

SÉPTIMA EDICIÓN



# ANÁLISIS DE SISTEMAS DISEÑO Y MÉTODOS

WHITTEN  
BENTLEY



Mc  
Graw  
Hill





# Análisis de sistemas: diseño y métodos



# Jeffrey L. Whitten Lonnie D. Bentley

*Purdue University  
West Lafayette, IN*

*Con la contribución de Gary Randolph,  
Purdue University*

## SÉPTIMA EDICIÓN

### Revisión técnica:

**Mayra Rico Valdovinos**  
*Profesora de cátedra de  
Organización computacional,  
Instituto Tecnológico y  
de Estudios Superiores  
de Monterrey,  
Campus Ciudad de México*

**Miguel Alejandro Orozco Malo**  
*Profesor de cátedra de Análisis y  
diseño de algoritmos,  
Instituto Tecnológico y  
de Estudios Superiores  
de Monterrey,  
Campus Ciudad de México*



**Director Higher Education:** Miguel Ángel Toledo Castellanos  
**Director editorial:** Ricardo Alejandro del Bosque Alayón  
**Editor sponsor:** Jesús Mares Chacón  
**Editor de desarrollo:** Edmundo Carlos Zúñiga Gutiérrez  
**Supervisor de producción:** Zeferino García García  
**Traductores:** M. Elizabeth Treviño R., Raúl Arrijo Juárez y Jorge L. Blanco y Correa M.

**ANÁLISIS DE SISTEMAS: DISEÑO Y MÉTODOS**  
**Séptima edición**

Prohibida la reproducción total o parcial de esta obra,  
por cualquier medio, sin la autorización escrita del editor.



DERECHOS RESERVADOS © 2008 respecto a la tercera edición en español por  
McGRAW-HILL/INTERAMERICANA EDITORES, S.A. de C.V.

*A Subsidiary of The McGraw-Hill Companies, Inc.*

Prolongación Paseo de la Reforma 1015, Torre A,  
Piso 17, Colonia Desarrollo Santa Fe,  
Delegación Álvaro Obregón,  
C.P. 01376, México, D.F.

Miembro de la Cámara Nacional de la Industria Editorial Mexicana, Reg. Núm. 736

**ISBN-10: 970-10-6614-6**  
**ISBN-13: 978-970-10-6614-0**  
(ISBN: 970-10-4283-2 anterior)

Traducido de la séptima edición en inglés de: **SYSTEMS ANALYSIS & DESIGN METHODS**  
Copyright © MMVII by The McGraw-Hill Companies, Inc. All rights reserved.

0-07-305233-7

0123456789

09765432108

Impreso en México

*Printed in Mexico*

# Dedicatoria

Para mi adorable esposa Cheryl y mis hijos Robert, Heath y Coty. Para mi coautor y buen amigo Jeff y nuestros veinte años de escribir juntos.

–Lonnie

Para mi padre. Tú me inculcaste la ética laboral, la perseverancia y la curiosidad de conocimiento que ha hecho posible este libro.

–Jeff

## > Público objetivo

*Análisis de sistemas: diseño y métodos*, séptima edición, pretende apoyar uno o más cursos sobre desarrollo de sistemas de información. Estos cursos normalmente son impartidos en los programas de Sistemas de Información y cursos optativos de programas de Administración a partir del segundo año, o en cursos de posgrado.

Recomendamos que los alumnos tomen un curso básico de computación (sistemas de información) antes de utilizar este texto. Aunque no sea requerido o se dé por supuesto, un curso sobre programación puede mejorar considerablemente la experiencia de aprendizaje que proporciona este libro de texto.

## > Por qué escribimos este libro

Más que nunca, los estudiantes de hoy están “orientados al consumidor”, debido, en parte, a la cambiante economía mundial, que promueve la calidad, la competencia y la actualización profesional. Ellos esperan salir de un curso con más que un grado y una promesa de que algún día apreciarán lo que han aprendido. Quieren “practicar” la aplicación de los conceptos, no sólo estudiar las aplicaciones de conceptos. Escribimos este libro: 1) para equilibrar la comprensión de conceptos, herramientas, técnicas y su aplicación; 2) proporcionar la mayor cantidad de ejemplos posible de diseño y análisis de sistemas disponibles en cualquier libro, y 3) equilibrar la comprensión de métodos clásicos con los métodos modernos de desarrollo de sistemas de información. Hemos escrito este libro en un tono de diálogo. Este método (y los diversos ejemplos) proporcionan un texto integral que conecta con el alumno a lo largo del proceso de aprendizaje. Además, nuestra meta es servir al lector al proporcionar referencias profesionales sobre las mejores prácticas actuales, que le sirvan más adelante. Esta nueva edición de *Análisis de sistemas: diseño y métodos* ofrece una gran cantidad de ejemplos de productos de análisis y diseño de sistemas disponibles en el mercado actual para apoyar la práctica del aprendizaje.

## > Cambios en la séptima edición

- **Reorganización para una mayor claridad:** Algunos capítulos han sido reorganizados internamente. Por ejemplo, el capítulo 8, en respuesta a comentarios de la revisión, pasó por una extensa reorganización. También el análisis sobre el desarrollo iterativo fue movido al capítulo 3 para colocarlo con los conceptos relacionados de metodología.
- **Ampliación del análisis de factibilidad:** El análisis de factibilidad ahora incluye la factibilidad legal y cultural (o política), así como nuestras cuatro pruebas tradicionales de factibilidad (operacional, económica, programa y técnica).
- **Uso de diagramas de contexto:** Incluso mientras continúa la migración del uso de los diagramas de flujo de datos hacia los diagramas UML, el diagrama de contexto continúa siendo importante como herramienta para entender el ámbito del sistema. Ha sido agregado a las herramientas utilizadas en el capítulo 4 y puede ser utilizado en el salón de clases como la primera tarea del modelado.
- **Referencias de tecnologías actualizadas:** Las extensas referencias para ejemplificar tecnologías han continuado en la séptima edición y han sido actualizadas para reflejar los cambios tecnológicos, actualizaciones de versiones, así como fusiones y adquisiciones de compañías de tecnología.
- **Revisión del caso SoundStage:** El caso SoundStage ha sido condensado, cambiado de un formato de diálogo a un formato de narrativa y ha sido integrado en la apertura de cada capítulo. El caso SoundStage presenta la perspectiva de un analista en sistemas recién graduado en su primera asignación, el caso SoundStage introduce brevemente los conceptos enseñados en cada capítulo y subraya su importancia en un proyecto de sistemas.

## > Código de pantallas

En esta séptima edición se ha adoptado un código de pantallas (tonos progresivos del gris al negro) para adaptar el sistema utilizado en la obra *Framework for Information Systems Architecture* de Zachman. La matriz de bloques de construcción de sistemas utiliza estos elementos para presentar conceptos recurrentes. Los modelos de sistemas refuerzan así estos conceptos con el uso consistente de las pantallas.



## > Organización

*Análisis de sistemas: diseño y métodos*, séptima edición, está dividido en tres partes. La organización del texto es lo suficientemente flexible para permitir que los instructores omitan o retomen capítulos de acuerdo con lo que decidan que es más importante para su audiencia. Se hicieron todos los esfuerzos posibles para hacer independientes los capítulos uno de otro y para ayudar a retomar la secuencia del material; incluso al grado de reintroducir conceptos y terminología específica.

En la Parte Uno, “El contexto de proyectos de desarrollo de sistemas”, se presenta el escenario y los procesos de desarrollo de sistemas de información. En los capítulos 1 al 3, al estudiante, se le presentan los analistas de sistemas, otros miembros del equipo del proyecto (incluidos usuarios y administradores), componentes de sistemas de información (basado en el marco de referencia de Zachman), y el ciclo de vida de desarrollo de sistemas contemporáneos. La Parte Uno puede ser cubierta relativamente rápido.

En la Parte Dos, “Métodos de análisis de sistemas”, se abarcan las actividades del ciclo de vida del front-end, las herramientas y técnicas para analizar problemas de negocios, especificación de requerimientos de negocios para un sistema de información y propuesta de solución de negocios y sistemas. Los capítulos 4 al 9 cubren los temas de requerimientos, casos de uso, modelado de datos con diagramas entidad-relación, modelado de procesos con diagramas de flujo de datos, e identificación de soluciones y propuesta de sistemas.

En la Parte Tres, “Métodos de diseño de sistemas”, se abarcan las actividades intermedias del ciclo de vida, herramientas y técnicas. En los capítulos del 10 al 15 se incluyen los temas del diseño general y detallado, con un énfasis particular en la arquitectura de aplicaciones, el desarrollo rápido y el desarrollo de prototipos, el diseño externo (entradas, salidas e interfaces), y el diseño interno (por ejemplo, bases de datos e ingeniería de software).

### Marco de trabajo de sistemas de información

Las pantallas y otros recursos se utilizan consistentemente a lo largo del marco de trabajo del texto para presentar los conceptos recurrentes.



representa datos y/o conocimiento



representa procesos



representa comunicación interface

Estamos en deuda con muchos individuos que contribuyeron al desarrollo de esta edición:

Grant Alexander, *Northeastern Oklahoma State University*  
 Richard J. Averbeck, *DeVry Institutes*  
 Emerson (Bill) Bailey, *Park University*  
 Jack Briner, *Charleston Southern University*  
 Jimmie Carraway, *Old Dominion University*  
 Casey Cegielski, *Auburn University*  
 Minder Chen, *George Mason University*  
 Glenn Dietrich, *University of Texas-San Antonio*  
 Dorothy Dologite, *Baruch College, CUNY*  
 Tom Erickson, *University of Virginia's Virginia Center for Continuing and Professional Education*  
 Bob Kilmer, *Messiah College*  
 Avram Malkin, *DeVry College of Technology*  
 Dat-Dao Nguyen, *California State University-Northridge*  
 Parag C. Pendharkar, *Penn State University*  
 Leah Pietron, *University of Nebraska-Omaha*  
 Charlene Riggle, *University of South Florida-Sarasota/Manatee*

Se extiende un agradecimiento especial a los siguientes participantes de grupos muestra:

Jeffrey Parsons, *Memorial University of Newfoundland*  
 Parag C. Pendharkar, *Penn State University*  
 Carl Scott, *University of Houston*  
 Ron Thompson, *Wake Forest University*  
 Steve Walczak, *Colorado University-Denver*

También estamos en deuda con muchos individuos que contribuyeron al desarrollo de las ediciones previas de este texto.

Jeanne M. Alm, *Moorhead State University*  
 Charles P. Bilbrey, *James Madison University*  
 Ned Chapin, *California State University-Hayward*  
 Carol Clark, *Middle Tennessee State University*  
 Gail Corbitt, *California State University-Chico*  
 Larry W. Cornwell, *Bradley University*  
 Barbara B. Denison, *Wright State University*  
 Linda Duxbury, *Carleton University*  
 Dana Edberg, *University of Nevada-Reno*  
 Craigh W. Fisher, *Marist College*  
 Raoul J. Freeman, *California State University-Dominguez Hills*  
 Dennis D. Gagnon, *Santa Barbara City College*  
 Abhijit Gopal, *University of Calgary*

Patricia J. Guinan, *Boston University*  
 Bill C. Hardgrave, *University of Arkansas-Fayetteville*  
 Alexander Hars, *University of Southern California*  
 Richard C. Housley, *Golden Gate University*  
 Constance Knapp, *Pace University*  
 Riki S. Kuchek, *Orange Coast College*  
 Thom Luce, *Ohio University*  
 Charles M. Lutz, *Utah State University*  
 Ross Malaga, *University of Maryland-Baltimore County*  
 Chip McGinnis, *Park College*  
 William H. Moates, *Indiana State University*  
 Ronald J. Norman, *San Diego State University*  
 Charles E. Paddock, *University of Nevada-Las Vegas*  
 June A. Parsons, *Northern Michigan University*  
 Harry Reif, *James Madison University*  
 Gail L. Rein, *SUNY-Buffalo*  
 Rebecca H. Rutherford, *Southern College of Technology*  
 Craig W. Slinkman, *University of Texas-Arlington*  
 John Smiley, *Holy Family College*  
 Mary Thurber, *Northern Alberta Institute of Technology*  
 Jerry Tillman, *Appalachian State University*  
 Jonathan Trower, *Baylor University*  
 Margaret S. Wu, *University of Iowa*  
 Jacqueline E. Wyatt, *Middle Tennessee State University*  
 Vincent C. Yen, *Wright State University*  
 Ahmed S. Zaki, *College of William and Mary*

Finalmente, reconocemos las contribuciones, aliento y paciencia del personal en McGraw-Hill. Un agradecimiento especial a Brent Gordon, editor, Paul Ducham, editor *sponsor*, Trina Hauger, editor de desarrollo; Greta Kleinert, gerente de marketing, Kristin Bradley, gerente de proyecto y Kami Carter, diseñador. También agradecemos a Judy Kausal, coordinadora de investigación de fotografía, Michael McCormick, supervisor de producción; Greg Bates, productor de medios y Rose Range, coordinadora de suplementos.

A todos ustedes que utilizaron nuestras ediciones anteriores, gracias por su continuo apoyo. Para aquellos que utilizan el texto por primera vez, esperamos que encuentren una diferencia en él. Esperamos ansiosamente sus reacciones, comentarios y sugerencias.

Jeffrey L. Whitten  
 Lonnie D. Bentley

# Síntesis del contenido

*Prefacio vi*

## PARTE UNO

### El contexto de proyectos de desarrollo de sistemas 3

- 1 El contexto de métodos de análisis y diseño de sistemas 5
- 2 Componentes de sistemas de información 23
- 3 Desarrollo de sistemas de información 47

## PARTE DOS

### Métodos de análisis de sistemas 99

- 4 Análisis de sistemas 101
- 5 Técnicas de exploración de hechos para identificación los requerimientos 149
- 6 Modelado de requerimientos del sistema con los casos de uso 185

- 7 Modelado y análisis de datos 211
- 8 Modelado de procesos 257
- 9 Análisis de factibilidad y propuesta del sistema 311

## PARTE TRES

### Métodos de diseño de sistemas 341

- 10 Diseño de sistemas 343
- 11 Modelado y arquitectura de la aplicación 373
- 12 Diseño de bases de datos 417
- 13 Diseño de salidas y elaboración de prototipos 451
- 14 Diseño de entradas y elaboración de prototipos 483
- 15 Diseño de la interfaz del usuario 515

*Créditos de fotografías 548*

*Glosario/Índice 549*



# Contenido

Prefacio vi

## PARTE UNO

El contexto de proyectos de desarrollo de sistemas 3

### 1 EL CONTEXTO DE MÉTODOS DE ANÁLISIS Y DISEÑO DE SISTEMAS 5

Introducción 6  
Marco de referencia para análisis y diseño de sistemas 6  
Los jugadores y los involucrados en el sistema 7

- Propietarios de sistemas* 7
- Usuarios de sistemas* 7
- Diseñadores de sistemas* 10
- Constructores de sistemas* 10
- Analistas de sistemas* 11
- Proveedores de servicio externo* 16
- El administrador de proyectos* 16

### 2 COMPONENTES DE SISTEMAS DE INFORMACIÓN 23

Introducción 24  
El producto (sistemas de información) 24  
Un marco de trabajo para la arquitectura de sistemas de información 26

- Componentes de CONOCIMIENTO* 27
- Componentes de PROCESO* 31
- Componentes de COMUNICACIONES* 35

Tecnologías de red y componentes de sistemas de información 38

### 3 DESARROLLO DE SISTEMAS DE INFORMACIÓN 47

Introducción 48  
El proceso de desarrollo de sistemas 48

- Modelo de Madurez de la Capacidad* 49
- Ciclo de vida frente a metodología* 50
- Principios fundamentales para el desarrollo de sistemas* 52

Proceso de desarrollo de sistemas 56

- ¿De dónde surgen los proyectos de desarrollo de sistemas?* 57

*Las fases del proyecto FAST* 57  
*Actividades transversales del ciclo de vida* 68  
*Desarrollo secuencial o iterativo* 69

Rutas y estrategias alternativas 72

- Estrategia de desarrollo basado en modelos* 74
- Estrategia de desarrollo rápido de aplicaciones* 78
- Estrategia de implantación de paquetes de aplicación comercial* 80
- Estrategias híbridas* 84
- Mantenimiento del sistema* 84

Herramientas y tecnología automatizada 87

- Ingeniería de sistemas asistida por computadora* 88
- Ambientes de desarrollo de aplicación* 89
- Administradores de proceso y proyecto* 91

## PARTE DOS

Métodos de análisis de sistemas 99

### 4 ANÁLISIS DE SISTEMAS 101

Introducción 102  
¿Qué es un análisis de sistemas? 102  
Enfoques de análisis de sistemas 103

- Enfoques de análisis basados en modelos* 103
- Enfoques de análisis de sistemas acelerados* 105
- Métodos para identificación de requerimientos* 107
- Métodos de rediseño de procesos de negocios* 108
- Estrategias de análisis de sistemas FAST* 108

Fase de definición de alcance 109

- Tarea 1.1: Identificar problemas y oportunidades básicas* 111
- Tarea 1.2: Negociar el alcance base* 114
- Tarea 1.3: Evaluar el beneficio del proyecto base* 115
- Tarea 1.4: Desarrollar un programa y presupuesto base* 115
- Tarea 1.5: Comunicar el plan del proyecto* 115

Fase de análisis del problema 116

- Tarea 2.1: Entender el dominio del problema* 117
- Tarea 2.2: Analizar problemas y oportunidades* 122
- Tarea 2.3: Analizar los procesos del negocio* 122
- Tarea 2.4: Establecer objetivos de mejora del sistema* 124

*Tarea 2.5: Actualizar o refinar el plan del proyecto* 125  
*Tarea 2.6: Comunicar resultados y propuestas* 125

Fase de análisis de requerimientos 127

*Tarea 3.1: Identificar y expresar los requerimientos del sistema* 127

*Tarea 3.2: Priorizar los requerimientos del sistema* 130

*Tarea 3.3: Actualizar o refinar el plan del proyecto* 130

*Tarea 3.4: Comunicar la definición de requerimientos* 131

*Manejo de requerimientos permanentes* 131

Fase de diseño lógico 131

*Tarea 4.1a: Requerimientos funcionales de estructura* 133

*Tarea 4.1b: Requerimientos funcionales del prototipo (alternativa)* 134

*Tarea 4.2: Validar requerimientos funcionales* 134

*Tarea 4.3: Definir casos de prueba de aceptación* 134

Fase de análisis de decisión 134

*Tarea 5.1: Identificar soluciones alternativas* 136

*Tarea 5.2: Analizar soluciones alternativas* 137

*Tarea 5.3: Comparar soluciones alternativas* 139

*Tarea 5.4: Actualizar el plan del proyecto* 139

*Tarea 5.5: Recomendar una solución del sistema* 139

## 5 TÉCNICAS DE EXPLORACIÓN DE HECHOS PARA IDENTIFICACIÓN DE LOS REQUERIMIENTOS 149

Introducción 150

Una introducción a la identificación de requerimientos 150

Proceso de identificación de requerimientos 152

*Identificación del problema y análisis* 152

*Identificación de los requerimientos* 154

*Requerimientos de documentación y de análisis* 154

*Administración de los requerimientos* 156

Técnicas de exploración 157

*Muestreo de la documentación, los formatos y los archivos existentes* 157

*Investigación y visitas al sitio* 159

*Observación del ambiente de trabajo* 160

*Cuestionarios* 162

*Entrevistas* 164

*Cómo conducir una entrevista* 166

*Elaboración de prototipos de identificación* 170

*Planeación conjunta de requerimientos* 171

Una estrategia de exploración 176

## 6 MODELADO DE REQUERIMIENTOS DEL SISTEMA CON LOS CASOS DE USO 185

Introducción 186

Una introducción a la modelación de casos de uso 186

Conceptos de sistemas en la modelación de casos de uso 188

*Los casos de uso* 188

*Actores* 189

*Relaciones* 190

El proceso de la modelación de los casos de uso para los requerimientos 193

*Paso 1: Identificar a los actores de negocios* 193

*Paso 2: Identificar los casos de uso para los requerimientos de negocios* 194

*Paso 3: Construir el diagrama de modelos de casos de uso* 196

*Paso 4: Narraciones de los casos de uso para los requerimientos de documentos para los negocios* 198

Los casos de uso y la administración de proyectos 202

*Cómo jerarquizar y evaluar los casos de uso* 202

*Identificación de las dependencias de los casos de uso* 203

## 7 MODELADO Y ANÁLISIS DE DATOS 211

Introducción 212

¿Qué es el modelado de datos? 212

Conceptos de sistemas para el modelado de datos 213

*Las entidades* 213

*Los atributos* 214

*Las relaciones* 216

El proceso del modelado lógico de datos 225

*El modelado estratégico de datos* 225

*El modelado de datos durante el análisis de sistemas* 227

*Mirando hacia delante en el diseño de sistemas* 228

*Las herramientas automatizadas para el modelado de datos* 228

Cómo construir modelos de datos	230
<i>El descubrimiento de las entidades</i>	231
<i>El modelo de datos de contexto</i>	232
<i>El modelo de datos basado en claves</i>	234
<i>Las jerarquías generalizadas</i>	237
<i>El modelo integral de datos</i>	237

Cómo analizar el modelo de datos	240
<i>¿Qué es un buen modelo de datos?</i>	240
<i>El análisis de datos</i>	241
<i>Ejemplo de normalización</i>	241

Cómo mapear los requerimientos de datos a las localidades	248
---	-----

## 8 MODELADO DE PROCESOS 257

Introducción	258
Una introducción a la modelación de procesos	258
Los conceptos de sistemas para la modelación de procesos	261
<i>Agentes externos</i>	261
<i>Los almacenes de datos</i>	262
<i>Conceptos de procesos</i>	263
<i>Flujos de datos</i>	267
El proceso de modelación lógica de procesos	276
<i>Planeación estratégica de sistemas</i>	276
<i>Modelado de procesos para el rediseño de los procesos de negocio</i>	276
<i>Modelado de procesos durante el análisis de los sistemas</i>	277
<i>Mirando hacia delante al diseño de sistemas</i>	279
<i>Descubrimiento de hechos y recopilación de información para el modelado de procesos</i>	279
<i>La ingeniería de sistemas asistida por computadora (CASE) para la modelación de procesos</i>	279
Cómo construir modelos de procesos	280
<i>El diagrama de flujo de datos de contexto</i>	280
<i>El diagrama de descomposición funcional</i>	281
<i>Lista de respuesta o los eventos o casos de uso</i>	283
<i>Los diagramas de descomposición de eventos</i>	284
<i>Los diagramas de eventos</i>	287
<i>El (los) diagrama(s) de sistema</i>	289
<i>Los diagramas elementales</i>	291
<i>Completando la especificación</i>	291
Sincronización de los modelos de sistemas	301
<i>Sincronización de los modelos de datos y de procesos</i>	301
<i>Distribución de procesos</i>	302

## 9 ANÁLISIS DE FACTIBILIDAD Y PROPUESTA DEL SISTEMA 311

Introducción	312
Análisis de factibilidad y la propuesta de sistema	312
<i>Análisis de factibilidad: una estrategia de compromiso creciente</i>	312
<i>Análisis de sistemas: control de la definición del alcance</i>	314
<i>Análisis de sistemas: control del análisis del problema</i>	314
<i>Diseño de sistemas: control de análisis de decisiones</i>	314
Seis pruebas de factibilidad	315
<i>Factibilidad operativa</i>	315
<i>Factibilidad cultural (o política)</i>	315
<i>Factibilidad técnica</i>	316
<i>Factibilidad del calendario</i>	316
<i>Factibilidad económica</i>	317
<i>Factibilidad legal</i>	317
<i>Resultado final</i>	317
Técnicas de análisis de costo-beneficio	317
<i>¿Cuánto costará el sistema?</i>	317
<i>¿Cuáles beneficios proporcionará el sistema?</i>	318
<i>¿Es rentable el sistema propuesto?</i>	320
Análisis de factibilidad de las soluciones alternativas del sistema	324
<i>Matriz de soluciones alternativas del sistema</i>	324
<i>Matriz de análisis de factibilidad de soluciones</i>	327
La propuesta del sistema	329
<i>Informe escrito</i>	329
<i>Presentación formal</i>	331

## PARTE TRES

### Métodos de diseño de sistemas 341

## 10 DISEÑO DE SISTEMAS 343

Introducción	344
¿Qué es el diseño de sistemas?	344
Estrategias del diseño de sistemas	344
<i>Estrategias basadas en modelos</i>	345
<i>Desarrollo rápido de aplicaciones</i>	349
<i>Estrategias de diseño de sistemas FAST</i>	351
Diseño de sistemas para desarrollo en la organización: la solución de “construcción”	351
<i>Tarea 5.1: Diseño de la arquitectura de la aplicación</i>	351

- Tarea 5.2: Diseño de las bases de datos del sistema* 355
- Tarea 5.3: Diseño de la interfaz del sistema* 355
- Tarea 5.4: Especificaciones de diseño del paquete* 357
- Tarea 5.5: Actualización del plan del proyecto* 358

Diseño de sistemas para integrar software comercial:  
La solución de “compra” 358

- Tarea 4.1: Investigación de criterios y opciones técnicas* 360
- Tarea 4.2: Solicitar propuestas o cotizaciones a los proveedores* 361
- Tarea 5A.1: Validación de las afirmaciones y desempeño de los proveedores* 363
- Tarea 5A.2: Evaluar y jerarquizar las propuestas de los proveedores* 363
- Tarea 5A.3: Otorgamiento del contrato y junta informativa a proveedores* 364
- Impacto de la decisión de compra en las fases restantes del ciclo de vida de sistemas* 364

## 11 MODELADO Y ARQUITECTURA DE LA APLICACIÓN 373

- Introducción 374
- La arquitectura de aplicación 374
- Diagramas de flujo de datos físicos 375
  - Los procesos físicos* 375
  - Los flujos de datos físicos* 379
  - Los agentes externos físicos* 379
  - Los almacenamientos de datos físicos* 379
- Arquitectura de tecnología de la información 381
  - Los sistemas distribuidos* 382
  - Arquitecturas de datos: bases de datos relacionales distribuidas* 392
  - Arquitectura de interfaces: entradas, salidas y middleware* 393
  - Arquitecturas de proceso: el ambiente de desarrollo de software* 398
- Estrategias de arquitecturas de aplicación para el diseño de sistemas 400
  - La estrategia de arquitectura de aplicación empresarial* 400
  - La estrategia táctica de arquitectura de aplicación* 401
- Modelando la arquitectura de aplicación de un sistema de información 401
  - Dibujo de diagramas de flujo de datos físicos* 402
  - Los prerrequisitos* 402
  - La arquitectura de red* 403
  - Distribución de datos y asignaciones de tecnología* 404

- Distribución de procesos y asignaciones de tecnología* 405
- Los límites entre persona/máquina* 408

## 12 DISEÑO DE BASES DE DATOS 417

- Introducción 418
- Archivos convencionales contra bases de datos 418
  - Pros y contras de los archivos convencionales* 418
  - Pros y contras de las bases de datos* 420
- Los conceptos de bases de datos para el analista de sistemas 420
  - Los campos* 421
  - Los registros* 421
  - Archivos y tablas* 422
  - Las bases de datos* 423
- Prerrequisito para el diseño de la base de datos: normalización 428
- Diseño convencional de archivos 429
- El diseño moderno de la base de datos 429
  - Metas y prerrequisitos del diseño de bases de datos* 430
  - El esquema de la base de datos* 430
  - Los datos y la integridad referencial* 435
  - Los roles* 438
  - Distribución y réplica de la base de datos* 438
  - Los prototipos de bases de datos* 439
  - Planificación de la capacidad de las bases de datos* 439
  - Generación de estructuras de bases de datos* 439

## 13 DISEÑO DE SALIDAS Y ELABORACIÓN DE PROTOTIPOS 451

- Introducción 452
- Conceptos y lineamientos del diseño de las salidas 452
  - La distribución y la audiencia de salidas* 452
  - Los métodos de implantación de las salidas* 455
- Cómo diseñar y desarrollar prototipos de las salidas 460
  - Las herramientas automatizadas para el diseño y el desarrollo de prototipos de las salidas* 460
  - Lineamientos del diseño de salidas* 461
  - El proceso del diseño de salida* 464
  - Las salidas basadas en la Web y el comercio electrónico* 472

## 14 DISEÑO DE ENTRADAS Y ELABORACIÓN DE PROTOTIPOS 483

- Introducción 484
- Conceptos y lineamientos de diseño de entradas 484
  - Captura de datos, entrada de datos y procesamiento de datos* 484
  - Métodos de entrada y su implementación* 487
  - Temas de usuario del sistema para diseño de entradas* 489
  - Controles internos: Edición de datos para entradas* 491
- Controles GUI para diseño de entradas 492
  - Controles GUI comunes para entradas* 494
  - Controles de entrada avanzados* 498
- Cómo diseñar y elaborar prototipos de entradas 500
  - Herramientas automatizadas para diseño de entradas y elaboración de prototipos* 500
  - Proceso de diseño de entradas* 501
  - Entradas basadas en la Web y el negocio electrónico (e-business)* 507

## 15 DISEÑO DE LA INTERFAZ DEL USUARIO 515

- Introducción 516
- Conceptos y lineamientos del diseño de la interfaz del usuario 516

- Tipos de usuarios de computadoras* 516
- Los factores humanos* 517
- Las directrices de la ergonomía* 518
- Terminología y tono del diálogo* 519

- Tecnología de la interfaz del usuario 520
  - Sistemas operativos y navegadores Web* 520
  - Monitor de pantalla* 520
  - Teclados y punteros* 521

- Estilos y consideraciones de la interfaz gráfica del usuario 521

- Ventanas y marcos* 522
- Las interfaces operadas por menú* 522
- Interfaces de comandos* 529
- Diálogos pregunta-respuesta* 531
- Consideraciones especiales para el diseño de la interfaz del usuario* 531

- Cómo diseñar y elaborar un prototipo para una interfaz del usuario 535

- Herramientas automatizadas para el diseño de la interfaz del usuario y la elaboración de prototipos* 536
- Proceso de diseño de la interfaz del usuario* 537
- Créditos de fotografías* 548
- Glosario/Índice* 549



# Análisis de sistemas: diseño y métodos



## El contexto de proyectos de desarrollo de sistemas

Éste es un libro práctico acerca de métodos de desarrollo de sistemas de información. Todas las empresas y las organizaciones desarrollan sistemas de información. Puede estar seguro de que usted tendrá algún papel en el análisis y diseño de esos sistemas, ya sea como cliente, usuario o desarrollador de los mismos. El análisis y diseño de sistemas trata acerca de la solución de problemas de negocios y de aplicaciones de cómputo. Los métodos que usted aprenderá en este texto pueden ser aplicados a una amplia diversidad de dominios de problemas, no sólo a los que incluyen computadoras.

Antes de comenzar, asumimos que usted ha completado un curso de introducción en sistemas de información basados en la computación. Muchos de ustedes también han completado uno o más cursos de programación (con el uso de tecnologías como *Access*, *Java*, *C/C++*, o *Visual Basic*). Eso será útil ya que el análisis y diseño de sistemas precede y/o se integra con esas actividades. Pero no se preocupe, revisaremos todos los principios necesarios en los que se basa el análisis y diseño de sistemas.

En la parte uno nos enfocamos en un aspecto general. Antes de que usted aprenda actividades, herramientas, técnicas, métodos y tecnología, tiene que entender el aspecto general. Cuando usted explore el contexto de análisis y

diseño de sistemas, presentaremos muchas ideas, herramientas y técnicas que son exploradas con mayor detalle posteriormente. Trate de tener eso en mente mientras explora el aspecto general.

El desarrollo de sistemas no es mágico. No hay secretos para el éxito ni herramientas, técnicas o métodos perfectos. Para estar seguros, hay habilidades que pueden ser dominadas. Pero la aplicación completa y consistente de esas habilidades es todavía un arte.

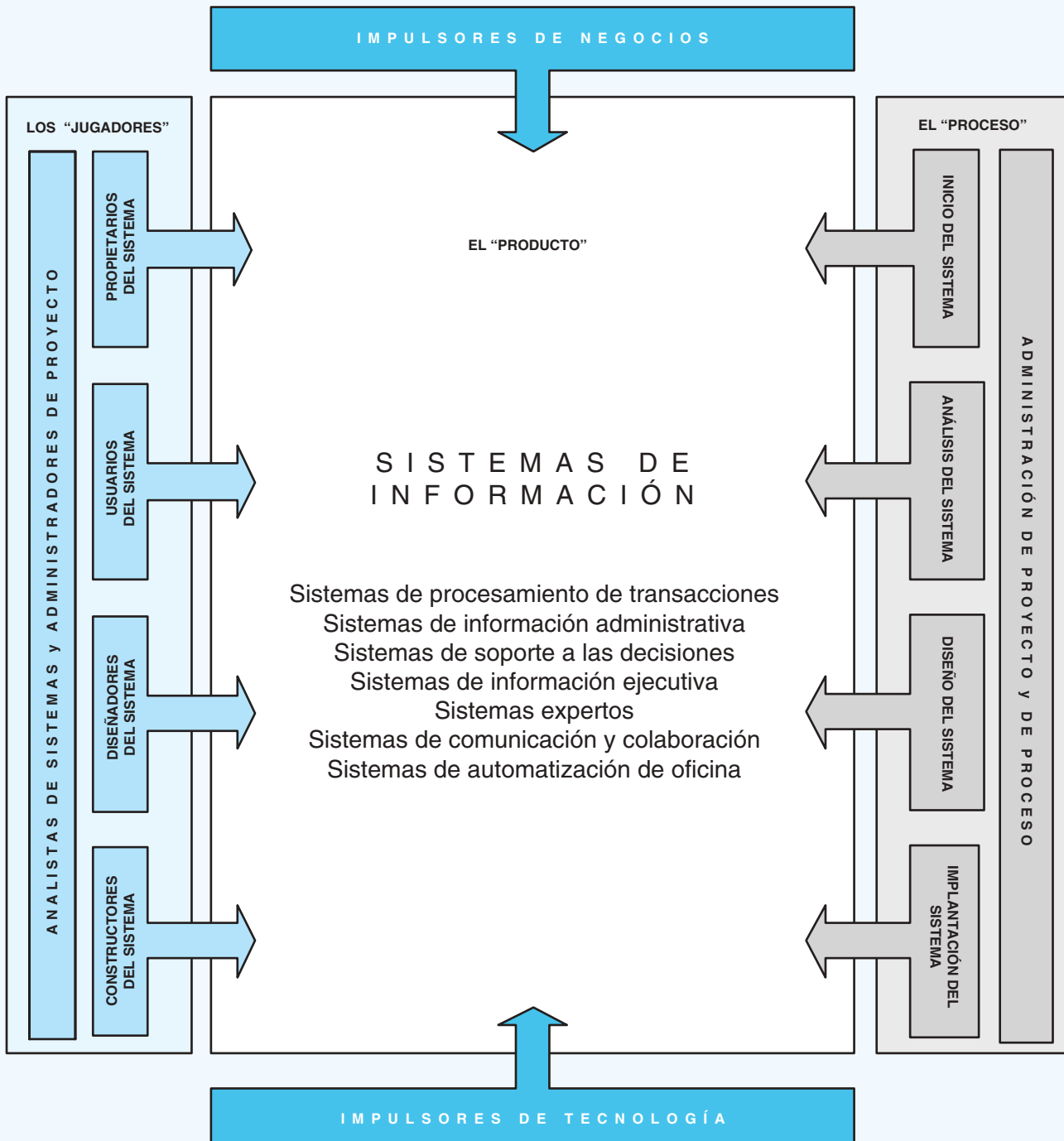
En la parte uno comenzamos con los conceptos fundamentales, filosofías y tendencias que proporcionan el contexto de análisis de sistemas y métodos de diseño; en otras palabras, ¡lo básico! Si usted entiende lo básico, será más capaz de aplicar, con confianza, las herramientas y técnicas que aprenderá en las partes dos y tres. También será capaz de adaptarse a nuevas situaciones y métodos.

Tres capítulos constituyen esta parte. En el capítulo 1, “El contexto de métodos de análisis y diseño de sistemas”, se presentan los *participantes* del análisis y diseño de sistemas con un énfasis especial en el analista de sistemas moderno como un facilitador del trabajo de sistemas. También aprenderá acerca de las relaciones entre los analistas de sistemas, usuarios finales, administradores y otros profesionales de sistemas de

información. Finalmente, aprenderá a prepararse para una carrera como analista (si ésa es su meta). Si no, usted entenderá la forma en que interactuará con este importante profesional.

En el capítulo 2, “Componentes de sistemas de información”, se presenta el *producto* que le enseñaremos a construir: *sistemas de información*. Específicamente, usted aprenderá a examinar los sistemas de información en términos de componentes comunes, CONOCIMIENTO, PROCESOS y COMUNICACIONES, cada uno desde la perspectiva de diferentes participantes o involucrados. Un marco de referencia de matriz visual le ayudará a organizar estos componentes o bloques de construcción para que usted los pueda ver aplicados en los capítulos subsecuentes.

En el capítulo 3, “Desarrollo de sistemas de información”, se presenta un proceso de alto nivel (es decir, general) para el desarrollo de sistemas de información. Esto es llamado *un ciclo de vida de desarrollo de sistemas*. Presentaremos el ciclo de vida en una forma en la que la mayoría de ustedes lo experimentará: *una metodología de desarrollo de sistemas*. Esta metodología será el contexto en el que usted aprenderá a utilizar y aplicar el análisis de sistemas y los métodos de diseño enseñados en el resto del texto.



**CAPÍTULO 1 PÁGINA DE INICIO** Cada capítulo en este texto comienza con una "página de inicio" similar a la de arriba. La página de inicio es como un mapa del capítulo, un marco visual de trabajo para el pensamiento de sistemas aplicable a ese capítulo. El capítulo 1 se enfoca en 1) los jugadores del juego de sistemas, 2) impulsores de negocios de interés para los jugadores de la organización, 3) impulsores y facilitadores de tecnología de interés para los jugadores técnicos y 4) el proceso utilizado para desarrollar sistemas. También examinaremos el papel crítico que juegan los analistas de sistemas al facilitar una comprensión de cómo se deben unir las cuatro perspectivas.

# 1

# El contexto de métodos de análisis y diseño de sistemas

## Panorámica y objetivos del capítulo

Éste es un libro acerca del análisis y diseño de sistemas como se aplican a los sistemas de información y a las aplicaciones de cómputo. Sin importar su ocupación o puesto en cualquier empresa, probablemente usted participará en un análisis y diseño de sistemas. Algunos de ustedes se volverán *analistas de sistemas*, los jugadores fundamentales en el análisis de sistemas y las actividades de diseño. El resto de ustedes trabajará *con* analistas de sistemas conforme los proyectos vayan y vengan en sus organizaciones. En este capítulo se presentan los sistemas de información desde cuatro perspectivas distintas. Usted comprenderá el contexto del análisis de sistemas y los métodos de diseño, donde podrá:

- Definir *sistema de información* y nombrar siete tipos de sistemas de información.
- Identificar distintos tipos de *involucrados* que utilicen o desarrollen sistemas de información y dar ejemplos de cada uno.
- Definir el papel único de los *analistas de sistemas* en el desarrollo de sistemas de información.
- Identificar aquellas *habilidades* necesarias para funcionar con éxito como analista de sistemas de información.
- Describir los *impulsores de negocios* actuales que influyen en el desarrollo de sistemas de información.
- Describir los *impulsores de tecnología* actuales que influyen en el desarrollo de sistemas de información.
- Describir brevemente un *proceso* simple para desarrollar sistemas de información.

## Introducción

Es la primera semana de trabajo de Bob Martínez como analista/programador. Recién salido de la universidad con un título en tecnología en sistemas de información de cómputo, está ansioso por trabajar con sistemas de información en el mundo real. Su empleador es SoundStage Entertainment Club, uno de los clubes de música y video de mayor crecimiento en Estados Unidos. SoundStage apenas comienza un trabajo de análisis y diseño de sistemas en una reingeniería de sus servicios de membresía de sistemas de información. Bob ha sido asignado al equipo de proyecto.

Esta mañana fue la junta de inicio para el proyecto, una reunión que incluyó al vicepresidente de servicios de membresía, al director del club de audio, al director del club de juegos, al director de marketing, al director de servicio al cliente y al director de las operaciones de almacén. Con esa alineación, Bob estuvo contento de mantenerse en silencio en la junta y de confiar en su jefa, Sandra Shepherd, una analista de sistemas *senior*. Él estaba sorprendido de lo bien que Sandra era capaz de hablar el lenguaje de cada uno de los participantes y explicar los planes para el nuevo sistema de información, en términos que ellos pudieran entender y con los beneficios que pudieran apreciar. Bob había pensado que al haber apenas salido de la universidad él sabría más acerca de la tecnología de punta que la mayoría de sus colaboradores. Pero Sandra parecía entenderlo todo acerca del comercio electrónico y el uso de tecnologías móviles, además de muchas cosas de las que Bob sólo tenía un vago conocimiento. Él hizo una nota para leer acerca de los sistemas ERP, cuando salieron en la discusión. Al final de la junta Bob tenía una nueva apreciación del puesto de analista de sistemas y de todas las cosas que aún tenía que aprender.

**sistema** Grupo de componentes interrelacionados que funcionan juntos para lograr un resultado deseado.

## Marco de referencia para análisis y diseño de sistemas

**sistema de información (IS)** Conjunto de personas, datos, procesos y **tecnología de la información** que interactúan para recopilar, procesar, guardar y proporcionar como salida la información necesaria para brindar soporte a una organización.

**tecnología de información (TI)** Término contemporáneo que describe la combinación de la tecnología de computadoras (hardware y software) con la de telecomunicaciones (redes de datos, imágenes y voz).

**sistema de procesamiento de transacciones (TPS)** Sistema de información en el que se capturan y procesan los datos relativos a transacciones de negocios.

**sistema de información administrativa (MIS)** Sistema de información que provee informes orientados a la administración basado en el procesamiento de las transacciones y operaciones de la organización.

Como el título lo sugiere, este es un libro acerca de *métodos de análisis y diseño de sistemas*. En este capítulo, presentaremos el tema por medio de un marco de referencia visual simple pero completo. Cada capítulo en este libro comienza con una *página de inicio* (vea la página 4) en la que se muestra rápida y visualmente qué aspectos del marco de referencia total analizaremos en el capítulo. Construiremos este marco de referencia visual lentamente sobre los primeros tres capítulos para evitar inundarlo tan pronto con demasiados detalles. A partir de ahí, cada capítulo resaltarán los aspectos del marco de referencia que son enseñados con mayor detalle dentro de ese capítulo.

Finalmente, este es un libro que trata acerca de “analizar” los requerimientos de negocios para los sistemas de información y “diseñar” los *sistemas de información* que satisfagan esos requerimientos de negocios. En otras palabras, el *producto* del análisis y diseño de sistemas es un sistema de información. Ese producto está representado visualmente en el marco de referencia visual como el rectángulo grande en el centro de la imagen.

Un **sistema** es un grupo de componentes interrelacionados que funcionan juntos para lograr un resultado deseado. Por ejemplo, usted puede ser propietario de un sistema de teatro en casa conformado por un aparato de DVD, un receptor, bocinas y el monitor.

Los **sistemas de información (IS)**, por sus siglas en inglés) en las organizaciones capturan y administran datos para producir información útil que respalda a una organización y sus empleados, clientes, proveedores y socios. Muchas organizaciones consideran que los sistemas de información son esenciales para su capacidad de competir u obtener una ventaja competitiva. La mayoría de las organizaciones se han percatado de que *todos* los trabajadores necesitan participar en el desarrollo de sistemas de información. Por tanto, el desarrollo de sistemas de información es un tema relevante para usted sin importar si estudia o no para convertirse en un profesional de sistemas de información.

Los sistemas de información vienen en todas formas y tamaños. Están tan entrelazados en la tela de los sistemas de negocios que respaldan que con frecuencia es difícil distinguir entre sistemas de negocios y sus sistemas de información de soporte. Basta con decir que los sistemas de información pueden ser clasificados de acuerdo con las funciones que atienden. Los **sistemas de procesamiento de transacciones (transaction processing systems, TPS)** procesan transacciones de negocios como pedidos, tarjetas de tiempo, pagos y reservaciones. Los **sistemas de información administrativa (management information systems, MIS)** utilizan los datos de transacción para producir información necesaria por los administradores para dirigir el negocio.

Los **sistemas de soporte de decisiones (decision support systems, DSS)** ayudan a diversos tomadores de decisiones a identificar y elegir entre opciones o decisiones. Los **sistemas de información ejecutiva (executive information systems, EIS)** están adaptados a las necesidades de información únicas de los ejecutivos que planean el negocio y evalúan el desempeño contra esos planes. Los **sistemas expertos** capturan y reproducen el conocimiento de un solucionador de problemas experto o un tomador de decisiones y luego simulan el “pensamiento” de ese experto. Los **sistemas de comunicación y colaboración** resaltan la comunicación y la colaboración entre las personas, tanto internas como externas de la organización. Finalmente, los **sistemas de automatización de oficina** ayudan a los empleados a crear y compartir documentos que respaldan las actividades diarias de oficina.

Como se ilustró en la página de inicio del capítulo, los sistemas de información pueden ser vistos desde diversas perspectivas, que incluyen:

- Los jugadores en el sistema de información (el “equipo”).
- Los impulsores de negocios que influyen en el sistema de información.
- Los impulsores de tecnología utilizados por el sistema de información.
- El proceso utilizado para desarrollar el sistema de información.

Examinemos cada una de estas perspectivas en las secciones restantes del capítulo.

## Los jugadores y los involucrados en el sistema

Supongamos que usted está en una posición para ayudar a construir un sistema de información. ¿Quiénes son los **involucrados** en este sistema? Los involucrados en los sistemas de información pueden ser clasificados ampliamente en los cinco grupos que se muestran en el lado izquierdo de la figura 1.1. Nótese que cada grupo de involucrados tiene una perspectiva diferente del mismo sistema de información. El *analista de sistemas* es un involucrado único en la figura 1.1. El analista de sistemas sirve como un facilitador o instructor, que construye puentes entre las brechas de comunicación que pueden desarrollarse en forma natural entre los propietarios del sistema no técnico y los usuarios así como los diseñadores y constructores del sistema técnico.

Todos los involucrados anteriores tienen una cosa en común: son lo que el departamento de trabajo estadounidense llama **trabajadores de la información**. La vida de los trabajadores de la información depende de las decisiones tomadas con base en la información. Actualmente, más del 60 por ciento de la fuerza laboral estadounidense participa en la producción, distribución y uso de la información. Examinemos los cinco grupos de trabajadores de la información con mayor detalle.

Analícemos brevemente las perspectivas de cada grupo. Pero antes, debemos señalar que estos grupos en realidad definen “papeles” jugados en el desarrollo de sistemas. En la práctica, cualquier persona puede tener más de uno de estos papeles. Por ejemplo, un propietario de sistemas podría también ser un usuario de sistemas. En forma similar, un analista de sistemas puede también ser visto como un diseñador de sistemas y un diseñador de sistemas puede ser visto como un constructor de sistemas. Cualquier combinación puede funcionar.

### > Propietarios de sistemas

Para cualquier sistema de información, grande o pequeño, habrá uno o más **propietarios del sistema**. Los propietarios del sistema comúnmente vienen de las filas de los administradores. Para los sistemas de información de medianos a grandes, los propietarios del sistema son, en general, administradores medios o ejecutivos. Para los sistemas pequeños, los propietarios del sistema pueden ser administradores medios o supervisores. Los propietarios del sistema tienden a estar involucrados en la línea de fondo; ¿Cuánto costará el sistema? ¿Cuánto valor o qué beneficios el sistema retornarán al negocio? El valor y los beneficios pueden ser medidos de distintas formas, como se señala en la lista de comprobación del margen.

### > Usuarios de sistemas

Los **usuarios del sistema** constituyen la vasta mayoría de los trabajadores de la información en cualquier sistema de información. A diferencia de los propietarios del sistema, los usuarios del sistema tienden a estar menos preocupados con los costos y los beneficios del

---

**sistema de soporte de decisiones (DSS)** Sistema de información que ayuda a identificar oportunidades de toma de decisiones o proporciona información que ayuda a tomarlas.

---

**sistema de información ejecutiva (EIS)** Sistema de información que brinda soporte a las necesidades de planeación y evaluación de los administradores de nivel ejecutivo.

---

**sistema experto** Sistema de información en el cual se captura la experiencia de los expertos humanos y luego simula esa experiencia para beneficio de quienes no son expertos.

---

**sistema de comunicación y colaboración** Sistema de información que posibilita la comunicación más efectiva entre los empleados, socios, clientes y proveedores, para mejorar su capacidad de colaboración.

---

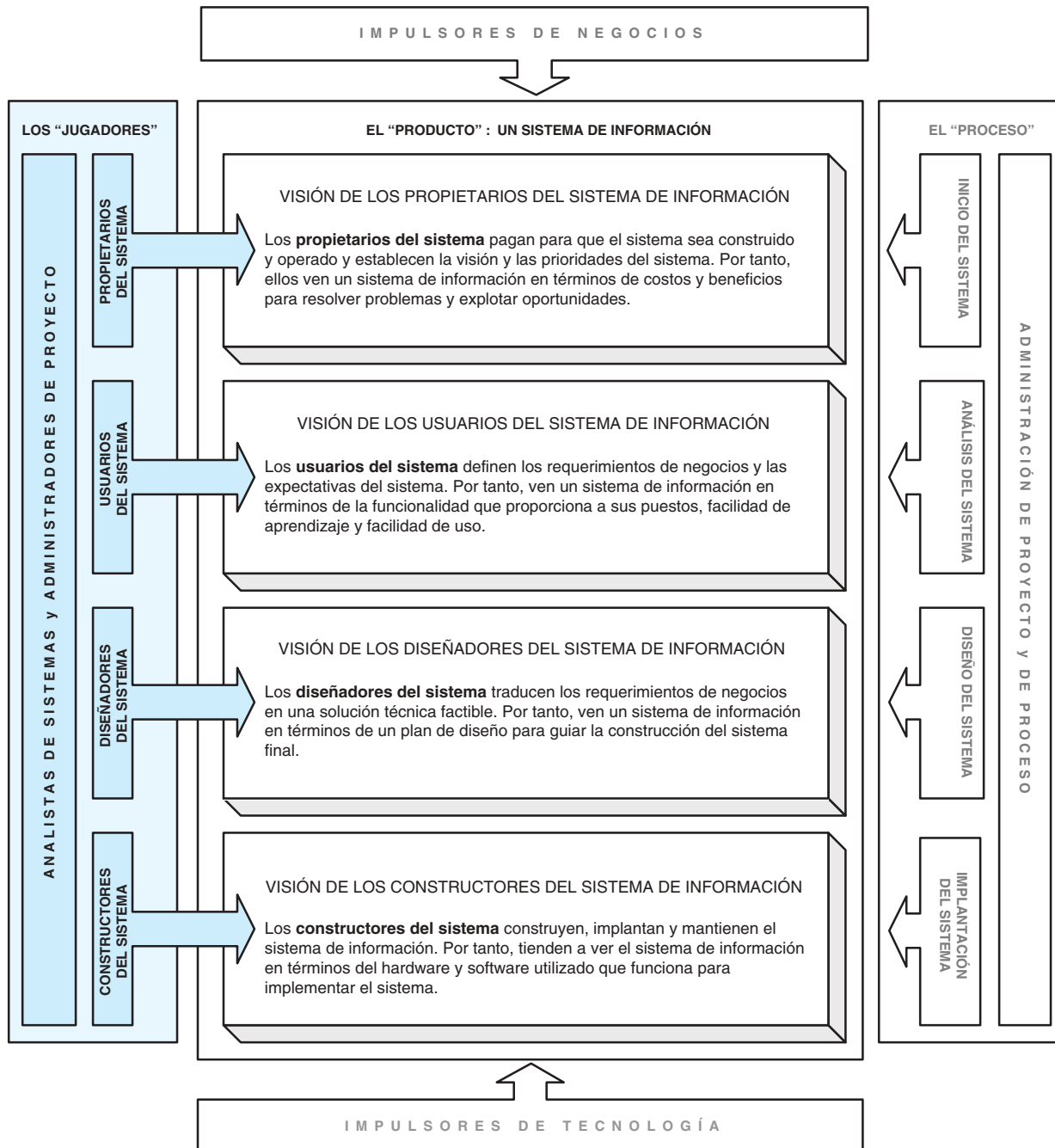
**sistema de automatización de oficina** Sistema de información que brinda soporte a la amplia gama de actividades de oficina de los negocios para mejorar el flujo de trabajo entre los empleados.

---

**involucrado** Toda persona que tiene interés en un sistema de información existente o propuesto. Los involucrados y grupos de interés pueden ser trabajadores técnicos y no técnicos. También puede tratarse de trabajadores internos y externos.

---

**trabajador de la información** Toda persona cuyo trabajo entraña la creación, recopilación, procesamiento, distribución y uso de información.



**FIGURA 1.1** Perspectiva de los involucrados sobre un sistema de información

**propietario del sistema**  
 Patrocinador y representante ejecutivo de un sistema de información, que generalmente se encarga del financiamiento del proyecto y del desarrollo, operación y mantenimiento del sistema de información.

sistema. En lugar de eso, como se ilustra en la figura 1.1, están preocupados por la funcionalidad que el sistema provee a sus puestos y, la facilidad de aprendizaje y uso del sistema. Aunque los usuarios se han vuelto más cultos en cuanto a la tecnología con el paso de los años, su principal preocupación es que el trabajo se realice. En consecuencia, las discusiones con la mayoría de los usuarios necesitan mantenerse al nivel de los requerimientos de negocios en contraste con el nivel de requerimientos técnicos. Gran parte de este libro está dedicada a enseñarle la forma de identificar y comunicar eficazmente los requerimientos de negocios para un sistema de información.

Existen muchas clases de usuarios de sistemas. Cada clase debe participar directamente en cualquier proyecto de desarrollo de sistemas de información que los afecte. Analicemos brevemente estas clases.



**Usuarios internos del sistema** Los usuarios internos del sistema son empleados del negocio para el cual se construyen la mayoría de los sistemas de información. Los usuarios internos constituyen el mayor porcentaje de usuarios de sistemas de información en la mayoría de las empresas. Los ejemplos incluyen:

- **Trabajadores de oficina y de servicio.** Desempeñan la mayoría del proceso de transacción diaria en el negocio promedio. Procesan pedidos, facturas, pagos y demás. Mecanografían y archivan la correspondencia. Llenan pedidos en el almacén. Fabrican productos en el piso del taller. La mayoría de los datos fundamentales en cualquier negocio son capturados o creados por estos trabajadores, muchos de los cuales desempeñan una labor manual además del proceso de datos. Los sistemas de información que se dirigen a estos trabajadores, generalmente se enfocan en la velocidad y precisión del proceso de transacciones.
- **Personal técnico y profesional.** Consiste en gran medida de especialistas de negocios e industriales que desempeñan un trabajo que requiere grandes habilidades y especialización. Ejemplos de esto pueden incluir abogados, contadores, ingenieros, científicos, analistas de mercado, diseñadores de publicidad y estadistas. Como su trabajo está basado en ramas del conocimiento bien definidas, a veces son llamados **trabajadores del conocimiento**. Los sistemas de información que se dirigen al personal técnico y profesional se enfocan en el análisis de datos así como en generar información oportuna para la solución de problemas.
- **Supervisores, administradores medios y administradores ejecutivos.** Son los tomadores de decisiones. Los supervisores tienden a enfocarse en la solución de problemas y en la toma de decisiones diaria. Los administradores intermedios están más preocupados por los problemas operacionales (corto plazo) y en la toma de decisiones. Los administradores ejecutivos están preocupados por la planeación y toma de decisiones estratégicas (a largo plazo). Los sistemas de información para los administradores tienden a enfocarse completamente en el acceso a la información. Los administradores necesitan el derecho a la información en el momento correcto para identificar y resolver problemas así como para tomar buenas decisiones.

**Usuarios externos del sistema** Internet ha permitido que las fronteras tradicionales de los sistemas de información se extiendan para incluir otros negocios o consumidores directos como usuarios del sistema. Estos usuarios externos del sistema constituyen un porcentaje cada vez más grande de usuarios de sistemas para los sistemas de información modernos. Los ejemplos incluyen:

- **Clientes.** Cualquier organización o individuo que compra nuestros productos y servicios. En la actualidad, nuestros clientes se pueden convertir en usuarios directos de nuestros sistemas de información cuando pueden ejecutar directamente pedidos y transacciones de ventas que antes requerían intervención de un usuario interno. Por ejemplo, si usted compró un producto de la compañía a través de Internet, se convirtió en un usuario externo del sistema de información de ventas de esa empresa. (No hubo necesidad de que un usuario interno por separado del negocio colocara su pedido.)
- **Proveedores.** Cualquier organización de la que nuestra compañía pueda comprar suministros y materia prima. Estos proveedores pueden interactuar de manera directa con los sistemas de información de nuestra compañía para determinar nuestras necesidades de suministros y automáticamente crear pedidos para satisfacer esas necesidades. Ya no hay necesidad de que un usuario inicie esos pedidos a un proveedor.
- **Socios.** Cualquier organización de la que nuestra compañía puede adquirir servicios o con la que se asocia. La mayoría de los negocios modernos contrata o subcontrata diversos servicios básicos como mantenimiento del sitio, administración de redes y muchos más. Y las empresas han aprendido a asociarse con otras para impulsar con mayor rapidez las fortalezas y construir mejores productos con más rapidez.
- **Empleados.** Son aquellos que trabajan en viajes o en su casa. Por ejemplo, los representantes de ventas generalmente pasan gran parte de su tiempo en viajes. También, muchas empresas permiten a los empleados trabajar a distancia (es decir, hacerlo desde casa) para reducir costos y mejorar la productividad. Como usuarios móviles o remotos, estos empleados requieren acceso a los mismos sistemas de información así como los requeridos por los usuarios internos.

### VENTAJAS Y BENEFICIOS DE LOS SISTEMAS DE INFORMACIÓN

Aumento en la utilidad del negocio  
 Reducción de los costos del negocio  
 Costos y beneficios del sistema  
 Aumento en la participación de mercado  
 Mejora en las relaciones con los clientes  
 Aumento en la eficiencia  
 Mejor toma de decisiones  
 Mejor cumplimiento de la normatividad  
 Menos errores  
 Mejor seguridad  
 Mayor capacidad

#### usuario del sistema

"Cliente" que usa con regularidad un sistema de información o se ve afectado por él, capturando, validando, introduciendo, respondiendo, almacenando e intercambiando datos e información.

**trabajador del conocimiento** Todo trabajador cuyas responsabilidades se centran en un área especializada de conocimientos.

**usuario remoto** Usuario que no se ubica físicamente en las instalaciones donde se encuentran los sistemas de información pero que necesita y tiene acceso a ellos.

**usuario móvil** Usuario cuya ubicación cambia constantemente pero requiere acceso a los sistemas de información desde cualquier lugar.

### diseñador de sistemas

Especialista técnico que traduce los requerimientos de negocios de los usuarios del sistema y las restricciones en soluciones técnicas. Diseña las bases de datos, entradas, salidas, pantallas, redes y software que podrán satisfacer los requerimientos de los usuarios del sistema.

### constructor del sistema

Especialista técnico que construye sistemas de información y sus componentes con base en las especificaciones de diseño que generan los diseñadores de sistemas.

Cada vez más los usuarios externos del sistema son denominados **usuarios remotos** y **usuarios móviles**. Se conectan a nuestros sistemas de información mediante computadoras portátiles, computadoras manuales y teléfonos inteligentes, con cables o inalámbricos. Diseñar sistemas de información para estos aparatos presenta algunos de los desafíos más contemporáneos que abordaremos en este texto.

## > Diseñadores de sistemas

Los **diseñadores de sistemas** son especialistas en tecnología de sistemas de información. Como se muestra en la figura 1.1 estos diseñadores están involucrados en opciones de tecnología de información y en el diseño de sistemas que utilizan tecnologías elegidas. Los diseñadores de sistemas actuales tienden a enfocarse en especialidades técnicas. Algunos de ustedes pueden estarse preparando para una de las siguientes especialidades técnicas, como:

- **Administradores de bases de datos.** Especialistas en tecnologías de bases de datos que diseñan y coordinan cambios a las bases de datos corporativas.
- **Arquitectos de redes.** Especialistas en creación de redes y tecnología de telecomunicaciones que diseñan, instalan, configuran, optimizan y respaldan redes locales y de áreas amplias, que incluyen conexiones a Internet y otras redes externas.
- **Artistas gráficos.** Relativamente nuevos en la mezcla de trabajadores de tecnología de la información (TI) actual, los especialistas en tecnología y métodos gráficos diseñan y construyen interfaces fascinantes y fáciles de utilizar para sistemas, que incluyen interfaces para computadoras personales, la Web, manuales y teléfonos inteligentes.
- **Expertos en seguridad.** Especialistas en la tecnología y los métodos utilizados para garantizar la seguridad (y privacidad) de datos y redes.
- **Especialistas en tecnología.** Expertos en la aplicación de tecnologías específicas que serán utilizadas en un sistema (por ejemplo, un paquete de software comercial específico o un tipo específico de hardware).

## > Constructores de sistemas

Los **constructores de sistemas** (de nuevo, vea la figura 1.1) son otra categoría de especialistas de tecnología para sistemas de información. Su papel es construir el sistema de acuerdo con las especificaciones de los diseñadores del sistema. En las organizaciones pequeñas o con sistemas de información pequeños, los diseñadores de sistemas y los constructores de sistemas con frecuencia son las mismas personas. Pero en las organizaciones grandes con sistemas de información grandes a menudo ocupan puestos separados. Algunos de ustedes pueden estarse preparando para alguna de las siguientes especialidades técnicas, como:

- **Programadores de aplicaciones.** Especialistas que convierten los requerimientos de negocios y las declaraciones de problemas y procedimientos en lenguajes de computadora. Desarrollan y prueban programas de cómputo para capturar y almacenar datos, localizar y recuperar datos para aplicaciones de cómputo.
- **Programadores de sistemas.** Especialistas que desarrollan, prueban e implementan software a nivel sistema operativo, utilerías de software y otros servicios. Con mayor frecuencia, también desarrollan “componentes” de software reutilizable para uso por parte de programadores de aplicaciones (arriba).
- **Programadores de bases de datos.** Especialistas en lenguajes y tecnología de bases de datos que construyen, modifican y prueban estructuras de bases de datos y los programas que las utilizan y las mantienen.
- **Administradores de red.** Especialistas que diseñan, instalan, prueban contra fallas y optimizan las redes de cómputo.
- **Administradores de seguridad.** Especialistas que diseñan, implementan, prueban contra fallas y manejan los controles de seguridad y privacidad en una red.

- *Webmasters*. Especialistas que codifican y mantienen los servidores Web.
- *Integradores de software*. Especialistas que integran paquetes de software con hardware, redes y otros paquetes de software.

Aunque con este texto no se pretende educar directamente al constructor del sistema, sí se busca enseñar a los diseñadores de sistemas la forma de comunicar mejor las especificaciones de diseño a los constructores del sistema.

## > Analistas de sistemas

Como usted ha visto, los propietarios, usuarios, diseñadores y constructores de sistemas con frecuencia tienen muy distintas perspectivas acerca de cualquier sistema de información que se va a construir y a utilizar. Algunos están involucrados en generalidades, mientras que otros se enfocan a los detalles. Algunos no son técnicos, mientras que otros son muy técnicos. Esto presenta una brecha de comunicación que siempre ha existido entre quienes necesitan soluciones de negocios basadas en computadora y quienes entienden la tecnología de la información. El **analista de sistemas** crea puentes para acortar esa brecha. Usted puede (y probablemente lo hará) tener un papel como analista de sistemas o alguien que trabaja con analistas de sistemas.

Como se ilustró en la figura 1.1, su papel intencionalmente se superpone con los papeles de todos los demás involucrados. Para los propietarios y usuarios del sistema, los analistas de sistemas identifican y validan los problemas y necesidades de negocios. Para los diseñadores y constructores del sistema, los analistas de sistemas se aseguran de que la solución técnica satisfaga las necesidades del negocio e integre la solución técnica dentro del mismo. En otras palabras, los analistas de sistemas *facilitan* el desarrollo de los sistemas de información a través de la interacción con los demás involucrados.

Existen diversas variaciones legítimas pero a menudo confusas en el título del puesto que llamamos “analista de sistemas”. Un *analista/programador* (o *programador/analista*) incluye las responsabilidades del programador de cómputo y del analista de sistemas. Un *analista de negocios* se enfoca sólo en los aspectos no técnicos de un análisis y diseño de sistemas. Otros sinónimos de “analista de sistemas” son consultor de sistemas, analistas de sistemas, arquitecto de sistemas, ingeniero de sistemas, ingeniero de información, analista de información e integrador de sistemas.

Algunos de ustedes se convertirán en analistas de sistemas. Los demás trabajarán rutinariamente con analistas de sistemas que les ayudarán a resolver sus problemas de negocios e industriales al crear y mejorar su acceso a los datos e información necesaria para hacer su trabajo. Demos un vistazo más de cerca a los analistas de sistemas como facilitadores fundamentales del desarrollo de sistemas de información.

**El papel del analista de sistemas** Los analistas de sistemas entienden tanto de negocios como de cómputo. Estudian los problemas y oportunidades del negocio y luego transforman los requerimientos de negocios y de información en especificaciones de sistemas de información que serán implementados por diversos especialistas técnicos que incluyen programadores de computadoras. Los sistemas de computadoras y de información son valiosos para una empresa sólo si ayudan a resolver problemas o si implementan mejoras.

Los analistas de sistemas inician el *cambio* dentro de una organización. Cada nuevo sistema transforma el negocio. Cada vez más, los mejores analistas de sistemas literalmente cambian sus organizaciones, al proporcionar información que puede ser utilizada para tener una ventaja competitiva, al encontrar nuevos mercados y servicios e incluso al cambiar y mejorar dramáticamente la forma en que la organización hace negocios.

El analista de sistemas es básicamente un *solucionador de problemas*. A través de este texto, el término *problema* será utilizado para describir muchas situaciones, que incluyen:

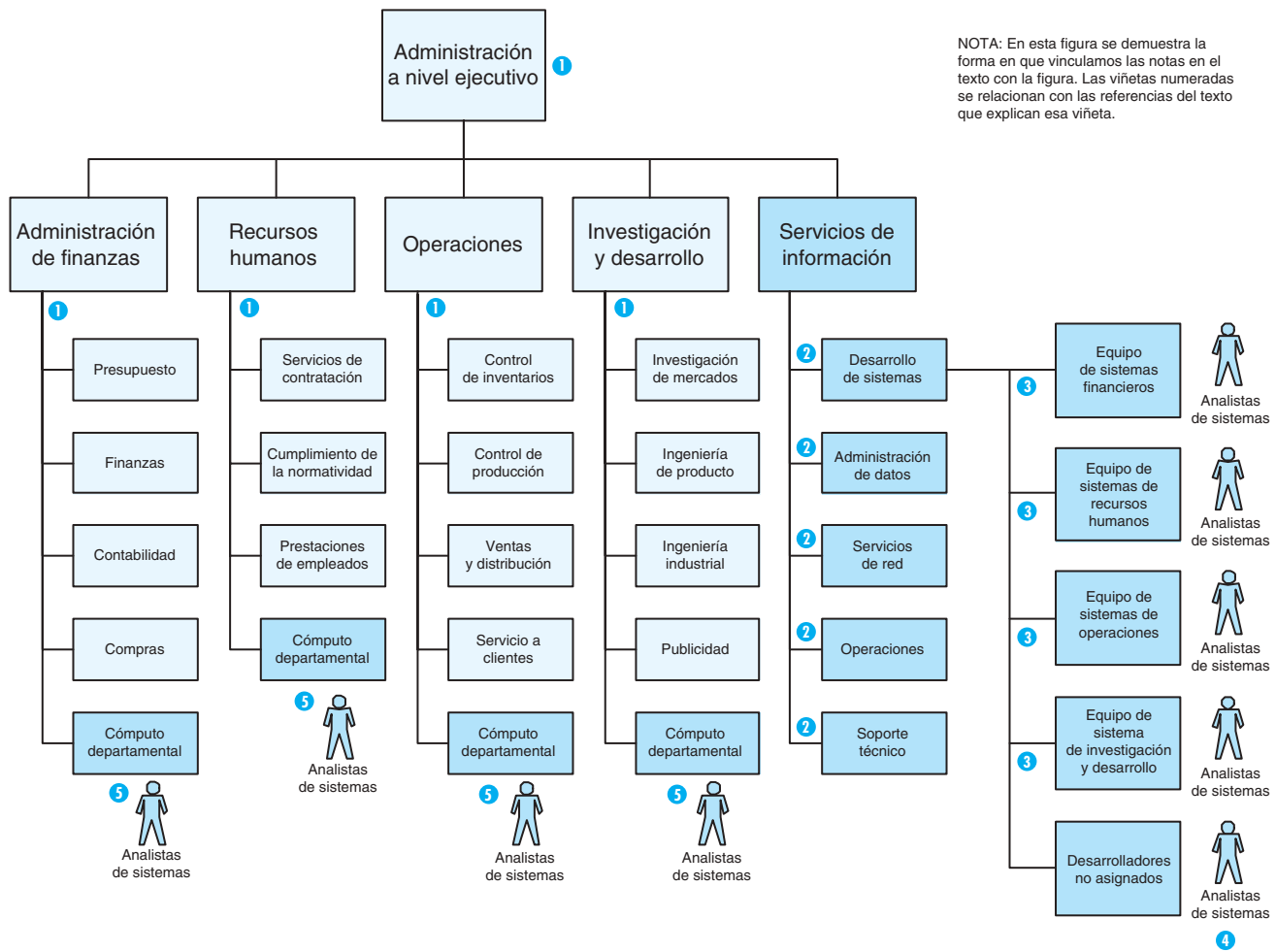
- Problemas, ya sean reales o anticipados, que requieren de una acción correctiva.
- Oportunidades para mejorar una situación a pesar de la ausencia de quejas.
- Directivas para cambiar una situación sin importar si alguien se ha quejado de la situación actual.

El puesto de analista de sistemas presenta un reto fascinante y emocionante para muchas personas. Ofrece una alta visibilidad administrativa y oportunidades para una

---

**analista de sistemas** Especialista que estudia los problemas y necesidades de una organización para determinar la forma en que las personas, los datos, los procesos y la tecnología de la información pueden lograr óptimamente mejoras para la empresa.





**FIGURA 1.2** Analistas de sistemas en una organización típica

toma de decisiones importante y una creatividad que puede afectar a una organización completa. Es más, este puesto puede ofrecer estos beneficios relativamente temprano en su carrera (comparado al nivel de ingreso de otros trabajos y carreras).

**¿Dónde trabajan los analistas de sistemas?** Cada empresa se organiza de manera única. Pero ciertos patrones de organización parecen ser recurrentes. La figura 1.2 es un organigrama representativo. Las siguientes viñetas numeradas hacen referencia y enfatizan los puntos fundamentales en la figura:

- 1 Los propietarios y usuarios de sistemas están ubicados en las unidades y subunidades funcionales del negocio, así como en la administración ejecutiva.
- 2 Los diseñadores y constructores de sistemas están ubicados generalmente en la unidad de sistemas de información del negocio. La mayoría de los analistas de sistemas también trabaja para la unidad de servicios de información de una organización.
- 3 Como se muestra en la figura, los analistas de sistemas (junto con los diseñadores y constructores de sistemas) pueden estar asignados permanentemente a un equipo que respalda una función de negocios específica (por ejemplo, sistemas financieros).

Los números 2 y 3 anteriores representan un enfoque tradicional para organizar a los analistas de sistemas y otros desarrolladores. Los números 4 y 5 posteriores representan estrategias que tienen la intención de enfatizar la eficiencia o la experiencia de negocios. Todas las estrategias pueden ser combinadas en una sola organización.

## La próxima generación: Prospectos de carrera para analistas de sistemas

Muchos de ustedes están considerando o se están preparando para una carrera como analista de sistemas. La vida de un analista de sistemas es desafiante y gratificante. Pero, ¿cuáles son las perspectivas para el futuro? ¿Las organizaciones necesitan analistas de sistemas? ¿Los necesitarán en el futuro inmediato? ¿Ese puesto está cambiando para el futuro?, y si es así, ¿en qué forma? Abordaremos estas preguntas en este recuadro.

De acuerdo con el Departamento de Trabajo estadounidense, los puestos relacionados con cómputo representan cinco de las 20 ocupaciones de más rápido crecimiento en la economía. Lo que es más, estas ocupaciones, relacionadas con cómputo, tienen rápido crecimiento y pagan mejor que muchos otros puestos.

En el 2002, 468 000 trabajadores eran clasificados como analistas de sistemas. Para el 2012, ese número crecerá a 653 000, un aumento de 39%. Esto significa que al menos 185 000 nuevos analistas de sistemas deben ser preparados y contratados (sin incluir a los que requieren ser reemplazados por retiro o por haberse cambiado a posiciones administrativas u otras ocupaciones). La necesidad va en aumento debido a que la industria necesita analistas de sistemas para satisfacer al parecer la infinita demanda por más sistemas de información y aplicaciones de software. Como algunos puestos de programación son subcontratados a contratistas independientes y a otros países, la necesidad aumenta todavía más para analistas de sistemas hábiles, que puedan crear especificaciones de diseño sólidas para equipos de desarrollo remotos. Las oportunidades para el éxito serán mejores para los analistas más preparados, calificados, hábiles y experimentados.

¿Qué sucede con el analista de sistemas exitoso? ¿Un puesto como analista de sistemas lleva a cualquier otra carrera? De hecho, existen muchos caminos de carrera. Algunos analistas dejan el campo de sistemas de información y se unen a la comunidad de usuarios. Su experiencia con el desarrollo de aplicaciones de negocios, combinada con su perspectiva de sistemas integral, puede hacer que los analistas de sistemas experimentados sean especialistas de negocios. Alternativamente, los analistas se pueden volver administradores de proyectos, administradores de sistemas de información o especialistas técnicos (de bases de datos, telecomunicaciones, microcomputadoras y demás).

Finalmente, los analistas de sistemas hábiles a menudo son reclutados por las industrias de consultoría y contratación externa. Las oportunidades de caminos de carrera son virtualmente ilimitadas.

Como en cualquier profesión, los analistas de sistemas pueden esperar cambios. Aunque siempre es peligroso pronosticar cambios, lo intentaremos. Creemos que las organizaciones se volverán cada vez más dependientes de las fuentes externas para sus analistas de sistemas, consultores y contratistas externos. Esto será dirigido por factores tales como la complejidad y el cambio rápido de tecnología, el deseo de acelerar el desarrollo de sistemas, y la dificultad continua en reclutar, retener y volver a entrenar analistas de sistemas hábiles (y otros profesionales de tecnología de la información). En muchos casos, los analistas de sistemas empleados manejarán proyectos a través de contratos de consultoría o subcontratación.

Creemos que un porcentaje cada vez mayor de los analistas de sistemas de mañana no trabajarán en el departamento de sistemas de información. En lugar de eso, trabajarán en forma directa para una unidad de negocios dentro de una organización. Esto les permitirá atender mejor a sus usuarios. También dará a los usuarios mayor poder sobre qué sistemas son construidos y respaldados.

Finalmente, también creemos que un mayor porcentaje de analistas de sistemas tendrá antecedentes distintos al cómputo. En algún momento, la mayoría de los analistas eran especialistas de cómputo. Actualmente, los graduados en cómputo se vuelven más conocedores en negocios. En forma similar, los recientes graduados en negocios y en carreras no relacionadas con el cómputo se vuelven cada vez más expertos de cómputo. Su ayuda y conocimientos de tiempo completo se necesitarán para satisfacer la demanda y proporcionar los antecedentes de negocios necesarios para las aplicaciones más complejas del mañana.

- 4 Los analistas de sistemas (junto con los diseñadores y constructores de sistemas) pueden también ser compartidos y temporalmente asignados a proyectos específicos para cualquier función de negocios según sea necesario. (Algunas organizaciones creen que este método arroja una mayor eficiencia debido a que los analistas y otros desarrolladores siempre son asignados a los proyectos de mayor prioridad sin importar la experiencia en el área de negocios.)
- 5 Algunos analistas de sistemas pueden trabajar para organizaciones de cómputo departamentales más pequeñas que respaldan y reportan a sus propias funciones de negocios específicas. (Algunas organizaciones creen que esta estructura resulta en analistas de sistemas que desarrollan una mayor experiencia en su área de negocios asignada para complementar su experiencia técnica.)

Todas las estrategias anteriores, desde luego, pueden reflejarse dentro de una sola organización.

Sin importar dónde se asignan los analistas de sistemas dentro de la organización, es importante estar conscientes de que se reúnen en *equipos de proyectos*. Éstos deben también incluir una representación apropiada de los demás involucrados que previamente analizamos (propietarios de sistemas, usuarios de sistemas, diseñadores de sistemas, y constructores de sistemas). De acuerdo con eso, enfatizaremos la construcción de equipos y el trabajo en equipo a lo largo de este libro.

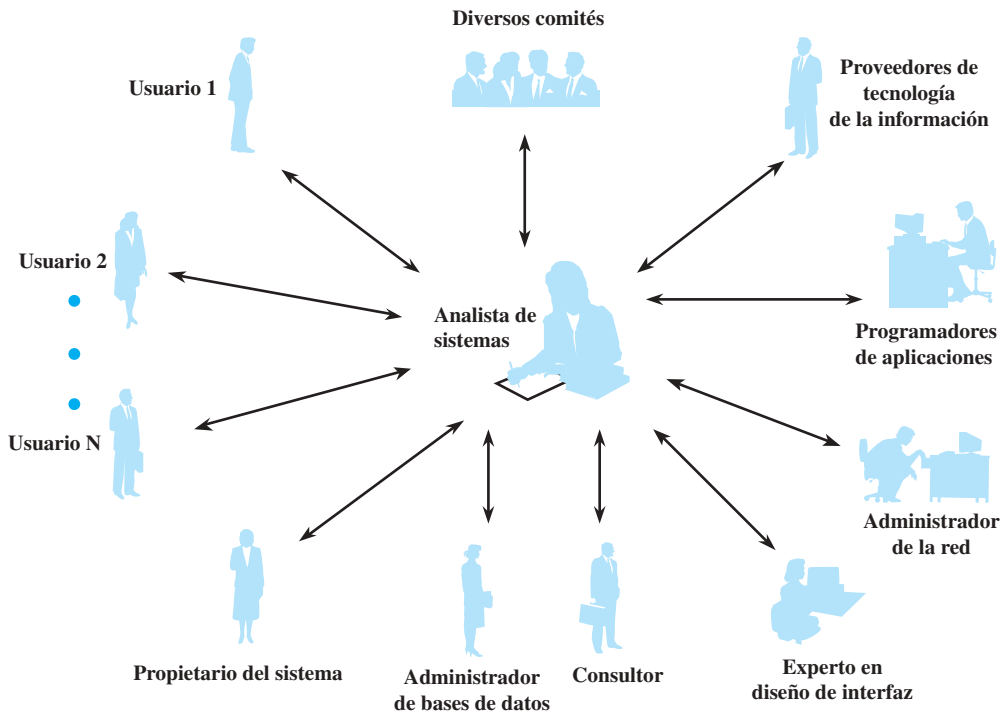
**Habilidades requeridas por el analista de sistemas** Para aquellos de ustedes con aspiraciones de convertirse en analistas de sistemas, en esta sección se describen las habilidades que necesitarán desarrollar. En este libro se presentan muchos análisis de sistemas y conceptos de diseño, herramientas y técnicas. Pero usted también necesitará habilidades y experiencias que ni este libro ni su curso de análisis y diseño pueden proporcionar completamente.

Cuando todo esto falla, el analista de sistemas que recuerda los conceptos básicos y los principios del “pensamiento de sistemas” aún tendrá éxito. No hay herramienta, técnica, proceso o metodología perfecta en todas las situaciones. Pero los conceptos y principios del pensamiento de sistemas siempre le ayudarán a adaptarse a situaciones nuevas y diferentes. En este libro se enfatiza el pensamiento de sistemas.

Hace poco tiempo se pensaba que las únicas herramientas reales de un analista de sistemas eran papel, lápiz y una plantilla de diagrama de flujos. Al paso de los años, varias herramientas y técnicas han sido desarrolladas para ayudar al analista de sistemas. Desafortunadamente, muchos libros enfatizan una clase específica de herramientas que es asociada con una metodología o enfoque al análisis y diseño de sistemas. Conforme usted lea este libro, su caja de herramientas aumentará para incluir muchas de las diferentes metodologías y enfoques para el análisis y diseño de sistemas. Subsecuentemente, usted deberá elegir y utilizar las herramientas con base en las distintas situaciones que encontrará como analista; ¡la herramienta correcta para el trabajo correcto!

Además de tener habilidades formales de análisis y diseño de sistemas, un analista de sistemas debe desarrollar o poseer otras habilidades, conocimientos y rasgos para completar el puesto. Éstos incluyen:

- **Conocimiento laboral de tecnologías de información.** El analista debe estar consciente de las tecnologías de información existentes y las que están en surgimiento. Dicho conocimiento puede ser adquirido en cursos universitarios, seminarios y cursos de desarrollo profesional y programas de capacitación corporativa internos. Los analistas practicantes también se mantienen al día a través de una lectura disciplinada y participación en las sociedades profesionales apropiadas. (Para empezar, véanse las lecturas recomendadas al final de este capítulo y los siguientes.)
- **Experiencia en programación de computadoras.** Es difícil imaginar cómo los analistas de sistemas podrían preparar adecuadamente especificaciones técnicas y de negocios para un programador si ellos no tuvieran alguna experiencia en programación. La mayoría de los analistas de sistemas necesitan estar capacitados en uno o más lenguajes de programación de alto nivel.
- **Conocimiento general de procesos y terminología de negocios.** Los analistas de sistemas deben ser capaces de comunicarse con los expertos de negocios para obtener una comprensión de sus problemas y necesidades. Para el analista, al menos parte de este conocimiento viene sólo mediante la experiencia. Al mismo tiempo, los que aspiran a ser analistas deben aprovechar todas las oportunidades para completar cursos de alfabetización en negocios básicos disponibles en las universidades de negocios. Los cursos importantes pueden incluir contabilidad financiera, administración o contabilidad de costos, finanzas, marketing, administración de manufactura u operaciones, administración de la calidad, economía y leyes de negocios.
- **Habilidades generales de solución de problemas.** El analista de sistemas debe ser capaz de tomar un problema de negocios grande, dividirlo en partes, determinar las causas y efectos del problema, y luego recomendar una solución. Los analistas deben evitar la tendencia a sugerir la solución antes de analizar el problema. Para los que aspiran a ser analistas, muchas universidades ofrecen cursos de filosofía que enseñan habilidades en la solución de problemas, pensamiento crítico y razonamiento. Estas “habilidades suaves o ligeras” servirán también a un analista.



**FIGURA 1.3** El analista de sistemas como facilitador

- **Buenas habilidades de comunicación interpersonal.** Un analista debe ser capaz de comunicarse eficazmente, tanto en forma oral como escrita. Casi sin excepción, sus habilidades de comunicación, y no las técnicas, probarán ser el factor más importante en el éxito o fracaso de su carrera. Estas habilidades se pueden aprender, pero la mayoría de nosotros debemos esforzarnos a obtener ayuda y a trabajar fuerte para mejorarlas. La mayoría las escuelas ofrecen cursos de escritura técnica y de negocios, de comunicación verbal técnica y de negocios, técnicas de entrevista y escucha; todas ellas habilidades útiles para el analista de sistemas. Estas habilidades se enseñan en el capítulo 6.
- **Buenas habilidades de relaciones interpersonales.** Como se ilustró en la figura 1.3, el analista de sistemas interactúa con todos los involucrados en un proyecto de desarrollo de sistemas. Estas interacciones requieren habilidades interpersonales eficaces que permiten al analista tratar con dinámicas de grupo, política de negocios, conflictos y cambio. Muchas escuelas ofrecen cursos de desarrollo de habilidades interpersonales en temas como trabajo en equipo, principios de persuasión, manejo del cambio, del conflicto y liderazgo.
- **Flexibilidad y adaptabilidad.** No hay dos proyectos iguales. Por eso, no hay un enfoque o estándar mágico que sea igualmente aplicable a todos los proyectos. Los analistas de sistemas exitosos aprenden a ser flexibles y a adaptarse a situaciones y retos únicos. Nuestro enfoque de la caja de herramientas antes mencionado tiene la intención de alentar la flexibilidad en el uso de herramientas y métodos de análisis y diseño de sistemas. Pero usted debe desarrollar una actitud de adaptabilidad para utilizar adecuadamente cualquier caja de herramientas.
- **Carácter y ética.** La naturaleza del puesto del analista de sistemas requiere un carácter fuerte y un sentido de lo correcto y lo incorrecto. Los analistas a menudo obtienen acceso a hechos e información sensible o confidencial que no debe ser revelada al público. También, los productos del análisis y diseño de sistemas son considerados generalmente como propiedad intelectual del empleador. Existen varios estándares para la ética de cómputo. Uno de ellos, del Instituto de Ética de Cómputo, es llamado “Los diez mandamientos de la ética de cómputo” y se muestra en la figura 1.4.

**FIGURA 1.4** Ética para los analistas de sistemas**Los diez mandamientos de la ética de cómputo**

1. No utilizará una computadora para dañar a otras personas.
2. No interferirá con el trabajo de cómputo de otras personas.
3. No se entrometerá en los archivos de cómputo de otras personas.
4. No utilizará una computadora para robar.
5. No utilizará una computadora para dar falso testimonio.
6. No copiará o utilizará software registrado por el que no haya pagado.
7. No utilizará los recursos de cómputo de otras personas sin autorización o compensación adecuada.
8. No se apropiará de la producción intelectual de otra persona.
9. Deberá pensar en las consecuencias sociales del programa que usted escribe o el sistema que usted diseña.
10. Siempre deberá usar una computadora en formas que aseguren consideración y respeto para el resto de la humanidad.

Fuente: Instituto de Ética de Cómputo.

**proveedor de servicio externo (ESP)**

Un analista, diseñador o constructor de sistemas que vende su conocimiento y experiencia a otras empresas para ayudarlas en la compra, el desarrollo o la integración de sus soluciones de sistemas de información; puede estar afiliado a una organización de consultoría o servicios.

**administrador de proyectos**

Profesional experimentado que acepta la responsabilidad de planear, monitorear y controlar proyectos con respecto a un calendario, presupuesto, entregables, satisfacción del cliente, normas técnicas y calidad del sistema.

**> Proveedores de servicio externo**

Aquellos lectores con algo de experiencia de cómputo pueden preguntarse dónde entran los consultores en nuestra taxonomía de los involucrados. No son aparentes inmediatamente en nuestro marco de referencia. Pero ¡están ahí! Cualquiera de los papeles de involucrados puede ser ocupado por trabajadores internos o externos. Los consultores son un ejemplo de un **proveedor de servicio externo (external service provider, ESP)**. La mayoría de los ESP son analistas de sistemas, diseñadores o constructores que son contratados para traer una experiencia especial a un proyecto específico. Ejemplos de esto pueden ser ingenieros en tecnología, ingenieros de ventas, consultores de sistemas, programadores de contratos e integradores de sistemas.

**> El administrador de proyectos**

Hemos presentado la mayoría de los participantes fundamentales en el desarrollo moderno de sistemas de información: propietarios de sistemas, usuarios, diseñadores, constructores y analistas. Debemos concluir enfatizando el hecho de que estos individuos deben trabajar juntos como equipo para construir con éxito sistemas de información y aplicaciones que beneficiarán el negocio. Los equipos requieren de liderazgo. Por esta razón, generalmente uno o más de estos involucrados toma el papel de **administrador de proyectos** para asegurar que los sistemas se desarrollen a tiempo, dentro del presupuesto y con una calidad aceptable. Como se indica en la figura 1.1, la mayoría de los administradores de proyectos son experimentados analistas de sistemas. Sin embargo, en algunas organizaciones, los administradores de proyectos son seleccionados de las filas de lo que hemos llamado “propietarios de sistemas”. Como sea, la mayoría de las organizaciones han aprendido que la administración de proyectos es un papel especializado que requiere de habilidades y experiencia distintas.



Cada capítulo proporcionará una guía para una instrucción de ritmo individual bajo el título de “Mapa de aprendizaje”. Al reconocer que los distintos alumnos y lectores tienen diferentes antecedentes e intereses, propondremos caminos de aprendizaje adecuados, la mayoría dentro de este texto, pero algunos más allá del alcance de este libro.

La mayoría de los lectores deben proseguir directamente al capítulo 2 porque los primeros cuatro capítulos proporcionan gran parte del contexto para el resto del libro. Varios temas recurrentes, marcos de trabajo y términos se presentan en esos capítulos para permitirle definir su propio camino de aprendizaje desde ese punto en adelante. En este capítulo nos enfocamos en los sistemas de información desde cuatro distintas perspectivas:

- Los *jugadores*. Desarrolladores y usuarios de los sistemas de información.
- Los *impulsores de negocios* que actualmente influyen en los sistemas de información.
- Los *impulsores de tecnología* que actualmente influyen en los sistemas de información.
- El *proceso* de desarrollo de sistemas de información.

En el capítulo 2 veremos más de cerca al *producto* mismo, los sistemas de información, desde una perspectiva de arquitectura apropiada para el desarrollo de sistemas. Definiremos cómo se visualiza un sistema de información por los distintos jugadores y las etapas de desarrollo.

Más adelante, en el capítulo 3, se analiza de cerca el *proceso* del desarrollo de sistemas.



## Resumen

1. Los sistemas de información en las organizaciones capturan y administran datos para producir información útil que respalda a una organización y a sus empleados, clientes, proveedores y asociados.
2. Los sistemas de información pueden ser clasificados de acuerdo con las funciones que atienden, lo que incluye:
  - a) Sistemas de proceso de transacción que procesan las transacciones de negocios como pedidos, tarjetas de tiempo (para registrar la entrada), pagos y reservaciones.
  - b) Sistemas de información administrativa que utilizan datos de transacciones para producir información requerida por administradores para dirigir el negocio.
  - c) Sistemas de soporte de decisiones que ayudan a diversos tomadores de decisiones a identificar y elegir entre opciones o decisiones.
  - d) Sistemas de información ejecutiva hechos a la medida para las necesidades de información únicas de ejecutivos que planean el negocio y evalúan el desempeño contra esos planes.
  - e) Sistemas expertos que capturan y reproducen el conocimiento de un solucionador de problemas o un tomador de decisiones experto y luego simulan el “pensamiento” de ese experto.
  - f) Sistemas de comunicación y colaboración que resaltan la comunicación y colaboración entre las personas, tanto internas como externas a la organización.
- g) Sistemas de automatización de oficina que ayudan a los empleados a crear y compartir documentos que respaldan las actividades de oficina diarias.
3. Los sistemas de información pueden ser vistos desde distintas perspectivas, incluida la perspectiva de los “jugadores”, de los “impulsores de negocios” que influyen en el sistema de información, los “impulsores de tecnología” utilizados por el sistema de información y el “proceso” utilizado para desarrollar el sistema de información.
4. Los trabajadores de la información son los involucrados en los sistemas de información. Incluyen a esas personas cuyos puestos abarcan la creación, el acopio, el proceso, la distribución y el uso de la información. Éstos son:
  - a) Propietarios del sistema, patrocinadores y principales defensores de los sistemas de información.
  - b) Usuarios del sistema, la gente que utiliza o que es impactada por el sistema de información en una base regular. Geográficamente, los usuarios del sistema pueden ser internos o externos.
  - c) Constructores del sistema, especialistas de tecnología que construyen el sistema de información basado en las especificaciones de diseño.

- d) Analistas de sistemas, que facilitan el desarrollo de los sistemas de información y las aplicaciones de cómputo. Ellos coordinan los esfuerzos de los propietarios, usuarios, diseñadores y constructores. Con frecuencia, pueden jugar uno de esos papeles también. Los analistas de sistemas realizan el análisis y diseño de sistemas.
5. Además de tener habilidades de análisis y diseño de sistemas formales, un analista de sistemas debe desarrollar o poseer las siguientes habilidades, conocimiento y rasgos:
- Conocimiento del trabajo de las tecnologías de la información.
  - Experiencia en programación de computadoras.
  - Conocimiento general de procesos y terminología de negocios.
  - Habilidades generales de solución de problemas.
- Habilidades buenas de comunicación interpersonal.
  - Habilidades buenas de relaciones interpersonales.
  - Flexibilidad y adaptabilidad.
  - Carácter y ética.
6. Cualquier papel de involucrado puede ser desempeñado por un trabajador interno o externo llamado proveedor de servicio externo (ESP). La mayoría de los ESP son analistas, diseñadores o constructores de sistemas que son contratados por su experiencia en un proyecto específico.
7. La mayoría de los proyectos de sistemas de información incluyen el trabajo en equipo. Generalmente uno o más de los involucrados (miembros del equipo) toman el papel de administrador de proyecto para asegurarse de que el sistema se desarrolle a tiempo, dentro del presupuesto y con una calidad aceptable. La mayoría de los administradores de proyecto son experimentados analistas de sistemas.

## Preguntas de repaso



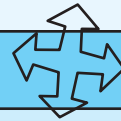
- ¿Por qué los sistemas de información (IS) son esenciales en las organizaciones?
- ¿Por qué los analistas de sistemas necesitan conocer quiénes son los involucrados en la organización?
- ¿Quiénes son los involucrados típicos en un sistema de información? ¿Cuáles son sus papeles?
- Favor de explicar cuáles son las consecuencias si un sistema de información carece de un propietario del sistema.
- ¿Cuáles son las diferencias entre los usuarios internos y los externos? Dé ejemplos.
- ¿Cuáles son las diferencias entre el papel del analista de sistemas y el del resto de los involucrados?
- ¿Qué tipo de conocimientos y habilidades debe poseer un analista de sistemas?
- Además del conocimiento del negocio y de cómputo que los analistas de sistemas deben poseer, ¿cuáles son las otras habilidades esenciales que necesitan para cumplir eficazmente con su trabajo?
- ¿Por qué las buenas habilidades de comunicación interpersonales son esenciales para los analistas de sistemas?

## Problemas y ejercicios



- Suponga que usted es un analista de sistemas que conducirá un análisis de requerimientos para una tienda de ventas tradicional de propiedad individual con un sistema de punto de ventas. Identifique quiénes serían los usuarios típicos internos y externos.
- Suponga que usted es un analista de sistemas de una empresa consultora y se le ha pedido ayudar al presidente y director ejecutivo de un banco regional. El banco recientemente implementó un plan para reducir la cantidad de personal, incluidos ejecutivos de préstamos, como una estrategia para mantener la rentabilidad. Posteriormente, el banco experimentó problemas crónicos con las solicitudes de préstamo retrasadas debido al número limitado de empleados de préstamos que eran capaces de revisar y aprobar o desaprobar los préstamos. El presidente y director ejecutivo del banco está involucrado en soluciones que permitirían que el proceso de aprobación se moviera con mayor rapidez sin aumentar el número de empleados de préstamos y ha contratado a su compañía para obtener soluciones. ¿Cuál es un tipo de sistema que usted podría recomendar al banco?
- ¿Cómo los sistemas de comunicación y colaboración mejoran la eficiencia y la eficacia? ¿Cuáles son algunos de los sistemas de comunicación y colaboración utilizados por un número cada vez mayor de organizaciones?
- Identifique el tipo de sistema de información que los empleados de oficina en una organización utilizarían generalmente y por qué.
- Conforme los sistemas de información aumentan en complejidad y profundidad, también aumentan los temas éticos en relación con el acceso y el uso

- de datos de estos sistemas. ¿Cuáles son algunos de estos temas éticos?
6. Aunque los procesos de desarrollo de sistemas y las metodologías pueden variar en gran medida, identifique y explique brevemente las fases “genéricas” del proceso de desarrollo del sistema que son descritas en el libro y que deben ser completadas para cualquier proyecto. Usted es un contratista con una compañía de integración de sistemas.
  7. Identifique a qué fase del proceso de desarrollo pertenecen las siguientes actividades:
    - a) Desarrollo del plano técnico o documento de diseño.
    - b) Programación de proyecto.
    - c) Pruebas de integración.
    - d) Entrevistar a los usuarios del sistema para definir los requerimientos del negocio.
  8. ¿Cuáles son las dos ventajas más importantes de las tecnologías de software orientadas a objetos sobre las tecnologías de software estructuradas?



## Proyectos e investigación

1. Investigue los salarios promedio de los profesionales de TI. Usted puede utilizar una diversidad de métodos para encontrar esta información, tales como búsqueda en la Web para sitios en línea que publican los resultados de encuestas salariales de profesionales de TI. También puede ver los anuncios clasificados en periódicos, revistas comerciales o en línea.
  - a) ¿Hay una diferencia significativa entre los salarios típicos de analistas, diseñadores y desarrolladores?
  - b) A grandes rasgos, ¿cuál es la diferencia en los salarios típicos de estos distintos grupos?
  - c) ¿Cuáles cree usted que son las razones para esta diferencia?
  - d) ¿Hay una brecha de género en los salarios de los profesionales de TI? Analice cualquier tendencia que haya encontrado y sus implicaciones.
2. Contacte a los directores ejecutivos de información (CIO) o a los gerentes de TI principales de diversas organizaciones locales o regionales. Pregunte acerca del proceso o la metodología que utilizan para el desarrollo de sistemas en sus organizaciones y por qué utilizan ese método en particular.
  - a) Describa y compare los distintos métodos que haya encontrado.
  - b) ¿Cuál método cree usted que sea el más eficaz?
  - c) ¿Por qué?
3. Opciones de carrera y habilidades personales:
  - a) En este punto en su educación, si usted tuviera que escoger entre convertirse en analista de sistemas, diseñador de sistemas o constructor de sistemas, ¿cuál elegiría?
  - b) ¿Por qué?
  - c) Ahora divida un pedazo de papel en dos columnas. En un lado, liste las habilidades personales y los rasgos que usted cree que son los más importantes para cada uno de estos tres grupos de analistas, diseñadores y constructores de sistemas. En la segunda columna, liste al menos cinco habilidades y rasgos que usted siente que serán los más fuertes, luego haga un mapa de ellos con las habilidades y rasgos que listó para cada uno de los tres grupos. ¿Con qué grupo tendría usted habilidades y rasgos en común?
  - d) ¿Es este grupo el mismo que usted elegiría en la pregunta 3a? ¿Por qué cree usted que este es (o no es) el caso?
4. La biblioteca de su escuela debe tener diarios y periódicos que datan de hace varias décadas o puede suscribirse a servicios de investigación en línea que lo hacen. Mire varios artículos recientes en diarios de tecnología de la información relacionados con análisis de sistemas, así como varios artículos de hace al menos 25 años.
  - a) Compare los artículos recientes con los viejos. ¿Parece haber diferencias significativas en el conocimiento, habilidades, capacidades o experiencia típicos que los analistas de sistemas necesitaban hace 25 años en comparación con la actualidad?
  - b) Si usted encontró diferencias, ¿cuáles son las que considera más importantes?
  - c) ¿Cuáles cree usted que son las razones para estos cambios?
  - d) Ahora saque su bola de cristal y vea al futuro dentro de 25 años. ¿Qué diferencias pronostica usted entre los analistas de sistemas actuales y los de dentro de 25 años?
5. Busque en la Web diversos artículos e información acerca de los temas éticos relacionados con profesionales de la tecnología de la información.
  - a) ¿Qué artículos encontró?
  - b) Con base en su investigación, ¿qué temas éticos cree usted que los analistas de sistemas podrían enfrentar durante sus carreras?

- c) Elija un tema ético en particular y explique los pasos que usted tomaría si fuera confrontado con este tema.
- d) ¿Qué haría usted si encontrara que su mejor amigo y compañero de trabajo ha cometido una seria violación ética? Los hechos que se deben considerar son: La violación puede no ser descubierta nunca, pero le costará a su compañía muchos miles de dólares en costos más altos durante los próximos años. Su compañía tiene una política estricta de despedir a empleados que cometen violaciones éticas serias. Asegúrese de explicar sus razones para las acciones que usted tomaría, si fuera el caso.
6. Busque en la Web o periódicos de negocios en su biblioteca como *Forbes Magazine* para información acerca de tres o cuatro ejecutivos de información de compañías u organizaciones grandes.
- a) ¿Qué sector, compañías y directores de la industria encontró usted?
- b) Para cada CIO que usted buscó, ¿cuál fue su experiencia predominante antes de convertirse en CIO; es decir, tenía algún antecedente de tecnología de información, antecedentes de negocios o ambos?
- c) Para cada CIO, ¿cuál fue su nivel de educación?
- d) ¿Cuántos años ha sido cada uno de ellos CIO y aproximadamente en cuántas compañías ha trabajado cada uno?
- e) Con base en su investigación, ¿qué conocimientos y habilidades necesita un CIO con el fin de ser exitoso y por qué?

## Casos breves



- ¿Qué cree usted que será posible tecnológicamente dentro de 10 años? ¿Y dentro de 20 o 30 años? Investigue una tecnología nueva e interesante que esté en la etapa de investigación y desarrollo. Prepare una presentación con el uso de un clip de película y PowerPoint acerca de esta tecnología y preséntelo a la clase. Emita un ensayo breve acerca de los impactos que esta nueva tecnología podría tener en la sociedad y/o los negocios.
- Considere las subcontrataciones: Muchas veces se da el caso de que al menos parte del proceso de desarrollo sea subcontratado. De hecho, los líderes de proyecto actuales deben ser capaces de manejar equipos geográficamente diversos así como restricciones de tiempo y de recursos. La subcontratación trae a la mesa el aumento en la eficiencia y ganancias económicas a las sociedades que interactúan. Sin embargo, estas ganancias no son notadas rápidamente y los impactos negativos en una sociedad que subcontrata pueden ser significativos desde la perspectiva de los puestos. El Dr. Mankiw, como consejero de economía del presidente Bush, públicamente promocionó los beneficios de la subcontratación, y fue duramente criticado por esta postura. ¿Cree usted que es bueno o malo para una empresa el subcontratar el trabajo? ¿Cree usted que existen dilemas éticos para las compañías que subcontratan? Encuentre al menos dos artículos acerca de los impactos de la subcontratación y llévelos para compartirlos en la clase.
- Usted es un administrador de red y como parte de su trabajo, vigila los correos electrónicos de los empleados. Usted descubre que su jefe está abreviando pasos en un sistema que su compañía desarrolla con el fin de terminar el proyecto más rápido y quedar dentro del presupuesto. Como resultado, hay una falla en el sistema y esta falla ocasionará una caída de la red si más de 20 personas están conectadas al mismo tiempo. El cliente espera aproximadamente 12 personas en la red en un momento dado. Usted está seguro, al parecer igual que su jefe, que el cliente no lo descubrirá hasta mucho después (si acaso) de que el proyecto haya sido aceptado. ¿Qué haría?
- Un analista de sistemas debe ser capaz de comunicarse técnica y exitosamente con los clientes. Desarrollar un buen sistema requiere una comprensión completa de los requerimientos del usuario. Muchas veces, los usuarios no saben lo que está disponible (tecnológicamente) o incluso lo que les gustaría obtener de un sistema. ¿Cuáles son las características de una buena comunicación?

## Ejercicios de equipo e individuales



- Reúnanse en grupos de dos personas. La persona uno decidirá acerca de una tarea que él o ella desea completar. Por ejemplo, sacar punta a un lápiz o escribir el nombre del profesor. Debe ser simple y directo. Esa persona debe comunicar en papel con el uso de diagramas sin palabras (verbales ni escritas) lo que quiere que se haga y dárselo a la persona número dos. La persona número dos debe entonces completar la tarea solicitada.
- ¿Qué descubrió de este ejercicio? ¿Cuánto tiempo le tomó hasta que la segunda persona entendió lo que la primera persona pedía? ¿Hubo una mala comuni-

cación? Escriba sus pensamientos y observaciones y compártalos con la clase.

3. Ejercicio individual: Imagine una tecnología increíble. El cielo es el límite y cualquier cosa es posible. ¿Cómo

4. Ejercicio individual: Recuerde cuál fue la última vez que alguien le dijo que algo *no podía* hacerse. ¿Qué fue? ¿Les prestó atención? ¿Por qué sí o por qué no?



## Lecturas recomendadas

Ambler, Scott. *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process*. Nueva York: John Wiley & Sons, 2002. Este libro ha dado forma significativa a nuestro pensamiento acerca del proceso de desarrollo de software. Aquellos de ustedes que critican el movimiento de “programación extrema” no deben temer que nuestro entusiasmo por esta lectura recomendada indique un apoyo de la programación extrema. Simplemente nos gusta el juicio que Scott trae al proceso del desarrollo de sistemas y software a través del uso de métodos flexibles dentro del contexto de un proceso iterativo. Haremos referencia a este libro en diversos capítulos.

Ernest, Kallman; John Grillo y James Linderman. *Ethical Decision Making and Information Technology: An Introduction with Cases*, 2a. ed. Burr Ridge, IL, McGraw Hill/Irwin, 1995. Es un texto excelente para enseñar ética en un currículo de MIS. Es una colección de casos de estudio que pueden complementar un curso de análisis y diseño de sistemas.

Simposio y exposición anual de TI de Gartner Group. La unidad de información administrativa de nuestra universidad ha estado suscrita desde hace tiempo al servicio de Gartner Group que reporta las tendencias de la industria, las probabilidades de éxito de tendencias y tecnologías y estrategias sugeridas para una transferencia de tecnología de la información. La investigación de Gartner ha tenido un papel significativo y continuo para ayudarnos a hacer diagramas de los controladores de negocios y de tecnología como se describen en este capítulo. También hemos tenido la fortuna de poder asistir al simposio anual de TI de Gartner. Los informes y simposios

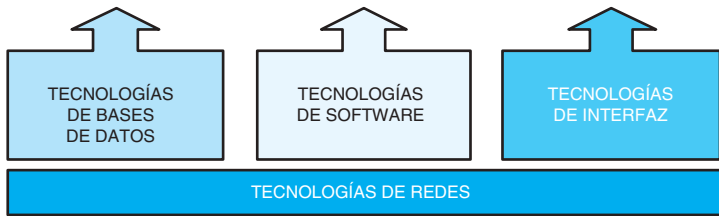
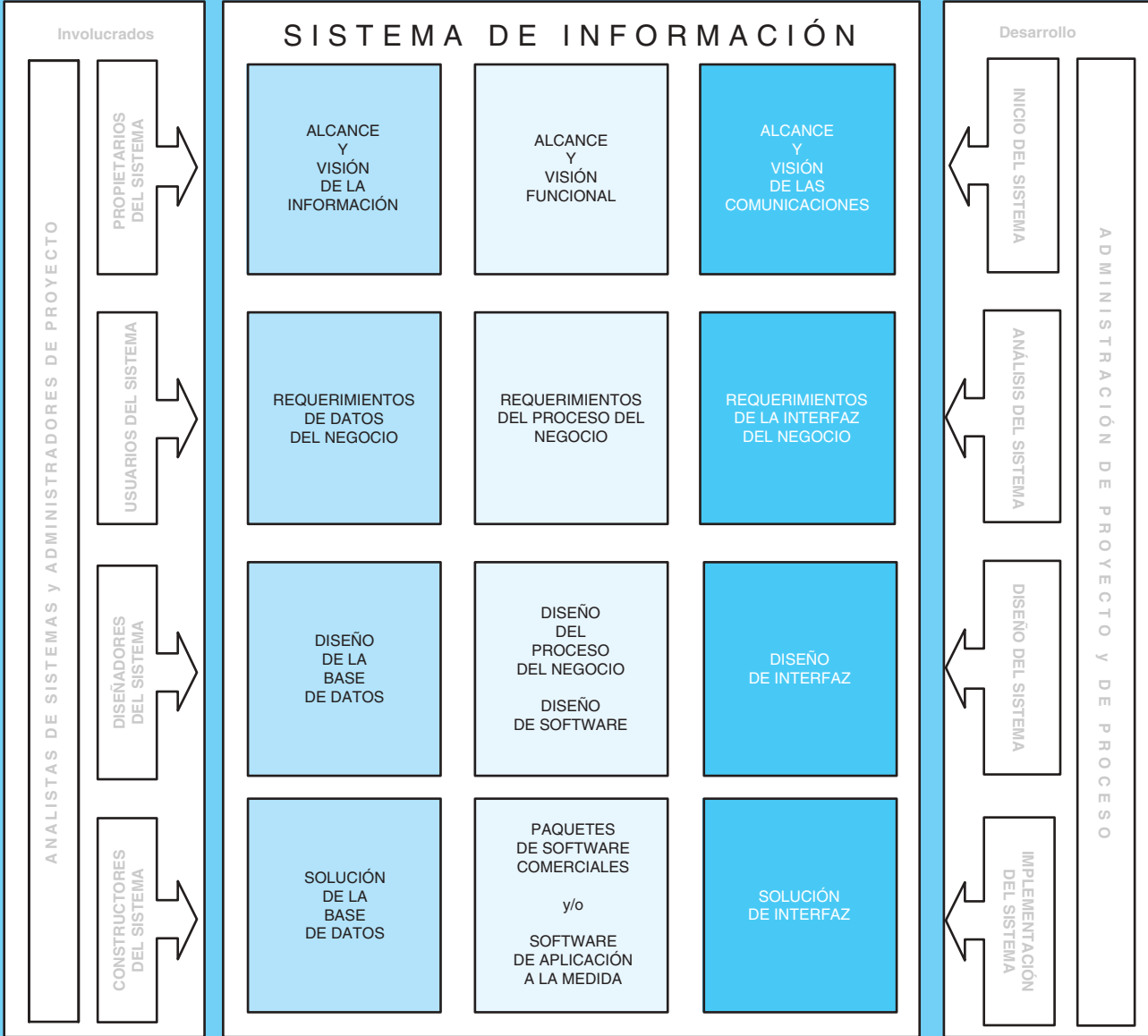
de Gartner Group han influido en cada edición de este libro. Para mayor información acerca de Gartner Group, véase [www.gartner.com](http://www.gartner.com).

Gause, Donald y Gerald Weinberg. *Are Your Lights On? How to Figure Out What the Problem REALLY Is*. Nueva York: Dorset House Publishing, 1990. Sí, este no es un libro reciente, pero tampoco lo son los básicos de solución de problemas. Aquí tenemos un libro breve y de lectura fácil acerca de solución general de problemas. Usted probablemente puede leer el libro completo en una noche y podría mejorar profundamente su potencial de solución de problemas como analista de sistemas (o, en su caso, en cualquier profesión).

Levine, Martin. *Effective Problem Solving*, 2a. ed., Englewood Cliffs, NJ, Prentice Hall, 1994. Éste es otro libro antiguo, pero como afirmamos antes, los métodos de solución de problemas son atemporales. Con sólo 146 páginas, este título puede servir como una excelente referencia profesional.

Weinberg, Gerald, *Rethinking Systems Analysis and Design*. Nueva York, Dorset House Publishing, 1988. No permita que la fecha lo engañe. Es uno de los mejores y más importantes libros acerca de este tema que hayan sido escritos. Este libro puede no enseñar ninguno de los métodos populares de análisis y diseño de nuestros días, pero desafía al lector a ir más allá de esos métodos para considerar algo mucho más importante, el lado de las personas del trabajo de sistemas. Las teorías y conceptos del Dr. Weinberg se presentan en el contexto de docenas de encantadoras fábulas y breves anécdotas. Le agradecemos nuestra fábula de análisis de sistemas favorita de todos los tiempos “The Three Ostriches”.

Impulsores de negocios



Impulsores de tecnología

# 2

## Componentes de sistemas de información

### Panorámica y objetivos del capítulo

Los métodos de análisis y diseño de sistemas se utilizan para desarrollar sistemas de información para organizaciones. Antes de conocer el *proceso* de construcción de sistemas, usted necesita una comprensión clara del *producto* que usted trata de construir. En este capítulo se toma un punto de vista de arquitectura de los sistemas de información y sus aplicaciones. Construiremos un marco de referencia para la arquitectura de los sistemas de información que posteriormente será utilizado para organizar y relacionar todos los capítulos en este texto. En el capítulo abordaremos las siguientes áreas:

- Diferenciar entre sistemas de información de *front-office* y de *back-office*.
- Describir las distintas clases de aplicaciones de sistemas de información (*de proceso de transacciones, información administrativa, soporte a las decisiones, experto, comunicación y colaboración y automatización de oficina*) y la forma en que interactúan para complementarse entre ellos.
- Describir el papel de la arquitectura de sistemas de información en el desarrollo de sistemas.
- Identificar tres metas de alto nivel que proporcionen a los propietarios de sistemas y a los usuarios de sistemas una perspectiva del sistema de información.
- Nombrar tres perspectivas orientadas a las metas de cualquier sistema de información.
- Identificar tres tecnologías que proporcionen a los diseñadores y constructores de sistemas una perspectiva del sistema de información.
- Describir cuatro componentes de la meta del CONOCIMIENTO de un sistema de información.
- Describir cuatro componentes de la meta del PROCESO de un sistema de información.
- Describir cuatro componentes de la meta de COMUNICACIONES de un sistema de información.
- Describir el papel de las tecnologías de red en relación con los componentes de CONOCIMIENTO, PROCESOS y COMUNICACIONES.

## Introducción

El proyecto de sistemas de servicios de membresía de SoundStage está en marcha. Bob Martínez ha sido asignado a la tarea de conducir las juntas iniciales con grupos de usuarios de sistemas para obtener su perspectiva del sistema y lo que debe lograr. Rápidamente, él descubrió que todos tenían una perspectiva distinta y la expresaban en un lenguaje diferente. En la universidad, al especializarse en tecnología de sistemas de información de cómputo, Bob tenía la tendencia a pensar en los sistemas de información en términos de lenguajes de programación, tecnologías de redes y bases de datos. Encontró que los demás no pensaban en esos términos. Los usuarios de sistemas hablaban del manual de instrucciones y de cómo estaban ocupados. Hablaban sobre políticas y procedimientos y de los informes que necesitaban. Cuando se reunió con los administradores, los escuchó hablar acerca de planes estratégicos y en la forma en que el sistema le podía dar a la organización una ventaja competitiva.

A Bob le recordó la historia que había escuchado acerca de tres hombres ciegos que encontraban un elefante. Uno tocó la trompa y concluyó que el elefante era como una víbora. Otro sintió una pata y concluyó que el elefante era como un árbol. El tercero tocó una oreja y concluyó que el elefante era como un abanico. A Bob no le tomó mucho darse cuenta de que las perspectivas de los propietarios y usuarios eran tan válidas como las suyas. Un sistema de información es más que sólo tecnología. Es principalmente una herramienta que sirve a las metas de la organización.

## El producto (sistemas de información)

En el capítulo 1 se presentaron los sistemas de información desde cuatro diferentes perspectivas, incluidas las de los involucrados, impulsores de negocios, impulsores de tecnología y el proceso de desarrollo de sistemas. Como se sugirió en la *página de inicio* (vea la página 22), en este capítulo se analizará más de cerca el “producto”, es decir, el sistema de información.

Las organizaciones son atendidas no por un solo sistema de información, sino por el contrario, por una federación de sistemas de información que respalda diversas funciones de negocios. Esta idea se ilustra en la figura 2.1. Nótese que la mayoría de las empresas

**FIGURA 2.1**

Una federación de sistemas de información



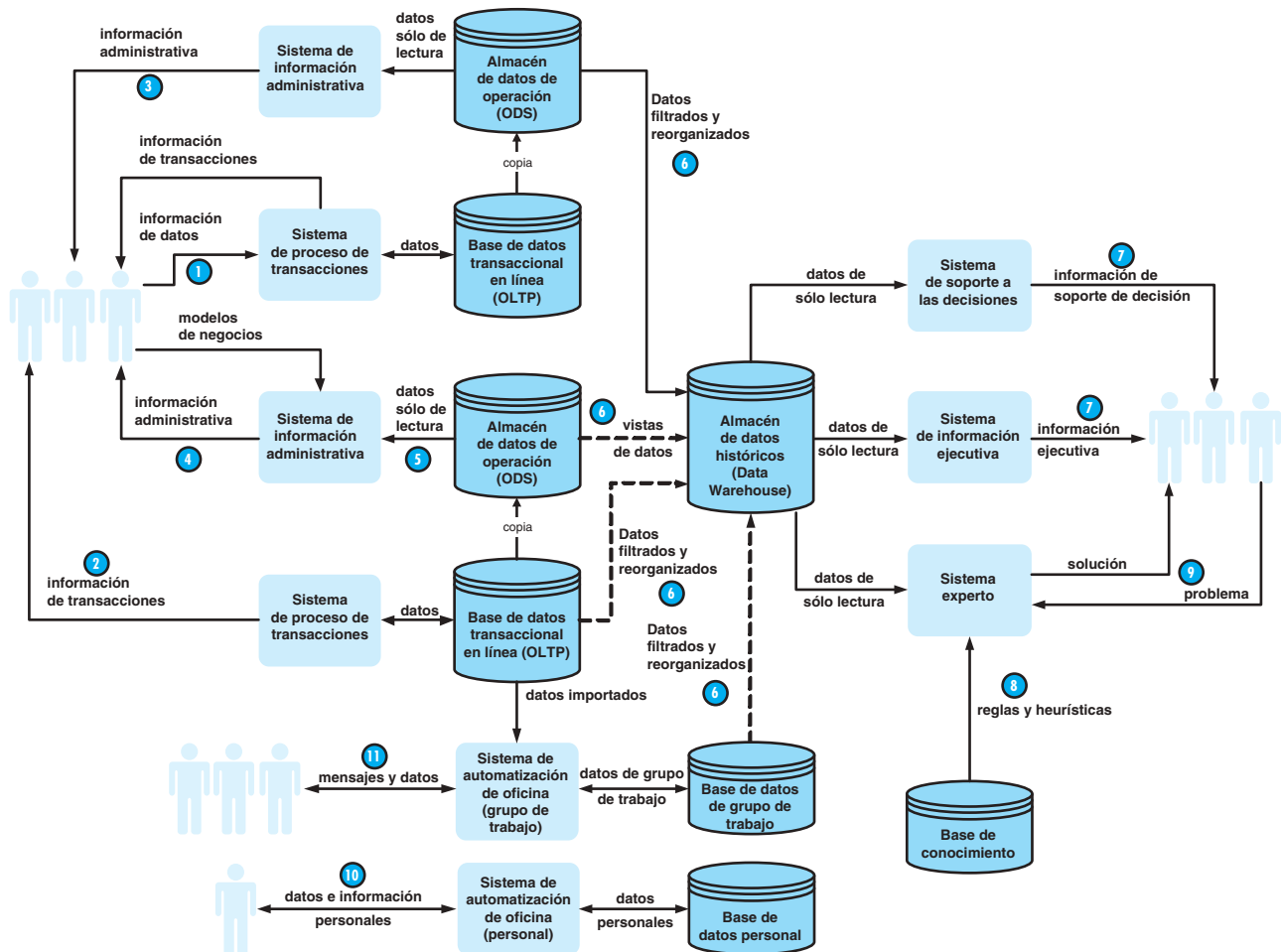


tienen tanto **sistemas de información de front-office** que soporta las funciones del negocio que se extienden hacia los clientes (y otros interesados), y **sistemas de información de back-office** que soportan las operaciones internas del negocio, al igual que interactúan con los proveedores. Estos sistemas de front-office y back-office alimentan de datos a sistemas de información administrativa y sistemas de soporte a las decisiones que respaldan las necesidades administrativas de las empresas. Los sistemas de información actuales se interconectan con clientes y proveedores por medio de la tecnología de comercio electrónico y aplicaciones de administración de la relación con los clientes (CRM) y administración de cadena de suministro (SCM) (véanse las descripciones en el capítulo 1) por Internet. Finalmente, la mayoría de las compañías tiene algún tipo de intranet (red interna para el negocio) para respaldar las comunicaciones entre los empleados y los sistemas de información.

En el capítulo 1 usted aprendió que hay diversas clases de aplicaciones de sistemas de información (véase la página opuesta). Cada clase atiende las necesidades de diferentes tipos de usuarios. En la práctica, estas clases se traslapan de manera que no siempre es fácil diferenciar una de otra. Las diversas aplicaciones deben interactuar de manera ideal para complementarse y soportarse. Tome unos minutos para estudiar la figura 2.2, en la que se ilustran los papeles típicos de los sistemas de información en una organización. Los rectángulos redondeados representan diversos sistemas de información. Nótese que una organización podrá y tendrá múltiples sistemas de proceso de transacciones, sistemas de automatización de oficina y demás. Las formas de “tambor” representan datos almacenados. Observe que una organización tiene múltiples conjuntos de datos almace-

**sistema de información de front-office** Un sistema de información que brinda soporte a funciones del negocio que se extienden hacia fuera, a los clientes de la organización.

**sistema de información de back-office** Sistema que brinda soporte a las operaciones internas del negocio, además de permitir el acceso a los proveedores.



**FIGURA 2.2** Aplicaciones de sistemas de información

### CLASES DE APLICACIONES DE SISTEMAS DE INFORMACIÓN

Sistema de proceso de transacciones (TPS)

Sistema de información administrativa (MIS)

Sistema de soporte a las decisiones (DSS)

Sistema de información ejecutiva (EIS)

Sistema experto

Sistema de comunicación y colaboración

Sistema de automatización de oficina

nados, y sólo algunos de ellos trabajan conjuntamente. Preste atención a las siguientes anotaciones numeradas en el diagrama:

- ① El primer sistema de proceso de transacción responde a una entrada de datos de transacción (por ejemplo, un pedido). Este produce información de la transacción para verificar el procesamiento correcto de la transacción de entrada.
- ② El segundo sistema de proceso de transacciones sólo produce una transacción de salida (por ejemplo, una factura). Tal sistema puede responder a algo tan simple como el paso del tiempo (por ejemplo, es fin de mes; por tanto, generará todas las facturas).
- ③ El primer sistema de información administrativa simplemente produce reportes o informes (por ejemplo, reportes de análisis de ventas) con el uso de datos almacenados en bases de datos de transacciones (mantenida por los sistemas de proceso de transacciones antes mencionados).
- ④ El segundo sistema de información administrativa utiliza modelos de negocios (por ejemplo, MRP) para producir información administrativa de las operaciones (por ejemplo, un programa de producción).
- ⑤ Nótese que un MIS puede utilizar datos de más de una base de datos transaccionales.
- ⑥ Observe que vistas de los datos de las bases de datos transaccionales componen un almacén de datos. Las vistas pueden ser tomadas en diversos intervalos y distintos subconjuntos de datos pueden ser incluidos en las diversas vistas. Los datos en el almacén serán organizados para asegurar un acceso fácil y consultas por parte de los administradores.
- ⑦ Las aplicaciones de soporte a las decisiones y sistemas de información ejecutiva generalmente proporcionarán acceso sólo de lectura a los almacenes de datos para producir información de soporte a las decisiones e información ejecutiva.
- ⑧ Un sistema experto requiere una base de datos especial que almacene experiencia en forma de reglas y heurísticas.
- ⑨ Un sistema experto acepta problemas como entradas (por ejemplo, ¿se debe otorgar crédito a un determinado cliente?) o problemas de razonamiento en el ambiente (por ejemplo, ¿el torno produce partes dentro de las especificaciones aceptables?) y luego responde a un problema con una solución apropiada con base en la experiencia del sistema.
- ⑩ Los sistemas de automatización de oficina personales tienden a girar alrededor de los datos y necesidades de procesamiento de negocios de cada individuo. Generalmente, dichos sistemas son desarrollados por los mismos usuarios (y operan en las computadoras personales).
- ⑪ Los sistemas de automatización de oficina de grupo se basan con frecuencia en mensajes (por ejemplo, con base en el correo electrónico) y son soluciones de menor escala a las necesidades departamentales. Como se muestra en la figura, pueden ingresar o importar datos de sistemas de proceso de transacciones más grandes.

En la empresa promedio, habrá muchos casos de cada una de estas distintas aplicaciones.

## Un marco de trabajo para la arquitectura de sistemas de información

**arquitectura de sistemas de información** Marco de referencia unificador en el que los diversos involucrados con perspectivas distintas, pueden organizar y ver los componentes fundamentales de los sistemas de información.

Se ha puesto de moda tratar con la complejidad de los sistemas de información modernos utilizando el término *arquitectura*. Los profesionales de la tecnología de información hablan de arquitecturas de datos, arquitecturas de aplicación, arquitecturas de red, arquitecturas de software y demás. Una **arquitectura de sistemas de información** sirve como marco de trabajo del más alto nivel para comprender los distintos puntos de vista de los componentes básicos de un sistema de información. Principalmente, la arquitectura de sistemas de información proporciona una base para organizar los diversos componentes de cualquier sistema de información que a usted le interese desarrollar.

Los distintos involucrados tienen diferentes perspectivas o puntos de vista de un sistema de información. Los propietarios de sistemas y los usuarios de sistemas tienden a enfocarse en tres metas del negocio comunes de cualquier sistema de información. Estas metas son establecidas generalmente en respuesta a uno o más de los impulsores del

negocio que usted leyó en el capítulo 1. Estas perspectivas orientadas a las metas de un sistema de información incluyen:

- La meta de mejorar el *conocimiento del negocio*. El conocimiento es producto de la información y los datos.
- La meta de mejorar *procesos de negocios* y servicios.
- La meta de mejorar las *comunicaciones del negocio* y la colaboración de las personas.

El papel de los diseñadores y constructores de sistemas es más técnico. Como tal, su enfoque tiende a ser ubicado más en las tecnologías que pueden ser utilizadas por el sistema de información con el fin de lograr la meta del negocio. Las perspectivas de los diseñadores y constructores del negocio de un sistema de información tienden a enfocarse más en:

- *Tecnologías de bases de datos* que soportan el crecimiento del negocio y utilizan el conocimiento del negocio.
- *Tecnologías de software* que automatizan y respaldan los procesos y servicios del negocio.
- *Tecnologías de interfaz* que respaldan la colaboración y la comunicación del negocio.

Como se muestra en la figura 2.3, la intersección de estas perspectivas (filas y columnas) define los *componentes* de un sistema de información. En la siguiente sección, describiremos todos estos componentes de sistemas de información.

**NOTA:** A lo largo de este libro se utiliza una gama consistente en pantallas que van del gris al negro para el marco de referencia y las diversas herramientas que relacionan o documentan los diversos componentes. Dicho código está basado en los componentes como sigue:

■	representa algo relacionado con el	CONOCIMIENTO
■	representa algo relacionado con los	PROCESOS
■	representa algo relacionado con las	COMUNICACIONES

Los componentes del sistema de información no existen de manera aislada. Deben estar cuidadosamente sincronizados para evitar inconsistencias e incompatibilidades dentro del sistema. Por ejemplo, un diseñador de bases de datos (un *diseñador de sistemas*) y un programador (*constructor de sistemas*) tienen sus propios puntos de vista arquitectónicos del sistema; sin embargo, estos puntos de vista deben ser compatibles y consistentes si se desea que el sistema trabaje adecuadamente. La sincronización ocurre tanto en forma horizontal (a través de una fila dada) como vertical (en forma descendente en una columna dada).

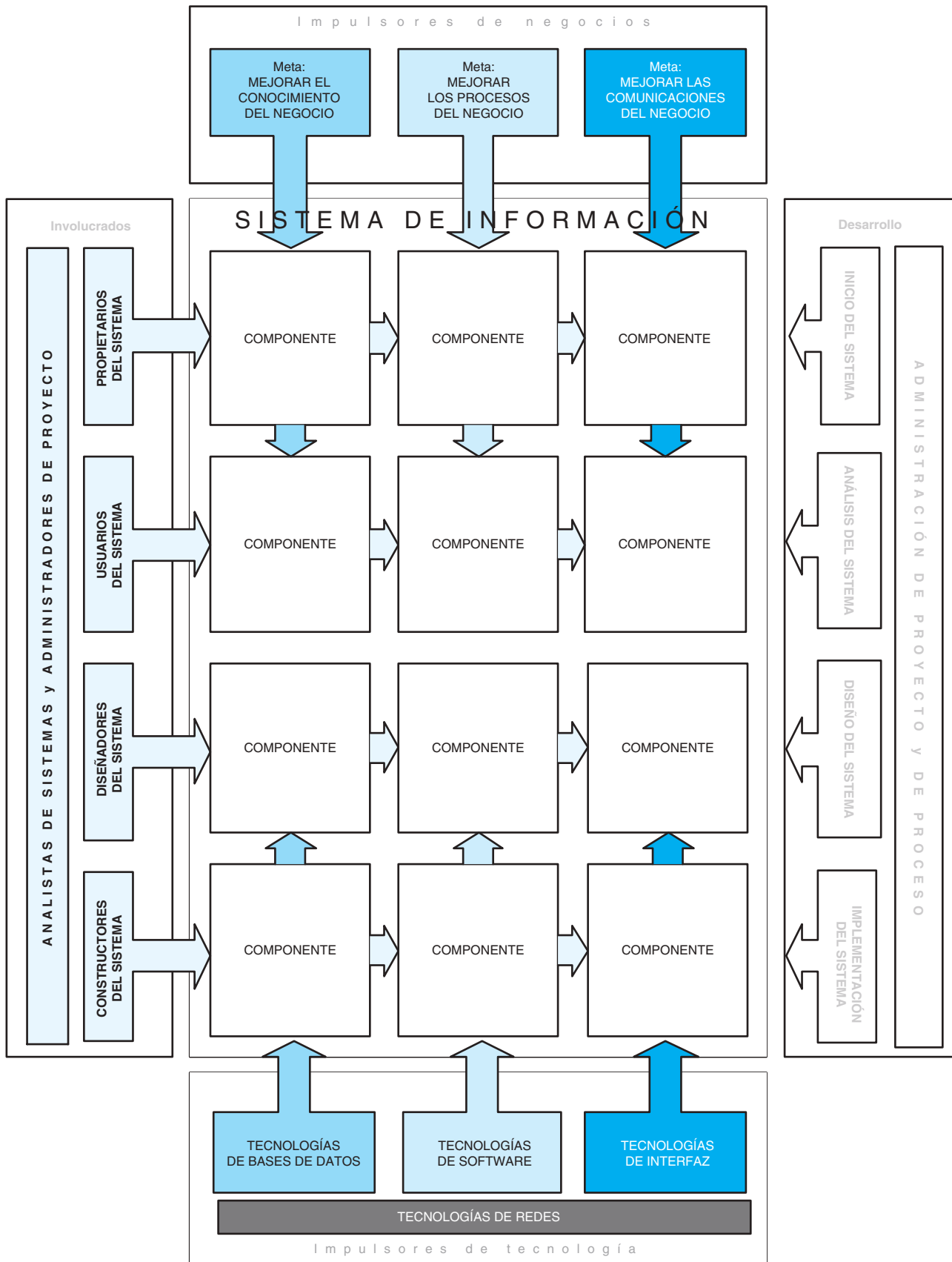
En el resto de este capítulo, analizaremos de manera breve cada enfoque y perspectiva de los **componentes** de los sistemas de información.

## > Componentes de CONOCIMIENTO

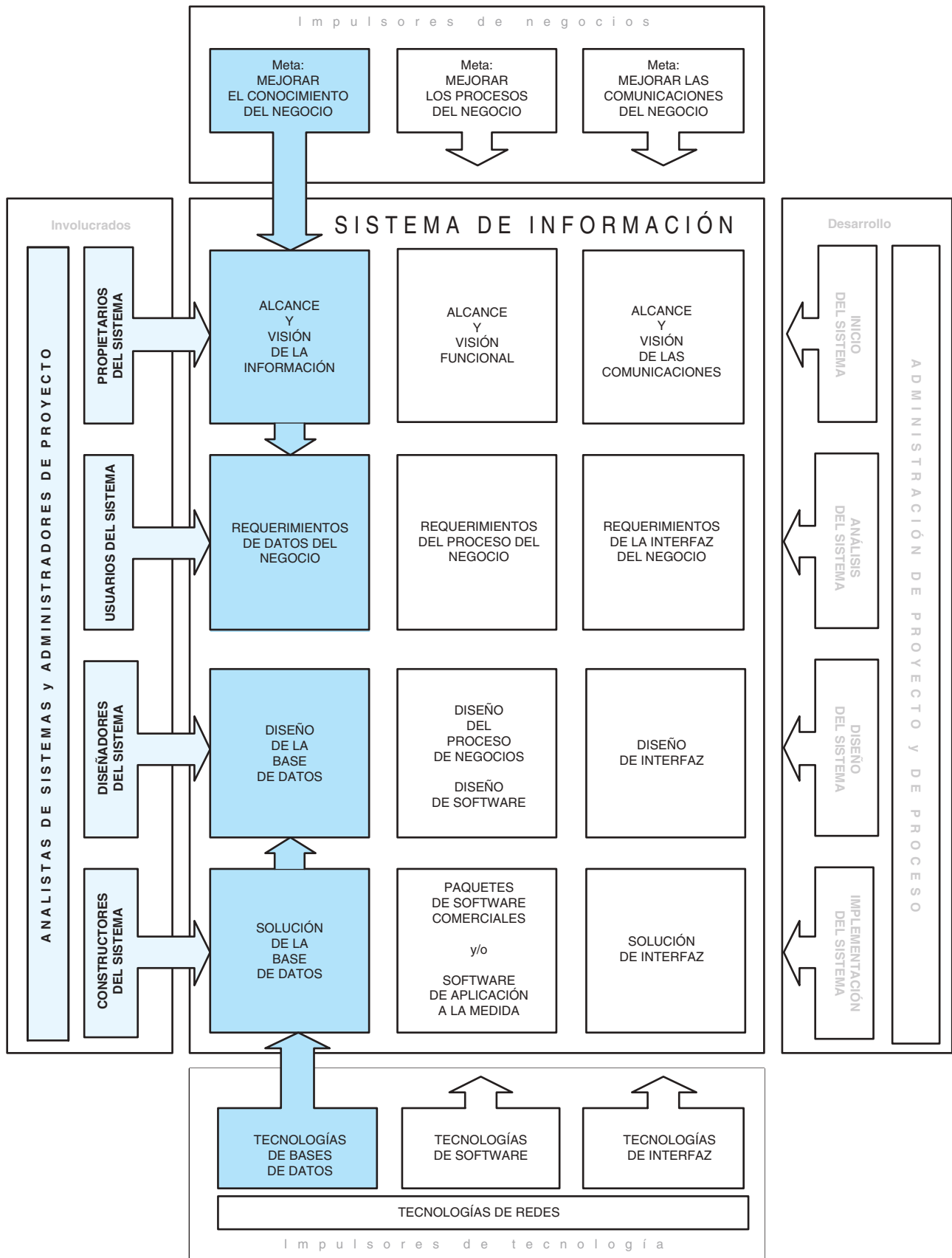
Mejorar el conocimiento del negocio es una meta fundamental para un sistema de información. Como usted aprendió en el capítulo 1, el conocimiento del negocio se deriva de los datos e información. A través del procesamiento, los datos son refinados para producir información que se convierte en conocimiento. El conocimiento permite a una compañía lograr su misión y visión.

La columna de CONOCIMIENTO de su marco de referencia se ilustra en la figura 2.4. Note que en la parte inferior de la columna del CONOCIMIENTO nuestra meta es capturar y almacenar datos del negocio a través de TECNOLOGÍAS DE BASES DE DATOS. La tecnología de bases de datos (tal como *Access*, *SQL Server*, *DB2* u *Oracle*) será utilizada para organizar y almacenar datos para todos los sistemas de información. También, al mirar la columna del CONOCIMIENTO, cada uno de nuestros distintos involucrados tiene diferentes perspectivas del sistema de información. Analicemos esos puntos de vista y revisemos su importancia para la columna del CONOCIMIENTO.

**Punto de vista de los propietarios del sistema acerca del CONOCIMIENTO** En general, el propietario del sistema no está interesado en datos sino en información que agrega nuevo conocimiento del negocio. El conocimiento y la información del negocio ayudan a



**FIGURA 2.3** Perspectivas y enfoques de los sistemas de información



**FIGURA 2.4** Perspectiva de CONOCIMIENTO DE NEGOCIOS de sistemas de información

los administradores a tomar decisiones inteligentes que respaldan la misión, las metas, los objetivos y la ventaja competitiva de la organización.

El conocimiento del negocio al principio puede tomar la forma de una simple lista de entidades y reglas del negocio. Ejemplos de entidades del negocio podrían incluir CLIENTES, PRODUCTOS, EQUIPO, EDIFICIOS, PEDIDOS Y PAGOS. ¿Qué tienen que ver las entidades del negocio con el conocimiento? Se produce información desde datos que describen estas entidades del negocio. Por tanto, tiene sentido que rápidamente debamos identificar dichas entidades acerca de las necesidades de captura y almacenamiento de datos.

También es útil entender simples asociaciones del negocio o reglas que describen la forma en que interactúan las entidades del negocio. Algunos ejemplos de reglas del negocio útiles para un sistema de ventas podrían incluir lo siguiente:

- Un CLIENTE puede colocar PEDIDOS, un pedido debe ser colocado por un CLIENTE.
- Un PEDIDO vende PRODUCTOS; un PRODUCTO puede ser vendido en un PEDIDO.

Intuitivamente, la base de datos de un sistema necesita rastrear estas entidades del negocio y reglas con el fin de producir información útil (por ejemplo, “¿El CLIENTE 2846 ha colocado algún PEDIDO no satisfecho?”).

Los propietarios del sistema están preocupados por la función general. Normalmente no están interesados en los detalles (tales como qué campos describen un CLIENTE o un PEDIDO). La función primaria de los propietarios del sistema en un proyecto de desarrollo de sistemas debe ser definir el alcance y la visión del proyecto. En cuanto al CONOCIMIENTO, el alcance se puede definir en términos simples como las entidades del negocio y reglas antes mencionadas. Los propietarios del sistema definen la visión del proyecto y las expectativas en términos de su conocimiento de los problemas, oportunidades y restricciones en lo que se relaciona con las entidades y reglas del negocio.

**Punto de vista de los usuarios del sistema acerca del CONOCIMIENTO** Los usuarios del sistema de información son conocedores de los datos que describen el negocio. Como trabajadores de la información, capturan, almacenan, procesan y editan y utilizan esos datos cada día. Con frecuencia ven los datos sólo en términos de la forma en que realmente se almacenan o cómo piensan que deben ser almacenados. Para ellos, los datos se registran en formatos, se guardan en archiveros, se registran en libros y carpetas, se organizan en hojas de cálculo o se almacenan en archivos de computadora y bases de datos. El desafío en el desarrollo de sistemas es identificar y verificar correctamente los requerimientos de datos del negocio de los usuarios. Los **requerimientos de datos** son una extensión de las entidades del negocio y reglas que fueron identificadas inicialmente por los propietarios del sistema. Los usuarios del sistema pueden identificar entidades y reglas adicionales debido a su mayor familiaridad con los datos. Más importante, los usuarios del sistema deben especificar los atributos de datos exactos que serán almacenados y las reglas del negocio precisas para mantener esos datos. Considere el siguiente ejemplo:

Un propietario del sistema puede identificar la necesidad de almacenar datos acerca de una entidad del negocio llamada CLIENTE. Los usuarios del sistema podrían decirnos que necesitamos diferenciar entre CLIENTES PROSPECTO, CLIENTES ACTIVOS Y CLIENTES INACTIVOS debido a que ellos saben qué tipos de datos ligeramente distintos describen cada clase de cliente. Los usuarios del sistema también nos pueden decir precisamente qué datos deben ser almacenados acerca de cada tipo de cliente. Por ejemplo, un CLIENTE ACTIVO podría requerir atributos de datos tales como NÚMERO DE CLIENTE, NOMBRE, DIRECCIÓN DE FACTURACIÓN, CLASIFICACIÓN DE CRÉDITO Y SALDO ACTUAL. Finalmente, los usuarios del sistema también tienen conocimientos acerca de las reglas precisas que rigen las entidades y las relaciones. Por ejemplo, ellos nos podrían decir que la clasificación de crédito de un CLIENTE ACTIVO debe ser PREFERENTE, NORMAL O A PRUEBA y que la clasificación fija para un cliente nuevo es NORMAL. También pueden especificar que solo un CLIENTE ACTIVO puede colocar un PEDIDO, pero un CLIENTE ACTIVO podría no necesariamente tener ningún PEDIDO actual en un momento dado.

Note que en el ejemplo anterior los requerimientos de datos de los usuarios pueden ser identificados en forma independiente de la TECNOLOGÍA DE BASES DE DATOS que puede ser o será utilizada para almacenar los datos. Los usuarios del sistema tienden a

### requerimientos de datos

Representación de los datos de los usuarios con base a entidades, atributos, relaciones y reglas.

enfocarse en los temas “negocios” en lo que se relaciona con los datos. Es importante que los usuarios del sistema proporcionen requerimientos de datos que sean consistentes y que complementen el alcance y la visión de la información proporcionada por los propietarios del sistema.

**Punto de vista de los diseñadores del sistema acerca del CONOCIMIENTO** La perspectiva del conocimiento de los diseñadores del sistema difiere significativamente de las perspectivas de los propietarios del sistema y los usuarios del sistema. El diseñador del sistema está más preocupado por la **TECNOLOGÍA DE BASES DE DATOS** que será utilizada por el sistema de información para respaldar el conocimiento del negocio. Los diseñadores de sistemas traducen los requerimientos de datos del negocio de los usuarios del sistema en diseños de bases de datos que posteriormente serán utilizados por los constructores de sistemas para desarrollar bases de datos de cómputo que estarán disponibles por medio del sistema de información. El punto de vista de datos de los diseñadores del sistema está restringido por las limitaciones de cualquier sistema de administración de bases de datos (DBMS, por sus siglas en inglés) que sea elegido. Con frecuencia, la elección ya ha sido hecha y los desarrolladores deben utilizar esa tecnología. Por ejemplo, muchas empresas han establecido como estándar un DBMS empresarial (tal como *Oracle*, *DB2* o *SQL Server*) y un DBMS para grupo de trabajo (tal como *Access*).

En cualquier caso, el punto de vista del diseñador del sistema acerca del conocimiento consiste en estructuras de datos, esquemas de bases de datos, campos, índices y otros componentes dependientes de tecnología. La mayoría de estas especificaciones técnicas son demasiado complejas para ser razonablemente entendidas por los usuarios del sistema. El analista de sistemas y/o el especialista en bases de datos diseñan y documentan estos puntos de vista técnicos de los datos. Este texto le enseñará herramientas y técnicas para transformar los requerimientos de datos del negocio en esquemas de bases de datos.

**Punto de vista de los constructores del sistema acerca del CONOCIMIENTO** El punto de vista final acerca del CONOCIMIENTO se refiere a los constructores del sistema. En la columna del CONOCIMIENTO de la figura 2.4, los constructores del conocimiento trabajan más cerca con la tecnología del sistema de administración de bases de datos. Deben representar los datos en lenguajes muy precisos. El lenguaje de bases de datos más comúnmente utilizado es *SQL (Structured Query Language)*. Adicionalmente, muchos sistemas de administración de bases de datos, como *Access* y *Visual FoxPro* incluyen lenguajes propietarios o facilidades para construir una nueva base de datos.

No todos los sistemas de información utilizan la tecnología de bases de datos para almacenar sus datos de negocios. Muchos sistemas heredados fueron construidos con tecnologías *flat-file* como VSAM. Estas estructuras de datos *flat-file* fueron construidas directamente dentro del lenguaje de programación utilizado para escribir los programas que usan esos archivos. Por ejemplo, en un programa *COBOL* las estructuras de datos *flat-file* se expresan como cláusulas de **IMAGEN** en una **DIVISIÓN DE DATOS**. No es la intención de este libro enseñar lenguajes de construcción de archivos planos o *flat-file*, sino sólo ponerlos en el contexto del componente del CONOCIMIENTO de los sistemas de información.

## > Componentes de PROCESO

Mejorar los procesos y servicios del negocio es otra meta fundamental de un sistema de información. Los procesos entregan la funcionalidad deseada de un sistema de información. Los procesos representan el *trabajo* en un sistema. La gente puede desempeñar algunos procesos, mientras que las computadoras y máquinas realizan otros.

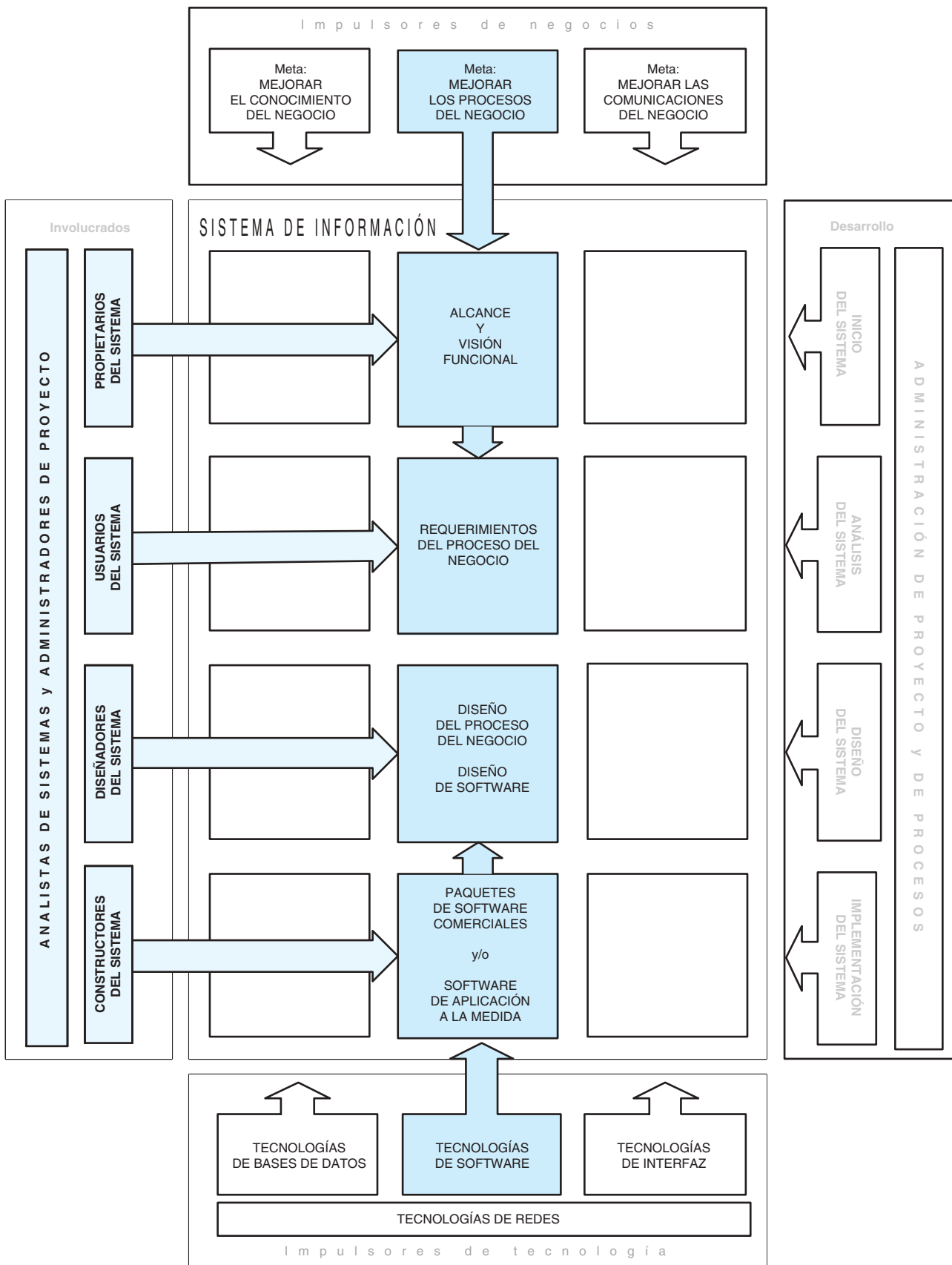
Los componentes de PROCESO de los sistemas de información se ilustran en la figura 2.5. Note que en la parte inferior de la columna de PROCESO las **TECNOLOGÍAS DE SOFTWARE** serán utilizadas para automatizar los procesos seleccionados. Si observa la columna de PROCESO, cada uno de nuestros diferentes involucrados tiene distintas perspectivas del sistema de información. Analicemos esos puntos de vista y discutamos su importancia para la columna de PROCESO.

**Punto de vista de los propietarios del sistema acerca de los PROCESOS** Como siempre, los propietarios del sistema están interesados en la función general. Tienden a enfocarse no tanto en el flujo de trabajo y los procedimientos como en las **funciones del negocio** de alto nivel, tales como las listadas al margen en la página 33. Las empresas a menudo están organizadas alrededor de estas funciones del negocio con un vicepresidente

---

### función del negocio

Grupo de procesos relacionados que brinda soporte al negocio. Las funciones se pueden descomponer en otras subfunciones y así sucesivamente hasta llegar a procesos que realizan tareas específicas.



**FIGURA 2.5** Perspectiva de PROCESO DE NEGOCIOS de sistemas de información



que supervisa cada función. A diferencia de los sucesos del negocio (tales como CLIENTE QUE REMITE UN PEDIDO) que tienen un inicio y un fin definidos, una función del negocio no tiene un tiempo de inicio o de terminación.

Históricamente, la mayoría de los sistemas de información estaban (o están) *centrados en las funciones*. Eso significa que el sistema respaldaba una función del negocio. Un ejemplo sería un SISTEMA DE INFORMACIÓN DE VENTAS que respalda sólo el proceso inicial de los pedidos de los clientes. Actualmente, muchos de estos sistemas de información de una sola función están siendo rediseñados como **sistemas de información transversal** que respaldan diversas funciones del negocio. Como una alternativa contemporánea al SISTEMA DE INFORMACIÓN DE VENTAS tradicional, un SISTEMA DE INFORMACIÓN DE CUMPLIMIENTO DE PEDIDO TRANSVERSAL también respaldaría todos los procesos importantes posteriores al proceso de pedidos del cliente. Esto incluiría llenar el pedido en el almacén, embarcar los productos al cliente, facturar al cliente y proporcionar cualquier servicio de seguimiento necesario para el cliente, en otras palabras, todos los procesos requeridos para asegurar una respuesta completa y satisfactoria para el pedido del cliente, sin importar qué departamentos participan.

Como se muestra en la figura 2.5, los propietarios del sistema visualizan los PROCESOS del negocio de un sistema en relación con el alcance funcional que se soporta por los sistemas y una visión o expectativa de mejoras. Las funciones del negocio del sistema están documentadas frecuentemente por los analistas de sistemas en términos de simples listas de sucesos del negocio y respuestas a esos sucesos. Algunos ejemplos de sucesos del negocio y respuestas son como sigue:

- Suceso: EL CLIENTE REMITE UN PEDIDO  
Respuesta: EL CLIENTE RECIBE LOS PRODUCTOS PEDIDOS
- Suceso: EL EMPLEADO REMITE UNA REQUISICIÓN DE COMPRA DE SUMINISTROS  
Respuesta: EL EMPLEADO RECIBE SUMINISTROS SOLICITADOS
- Suceso: FIN DE MES  
Respuesta: FACTURAR A LOS CLIENTES CONTRA LAS FACTURAS

Con relación a cada suceso y respuesta identificados, los propietarios del sistema identificarían problemas, oportunidades, metas, objetivos y restricciones percibidas. Los costos y beneficios de desarrollar sistemas de información para respaldar funciones del negocio también se analizarían. Como fue el caso con el CONOCIMIENTO, los propietarios del sistema no están preocupados por los detalles del PROCESO. Ese nivel de detalle se identifica y documenta como parte de la visión de procesos de los usuarios del sistema.

Los propietarios del sistema también identifican frecuentemente servicios y niveles de servicio que buscan proporcionar a clientes, proveedores y empleados. Un ejemplo popular es el *autoservicio* a clientes, proveedores o empleados. Los sistemas de recursos humanos, por ejemplo, cada vez más proporcionan a los empleados la capacidad de ingresar a sus propias transacciones como un cambio de dirección, reclamaciones médicas y solicitudes de capacitación. Los propietarios del sistema también identifican necesidades del sistema de información para mejorar el servicio y reducir errores.

Este libro le enseñará cómo identificar y documentar el alcance del proyecto en términos de funciones del negocio pertinentes, sucesos del negocio y respuestas.

**Punto de vista de los usuarios del sistema acerca de los PROCESOS** De nuevo en la figura 2.5, estamos listos para examinar el punto de vista de los usuarios de sistemas acerca de los procesos. Los usuarios están preocupados por los procesos del negocio o “trabajo”, que debe ser realizado con el fin de proporcionar las respuestas apropiadas a los sucesos del negocio. Los usuarios de sistemas especifican el proceso del negocio en términos de **requerimientos de proceso** para un nuevo sistema. Los requerimientos de proceso con frecuencia están documentados en términos de actividades, flujos de datos o flujo de trabajo.

Estos requerimientos de proceso deben ser especificados con precisión, especialmente si serán automatizados o respaldados por tecnología de software. Los requerimientos de proceso del negocio frecuentemente se definen en términos de **políticas** y **procedimientos**. Las políticas son reglas explícitas a las que debe adherirse al completar un proceso del negocio. Los procedimientos son los pasos precisos que se deben seguir al completar el proceso del negocio. Considere el siguiente ejemplo:

## FUNCIONES DE NEGOCIOS TÍPICAS

Ventas  
Servicio  
Manufactura  
Embarques  
Recepción  
Contabilidad

### sistemas de información transversal

Sistema que brinda soporte a los procesos del negocio pertinentes de varias funciones del negocio sin importar los límites organizacionales tradicionales, como las divisiones, los departamentos, los centros y las oficinas.

### requerimientos de proceso

Expectativas del usuario acerca de los requerimientos de un proceso del negocio y sus sistemas de información.

**política** Conjunto de reglas que rige la manera de realizar un proceso.

**procedimiento** Conjunto de instrucciones paso a paso y lógica para lograr un proceso del negocio.

La APROBACIÓN DE CRÉDITO es una política. Establece un conjunto de reglas para determinar si se extiende o no crédito a un cliente. La política de aprobación de crédito generalmente se aplica dentro del contexto de un procedimiento de REVISIÓN DE CRÉDITO específica que estableció los pasos correctos para revisar el crédito contra la política de crédito.

**flujo de trabajo** El flujo de transacciones a través de los procesos del negocio para garantizar que se realicen las comprobaciones y aprobaciones debidas.

Los requerimientos de proceso también son especificados frecuentemente en términos de **flujo de trabajo**. La mayoría de las empresas son muy dependientes de revisiones y autorizaciones para implementar el flujo de trabajo. Por ejemplo, una requisición de compra puede ser iniciada por cualquier empleado, pero sigue un flujo de trabajo específico de aprobaciones y revisiones antes de que se convierta en una transacción de orden de compra que se ingresa a un sistema de procesamiento de información. Desde luego, estas revisiones y estos balances pueden volverse engorrosos y burocráticos. Los analistas de sistemas y los usuarios buscan un equilibrio apropiado entre revisiones y autorizaciones, servicio y desempeño.

Una vez más, el desafío en el desarrollo de sistemas es identificar, expresar y analizar los requerimientos de procesos del negocio exclusivamente en términos de negocios que puedan ser entendidos por los usuarios de sistemas. En este texto se enseñan de manera amplia herramientas y técnicas para el modelado de procesos y la documentación de políticas y procedimientos.

**Punto de vista de los diseñadores del sistema acerca de los PROCESOS** Como fue el caso con el componente del conocimiento, el punto de vista de los diseñadores del sistema acerca de los procesos del negocio está restringido por las limitaciones de las tecnologías de desarrollo de aplicación como *Java*, *Visual Basic.NET*, *C++* y *C++*. Algunas veces el analista es capaz de elegir la tecnología de software; sin embargo, a menudo las elecciones están limitadas por estándares de arquitectura de software que especifican qué tecnologías de software y hardware deben utilizarse. En cualquier caso, el punto de vista de los diseñadores de procesos es técnico.

Dados los procesos del negocio desde el punto de vista del usuario, el diseñador debe primero determinar qué procesos automatizar y la mejor forma de hacerlo. Se dibujan modelos para documentar y comunicar cómo son o cómo serán los procesos del negocio seleccionado, implementados con el software y el hardware.

Actualmente muchas empresas adquieren paquetes de aplicación comercial del anaquele (comercial off-the-shelf, COTS) en lugar de construir ese software de manera interna. De hecho, muchas empresas afirman que el software que puede ser comprado nunca debe ser construido o que sólo el software que proporciona una verdadera ventaja competitiva debe ser construido. En el caso de adquisición de software, los procesos del negocio en general deben ser cambiados o adaptados para trabajar con el software. Por tanto, en este escenario las especificaciones de diseño de procesos del negocio deben documentar la forma en que el paquete de software será integrado a la empresa.

En forma alternativa, en el caso de construir software internamente, en general los procesos del negocio son primeramente designados. Y las especificaciones de procesos del negocio deben entonces ser soportadas con **especificaciones de software** que documentan el diseño técnico de los programas de cómputo que serán escritos. Usted puede haber encontrado algunas de estas especificaciones de software en un curso de programación. Como fue el caso con el CONOCIMIENTO, algunos de estos puntos de vista de los PROCESOS pueden entenderse por los usuarios, pero la mayoría no. La intención de los diseñadores es preparar especificaciones de software que 1) satisfagan los requerimientos del proceso del negocio de los usuarios del sistema, y 2) proporcionen suficiente detalle y consistencia para comunicar el diseño del software a los constructores del sistema. Los capítulos de diseño de sistemas en este texto enseñan herramientas y técnicas para transformar requerimientos de proceso del negocio en aplicaciones de diseño del proceso del negocio y diseño de software.

**especificaciones de software** Diseño técnico de los procesos del negocio para automatizarlos o brindarles soporte mediante programas de computadora que escriben los constructores del sistema.

**Punto de vista de los constructores del sistema acerca de los PROCESOS** Los constructores del sistema representan los PROCESOS con lenguajes de programación de cómputo precisos o ambientes de desarrollo de aplicaciones (application development environments, ADE) que describen entradas, salidas, lógica y control. Ejemplos pueden ser *C++*, *Visual Basic.NET*, *C#* (parte de, ADE *Visual Studio .NET* de Microsoft) y *Java* (disponible en ADE tales como IBM *WebSphere* y BEA *WebLogic*). Adicionalmente, algunos sistemas administrativos de aplicaciones y manejadores de bases de datos proporcionan sus pro-

pios lenguajes internos para programación. Ejemplos de esto pueden ser *Visual Basic for Applications* (en *Access*) y *PL-SQL* (en *Oracle*). Todos estos lenguajes se utilizan para escribir **programas de aplicación** personalizados que automatizan los procesos del negocio.

Este libro no intenta enseñar programación de aplicaciones. Sin embargo, mostraremos cómo algunos de estos lenguajes proporcionan un ambiente excelente para desarrollar rápidamente un sistema con un software de elaboración de prototipos. La **elaboración de prototipos** se ha vuelto la opción de técnica de diseño para muchos diseñadores y constructores. Los prototipos generalmente evolucionan hasta la versión final del sistema o la aplicación.

Como se mencionó antes, algunas veces las decisiones pueden incluir comprar un paquete de software comercial como una solución del sistema. En este escenario, el constructor del sistema puede necesitar enfocarse en una personalización que debe ser hecha al paquete de software. Puede también esperarse que desarrolle programas de aplicación que deben ser integrados con el paquete comercial para extender las capacidades funcionales del paquete. Finalmente, debe además enfocarse en los servicios del programa que deben ser escritos para ayudar con la conversión e integración del programa comercial y los sistemas existentes.

## > Componentes de COMUNICACIONES

Analicemos nuestro componente final, COMUNICACIONES. Una meta común de la mayoría de las organizaciones es mejorar las comunicaciones y la colaboración del negocio entre empleados y otros integrantes. Las mejoras de comunicación en los sistemas de información en general son dirigidas hacia dos metas de interfaz críticas para un sistema de información:

- Los sistemas de información deben proporcionar interfaces de comunicación eficaces y eficientes para los usuarios del sistema. Estas interfaces deben promover el trabajo en equipo y la coordinación de actividades.
- Los sistemas de información deben tener una interfaz eficaz y eficiente con otros sistemas de información. Con aquellos dentro del negocio y cada vez más con los sistemas de información de otros negocios.

Los componentes de COMUNICACIONES de los sistemas de información se ilustran en nuestro marco de referencia en la figura 2.6. Nótese en la parte inferior de la columna COMUNICACIÓN que utiliza las TECNOLOGÍAS DE INTERFAZ para implementar las interfaces de comunicación. Y una vez más, observe en la columna de COMUNICACIÓN, que cada uno de nuestros distintos involucrados tienen diferentes puntos de vista del sistema. Analicemos esos puntos de vista y discutamos su pertinencia para el desarrollo de sistemas.

**Punto de vista de los propietarios del sistema acerca de la COMUNICACIÓN** El punto de vista de los propietarios del sistema acerca de la COMUNICACIÓN es relativamente simple. Temprano en un proyecto de desarrollo de sistemas, los propietarios del sistema necesitan especificar:

- ¿Con qué unidades del negocio, empleados, clientes y negocios externos debe tener una interfaz el nuevo sistema?
- ¿Dónde están ubicadas estas unidades del negocio, empleados, clientes y negocios externos?
- ¿El sistema deberá tener una interfaz con cualquier otra información, computadora o sistema automatizado?

Las respuestas a estas preguntas ayudan a definir el alcance de comunicaciones de un proyecto de desarrollo de sistemas de información. Como mínimo, un adecuado punto de vista de un propietario del sistema debe incluir el alcance y la visión de comunicación, y podría ser expresado como una simple lista de ubicaciones de negocios o sistemas con los que el sistema de información debe tener una interfaz. De nuevo, los problemas relevantes, las oportunidades y las restricciones pueden ser identificados y analizados.

**Punto de vista de los usuarios del sistema acerca de la COMUNICACIÓN** El punto de vista de los usuarios acerca de la COMUNICACIÓN está dado más en términos de las entradas y salidas del sistema de información. Esas entradas y salidas pueden tomar muchas formas; sin embargo, los requerimientos de la interfaz del negocio son más importantes que el formato

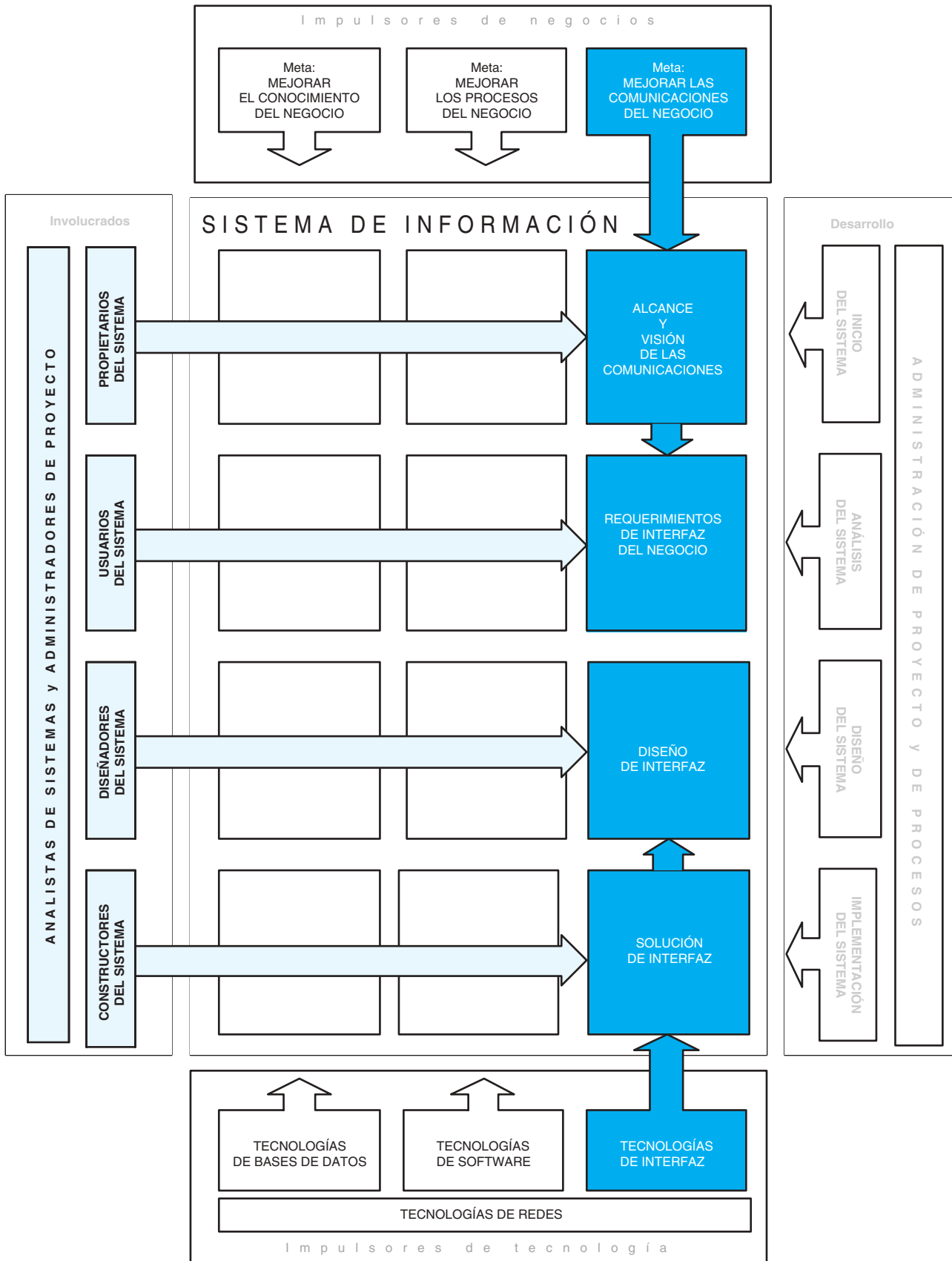
### programa de aplicación

Representación basada en un lenguaje legible para la computadora sobre lo que se supone que debe hacer un proceso de software o la forma en que éste presuntamente ha de realizar su tarea.

### elaboración de prototipos

Técnica de construcción rápida de un modelo funcional y a la vez incompleto de un sistema de información con las herramientas de desarrollo rápido de aplicaciones.





**FIGURA 2.6** Perspectiva de COMUNICACIONES DEL NEGOCIO de los sistemas de información

técnico. Las entradas y salidas representan la forma en que el sistema propuesto interactuaría con los usuarios, empleados, unidades del negocio, clientes y otros negocios.

Los detalles de esas entradas y salidas son importantes. Los usuarios del sistema podrían especificar los detalles en forma de un listado de campos (y sus valores) que constituyen las entradas o salidas. En forma alternativa, y debido a que los usuarios de sistemas se han sentido más cómodos con la interfaz de usuario gráfico (por ejemplo, *Windows* o buscadores Web) para el sistema, los detalles podrían ser especificados en forma de prototipos. Los usuarios de sistemas cada vez más exigen que sus aplicaciones de sistemas de información personalizadas tengan la misma “visión y sensación” como las herramientas favoritas de su PC como procesadores de palabras y hojas de cálculo. Esta interfaz gráfica de usuario común hace que cada nueva aplicación sea más fácil de aprender y usar.

Los métodos de lista y de prototipo para documentar el punto de vista de los usuarios del sistema acerca de la COMUNICACIÓN serán abordados en diversos capítulos de este texto.

**Punto de vista de los diseñadores del sistema acerca de la COMUNICACIÓN** Los diseñadores del sistema deben estar preocupados por el diseño técnico del usuario y de las interfaces de comunicación de sistema a sistema. A esto le llamamos **especificaciones de interfaz**. Comencemos con la interfaz de usuario.

Los usuarios y los diseñadores pueden estar involucrados en el diseño de interfaz. Pero mientras los usuarios de sistemas están interesados en los requerimientos y el formato, los diseñadores de sistemas tienen otros intereses como consistencia, compatibilidad, integridad y diálogos de usuario. El **diálogo de usuario** (a veces llamado *navegación de interfaz*) especifica la forma en que el usuario navegará a través de una aplicación para desarrollar un trabajo útil.

La tendencia hacia las interfaces de usuario gráfico (GUI, por sus siglas en inglés) como *Windows* y buscadores Web ha simplificado la vida para los usuarios de sistemas pero ha complicado el proceso de diseño para los diseñadores de sistemas. En una aplicación de *Windows* típica, existen cosas muy diferentes que los usuarios pueden hacer en un momento dado, mecanografiar algo, dar un clic con el botón izquierdo del ratón en un punto del menú o de la barra de herramientas, oprimir la tecla F1 para ayuda, maximizar o minimizar la ventana actual, cambiar a un programa diferente y muchos más. En consecuencia, el diseñador del sistema visualiza la interfaz en términos de diversos estados de sistema, sucesos que cambian el sistema de un estado a otro y respuestas a esos sucesos. Actualmente, hay muchas más decisiones de diseño y consideraciones que el diseñador de sistemas debe abordar para documentar el diálogo de una solución de interfaz de usuario gráfico. Las herramientas utilizadas para documentar los diálogos de usuario serán analizadas en la unidad de diseño de este texto.

Las interfaces Web han complicado aún más las actividades del diseñador. La sociedad ha llegado a esperar más ostentación en las interfaces Web. Por esa razón, no es del todo raro para el equipo de diseño el incluir a especialistas de diseño gráfico y de interfaz humano-computadora para asegurarse de que la interfaz para un servidor Web sea atractiva y fácil de utilizar.

Aunque no se describe en la figura 2.6, los diseñadores de sistemas modernos pueden también diseñar *interfaces “sin teclado”* como códigos de barras, reconocimiento óptico de caracteres, reconocimiento de pluma, voz o escritura. Estas alternativas reducen errores al eliminar el teclado como una fuente de errores humanos. Sin embargo, estas interfaces, al igual que las de usuarios gráficos, deben ser cuidadosamente diseñadas para explotar la tecnología y maximizar el retorno de lo que puede ser una considerable inversión.

Finalmente y como se sugirió antes, los diseñadores de sistemas también están preocupados por las interfaces de sistema a sistema. Cada vez son más difíciles de diseñar e implementar. Por ejemplo, considere un sistema de información de compras que se utiliza para iniciar y adquirir todo desde suministros hasta equipo. Un sistema de compras debe tener una interfaz con otros sistemas de información como recursos humanos (para determinar autoridad de compra y aprobación de pedidos), contabilidad (para determinar si hay fondos disponibles contra una cuenta), almacén (para determinar si los productos ordenados fueron recibidos y cuentas por pagar (para iniciar el pago)). Esos sistemas de interfaces pueden utilizar software y bases de datos muy diferentes. Esto puede complicar enormemente el diseño de interfaz de sistema. Las interfaces de sistema se vuelven cada vez más complejas cuando la interfaz está entre los sistemas de información en distintos negocios. Por ejemplo, en el sistema antes mencionado, podríamos querer permitir que nuestro sistema de compras directamente tenga una interfaz con el sistema de pedidos de un proveedor.

---

### especificaciones de interfaz

Diseños técnicos en los que se documenta la manera en que los usuarios del sistema interactuarán con él y la forma en que el sistema mismo lo hará con otros sistemas.

---

### diálogo de usuario

Especificación de la forma en cómo el usuario se mueve de una ventana o página a otra e interactúa con los programas de aplicaciones para realizar tareas útiles.

En la mayoría de los negocios, cada uno de los sistemas de información heredados fueron construidos con las tecnologías y técnicas que representaron las mejores prácticas en el momento en que fueron desarrollados. Algunos sistemas fueron construidos internamente. Otros fueron adquiridos de proveedores de software o desarrollados con consultores. Como resultado, la integración de estos sistemas heterogéneos puede ser difícil. En consecuencia, la necesidad de que sistemas distintos interactúen es constante. Asimismo, el tiempo que los diseñadores de sistemas emplean en la integración de sistema a sistema es frecuentemente tanto o más que el que emplean en el desarrollo de un sistema. La misión del diseñador del sistema es encontrar o construir interfaces entre estos sistemas que 1) no creen proyectos de mantenimiento para los sistemas heredados, 2) no comprometan las mejores tecnologías y el diseño de los nuevos sistemas y 3) sean idealmente transparentes para los usuarios de sistemas.

**Punto de vista de los constructores del sistema acerca de la COMUNICACIÓN** Los constructores del sistema construyen, instalan, prueban e implementan tanto soluciones de interfaz de usuario como de sistema a sistema por medio de TECNOLOGÍA DE INTERFAZ (vea la figura 2.6). Para interfaces de usuario, la tecnología de interfaz frecuentemente está incrustada en el *ambiente de desarrollo de aplicación (ADE)* utilizado para construir software para el sistema. Por ejemplo, un ADE como el de *Visual Studio .NET* y *Powerbuilder* incluyen toda la tecnología de interfaz requerida para construir una interfaz de usuario gráfica (GUI). Un ADE como los de *Java* y *Cold Fusion* proporcionan una funcionalidad similar para interfaces Web. En forma alternativa, la interfaz de usuario podría ser construida con una tecnología de interfaz independiente que respalda *xHTML* (por ejemplo, *Dreamweaver* de Macromedia).

Las interfaces sistema a sistema son considerablemente más complejas de construir o implementar que las de usuario. Una tecnología de interfaz de sistema a sistema que es popular actualmente es el **middleware**. El **middleware** es una capa de software utilitario que se encuentra en medio del software de aplicación y el de sistemas para integrar en forma transparente las tecnologías que difieren para que puedan interactuar.

Un ejemplo común de middleware son las herramientas de conectividad de bases de datos abierta (open database connectivity, ODBC) que permiten que los programas de aplicación trabajen con diferentes sistemas administrativos de bases de datos sin que tengan que ser reescritos para considerar los detalles y las diferencias de aquellos sistemas administrativos de bases de datos. Los programas escritos con comandos de ODBC pueden, en su mayor parte, trabajar con cualquier base de datos compatible con ODBC (que incluye docenas de distintos sistemas administrativos de bases de datos). Existen productos middleware similares para cada una de las columnas de nuestro marco de referencia de sistema de información. Los diseñadores de sistemas ayudan a elegir y aplicar estos productos para integrar sistemas.

Al momento de escribir esto, ha surgido *XML (eXtensible Markup Language)* como un estándar en evolución para la comunicación de sistema a sistema. *XML* es único en su capacidad de compartir datos entre sistemas a través de flujos de datos que no sólo incluyen los datos sino también el significado y las definiciones estructurales para esos datos. Las capacidades *XML* son la nueva frontera para el software que implementa el intercambio de datos electrónicos por la Web.

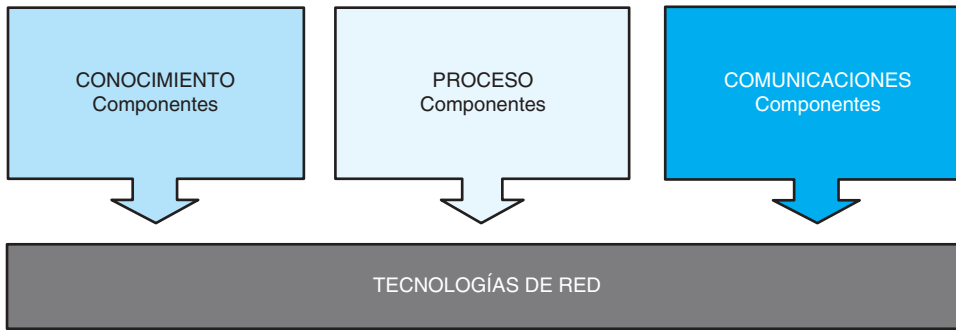
Una vez más, mencionamos que este texto no es sobre la construcción de sistemas; sin embargo, presentamos el punto de vista de los constructores de sistemas porque los demás puntos de vista de la COMUNICACIÓN llevan a la construcción de las interfaces de comunicación reales.

**middleware** Software de utilería que permite interactuar a software de aplicación y de sistemas que usan distintas tecnologías.

## Tecnologías de red y componentes de sistemas de información

En este capítulo, presentamos un marco de referencia para la arquitectura de sistemas de información que fue inicialmente inspirado por el trabajo de John Zachman.<sup>1</sup> “Framework for Information Systems Architecture” de Zachman logró un reconocimiento y uso internacional. El marco de referencia Zachman es una matriz (similar al mapa al inicio de este capítulo). Las filas corresponden a lo que Zachman llama *perspectivas* de diferentes

<sup>1</sup> John A. Zachman, “A Framework for Information Systems Architecture”, *IBM Systems Journal*, 26, núm. 3, 1987, pp. 276-292.

**FIGURA 2.7**

El papel de la red en los sistemas de información

personas que participan en el desarrollo y uso del desarrollo de sistemas. Las columnas corresponden a *enfoques* en distintos aspectos del sistema de información. La arquitectura de Zachman incluye una columna separada que se equipara muy de cerca de lo que nuestro marco de referencia reconoce como TECNOLOGÍAS DE RED. (Hemos elegido omitir esa columna debido a que el tema de redes está por lo regular cubierto en libros de texto de comunicaciones de datos y de redes y tienden a enfocarse en el marco de referencia de Interconexión de Sistemas Abiertos (Open Systems Interconnect, OSI) en contraste con el de Zachman.)

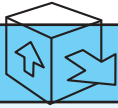
Pero indudablemente, los sistemas de información actuales están contruidos sobre las redes. En la figura 2.7 se muestra un marco de referencia moderno de sistemas de información de alto nivel que demuestra la distribución contemporánea de capas de los componentes del CONOCIMIENTO, PROCESOS y COMUNICACIÓN de un sistema de información. Los sistemas de información mejor diseñados de hoy tienden a separar estas capas y a forzarlas a comunicarse a través de la red. Este método *distribución en capas limpias* permite que cualquier componente sea reemplazado con otro con poco o ningún impacto para los demás componentes. Por ejemplo, la TECNOLOGÍA DE BASES DE DATOS, la TECNOLOGÍA DE SOFTWARE o la TECNOLOGÍA DE INTERFAZ, podría ser cambiada sin impactar los demás componentes.

No se pretende en este texto enseñar tecnologías de red. La mayoría de los programas de sistemas de información y de tecnología ofrecen cursos que pueden expandir su comprensión de la tecnología de redes.

¿Así que dónde estamos ahora? Si usted leyó el capítulo 1, aprendió acerca de los proyectos de desarrollo de sistemas de información con un enfoque en los involucrados, los procesos y los impulsores de negocios y de tecnología que influyen en la necesidad de sistemas nuevos. Si usted no lo ha leído, al menos debe hojear el capítulo 1 para aprender acerca del *contexto de los métodos de análisis y diseño de sistemas*.

En el capítulo 2, usted aprendió acerca del producto mismo, los sistemas de información, en términos de componentes básicos. Esta perspectiva de arquitectura se enfocó en los distintos puntos de vista de sistemas de información de los diversos involucrados. Usted aprendió que los propietarios y usuarios de sistemas ven a los sistemas de información desde un punto de partida de cumplimiento de metas (mejorar el conocimiento, los procesos y las comunicaciones de negocios), mientras que los diseñadores y constructores de sistemas los ven en términos de tecnología que respalda el cumplimiento de metas.

La mayoría de los lectores deben proceder directamente al capítulo 3, en el que se presenta el proceso de desarrollo de un sistema de información. Usted aprenderá acerca de la solución de problemas, metodologías y tecnología de desarrollo de los sistemas de información, al tiempo que expande su preparación en las bases para los métodos de análisis y diseño de sistemas.



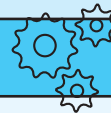
1. Las organizaciones son atendidas por una federación de sistemas de información que respaldan diversas funciones de negocios. Las empresas tienen sistemas de información de front-office que respaldan las funciones de negocios que se extienden a sus clientes y sistemas de información de back-office que respaldan las operaciones de negocios internas e interactúan con los proveedores.
2. Las variadas clases de aplicaciones de sistemas de información se traslapan e interactúan para complementarse y suplementarse entre ellas.
3. La arquitectura de sistemas de información proporciona un unificador marco de referencia en el cual diversos involucrados con distintas perspectivas pueden organizar y ver los componentes fundamentales de los sistemas de información:
  - a) Los propietarios de sistemas y los usuarios de sistemas tienden a enfocarse en tres metas de negocios comunes en cualquier sistema de información; mejoramientos en el conocimiento del negocio, procesos del negocio y comunicaciones del negocio.
  - b) Los diseñadores y los constructores de sistemas tienden a enfocarse en las tecnologías utilizadas por el sistema de información con el fin de lograr las metas del negocio. Se enfocan en tecnologías de bases de datos que respaldan el conocimiento del negocio, las tecnologías de software que respaldan los procesos del negocio y las tecnologías de interfaz que respaldan las comunicaciones del negocio.
4. Los tres puntos de vista representados en el modelo son:
  - a) CONOCIMIENTO. El conocimiento del negocio que ayuda a los administradores a tomar decisiones inteligentes.
  - b) PROCESOS. Las actividades (incluida la administración) que realizan la misión del negocio.
  - c) COMUNICACIONES. La forma en que el sistema realiza la interfaz con los usuarios y con otros sistemas de información.
5. Mejorar el conocimiento del negocio es una meta fundamental en un sistema de información:
  - a) El propietario del sistema está involucrado en la información que agrega nuevo conocimiento del negocio.
  - b) Los usuarios del sistema de información son conocedores de los datos que describen el negocio. Estos datos se utilizan para crear posteriormente información y el conocimiento del negocio.
  - c) Los diseñadores del sistema están preocupados por la tecnología de bases de datos que será utilizada por el sistema de información para respaldar el conocimiento del negocio.
  - d) Los constructores del sistema se enfocan en la tecnología de sistemas de administración de bases de datos real utilizada para almacenar los datos del negocio que respaldarán el conocimiento del mismo.
6. Mejorar los procesos del negocio es una meta fundamental de un sistema de información:
  - a) Los propietarios del sistema están interesados en las funciones del negocio y los grupos de procesos relacionados que respaldan un negocio.
  - b) Los usuarios del sistema especifican el proceso del negocio en términos de los requerimientos de proceso para un nuevo sistema. Los requerimientos de procesos del negocio son definidos frecuentemente en términos de políticas y procedimientos. Las políticas son reglas explícitas a las que hay que adherirse cuando se completan los procesos del negocio. Los procedimientos son los pasos precisos que se deben seguir al completar los procesos del negocio.
  - c) Los diseñadores del sistema ven los procesos del negocio en términos del ambiente de desarrollo de aplicación y de la tecnología de software utilizada para desarrollar el sistema. Muchas empresas compran soluciones de software comerciales en lugar de construir software internamente.
  - d) Los constructores del sistema se enfocan en los programas de aplicaciones personalizados que automatizan los procesos del negocio.
7. Una meta común de la mayoría de las organizaciones es mejorar las comunicaciones del negocio:
  - a) Los propietarios del sistema definen el alcance de las comunicaciones de un proyecto de desarrollo de sistemas de información.
  - b) Los usuarios del sistema ven las comunicaciones en términos de las entradas y salidas del sistema de información.
  - c) Los diseñadores del sistema están preocupados por el diseño técnico del usuario y de las interfaces de comunicación de sistema a sistema.
  - d) Los constructores del sistema están preocupados por la tecnología de interfaz que utilizan para implementar las interfaces de comunicación de usuario y de sistema a sistema.
8. Los sistemas de información actuales están contruidos en las redes. La tecnología de redes permite a los sistemas de información adecuadamente diseñados separar los componentes del conocimiento, PROCESO y la COMUNICACIÓN y forzarlos a comunicarse a través de la red.





## Preguntas de repaso

1. ¿Cuál es la diferencia entre los sistemas de información de front-office y los sistemas de información de back-office?
2. ¿Cómo interactúan los sistemas de procesamiento de transacciones (TPS), los sistemas de información administrativa (MIS) y los sistemas de soporte a las decisiones (DSS) entre ellos?
3. ¿Por qué necesitamos identificar la arquitectura del sistema de información?
4. ¿Cuáles son las tres perspectivas o puntos de vista orientados a las metas del negocio de un sistema de información en el que los propietarios del sistema y los usuarios del sistema tienden a enfocarse? ¿Cuáles son las tres perspectivas tecnológicas en las que los diseñadores y los constructores del sistema tienden a enfocarse?
5. ¿Cómo se relacionan las perspectivas del negocio y las perspectivas de tecnología de un sistema de información?
6. En cualquier componente dado de un sistema de información, los puntos de vista de cuatro grupos de involucrados necesitan ser considerados durante el desarrollo del sistema. ¿Cuáles son estos cuatro grupos de involucrados?
7. Describa brevemente la forma en que los diseñadores del sistema y los constructores del sistema tienden a visualizar el conocimiento en un sistema.
8. Comprender las funciones de negocios es esencial en el componente del proceso de un sistema de información. ¿Cuáles son seis funciones de negocios de alto nivel típicas de muchas compañías?
9. Si usted fuera un propietario del sistema de una tienda de discos compactos en línea, liste dos funciones de negocios de su tienda en términos de sucesos de negocios y respuestas a esos sucesos.
10. Dé un ejemplo de una política y los procedimientos necesarios para implementarla.
11. ¿Qué es la elaboración de prototipos? ¿Por qué necesitamos esa técnica?
12. ¿Cuáles son las dos metas más críticas en los componentes de comunicación?
13. ¿Cuál es el diálogo del usuario?
14. ¿Por qué el creciente uso de interfaces de usuario gráficas (GUI) ha complicado el proceso de diseño para los diseñadores de sistemas?
15. ¿Qué papel tiene la tecnología de red en el desarrollo de un sistema de información?



## Problemas y ejercicios

1. Las compañías generalmente necesitan utilizar más de un sistema de información para respaldar sus distintas funciones de negocios. Estas funciones con frecuencia son llamadas sistemas de información de front-office o sistema de información de back-office. Defina cada uno de estos dos tipos de sistemas e identifique algunas de las funciones de negocios típicas respaldadas por ellos.
2. Como analista, diseñador o constructor de sistemas, frecuentemente participará en la arquitectura de sistemas de información de su organización. ¿Qué es la arquitectura de sistemas de información? ¿Y cuál es su propósito?
3. Aunque los propietarios del sistema y los usuarios del sistema generalmente tienen distintas perspectivas del sistema de información de su organización, ambos grupos tienden a enfocarse en tres metas del negocio que son comunes para un sistema de información. ¿Cuáles son estas perspectivas orientadas a las metas? ¿Y cuál es su importancia?
4. En un sistema de información, los componentes de proceso representan el trabajo que ocurre en un sistema, que puede ser desempeñado por personas o por computadoras y máquinas. Los involucrados tienden a tener distintos puntos de vista o perspectivas de estos componentes. ¿Cuáles son estas distintas perspectivas de los involucrados en relación con los procesos y cómo difieren de cada uno?
5. Asuma que usted es un diseñador de sistemas y que su organización construye un nuevo sistema de administración de inventarios. Al revisar los requerimientos de documentación, parece que se cometió un error y que algunos elementos de datos que son necesarios para satisfacer los objetivos del negocio o técnicos del sistema de administración de inventarios se dejaron fuera. ¿Qué **no** debe usted hacer en este punto?
6. Suponga que usted diseña un sistema de punto de venta (POS) para su empresa. ¿Cuáles son las interfaces del sistema típicas de un sistema de punto de venta que necesitan ser consideradas al diseñar el sistema POS?
7. Una tecnología de negocios se vuelve más poderosa y sofisticada, muchas empresas rediseñan sus sistemas de información de función simple, como ventas, en sistemas de información transversal que proporcionan un soporte integrado por separado, para funciones de negocios separadas pero relacionadas. Suponga que usted diseña un sistema de

- administración de pedidos que integrará todas las funciones de negocios disparadas por la emisión de un pedido de ventas. ¿Qué funciones de negocios típicas serían incluidas en un sistema de información transversal?
8. El middleware frecuentemente es utilizado en proyectos de integración de sistemas cuando distintos sistemas de información están unidos para intercambiar datos vía interfaces de sistema a sistema. Defina brevemente middleware, explique sus beneficios y proporcione un ejemplo.
  9. Al identificar y documentar los requerimientos del negocio, los analistas del sistema necesitan ser capaces de distinguir entre leyes, políticas y procedimientos. ¿Por qué es esto importante?
  10. ¿Es común que los propietarios del sistema y los usuarios del sistema tengan muy distintos puntos de vista de los mismos procesos del negocio utilizados en un sistema de información? ¿Por qué cree usted que esto sucede? Considere una aerolínea que desarrolla un sistema de autoservicio de registro de clientes en los aeropuertos. ¿Cuál cree usted que sea la perspectiva de los dueños del sistema? ¿Y de los usuarios del sistema? Dé ejemplos de cómo los procesos del negocio del sistema de registro de una aerolínea serán vistos por los dueños del sistema y los usuarios del sistema.
  11. Los diseñadores del sistema y los constructores del sistema también tienden a tener muy distintos puntos de vista acerca de los componentes del sistema. Explique las distintas formas en que los diseñadores y los constructores podrían visualizar los componentes de comunicación al usar el escenario de autoservicio de registro de clientes descrito en la pregunta anterior.
  12. Los diseñadores del sistema con frecuencia tienen diversas opciones de diseño técnicas para elegir cuando diseñan interfaces entre distintos sistemas y aplicaciones. ¿Qué deben tener en mente siempre los diseñadores al planear estas interfaces?
  13. El marco de referencia para la arquitectura de sistemas de información utilizado en este libro de texto se deriva de un marco de referencia pionero desarrollado por John Zachman. ¿Cuál es una de las ventajas de diseñar sistemas con base en este marco de referencia u otros similares?
  14. En ocasiones, una organización puede elegir adquirir un paquete de software comercial (COTS). ¿Cuáles cree usted que sean las ventajas y desventajas de utilizar aplicaciones comerciales en comparación con las aplicaciones personalizadas?
  15. Si una organización elige un paquete de COTS como su solución, ¿el punto de vista del constructor del sistema sería el mismo que para una aplicación personalizada? Si no es así, ¿en qué forma sería distinto?

## Proyectos e investigación



1. Seleccione una organización de mediana a grande. La organización puede ser del sector público o privado y puede ser una con la que usted esté personalmente familiarizado o una en la que la información esté disponible.
  - a) Describa la organización que ha elegido (tipo de organización, misión, productos o servicios, tamaño, ventas o ingresos anuales, etcétera).
  - b) Seleccione uno de los sistemas de información más importantes que la organización utiliza o desarrolla y descríbalos.
  - c) En la organización que usted eligió, ¿típicamente quién sería el propietario del sistema?
  - d) Describa, desde el punto de vista del propietario, la información producida por este sistema.
  - e) Si la organización iniciara un proyecto para reemplazar o modificar este sistema, ¿cómo podría el propietario del sistema definir el alcance y la visión del proyecto dentro del contexto de la organización que usted eligió?
  - f) ¿Quiénes son los usuarios típicos de este sistema?
  - g) Describa, desde la perspectiva de los usuarios, la información producida por este sistema.
  - h) ¿Cuál es una diferencia esencial en la forma en que los dueños y usuarios del sistema visualizan la información producida por el sistema?
2. Contacte y entreviste a dos o tres analistas de sistemas, en distintas organizaciones de ser posible, en relación con el tema tratado en este capítulo acerca de los componentes de la comunicación.
  - a) Describa la naturaleza del analista de cada compañía u organización, su misión y sus problemas o necesidades actuales.
  - b) ¿Qué tan importante considera cada analista de sistemas las comunicaciones con los usuarios del sistema contra los constructores del sistema?; es decir, ¿cuál es más importante y por qué?
  - c) ¿Les parece más difícil comunicarse con los usuarios del sistema o con los constructores del sistema? ¿Por qué?
  - d) Si fueran CIO por un día, ¿qué cambiaría cada uno de ellos acerca de la forma en que los diseñadores se comunican con los usuarios y los constructores de sistemas?

- e) ¿Con qué puntos de vista de las personas que entrevistó está usted de acuerdo o con cuáles está en total desacuerdo? Justifique su respuesta.
3. Seleccione un sistema de información utilizado por una organización de mediana a grande. Puede ser una con la que usted esté personalmente familiarizado o alguna cuya estructura organizacional y sistema de información usted haya investigado.
- ¿Cuál es la naturaleza de la organización que usted ha elegido, su misión y el propósito de alto nivel de su sistema de información?
  - ¿Quién es el propietario del sistema?
  - Si usted fuera el propietario del sistema, describa cómo vería los procesos del sistema desde ese punto de vista.
  - ¿Quiénes son los usuarios del sistema?
  - Si usted fuera uno de los usuarios del sistema, describa cómo vería usted los procesos del sistema desde ese punto de vista.
  - Si usted fuera el diseñador del sistema, describa los procesos del sistema desde ese punto de vista.
  - ¿Cuáles son las diferencias esenciales en los puntos de vista?
4. Imagine que usted es el propietario de un negocio pequeño y que busca en la Web una compañía que pueda proveer productos o servicios necesitados por su negocio. Encuentre varios sitios de empresa a empresa (B2B) que ofrezcan productos o servicios que usted busca para su negocio. Familiarícese con sus sitios Web desde el punto de vista de un cliente de negocios típico que visita esos sitios por primera vez.
- ¿Cuál es la naturaleza de su organización y qué tipo de productos o servicios busca usted?
  - ¿Qué sitios B2B encontró usted en la Web?
  - Compare los diferentes sitios. Si todas las demás cosas fueran iguales (precio, disponibilidad, marcas ofrecidas, etcétera), ¿sería más probable que usted adquiriera productos o servicios de uno y no de otro sólo por diferencias en sus sitios Web? ¿Por qué sí o por qué no?
  - Desde el punto de vista de un cliente de negocios, ¿qué cree que es más importante para un sitio Web, el diseño o su funcionalidad? Explique su respuesta.
- e) Desde el punto de vista de un consumidor, ¿su respuesta sería la misma que en la pregunta anterior? Explique.
5. Investigue varios artículos publicados en los últimos años en su biblioteca y/o en la Web que analicen temas éticos relacionados con el diseño de sistemas.
- ¿Qué artículos pudo encontrar?
  - Describa algunas de las situaciones y temas éticos que podrían surgir de tiempo en tiempo en el diseño de sistemas.
  - Elija una de las situaciones descritas en el inciso b) y describa la que usted cree que es la obligación ética del diseñador del sistema, si la hay.
  - (Crédito adicional) ¿Cree usted que requerir a los profesionales de TI, es decir, analistas, diseñadores y constructores de sistemas que sean licenciados o certificados aumentaría el profesionalismo y/o reducirían el comportamiento poco ético? ¿Por qué sí o por qué no?
6. El libro de texto utiliza un marco de referencia para describir la arquitectura de sistemas de información que está basada en el modelo “Framework for Information Systems Architecture” de John Zachman. Por medio de la Web o de su biblioteca escolar, investigue otros marcos de trabajo para describir las arquitecturas de IS y elija uno, tal como el Open Systems Interconnect (OSI).
- ¿Qué marcos de trabajo encontró y cuál eligió?
  - Describa su enfoque a la arquitectura de sistemas de comunicación. Incluya un diagrama si es aplicable.
  - ¿Cuáles son las similitudes con el marco de referencia que usted utilizó en el libro de texto?
  - ¿Cuáles son sus diferencias?
  - Si usted fuera el propietario del sistema, ¿cuál encontraría más fácil de entender?
  - Si usted fuera un analista de sistemas, ¿cuál encontraría más fácil de entender?



## Casos breves

1. Un administrador de TI solicita una cantidad de fondos para mejorar el servidor de correo electrónico. Sin la mejora necesaria, el servidor estará sobrecargado por la gran cantidad de correo electrónico y correrá el riesgo de colapsarse. El administrador de la empresa niega la solicitud, al citar la confiabilidad pasada del servidor y expresa su preocupación por gastos grandes recientes de TI. El administrador de la empresa deja la conversación

y se pregunta qué inversiones de TI son realmente necesarias y si el administrador de TI sólo desea crear una “seguridad laboral”.

Igualmente, el administrador de TI sale de la junta frustrado por no tener las herramientas que él o ella necesita para hacer su trabajo adecuadamente. El administrador de TI sabe que cuando el servidor colapse, será su responsabilidad arreglarlo.

- a) ¿Cree usted que esto sucede con frecuencia en las empresas?
  - b) ¿Qué perspectivas cree usted que cada uno toma en este problema?
  - c) ¿Cómo podía cada uno de ellos haber comunicado esas perspectivas y necesidades del negocio de una mejor manera?
2. Entreviste al menos a una persona de marketing, servicios a clientes y contabilidad/nóminas en la misma compañía. ¿Qué tipos de información manejan? ¿Comparten información a través de los departamentos? ¿Nota usted un traslape en información o en entradas de datos?
  3. Los departamentos de servicios gubernamentales están profundamente sobrecargados por la cantidad de datos que tienen y que procesan. Entreviste a alguien de un departamento de servicios y prepare un breve ensayo. Los departamentos de servicio que deben examinar a fondo una gran cantidad de datos son los que tratan con, por ejemplo, personas extraviadas, servicios de protección de infantes, departamento de control vehicular y rastreo de personas con libertad bajo fianza después de un crimen. Usted debe incluir pero no limitarse a temas como:
    - ¿Cuál es el trabajo del departamento (o la persona)?
    - ¿Qué tipo de datos recopilan y analizan?
    - ¿Qué tipo de análisis hacen a los datos?
    - ¿Cuánta información recolectan, de quién y qué programas utilizan?
  4. La tienda de abarrotes de su barrio, Wow Grocery, siempre parece quedarse sin su helado favorito. Con frustración, usted le pregunta al gerente de la tienda por qué siempre se les termina. El gerente de la tienda, Bob, le dice que una tienda pequeña no puede permitirse un sistema de administración de inventario, así que el inventario se actualiza en forma manual. Esto significa que con frecuencia la tienda debe tener cantidades adicionales de los artículos muy buscados o arriesgarse a quedarse sin ellos. Desafortunadamente, Wow Grocery no tiene un congelador lo suficientemente grande para almacenar suministros adicionales de helado. Como resultado, la tienda se queda sin helado con bastante frecuencia.
  5. ¿Qué puede hacer Wow Grocery para automatizar o administrar sus sistemas de inventarios sin gastar mucho dinero? Haga un borrador de su solución en un breve escrito.

## Ejercicios de equipo e individuales



1. ¿Cómo resuelve usted un problema al parecer irreparable? Como equipo, desarrolle una metodología para resolver la siguiente pregunta: ¿Cuántas casas hay en Estados Unidos que estén pintadas de amarillo?
2. Intente algo que usted no haya hecho antes (legal, no peligroso y de clasificación G). Comparta con la clase lo que haya hecho y por qué. ¿Por qué es importante experimentar cosas nuevas?
3. Comparta con su equipo un incidente poco ético que usted haya observado. ¿Cómo le afectó directamente ese incidente? ¿Qué impacto indirecto tuvo en los demás?

## Lecturas recomendadas

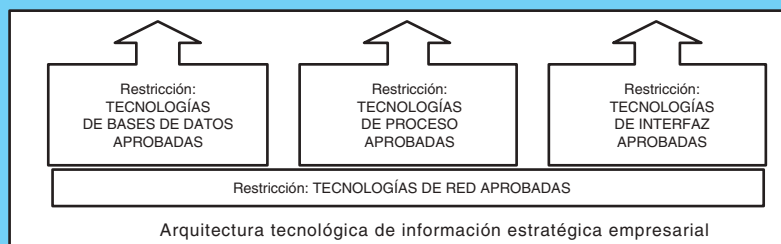
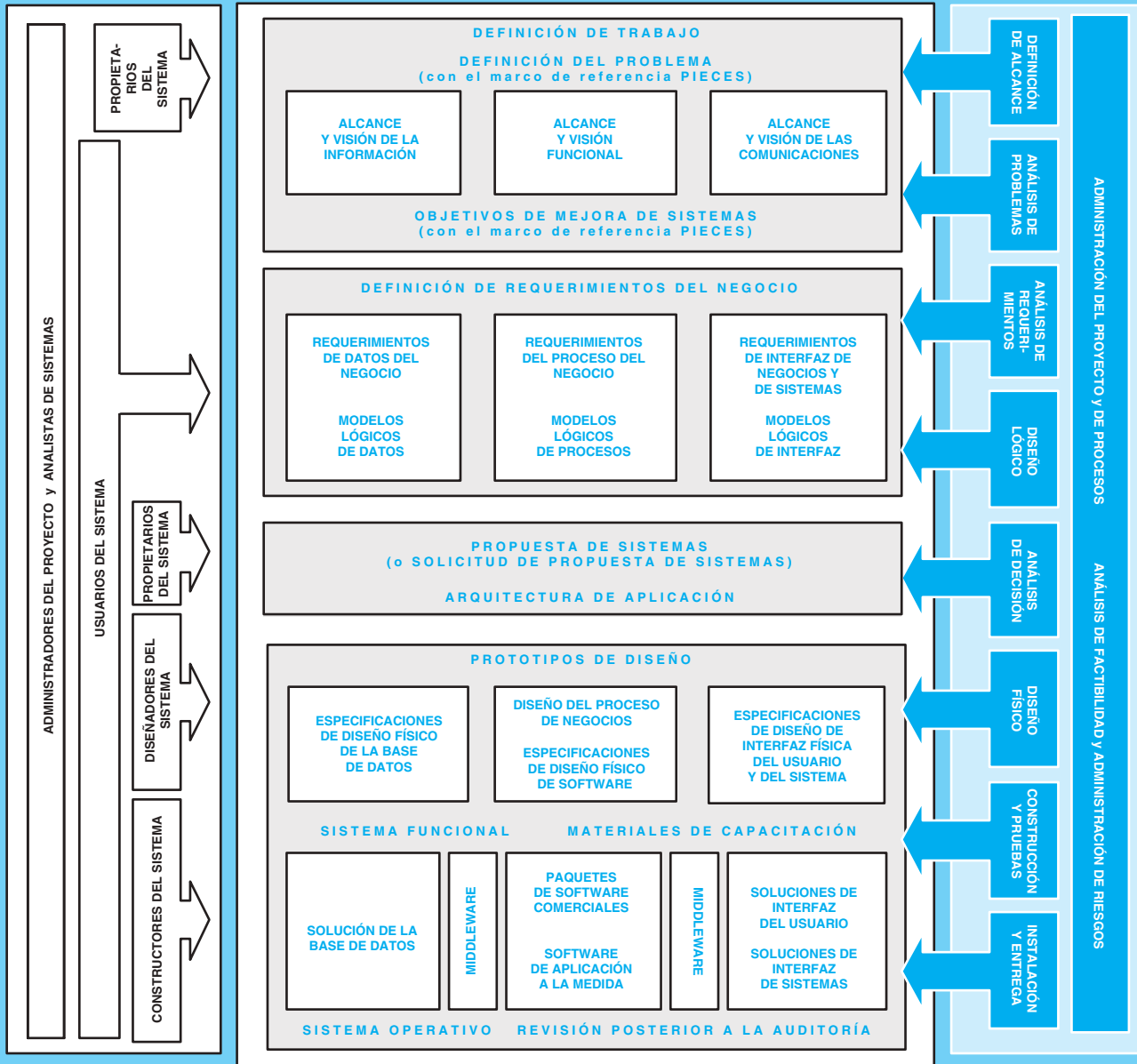
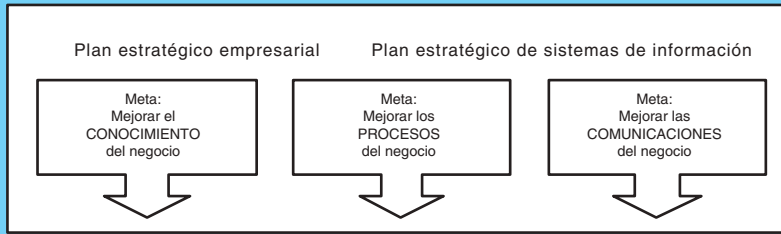


- Galitz Wilbert O. *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques*, 2a. ed., Nueva York, John Wiley & Sons, 2002.
- Goldman, James E., Phillip T. Rawles y Julie R. Mariga. *Client/Server Information Systems: A Business-Oriented Approach*. Nueva York, John Wiley & Sons, 1999. Para estudiantes que están en busca de una introducción orientada a ellos acerca de la arquitectura de tecnología de información y comunicaciones de datos, recomendamos el libro de nuestros colegas debido a que fue escrito para especializaciones en negocios y sistemas de información para proporcionar una revisión completa de la tecnología que respalda los sistemas de información actuales.
- Inmon, W.H. *Building the Data Warehouse*, 3a. ed. Nueva York, John Wiley & Sons, 2002.
- Sethi, Vikram y William R. King. *Organizational Transformation through Business Process Reengineering*. Upper Saddle River, NJ, Prentice Hall, 1998.
- Taylor, David y Alyse D. Terhune. *Doing E-Business: Strategies for Thriving in an Electronic Marketplace*. Nueva York: John Wiley & Sons, 2000.
- Zachman, John A. "A Framework for Information System Architecture". *IBM Systems Journal* 26, núm. 3 (1987). Adaptamos el modelo matriz para los componentes de sistemas de información del marco de referencia conceptual del señor Zachman. Primero encontramos a John Zachman en el circuito de lectura, donde él da una plática sorprendentemente informativa y entretenida acerca del mismo tema de este artículo. El marco de referencia del señor Zachman ha obtenido reconocimientos profesionales y ha inspirado al menos una conferencia acerca de su modelo. Su marco de referencia está basado en el concepto de que la arquitectura significa algo diferente para cada persona y sugiere que los sistemas de información consisten de tres distintos puntos de vista "orientados a los productos", datos, procesos y tecnología (que nosotros renom-

bramos como *comunicaciones*), a los que nosotros agregamos un cuarto punto de vista, “interfaz”. El marco de referencia de Zachman ofrece seis distintos puntos de vista para el público específico, para cada uno de esos puntos de vista del producto, los puntos de vista general y del propietario (que

nosotros renombramos como *puntos de vista del propietario y del usuario*, respectivamente), puntos de vista del diseñador y del constructor (que nosotros combinamos en nuestro *punto de vista del diseñador*) y un punto de vista fuera de contexto (que nosotros llamamos *punto de vista del constructor*).





# 3

## Desarrollo de sistemas de información

### Panorámica y objetivos del capítulo

Este capítulo analiza más de cerca el proceso de desarrollo de sistemas que se presentó en el capítulo 1. El desarrollo de sistemas exitoso está regido por algunos principios fundamentales que presentamos en este capítulo. También presentamos una metodología de desarrollo de sistemas típica como una estrategia que requiere disciplina para desarrollar sistemas de información. Aunque dicha estrategia no garantizará el éxito, sí mejorará en gran medida las oportunidades de éxito. Usted sabrá que comprende el desarrollo de sistemas de información cuando pueda:

- Describir la motivación de un proceso de desarrollo de sistemas estándar en términos del Modelo de Madurez de Capacidad (Capability Maturity Model, CMM) para realizar una administración con calidad.
- Diferenciar entre el ciclo de vida del sistema y una metodología de desarrollo de sistemas.
- Describir 10 principios básicos del desarrollo de sistemas.
- Definir problemas, oportunidades y directrices (los triggers de los proyectos de desarrollo de sistemas).
- Describir el marco de referencia PIECES para clasificar problemas, oportunidades y directrices.
- Describir las fases esenciales del desarrollo de sistemas. En cada fase, describir su propósito, entradas y salidas.
- Describir las actividades transversales del ciclo de vida que se traslapan en múltiples fases del desarrollo de sistemas.
- Describir las “rutas” alternativas típicas, a través de las fases esenciales del desarrollo de sistemas. Describir cómo las rutas pueden combinarse o personalizarse para distintos tipos de proyectos.
- Describir diversas herramientas automatizadas para el desarrollo de sistemas.

## Introducción

El trabajo va en marcha en SoundStage Entertainment Club para el análisis y diseño de sistemas de su sistema de información de servicios de membresía. Pero entre más aprende Bob Martínez acerca del sistema, más confundido está. Bob recuerda algunas de sus tareas de programación en la universidad. La mayoría de ellas eran sólo una página o dos de listados con viñetas de los puntos con las características requeridas. Era bastante fácil hacer eso. Pero el nuevo sistema de SoundStage incluirá rastrear contactos de miembros y solicitudes de compras, promociones, ventas, embarques, inventarios, almacenes múltiples, sitios Web y más. Bob se pregunta cómo podrán listar todos los requerimientos, no sólo tenerlos en orden. ¿Cómo sabrán qué datos necesitan rastrear? ¿Cómo sabrán lo que cada pieza de programación necesita hacer? Le mencionó eso a su jefa, Sandra. Ella dijo que todo se trataba de seguir “la metodología”. Él recordó algo acerca de la metodología de una clase de análisis de sistemas. En ese tiempo parecía como mucho trabajo innecesario. Pero ahora comienza a ver que en un proyecto grande, seguir un método establecido puede ser el único camino seguro a recorrer.

## El proceso de desarrollo de sistemas

**proceso de desarrollo de sistemas** Conjunto de actividades, métodos, mejores prácticas, productos y herramientas automatizadas que los interesados (del capítulo 1) usarán para desarrollar y mejorar software y sistemas de información (de los capítulos 1 y 2).

En este capítulo se presenta un enfoque en el desarrollo de sistemas de información. Analizaremos un **proceso de desarrollo de sistemas**. Nótese que no dijimos “el” proceso; hay tantas variaciones en procesos como hay expertos y autores. Presentaremos un proceso típico y lo utilizaremos en forma consistente a través de este libro. La página de inicio del capítulo muestra un número expandido de fases comparado con la página de inicio del capítulo 1. Esto es porque hemos dividido las fases de alto nivel, como análisis y diseño de sistemas del capítulo 1 en múltiples fases y actividades. También hemos refinado el tamaño y lugar de los papeles de los interesados para que reflejen “participación” en contraste con el énfasis o la prioridad. Y hemos editado y mejorado los componentes para indicar productos y herramientas del desarrollo de sistemas. Todas estas modificaciones se explicarán a su debido tiempo.

¿Por qué las organizaciones se enfocan en procesos estandarizados para desarrollar sistemas de información? Los sistemas de información son un producto complejo. Recuerde del capítulo 2 que un sistema de información incluye componentes y tecnologías de datos, procesos y comunicaciones, que deben atender las necesidades de una diversidad de interesados. Tal vez esto explique por qué tanto como 70 por ciento o más de los proyectos de desarrollo de sistemas de información fracasan en cumplir con las expectativas, cuestan más de lo presupuestado o son entregados mucho tiempo después de lo prometido. El Gartner Group sugiere que “una adherencia consistente a lineamientos de una metodología moderadamente rigurosa puede proporcionar a 70 por ciento de las organizaciones [desarrollo de sistemas] una mejora de productividad de al menos 30 por ciento en dos años”.<sup>1</sup>

Cada vez más, las organizaciones no tienen otra elección más que adoptar y seguir un proceso estandarizado de desarrollo de sistemas. Primero, utilizar un proceso consistente de desarrollo de sistemas crea eficiencias que permiten a la administración compartir recursos entre proyectos. Segundo, una metodología consistente produce documentación consistente que reduce los costos del tiempo de vida para mantener los sistemas. Finalmente, el gobierno estadounidense ha ordenado que cualquier organización que busque desarrollar software para el gobierno, debe apearse a ciertos requerimientos de administración de calidad. Un proceso consistente promueve la calidad. Y muchas otras organizaciones se han comprometido agresivamente con metas de administración de calidad total con el fin de aumentar su ventaja competitiva. Con el fin de realizar mejoras de calidad y de productividad, muchas organizaciones han cambiado a marcos de referencia de administración de procesos y de proyectos como el Modelo de Madurez de la Capacidad, analizado en la siguiente sección.

<sup>1</sup> Richard Hunter, “AD Project Portfolio Management”, en *Proceedings of the Gartner Group IT98 Symposium/Expo* (CD-ROM). El Gartner Group es un observador de la industria de TI y un grupo de investigación que investiga y proyecta las tendencias de la industria en cuanto a los gerentes de TI.



## > Modelo de Madurez de la Capacidad

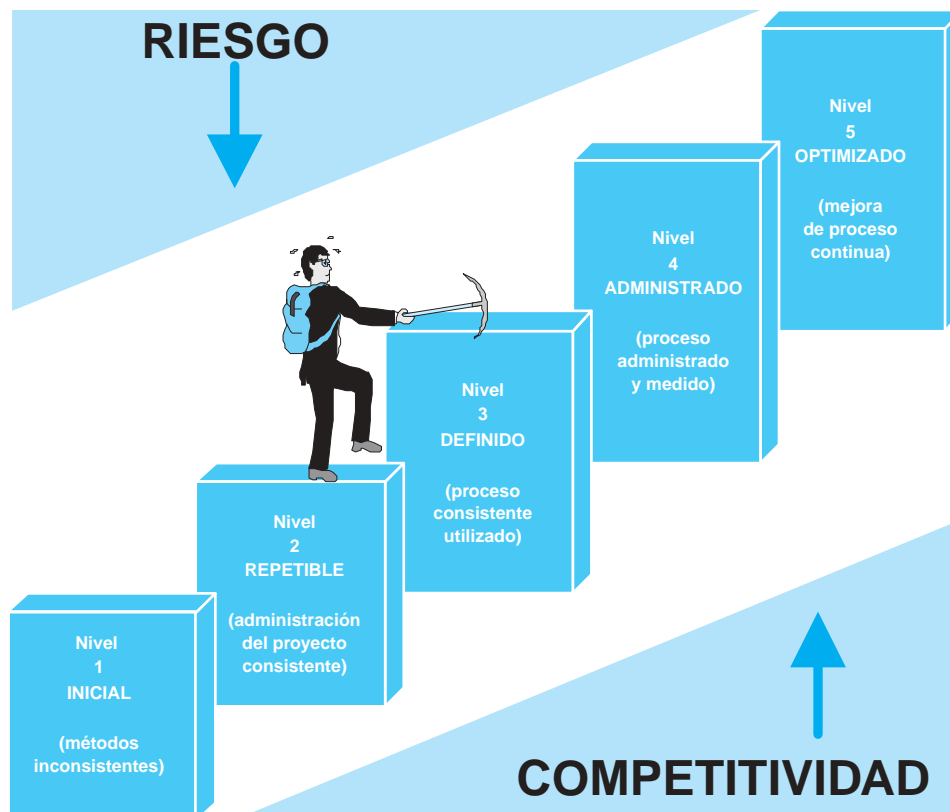
Se ha demostrado que conforme un proceso de desarrollo de sistemas de información de una organización madura, la duración y el costo del proyecto disminuyen al tiempo que la productividad y la calidad aumentan. El Software Engineering Institute en Carnegie Mellon University ha observado y medido este fenómeno y desarrolló el **Modelo de Madurez de la Capacidad (CMM)** para ayudar a todas las organizaciones a lograr estos beneficios. El CMM ha conseguido una gran aceptación, tanto de la industria como del gobierno. Una evaluación de software basada en CMM está siendo utilizada para evaluar a contratistas de tecnología de la información para la mayoría de los proyectos gubernamentales federales estadounidenses.

Con el marco de referencia de CMM para sistemas y software se tiene la intención de ayudar a las organizaciones a mejorar la madurez de sus procesos de desarrollo de sistemas. El CMM está organizado en cinco niveles de madurez (véase la figura 3.1):

- **Nivel 1, Inicial.** Esto a veces es llamado anarquía o caos. En este nivel, los proyectos de desarrollo de sistemas no siguen un proceso consistente. Cada equipo de desarrollo utiliza sus propias herramientas y métodos. El éxito o el fracaso generalmente es una función de la habilidad y experiencia del equipo. El proceso es impredecible y no es repetible. Un proyecto en general encuentra muchas crisis y con frecuencia está por encima del presupuesto y retrasado en el programa. La documentación es esporádica o no es consistente de un proyecto al siguiente, por lo tanto crea problemas para los que deben mantener un sistema durante su tiempo de vida. Casi todas las organizaciones empiezan en el nivel 1.
- **Nivel 2, Repetible.** En este nivel los procesos y las prácticas de administración de proyectos están establecidos para rastrear costos, programas y funcionalidad del proyecto. El enfoque está en la administración del proyecto. Un proceso de desarrollo de sistemas se sigue siempre, pero puede variar de proyecto a proyecto. El éxito o el fracaso está todavía en función de la habilidad y experiencia del equipo del proyecto; sin embargo, se hace un esfuerzo concertado por repetir éxitos de proyectos anteriores. Las prácticas de administración de proyectos eficaces establecen la base para procesos estandarizados en el siguiente nivel.

### Modelo de Madurez de la Capacidad (CMM)

Marco estándar de referencia para evaluar el nivel de madurez del desarrollo de sistemas de información de una organización y sus procesos y productos administrativos. Consiste en cinco niveles de madurez.



**FIGURA 3.1**

Modelo de Madurez de la Capacidad (CMM)

**TABLA 3.1** Impacto del “proceso” de desarrollo de sistemas en la calidad

Estadísticas del proyecto CMM para un proyecto que resulta en 200 000 líneas de código

Nivel CMM de la organización	Duración del proyecto (meses)	Personas en el proyecto por mes	Número de defectos entregados	Costo medio (en millones de dólares)	Costo más bajo (en millones de dólares)	Costo más alto (en millones de dólares)
1	30	600	61	5.5	1.8	100+
2	18.5	143	12	1.3	.96	1.7
3	15	80	7	.728	.518	.933

Fuente: Master Systems, Inc.

**metodología de desarrollo de sistemas** Estrategia formalizada del proceso de desarrollo de sistemas; un proceso estandarizado que incluye actividades, métodos, mejores prácticas, productos y herramientas automatizadas que se usan en el desarrollo de sistemas de información.

- **Nivel 3, Definido.** En este nivel se adquiere o se desarrolla un proceso de **desarrollo de sistemas** estándar (a veces llamado *metodología*). Todos los proyectos utilizan una versión personalizada de este proceso para desarrollar y mantener sistemas de información y software. Como resultado de utilizar el proceso estandarizado para todos los proyectos, cada proyecto resulta en documentación y productos consistentes y de alta calidad. El proceso es estable, predecible y repetible.
- **Nivel 4, Administrado.** En este nivel se establecen metas mensurables para la calidad y la productividad. Las mediciones detalladas del proceso de desarrollo de sistemas estándar y la calidad del producto se recolectan y almacenan rutinariamente en una base de datos. Hay un esfuerzo por mejorar la administración de proyectos individuales basado en estos datos recopilados. Por tanto, la administración busca volverse más proactiva que reactiva a los problemas de desarrollo de sistemas (como elevación de costos, ampliación del alcance, retrasos de programa, etcétera). Incluso cuando un proyecto encuentra problemas o asuntos inesperados, el proceso puede ser ajustado con base en impactos predecibles y medibles.
- **Nivel 5, Optimizado.** En este nivel el proceso de desarrollo del sistema estandarizado es vigilado continuamente y mejorado con base en medidas y análisis de datos establecidos en el nivel 4. Esto puede incluir cambiar la tecnología y las mejores prácticas utilizadas para realizar actividades requeridas en el proceso estándar de desarrollo del sistema, así como ajustar el proceso mismo. Las lecciones aprendidas se comparten a través de la organización, con un énfasis especial en la eliminación de ineficiencias en el proceso de desarrollo del sistema al tiempo que se mantiene la calidad. En resumen, la organización ha institucionalizado una mejora continua del proceso de desarrollo de sistemas.

Es muy importante reconocer que cada nivel es un requerimiento previo para el siguiente nivel.

Actualmente, muchas organizaciones están trabajando fuerte para lograr al menos un nivel 3 de CMM (a veces impulsado por un mandato gubernamental u organizacional). Un tema central para lograr el nivel 3 (definido) es el uso de un proceso o una metodología estándar para construir o integrar sistemas. Como se muestra en la tabla 3.1, una organización puede realizar mejoras significativas en programa y costo al institucionalizar las mejoras de proceso de nivel 3 de CMM.<sup>2</sup>

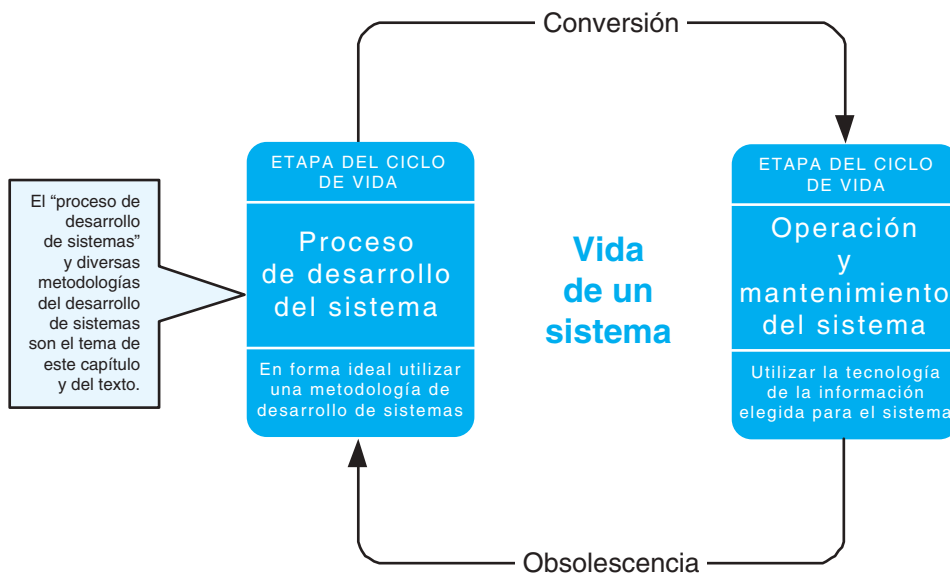
### ciclo de vida del sistema

El ciclo de vida de un sistema de información se divide en dos etapas: 1) desarrollo de sistemas, y 2) operación y mantenimiento de sistemas; primero se construye y, luego se usa y se mantiene. Tarde o temprano el ciclo culmina con el desarrollo de un nuevo sistema.

### > Ciclo de vida frente a metodología

Los términos *ciclo de vida del sistema* y *metodología de desarrollo del sistema* con frecuencia son intercambiados incorrectamente. La mayoría de los procesos de desarrollo de sistemas se derivan de un **ciclo de vida del sistema** natural. El ciclo de vida del sistema sólo sucede. En la figura 3.2 se ilustran dos etapas de ciclo de vida. Nótese que hay dos eventos fundamentales que disparan un cambio de una etapa a otra:

<sup>2</sup> White Paper, “Rapidly Improving Process Maturity: Moving Up the Capability Maturity Model through Outsourcing” (Boston, Keane, 1997, 1998, p. 11).

**FIGURA 3.2**

El ciclo de vida del sistema

- Cuando un sistema evoluciona del desarrollo a la operación y al mantenimiento, debe darse una *conversión*.
- En algún punto en el tiempo, se vuelve *obsoleto* (o se vuelve inminente) y el ciclo cambia de la operación y el mantenimiento al redesarrollo.

En realidad, un sistema puede estar en más de una etapa al mismo tiempo. Por ejemplo, una versión puede estar en *operación y soporte* mientras que la siguiente versión está en *desarrollo*.

¿Cómo contrasta esto con una metodología de desarrollo de sistemas? Una metodología de desarrollo de sistemas "ejecuta" la etapa de desarrollo de sistemas del ciclo de vida del sistema. Cada sistema de información tiene su propio ciclo de vida. La metodología es el proceso estándar para construir y mantener ese sistema y todos los demás sistemas de información a través de sus ciclos de vida. En consistencia con las metas de CMM, las metodologías aseguran que:

- Un método consistente y reproducible se aplique a todos los proyectos.
- Hay un riesgo reducido asociado con las omisiones y los errores.
- Se produzca documentación completa y consistente de un proyecto al otro.
- Analistas de sistemas, diseñadores y constructores puedan ser reasignados rápidamente entre proyectos debido a que todos usan el mismo proceso.
- Como los equipos y el personal de desarrollo cambian constantemente, los resultados del trabajo anterior pueden ser encontrados con facilidad y entendidos por las personas que les siguen.

Las metodologías se pueden adquirir o crear internamente. ¿Por qué comprar una metodología? Muchas organizaciones de sistemas de información no pueden permitirse el dedicar personal al desarrollo y mejora continua de la metodología interna. Los proveedores de metodología han mostrado interés en mantener sus metodologías actualizadas con las más recientes tendencias de negocios y de tecnología. Las metodologías internas en general están basadas en metodologías genéricas y técnicas que están bien documentadas en libros y en sitios Web. Algunos ejemplos de metodologías de desarrollo de sistemas están listados en el margen de la siguiente página. Usted debe ser capaz de investigar la mayoría de ellas en la Web. Muchos de sus métodos fundamentales se enseñarán en este texto.

A lo largo de este libro utilizaremos la metodología llamada **FAST**, que significa *Framework for the Application of Systems Thinking*. **FAST** no es una metodología comercial del mundo real. La desarrollamos como una mezcla de las mejores prácticas que hemos encontrado en muchas metodologías comerciales y de referencia. A diferencia de muchas

**FAST** Método teórico usado en este libro para demostrar un proceso de desarrollo de sistemas representativo. Las letras de las siglas corresponden a *Framework for the Application of Systems Thinking* ("Marco de referencia para la Aplicación del Pensamiento de Sistemas").

## METODOLOGÍAS REPRESENTATIVAS DEL DESARROLLO DE SISTEMAS

Arquitectura de desarrollo rápido de aplicaciones (arquitectura RAD)

Metodología de desarrollo de Sistemas dinámicos (DSDM)

Desarrollo Conjunto de Aplicación (JAD)

Ingeniería de información (IE)

Desarrollo rápido de aplicación (RAD)

Proceso unificado (RUP)

Análisis y diseño estructurado (viejo, pero todavía se encuentra ocasionalmente)

Programación eXtrema (XP)

Nota: Existen muchas metodologías comerciales y herramientas de software (a veces llamadas *methodware*) con base en las metodologías generales anteriores.

metodologías comerciales, *FAST* no es estricta. Esto es, *FAST* es un marco de referencia que es ágil y lo suficientemente flexible para proporcionar distintos tipos de proyectos y estrategias. Pero lo más importante, *FAST* tiene mucho en común con las metodologías basadas en los libros y las comerciales que usted encontrará en la práctica.

## > Principios fundamentales para el desarrollo de sistemas

Antes de que examinemos la metodología *FAST*, presentemos algunos principios generales que deben estar en todas las metodologías del desarrollo de sistemas.

**Principio 1: Hacer participar a los usuarios del sistema** Los analistas, programadores y otros especialistas de la tecnología de la información frecuentemente se refieren a “mi sistema”. Esta actitud, en parte, ha creado un conflicto de “nosotros contra ellos” entre el personal técnico y los usuarios y la administración. Aunque los analistas y programadores trabajan fuerte para crear soluciones impresionantes de tecnología, esas soluciones a menudo les ocasionan problemas porque no abordan los problemas reales de la organización. Algunas veces hasta presentan nuevos problemas en la organización. Por esta razón, la participación del usuario del sistema es una necesidad absoluta para el desarrollo de sistemas exitosos. Piense en el desarrollo de sistemas como una sociedad entre los usuarios, analistas, diseñadores y constructores del sistema. Los analistas, diseñadores y constructores son responsables por el desarrollo del sistema, pero deben hacer participar a sus propietarios y usuarios, insistir en su participación y buscar un acuerdo por parte de todos los interesados en relación con las decisiones que puedan afectarlos.

La mala comunicación y los malos entendidos continúan como un problema significativo en muchos proyectos de desarrollo de sistemas. Sin embargo, la participación entre propietarios y usuarios y la cultura minimiza dichos problemas y ayuda a ganar aceptación de nuevas ideas y cambio tecnológico. Debido a que la gente tiende a resistirse al cambio, la tecnología de la información a menudo es vista como una amenaza. La mejor forma de contrarrestar esa amenaza es mediante una comunicación constante y completa con los propietarios y los usuarios.

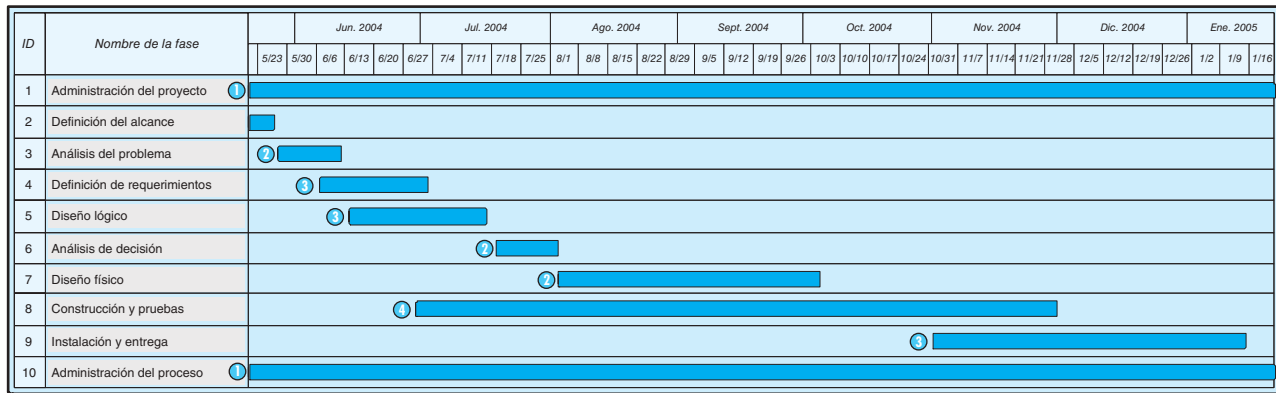
**Principio 2: Utilizar un método de solución de problemas** Una metodología de desarrollo de sistemas es, primero y sobretodo, un enfoque de solución de problemas de la construcción de sistemas. El término *problema* es ampliamente utilizado a lo largo de este libro para incluir 1) problemas reales, 2) oportunidades para mejorar, y 3) directrices de la administración. El método clásico de solución de problemas es el siguiente:

1. Estudiar y entender el problema, su contexto y su impacto.
2. Definir los requerimientos que deben satisfacerse para alcanzar *una* solución.
3. Identificar alternativas de soluciones que satisfagan los requerimientos y elegir la mejor solución.
4. Diseñar y/o implantar la solución elegida.
5. Observar y evaluar el impacto de la solución y depurarla.

Los analistas de sistemas deben abordar todos los proyectos por medio de alguna variación de este método de solución de problemas.

Los solucionadores de problemas inexpertos o poco exitosos tienden a eliminar o abreviar uno o más de los pasos anteriores. Por ejemplo, fracasan en entender por completo el problema o prematuramente se comprometen con la primera solución que se les ocurre. El resultado puede ir desde 1) solucionar el problema equivocado, 2) solucionar incorrectamente el problema, 3) elegir la solución equivocada, o 4) elegir una solución menos óptima. El proceso de solución de problemas de una metodología, cuando se aplica correctamente, puede reducir o eliminar estos riesgos.

**Principio 3: Establecer fases y actividades** Todas las metodologías tienen fases y actividades. El número y el alcance de ambas varían de autor en autor, de experto en experto, de metodología en metodología y de empresa a empresa. La página de inicio de este capítulo ilustra las ocho fases de nuestra metodología *FAST* en el contexto de su marco de referencia de sistemas de información. Las fases se listan en el extremo derecho de la ilustración. En cada fase, el enfoque está en esos componentes y en los interesados que están alineados a la izquierda de esa fase.



**FIGURA 3.3** Traslape de fases y actividades de desarrollo de sistemas

Las fases son: definición del alcance, análisis del problema, análisis de requerimientos, diseño lógico, análisis de decisión, diseño físico e integración, construcción y pruebas, instalación y entrega. Cada una de estas fases será analizada posteriormente en este capítulo. Estas fases no son absolutamente secuenciales, tienden a traslaparse entre ellas, como se ilustra en la figura 3.3. Además, las fases pueden ser personalizadas para las necesidades especiales de un proyecto dado (por ejemplo, vencimientos, complejidad, estrategia, recursos). En este capítulo, describiremos cada personalización como rutas alternativas a través de la metodología y el proceso de solución de problemas.

**Principio 4: Documentar a través del desarrollo** ¿Cuándo documenta los programas que escribe? Sea honesto. Debemos confesar que, al igual que la mayoría de los estudiantes, hicimos nuestra documentación después de escribir los programas. Sabíamos que no era lo indicado, pero de todas formas lo hicimos al final. Eso simplemente no funciona en el mundo de los negocios. En las organizaciones de medianas a grandes, los propietarios, usuarios, analistas, diseñadores y constructores de sistemas van y vienen. Algunos serán promovidos; otros se irán por incapacidades médicas; otros renunciarán a la organización y otros más serán reasignados. Para promover una buena comunicación entre interesados constantemente en cambio, la documentación debe ser un producto alterno del esfuerzo completo del desarrollo de sistemas.

La documentación mejora las comunicaciones y la aceptación. La documentación revela fortalezas y debilidades del sistema para los múltiples interesados. Estimula la participación de los usuarios y reasegura la administración acerca del progreso. Al mismo tiempo, algunas metodologías han sido criticadas por esperar demasiada documentación que agrega poco valor al proceso o sistema resultante. Nuestra metodología *FAST* defiende un equilibrio entre el valor de la documentación y el esfuerzo por producirla. Los expertos llaman a esto *modelado acelerado*.

**Principio 5: Establecer estándares** En un mundo perfecto, todos los sistemas de información estarían integrados de tal forma que se comportaran como un solo sistema. Desafortunadamente, esto nunca sucede debido a que los sistemas de información son desarrollados y reemplazados durante un largo periodo. Incluso las organizaciones que adquieren e instalan un producto de planeación de recursos de la empresa (enterprise resource planning, ERP) generalmente descubren que hay aplicaciones y necesidades que quedan fuera del sistema ERP. La integración de sistemas se ha vuelto crítica para el éxito de los sistemas de información de cualquier organización.

Para lograr o mejorar la integración de sistemas, las organizaciones se apegan a estándares. En muchas organizaciones, éstos toman forma de arquitectura tecnológica de información empresarial. Una arquitectura de TI establece estándares que sirven para dirigir soluciones de tecnología y sistemas de información hacia una visión o configuración de tecnología común.

Una arquitectura de tecnología de información generalmente se estandariza acerca de lo siguiente (Nota: no es importante que usted sepa lo que son todas estas tecnologías de muestra):

- **Tecnología de bases de datos.** ¿Qué motores de bases de datos se utilizarán (por ejemplo, Oracle, IBM, DB2, SQL Server de Microsoft)? ¿En qué plataformas se operarán (por ejemplo UNIX, Linux, Windows XP, MVS)? ¿Qué tecnologías se utilizarán para cargar datos en las bases de datos de procesamiento de transacciones en línea (OLTP), almacenes de datos operacionales y almacenes de datos (por ejemplo, Extract Transform y Load [ETL])?
- **Tecnología de software.** ¿Qué ambientes/lenguajes de desarrollo de aplicaciones serán utilizados para escribir software (por ejemplo Websphere de IBM con Java, Visual Studio .NET de Microsoft con Visual Basic .NET, Visual C++ o Visual C++; Powerbuilder de Sybase, Oracle Forms de Oracle)?
- **Tecnología de interfaz.** ¿Cómo serán desarrolladas las interfaces de usuario, con componentes de MS Windows o lenguajes Web y componentes (por ejemplo, un editor xHTML como Dreamweaver de Macromedia, un motor de portal como el Websphere de IBM)? ¿Cómo se intercambiarán los datos entre los distintos sistemas de información (por ejemplo, un corredor de datos como MQ Messaging de IBM, un intercambio de datos basado en XML o una interfaz programada personalizada)?

Note cómo estas preguntas de arquitectura se aproximan a los controladores de tecnología en su modelo de sistema de información.

En ausencia de una arquitectura TI, cada sistema de información y aplicación puede ser construido con tecnologías radicalmente diferentes. Esto no sólo hace difícil integrar las aplicaciones, sino que crea problemas de administración de recursos; las organizaciones de TI no pueden tan fácilmente mover a los desarrolladores entre proyectos conforme cambian las prioridades u ocurren emergencias, debido a que diferentes equipos están formados con competencias técnicas basadas en las diversas tecnologías empleadas para desarrollar sistemas de información. Crear la arquitectura TI de una empresa y conducir proyectos y equipos hacia esa arquitectura tiene más sentido.

Conforme surgen las nuevas tecnologías, una arquitectura TI debe cambiar. Pero ese cambio puede ser manejado. El director de tecnología (CTO, por sus siglas en inglés) en una organización, a menudo se encarga de la exploración de tecnología y la administración de la arquitectura de TI. Dada esa arquitectura, todos los proyectos de sistemas de información están restringidos a implantar nuevos sistemas que se adaptan a la arquitectura (a menos que el CTO lo apruebe de otra manera).

---

**administración de procesos** Actividad continua que documenta, enseña, supervisa y mejora el uso de los métodos (el "proceso") y estándares que elige una organización para el desarrollo de sistemas. La administración de procesos se relaciona con fases, actividades, productos y normas de calidad que se deben aplicar de manera constante a todos los proyectos.

---

**administración de proyectos** Proceso de definir el alcance, planear, dotar de personal, organizar, dirigir y controlar un proyecto para desarrollar un sistema de información con costo mínimo, dentro del tiempo especificado y con calidad aceptable.

**Principio 6: Administrar el proceso y los proyectos** La mayoría de las organizaciones tienen un proceso o una metodología de desarrollo de sistemas, pero no siempre lo utilizan en forma consistente en los proyectos. Tanto el proceso como los proyectos que lo utilizan deben ser administrados. La **administración de procesos** asegura que el proceso o administración elegidos por una organización se utilice en forma consistente en todos los proyectos. La administración de procesos también define y mejora el proceso o metodología elegido con el paso del tiempo. La **administración de proyectos** asegura que un sistema de información sea desarrollado a un costo mínimo, dentro de un marco de tiempo específico y con una calidad aceptable (con un proceso o metodología estándar de desarrollo de sistema). La administración de proyectos eficaz es esencial para lograr un éxito de nivel 2 CMM. El uso de un proceso repetible nos lleva a un nivel 3 CMM. Los niveles 4 y 5 CMM requieren una administración de procesos eficaz. La administración de proyectos puede ocurrir sin un proceso estándar, pero en las organizaciones maduras todos los proyectos están basados en un proceso estandarizado y administrado.

La administración de procesos y la administración de proyectos están influidas por la necesidad de la administración de calidad. Los estándares de calidad están construidos en un proceso para asegurar que las actividades y los productos de cada fase contribuirán al desarrollo de un sistema de información de alta calidad. Reducen la probabilidad de problemas y requerimientos omitidos, así como diseños con fallas y errores de programas. Los estándares también hacen que la organización de TI sea más ágil. Mientras ocurren los cambios de personal, éste puede ser reubicado entre proyectos con la seguridad de que cada proyecto sigue un proceso entendido y aceptado.

**Principio 7: Justificar sistemas de información como inversiones de capital** Los sistemas de información son inversiones de capital, tal como una flota de camiones o un nuevo edificio. Los propietarios del sistema se comprometen con esta inversión. Nótese que el compromiso inicial ocurre al inicio de un proyecto, cuando los propietarios del sistema acuerdan patrocinar y financiar el proyecto. Posteriormente (durante la fase llamada *análisis de decisión*), los propietarios del sistema se vuelven a comprometer con las decisiones técnicas más costosas. Al considerar una inversión de capital, se deben abordar dos temas:

1. Para cualquier problema, es probable que existan varias soluciones. El analista de sistemas y otros interesados no deben aceptar a ciegas la primera solución sugerida. El analista que fracasa en observar alternativas puede estar dando un mal servicio al negocio.
2. Luego de identificar soluciones alternativas, el analista de sistemas debe evaluar cada posible solución en busca de factibilidad, especialmente para una **efectividad de costos**. Ésta se mide con una técnica llamada *análisis de costo-beneficio*.

Al igual que la administración de proyectos y de procesos, el análisis de costo-beneficio se realiza a lo largo del proceso de desarrollo de sistemas.

Una ventaja significativa del enfoque de fases al desarrollo de sistemas es que proporciona diversas oportunidades para reevaluar la efectividad de costos, el riesgo y la factibilidad. A menudo está la tentación de continuar un proyecto debido a la inversión ya realizada. A la larga, los proyectos cancelados son generalmente menos costosos que los desastres implantados. Es extremadamente importante que los analistas jóvenes recuerden esto.

La mayoría de los propietarios de sistemas quieren más de sus sistemas de lo que pueden permitirse o de lo que están dispuestos a pagar. Es más, el alcance de la mayoría de los proyectos de sistemas de información aumenta conforme el analista aprende más acerca de los problemas y requerimientos del negocio conforme el proyecto progresa. Desafortunadamente, la mayoría de los analistas fracasan en ajustar los costos estimados y los programas conforme aumenta el alcance. Como resultado, el analista frecuentemente y sin necesidad acepta la responsabilidad de los excedentes en costos y en tiempo.

Como los sistemas de información son reconocidos como inversiones de capital, los proyectos de desarrollo de sistemas a menudo son conducidos por la planeación empresarial. Muchas unidades de negocio de tecnología de información contemporánea crean y mantienen un **plan estratégico de sistemas de información**. Dicho plan identifica y prioriza los proyectos de desarrollo de sistemas de información. De manera ideal, un plan de sistemas de información estratégico es conducido por un **plan estratégico de la empresa** que define una dirección para todo el negocio.

**Principio 8: No tema cancelar o revisar el alcance** Hay un dicho: “No tire dinero bueno tras el dinero malo”. En otras palabras, no tema cancelar un proyecto o revisar un alcance, sin importar cuánto dinero se haya gastado hasta el momento; recorte sus pérdidas. Para este fin, recomendamos un método de **compromiso ajustado** para el desarrollo de sistemas.<sup>3</sup> Con el método de compromiso ajustado se construyen múltiples puntos de revisión de factibilidad en cualquier metodología de desarrollo de sistemas. En cada punto de revisión se reevalúa la factibilidad. Todos los costos previamente realizados se consideran hundidos (lo que significa irrecuperables). Por lo tanto, son irrelevantes para la toma de decisiones. Así, el proyecto debe ser reevaluado en cada punto de revisión para determinar si continúa siendo factible para seguir con la inversión de tiempo, esfuerzo y recursos en ese proyecto. En cada punto de revisión, el analista debe considerar las siguientes opciones:

- Cancelar el proyecto si ya no es factible.
- Reevaluar y ajustar los costos y el programa si el alcance del proyecto se debe incrementar.
- Reducir el alcance si el presupuesto del proyecto y el programa están congelados y no son suficientes para cubrir todos los objetivos del proyecto.

El concepto de costos hundidos es más o menos familiar para la mayoría de los analistas financieros, pero con frecuencia es olvidado o no utilizado por la mayor parte de los analistas de sistemas, usuarios de sistemas y aún más por los propietarios de sistemas.

---

#### efectividad de costos

Resultado obtenido al lograr el equilibrio entre los costos de desarrollo, mantenimiento y operación de un sistema de información y los beneficios derivados de éste durante su ciclo de vida. La actividad de costos se mide con el análisis de costo-beneficio.

---

#### plan estratégico de sistemas de información

Plan formal (tres a cinco años) para construir y mejorar la infraestructura de tecnología de la información y las aplicaciones de sistemas de información que usan dicha infraestructura.

---

#### plan estratégico de la empresa

Plan formal (de tres a cinco años) de todo negocio que define su misión, visión, objetivos, estrategias, normas de comparación y mediciones de avance y logros. Es usual que el plan estratégico se complementa con planes estratégicos de unidades del negocio, en los que se define la manera en que cada unidad del negocio contribuirá al plan de la empresa. El plan de sistemas de información es uno de esos planes en el nivel de unidad.

---

#### compromiso ajustado

Estrategia en la que se reevalúan continuamente la factibilidad y los riesgos a lo largo de un proyecto. El presupuesto y fecha límite del proyecto se ajustan en concordancia.

<sup>3</sup> Thomas Gildersleeve, *Successful Data Processing Systems Analysis*, 2a. ed., Englewood Cliffs, NJ, Prentice Hall, 1985, pp. 5-7.

**administración de riesgos** Proceso de identificar, evaluar y controlar los problemas que podrían surgir con un proyecto antes de que se conviertan en una amenaza para la terminación exitosa del proyecto o la puesta en práctica del sistema de información. La administración de riesgos se lleva a cabo mediante el análisis o evaluación de riesgos.

### PRINCIPIOS DEL DESARROLLO DE SISTEMAS

- Hacer que los usuarios participen
- Utilizar un método de solución de problemas
- Establecer fases y actividades
- Documentar a través del desarrollo
- Establecer estándares
- Administrar el proceso y los proyectos
- Justificar los sistemas de información como inversiones de capital.
- No temer cancelar o revisar el alcance.
- Dividir y conquistar.
- Diseñar sistemas para el crecimiento y el cambio.

Además de administrar la factibilidad a lo largo del proyecto, debemos administrar el riesgo. La **administración de riesgos** busca equilibrar riesgo y beneficio. Las diferentes organizaciones son más o menos reacias al riesgo, lo que significa que algunas están dispuestas a tomar mayores riesgos que otras con el fin de lograr mayores beneficios.

**Principio 9: Divida y vencerá** Ya sea que usted esté consciente o no, ha aprendido el método llamado “divide y vencerás” a lo largo de su educación. Desde la secundaria, a usted le ha sido enseñado a delinear un trabajo antes de escribirlo. Delinear es el enfoque de “dividir y vencer” de la escritura. Un enfoque similar se utiliza en el desarrollo de sistemas. Dividimos un sistema en subsistemas y componentes con el fin de conquistar con mayor facilidad el problema y construir un sistema más grande. En el análisis de sistemas, a menudo llamamos a esto *dividir*. Al dividir repetidamente un problema más grande (sistema) en piezas más manejables (subsistemas), el analista puede simplificar el proceso de solución de problemas. Este método de dividir y vencer también complementa la comunicación y la administración de proyectos al permitir que diferentes piezas del sistema sean comunicadas a interesados distintos y más apropiados.

Los componentes del marco de referencia del sistema de información le proporcionan una base para dividir y conquistar el desarrollo de un sistema de información. Utilizaremos este marco de referencia a lo largo del texto.

**Principio 10: Diseñar sistemas para crecimiento y cambio** Los negocios cambian con el paso del tiempo. Sus necesidades cambian. Sus prioridades cambian. En consecuencia, los sistemas de información que respaldan a un negocio deben cambiar con el paso del tiempo. Por esta razón, buenas metodologías deben abrazar la realidad del cambio. Los sistemas deben ser diseñados para incorporar tanto los requerimientos de crecimiento como de cambio. En otras palabras, los sistemas de información bien diseñados pueden escalar y adaptarse al negocio. Pero sin importar qué tan bien diseñemos los sistemas para crecimiento y cambio, siempre llegará el momento cuando simplemente ya no puedan soportar el negocio.

Los expertos de sistemas describen la caída natural e inevitable de todos los sistemas por el paso del tiempo como *entropía*. Como ya se describió en esta sección, luego de que un sistema se implanta, entra en la etapa de *operaciones y mantenimiento* de su ciclo de vida. Durante este periodo, el analista encuentra la necesidad de cambios que van desde corregir errores simples hasta rediseñar el sistema para incorporar la cambiante tecnología y hacer modificaciones para soportar requerimientos cambiantes de los usuarios. Dichos cambios dirigen al analista y a los programadores a trabajar de nuevo las fases antes completadas del ciclo de vida. Eventualmente, el costo de mantener el sistema actual supera los costos de desarrollar un sistema de reemplazo; el sistema actual ha llegado a un nivel de entropía inadecuado y se vuelve obsoleto.

Pero la entropía del sistema puede ser administrada. Las herramientas y técnicas actuales hacen posible diseñar sistemas que pueden crecer y cambiar conforme los requerimientos crecen y cambian. Este libro le enseñará muchas de esas herramientas y técnicas. Por ahora, es más importante reconocer simplemente que la flexibilidad y la adaptabilidad no suceden por accidente; deben ser construidas en un sistema.

Hemos presentado 10 principios que deben estar en toda metodología. Estos principios se resumen al margen y pueden ser utilizados para evaluar cualquier metodología, incluida nuestra metodología *FAST*.

## Proceso de desarrollo de sistemas

En esta sección analizaremos un proceso lógico para el desarrollo de sistemas. (Recordatorio: *FAST* es una metodología hipotética creada para fines de aprendizaje.) Comenzaremos por estudiar tipos de proyectos de sistemas y la forma en que se inician. Luego presentaremos las ocho fases de *FAST*. Finalmente, analizaremos variaciones alternativas o “rutas” a través de las fases, para distintos tipos de estrategias de proyectos y desarrollo.



## > ¿De dónde surgen los proyectos de desarrollo de sistemas?

Los propietarios y los usuarios de sistemas inician la mayoría de los proyectos. El ímpetu de casi todos los proyectos es alguna combinación de **problemas, oportunidades y directrices**. Para simplificar este análisis, con frecuencia utilizaremos el término *problema* para referirnos colectivamente a problemas, oportunidades y directrices. En consecuencia, la *solución de problemas* se refiere a solucionar problemas, explotar oportunidades y satisfacer directrices.

Hay demasiados problemas potenciales de sistemas para listarlos todos en este texto. Sin embargo, James Wetherbe desarrolló un marco de referencia útil para clasificar problemas.<sup>4</sup> Él le llama *PIECES* debido a las letras en inglés de cada una de las seis categorías, que cuando se unen, deletrean la palabra “pieces” (significa “piezas”). Las categorías son:

- P** (*performance*) la necesidad de corregir o mejorar el *desempeño*
- I** (*information*) la necesidad de corregir o mejorar la *información* (y datos)
- E** (*economics*) la necesidad de corregir o mejorar la *economía*, controlar costos o aumentar las utilidades.
- C** (*control*) la necesidad de corregir o mejorar el *control* o la seguridad.
- E** (*eficiency*) la necesidad de corregir o mejorar la *eficiencia* de las personas y los procesos.
- S** (*service*) la necesidad de corregir o mejorar el *servicio* a clientes, proveedores, socios, empleados y demás.

En la figura 3.4 se amplía la información de cada una de las categorías de PIECES.

Las categorías del marco de referencia de PIECES no son ni exhaustivas ni mutuamente exclusivas, se traslapan. Cualquier proyecto se caracteriza generalmente por una o más categorías y cualquier problema u oportunidad puede tener implicaciones en relación con más de una categoría. Pero PIECES es un marco de referencia práctico (utilizado en *FAST*), no sólo un ejercicio académico. Volveremos a mencionar PIECES varias veces en este texto.

Los proyectos pueden ser planeados o no planeados. Un *proyecto planeado* es el resultado de una de las siguientes opciones:

- En un *plan de estrategia de sistemas de información* se ha analizado el negocio como un todo para identificar aquellos proyectos de desarrollo de sistemas que tendrán el mayor retorno de valor estratégico (largo plazo) para el negocio.
- En un *rediseño de proceso de negocios* se ha analizado profundamente una serie de procesos del negocio para eliminar la redundancia y la burocracia y para mejorar la eficiencia y el valor agregado. Ahora es el momento de rediseñar el sistema de información de soporte para los procesos rediseñados del negocio.

Lo opuesto de los proyectos planeados son los *proyectos no planeados*, aquellos que se disparan por un problema, una oportunidad o una directriz en específico que sucede en el curso de hacer negocios. La mayoría de las organizaciones no tienen pocos proyectos planeados. Cualquiera puede remitir un proyecto propuesto con base en algo que sucede en el negocio. El número de propuestas de proyecto no planeadas puede fácilmente abrumar hasta a la organización de sistemas de información más grande; por lo tanto, con frecuencia son vigilados y priorizados por un **comité de dirección** de propietarios de sistemas y de gerentes de TI para determinar qué solicitudes son aprobadas. Las requisiciones que no son aprobadas son puestas en una **reserva** hasta que haya recursos disponibles (lo que a veces nunca sucede).

Los proyectos planeados y los no planeados pasan por el mismo proceso esencial de desarrollo de sistemas. Analicemos ahora las fases de proyecto en mayor detalle.

## > Las fases del proyecto FAST

*FAST*, al igual que la mayoría de las metodologías, consiste de fases. El número de ellas variará de una metodología a otra. En el capítulo 1 se le presentaron las cuatro fases clásicas del ciclo de vida de desarrollo de sistemas. La metodología *FAST* utiliza ocho fases para

---

**problema** Situación indeseable que impide a la organización lograr plenamente su misión, visión, metas y objetivos.

---

**oportunidad** Ocasión de mejorar la organización, incluso en ausencia de un problema identificado.

---

**directriz** Un nuevo requerimiento que imponen los administradores, organismos gubernamentales o alguna otra influencia externa.

---

### comité de dirección

Cuerpo administrativo de propietarios del sistema y ejecutivos de tecnología de la información que asigna prioridades y aprueba los proyectos de desarrollo de sistemas candidatos.

---

**reserva** (Proyectos en cartera) Repositorio de propuestas de proyectos que no se pueden financiar ni dotar de personal porque su prioridad es más baja que la de los proyectos aprobados para desarrollo de sistemas. Nótese que la prioridad cambia al paso del tiempo, por lo que un proyecto en reserva podría ser aprobado en una fecha futura.

<sup>4</sup> James Wetherbe y Nicholas P. Vitallari, *Systems Analysis and Design: Traditional, Best Practices*, 4a. ed., St. Paul, MN, West Publishing, 1994, pp. 196-199. James Wetherbe es un respetado educador, investigador y consultor en sistemas de información.

## Marco de referencia de solución de problemas y lista de revisión de PIECES

La siguiente lista de revisión de identificación de problemas, oportunidades y directrices utiliza el marco de referencia PIECES. Nótese que las categorías de PIECES no son mutuamente exclusivas; algunos problemas posibles aparecen en múltiples listas. También, la lista de posibles problemas no es exhaustiva. El marco de referencia PIECES es igualmente apropiado para analizar los sistemas y aplicaciones manuales y computarizados.

### DESEMPEÑO

- A. Caudal. Cantidad de trabajo que se desempeña durante un periodo
- B. Tiempos de respuesta. El retraso promedio entre una transacción o solicitud y una respuesta a esa transacción o solicitud

### INFORMACIÓN (y datos)

- A. Salidas
  - 1. Falta de cualquier información
  - 2. Falta de la información necesaria
  - 3. Falta de información relevante
  - 4. Demasiada información, "sobrecarga" de información
  - 5. Información que no está en un formato útil
  - 6. Información que no es precisa
  - 7. Información que es difícil de producir
  - 8. Información que no es oportuna para su uso posterior
- B. Entradas
  - 1. Los datos no se capturan
  - 2. Los datos no se capturan a tiempo para ser útiles
  - 3. Los datos no se capturan con precisión, contienen errores
  - 4. Los datos son difíciles de capturar
  - 5. Los datos se capturan en forma redundante, los mismos datos son capturados más de una vez
  - 6. Se capturan demasiados datos
  - 7. Se capturan datos ilegales
- C. Datos almacenados
  - 1. Los datos son almacenados en forma redundante en múltiples archivos o bases de datos
  - 2. Los mismos temas de datos tienen distintos valores en diferentes archivos (mala integración de datos)
  - 3. Los datos almacenados no son precisos
  - 4. Los datos no están seguros contra accidentes o vandalismo
  - 5. Los datos no están bien organizados
  - 6. Los datos no son flexibles, no es fácil satisfacer las necesidades de información nuevas con los datos almacenados
  - 7. Los datos no son accesibles

### ECONOMÍA

- A. Costos
  - 1. Los costos son desconocidos
  - 2. Los costos no se pueden rastrear hasta su fuente
  - 3. Los costos son demasiado altos
- B. Utilidades
  - 1. Nuevos mercados pueden ser explorados

- 2. El marketing actual puede ser mejorado
- 3. Los pedidos pueden ser aumentados

### CONTROL (y seguridad)

- A. Muy poca seguridad y control
  - 1. La entrada de datos no está editada adecuadamente
  - 2. Crímenes (por ejemplo, fraude, fraude informático) son (o pueden ser) cometidos contra los datos
  - 3. Se falta a la ética acerca de datos o información, se refiere a datos o información que llega a personas no autorizadas
  - 4. Los datos almacenados en forma redundante son inconsistentes en distintos archivos o bases de datos
  - 5. Las regulaciones o lineamientos de privacidad de datos se violan (o pueden ser violados)
  - 6. Suceden errores de proceso (ya sea por las personas, las máquinas o el software)
  - 7. Ocurren errores de toma de decisiones
- B. Demasiado control o seguridad
  - 1. La burocracia frena el sistema
  - 2. Los controles causan inconveniencia a los clientes o a los empleados
  - 3. Los controles excesivos pueden ocasionar retrasos de proceso

### EFICIENCIA

- A. Las personas, máquinas o computadoras pierden el tiempo
  - 1. Los datos se ingresan o copian en forma redundante
  - 2. Los datos se procesan en forma redundante
  - 3. La información se genera en forma redundante
- B. Las personas, máquinas o computadoras desperdician materiales o suministros
- C. El esfuerzo requerido para las tareas es excesivo
- D. El material requerido para las tareas es excesivo

### SERVICIO

- A. El sistema produce resultados imprecisos
- B. El sistema produce resultados inconsistentes
- C. El sistema produce resultados poco confiables
- D. El sistema no es fácil de aprender
- E. El sistema no es fácil de utilizar
- F. El sistema tiene un uso torpe
- G. El sistema es inflexible a situaciones nuevas o excepcionales
- H. El sistema es inflexible al cambio
- I. El sistema es incompatible con otros sistemas

**FIGURA 3.4** Marco de referencia PIECES para identificación de problemas

definir mejor los hitos periódicos y los productos. En la cuadrícula siguiente se comparan las fases *FAST* con las fases clásicas. Como usted puede ver, ambos conjuntos cubren el mismo terreno, pero *FAST* es más detallado.

<b>Fases <i>FAST</i></b>	<b>Fases clásicas</b>			
	Inicio de proyecto	Análisis del sistema	Diseño del sistema	Implantación del sistema
Definición de alcance	X			
Análisis del problema	X	X		
Análisis de requerimientos		X		
Diseño lógico		X		
Análisis de decisión	(fase de transición de análisis del sistema)			
Diseño físico e integración			X	
Construcción y pruebas			X	X
Instalación y entrega				X

En la figura 3.5 se ilustran las fases de la metodología *FAST*. Cada una tiene productos que se pasan a la siguiente fase. Y la documentación se acumula conforme usted completa cada fase. Nótese que hemos incluido una representación icónica de los componentes para simbolizar esta acumulación de conocimiento y de artefactos de trabajo en proceso durante el desarrollo de sistemas. Nótese también que un proyecto comienza con alguna combinación de PROBLEMAS, OPORTUNIDADES y DIRECTRICES desde la comunidad del usuario (la flecha punteada) y termina con una SOLUCIÓN DEL NEGOCIO FUNCIONAL (la flecha discontinua) para la comunidad del usuario.

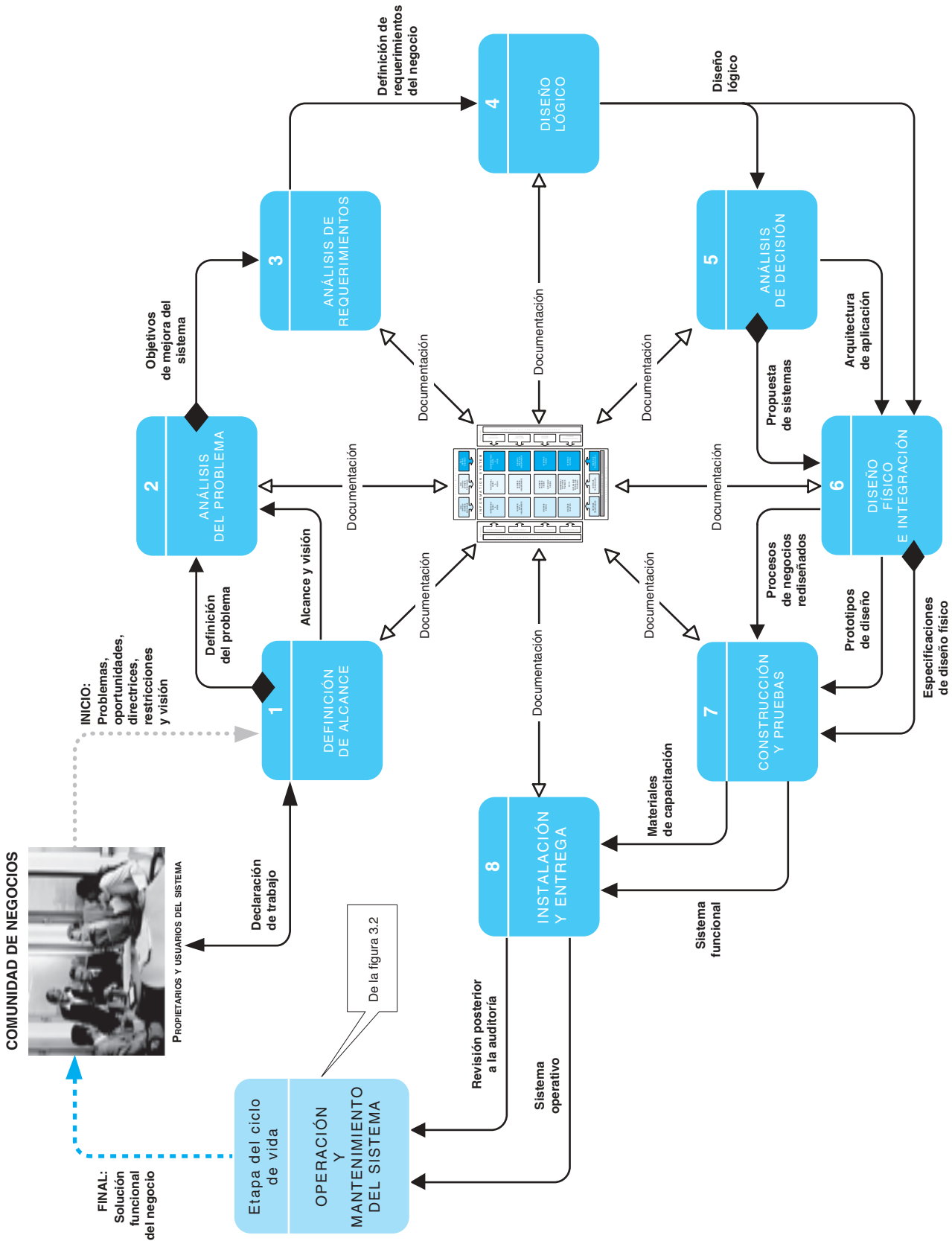
En la figura 3.6 se muestra la metodología *FAST* desde la perspectiva de los componentes del sistema de información que usted aprendió en los capítulos 1 y 2. Las fases están en el lado derecho. Los productos se construyen alrededor de los componentes del conocimiento, los procesos y las comunicaciones. Los interesados están en el lado izquierdo. Nótese cómo hemos expandido y duplicado a algunos interesados para reflejar su participación en el otro extremo a las fases en las que participan primordialmente.

**NOTA:** En el resto de esta sección se describe brevemente cada una de las ocho fases básicas. A lo largo de este análisis, nos referiremos al diagrama de flujo del proceso de la figura 3.5, así como al punto de vista de los componentes del proceso en la figura 3.6. Todos los términos impresos en VERSALITAS (mayúsculas pequeñas) se refieren a las fases, los requerimientos previos (entradas) y los productos (salidas) que se muestran en las figuras 3.5 y 3.6.

**Definición de alcance** La primera fase de un proyecto típico es la DEFINICIÓN DE ALCANCE. El propósito de la fase de definición de alcance es de dos sentidos. Primero, responde la pregunta, “¿vale la pena atender este problema?”. Segundo y suponiendo que el problema vale la pena, establece el tamaño y las fronteras del proyecto, la visión del proyecto, cualquier restricción o limitación, los participantes requeridos del proyecto y finalmente, el presupuesto y el programa.

En la figura 3.6, vemos que los participantes en la fase de definición de alcance primeramente incluyen PROPIETARIOS DEL SISTEMA, ADMINISTRADORES DEL PROYECTO y ANALISTAS DEL SISTEMA. Los usuarios del sistema generalmente son excluidos debido a que es muy pronto para entrar al nivel de detalle que eventualmente aportarán al proyecto.

En la figura 3.5 vemos que la fase de definición de alcance es disparada por alguna combinación de PROBLEMAS, OPORTUNIDADES y DIRECTRICES (a lo que añadiremos RESTRIC-



**FIGURA 3.5** Imagen del proceso de desarrollo de sistemas

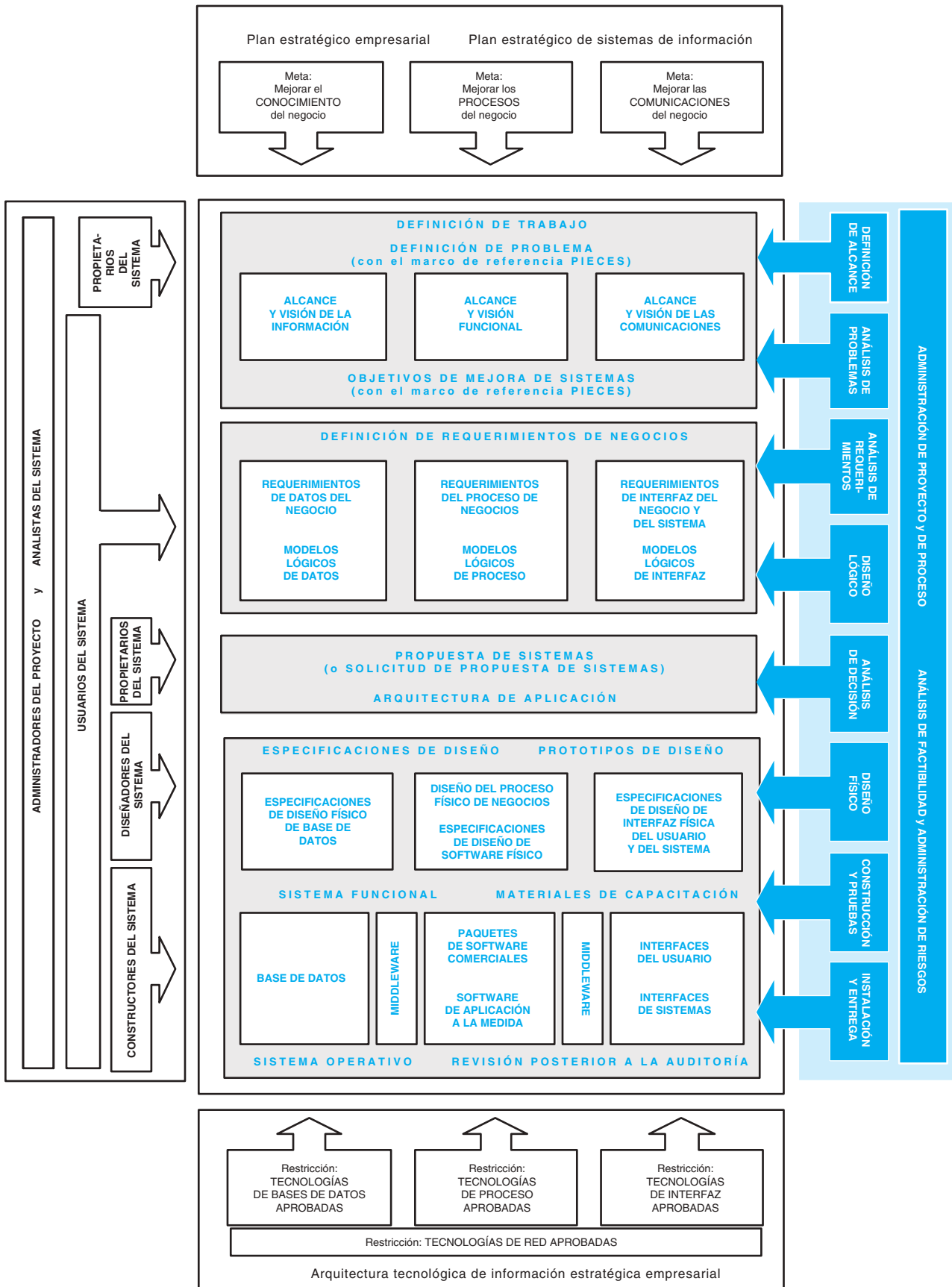


FIGURA 3.6 Vista de los componentes del desarrollo de sistemas

**definición del problema**

Definición y categorización de problemas, oportunidades y directrices; también suele incluir las restricciones y una decisión inicial de la solución. Sus sinónimos abarcan *estudio preliminar* y *evaluación de factibilidad*.

**restricción** Cualquier factor o limitación de una solución o proceso de solución de problemas.

**ampliación del alcance**

Fenómeno común en el que se incrementan los requerimientos y expectativas de un proyecto, frecuentemente sin considerar el efecto en su presupuesto y calendario (programa) de trabajo.

**declaración de trabajo**

Contrato de los administradores y la comunidad de usuarios para desarrollar o mejorar un sistema de información; define la visión, alcance, restricciones, requerimientos de usuarios de alto nivel, plan y presupuesto. Su sinónimo es *carta de definición de proyecto*, *plan de proyecto* y *convención del nivel de servicio*.

CIONES y VISIÓN). Existen varios productos o resultados de una definición del alcance. Un resultado importante es una DEFINICIÓN DEL PROBLEMA, una visión general de los problemas, oportunidades o directrices que disparan el proyecto. El marco de referencia PIECES proporciona un marco de referencia excelente para una **definición del problema**. La meta aquí no es resolver problemas, oportunidades y directrices sino sólo catalogarlos y clasificarlos. Debemos también identificar cualquier **restricción** que pueda impactar el proyecto propuesto. Algunos ejemplos de restricciones pueden incluir limitaciones de presupuesto, vencimientos, recursos humanos disponibles o no disponibles, políticas del negocio o regulaciones gubernamentales y estándares de tecnología. Finalmente, se debe pedir a los propietarios del sistema que proporcionen al menos una visión de alto nivel para las mejoras que buscan del sistema.

Dado un entendimiento básico de los problemas, oportunidades, directrices, restricciones y visión, necesitamos establecer un alcance inicial. Por tanto, una DEFINICIÓN DEL ALCANCE es otro resultado importante de esta fase. El alcance define qué tan grande creemos que es el proyecto. Sus componentes del sistema de información proporcionan un marco de referencia útil para definir el alcance. En la figura 3.6 se ilustra que el alcance y la visión pueden definirse en términos de INFORMACIÓN, FUNCIONES e INTERFACES. El alcance puede, y frecuentemente lo hace, cambiar durante un proyecto. Pero al documentar el alcance inicial, usted establece una línea de partida para controlar la **ampliación del alcance** tanto en el presupuesto como en el programa.

Dado el problema inicial y la definición del alcance del proyecto, el analista puede conseguir el personal para el equipo del proyecto, calcular el presupuesto para el desarrollo del sistema y preparar un programa para las fases restantes. Finalmente, esta fase concluye con una decisión de seguir “adelante o detenerse” por parte de los propietarios del sistema ya sea que estén de acuerdo con el alcance, presupuesto y programa propuesto para el proyecto o bien deban reducir el alcance (para reducir costos y tiempo) o cancelar el proyecto. Este punto de revisión de factibilidad es ilustrado en la figura 3.5 como un diamante.

El producto final y más importante es una DECLARACIÓN DE TRABAJO. Una **declaración de trabajo** es un contrato o acuerdo para desarrollar el sistema de información. Consolida la definición de problema, la definición del alcance y el programa y presupuesto para todas las partes que participarán en el proyecto.

**Análisis del problema** Siempre hay un sistema existente, sin importar si actualmente utiliza tecnología de la información. La fase del ANÁLISIS DEL PROBLEMA estudia el sistema existente y analiza los resultados que proporciona al equipo del proyecto con una comprensión más completa de los problemas que dispararon el proyecto. El analista con frecuencia descubre nuevos problemas y responde la pregunta más importante, “¿los beneficios de solucionar estos problemas exceden los costos de construir el sistema para resolver estos problemas?”

Una vez más, en la figura 3.6 se proporciona una visión general gráfica de la fase del análisis del problema en términos de los componentes de su sistema de información. Nótese que los participantes aún incluyen a los PROPIETARIOS DEL SISTEMA pero que en esta fase también comienzan a participar activamente los USUARIOS DEL SISTEMA; ellos son los expertos del tema de negocios en cualquier proyecto. (Nótese la expansión intencional de la perspectiva de los usuarios de sistemas que traslapan muchas fases, recuerde el principio 1: “Haga que participen los usuarios de sistemas”.) Desde luego, los ADMINISTRADORES DEL PROYECTO y los ANALISTAS DEL SISTEMA siempre participan en todas las fases de un proyecto.

Como se muestra en la figura 3.5, los requerimientos previos de la fase del análisis del problema son el ALCANCE y la DEFINICIÓN DEL PROBLEMA como se definen y aprueban en la fase de definición del alcance. El producto de la fase de análisis del problema es un conjunto de OBJETIVOS DE MEJORA DEL SISTEMA derivados de una comprensión profunda de los problemas del negocio. Estos objetivos no definen entradas, salidas o procesos. En su lugar, definen el criterio de negocios en el que cualquier sistema nuevo será evaluado. Por ejemplo, podríamos definir un objetivo de mejoramiento de sistemas como cualquiera de los siguientes:

Reducir el tiempo entre el proceso de pedidos y embarque en tres días.

Reducir las pérdidas por mal crédito en 45 por ciento.

Cumplir con nuevos requerimientos de calificación federal de ayuda financiera para el 1 de enero.

Piense en los objetivos de mejora de sistemas como el *criterio de puntuación* para evaluar cualquier sistema nuevo que usted eventualmente podría diseñar e implantar. Los objetivos de mejora del sistema pueden ser presentados a los propietarios y usuarios de sistemas como recomendaciones escritas o en una presentación oral.

Según la complejidad del problema y el programa del proyecto, el equipo puede o no elegir documentar formalmente el sistema existente. Dicha documentación con frecuencia ocurre cuando los procesos del negocio son considerados anticuados o demasiado burocráticos. La documentación del sistema existente a veces es llamada un MODELO DEL NEGOCIO “TAL COMO ES”. El “modelo tal como es” puede ir acompañado por análisis que demuestran ineficiencias, cuellos de botella u otros problemas relacionados con los procesos del negocio.

Cada sistema existente tiene su propia terminología, historia, cultura y detalles. Conocer esos aspectos del sistema es un importante producto derivado de esta fase. De toda la información recopilada, el equipo del proyecto obtiene una mejor comprensión de los problemas y oportunidades existentes del sistema. Luego de revisar los resultados, los propietarios del sistema estarán de acuerdo o en desacuerdo con los objetivos de mejoramiento del sistema recomendados. Y en consistencia con el principio de compromiso ajustado, incluimos otro punto de revisión de factibilidad de “adelante o detenerse” (diamante negro) al final de la fase. El proyecto puede ser:

- Cancelado si se considera que no vale la pena resolver los problemas.
- Aprobado para continuar a la siguiente fase.
- Reducido o expandido en alcance (con modificaciones de presupuesto y programas) y luego aprobado para continuar en la siguiente fase.

**Análisis de requerimientos** Dada la aprobación del propietario para continuar la fase de análisis del problema, ahora usted puede diseñar un nuevo sistema, ¿correcto? ¡No, aún no! ¿Qué capacidades debe proporcionar el nuevo sistema para sus usuarios? ¿Qué datos deben ser capturados y almacenados? ¿Qué nivel de desempeño se espera? ¡Cuidado! Esto requiere decisiones acerca de *lo que* el sistema debe hacer, no *cómo* debe hacerlo. La fase de ANÁLISIS DE REQUERIMIENTOS define y prioriza los requerimientos del *negocio*. Dicho de manera simple, el analista se aproxima a los usuarios para averiguar lo que necesitan o requieren del nuevo sistema, al evitar cuidadosamente cualquier discusión de tecnología o implantación técnica. Ésta es tal vez la fase más importante del desarrollo de sistemas. Errores y omisiones en el análisis de requerimientos resultarán en la insatisfacción del usuario con el sistema final y modificaciones costosas.

De nuevo en la figura 3.6, nótese que los participantes principalmente incluyen tanto USUARIOS DEL SISTEMA (que puede incluir propietarios que en realidad *utilizarán* el sistema) como ANALISTAS DEL SISTEMA. También participan ADMINISTRADORES DEL PROYECTO. LOS DISEÑADORES DEL SISTEMA se omiten en esta fase con el fin de evitar una atención prematura a las soluciones de tecnología. Los componentes pueden por sí mismos proveer el marco de referencia para definir muchos requerimientos del negocio, incluidos REQUERIMIENTOS DE DATOS DEL NEGOCIO, REQUERIMIENTOS DE PROCESOS DEL NEGOCIO Y REQUERIMIENTOS DE INTERFAZ DEL NEGOCIO Y SISTEMAS. Como se pretende que los requerimientos del negocio resuelvan los problemas, el marco de referencia de PIECES también puede proporcionar un útil ámbito, en esta ocasión para una definición de requerimientos.

En la figura 3.5 vemos que los OBJETIVOS DE MEJORA DE SISTEMAS de la fase de análisis de problema son el requerimiento previo para la fase de análisis de requerimientos. El producto es una DEFINICIÓN DE REQUERIMIENTOS DEL NEGOCIO. De nuevo, la definición de requerimientos no especifica ninguna posibilidad o solución técnica; puede ser un documento tan pequeño como unas cuantas páginas o ser extenso con una página o más documentos por requerimiento.

Para producir una definición de requerimientos del negocio, el analista de sistemas trabaja muy de cerca con los usuarios del sistema para identificar necesidades y prioridades. Esta información se recopila por medio de entrevistas, cuestionarios y juntas. El desafío para el equipo es validar esos requerimientos. Los objetivos de mejora del sistema proporcionan la “clave de clasificación” para requerimientos del negocio: *¿Cada requerimiento contribuye a reunir uno o más objetivos de mejora del sistema?* En los capítulos 6 y 7 se presentarán herramientas y técnicas de sistemas para identificar y documentar los requerimientos de usuario.

Típicamente, los requerimientos también deben ser priorizados. Las prioridades sirven a dos propósitos. Primero, si los vencimientos del proyecto se complican, se pueden utilizar las prioridades de requerimientos para redefinir el alcance del proyecto. Segundo, las prioridades con frecuencia pueden utilizarse para definir iteraciones de diseño y construcción para crear lanzamientos o versiones actualizadas del producto final.

La fase de análisis de requerimientos nunca se debe saltar o abreviar. Una de las quejas más comunes acerca de los nuevos sistemas y aplicaciones es que realmente no satisfacen las necesidades del usuario. Esto en general sucede cuando los diseñadores y los constructores de sistemas se preocupan por una solución técnica antes de entender por completo las necesidades del negocio. Los diseñadores y los constructores de sistemas dependen de analistas de sistemas competentes para trabajar con los usuarios, definir y documentar requerimientos del negocio completos y precisos antes de aplicar cualquier tecnología.

**Diseño lógico** Los requerimientos del negocio (antes mencionados) generalmente son expresados en palabras. Los analistas del sistema han encontrado que es útil traducir esas palabras en imágenes llamadas **modelos de sistemas** para validar los requerimientos con el fin de que estén completos y sean consistentes. (La figura 3.5 es un ejemplo de un modelo de sistemas común llamado *diagrama de flujo de datos*.) La elaboración de modelos de sistemas es un concepto eterno: “Una imagen vale más que mil palabras”.

LA FASE DEL DISEÑO LÓGICO traduce los requerimientos del negocio a modelos de sistemas. El término **diseño lógico** debe ser interpretado como “de tecnología independiente”, lo que significa que las imágenes ilustran el sistema en forma independiente de cualquier solución técnica posible, por tanto, modelan requerimientos del negocio que deben ser satisfechos mediante cualquier solución técnica que quisiéramos considerar.

Distintas metodologías requieren o recomiendan diferentes cantidades y grados de modelado de sistemas o diseño lógico. Metodologías estrictas como *análisis y diseño estructurado*, *ingeniería de la información* y el *Rational Unified Process (RUP)* normalmente requieren que muchos tipos o ejemplos de modelos de sistemas sean dibujados con distintos niveles de detalles. Por fortuna, las herramientas automatizadas de cómputo están disponibles para ayudar al analista de sistemas en estas tareas de dibujo. En forma alternativa, las metodologías ágiles como *desarrollo rápido de aplicaciones y programación extrema* recomiendan “apenas suficiente modelado”. Este tan llamado **modelado ágil** busca prevenir que el proyecto degenere en una condición llamada **parálisis de análisis**. En este texto nos inclinamos hacia métodos ágiles pero reconocemos que los problemas complejos pueden ser resueltos de una mejor manera por medio de métodos más estrictos.

En la figura 3.6 se puede ver que los participantes incluyen ANALISTAS DE SISTEMAS (quienes dibujan los modelos) y USUARIOS DE SISTEMAS (quienes validan los modelos). Los ADMINISTRADORES DE PROYECTO siempre están incluidos para asegurar que la elaboración de modelos cumple con los estándares y no afecta el progreso del proyecto en general. Podemos dibujar 1) MODELOS LÓGICOS DE DATOS que describan datos y requerimientos de información, 2) MODELOS LÓGICOS DE PROCESO que describan los requerimientos de procesos del negocio, y 3) MODELOS LÓGICOS DE INTERFAZ que describan el negocio y los requerimientos de interfaz del sistema.<sup>5</sup>

En la figura 3.5, podemos ver que el requerimiento previo para un diseño lógico es la DEFINICIÓN DE REQUERIMIENTOS DEL NEGOCIO de la fase anterior. En la práctica, las fases de análisis de requerimientos y de diseño lógico casi siempre tienen un traslape considerable. En otras palabras, mientras los requerimientos del negocio se identifican y se documentan, pueden ser modelados. Los productos del diseño lógico son los mismos MODELOS Y ESPECIFICACIONES DEL SISTEMA LÓGICO. Según la metodología utilizada, el nivel de detalle en las especificaciones variará. Por ejemplo, podemos definir una regla del negocio que especifique los valores legítimos para un atributo de datos tal como *evaluación de crédito* o una regla que especifique la política del negocio de una *revisión de crédito*.

**modelo del sistema** Imagen de un sistema que corresponde a la realidad o a una realidad deseada. Los modelos de sistemas facilitan el mejoramiento de la comunicación entre los usuarios, analistas, diseñadores y constructores del sistema.

**diseño lógico** La traducción de los requerimientos de los usuarios del negocio en un modelo de sistemas que muestre sólo los requerimientos del negocio y no cualquier posible diseño o ejecución técnica de esos requerimientos. Sus sinónimos frecuentes abarcan *diseño conceptual* y *diseño esencial*, de los cuales el segundo se refiere a modelar la “esencia” de un sistema o sus “requerimientos esenciales” con independencia de cualquier tecnología. El antónimo de diseño lógico es *diseño físico* (definido más adelante en este capítulo).

**parálisis de análisis** Término satírico acuñado para describir una situación común en proyectos, en la que el modelado excesivo del sistema desacelera el avance hacia la puesta en práctica de la solución de sistema que se pretende implantar.

<sup>5</sup> Quienes ya estén familiarizados con la elaboración de modelos orientados a objetos deben notar que los modelos de objetos tienden a desvanecer las fronteras de nuestro marco de referencia en cierto modo, aun así éste puede ser aplicado al problema que se va a resolver porque todavía está conducido por las tres metas de negocios fundamentales ilustradas en nuestro marco de trabajo. Esto se mostrará en los capítulos de este libro que tratan acerca del análisis y diseño orientado a objetos.



Antes de continuar a la siguiente fase, debemos señalar que la DEFINICIÓN DE ALCANCE, el ANÁLISIS DE PROBLEMA, LOS ANÁLISIS DE REQUERIMIENTOS y las FASES DE DISEÑO LÓGICO en conjunto son reconocidos por la mayoría de los expertos como el *análisis de sistema*. Algunos expertos también incluirían nuestra siguiente fase, ANÁLISIS DE DECISIÓN. Pero lo consideramos como una fase de transición de la fase de análisis a la fase de diseño porque hace la transición de los asuntos de negocios de los propietarios y los usuarios de sistemas a los asuntos de tecnología de los diseñadores y constructores de sistemas. Y desde luego, los analistas de sistemas son el factor que asegura la continuidad al hacer esta transición. Analicemos la transición.

**Análisis de decisión** Dados los requerimientos de negocios y los modelos de sistema lógicos, normalmente hay diversas alternativas para diseñar un nuevo sistema de información que satisfagan esos requerimientos. Algunas de las preguntas pertinentes incluyen lo siguiente:

- ¿Qué tanto del sistema debe ser automatizado con tecnología de la información?
- ¿Debemos comprar software o construirlo nosotros (llamado *decisión de comprar o desarrollar*)?
- ¿Debemos diseñar el sistema para una red interna o debemos diseñar una solución basada en la Web?
- ¿Qué tecnologías de información (posiblemente en surgimiento) podrían ser útiles para esta aplicación?

Estas preguntas se responden en la fase de ANÁLISIS DE LA DECISIÓN de la metodología. El propósito de esta fase es: 1) identificar soluciones técnicas candidatas, 2) analizar esas soluciones candidatas para evaluar su factibilidad, y 3) recomendar un sistema candidato como la solución objetivo para ser diseñada.

En la figura 3.6 podemos ver que la fase de análisis de la decisión está posicionada a la mitad a lo largo del proceso de desarrollo. La mitad de los componentes está posicionada más alto y la mitad más bajo. Esto es consistente con el papel de la fase de análisis de decisión como una transición del análisis al diseño y de los asuntos de negocios de los USUARIOS DEL SISTEMA a aquellos de los DISEÑADORES DEL SISTEMA (y finalmente, constructores de sistemas). Los diseñadores (los expertos técnicos en las tecnologías específicas) comienzan a tener un papel aquí junto con los usuarios del sistema y los ANALISTAS DEL SISTEMA. Los analistas ayudan a definir y analizar las alternativas. Las decisiones se toman en relación con las tecnologías que se utilizarán como parte de la arquitectura de la aplicación. Finalmente, los PROPIETARIOS DEL SISTEMA tendrán que aprobar o desaprobar las decisiones aprobadas ya que ellos pagan por el proyecto.

En la figura 3.5 se muestra que un análisis de decisión se dispara por los requerimientos de negocios validados más cualquier modelo lógico de sistemas y las especificaciones que se generan de esos requerimientos. El equipo del proyecto solicita ideas y opiniones para el diseño técnico e implantación hacia un público diverso, que posiblemente incluye proveedores de software de TI. Las soluciones candidatas se identifican y se clasifican de acuerdo con diversos criterios. Debe señalarse que muchas organizaciones modernas tienen estándares de arquitectura y de tecnología de la información que restringen el número de soluciones candidatas que podrían ser consideradas y analizadas. (La existencia de dichos estándares se ilustra al final del modelo de componentes de su sistema de información en la figura 3.6.) Luego de que las soluciones candidatas han sido identificadas, cada una se evalúa con los siguientes criterios:

- **Factibilidad técnica.** ¿Es la solución técnicamente práctica? ¿Nuestro personal tiene la experiencia técnica para diseñar y construir esa solución?
- **Factibilidad operacional.** ¿La solución podrá satisfacer los requerimientos de los usuarios? ¿A qué grado? ¿Cómo cambiará la solución el ambiente de trabajo del usuario? ¿Cómo se sienten los usuarios acerca de dicha solución?
- **Factibilidad económica.** ¿Tiene la solución un costo aceptable (como se definió anteriormente en el capítulo)?
- **Factibilidad de programa.** ¿Puede la solución ser diseñada e implantada dentro de un periodo aceptable?
- **Factibilidad de riesgos.** ¿Cuál es la probabilidad de una implantación exitosa al utilizar la tecnología y su enfoque?

El equipo de proyecto normalmente busca la solución *más* factible, aquella que ofrece la mejor combinación de factibilidad técnica, operacional, económica, de programa y de riesgo. Distintas soluciones candidatas pueden ser *más* factibles en un solo criterio; sin embargo, una solución normalmente demostrará ser la *más* factible cuando se consideran todos los criterios.

El producto fundamental de la fase de análisis de decisión es una PROPUESTA DEL SISTEMA que puede ser presentada por escrito o verbalmente. Diversos resultados son posibles. El punto de revisión de factibilidad del compromiso ajustado (de nuevo, el diamante negro) puede resultar en cualquiera de las siguientes opciones:

- Aprobar y financiar la propuesta de sistema para el diseño y la construcción (posiblemente al incluir un aumento en el presupuesto y cambios en el cronograma si el alcance ha sido expandido en forma considerable).
- Aprobar o financiar una de las soluciones candidatas alternativas.
- Rechazar todas las soluciones candidatas y ya sea cancelar el proyecto o regresarlo para nuevas recomendaciones.
- Aprobar una versión de alcance reducido de la solución propuesta.

Opcionalmente, la fase del análisis de decisión también puede producir una ARQUITECTURA DE APLICACIÓN para la solución aprobada. Tal modelo sirve como un plano de alto nivel (descripción general) para la propuesta recomendada o aprobada.

Antes de continuar, usted debe haber notado una variación en la figura 3.6 en el producto de la PROPUESTA DEL SISTEMA (O RFP). Esta variación es por una recomendación de adquirir la solución de hardware o software en lugar de construirlo internamente. Postergaremos cualquier análisis adicional de esta opción hasta más adelante en el capítulo cuando discutamos la variación de integración de paquete comercial de nuestro proceso básico.

**Diseño físico e integración** Dada la aprobación de la PROPUESTA DEL SISTEMA de la fase de análisis de decisión, al fin, usted puede diseñar el nuevo sistema. El propósito de la fase de DISEÑO FÍSICO E INTEGRACIÓN es transformar los requerimientos de negocios (representados en parte por los MODELOS LÓGICOS DEL SISTEMA) en las ESPECIFICACIONES DE DISEÑO FÍSICO que guiarán la construcción del sistema. En otras palabras, el diseño físico aborda con mayor detalle el *cómo* la tecnología será utilizada en el nuevo sistema. El diseño será restringido por el MODELO DE ARQUITECTURA desde la fase anterior. También, el diseño requiere apegarse a cualquier estándar de diseño técnico interno que asegure que sea completo, útil, confiable, con buen desempeño y calidad.

El **diseño físico** es lo opuesto del diseño lógico. Mientras que el diseño lógico trataba exclusivamente con requerimientos de negocios independientes de cualquier solución técnica, el diseño físico representa una solución técnica específica. En la figura 3.6 se demuestra la fase de diseño físico desde la perspectiva de sus componentes. Nótese que la fase de diseño se ocupa de los puntos de vista basados en la tecnología del sistema: 1) ESPECIFICACIONES DE DISEÑO FÍSICO DE BASES DE DATOS, 2) PROCESO FÍSICO DE NEGOCIOS Y ESPECIFICACIONES DE DISEÑO DE SOFTWARE Y 3) ESPECIFICACIONES DEL USUARIO FÍSICO E INTERFAZ DE SISTEMAS. EL DISEÑADOR DE SISTEMAS y el ANALISTA DE SISTEMAS (papeles que tal vez se traslapan con algunos de los mismos individuos) son los participantes fundamentales; sin embargo, ciertos aspectos del diseño normalmente tienen que ser compartidos con los USUARIOS DEL SISTEMA (por ejemplo, los diseños de pantalla y el flujo de trabajo). Usted puede haber tenido alguna experiencia en las especificaciones de diseño físico, ya sea en cursos de programación o de bases de datos.

Hay dos filosofías extremas del diseño físico.

- **Diseño por especificación.** Los modelos de sistema físico y especificaciones detallados son producidos como una serie de planos escritos (o generados por computadora) para la construcción.
- **Diseño por elaboración de prototipos.** Las aplicaciones o subsistemas incompletos pero que funcionan (llamados prototipos) se construyen y se refinan con base en la retroalimentación de los usuarios y otros diseñadores.

En la práctica, normalmente se realiza alguna combinación de estos extremos.

No existe ningún sistema de información nuevo en aislamiento de los demás sistemas de información existentes en una organización. En consecuencia, un diseño también debe reflejar los temas de integración de sistemas. El nuevo sistema debe ser integrado tanto

---

**diseño físico** La traducción de los requerimientos de un usuario de negocios en un modelo de sistemas que muestra una ejecución técnica de tales requerimientos. Sus sinónimos comunes abarcan *diseño técnico* o, para describir las salidas, *modelo de ejecución*. El antónimo del diseño físico es el diseño lógico (descrito con antelación en este mismo capítulo).

con otros sistemas de información como con los procesos de negocios. La integración normalmente se refleja en los modelos de sistema físicos y en las especificaciones de diseño.

En resumen, en la figura 3.5 se muestra que los productos de diseño físico y fase de integración incluyen alguna combinación de MODELOS DE DISEÑO FÍSICO Y ESPECIFICACIONES, PROTOTIPOS DE DISEÑO Y PROCESOS DE NEGOCIOS REDISEÑADOS. Nótese que hemos incluido una revisión de factibilidad de adelante o detenerse para el proyecto (el diamante negro). Un proyecto rara vez se cancela después de la fase de diseño a menos que esté demasiado por encima de lo presupuestado o retrasado en el programa. Por otro lado, el alcance puede ser disminuido para lograr un producto con un mínimo aceptable en un marco de tiempo especificado. O se puede extender el programa para construir una solución más completa en múltiples versiones. El plan de proyecto (programa y presupuesto) necesitaría ser ajustado para reflejar estas decisiones.

Debe señalarse que en las metodologías modernas, existe una tendencia hacia la fusión de la fase de diseño con nuestra siguiente fase, construcción. En otras palabras, las fases de diseño y construcción normalmente se traslapan.

**Construcción y pruebas** Dado algún nivel de MODELOS Y ESPECIFICACIONES DE DISEÑO FÍSICO (O PROTOTIPOS DE DISEÑO), podemos comenzar a construir y probar los componentes del sistema para ese diseño. En la figura 3.5 se muestra que el producto primario de la fase de PRUEBA Y CONSTRUCCIÓN ES UN SISTEMA FUNCIONAL que está listo para implantación. El propósito de la construcción y la fase de pruebas es doble: 1) Construir y probar un sistema que satisfaga los requerimientos de negocios y las especificaciones de diseño físico, y 2) implantar las interfaces entre el nuevo sistema y los sistemas existentes. Además, la DOCUMENTACIÓN FINAL (por ejemplo, sistemas de ayuda, manuales de capacitación, soporte de escritorio de ayuda, instrucciones de control de producción) se desarrollará en preparación para la capacitación y la operación del sistema. La fase de construcción puede también incluir una instalación de software adquirido.

En el marco de referencia de su sistema de información (figura 3.6) se identifican los componentes pertinentes y las actividades para la fase de construcción. El enfoque está en la última fila de componentes. El equipo del proyecto debe construir o instalar:

- BASES DE DATOS. Las bases de datos pueden incluir *bases de datos de procesamiento de transacciones en línea (online transaction processing, OLTP)* para respaldar las transacciones de negocios diarias, *almacenes de datos operacionales (operational data stores, ODS)* para soportar los informes, solicitudes diarias, y almacenes de datos que soporten las necesidades de análisis de datos y de toma de decisiones.
- PAQUETES DE SOFTWARE COMERCIAL Y/O SOFTWARE DESARROLLADO A LA MEDIDA. Los paquetes se instalan y se personalizan según sea necesario. Los programas de aplicación se desarrollan de acuerdo con el diseño físico o los prototipos de la fase anterior. Tanto los paquetes como el software personalizado deben ser probados completamente.
- INTERFACES DE USUARIO Y DE SISTEMA. Las interfaces de usuario (por ejemplo Windows e interfaces Web) deben ser construidas y probadas para su funcionalidad y estabilidad. Las interfaces de sistema a sistema deben ser construidas o implantadas con medio de tecnologías de integración. Nótese que el MIDDLEWARE (un tipo de software de sistemas) con frecuencia se utiliza para integrar diversas bases de datos, software y tecnologías de interfaz. Hablaremos más acerca del middleware en la fase de diseño de este texto.

En la figura 3.6 también se identifica a los participantes de esta fase como CONSTRUCTORES DE SISTEMAS, ANALISTAS DE SISTEMAS, USUARIOS DE SISTEMAS Y ADMINISTRADORES DEL PROYECTO. LOS DISEÑADORES DE SISTEMAS también pueden participar en la aclaración de especificaciones de diseño.

Usted probablemente tenga alguna experiencia con parte de esta actividad: Programación de aplicaciones. Los programas pueden ser escritos en muchos lenguajes distintos, pero la tendencia actual es hacia el uso de lenguajes de programación orientados a objetos tales como *Java*, *C++*, o *Visual Basic*. Conforme se construyen los componentes, generalmente se muestran a los usuarios con el fin de solicitar una retroalimentación.

Uno de los aspectos más importantes de construcción es la realización de pruebas de componentes de sistema individuales y el sistema en general. Una vez probado, un sistema (o versión de un sistema) está listo para INSTALACIÓN Y ENTREGA.

**Instalación y entrega** ¿Qué queda por hacer? Los nuevos sistemas normalmente representan un resultado de la forma en que los negocios se hacen hoy, por lo tanto, el analista

debe proporcionar una transición suave del viejo sistema al nuevo y ayudar a los usuarios a lidiar con los problemas normales de inicio. Por tanto, la fase de **INSTALACIÓN Y ENTREGA** sirve para entregar el sistema en operación (a veces llamado *producción*).

En la figura 3.5 el **SISTEMA FUNCIONAL** de la fase de construcción y pruebas es la entrada fundamental para la fase de **INSTALACIÓN Y ENTREGA**. El producto es un **SISTEMA OPERATIVO**. Los **CONSTRUCTORES DE SISTEMAS** instalan el sistema de su ambiente de desarrollo al ambiente de producción. Los **ANALISTAS DE SISTEMAS** deben capacitar a los **USUARIOS DE SISTEMAS**, escribir diversos manuales de control de producción y de usuario, convertir los archivos y bases de datos existentes a las nuevas bases de datos y realizar las pruebas finales del sistema. Los usuarios de sistemas proporcionan una retroalimentación continua conforme surgen los problemas y los diversos asuntos. Esencialmente, en la fase de instalación y entrega se consideran los mismos componentes que en la fase de construcción.

Para proporcionar una transición suave hacia el nuevo sistema se debe preparar un plan de conversión. Este plan puede requerir un cambio abrupto, donde se termina el sistema viejo y se reemplaza por el nuevo en una fecha específica. En forma alternativa, el plan puede mantener ejecutándose los sistemas viejo y nuevo en paralelo hasta que el nuevo sistema se considere aceptable para reemplazar al viejo.

La fase de instalación y entrega también incluye capacitar a los individuos que utilizarán el sistema final y desarrollar documentación para ayudar a los usuarios de sistemas. La fase de implantación normalmente incluye alguna forma de **REVISIÓN POSTERIOR A LA AUDITORÍA** para calcular el éxito del proyecto de sistemas terminado. Esta actividad promueve una mejora continua del proceso y una administración futura del proyecto.

**Operación del sistema y mantenimiento** Una vez que el sistema esté puesto en operación, requerirá un **soporte del sistema** continuo para el resto de su vida útil y productiva. El soporte de sistemas consiste en las siguientes actividades continuas:

- *Ayuda a usuarios.* Sin importar qué tan bien hayan sido capacitados los usuarios y qué tan completa y clara sea la documentación del usuario final, eventualmente los usuarios requerirán ayuda adicional conforme surjan los problemas no anticipados, se agreguen nuevos usuarios y demás.
- *Arreglar los defectos de software.* Los defectos de software son errores que se pasaron por alto en las pruebas del software. Éstos son inevitables, pero normalmente pueden ser resueltos, en la mayoría de los casos, con el soporte de un experto.
- *Recuperación del sistema.* Ocasionalmente, una falla del sistema puede resultar en un “colapso” de sistema y/o pérdida de datos. Los errores humanos o fallas en el hardware y el software pueden ocasionar esto. El analista de sistemas o los especialistas de soporte técnico pueden ser llamados para recuperar el sistema, es decir, restablecer los archivos y bases de datos del sistema y reiniciarlo.
- *Adaptar el sistema a requerimientos nuevos.* Los requerimientos nuevos pueden ser nuevos problemas de negocios, nuevos requerimientos de negocios, nuevos problemas técnicos o nuevos requerimientos de tecnología.

Eventualmente, se espera que la retroalimentación del usuario y los problemas o las necesidades de negocios cambiantes indiquen que es el momento de empezar de nuevo y reinventar el sistema. En otras palabras, el sistema ha llegado a una entropía y se debe iniciar un nuevo proyecto para crear un proceso de desarrollo de sistema completamente nuevo.

## > Actividades transversales del ciclo de vida

El desarrollo del sistema también incluye diversas **actividades transversales del ciclo de vida**. Estas actividades, listadas en la definición al margen, no se describen en forma explícita en la figura 3.5, pero son vitales para el éxito de cualquier proyecto. Analicemos brevemente cada una de estas actividades.

**Identificación de hechos** Hay muchas ocasiones para una **identificación de hechos** durante un proyecto. La identificación de hechos es más crucial para las primeras fases de un proyecto. Es durante estas fases que el equipo de proyecto aprende acerca del vocabulario del negocio, los problemas, oportunidades, restricciones, requerimientos y prio-

**soporte del sistema** Soporte de naturaleza técnica y continua para los usuarios de un sistema, así como el mantenimiento necesario para hacer frente a errores, omisiones o requerimientos nuevos que pueden surgir.

**actividad transversal del ciclo de vida** Toda actividad que se traslapa en múltiples fases del proceso de desarrollo de sistemas. Entre sus ejemplos, se cuentan la *identificación de hechos, documentación, presentación, estimación, análisis de factibilidad, administración de proyectos y procesos, administración del cambio y administración de la calidad.*

**identificación de hechos** Proceso formal de usar la investigación, entrevistas, juntas, cuestionarios, muestreo y otras técnicas para recopilar información acerca de problemas, requerimientos y preferencias concernientes al sistema. También se llama *recopilación de información o recopilación de datos.*

ridades. Pero la identificación de hechos también se utiliza durante las fases de análisis de decisión, diseño físico, construcción y pruebas e instalación y entrega, sólo que a un menor grado. Es durante estas últimas fases que el equipo de proyecto investiga alternativas técnicas y solicita una retroalimentación acerca de los diseños técnicos, estándares y componentes de trabajo.

**Documentación y presentación** Las habilidades de comunicación son esenciales para el cumplimiento exitoso de cualquier proyecto. De hecho, una mala comunicación con frecuencia es citada como la principal causa de retrasos de proyectos y repetición del trabajo. Dos formas de comunicación que son comunes en los proyectos de desarrollo de sistemas son la **documentación** y la **presentación**.

Claramente, las oportunidades de comunicación abarcan todas las fases. En la figura 3.7, las flechas negras representan diversos ejemplos de documentación de una fase. Las flechas discontinuas representan ejemplos donde las presentaciones se requieren con frecuencia. Finalmente, las flechas punteadas representan el almacenamiento de la documentación y otros artefactos de desarrollo de sistemas en un **repositorio**. Un repositorio guarda documentación para reutilizar y volver a trabajar en ella conforme sea necesario.

**Análisis de factibilidad** En consistencia con nuestro enfoque de compromiso ajustado al desarrollo de sistemas, el **análisis de factibilidad** es una actividad transversal del ciclo de vida. Diferentes mediciones de **factibilidad** son aplicables a distintas fases de la metodología. Estas mediciones incluyen factibilidad técnica, operacional, económica, de programa y de riesgo, como se describió cuando presentamos la fase de análisis de decisión. El análisis de factibilidad requiere buenas técnicas de **estimación**.

**Administración de proyecto y de proceso** Recuerde que CMM considera el desarrollo de sistemas como un *proceso* que debe ser manejado de proyecto en proyecto. Por ésta y otras razones, la administración de proceso y la administración de proyecto son actividades continuas, transversales del ciclo de vida. Ambos tipos de administración se presentaron anteriormente, pero sus definiciones se repiten en el margen de esta página y en el margen de la página 72 para su conveniencia. La **administración de proceso** define la metodología que se utilizará en cada proyecto, piense en ella como la *receta* para construir un sistema. La **administración de proyecto** está preocupada por administrar un solo ejemplo del proceso, en relación con un solo proyecto.

Los fracasos y éxitos limitados de los proyectos de desarrollo de sistemas a menudo exceden a los proyectos exitosos. ¿Por qué es eso? Una razón es que muchos analistas de sistemas no están familiarizados o son indisciplinados acerca de la adecuada aplicación de herramientas y técnicas del desarrollo de sistemas. Pero la mayoría de los fracasos son atribuidos a un mal liderazgo y administración. Esta mala administración ocasiona requerimientos no satisfechos o no identificados, exceso en los costos y entregas tardías.

## > Desarrollo secuencial o iterativo

El análisis anterior de las fases podría llevarle a asumir que el desarrollo de sistemas es un proceso naturalmente secuencial, que se mueve en una dirección de un solo sentido de fase en fase. Dicho desarrollo secuencial es, de hecho, una alternativa. Este método se describe en la parte *a*) en la figura 3.8. Para mayor simplicidad en la figura hemos utilizado las cuatro fases clásicas en lugar de las ocho fases *FAST*. Esta estrategia requiere que cada fase sea “completada” una después de la otra hasta que el sistema de información esté terminado. En la realidad, las fases pueden traslaparse de algún modo en algún momento. Por ejemplo, algún diseño de sistemas puede ser iniciado antes del cumplimiento del análisis de sistemas. Dada su apariencia visual de tipo cascada, este método a menudo es llamado **estrategia de desarrollo en cascada**.

La estrategia en cascada ha perdido adeptos con la mayoría de los desarrolladores de sistema modernos. Una estrategia más popular, que se muestra en el inciso *b*) de la figura 3.8, es referida comúnmente como la **estrategia de desarrollo iterativo** o proceso de desarrollo incremental. Este método requiere completar suficiente análisis, diseño e implantación para ser capaz de desarrollar completamente una *parte* del nuevo sistema y de colocarlo en operación tan rápido como sea posible. Una vez que la versión del sistema se implanta, la estrategia entonces es realizar un análisis adicional, diseño e implantación

**documentación** Actividad continua de registrar hechos y especificaciones de un sistema para referencia actual y futura.

**presentación** Actividad continua de comunicar hallazgos, recomendaciones y documentación para su revisión por los usuarios interesados y administradores. Las presentaciones pueden ser escritas o verbales.

**repositorio** Base de datos o directorio de archivos donde los desarrolladores de sistemas guardan toda la documentación, conocimientos y artefactos de uno o más proyectos o sistemas de información. Los repositorios usualmente se automatizan para facilitar el almacenamiento, recuperación, así como para compartir la información.

### análisis de factibilidad

Actividad en la que se mide y evalúa la factibilidad.

### factibilidad

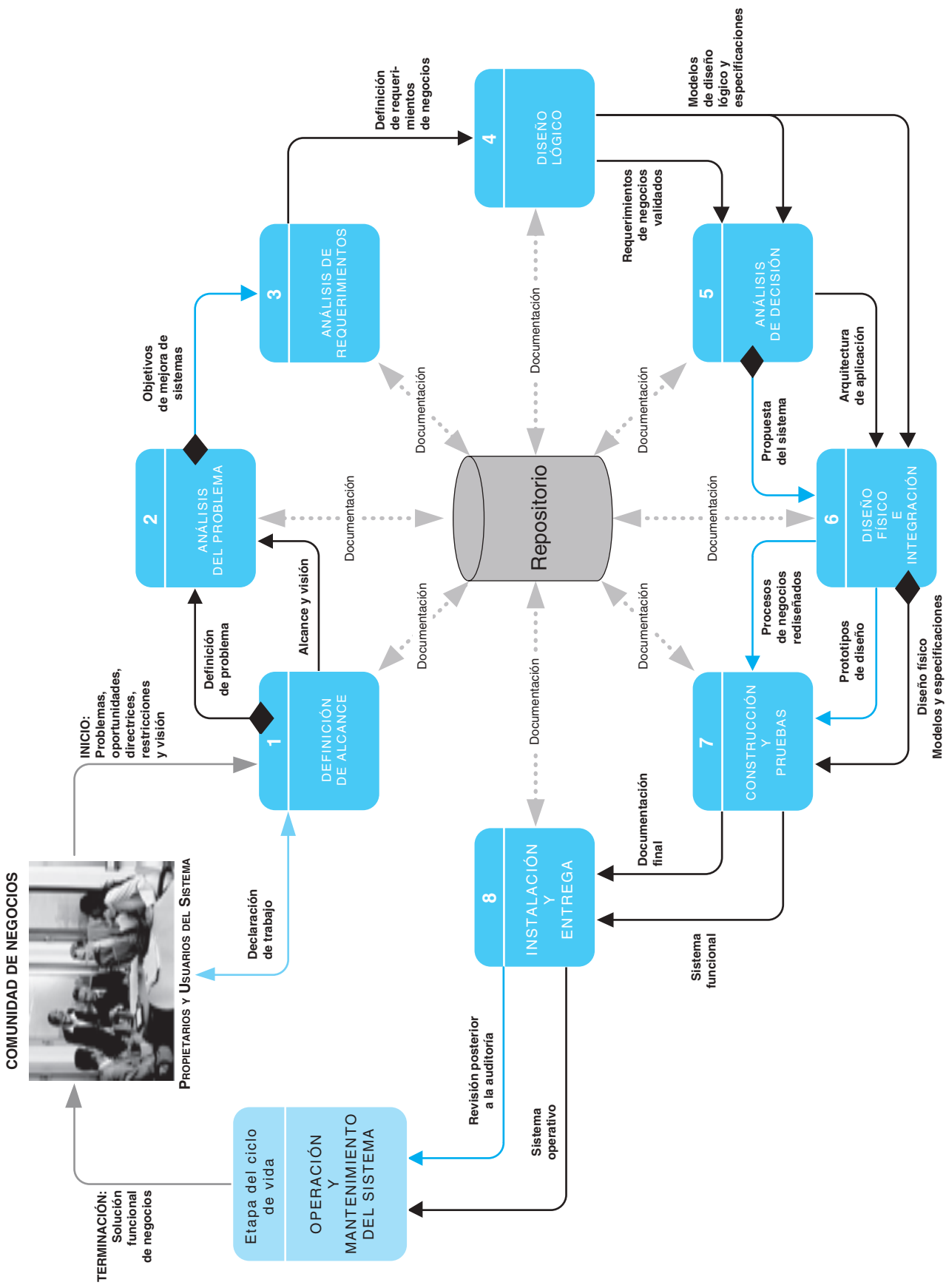
Medición de cuán benéfico sería el desarrollo de un sistema de información para una organización.

### estimación

Predicción calculada de los costos y esfuerzos necesarios para el desarrollo de un sistema. Un sinónimo hasta cierto punto menos pretencioso es *conjetura informada*, que normalmente significa que la estimación se basa en la experiencia como datos empíricos, si bien carece de rigor, en otras palabras, una simple conjetura.

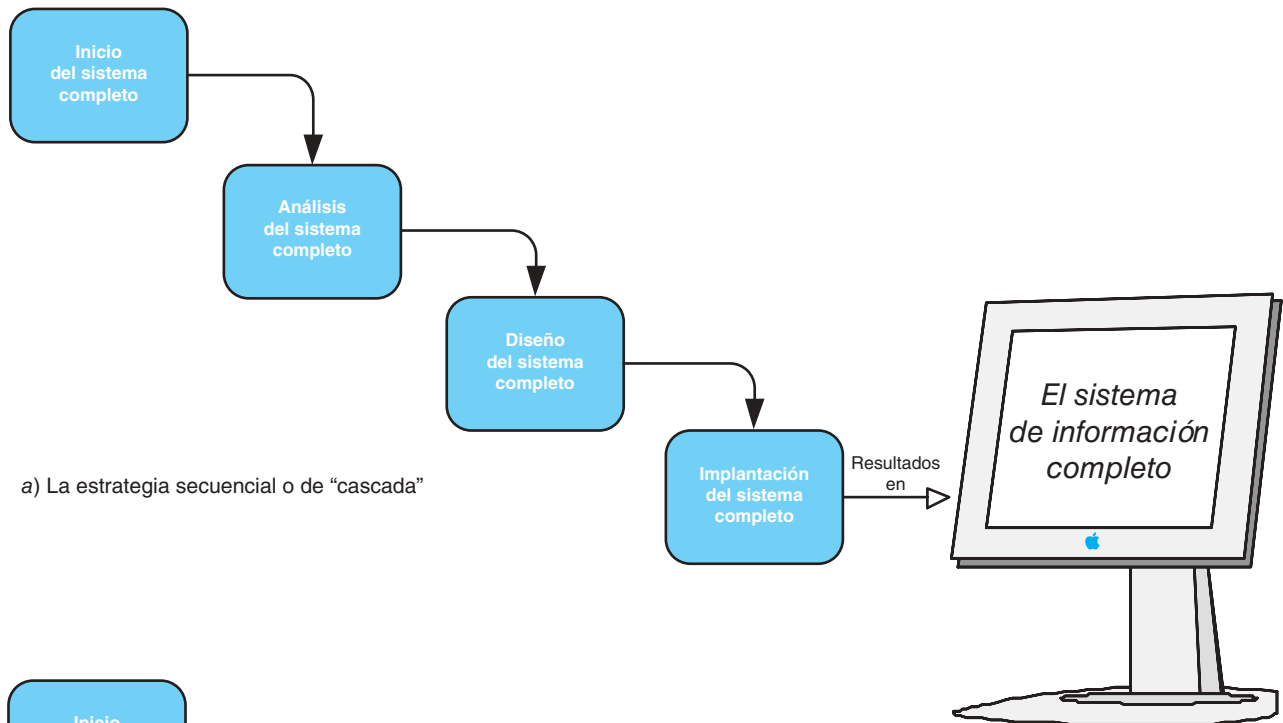
### administración de proceso

Actividad continua que documenta, enseña, supervisa y mejora el uso de los métodos (el “proceso”) y estándares que elige una organización para el desarrollo de sistemas. La administración de procesos se relaciona con fases, actividades, productos y normas de calidad que se deben aplicar de manera constante a todos los proyectos.

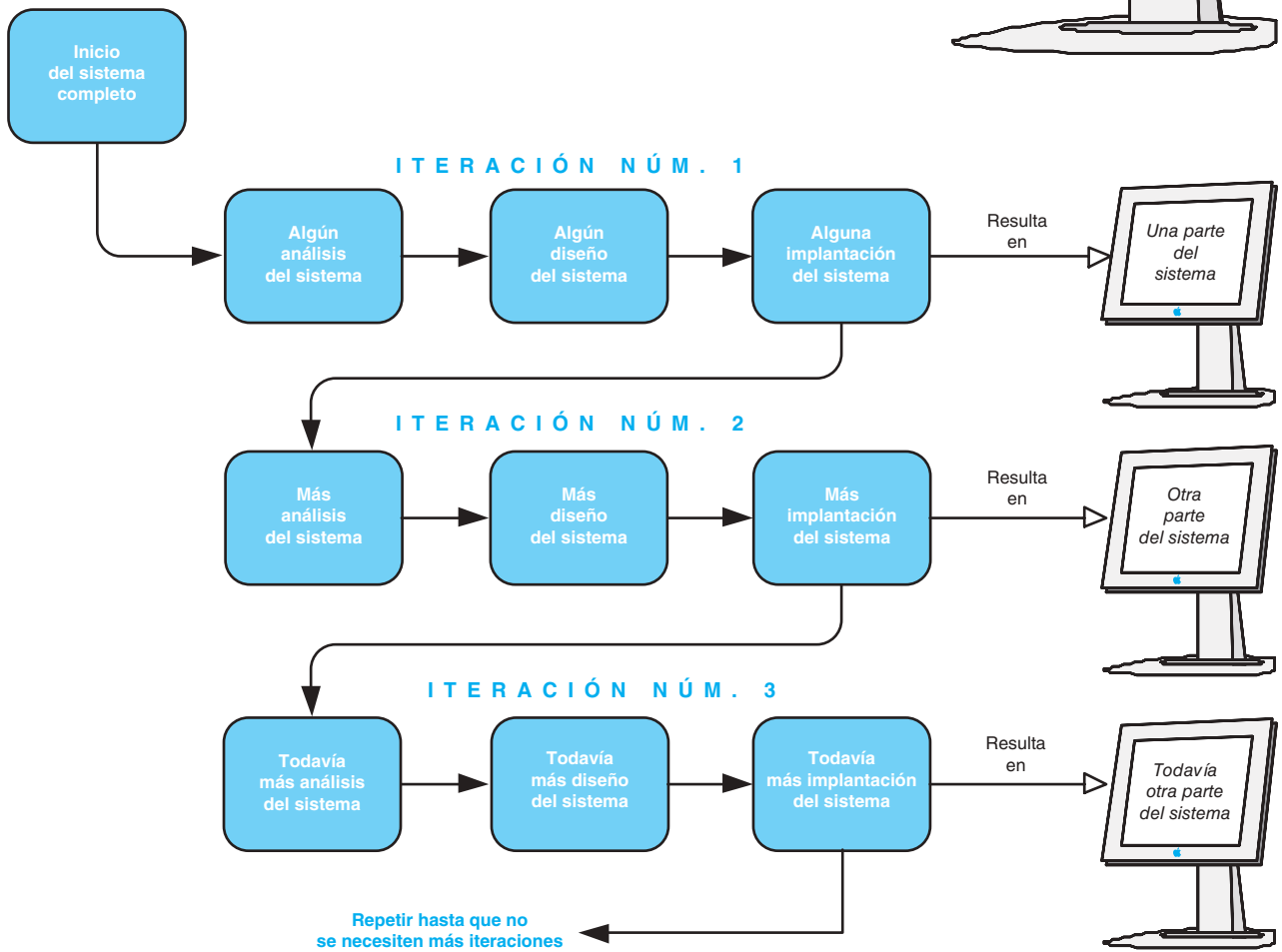


**FIGURA 3.7** Documentación, repositorio y presentación del desarrollo de sistemas





a) La estrategia secuencial o de "cascada"



b) La estrategia iterativa o incremental

**FIGURA 3.8** Método de desarrollo de sistemas secuencial o iterativo

**administración de proyecto** Proceso de definir el alcance, planear, dotar de personal, organizar, dirigir y controlar un proyecto para desarrollar un sistema de información con costo mínimo, dentro del tiempo especificado y con calidad aceptable.

para lanzar la siguiente versión del sistema. Estas iteraciones continúan hasta que todas las partes del sistema de información completo han sido implantadas. La popularidad de este proceso iterativo e incremental puede ser explicada de manera simple: los propietarios y los usuarios del sistema se han quejado por mucho tiempo acerca del tiempo excesivo requerido para desarrollar e implantar sistemas de información con la estrategia de cascada. El método iterativo permite que versiones de sistemas de información sean entregadas en marcos temporales regulares y más breves. Esto ocasiona un aumento de la satisfacción de los clientes (propietarios y usuarios del sistema).

## Rutas y estrategias alternativas

**estrategia de desarrollo en cascada** Una forma de análisis y diseño de sistemas que completa cada fase una después de otra y solamente una vez.

**estrategia de desarrollo iterativo** Estrategia de análisis y diseño de sistemas que completa el sistema de información en iteraciones sucesivas. Cada iteración realiza una cierta parte del análisis, diseño y construcción. Sus sinónimos incluyen estrategia de desarrollo incremental y espiral.

Al definir algún destino, hay muchas rutas hacia él y muchos modos de transporte. Usted podría tomar la supercarretera, la carretera o el camino secundario, o usted podría volar. Decidir qué ruta es la mejor depende de sus metas y prioridades. ¿Quiere llegar ahí rápido o quiere ver el paisaje? ¿Cuánto está dispuesto a gastar? ¿Se siente cómodo con la forma de viajar? Al igual que elegiría su ruta y medio de transporte a algún destino, usted puede y debe elegir una ruta y medios para un destino de desarrollo de sistemas.

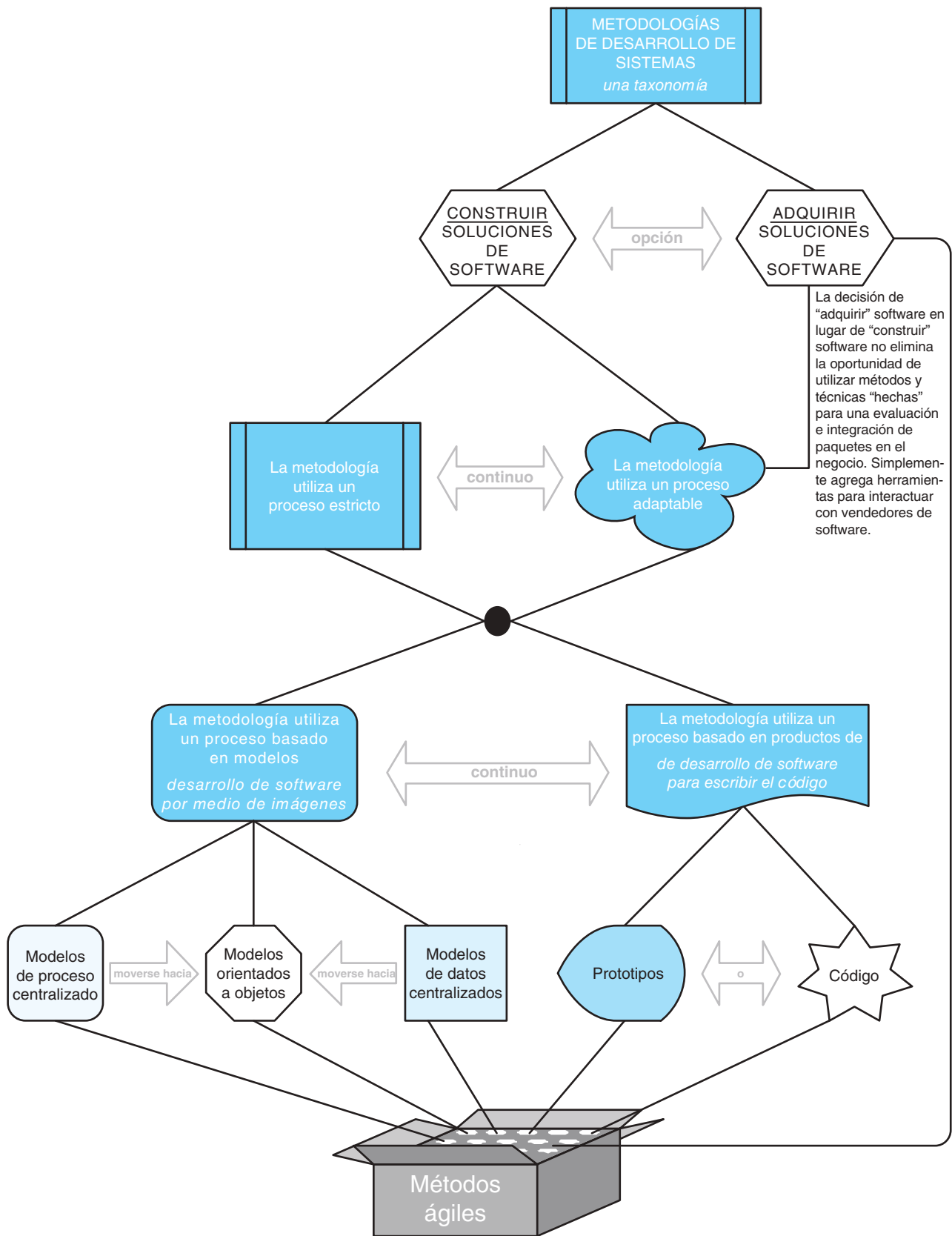
Hasta ahora, hemos descrito un conjunto básico de fases que comprenden nuestra metodología *FAST*. En alguna ocasión, una metodología de “una talla para todos” era común para la mayoría de los proyectos; sin embargo, actualmente existe una diversidad de tipos de proyectos, tecnologías y estrategias de desarrollo, ¡ya no hay una sola talla para todos los proyectos! Al igual que muchas metodologías contemporáneas, *FAST* proporciona rutas y estrategias alternativas para acomodar distintos tipos de proyectos, metas de tecnología, habilidades de desarrollo y paradigmas del desarrollo.

En esta sección, describiremos diversas rutas y estrategias *FAST*. Antes de hacerlo, analice la figura 3.9 en la siguiente página. En ella se ilustra una taxonomía o un esquema de clasificación para estrategias de metodología. Nótese lo siguiente:

- Las metodologías y las rutas pueden soportar la opción de ya sea construir *soluciones de software internas* o *adquirir una solución de software comercial* de un proveedor de software. Generalmente, gran parte de los mismos métodos y técnicas son aplicables a ambas opciones.
- Las metodologías pueden ser muy *estrictas* (“Tocar todas las bases; seguir todas las reglas”) o relativamente *adaptables* (“cambiar conforme sea necesario dentro de ciertos lineamientos”).
- Las metodologías pueden también ser caracterizadas como *conducidas por el modelo* (“dibujar imágenes del sistema”) o *conducidas por el producto* (“construir el producto y ver cómo reaccionan los usuarios”).
- Las metodologías conducidas por el modelo se mueven rápidamente para enfocarse en las tecnologías *orientadas a objetos* que se utilizan para construir la mayoría de los sistemas actuales (más acerca de esto posteriormente). Antes los métodos conducidos por el modelo enfatizaban ya sea la elaboración de modelos de proceso o la elaboración de modelos de datos.
- Finalmente, los enfoques conducidos por el producto tienden a enfatizar su *elaboración de prototipos* acelerados o a escribir el *código* de programa lo más pronto posible (tal vez usted ha escuchado el término *programación extrema*).

¡Cuántas estrategias! ¿Cuál debe elegir? Se está formando un movimiento conocido como *métodos ágiles*. En una palabra, los defensores de los métodos ágiles sugieren que los analistas y programadores de sistemas deben tener una caja de herramientas de métodos que incluyen herramientas y técnicas de todas las metodologías antes mencionadas. Deben elegir sus herramientas y técnicas con base en el problema y la situación. *FAST* es una metodología ágil. Defiende el uso integrado de las herramientas y técnicas de muchas metodologías, aplicado en el contexto de procesos repetibles (como en el nivel 3 CMM). Dicho esto, analicemos algunas de las variaciones y estrategias de ruta para el proceso *FAST*. Conforme navegamos por cada ruta, utilizaremos textos en tipo *itálico* (o cursivo) y flechas con líneas discontinuas para resaltar aquellos aspectos de la ruta que difieren de la ruta básica que usted ya ha aprendido.





**FIGURA 3.9** Una taxonomía para metodologías y estrategias de desarrollo de sistemas

## > Estrategia de desarrollo basado en modelos

Uno de los métodos más antiguos y más comúnmente utilizados para analizar y diseñar sistemas de información está basado en el modelado de sistemas. Como recordatorio, un modelo de sistemas es la imagen de un sistema que representa la realidad o una realidad deseada. Los modelos de sistemas facilitan una comunicación mejorada entre los usuarios de sistemas, analistas de sistemas, diseñadores de sistemas y constructores de sistemas. En la metodología *FAST*, los modelos de sistemas se utilizan para ilustrar y comunicar los componentes del CONOCIMIENTO, PROCESO O INTERFAZ de los sistemas de información. Este método se llama **desarrollo basado en modelos**.

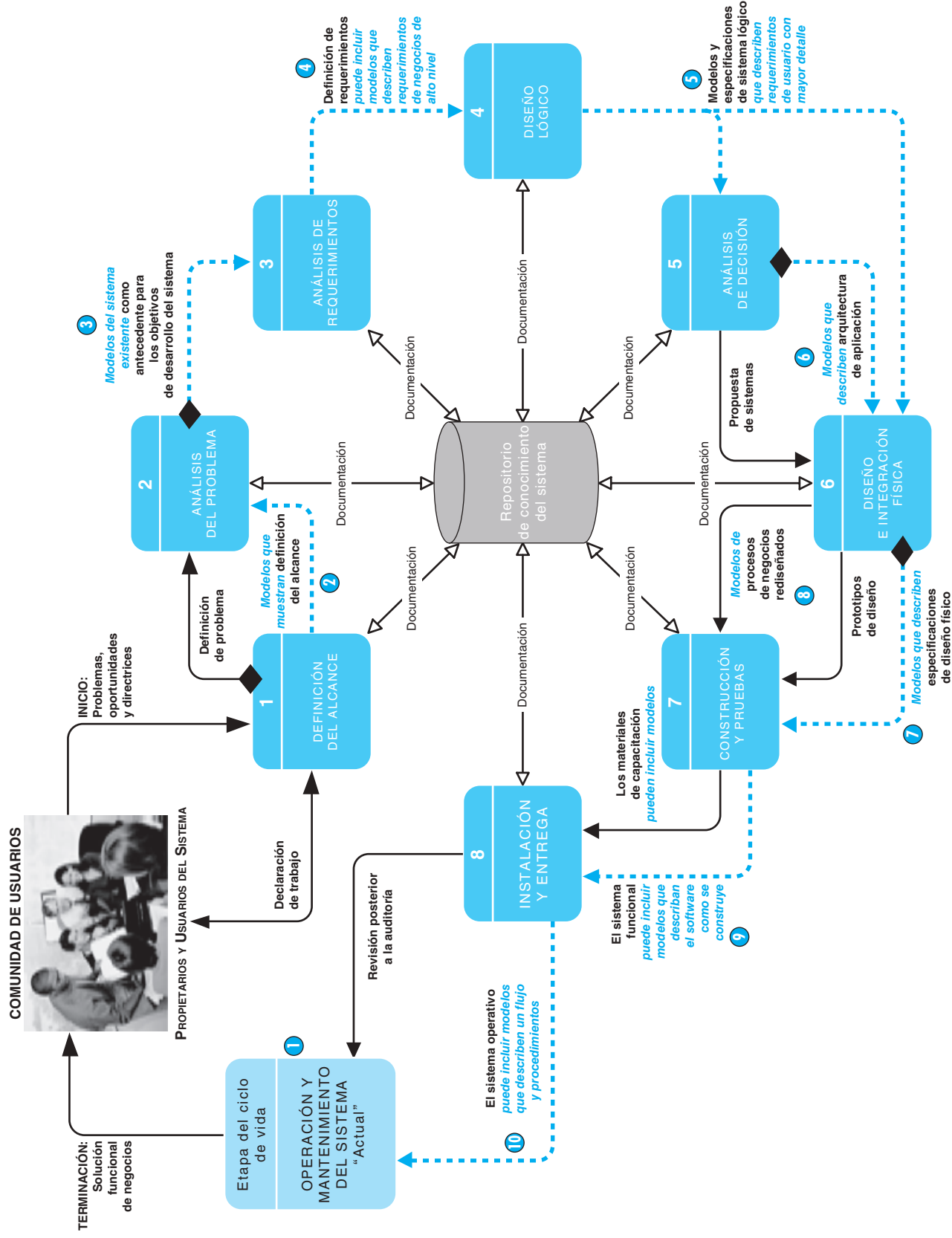
La ruta del desarrollo basado en modelos de *FAST* se ilustra en la figura 3.10. El método basado en modelos no varía mucho de las fases básicas que describimos anteriormente. Preste atención a las siguientes notas que corresponden a las viñetas numeradas:

- ❶ Los modelos de sistemas pueden existir del proyecto que creó el sistema actual. ¡Cuidado! Estos modelos son notorios por ser anticuados. Pero todavía pueden ser útiles como punto de partida.
- ❷ Anteriormente usted aprendió que es importante definir el alcance de un proyecto. Una de las formas más simples de comunicar el alcance es al dibujar **MODELOS QUE MUESTREN UNA DEFINICIÓN DEL ALCANCE**. Los modelos de alcance muestran cuáles aspectos de un problema están dentro del alcance y cuáles están fuera. Esto a veces se llama *diagrama de contexto* o modelo de contexto.
- ❸ Algunas técnicas de modelado de sistemas requieren extensos **MODELOS DEL SISTEMA EXISTENTE** para identificar problemas y oportunidades para la mejora del sistema. Esto a veces es llamado *modelo de sistema tal como está*. Hoy en día, la elaboración de modelos del sistema actual ha declinado en popularidad. Muchos administradores de proyectos y analistas lo ven como contraproducente o de poco valor. La excepción es la elaboración de procesos de negocios “tal como es”, con el fin del rediseño de procesos de negocio.
- ❹ La definición de requerimientos es uno de los productos más importantes del desarrollo de sistemas. A veces incluye **MODELOS QUE DESCRIBEN REQUERIMIENTOS DE ALTO NIVEL**. Una de las técnicas de elaboración de modelos más populares actualmente es llamada *caso de uso* (presentada en el capítulo 6). Los casos de uso (o casos prácticos) identifican requerimientos y rastrean su cumplimiento a lo largo del ciclo de vida.
- ❺ La mayoría de las técnicas basadas en modelos requieren que analistas documenten los requerimientos de negocios con **modelos lógicos** (definidos anteriormente). Los requerimientos de negocios con frecuencia se expresan en **MODELOS LÓGICOS QUE DESCRIBEN REQUERIMIENTOS DEL USUARIO MÁS DETALLADOS**. Ellos muestran sólo *lo que* un sistema debe ser o hacer. Son de implantación *independiente*; es decir, describen el sistema independiente de cualquier implantación técnica posible. Por tanto, son útiles para describir y validar requerimientos de negocios.
- ❻ Como resultado de la fase de análisis de decisión, el analista puede producir **MODELOS QUE DESCRIBEN LA ARQUITECTURA DE APLICACIÓN**. Dichos modelos ilustran la implantación técnica planeada de un sistema.
- ❼ Muchas técnicas basadas en modelos requieren que los analistas desarrollen **MODELOS QUE DESCRIBEN ESPECIFICACIONES FÍSICAS DE DISEÑO** (definidas anteriormente en este capítulo). Recuerde que los **modelos físicos** muestran no sólo lo que un sistema es o hace, sino también *cómo* se implanta el sistema con la tecnología. Dependen de la implantación debido a que reflejan las opciones de tecnología y las limitaciones de aquellas opciones de tecnología. Algunos ejemplos incluyen esquemas de base de datos, cuadros de estructura y diagramas de flujo. Sirven como un plano para la construcción del nuevo sistema.
- ❽ Los nuevos sistemas de información deben ser “entrelazados en la tela” de los procesos de negocios de una organización. En consecuencia, el analista y los usuarios pueden desarrollar **MODELOS DE PROCESOS REDISEÑADOS DEL NEGOCIO**.

**desarrollo basado en modelos** Estrategia de desarrollo de sistemas que hace énfasis en trazar modelos de sistemas como ayuda para visualizar y analizar los problemas, definir los requerimientos de negocios y diseñar los sistemas de información.

**modelo lógico** Representación visual de *qué es* o *hace* un sistema. Sus sinónimos son modelo esencial, modelo conceptual y modelo de negocios.

**modelo físico** Representación visual técnica que describe *qué es* un sistema o *qué hace* y *cómo* ponerlo en práctica. Sus sinónimos son modelo de implantación y modelo técnico.



**FIGURA 3.10** Estrategia de desarrollo de sistemas basada en modelos

- 9 La construcción traduce los modelos de sistema físicos en software. En algunos casos, existen herramientas automatizadas para traducir en forma automática el software en **MODELOS FÍSICOS QUE DESCRIBEN EL SOFTWARE CONSTRUIDO**. Esto se llama *ingeniería inversa*.
- 10 Finalmente, el sistema operacional puede incluir **MODELOS QUE DESCRIBEN FLUJO Y PROCEDIMIENTO**. Por ejemplo, los modelos de sistemas pueden documentar procedimientos de respaldo y de recuperación.

En resumen, los modelos de sistemas pueden ser producidos como una porción de los productos de la mayoría de las fases. Los métodos basados en modelos enfatizan la elaboración de modelos de sistemas. Una vez implantados, los modelos de sistemas sirven como documentación para cualquier cambio que pueda ser necesario durante la etapa de operación y soporte del ciclo de vida.

Se cree que el método basado en modelos ofrece diversas ventajas y desventajas, como se lista a continuación:

### Ventajas

- La especificación de requerimientos tiende a ser más completa y mejor documentada.
- Los requerimientos de negocios y diseños de sistemas son más fáciles de validar con imágenes que con palabras.
- Es más fácil identificar, conceptualizar y analizar soluciones técnicas alternativas.
- Las especificaciones de diseño tienden a ser más profundas, estables, adaptables y flexibles debido a que están basadas en modelos y a que son analizadas más profundamente *antes* de ser construidas.
- Los sistemas pueden ser construidos en forma más correcta la primera vez cuando se construyen de especificaciones basadas en modelos claros y completos. Algunos afirman que el software de generación de código puede generar automáticamente un esquema o un código casi completo conforme a modelos adecuados de sistemas.

### Desventajas

- Consume mucho tiempo. Toma tiempo recopilar los hechos, dibujar los modelos y validar esos modelos. Esto es especialmente verdadero si los usuarios están inciertos o si son imprecisos acerca de sus requerimientos de sistemas.
- Los modelos pueden ser sólo tan buenos como la comprensión de los usuarios de esos requerimientos.
- Las imágenes no son software; algunos afirman que esto reduce el papel de los usuarios en un proyecto a una participación pasiva. La mayoría de los usuarios no se emociona acerca de las imágenes. En lugar de eso, quieren ver un software en funcionamiento y evalúan el progreso del proyecto por la existencia (o ausencia) de software.
- El método basado en modelos es considerado como inflexible por algunos: los usuarios deben especificar por completo los requerimientos antes del diseño, el diseño debe documentar completamente las especificaciones técnicas antes de la construcción y demás. Algunos ven dicha rigidez como poco práctica.

### modelado de proceso

Técnica centrada en procesos que se popularizó con el *análisis estructurado* y la metodología *de diseño* en los que se aplican modelos de requerimientos de procesos de negocios con el fin de derivar diseños de software efectivos para un sistema. Con el análisis estructurado, se introdujo una herramienta de modelado que se llama *diagrama de flujo de datos* para ilustrar el flujo por una sucesión de procesos de negocios. El diseño estructurado convirtió los diagramas de flujo de datos en un modelo de procesos llamado *gráficos estructurales*, para ilustrar una estructura descendente de software que satisface los requerimientos de negocios.

El desarrollo basado en modelos es más efectivo para sistemas para los que los requerimientos están bien entendidos y que son tan complejos que requieren grandes equipos de proyecto para completar. El método también funciona bien cuando la satisfacción de las expectativas del usuario y la calidad son más importantes que el costo y el programa.

Hay varias técnicas distintas basadas en modelos. Difieren principalmente en términos de los tipos de modelos que requieren que dibuje y valide el analista de sistemas. Analicemos brevemente tres de las técnicas de desarrollo basadas en modelos más populares que serán enseñadas en este texto. Nótese que aquí presentamos sólo las técnicas, no los modelos. Enseñaremos los modelos más adelante, en los capítulos del “cómo”.

**Modelado de procesos** La **elaboración de modelos de procesos** fue fundada en el análisis estructurado y metodologías de diseño en 1978. Mientras que el análisis y diseño estructurado ha perdido adeptos como metodología, la elaboración de modelos de pro-

ceso continúa como una técnica viable e importante. Recuerde que los componentes de sistemas de información incluyen varios posibles enfoques: CONOCIMIENTO, PROCESOS e INTERFACES. La elaboración de modelos de proceso se enfoca en la columna del PROCESO de los componentes. Los *diagramas de flujo* son un tipo de modelo de proceso (utilizado principalmente por los CONSTRUCTORES DE SISTEMAS) que usted puede haber encontrado en un curso de programación. La elaboración de modelos de procesos ha disfrutado un cierto renacimiento con el surgimiento del rediseño de proceso de negocios (presentado en el capítulo 1).

Los diagramas de flujo de datos y los cuadros de estructura han contribuido significativamente a reducir la brecha de comunicaciones que con frecuencia existe entre los propietarios y usuarios de sistemas no técnicos y los diseñadores y constructores técnicos de sistemas. La elaboración de modelos de proceso se enseña en este texto.

**Modelado de datos** Recuerde que la mejora del CONOCIMIENTO es una meta fundamental y un conjunto de componentes en su marco de referencia. El conocimiento es el producto de la *información*, que a su vez es el producto de los *datos*. Los métodos de **elaboración de modelado de datos** enfatizan los componentes del conocimiento, especialmente datos. En el método de elaboración de modelos de datos, el énfasis se coloca en los diagramas que capturan requerimientos de datos de negocios y los traducen en diseños de bases de datos. Posiblemente, la elaboración de modelos es la técnica de elaboración de modelos de sistemas más ampliamente practicada. Por tanto, la enseñaremos en este libro.

**Elaboración de modelos de objetos** La elaboración de modelos de objetos es el resultado del avance técnico. Actualmente la mayoría de los lenguajes y métodos de programación están basados en el surgimiento de la tecnología de objetos. Aunque los conceptos de la tecnología de objetos están cubiertos ampliamente a lo largo de este libro, aquí es apropiado hacer una introducción simplificada.

Durante los últimos 30 años, técnicas como la elaboración de modelos de proceso y de datos deliberadamente separaban los asuntos de PROCESOS de los de DATOS. En otras palabras, los modelos de proceso y de datos eran separados y distintos. Como casi todos los sistemas incluían procesos y datos, las técnicas con frecuencia se utilizaban en paralelo y los modelos tenían que ser cuidadosamente sincronizados. Las técnicas de objeto son un intento por eliminar la separación de ambos aspectos y por tanto, la necesidad de sincronización de asuntos de datos y de proceso. Esto ha provocado un aumento en los métodos de **elaboración de modelos de objetos**.

Los objetos de negocios corresponden a cosas reales de importancia en el negocio como *clientes* y *pedidos* que colocan de *productos*. Cada objeto consiste tanto de los datos que describen el objeto como los procesos que pueden crear, leer, actualizar y eliminar ese objeto. Con relación a los componentes de su sistema de información, el análisis y diseño orientado a objetos (object-oriented analysis and design, OOAD) cambia significativamente el paradigma. Las columnas de DATOS y PROCESOS (y tal vez la columna de INTERFAZ también) son fusionadas en una sola columna de OBJETO. Los modelos entonces se enfocan en identificar, construir y ensamblar objetos apropiados, como con los *Legos*, para lograr sistemas de información útiles.

La popularidad actual de la tecnología de objetos está dirigiendo el interés a los modelos de objetos y al OOAD. Por ejemplo, actualmente la mayoría de los sistemas operativos más populares como son *Windows* de Microsoft y *Mac/OS* de Apple tienen interfaces de usuario orientadas a objetos ("señale y haga clic", mediante objetos como ventanas, marcos, menús descendentes, botones de radio, casillas de verificación, barras de desplazamiento y demás). Las interfaces de usuario Web como *Internet Explorer* de Microsoft y el *Navigator* de Netscape también están basadas en la tecnología de objetos. Los lenguajes de programación de objetos como *Java*, *C++*, *C++*, *Smalltalk* y *Visual Basic.NET* se utilizan para construir y ensamblar dichos sistemas operativos y aplicaciones orientados a los objetos. Y esos mismos lenguajes se han convertido en las herramientas de elección para construir aplicaciones de sistemas de información de la siguiente generación. No es de sorprender que las técnicas de modelado de objetos hayan sido creadas para expresar requerimientos de negocios y de software y diseños en términos de objetos. Esta edición del texto integra de manera extensa las técnicas de elaboración de modelos más populares para prepararlo para el análisis y diseño de sistemas que finalmente produce los sistemas y aplicaciones de información basadas en objetos.

---

**modelado de datos** Técnica centrada en datos que se usa para modelar los requerimientos de datos de las empresas y diseñar sistemas de bases de datos que satisfagan esos requerimientos. Los modelos de datos más frecuentes son los *diagramas de relaciones de entidades*.

---

**elaboración de modelos de objetos** Técnica que intenta fusionar los datos y procesos en un solo concepto, llamado *objeto*. Los modelos de objetos son diagramas que documentan un sistema en cuanto a sus objetos y sus interacciones. El modelado de objetos es la base del *análisis orientado a objetos* y de las metodologías de *diseño*.

### desarrollo rápido de aplicaciones (RAD)

Estrategia de desarrollo de sistemas que hace énfasis en la rapidez de desarrollo mediante la participación amplia del usuario en la construcción rápida, iterativa e incremental de una sucesión de **prototipos** funcionales de un sistema, que tarde o temprano se convierten en el sistema definitivo (o en una versión de él).

**prototipo** Modelo a pequeña escala, representativo o de trabajo de los requerimientos de los usuarios o de un diseño propuesto de un sistema de información. Todo prototipo puede omitir ciertas funciones o características hasta el momento en que haya evolucionado en grado suficiente para convertirse en una puesta en práctica aceptable de los requerimientos.

## > Estrategia de desarrollo rápido de aplicaciones

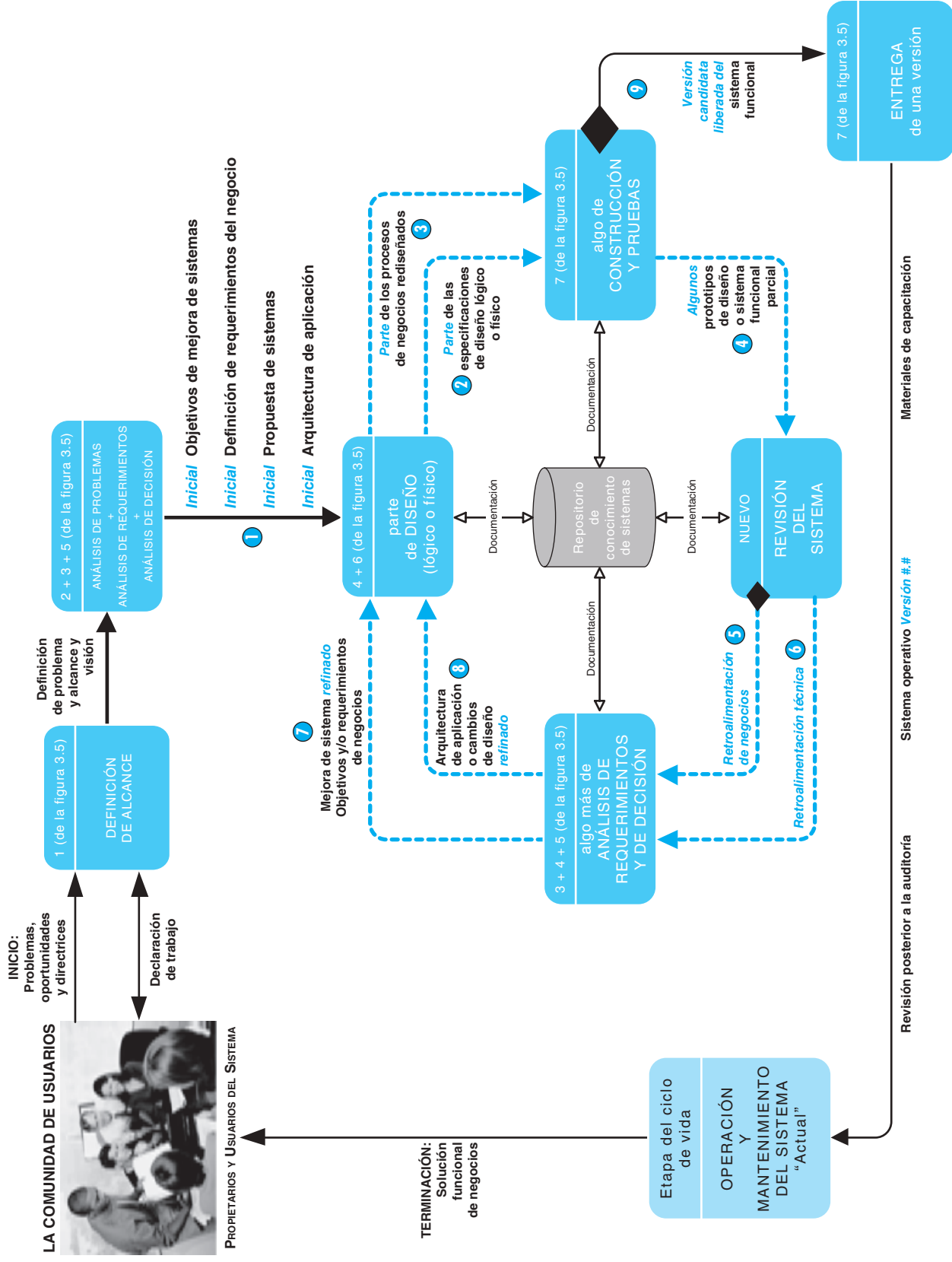
En respuesta al paso más rápido de la economía en general, el **desarrollo rápido de aplicaciones (rapid application development, RAD)** se ha convertido en una ruta popular para acelerar el desarrollo de sistemas. Las ideas básicas del RAD son:

- Incluir de manera más activa a los usuarios en las actividades de análisis, diseño y construcción.
- Organizar el desarrollo de sistemas en una serie de talleres intensos y enfocados que juntos incluyen PROPIETARIOS, USUARIOS, ANALISTAS, DISEÑADORES y CONSTRUCTORES DE SISTEMAS.
- Acelerar los análisis de requerimientos y fases de diseño a través de un método de construcción iterativo.
- Reducir la cantidad de tiempo que pasa antes de que los usuarios comiencen a ver un sistema en funcionamiento.

El principio básico detrás de la elaboración de prototipos es que los usuarios saben lo que quieren cuando lo ven en funcionamiento. En RAD, un **prototipo** eventualmente evoluciona hacia el sistema de información final. La ruta RAD para *FAST* se ilustra en la figura 3.11. De nuevo, el texto en tipo cursivo o itálico y los flujos indican las desviaciones del proceso *FAST* básico. Preste atención a las siguientes notas que corresponden a las viñetas numeradas:

- 1 El énfasis está en reducir tiempo y en desarrollar aplicaciones y sistemas; por tanto, las fases de análisis de problema inicial, análisis de requerimientos y análisis de decisión se consolidan y se aceleran. Los productos son típicamente reducidos, de nuevo por el interés del tiempo. Se dice que los productos son *INICIALES*, lo que significa “se espera que cambien” conforme el proyecto progresa.
 

Luego del análisis inicial anterior, el RAD utiliza un método iterativo, como se discutió antes en el capítulo. Cada iteración enfatiza una nueva funcionalidad que se debe lograr en pocas semanas.
- 2 Las *ESPECIFICACIONES DE DISEÑO LÓGICO Y FÍSICO* normalmente se abrevian y se aceleran de manera significativa. En cada iteración del ciclo, sólo se considerarán algunas especificaciones de diseño. Mientras que algunos modelos de sistemas pueden ser dibujados, se eligen en forma selectiva y el énfasis continúa en un rápido desarrollo. La suposición es que los errores pueden detectarse y arreglarse en la siguiente iteración.
- 3 En algunas iteraciones, aunque rara vez en todas, algunos procesos de negocios pueden necesitar ser rediseñados para reflejar la integración probable de la aplicación del software en evolución.
- 4 En cada iteración del ciclo, *ALGUNOS PROTOTIPOS DE DISEÑO* o algunos elementos de *SISTEMA FUNCIONAL PARCIAL* se construyen y se prueban. Eventualmente, la aplicación completada resultará de la iteración final a través del ciclo.
- 5 Después de que se construye y se prueba cada prototipo o subsistema funcional, los usuarios del sistema reciben la oportunidad de experimentar el trabajar con ese prototipo. La expectativa es que los usuarios aclararán los requerimientos, identificarán otros nuevos y proporcionarán *RETROALIMENTACIÓN DE NEGOCIOS* acerca del diseño (por ejemplo, facilidad de aprendizaje, de uso) para la siguiente iteración a través del ciclo RAD.
- 6 Después de que se construye y se prueba cada prototipo o subsistema funcional, los analistas y diseñadores de sistemas revisarán la arquitectura de aplicación y el diseño para proporcionar *RETROALIMENTACIÓN TÉCNICA* y dirección para la siguiente iteración a través del ciclo RAD.
- 7 Con base en la retroalimentación, los analistas de sistemas identificarán *OBJETIVOS DE MEJORA DEL SISTEMA REFINADO* y/o *REQUERIMIENTOS DEL NEGOCIO*. Este análisis tiende a enfocarse en la revisión o expansión de objetivos y requerimientos y en identificar los asuntos de los usuarios con el diseño.
- 8 Con base en la retroalimentación, los analistas de sistemas y los diseñadores de sistemas identificarán una *ARQUITECTURA DE APLICACIÓN REFINADA* y/o *CAMBIOS DE DISEÑO*.
- 9 Eventualmente, el sistema (o una versión del mismo) se considerará digno de implantarse. Esta *VERSIÓN DEL SISTEMA FUNCIONAL* se prueba sistemáticamente y se pone en operación. La siguiente versión del sistema puede continuar la iteración a través del ciclo RAD.



**FIGURA 3.11** Estrategia de desarrollo rápido de aplicación (RAD)

**timeboxing** Imposición de un periodo no prorrogable, normalmente de 60 a 90 días, al término del cual debe entregarse para su funcionamiento la primera (o siguiente) versión de un sistema.

Aunque no es un requerimiento rígido para el RAD, la duración del ciclo de elaboración de prototipos puede ser limitada mediante una técnica llamada **timeboxing**. El **timeboxing** busca entregar un sistema operativo a los usuarios y a la administración en forma regular y constante. Los defensores del **timeboxing** alegan que el entusiasmo de la administración y el usuario por un proyecto puede ser mejorado y sostenido porque una nueva versión funcional del sistema se implanta constantemente.

El método RAD ofrece varias ventajas y desventajas:

#### Ventajas

- Es útil para proyectos en los que los requerimientos del usuario son inciertos o imprecisos.
- Alienta la participación activa de usuarios y administración (al contrario de una reacción pasiva a los modelos de sistemas no funcionales). Esto aumenta el entusiasmo del usuario final por el proyecto.
- Los proyectos tienen una mayor visibilidad y soporte debido a la extensa participación del usuario a lo largo del proceso.
- Los usuarios y la administración ven soluciones en funcionamiento, basadas en el software con mayor rapidez que en el desarrollo basado en modelos.
- Los errores y omisiones tienden a ser detectados más temprano en los prototipos que en los modelos de sistemas.
- Las pruebas y la capacitación son productos derivados naturales del método fundamental de elaboración de prototipos.
- El método iterativo es un proceso más natural debido a que el cambio es un factor esperado durante el desarrollo.

#### Desventajas

- Algunos afirman que el RAD alienta una mentalidad de “codificar, implantar y reparar” que incrementa los costos de la vida requeridos para operar, soportar y mantener el sistema.
- La elaboración de prototipos RAD puede fácilmente resolver los problemas equivocados ya que el análisis del problema se abrevia o se ignora.
- Un prototipo basado en RAD puede desalentar a los analistas de considerar otras alternativas con mayor valor técnico.
- Algunas veces es mejor lanzar un prototipo, pero los interesados a menudo están renuentes a hacerlo porque ven esto como una pérdida de tiempo y esfuerzo en el producto actual.
- El énfasis en la velocidad puede impactar en forma adversa la calidad debido a errores y omisiones.

RAD es muy popular para proyectos de tamaño pequeño a mediano. Demostraremos las técnicas de elaboración de prototipos y RAD en los capítulos apropiados en este libro.

### > Estrategia de implantación de paquetes de aplicación comercial

Algunas veces tiene más sentido comprar una solución de sistema de información que construir uno internamente. De hecho, cada vez más, muchas organizaciones esperan construir software sólo cuando hay una ventaja competitiva que obtener. Y para muchas aplicaciones centrales, tal como recursos humanos, financieros, compras, manufactura y distribución, existe poco valor competitivo en construir su propio sistema; por tanto se adquiere un **paquete de aplicación comercial**. En consecuencia, nuestra metodología *FAST* incluye una ruta de paquete de software comercial.

La solución comercial final es la planeación de recursos empresariales o ERP (definida en el capítulo 1). Las soluciones ERP proporcionan *todas* las aplicaciones de sistema de información centrales para un negocio completo. Para la mayoría de las organizaciones, una implantación ERP representa el proyecto de sistemas de información más grande alguna vez realizado por la organización. Puede costar decenas o cientos de millones de dólares y requerir un pequeño ejército de administradores, usuarios, analistas, especialistas técnicos, programadores, consultores y contratistas.

**paquete de aplicación comercial** Una aplicación de software que se puede comprar y personalizar (dentro de ciertos límites) para satisfacer las necesidades de negocios de numerosas organizaciones o de un ramo específico. Sinónimo de *sistema comercial listo para usarse (COTS)*.



La ruta de metodología *FAST* para una integración de paquete de aplicación comercial realmente no está dirigida a los proyectos ERP. De hecho, la mayoría de los proveedores ERP proporcionan su propia metodología de implantación (y socios consultores) para ayudar a sus clientes a implantar dicha solución de software masiva. En lugar de eso, nuestra metodología *FAST* proporciona una ruta para implantar todos los demás tipos de soluciones de sistema de información que pueden ser adquiridos por una empresa. Por ejemplo, una organización puede adquirir un paquete de aplicación comercial para una sola función de negocios como contabilidad, recursos humanos o compras. El paquete debe ser seleccionado, instalado, personalizado e integrado en el negocio y sus demás sistemas de información.

Las ideas básicas detrás de nuestra ruta de implantación de paquete de aplicación comercial son:

- Las soluciones de software comercial deben ser cuidadosamente seleccionadas para satisfacer las necesidades del negocio, “usted obtiene lo que pide y lo que paga”.
- Las soluciones de software comercial no sólo son costosas de adquirir sino que también pueden ser costosas de implantar. De hecho, el implantar un paquete comercial puede en realidad ser más costoso de implantar que un desarrollo interno.
- Los paquetes de software normalmente deben ser personalizados e integrados en el negocio. Además, los paquetes de software requieren el rediseño de los procesos de negocios existentes para adaptarse al software.
- Los paquetes de software rara vez satisfacen todos los requerimientos de negocios para la completa satisfacción de los usuarios. Por tanto, es necesario cierto nivel de desarrollo de sistemas internos con el fin de cumplir con los requerimientos no satisfechos.

La ruta de implantación de paquete de aplicación comercial se ilustra en la figura 3.12. Una vez más, el texto *italico* o *cursivo* y las flechas discontinuas indican diferencias del proceso básico *FAST*. Preste atención a las siguientes notas que corresponden a las viñetas numeradas:

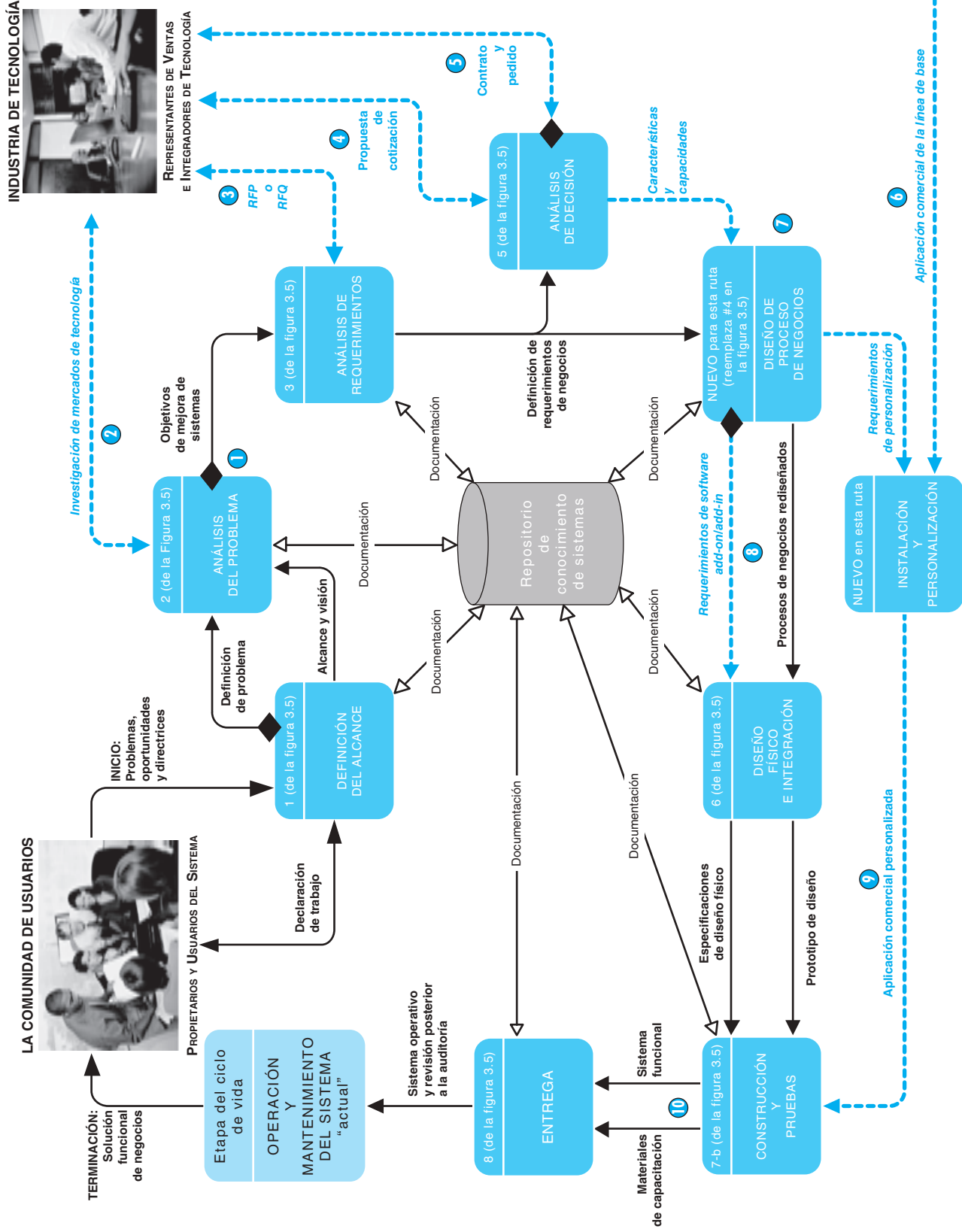
- 1 Debe señalarse que la decisión de comprar un paquete se determina en la fase de análisis de problema. El diamante negro representa la decisión “desarrollar o comprar”. El resto de este análisis asume que ha sido aprobada una decisión de compra.
- 2 El análisis del problema normalmente incluye cierta *INVESTIGACIÓN DE MERCADOS DE TECNOLOGÍA* para identificar qué soluciones de paquete existen, qué características hay en el software y qué criterio debe ser utilizado para evaluar dichos paquetes de aplicación. Esta investigación puede incluir proveedores de software, servicios de investigación TI (tales como el Gartner Group) o consultores.
- 3 Luego de definir los requerimientos de negocios, éstos deben ser comunicados a los proveedores de software que ofrecen soluciones de aplicación viables. Los requerimientos de negocios (y técnicos) son organizados y comunicados a los proveedores candidatos del software ya sea como **SOLICITUD DE PROPUESTA (REQUEST FOR PROPOSAL, RFP)** o **SOLICITUD DE COTIZACIÓN (REQUEST FOR QUOTATION, RFQ)**. La flecha de doble punta implica que puede necesitarse alguna aclaración de requerimientos y criterios.
- 4 Los proveedores envían *PROPUESTAS* o *COTIZACIONES* para sus soluciones de aplicación. Estas propuestas son evaluadas en contra de los requerimientos técnicos y de negocios especificados en RFP. La flecha de doble punta indica que las características y capacidades anunciadas deben ser validadas y en algunos casos aclaradas. Esto ocurre durante la fase de análisis de decisiones.
- 5 Se negocia un *CONTRATO Y PEDIDO* con el proveedor ganador del software, así como por los servicios necesarios para instalarlo y mantenerlo.
- 6 El proveedor proporciona el software y la documentación de *APLICACIÓN COMERCIAL DE LÍNEA*. Los servicios para instalación e implantación del software con frecuencia son proporcionados por el proveedor o sus proveedores de servicio (consultores certificados).
- 7 Cuando se adquiere un paquete de aplicación, la organización casi siempre debe cambiar sus procesos y prácticas de negocios para funcionar de manera eficaz con el paquete. La necesidad de *PROCESOS DE NEGOCIOS REDISEÑADOS* rara vez es recibida con gran entusiasmo, pero normalmente es necesario. En muchos casos, los cambios requeridos no están equivocados, sólo son distintos y poco familiares. Los usuarios de

---

**solicitud de propuesta (RFP)** Documento formal que comunica los requerimientos de negocios, técnicos y de soporte de un paquete de software de aplicación a los proveedores que desean competir por la venta de ese paquete de aplicación y los servicios correspondientes.

---

**solicitud de cotización (RFQ)** Documento formal que comunica los requerimientos de negocios, técnicos y de soporte de un paquete de software de aplicación a un solo proveedor, al que se ha considerado capaz de proporcionar el paquete de aplicación y servicios respectivos.



**FIGURA 3.12** Estrategia de implantación de paquetes de aplicación comercial

sistemas tienden a estar incómodos con cambiar la forma en que siempre han hecho algo.

- 8 Un paquete de aplicación rara vez cumple con todos los requerimientos de negocios en la instalación. Típicamente, se debe realizar un **análisis de diferencias** para determinar qué requerimientos de negocios no son satisfechos por las capacidades y características del paquete. Existen las siguientes opciones para los requerimientos que no son satisfechos:
- Solicitar una personalización del paquete dentro de los límites permisibles de acuerdo con lo especificado por el proveedor del software. La mayoría de los paquetes de aplicación comerciales permiten al comprador establecer opciones específicas, preferencias y valores definidos y rangos para ciertos parámetros. Dentro de ciertos límites, estas personalizaciones le permiten “adecuar” el paquete a la contabilidad del negocio y a las prácticas del mismo. Dichos REQUERIMIENTOS DE PERSONALIZACIÓN deben ser especificados.
  - Definir REQUERIMIENTOS DE SOFTWARE ADD-ON. Los requerimientos de software agregado especifican programas que deben ser diseñados y construidos para aumentar el paquete de aplicación comercial y entregar una funcionalidad adicional. Debe señalarse que los programas agregados llevan cierto riesgo de que pueda ser necesario modificarlos en el futuro cuando esté disponible una nueva versión del software. Pero este riesgo es nominal y la mayoría de las organizaciones lo toman con el fin de proporcionar una funcionalidad adicional.
  - Definir REQUERIMIENTOS DE SOFTWARE ADD-IN. ¡Los requerimientos de software add-in son muy peligrosos! Especifican cambios al paquete de aplicación comercial para satisfacer los requerimientos de negocios. En otras palabras, los usuarios solicitan que se hagan cambios al software adquirido, a su base de datos o a sus interfaces de usuario. En el mejor de los casos, los cambios pueden hacer que las mejoras de versión sean en extremo difíciles y exageradamente costosas. En el peor de los casos, dichos cambios pueden invalidar el soporte técnico del proveedor. (Y la mayoría de los proveedores alientan el mantener las versiones relativamente actuales al cancelar el soporte técnico en las versiones antiguas.) Debe desalentarse el cambiar el código del programa y las estructuras de bases de datos. La insistencia de los usuarios con frecuencia es un síntoma de la mala disposición para adaptar los procesos de negocios para que funcionen con la aplicación. Muchas organizaciones prohíben cambios a los paquetes de aplicación y obligan a los usuarios a adaptarse con el fin de preservar su proceso de mejora.
- 9 La APLICACIÓN COMERCIAL DE LÍNEA se instala y se prueba. Los cambios permitidos con base en opciones, preferencias y parámetros son completados y probados. Nota: Estas personalizaciones dentro de los límites especificados por el proveedor del software, generalmente llevarán a mejoras de la versión. En la mayoría de los casos, el proveedor ha considerado este nivel de APLICACIÓN COMERCIAL PERSONALIZADA.
- 10 Cualquier cambio de software add-on (o add-in) se diseña y construye para satisfacer los requerimientos de negocios adicionales. Después, el sistema es probado y puesto en operación por medio de las mismas actividades descritas en el proceso básico *FAST*.

La estrategia de paquete de aplicación comercial ofrece sus propias ventajas y desventajas:

#### Ventajas

- Los nuevos sistemas normalmente pueden ser implantados con mayor rapidez debido a que no se requiere una programación extensa.
- Muchas empresas no pueden permitirse el personal y la experiencia requerida para desarrollar soluciones internas.
- Los proveedores de aplicaciones distribuyen sus costos de desarrollo entre todos los clientes que adquieren su software. Por tanto, pueden inver-

#### Desventajas

- Una implantación COTS exitosa depende del éxito y viabilidad a largo plazo del proveedor de aplicación; si el proveedor queda fuera del negocio, usted pierde su soporte técnico y mejoras futuras.
- Un sistema adquirido rara vez refleja la solución ideal que el negocio podría lograr con un sistema desarrollado internamente, que se podría personalizar para expectativas precisas de la administración y los usuarios.

#### análisis de diferencias

Comparación de los requerimientos de negocios y técnicos de un paquete de aplicación comercial contra las características y capacidades de un paquete de aplicación comercial específico para definir los requerimientos que no se pueden satisfacer.

- tir en mejoras continuas en características, capacidades y funcionalidad que individualmente las empresas no siempre pueden pagar.
- El proveedor de aplicación asume la responsabilidad por mejoras de sistema significativas y correcciones de errores.
  - Muchas funciones de negocios son más similares que distintas en todas las empresas en una industria dada. Por ejemplo, las funciones de negocios a través de las organizaciones en la industria de cuidados médicos son más parecidas que diferentes. No tiene un buen sentido de negocios para cada organización el “reinventar la rueda”.
  - Casi siempre hay al menos cierta resistencia a cambiar los procesos de negocios para adaptarse al software. Algunos usuarios tendrán que ceder o asumir algunas responsabilidades. Y algunas personas pueden resentir que los cambios que perciben están basados en la tecnología en lugar de en los negocios.

Sin embargo, la tendencia hacia los paquetes de aplicación comercial adquiridos no puede ser ignorada. Actualmente, muchas empresas requieren que un paquete alternativo sea considerado antes de comenzar ningún tipo de proyecto de desarrollo interno. En este texto enseñaremos herramientas y técnicas de análisis de sistemas necesitados por los analistas de sistemas para funcionar en este ambiente.

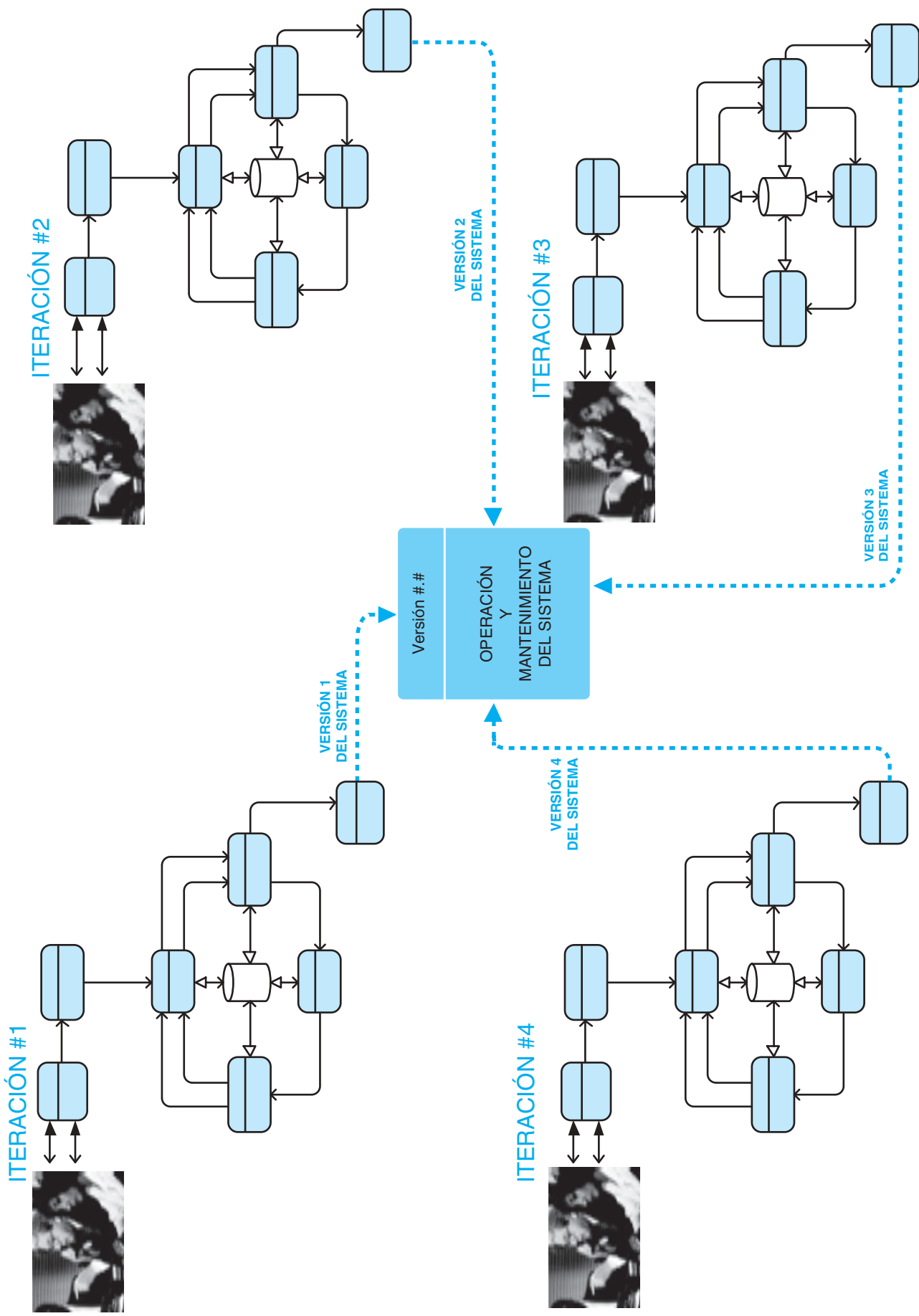
### > Estrategias híbridas

Las rutas *FAST* no son mutuamente exclusivas. Cualquier proyecto dado puede elegir o necesitar usar una combinación o variación de más de una ruta. La ruta que se utilizará siempre es seleccionada durante la fase de *definición de alcance* y es negociada como parte de la *declaración de trabajo*. Una estrategia que se aplica comúnmente a las rutas de desarrollo de aplicación rápida y basada en los modelos es una estrategia de incremento. En la figura 3.13 se ilustra una posible implantación de una estrategia de incremento en combinación con un desarrollo de aplicación rápido. El proyecto entrega el sistema de información en operación en cuatro etapas. Cada una implanta una versión del sistema final que utiliza una ruta RAD. Son posibles otras variaciones en las rutas.

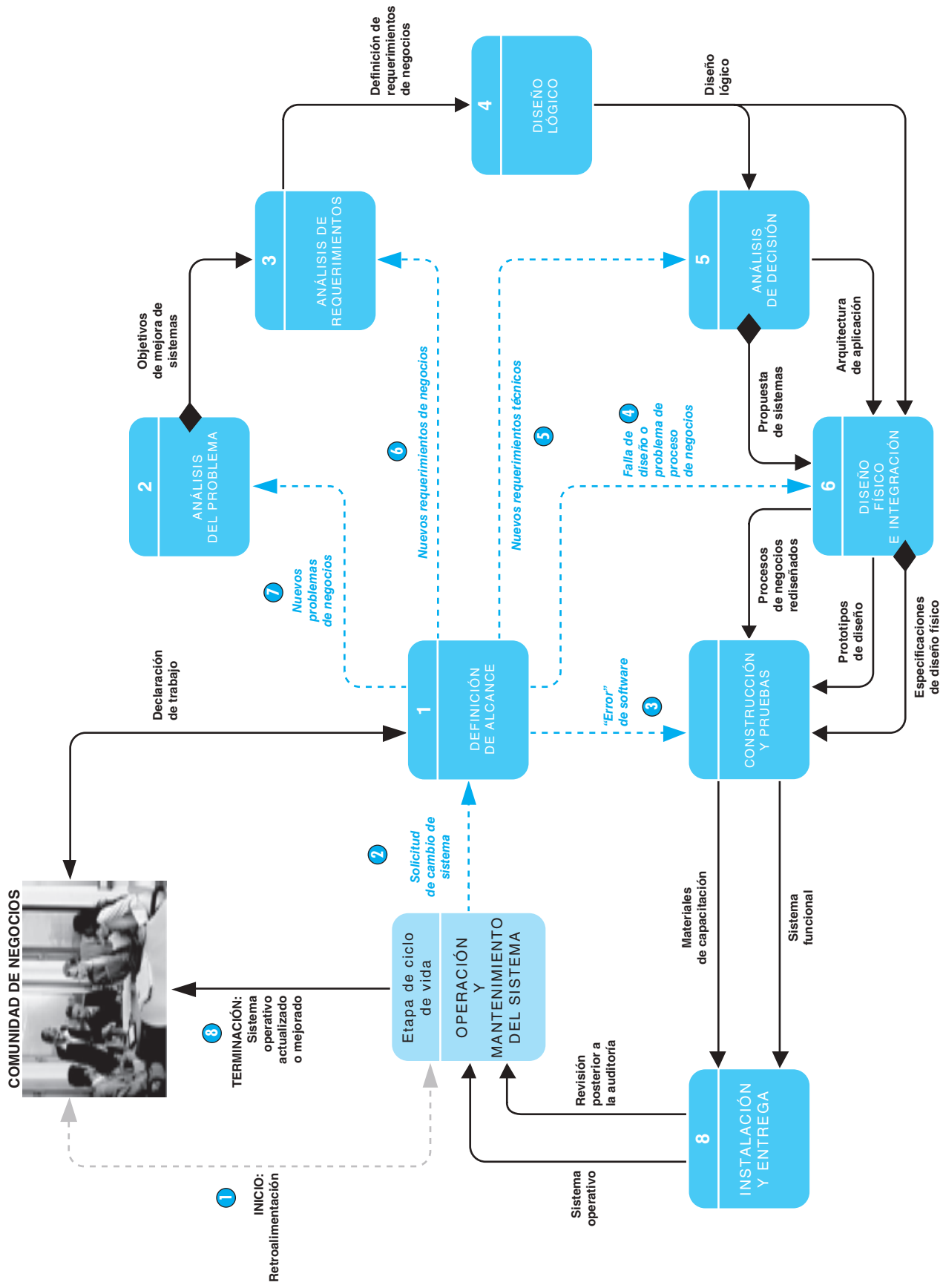
### > Mantenimiento del sistema

Todas las rutas finalmente resultan en la colocación de un nuevo sistema en operación. Con el mantenimiento de sistemas se pretende guiar los proyectos a través de la etapa de operación y soporte de su ciclo de vida, ¡que podría durar décadas! En la figura 3.14 se coloca un mantenimiento de sistemas en perspectiva. El mantenimiento de sistemas en *FAST* no es en realidad una ruta única. Como se ilustró en la figura, es apenas una versión de escala más pequeña del mismo proceso (o ruta) *FAST* que se utilizó para desarrollar originalmente el sistema. La figura demuestra que el punto de inicio para el mantenimiento de sistemas depende del problema que se quiere resolver. Preste atención a las siguientes viñetas numeradas en la figura:

- 1 Los proyectos de mantenimiento y reingeniería son disparados por alguna combinación de retroalimentación del usuario y técnica. Dicha retroalimentación puede identificar nuevos problemas, oportunidades o directrices.
- 2 El proyecto de mantenimiento se inicia por una SOLICITUD DE CAMBIO DE SISTEMA que indica los problemas, oportunidades o directrices.
- 3 Los arreglos más simples SON ERRORES DE SOFTWARE. Uno de dichos proyectos típicamente va directo hacia una fase de RECONSTRUCCIÓN Y REPETICIÓN DE PRUEBAS y se resuelve relativamente rápido.
- 4 A veces una FALLA DE DISEÑO en el sistema se vuelve aparente después de la implantación. Por ejemplo, los usuarios con frecuencia pueden cometer el mismo error debido a una pantalla de diseño confusa. Para este tipo de proyecto de mantenimiento, la fase de DISEÑO FÍSICO E INTEGRACIÓN necesitaría ser revisada de nuevo y desde luego seguida por las fases de construcción y entrega.



**FIGURA 3.13** Una estrategia de implantación incremental



**FIGURA 3.14** Perspectiva de mantenimiento de un sistema



En algunos casos un TEMA DE PROCESOS DE NEGOCIOS puede ser aparente. En este caso, sólo el proceso de negocios necesita ser rediseñado.

- 5 En alguna ocasión, NUEVOS REQUERIMIENTOS TÉCNICOS pueden dictar un cambio. Por ejemplo, una organización puede estandarizarse en la versión más nueva de un sistema de administración de bases de datos tal como el *SQL Server* u *Oracle*. Para este tipo de proyecto, la fase de ANÁLISIS DE DECISIÓN puede necesitar primero determinar el riesgo y la factibilidad de convertir la base de datos operacional existente a la nueva versión. Conforme sea apropiado, subsecuentemente dicho proyecto avanzaría a las fases de diseño físico, construcción y entrega como se necesite.
- 6 Los negocios cambian constantemente; por lo tanto, los requerimientos de negocios para un sistema también cambian. Uno de los disparadores más comunes para un proyecto de reingeniería es un NUEVO REQUERIMIENTO DE NEGOCIOS (o revisado). Dado el requerimiento, la fase de ANÁLISIS DE REQUERIMIENTOS debe ser revisada con un enfoque basado en el impacto del nuevo requerimiento del sistema existente. Con base en análisis de requerimientos, el proyecto entonces procedería a las fases de diseño lógico, análisis de decisión, diseño físico, construcción y entrega.  
¡Un momento! Ahora, usted se ha percatado de que los proyectos de mantenimiento y de reingeniería inician en distintas fases de la metodología básica. Podría estar preocupado de que repetir estas fases requeriría tiempo y esfuerzo excesivos. Sin embargo, tenga en mente que el alcance de los proyectos de mantenimiento y de reingeniería es mucho más pequeño que el proyecto original que creó el sistema de operación. Por tanto, el trabajo requerido en cada fase es mucho menor del que usted se imaginaria.
- 7 De nuevo, conforme cambian los negocios, se pueden encontrar PROBLEMAS DE NEGOCIOS NUEVOS, oportunidades y restricciones significativas. En este tipo de proyecto, el trabajo comienza con la fase de ANÁLISIS DEL PROBLEMA y luego sigue conforme sea necesario a las fases siguientes.
- 8 En todos los casos, el resultado final de cualquier tipo de proyecto de mantenimiento o reingeniería es un sistema de negocios operacionales actualizados que entrega el valor mejorado a los usuarios y propietarios del sistema. Las actualizaciones pueden incluir programas revisados, bases de datos, interfaces o procesos de negocios.

Como se describió anteriormente en el capítulo, esperamos que todos los sistemas eventualmente lleguen a una entropía. Los problemas de negocios o técnicos y los requerimientos se han vuelto tan problemáticos como para garantizar “empezar de nuevo”.

Eso completa nuestra introducción a la metodología de desarrollo de sistemas y sus rutas. Antes de que terminemos este capítulo, debemos presentar el papel de las herramientas automatizadas que soportan el desarrollo de sistemas.

## Herramientas y tecnología automatizada

Tal vez conozca la vieja fábula del zapatero cuyos niños no tenían zapatos. Esta situación no es distinta a la que enfrentan algunos desarrolladores de sistemas. Durante años hemos estado aplicando la tecnología de la información para resolver los problemas de negocios de nuestros usuarios; sin embargo, a veces hemos estado lentos para aplicar esa misma tecnología a nuestro propio problema, desarrollar sistemas de información. En el pasado no muy lejano, las herramientas principales del analista de sistemas eran papel, lápiz y una plantilla de diagrama de flujos.

Actualmente, conjuntos de herramientas automatizadas han sido desarrolladas, comercializadas e instaladas para ayudar a los desarrolladores de sistemas. Mientras que las metodologías de desarrollo de sistemas no siempre requieren herramientas automatizadas, la mayoría de las metodologías se benefician de dicha tecnología. Algunos de los beneficios más comúnmente citados incluyen:

- Productividad mejorada, a través de la automatización de tareas.
- Calidad mejorada, debido a que las herramientas automatizadas revisan la integridad, consistencia y contradicciones.
- Documentación mejor y más consistente, debido a que las herramientas facilitan el crear y ensamblar documentación consistente y de alta calidad.
- Mantenimiento de tiempo de vida reducido, debido a las mejoras de calidad del mencionado sistema combinado con una mejor documentación.

## HERRAMIENTAS CASE REPRESENTATIVAS

Erwin de Computer Associates

Designer 2000 de Oracle

System Architect Rational ROSE de Popkin

Visible Analyst de Visible Systems

### ingeniería de software asistida por computadora (CASE)

Uso de herramientas de software automatizadas que brindan soporte al dibujo y análisis de modelos de sistemas y a las especificaciones correspondientes. Algunas herramientas de CASE también posibilitan la elaboración de prototipos y generación de código.

**repositorio CASE** Base de datos de desarrolladores de sistemas donde pueden almacenar modelos de sistemas, descripciones y especificaciones detalladas, y otros productos del desarrollo de sistemas. Sus sinónimos abarcan *enciclopedia* y *diccionario de datos*.

### ingeniería hacia adelante

Una capacidad de herramienta CASE que permite generar código inicial del software o base de datos directamente a partir de modelos de sistemas.

**ingeniería inversa** Función de las herramientas CASE que permite generar automáticamente modelos de sistemas iniciales a partir de software o código de bases de datos.

## El contexto de proyectos de desarrollo de sistemas

- Las metodologías que en realidad funcionan a través de aplicación de reglas y experiencia construida internamente.

Es posible que su futuro empleador utilice o utilizará esta tecnología para desarrollar sistemas. Demostraremos el uso de diversas herramientas automatizadas a lo largo del presente texto. Hay tres clases de herramientas automatizadas para los desarrolladores:

- Elaboración de modelos de sistemas asistidos por computadora.
- Ambientes de desarrollo de aplicación.
- Administradores de proyecto y de proceso.

Analicemos brevemente cada una de estas clases de herramientas automatizadas.

### > Ingeniería de sistemas asistida por computadora

Los desarrolladores de sistemas han aspirado desde hace mucho a transformar el desarrollo de los sistemas de información y el software en disciplinas estilo ingeniería. Los términos *ingeniería de sistemas* e *ingeniería de software* están basados en una visión de que los sistemas y el desarrollo de software pueden y deben desempeñarse con precisión y rigor tipo ingeniería. Dicha precisión y rigor son consistentes con el método basado en los modelos para el desarrollo de sistemas. Para ayudar a que los analistas de sistemas desempeñen mejor la elaboración de modelos de sistemas, la industria desarrolló unas herramientas automatizadas llamadas de **ingeniería de software asistida por computadora (computer-assisted software engineering, CASE)**. Piense en la tecnología CASE como software que es utilizado para diseñar e implantar otro software. Esto es muy similar a la tecnología de diseño asistida por computadora (computer-aided design, CAD) utilizada por la mayoría de los ingenieros contemporáneos para diseñar productos como vehículos, estructuras, máquinas y demás. Los productos representativos de elaboración de modelos se listan al margen. La mayoría de los productos de modelado corren en las computadoras personales, como se muestra en la figura 3.15.

**Repositorios CASE** Al centro de cualquier arquitectura de herramientas CASE está una base de datos de un desarrollador llamada **repositorio CASE**. El concepto de repositorio fue presentado anteriormente (véase la figura 3.7).

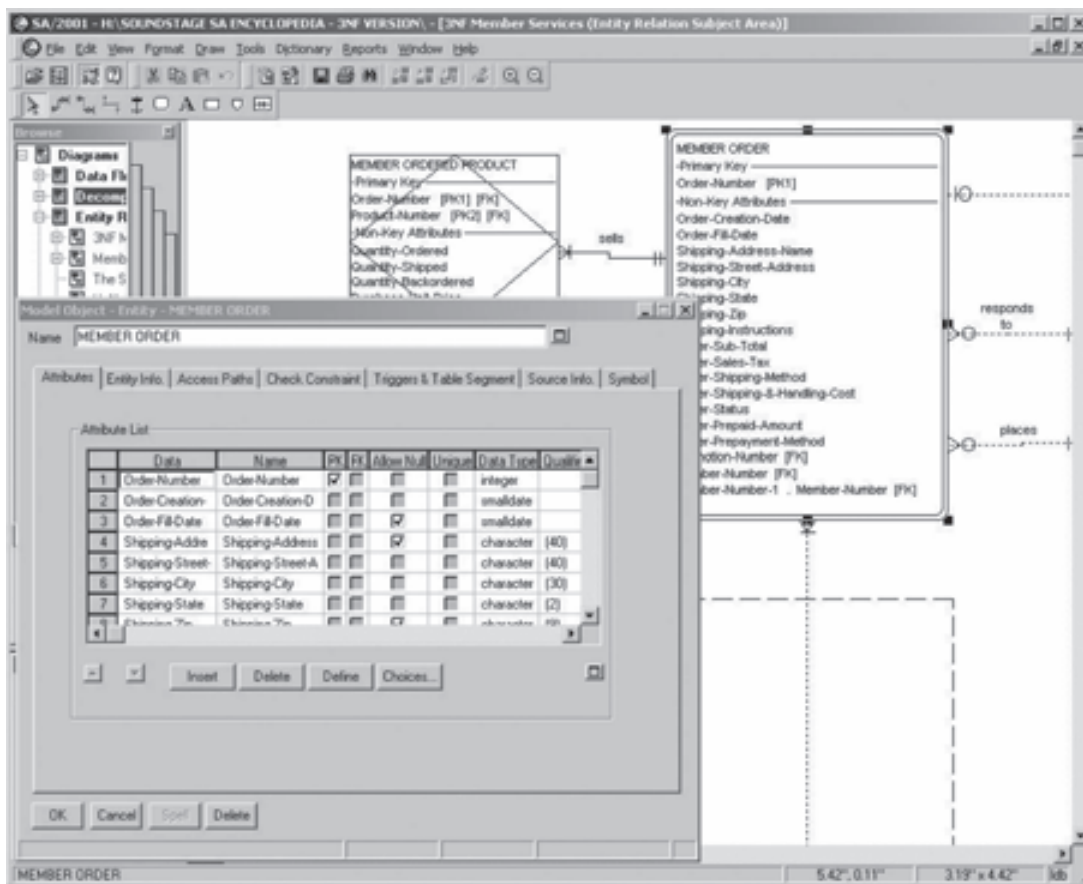
Alrededor del repositorio CASE está un grupo de herramientas o instalaciones para crear modelos de sistemas y la documentación.

**Ventajas de CASE** Para utilizar el repositorio, las herramientas CASE proporcionan alguna combinación de las siguientes ventajas, ilustradas en la figura 3.16:

- Las *herramientas de elaboración de diagramas* se utilizan para dibujar los modelos de sistemas requeridos o recomendados en la mayoría de las metodologías de desarrollo de sistemas. Normalmente, las formas en un modelo de sistemas pueden ser vinculadas a otros modelos de sistemas y a descripciones detalladas (véase el tema siguiente).
- Las *herramientas de diccionario* se utilizan para grabar, eliminar, editar y producir documentación y especificaciones detalladas. Estas descripciones pueden ser asociadas con formas que aparecen en los modelos de sistema que fueron dibujadas con las herramientas de elaboración de diagramas.
- Las *herramientas de diseño* pueden ser utilizadas para desarrollar muestras de los componentes de sistemas como entradas y salidas. Estas entradas y salidas pueden ser asociadas con los modelos de sistemas antes mencionados y con las descripciones.
- Con las *herramientas de administración de calidad* se analizan los modelos de sistemas, las descripciones y especificaciones y los diseños para integridad, consistencia y cumplimiento a las reglas y metodologías aceptadas.
- Las *herramientas de documentación* se utilizan para ensamblar, organizar e informar acerca de los modelos de sistemas, descripciones y especificaciones y prototipos que pueden ser revisados por los propietarios, usuarios, diseñadores y constructores de sistemas.
- Las *herramientas generadoras de diseño y código* automáticamente generan diseños de bases de datos y programas de aplicación o porciones significativas de esos programas.
- Las *herramientas de prueba* estimulan transacciones o tráfico de datos, miden el desempeño y proporcionan una administración de configuración de los planes de prueba y los guiones de prueba.

**Ingeniería hacia adelante e inversa** Como se afirmó antes, la tecnología CASE automatiza la elaboración de sistemas. Las herramientas CASE actuales proporcionan dos





**FIGURA 3.15** Captura de pantalla de la herramienta CASE *System Architect*

formas para desarrollar los modelos de sistemas; **ingeniería hacia adelante** e **ingeniería inversa**. Piense que la ingeniería inversa le permite generar un diagrama de flujo desde un programa existente y que la ingeniería hacia adelante le permite generar un programa directamente desde un diagrama de flujo. Las herramientas CASE que permiten una ingeniería bidireccional, hacia adelante e inversa se dice que proporcionan una “ingeniería de ida y vuelta”. Por ejemplo, usted aplica una ingeniería inversa a un sistema mal diseñado en un modelo de sistema, edita y mejora ese modelo y luego le aplica una ingeniería hacia adelante al nuevo modelo para un sistema mejorado.

### > Ambientes de desarrollo de aplicación

El énfasis en la velocidad y la calidad en el desarrollo del software ha resultado en los métodos RAD. El potencial para el RAD ha sido amplificado por la transformación de los compiladores de lenguaje de programación en **ambientes de desarrollo de aplicaciones (application development environments, ADE)**. Los ADE hacen que la programación sea más simple y más eficaz. De hecho, la mayoría de los compiladores de lenguaje de programación están ahora integrados en un ADE. Ejemplos de ADE (y los lenguajes de programación que respaldan, cuando aplican) se listan al margen.

Los ambientes de desarrollo de aplicación proporcionan diversas instalaciones de administración de productividad y calidad. El proveedor ADE proporciona algunas de estas instalaciones. Los proveedores terceros proporcionan muchas otras instalaciones que pueden integrar en el ADE.

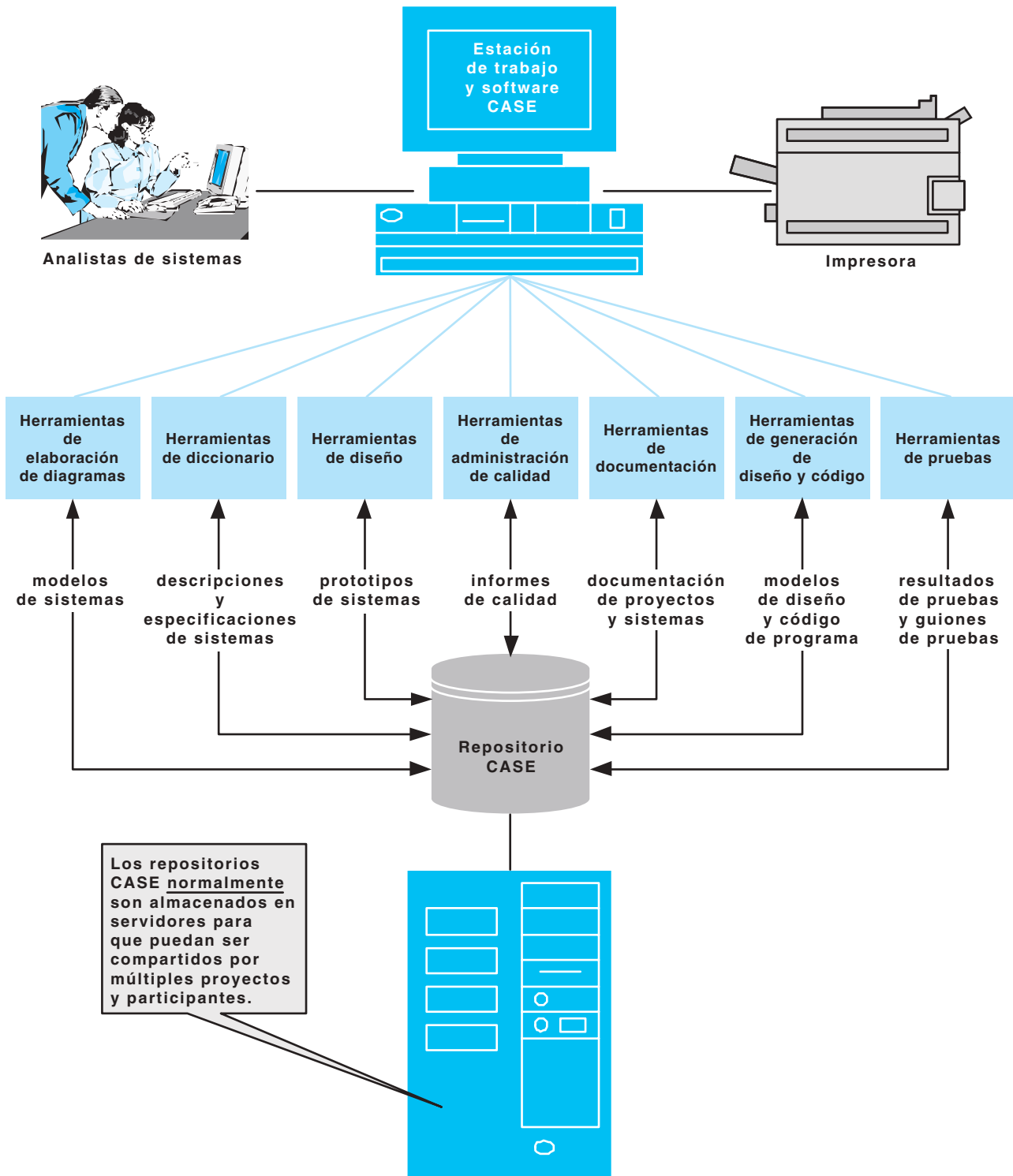
- Los *lenguajes de programación* o *intérpretes* están en el corazón de un ADE. Las poderosas características de limpieza de errores y de ayuda se proporcionan para ayudar a los programadores a identificar con rapidez y a resolver problemas de programación.
- Las *herramientas de construcción de interfaz* ayudan a los programadores a construir con rapidez las interfaces de usuario que utilizan una biblioteca de componente.

#### ADE REPRESENTATIVOS

*Websphere (Java)* IBM  
*J Builder (Java)* de Borland  
*Cold Fusion* de Macromedia  
*Visual Studio.NET (VB.NET, C++, C++.NET)* de Microsoft  
*Developer* de Oracle  
*Powerbuilder* de Sybase

#### ambiente de desarrollo de aplicaciones (ADE)

Herramienta integrada de desarrollo de software que proporciona todas las funciones necesarias para el desarrollo de nuevo software de aplicaciones con rapidez y calidad máximas. Un sinónimo muy usado es *ambiente de desarrollo integrado (IDE)*.



**FIGURA 3.16** Arquitectura de herramientas CASE

- El *middleware* es el software que ayuda a los programadores a integrar el software que se desarrolla con las diversas bases de datos y redes de cómputo.
- Las *herramientas de pruebas* se utilizan para construir y ejecutar los guiones de pruebas que pueden probar el software consistente y profundamente.
- Las *herramientas de control de versión* ayudan a múltiples equipos de programadores a manejar múltiples versiones de un programa, tanto durante el desarrollo como después de su implantación.
- Las *herramientas de la autoría de ayuda* se utilizan para escribir sistemas de ayuda en línea, manuales del usuario y capacitación en línea.
- Los *vínculos del repositorio* permiten que los productos de herramientas ADE se integren con los de CASE al igual que con otras ADE y herramientas de desarrollo.

## > Administradores de proceso y proyecto

Una tercera clase de herramientas automatizadas nos ayuda a manejar la metodología y los proyectos de desarrollo de sistemas que utilizan esa metodología. Mientras que las herramientas CASE y los ADE fueron hechos para soportar el análisis, diseño y construcción de los nuevos sistemas de información y software, las herramientas de **aplicación de administrador de proceso y aplicación de administrador de proyecto** fueron hechas para soportar las actividades transversales del ciclo de vida. El *Project* de Microsoft y *Open Workbench* y *Project Manager* de Niku son ejemplos de herramientas de administración de proyectos automatizadas.

### aplicación de administración de proceso

Herramienta automatizada que ayuda a documentar y gestionar un método y sus rutas, productos y normas de administración de la calidad. Un sinónimo emergente es *methodware*.

### aplicación de administración de proyecto

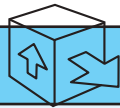
Herramienta automatizada que ayuda a planear las actividades de desarrollo de sistemas (de preferencia, con el uso de métodos aprobados), estimar y asignar.

Este capítulo, junto con los primeros dos, completa el contexto mínimo para estudiar el análisis y diseño de sistemas. Hemos descrito ese contexto en términos de las tres pes, los participantes (los interesados del capítulo 1), el producto (el sistema de información del capítulo 2) y el proceso (el desarrollo del sistema; capítulo 3). Armado con esta comprensión, usted está ahora listo para estudiar el análisis de sistemas o métodos de diseño.

La próxima tarea del lector dependerá de cuál sea su meta:

- Una revisión completa del desarrollo de sistemas. Recomendamos que siga su camino secuencial al capítulo 4, “Análisis de sistemas”. En ese capítulo estudiará con mayor profundidad la definición de alcance, el análisis de problemas, el análisis de requerimientos y las fases de diseño lógico que fueron presentadas en el capítulo 3.
- El estudio de las técnicas de análisis de sistemas. De nuevo, le recomendamos continuar el camino secuencial al capítulo 4, “Análisis de sistemas”. En dicho capítulo se presenta un contexto para estudiar posteriormente las herramientas y técnicas del análisis de sistemas.
- Estudio de técnicas de diseño de sistemas. Usted todavía podría dar un repaso rápido al capítulo 4, “Análisis de sistemas”, para tener el contexto. Luego continúe su estudio detallado en el capítulo 10, “Diseño de sistemas”. En ese capítulo usted aprenderá más acerca de los procesos y las estrategias para el diseño y la construcción de los sistemas de información.

## Resumen

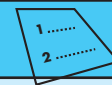


1. Un proceso de desarrollo de sistemas es un conjunto de actividades, métodos, mejores prácticas, productos y herramientas automatizadas que los interesados utilizan para desarrollar y mejorar continuamente los sistemas de información y el software.
2. El Modelo de Madurez de la Capacidad (CMM) es un marco de referencia para evaluar el nivel de madurez del desarrollo de sistemas de información y la administración de procesos y productos de una organización. Define la necesidad de un proceso de desarrollo de sistemas.
3. Un ciclo de vida de sistemas divide la vida de un sistema de información en dos etapas, *desarrollo de sistemas, operación y mantenimiento de sistemas*.
4. Una metodología de desarrollo de sistemas es un proceso de la etapa de desarrollo de sistemas. Define un conjunto de actividades, métodos, mejores prácticas, productos y herramientas automatizadas que los desarrolladores de sistemas y los administradores de proyecto utilizarán para desarrollar y mantener los sistemas de información y software.
5. Los siguientes principios deben fundamentar todas las metodologías de desarrollo de sistemas:
  - a) Hacer que los usuarios del sistema participen.
  - b) Utilizar un método de solución de problemas.
  - c) Establecer fases y actividades.
  - d) Documentar a través del desarrollo.
  - e) Establecer estándares.
  - f) Administrar el proceso y los proyectos.
  - g) Justificar los sistemas de información como inversiones de capital.
  - h) No temer cancelar o revisar el alcance.
  - i) Dividir y conquistar.
  - j) Diseñar sistemas para crecimiento y cambio.
6. Los proyectos de desarrollo de sistemas son disparados por problemas, oportunidades y directrices:
  - a) Los problemas son situaciones indeseables que evitan que la organización logre completamente su propósito, metas u objetivos.
  - b) Las oportunidades sirven para mejorar la organización incluso en ausencia de problemas específicos.
  - c) Las directrices son requerimientos nuevos impuestos por la administración, el gobierno o alguna influencia externa.
7. El marco de referencia PIECES de Whetherbe es útil para clasificar problemas, oportunidades y directrices. Las letras de PIECES corresponden en inglés a Performance (desempeño), Information (información), Economics (economía), Control (control), Efficiency (eficiencia) y Service (servicio).
8. Las fases básicas del desarrollo de sistemas incluyen:
  - a) Definición del alcance.
  - b) Análisis del problema.
  - c) Análisis de requerimientos.
  - d) Diseño lógico.
  - e) Análisis de decisión.
  - f) Diseño físico e integración.
  - g) Construcción y pruebas.
  - h) Instalación y entregas.
9. Las actividades transversales del ciclo de vida son aquellas que se traslapan en muchas o en todas las actividades de la metodología. Pueden incluir:
  - a) Definición de hechos, el proceso formal de utilizar investigación, entrevistas, juntas, cuestionarios, muestreos y otras técnicas para recolectar información acerca de los sistemas, requerimientos y preferencias.
  - b) Documentación, la actividad de registrar hechos y especificaciones de un sistema para referencia actual y futura. La documentación se almacena con frecuencia en un repositorio, una base de datos donde los desarrolladores de sistemas almacenan toda la documentación, el conocimiento y los productos para uno o más sistemas de información o proyectos.
  - c) Presentación, la actividad de comunicar resultados, recomendaciones y documentación para revisión por parte de los usuarios y administradores interesados. Las presentaciones pueden ser verbales o escritas.
  - d) El análisis de factibilidad, la actividad por la cual se mide y se evalúa la factibilidad, una medida de qué tan benéfico sería el desarrollo de un sistema de información para una organización.
  - e) Administración de procesos, la actividad continua que documenta y mejora el uso de la metodología elegida por una organización ("el proceso") para el desarrollo de sistemas.
  - f) La administración de proyecto, la actividad de definir, planear, dirigir, vigilar y controlar un proyecto para desarrollar un sistema aceptable dentro del tiempo y presupuesto asignados.
10. Hay diferentes rutas a través de las fases de desarrollo de sistemas básicos. Se selecciona una ruta apropiada durante la fase de definición de alcance. Las rutas típicas incluyen:
  - a) Estrategias de desarrollo basadas en modelos, que enfatizan el dibujo de diagramas para ayudar a visualizar y analizar problemas, definir requerimientos del negocio y diseñar sistemas de información. Las estrategias alternativas basadas en modelos incluyen:

- i)* Modelado de procesos
  - ii)* Modelado de datos
  - iii)* Elaboración de modelos de objetos
- b)* Las estrategias de desarrollo rápido de aplicaciones (RAD), que enfatizan la participación extensa de los usuarios en la construcción rápida y evolutiva de prototipos en funcionamiento de un sistema para acelerar el proceso de desarrollo del sistema.
  - c)* Las estrategias de implantación de paquete de aplicación comercial, que se enfocan en la compra e integración de un paquete o solución de software para soportar una o más funciones de negocios y sistemas de información.
  - d)* El mantenimiento del sistema ocurre luego de que se implanta un sistema y se realiza durante toda la vida del sistema. Esencialmente, el mantenimiento del sistema ejecuta una versión de menor escala del proceso de desarrollo con distintos puntos de inicio según el tipo de problema que se va a resolver.
11. Las herramientas automatizadas soportan todas las fases de desarrollo de sistemas:
- a)* Las herramientas de ingeniería de sistemas asistida por computadora (CASE) son programas de software que automatizan o soportan el dibujo y análisis de modelos de sistemas y permiten la transición de los modelos de sistemas a programas de aplicación.
    - i)* Un repositorio CASE es una base de datos de los desarrolladores de sistemas. Es un

lugar donde los desarrolladores pueden almacenar modelos de sistemas, descripciones y especificaciones detalladas y otros productos del desarrollo de sistemas.

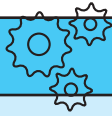
- ii)* La ingeniería hacia adelante requiere que el analista de sistemas dibuje modelos de sistemas, ya sea desde el inicio o con plantillas. Los modelos resultantes son transformados posteriormente en un código de programa.
  - iii)* La ingeniería inversa permite que una herramienta CASE lea un código de programa existente y transforme ese código en un modelo de sistema representativo que puede ser editado y refinado por el analista de sistemas.
- b)* Los ambientes de desarrollo de aplicación (ADE) son herramientas de desarrollo de software integrado que proporcionan todas las facilidades necesarias para desarrollar un software de aplicación nuevo con la máxima velocidad y calidad.
  - c)* Las herramientas de administración de procesos ayudan a documentar y manejar una metodología y rutas, sus productos y los estándares de administración de calidad.
  - d)* Las herramientas de administración de proyectos ayudan a planear actividades de desarrollo de sistemas (preferentemente por medio de una metodología aprobada), estimar y asignar recursos (incluidas las personas y los costos), programar actividades y recursos, vigilar el progreso en contra del programa y el presupuesto, controlar y modificar el programa y los recursos y reportar el progreso del proyecto.



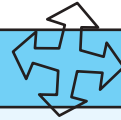
## Preguntas de repaso

1. Explique por qué es importante tener un proceso de desarrollo de sistemas estandarizado para una organización.
2. ¿Cómo se relacionan el ciclo de vida del sistema y la metodología de desarrollo de sistemas?
3. ¿Cuáles son los 10 principios fundamentales para el desarrollo de sistemas?
4. ¿Por qué es importante la documentación a lo largo del proceso de desarrollo?
5. ¿Por qué son necesarios la administración de proceso y la administración de proyecto?
6. ¿Qué es administración de riesgos? ¿Por qué es necesaria?
7. ¿Qué interesados inician la mayoría de los proyectos? ¿Cuál es el ímpetu para la mayoría de los proyectos?
8. ¿Quiénes son los principales participantes en la definición de alcance? ¿Cuáles son sus metas en la definición de alcance?
9. ¿Cuáles son los tres productos más importantes en la definición de alcance?
10. ¿Quiénes son los principales participantes en la fase de análisis de requerimientos? ¿Por qué son ellos los participantes principales?
11. ¿Qué análisis de factibilidad se realiza en el análisis de decisión?
12. ¿Qué es el desarrollo basado en modelos?
13. ¿Por qué es popular el desarrollo basado en modelos?
14. ¿Qué es el desarrollo de aplicación rápida (RAD)?
15. ¿Qué beneficios puede traer el RAD al proceso de desarrollo de sistemas?
16. ¿Qué es la ingeniería de software asistida por computadora (CASE)? Liste algunos ejemplos de CASE.

## Problemas y ejercicios



1. El Modelo de Madurez de la Capacidad (CMM) fue desarrollado por el Software Engineering Institute en Carnegie Mellon y es ampliamente utilizado por los sectores público y privado. ¿Cuál es el propósito del marco de referencia CMM y cómo lo logra?
2. Liste los cinco niveles de madurez y describa brevemente cada uno de ellos.
3. En la tabla 3.1 en este libro se ilustra la diferencia en la duración de un proyecto típico, persona-meses, calidad y costo, lo que depende de si el proceso de desarrollo de sistema de una organización está al nivel 1, 2 o 3 de CMM. ¿Entre cuáles dos niveles de CMM una organización obtiene el mayor beneficio en términos de porcentaje de mejora? ¿Cuál cree usted que es la razón para esto?
4. La *metodología de desarrollo de sistemas* y el *ciclo de vida de sistemas* son dos términos que se utilizan muy a menudo y con la misma frecuencia son mal utilizados. ¿Cuál es la diferencia entre los dos términos?
5. Describa cómo el usar una metodología de desarrollo de sistemas está en línea con las metas CMM y puede ayudar a una organización a aumentar su nivel de madurez.
6. Algunos principios fundamentales son comunes a todas las metodologías de desarrollo de sistemas. Identifique esos principios fundamentales y explíquelos.
7. El marco de referencia PIECES fue desarrollado por James Wetherbe como medio para clasificar problemas. Identifique las categorías, luego clasifique los siguientes problemas por medio del marco de referencia PIECES:
  - a) Los datos duplicados son almacenados en todo el sistema.
  - b) Hay necesidad de hacer un puerto para una aplicación existente con los aparados de PDA.
  - c) Los informes de ventas trimestrales necesitan generarse automáticamente.
  - d) Los empleados pueden obtener acceso a porciones confidenciales del sistema de personal.
  - e) Las interfaces de usuario del sistema de inventarios son difíciles y confusas, lo que resulta en una alta frecuencia de pedidos incorrectos.
8. Cada fase de un proyecto incluye productos específicos que deben ser producidos y entregados en la siguiente fase. Con la metodología *FAST* hipotética del libro de texto, ¿cuáles son los productos para las fases de integración de análisis de requerimientos, diseño lógico y diseño físico?
9. La definición de alcance es la primera fase de la metodología *FAST* y es ya sea la primera fase o parte de la primera fase de la mayoría de las metodologías. ¿Qué dispara la fase de alcance? ¿Qué interesados participan en esta fase? ¿Cuáles dos preguntas esenciales necesitan ser respondidas?, y ¿cuáles son los tres productos importantes que salen de esta fase?
10. La fase de análisis de requerimientos es una parte esencial de una metodología de desarrollo de sistema. De acuerdo con la metodología *FAST*, ¿qué interesados participan típicamente en esta fase? ¿Cuál es el enfoque principal del análisis de requerimientos? ¿Cuál *no es* el enfoque? ¿Por qué cada requerimiento propuesto debe ser evaluado? ¿Qué error crítico debe ser evitado?
11. En la metodología *FAST*, así como en la mayoría de las metodologías de sistemas, los propietarios de sistemas y los diseñadores de sistemas no participan en la fase de análisis de requerimientos. ¿Cuál cree usted que sea la razón para esto?
12. ¿Cuál es el propósito esencial de la fase de diseño lógico? ¿Cómo lo cumple? ¿Cómo se incorporan las soluciones tecnológicas en esta fase? ¿Cuáles son algunos sinónimos comunes para esta fase utilizada por otras metodologías? ¿Quiénes son los participantes típicos en esta fase? ¿Qué es el modelado ágil y cuál es su propósito? ¿Cuáles son los productos que salen de esta fase? En términos del equipo de desarrollo, ¿qué transición crítica se realiza para el final de esta fase?
13. ¿Cuál es el propósito esencial de la fase de diseño físico? ¿Quién debe participar en esta fase y quién puede participar? ¿Cuáles son dos filosofías de diseño físico en los distintos extremos del continuo y en qué son diferentes? ¿Es esta una fase probable en la que un proyecto puede ser cancelado? ¿Con qué otra fase es posible que exista un traslape y cuál cree usted que sea la razón para esto?
14. Un cliente ha contratado a su compañía de desarrollo de software para desarrollar un nuevo sistema de procesamiento de pedidos. Sin embargo, los marcos de tiempo son muy cortos y son inflexibles para la entrega de al menos la parte básica del nuevo sistema. Es más, los requerimientos del usuario no están bien definidos ni son claros. ¿Cuáles son dos estrategias de desarrollo de sistemas que podría ser provechoso utilizar en este trabajo?
15. ¿Cuál es la desventaja potencial de utilizar las estrategias descritas en la pregunta anterior?



## Proyectos e investigación

1. El Software Engineering Institute (SEI) en Carnegie Mellon University ha desarrollado una serie de Modelos de Madurez de la Capacidad (CMM) relacionados. Usted puede leer acerca de estos distintos productos de CMM en el sitio Web de SEI (<http://www.sei.cmu.edu>).
  - a) Identificar los productos actuales CMM que son mantenidos o desarrollados por el SEI.
  - b) ¿Cuáles son sus diferencias y similitudes?
  - c) Si usted fuera a evaluar a su organización o a una organización con la que usted esté familiarizado, por medio del CMM descrito en el texto, ¿en qué nivel sería? ¿Por qué?
  - d) ¿Qué pasos recomendaría usted que su organización tomara con el fin de avanzar al siguiente nivel de CMM?
  - e) ¿Siente usted que el tiempo, el costo y los recursos para avanzar al siguiente nivel valdrían los beneficios percibidos de su organización? ¿Por qué sí o por qué no?
2. Usted es un nuevo administrador del proyecto y ha recibido la asignación de la responsabilidad de un proyecto de sistemas de información empresarial que toca cada división de su organización. El presidente y director ejecutivo afirmó al inicio del proyecto que era la prioridad número uno de su organización. El proyecto está en medio de la fase de análisis de requerimientos. Aunque van de acuerdo con el programa, usted nota que la asistencia de los usuarios y propietarios del sistema a las juntas acerca de los requerimientos ha disminuido. Un administrador de proyectos más experimentado le ha dicho que no se preocupe, que esto es normal. ¿Debe usted preocuparse?
3. Hay muchas metodologías de desarrollo de sistemas distintas en uso, cada una con su propia terminología y número y alcance de fases. Busque en la Web para la información de dos o tres otras metodologías de desarrollo de sistemas y luego haga lo siguiente:
  - a) Revise las metodologías de desarrollo de sistemas que usted encontró. ¿Cuándo, por quién y por qué fueron desarrolladas?
  - b) ¿Qué fases y terminología emplean?
  - c) Dibuje una matriz y compare sus fases con la metodología *FAST* del texto.
  - d) ¿Ve usted alguna ventaja o desventaja en cualquiera de las metodologías que encontró en comparación con la metodología *FAST*?
4. El marco de referencia de *PIECES*, que fue desarrollado por James Wetherbe y se describe en el texto, está hecho con la intención de ser un marco de referencia para clasificar problemas, oportunidades y directrices.
 

Contacte a uno de los analistas de sistemas de su organización, su escuela y otra organización. Pregunte acerca de los sistemas de información utilizados en su organización y que describan cuáles son los problemas en términos generales. Seleccione tres de estos sistemas:

  - a) Describa los sistemas que usted eligió, sus problemas y las organizaciones que los utilizan.
  - b) Use el marco de referencia *PIECES* en el libro de texto para clasificar los problemas de cada sistema.
  - c) Describa la categoría o categorías de *PIECES* que usted encontró para cada problema. ¿Cada problema generalmente tenía una o más categorías asociadas con él?
  - d) Para los sistemas utilizados por distintas organizaciones, ¿qué cosa en común encontró en las clasificaciones de problemas? Si usted encontró una gran cantidad de cosas comunes en las categorías, ¿cree usted que esto es significativo o sólo coincidencia?
  - e) En el ciclo de vida del desarrollo de un sistema, ¿dónde cree usted que un marco de referencia *PIECES* sería de mayor valor?
5. Las herramientas de ingeniería de software asistida por computadora (CASE) pueden ayudar significativamente a los desarrolladores a mejorar la productividad, la calidad y la documentación. Realice una encuesta informal acerca de una docena de departamentos de tecnología de información en relación con si utilizan herramientas CASE y si es así, de qué tipo. También, averigüe por cuánto tiempo han estado utilizando las herramientas CASE y si son utilizadas para todos los proyectos o sólo para algunos. Agregue cualquier otra pregunta que le pueda parecer significativa. Trate de dividir su investigación entre agencias del sector público y del sector privado y organizaciones grandes y pequeñas.
  - a) ¿Qué tipo de organizaciones investigó?
  - b) ¿Qué preguntó?
  - c) ¿Cuáles fueron los resultados?
  - d) Dada la naturaleza limitada e informal de esta investigación, ¿pudieron encontrar algún patrón o tendencia?
  - e) Con base en sus lecturas y en su investigación, ¿cuáles son sus sentimientos en relación con el uso de las herramientas CASE?
6. Los proyectos en ocasiones son cancelados o abandonados, algunas veces por elección, algunas veces no. Busque artículos en la Web acerca de estrategias de abandono de proyectos y seleccione dos de ellas.

- a) ¿Qué artículo seleccionó usted?
- b) ¿Cuáles son sus temas centrales, resultados y recomendaciones en relación con el abandono del proyecto?
- c) Compare y contraste sus resultados y sus recomendaciones. ¿Qué estrategia elegiría, si fuera el caso?
- d) ¿Cree usted que abandonar un proyecto es siempre evitable o siempre representa un fracaso?

## Casos breves



1. Entreviste al menos a dos administradores de proyecto. ¿Cuáles son sus experiencias con el *acortamiento del alcance*?
2. George es el presidente y director ejecutivo de una importante corporación que ha tratado de desarrollar un programa que capture los golpes del teclado de todos los empleados en sus computadoras. El proyecto está actualmente excedido en 100 000 dólares por encima del presupuesto y detrás del programa y requerirá al menos otros 50 000 dólares y seis meses para concluirse. El presidente y director ejecutivo quiere continuar con el proyecto debido a que ya se ha invertido mucho en él. ¿Cuál sería su recomendación y por qué?
3. Beatriz es una administradora excelente, es muy capaz de administrar el proceso burocrático y de seguir las reglas de negocios en su corporación. Es una persona “que sigue las reglas” en la que siempre se puede contar para hacer las cosas “bien”. ¿Beatriz sería una buena *administradora del proyecto*? ¿Por qué sí o por qué no?
4. Una compañía está tratando de decidir entre utilizar un programa listo para usarse o desarrollar uno personalizado para la administración de inventarios. El producto listo para usarse es menos costoso que la solución personalizada y aún así tiene la mayor parte de la funcionalidad requerida. El presidente cree que la capacidad que falta puede ser abordada al modificar el programa una vez que lo hayan adquirido. Como el director ejecutivo de información de la empresa, ¿cuáles son sus preocupaciones y sus recomendaciones para el presidente?

## Ejercicios de equipo e individuales



1. (Equipo) Realice juntas de equipo por medio de distintos medios de comunicación. Por ejemplo: teléfono, correo electrónico, ambiente virtual. ¿Qué notó acerca del impacto de la tecnología de la junta? ¿Fue la productividad de las juntas la misma con cada medio? ¿Cómo se sintió el equipo acerca del impacto de las tecnologías en las relaciones de equipo?
2. (Individual) El análisis y diseño de un sistema de información, cuando está bien hecho, a menudo elimina puestos en una compañía. La teoría económica respalda fuertemente la creación de nuevos empleos cuando esto sucede, pero en general hay un retraso entre la pérdida estructural de los puestos y la creación de nuevos puestos. ¿Cómo se siente acerca de eso?
3. (Equipo o individual) Visite una unidad de cuidado intensivo neonatal en un centro médico muy reconocido. Escriba un ensayo breve acerca de sus pensamientos de la tecnología involucrada y el impacto que tiene acerca de la tecnología que participa y el impacto que tiene en las vidas de la gente. Comparta con la clase a criterio del profesor.

## Lecturas recomendadas



Ambler, Scott. *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process*. Nueva York: John Wiley & Sons, 2002. Éste es el libro definitivo acerca de los métodos ágiles y el modelado.

*Application Development Trends* (periódico mensual). Framingham, MA: Software Productivity Group, Inc. Éste es nuestro periódico favorito para mantenerse actualizado con las últimas tendencias en la metodología y herramientas automatizadas. Cada mes presenta diversos artículos acerca de diferentes temas y productos.

DeMarco, Tom, *Structured Analysis and System Specification*, Englewood Cliffs, NJ, Prentice Hall, 1978. Éste es el libro

clásico acerca de análisis de sistemas estructurados, una metodología centrada en los procesos y basada en modelos.

Gane, Chris, *Rapid Systems Development*, Englewood Cliffs, NJ Prentice Hall, 1998. Este libro presenta una buena visión general de RAD que combina un desarrollo basado en modelos y la elaboración de prototipos en el equilibrio correcto.

Gildersleeve, Thomas, *Successful Data Processing Systems Analysis*, 2a. ed., Englewood Cliffs, NJ, Prentice Hall, 1985. Estamos en deuda con Gildersleeve por el método del compromiso ajustado. Los clásicos nunca se vuelven obsoletos.

Jacobson, Ivar; Grady Booch y James Rumbaugh, *The Unified Software Development Process*, Reading, MA, Addison-Wesley,



1999. El Rational Unified Process actualmente es un ejemplo popular de una metodología orientada a objetos y basada en los modelos.

McConnell, Steve, *Rapid Development*. Redmond, WA, Microsoft Press, 1996. En el capítulo 7 de este excelente libro de referencia se proporciona lo que podría ser el resumen definitivo del ciclo de vida de desarrollo de un sistema y las variaciones de metodología que llamamos "rutas" en nuestro libro.

Orr, Ken, *The One Minute Methodology*, Nueva York, Dorsett House, 1990. Lectura obligada para los interesados en explorar la necesidad de una metodología. Este libro muy pequeño puede ser leído en una noche. Sigue la historia satírica de la

búsqueda de un analista de la bala de plata del desarrollo, "la metodología de un minuto".

Paulk, Mark C; Charles V. Weber; Bill Curtis y Mary Beth Chrissis, *The Capability Maturity Model: Guidelines for Improving the Software Process*, Reading, MA, Addison-Wesley, 1995. Este libro describe completamente la versión 1.1 del Modelo de Madurez de Capacidad. Nótese que la versión 2.0 estaba en desarrollo al momento de escribir este capítulo.

Wetherbe, James. *Systems Analysis and Design: Best Practices*, 4a. ed. St. Paul, MN, West, 1994. Estamos en deuda con Wetherbe por el marco de referencia PIECES.





## Métodos de análisis de sistemas

En los seis capítulos de la parte dos le presentamos las actividades y métodos de análisis de sistemas. En el capítulo 4, “Análisis de sistemas”, se presenta el contexto para todos los capítulos subsecuentes al introducir las actividades del *análisis de sistemas*. Este análisis es la fase más crítica de un proyecto. Durante el análisis de sistemas aprendemos acerca del sistema de negocios existente, tratamos de entender sus problemas, definimos los objetivos de mejora y detallamos los requerimientos de negocio que deben ser satisfechos por *cualquier* solución técnica posterior. Es claro que cualquier diseño e implementación del nuevo sistema depende de la calidad del análisis de sistema precedente. El análisis de sistemas a menudo se abrevia en un proyecto debido a que: 1) muchos analistas no son hábiles en los conceptos y técnicas de modelado lógico que se deben utilizar, y 2) muchos analistas no entienden el impacto significativo de esas abreviaciones. En el capítulo 4 se presentan los análisis de sistemas y su importancia general

en un proyecto. Los capítulos posteriores le enseñan habilidades específicas de análisis de sistemas en una elaboración de modelos lógicos de sistemas.

En el capítulo 5, “Técnicas de exploración de hechos para identificación de los requerimientos”, se enseñan diversas técnicas y estrategias utilizadas para obtener requerimientos del usuario para un nuevo sistema.

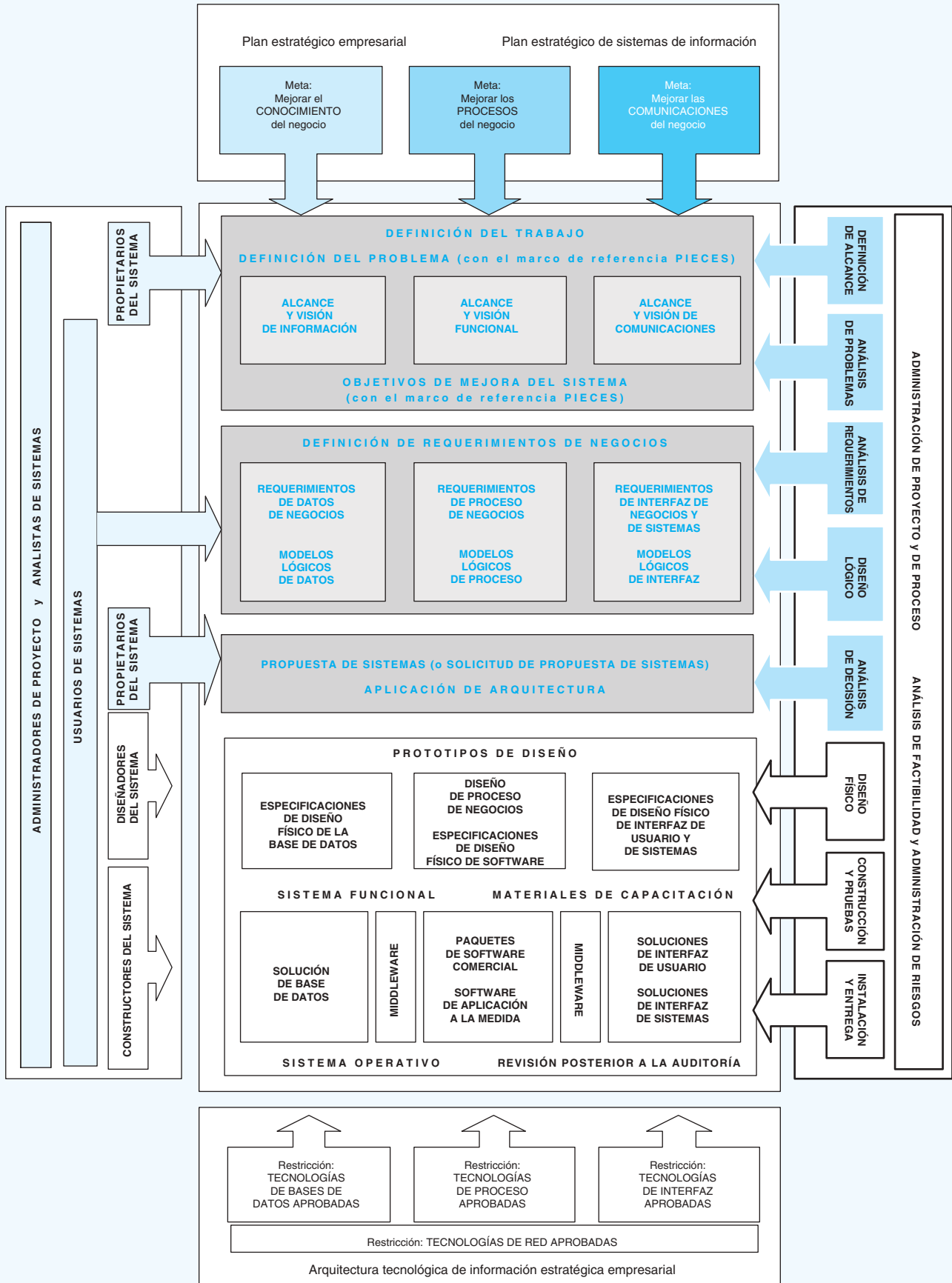
En el capítulo 6, “Modelado de requerimientos del sistema con los casos de uso”, aprenderá acerca de las herramientas y técnicas necesarias para realizar una elaboración de modelos de casos de uso (prácticos) para documentar los requerimientos del sistema.

En el capítulo 7, “Modelado y análisis de datos”, le enseñamos la *elaboración de modelos de datos*, una técnica para organizar y documentar los requerimientos de datos almacenados para un sistema. Aprenderá a dibujar diagramas de relaciones de entidades como una herramienta para estructurar datos de negocios que eventualmente se-

rán diseñados como una base de datos. Estos modelos capturarán las asociaciones y reglas de negocios que deben regir esos datos.

En el capítulo 8, “Modelado de procesos”, se presenta la *elaboración de modelos de proceso*. Se explica cómo se pueden utilizar los diagramas de flujo de datos para describir los procesos de negocios básicos en un sistema, el flujo de datos en un sistema y las políticas y los procedimientos que se implementarán por medio de los procesos. Si usted ha programado algún sistema, reconocerá la importancia de entender los procesos de negocios para los que usted trata de escribir programas.

En el capítulo 9, “Análisis de factibilidad y propuesta del sistema”, se enseñará la forma de obtener soluciones posibles del sistema por medio de una lluvia de ideas, analizar esas soluciones para conocer su factibilidad, elegir la mejor solución general y luego presentar su recomendación en forma de propuesta oral y escrita a la administración.



# 4

## Análisis de sistemas

### Panorámica y objetivos del capítulo

En este capítulo, usted aprenderá más acerca de las fases de análisis de sistemas en un proyecto de desarrollo de sistemas; a saber, definición de alcance, análisis de problema, análisis de requerimientos y fases de análisis de decisión. Las primeras tres fases son llamadas en forma colectiva *análisis de sistemas*. La última fase proporciona una transición entre el análisis de sistemas y el diseño de sistemas. Usted sabrá que comprende el proceso del análisis de sistemas cuando pueda:

- Definir análisis de sistemas y relacionar el término con la definición de alcance, análisis de problema, análisis de requerimientos, diseño lógico y fases de análisis de decisión de la metodología de desarrollo de sistemas de este texto.
- Describir diversos métodos de análisis de sistemas para resolver problemas de sistemas de negocios.
- Describir la definición de alcance, análisis de problemas, análisis de requerimientos, diseño lógico y las fases de análisis de decisión en términos de los bloques de construcción de su sistema de información.
- Describir la definición de alcance, análisis de problemas, análisis de requerimientos, diseño lógico y las fases de análisis de decisión en términos de propósito, participantes, entradas, salidas, técnicas y pasos.
- Identificar en este libro los capítulos que pueden ayudarle a aprender herramientas y técnicas específicas de análisis de sistemas.

NOTA: Aunque en este capítulo se revisan algunas herramientas y técnicas del análisis de sistemas, *no* es el propósito de este capítulo enseñarlas; en él se revisa sólo el *proceso* del análisis de sistemas. Las herramientas y técnicas se enseñarán en los cinco capítulos posteriores.

## Introducción

Bob Martínez recuerda haber aprendido en la universidad que el análisis de sistemas define lo que un sistema de información necesita hacer mientras que un diseño de sistemas define la forma en que tiene que hacerlo. En ese tiempo, sonaba como un proceso simple de dos pasos. Ahora, mientras comienza a trabajar en el proyecto de sistemas de servicios de membresía de SoundStage, puede ver que hay múltiples fases y varios pasos dentro del análisis de sistemas y del diseño de sistemas.

El proyecto de SoundStage está al inicio del análisis de sistemas, en lo que Sandra, su jefa, llama la fase de definición de alcance. Después de eso, harán un análisis del problema, análisis de requerimientos y análisis de decisiones. Suena como mucho trabajo sólo para entender *qué* debe hacer el sistema. Pero este es un sistema complicado. Como dice Sandra, ¿construiría usted una casa sin un buen juego de planos?

## ¿Qué es un análisis de sistemas?

**análisis de sistemas** Técnica de solución de problemas que descompone el sistema en sus componentes para estudiar el grado en que éstos funcionan e interactúan para lograr su propósito.

**diseño de sistemas** Técnica complementaria (de la de análisis de sistemas) de solución de problemas que reensambla los componentes de un sistema en el sistema completo, con la esperanza de mejorarlo. Ello puede abarcar la adición, la eliminación y el cambio de componentes en relación con el sistema original.

**análisis de sistemas de información** Las fases de desarrollo de un proyecto de desarrollo de sistemas de información que se centran principalmente en los problemas y requerimientos de negocios, con independencia de la tecnología que pueda usarse o se use para implantar una solución al problema.

**repositorio** Base de datos o directorio de archivos donde los desarrolladores de sistemas guardan toda la documentación, conocimientos y herramientas de uno o más proyectos o sistemas de información. Los depósitos usualmente se automatizan para facilitar el almacenamiento y la recuperación, así como para compartir la información.

En el capítulo 3 usted aprendió acerca del proceso de desarrollo de sistemas y, a propósito, limitamos nuestro análisis a considerar de manera breve cada fase. En este capítulo, damos un vistazo más cercano a esas fases que en forma colectiva se llaman **análisis de sistemas**. Con la definición formal al margen, el análisis de sistemas es el estudio de un sistema y sus componentes. Es un requerimiento previo para el **diseño de sistemas**, la especificación de un sistema nuevo y mejorado. Este capítulo se enfocará en el análisis de sistemas. En el capítulo 10 se hará lo mismo para el diseño de sistemas.

Al pasar de esta definición clásica del análisis de sistemas hacia algo un poco más contemporáneo, vemos que el *análisis de sistemas* es un término que en forma colectiva describe las fases iniciales del desarrollo de sistemas. En la figura 4.1 se marcaron las etapas 1 a 5 para identificar las fases de análisis de sistemas en el contexto de la ruta clásica completa para nuestra metodología *FAST* (del capítulo 3). Nunca ha habido una definición universalmente aceptada del análisis de sistemas. De hecho, nunca ha habido un acuerdo universal acerca de cuándo termina un análisis de sistema de información y cuándo comienza el diseño de sistemas de información. Para el propósito de este libro, con el **análisis de sistemas de información** se enfatizan los temas de *negocios*, no los asuntos técnicos o de implementación.

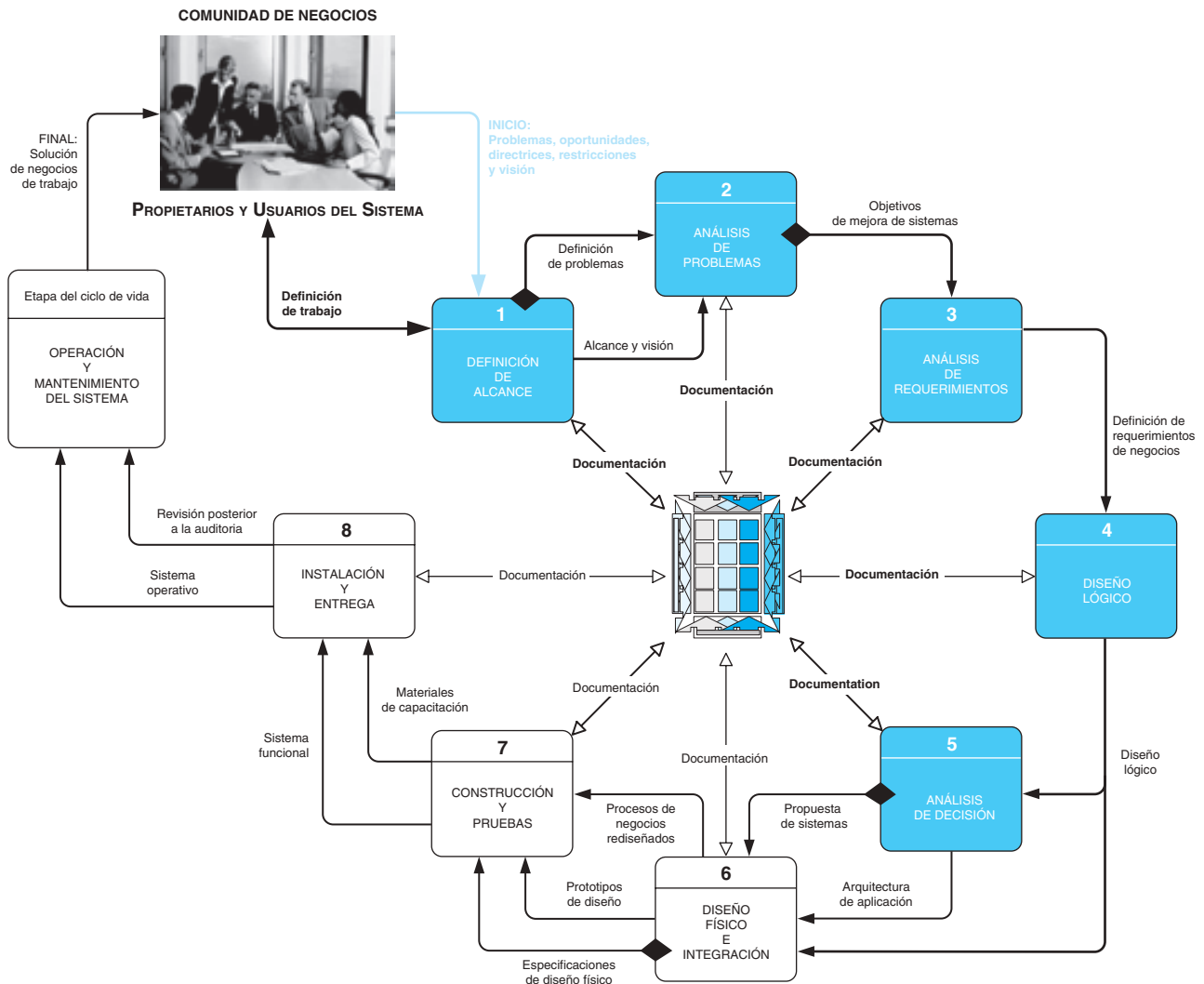
El análisis de sistemas es impulsado por los asuntos de negocios de los PROPIETARIOS DE SISTEMAS y LOS USUARIOS DE SISTEMAS. Por tanto, aborda los bloques de construcción de CONOCIMIENTO, PROCESO Y COMUNICACIONES desde las perspectivas de LOS PROPIETARIOS DEL SISTEMA y LOS USUARIOS DE SISTEMAS. LOS ANALISTAS DE SISTEMAS sirven como facilitadores del análisis de sistemas. Este contexto se ilustra en la página de inicio del capítulo que precedió a los objetivos del mismo.

La documentación y los productos obtenidos por las tareas de análisis de sistemas por lo general se almacenan en un repositorio. Éste puede ser creado para un solo proyecto o ser compartido por todos los proyectos y sistemas. Por lo general un repositorio se implementa como una combinación de lo siguiente:

- Un directorio de red de procesadores de palabras, hojas de cálculo y otros archivos generados en computadoras que contienen correspondencia de proyectos, informes y datos.
- Uno o más diccionarios o enciclopedias de herramientas CASE (como se analizó en el capítulo 3).
- Documentación impresa (como la almacenada en carpetas y bibliotecas de sistemas).
- Una interfaz de sitio Web de *intranet* para los componentes anteriores (útil para la comunicación).

De aquí en adelante, nos referiremos a los componentes en forma colectiva como **repositorio**.

En este capítulo analizaremos cada una de nuestras cinco fases de análisis de sistemas con mayor detalle. Pero primero, analicemos algunas estrategias generales para el análisis de sistemas.



**FIGURA 4.1** El contexto del análisis de sistemas

## Enfoques de análisis de sistemas

De manera fundamental, el análisis de sistemas trata acerca de la *solución de problemas*. Hay muchos métodos para ello; por lo tanto, no le debe sorprender que haya muchos enfoques para el análisis de sistemas. Estos enfoques con frecuencia son vistos como alternativas en *competencia*. En realidad, ciertas combinaciones pueden y deben complementarse entre ellas. Esto se presentó en el capítulo 3 como *métodos acelerados*. Analicemos de manera breve los diversos enfoques.

NOTA: La intención aquí es sólo desarrollar una comprensión de alto nivel. En los capítulos posteriores a esta unidad, se enseñarán las técnicas subyacentes.

### > Enfoques de análisis basados en modelos

El análisis estructurado, la ingeniería de información y el análisis orientado a objetos son ejemplos de un **análisis basado en modelos**. Este análisis utiliza imágenes para comu-

#### **análisis basado en modelos**

Una estrategia de solución de problemas que hace énfasis en trazar modelos de sistemas de imágenes para documentar y validar los sistemas existentes o propuestos. En última instancia, el modelo de sistema se convierte en el plano para diseñar y construir un sistema mejorado.

**modelo** Representación de la realidad. Puesto que “una imagen vale más que mis palabras”, en muchos modelos se usan imágenes para representar la realidad.

nicar problemas de negocios, requerimientos y soluciones. Ejemplos de **modelos** con los que puede ya estar familiarizado incluyen diagramas de flujo, cuadros de estructura o jerarquías y organigramas.

En la actualidad, los enfoques basados en modelos casi siempre se resaltan por el uso de herramientas automatizadas. Algunos analistas dibujan modelos de sistemas con software gráfico de propósitos generales como *Visio* de Microsoft. Otros analistas y organizaciones requieren el uso de herramientas basadas en repositorios CASE o de elaboración de modelos como *System Architect*, *Visible Architect*, *Visible Analyst* o *Rational ROSE*. Las herramientas CASE ofrecen la ventaja del análisis consistente y completo así como una revisión de errores basada en reglas.

Los enfoques de análisis basados en modelos se presentan en las metodologías y rutas basadas en modelos que se presentaron en el capítulo 3. Analicemos de manera breve los tres enfoques más populares de análisis basados en modelos.

**Métodos tradicionales** Al inicio de la década de los setenta se desarrollaron diversos métodos tradicionales para el análisis y el diseño de sistemas. Uno de los primeros métodos formales, que aún es ampliamente utilizado en la actualidad es el *análisis estructurado*. El **análisis estructurado** se enfoca en el flujo de datos a través de los procesos de negocios y de software. Se dice que está *centrado en el proceso*. Por centrado en el proceso queremos decir que el énfasis está en los componentes (bloques de construcción) del PROCESO en su marco de referencia del sistema de información.

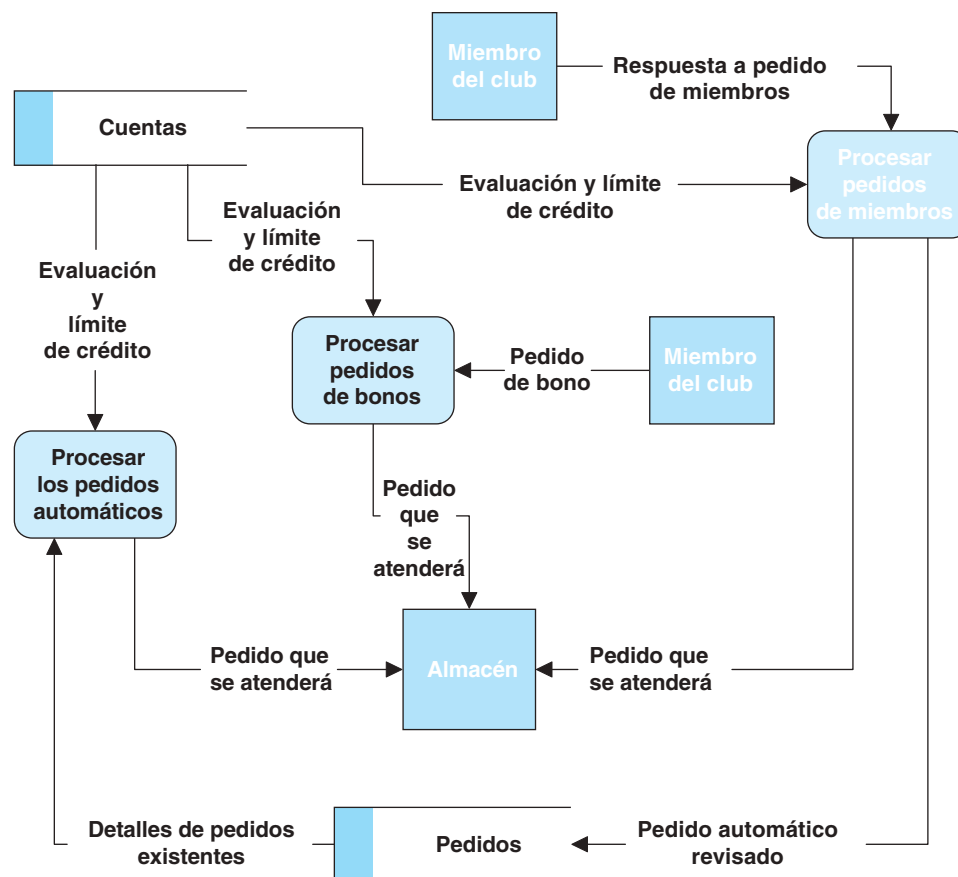
Una de las herramientas clave utilizadas para elaborar modelos de procesos es el *diagrama de flujo de datos* (figura 4.2) que describe los procesos existentes o propuestos en un sistema junto con sus entradas, salidas y datos. Los modelos muestran el flujo de datos entre y a través de los procesos y los lugares donde se almacenan los datos. Por último, estos modelos de procesos sirven como planos para que los procesos de negocios sean implementados y que el software se adquiera o se construya.

La práctica de análisis estructurado para el diseño de software ha disminuido mucho en favor de los métodos orientados a objetos. Sin embargo, la elaboración de modelos

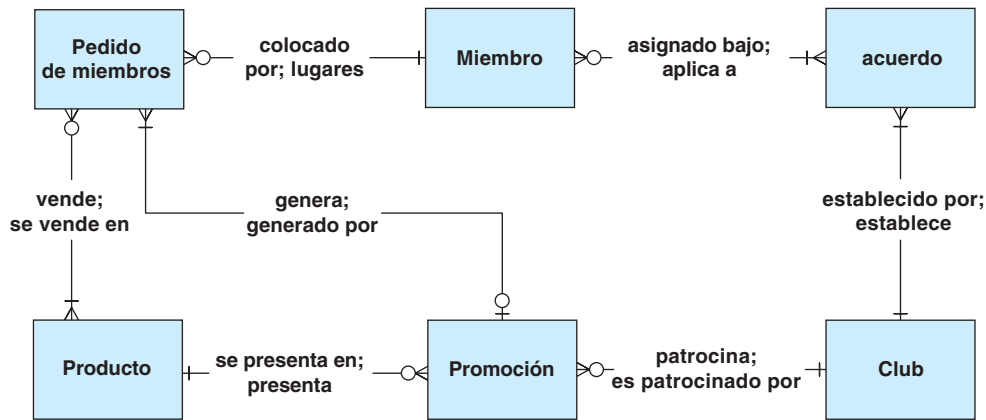
**análisis estructurado** Técnica centrada en PROCESOS y operada por modelos que se usa para analizar un sistema existente, para definir los requerimientos de negocios de un nuevo sistema o para ambos objetivos. Los modelos son imágenes que ilustran los componentes del sistema: procesos, entradas, salidas y archivos.

**FIGURA 4.2**

Modelo de proceso simple (también llamado diagrama de flujo de datos)





**FIGURA 4.3**

Modelo de datos simples (también llamado diagrama de entidad-relación)

de proceso disfruta un cierto reavivamiento gracias el énfasis renovado en el rediseño de procesos de negocios, que se analizará posteriormente en este capítulo.

Otro método tradicional, llamado **ingeniería de información (IE)**, se enfoca en la estructura de datos almacenados en un sistema más que en los procesos. Por ello, se dijo que era *centrado en los datos*, al enfatizar el análisis de los requerimientos de CONOCIMIENTO (o datos). La herramienta fundamental para modelar requerimientos de datos es el diagrama de relación de entidad (figura 4.3). Los diagramas de relación de entidades aún son muy utilizados en el diseño de bases de datos de relaciones.

En un principio, la ingeniería de información era vista como un método en competencia con el análisis estructurado. Con el paso del tiempo muchas personas los hicieron complementarios; utilizan diagramas de flujo de datos para modelar los procesos de sistemas y los diagramas de relaciones de entidades para modelar un sistema de datos.

**Estrategia orientada a objetos** Los métodos tradicionales separaban las preocupaciones en forma deliberada del CONOCIMIENTO (datos) de los de PROCESOS. Aunque la mayoría de los métodos de análisis de sistemas intentaron sincronizar modelos de datos y procesos, ese intento no siempre funcionaba bien en la práctica. Las tecnologías de **objeto** han surgido desde entonces para eliminar esta separación artificial de datos y procesos. La **estrategia orientada a objetos** ve los sistemas de información no como datos y procesos, sino como una colección de objetos que encapsulan datos y procesos. Los objetos pueden contener atributos de datos. Sin embargo, la única forma de crear, leer, actualizar o eliminar los datos de un objeto es a través de uno de sus procesos incrustados (llamado métodos). Los lenguajes de programación orientados a objetos como *Java*, *C++* y los lenguajes *.NET*, se vuelven cada vez más populares.

La estrategia orientada a objetos tiene un conjunto completo de herramientas de elaboración de modelos conocido como Unified Modeling Language (UML). Uno de los diagramas UML de clase de objetos, se muestra en la figura 4.4. Algunas de las herramientas UML han ganado aceptación por los proyectos de sistemas incluso cuando el sistema de información no sea implementado con tecnologías orientadas a objetos.

### > Enfoques de análisis de sistemas acelerados

El desarrollo de la elaboración de prototipos de identificación y de arquitectura rápida son ejemplos de enfoques de sistemas acelerados que enfatizan la construcción de prototipos para identificar con más rapidez los requerimientos de negocios y de usuarios para un sistema nuevo. La mayoría de dichos métodos se derivan de alguna variación en la construcción de **prototipos**, muestras funcionales pero incompletas de un sistema deseado. Los prototipos sirven a la forma de pensar de “sabré lo que quiero cuando lo vea”, que es característico de muchos usuarios y administradores. Por “incompleto” queremos decir que un prototipo no incluirá la revisión de errores, validación de entrada de datos, seguridad y totalidad de proceso de una aplicación terminada. Ni tampoco estará tan pulida ni ofrecerá ayuda al usuario como en el sistema final. Pero como puede ser desarrollado con rapidez, de igual forma puede identificar los requerimientos más cruciales de nivel de negocios. A veces, los prototipos pueden evolucionar para convertirse en los sistemas finales, es decir, las aplicaciones completas.

**ingeniería de información (IE)** Una técnica operada por modelos y centrada en DATOS, pero sensible a PROCESOS, para la planeación, el análisis y el diseño de sistemas de información. Los modelos de IE son imágenes que ilustran y sincronizan los datos y procesos del sistema.

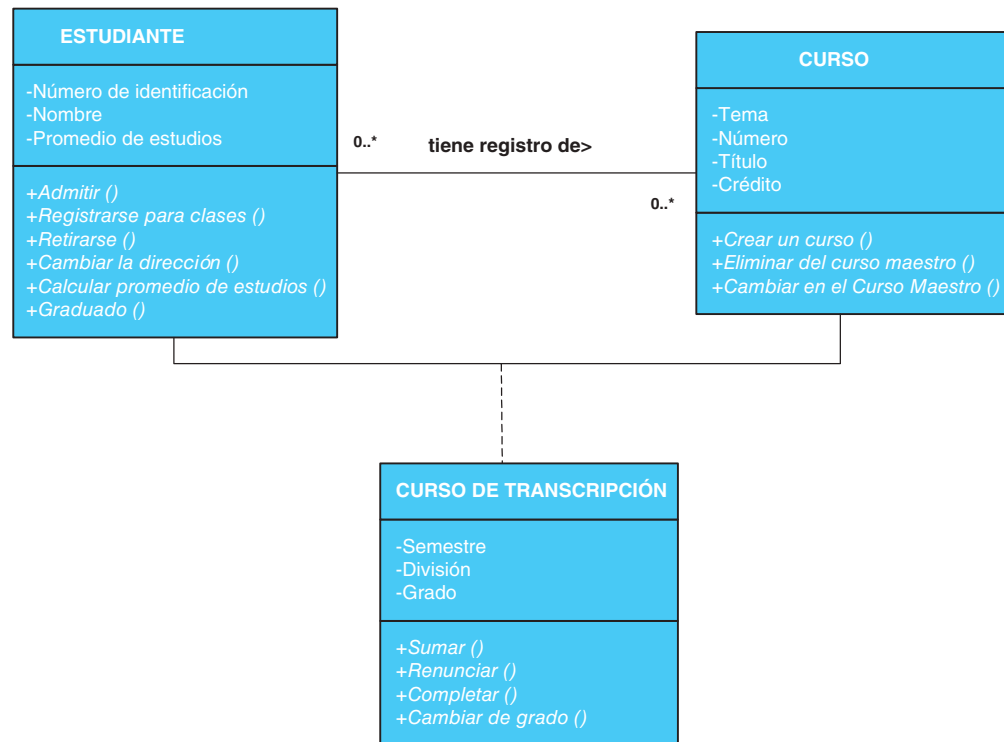
**objeto** Encapsulación de datos (llamados *propiedades*) que describen a una persona, objeto, sitio o evento, con todos los procesos (llamados *métodos*) permitidos para usar o actualizar los datos y propiedades. La única forma de tener acceso a los datos del objeto o actualizarlos es usar los procesos predefinidos del objeto.

**estrategia orientada a objetos** Técnica basada en modelos que integra los datos y procesos en conceptos llamados *objetos*. Los modelos de objetos son imágenes que ilustran los objetos del sistema desde diversas perspectivas, como la estructura, el comportamiento y las interacciones entre ellos.

**prototipo** Muestra a pequeña escala, un ejemplo incompleto pero funcional de un sistema deseado.

FIGURA 4.4

Modelo de objeto  
(con el estándar  
del lenguaje de  
elaboración de  
modelos unificado)



En un sentido, los métodos de análisis acelerado ponen mucho énfasis en los componentes de COMUNICACIONES en su marco de referencia de sistema de información al construir formas e informes de muestras. Al mismo tiempo, las herramientas de software utilizadas para construir prototipos también incluyen los componentes de DATOS y PROCESOS.

Estos métodos acelerados son comunes en las metodologías de desarrollo rápido de aplicaciones (rapid application development, RAD) y las rutas que fueron presentadas en el capítulo 3. Los métodos RAD requieren herramientas automatizadas mientras que algunas herramientas CASE basadas en repositorios incluyen instalaciones RAD muy simples, la mayoría de los analistas utilizan verdaderos ambientes de programación RAD, como *Powerbuilder* de Sybase, *Access* de Microsoft, *Visual Basic .NET* de Microsoft o *Websphere Studio for Application Development* de IBM (basado en *Java*).

Analicemos de manera breve dos métodos populares de análisis acelerado.

#### elaboración de prototipos de identificación

Técnica usada para identificar los requerimientos de negocios de los usuarios al hacerlos reaccionar a una implantación rápida y no acabada de esos requerimientos.

**Elaboración de prototipos de identificación** La elaboración de prototipos de identificación utiliza tecnología de desarrollo rápido para ayudar a los usuarios a identificar sus requerimientos de negocios. Por ejemplo, es muy común que los analistas de sistemas utilicen una herramienta de desarrollo simple como *Access* de Microsoft para crear en forma rápida una base de datos simple, formatos de entrada de usuario e informes de muestra para solicitar respuestas de usuarios en cuanto a si la base de datos, formatos e informes representan en realidad los requerimientos de negocios. La intención es desarrollar de manera normal el nuevo sistema final en una herramienta y un lenguaje de desarrollo de aplicación más sofisticado, pero la herramienta más simple permite al analista desarrollar los prototipos de acuerdo con los requerimientos de los usuarios con una mayor rapidez.

En la elaboración de prototipos para identificación, tratamos de desalentar a los usuarios de preocuparse por la “visión y sensación” final de los prototipos de sistemas, ¡que pueden cambiarse durante el diseño del sistema! Aquí reside la crítica primaria de la elaboración de prototipos; las plantillas de software existen en las herramientas de elaboración de prototipos para generar en forma rápida algunos prototipos muy elegantes y visualmente atractivos. La desventaja es que esto puede alentar un enfoque y compromiso prematuro con el diseño representado en el prototipo. Los usuarios también pueden ser llevados a creer en forma equivocada: 1) que el sistema completo puede construirse con la misma rapidez, o 2) que herramientas como *Access* pueden utilizarse para construir el sistema final. Aunque herramientas como *Access* pueden en realidad acelerar el desarrollo de sistemas, su uso en la elaboración de prototipos es rápido sólo porque se omite la

mayor parte del detalle de la base de datos y la programación de aplicaciones requerida para una aplicación completa y segura. También, por lo general, herramientas como *Access* no pueden soportar los tamaños, número de usuarios y tráfico de red de las bases de datos que se requieren en la mayoría de las aplicaciones empresariales.

No obstante, la elaboración de prototipos de identificación es un método preferido y recomendado. Desafortunadamente, algunos analistas y desarrolladores de sistemas utilizan la elaboración de prototipos de identificación para reemplazar en forma total el diseño basado en modelos, sólo para darse cuenta lo que los verdaderos ingenieros han sabido durante años: No se puede elaborar un prototipo sin cierta cantidad de un diseño más formal... y es aquí donde entra el análisis rápido con arquitectura.

**Análisis rápido de arquitectura** El **análisis rápido de arquitectura** es un método de análisis acelerado que también construye modelos de sistemas. El análisis rápido de arquitectura se hace posible mediante la tecnología de **ingeniería inversa** que se incluye en muchas herramientas automatizadas como CASE y lenguajes de programación (como se presentaron en el capítulo 3). Las herramientas de ingeniería inversa generan modelos de sistemas de las aplicaciones de software existentes o los prototipos de sistemas. Los modelos de sistemas resultantes pueden entonces ser editados y mejorados por los analistas de sistemas y los usuarios para proporcionar un plan para un sistema nuevo y mejorado. Debe ser evidente que el análisis rápido con arquitectura es una mezcla de los enfoques basados en modelos y de análisis acelerado.

Hay dos técnicas diferentes para aplicar un análisis de arquitectura rápido:

- La mayoría de los sistemas ya han sido automatizados hasta cierto grado y existen como sistema de información de heredado. Muchas herramientas CASE pueden leer las estructuras de base de datos subyacentes y/o los programas de aplicación, y luego, con ingeniería inversa, regresarlos en diferentes modelos de sistemas. Esos modelos sirven como punto de partida para definir un análisis de requerimientos de usuario basado en modelos.
- Si los prototipos han sido construidos en herramientas como *Access* o *Visual Basic* de Microsoft, esos prototipos a veces pueden ser revertidos con la ingeniería inversa a su equivalente en modelos de sistemas. Estos últimos por lo general se prestan para analizar los requerimientos de los usuarios en cuanto a su consistencia, totalidad, estabilidad, escalabilidad y flexibilidad al cambio futuro. También, los modelos de sistemas con frecuencia pueden ser cambiados por la ingeniería hacia adelante por medio de las mismas herramientas CASE y los ADE (ambientes de desarrollo de aplicación) en las bases de datos y plantillas de aplicación o los esqueletos que utilizarán las robustas bases de datos de nivel empresarial y tecnología de programación.

Ambas técnicas abordan el tema anterior de que los ingenieros rara vez elaboran prototipos en la ausencia total de un diseño más formal y al mismo tiempo, preservan las ventajas de acelerar las fases de análisis de sistemas.

## > Métodos para identificación de requerimientos

Los métodos de análisis de sistemas acelerados y el basado en modelos intentan expresar los requerimientos de usuarios para un nuevo sistema, ya sea como modelos o como prototipos. Pero ambos métodos son, a su vez, dependientes de la necesidad más sutil de identificar y administrar en realidad esos requerimientos. Más aún, los requerimientos para los sistemas dependen de la capacidad de los analistas para identificar los problemas y oportunidades que existen en el sistema actual, por tanto, los analistas deben volverse hábiles para identificar problemas, oportunidades y requerimientos. En consecuencia, todas las estrategias para análisis de sistemas requieren alguna forma de **identificación de requerimientos**. Revisemos de manera breve un par de estrategias para identificación de requerimientos.

**Técnicas de identificación de hechos** La **identificación de hechos** es una habilidad esencial para todos los analistas de sistemas. Las técnicas de identificación de hechos que se abarcan en este texto (de hecho, en el siguiente capítulo) incluyen:

- El muestreo de la documentación existente, informes, formatos, archivos, bases de datos y memorandos.
- Investigación de bibliografía relevante, sondeo en el mercado de otras soluciones y visitas a sitios.

---

**análisis rápido de arquitectura** Estrategia que intenta derivar modelos de sistemas (como se describe con antelación en la misma sección del capítulo) a partir de sistemas existentes o prototipos de identificación.

---

**ingeniería inversa** Uso de tecnología que lee el código de programas a partir de bases de datos, programas de aplicación o interfaces de usuarios existentes y genera automáticamente el modelo de sistemas equivalente.

---

**identificación de requerimientos** Proceso que usan los analistas de sistemas para identificar o extraer problemas de sistemas y requerimientos de solución de la comunidad de usuarios.

---

**identificación de hechos** Proceso de recopilar información acerca de problemas, oportunidades, requerimientos de solución y prioridades del sistema. También denominado *recopilación de información*.

- Observación del sistema actual en acción y el ambiente de trabajo.
- Cuestionarios y encuestas de la administración y la comunidad de usuarios.
- Entrevistas de administradores, usuarios y personal técnico apropiado.

### planeación conjunta de requerimientos (JRP)

Uso de talleres facilitados para reunir a todos los propietarios, usuarios y analistas de un sistema y a ciertos diseñadores y constructores de sistemas con el fin de realizar conjuntamente el análisis de sistemas. La JRP por lo general se considera como parte de un método más amplio, llamado *desarrollo conjunto de aplicaciones (JAD)*, que es una aplicación más completa de las técnicas de JRP al proceso de desarrollo de sistemas en su totalidad.

### rediseño de procesos de negocios (BPR)

Aplicación de métodos de análisis de sistemas con el objetivo de cambiar y mejorar significativamente los procesos de negocios fundamentales de una organización, con independencia de la tecnología de la información.

**Planeación conjunta de requerimientos** Las técnicas de identificación de hechos listadas con anterioridad son invaluable; sin embargo, pueden consumir mucho tiempo en sus formas clásicas. De manera alternativa, la identificación de requerimientos y la administración pueden ser acelerados en forma significativa al usar una técnica de **planeación conjunta de requerimientos (joint requirements planning, JRP)**. Un analista capacitado o certificado en JRP en general desempeña el papel de *facilitador* en un taller que normalmente toma de tres a cinco días completos de trabajo. Este taller puede reemplazar semanas o meses de las clásicas juntas de identificación de hechos y su seguimiento.

Una JRP proporciona un ambiente de trabajo en el cual se aceleran todas las tareas y los productos del análisis de sistemas. Promueve una participación mejorada del PROPIETARIO DE SISTEMAS y del USUARIO DE SISTEMAS en el análisis del sistema. Pero también requiere que un facilitador con excelentes habilidades de negociación y conciliación se asegure de que todas las partes reciban las oportunidades apropiadas para contribuir al desarrollo del sistema.

La JRP por lo general se utiliza en conjunto con los métodos de análisis basados en sistemas que describimos con anterioridad y casi siempre se incorpora en las metodologías y rutas de desarrollo rápido de aplicación (RAD), que se presentaron en el capítulo 3.

## > Métodos de rediseño de procesos de negocios

Una de las aplicaciones contemporáneas más interesantes de los métodos de análisis de sistemas es el **rediseño de procesos de negocios (business process redesign, BPR)**. El interés en el BPR fue dirigido por el descubrimiento de que los sistemas de información y aplicaciones más actuales sólo han automatizado los procesos de negocios existentes e ineficientes. La burocracia automatizada sigue siendo burocracia; la automatización no necesariamente aporta valor al negocio y en verdad puede sustraer valor al negocio. Presentado en el capítulo 1, el BPR es uno de muchos tipos de proyectos disparados por las tendencias que llamamos de *administración de calidad total (total quality management, TQM)* y *mejora continua de procesos (continuous process improvement, CPI)*.

Algunos proyectos de BPR se enfocan en todos los procesos de negocios, sin importar su automatización. Cada proceso de negocios se estudia y analiza con profundidad en busca de cuellos de botella, valor de retorno y oportunidades de eliminación o agilización. Los modelos de procesos, tales como diagramas de flujo de datos (analizados con anterioridad) ayudan a las organizaciones a visualizar sus procesos. Una vez que los procesos de negocios han sido rediseñados, la mayoría de los BPR concluyen al examinar cómo sería mejor aplicada la tecnología de la información a los procesos de negocios mejorados. Esto puede crear nuevos proyectos de desarrollo de sistemas de información y aplicación para implementar o soportar los nuevos procesos de negocios.

BPR también se aplica dentro del contexto de los proyectos de desarrollo de información. Es muy común que los proyectos de IS incluyan un estudio de los procesos de negocios existentes para identificar problemas, burocracia e ineficiencias que puedan ser abordados en requerimientos para nuevos y mejorados sistemas de información y aplicaciones de cómputo.

BPR también se ha vuelto común en los proyectos de IS que estarán basados en la compra e integración de software listo para usarse (commercial off-the-shelf, COTS). La adquisición de software COTS en general requiere que un negocio adapte sus procesos de negocios para encajar en el software. Un análisis de procesos de negocios existentes durante el análisis de sistemas casi siempre es parte de dichos proyectos.

## > Estrategias de análisis de sistemas FAST

Al igual que la mayoría de las metodologías comerciales, nuestra metodología hipotética *FAST* no impone un solo método en los analistas de sistemas. En su lugar, integra todos los métodos populares presentados en los párrafos anteriores en una colección de **métodos acelerados**. El caso de estudio de SoundStage demostrará estos métodos en el contexto de una primera asignación de un analista de sistemas. Las técnicas de análisis de sistemas serán aplicadas dentro del marco de referencia de:

**método acelerado** Integración de diversos enfoques del análisis y diseño de sistemas para su aplicación según se considere apropiado al problema que se intenta resolver y el sistema que se está desarrollando.

- Los componentes de su sistema de información (del capítulo 2).
- Las fases de *FAST* (del capítulo 3).
- Las tareas de *FAST* que implementan una fase (descritas en este capítulo).

Dado este contexto para estudiar análisis de sistemas, ahora podemos explorar las fases y tareas del análisis de sistemas.

## Fase de definición de alcance

Recuerde del capítulo 3 que la *fase de definición de alcance* es la primera del proceso de desarrollo de sistemas clásico. En otras metodologías esto podría ser llamado *fase de investigación preliminar*, *fase de estudio inicial*, *fase de investigación* o *fase de planeación*. La fase de definición de alcance responde la pregunta, “¿vale la pena realizar este proyecto?”. Para responder esta pregunta, debemos definir el alcance del proyecto y los problemas percibidos, oportunidades y directrices que dispararon el proyecto. Supongamos que el problema *se considera* digno de investigarse, la fase de definición de alcance debe también establecer el plan de proyecto en términos de escala, estrategia de desarrollo, programación, requerimientos de recursos y presupuesto.<sup>1</sup>

El contexto para la fase de definición de alcance aparece en un rectángulo punteado en la figura 4.5. Nótese que la fase de definición de alcance tiene como prioridad el punto de vista del PROPIETARIO DEL SISTEMA del ya existente y los problemas y oportunidades que dispararon el interés. Los propietarios de sistemas tienden a estar preocupados por la imagen general, no por los detalles. Más aún, ellos determinan si los recursos pueden y serán comprometidos con el proyecto.

En la figura 4.6 se encuentra el primero de cinco diagramas de tareas que presentaremos en este capítulo para dar un vistazo más de cerca a cada fase de análisis de sistemas. Un *diagrama de tareas* muestra el trabajo (= tareas) que debe ser realizado para completar una fase. Nuestros diagramas de tareas no indican una metodología específica, pero describiremos en los párrafos anexos los métodos, herramientas y técnicas que usted podría considerar para cada tarea. En la figura 4.6 se muestran las tareas requeridas para la fase de definición de alcance. Es importante recordar que estos diagramas de tareas son sólo plantillas. El equipo de proyecto y el administrador del proyecto pueden expandir o alterar estas plantillas para reflejar las necesidades únicas de cualquier proyecto dado.

Como se muestra en la figura 4.6, el producto final de la fase de investigación preliminar es el cumplimiento de una CARTA DE DEFINICIÓN DEL PROYECTO. (Esos importantes productos están indicados en cada diagrama de tareas en letras mayúsculas.) Una carta de definición de proyecto define el alcance de proyecto, el plan, la metodología, los estándares y demás. El completar la carta de definición de proyecto representa el primer hito en un proyecto.

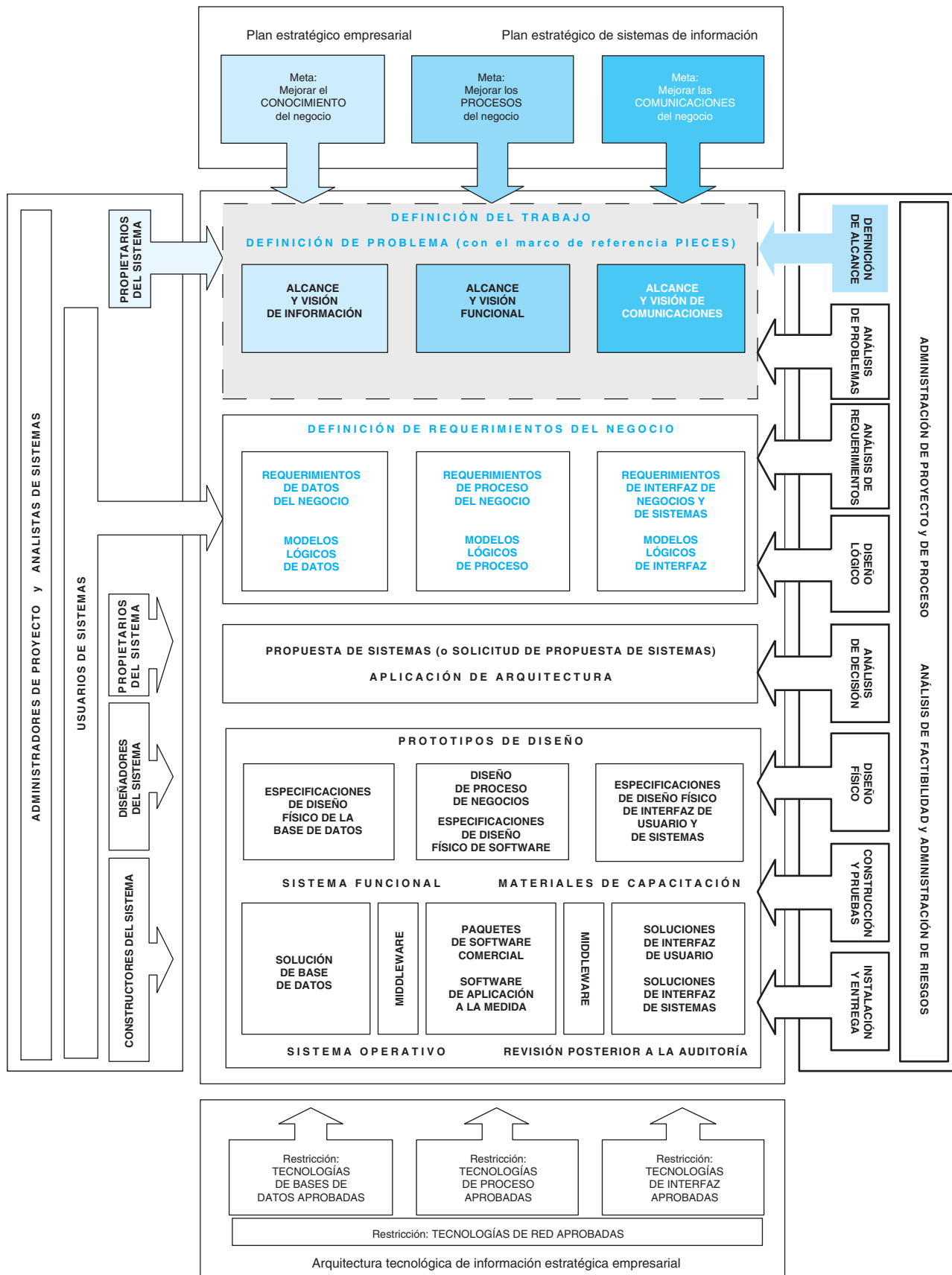
Se pretende que la fase de definición de alcance sea rápida. La fase completa no debe exceder dos o tres días para la mayoría de los proyectos. En general, la fase incluye las siguientes tareas:

- 1.1 Identificar problemas y oportunidades básicos.
- 1.2 Negociar alcance base.
- 1.3 Considerar el valor del proyecto base.
- 1.4 Desarrollar un programa y presupuesto iniciales.
- 1.5 Comunicar el plan de proyecto.

Ahora analicemos cada una de estas tareas con mayor detalle.

<sup>1</sup> Estos elementos de planeación forman parte del proceso utilizado por los administradores de proyecto para desarrollar un plan de proyecto.\*

\* Nota del R.T.: Para el lector interesado, se recomienda consultar un libro sobre administración de proyectos, como el PMBOK del PMI.



**FIGURA 4.5** El contexto de la fase de definición de alcance del análisis de sistemas

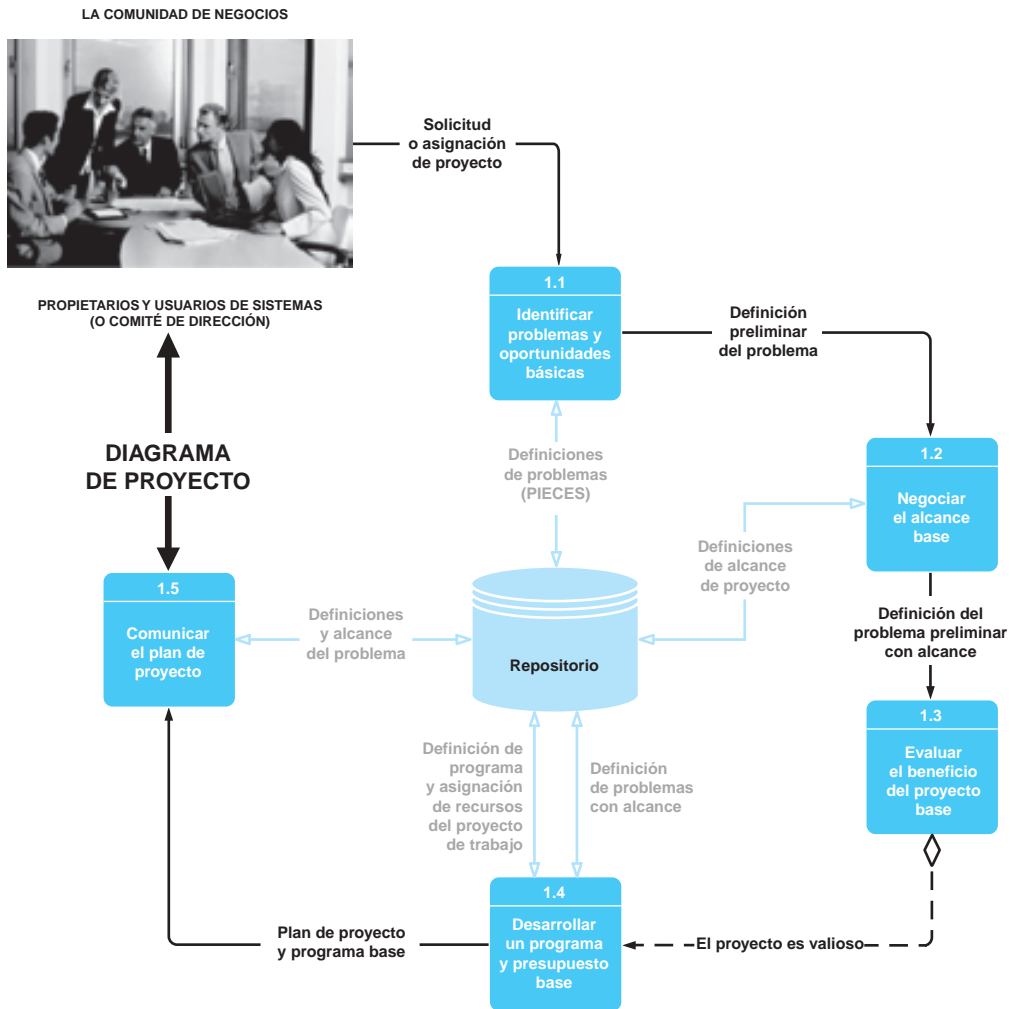


FIGURA 4.6

Tareas para la fase de definición de alcance del análisis de sistemas

## > Tarea 1.1: Identificar problemas y oportunidades básicas

Una de las tareas más importantes de la fase de definición de **alcance** es establecer una base inicial de los problemas, oportunidades o directrices que dispararon el proyecto. Cada problema, oportunidad y directriz está evaluado con relación a la urgencia, visibilidad, beneficios tangibles y prioridad. Cualquier análisis adicional y detallado no es relevante en esta etapa del proyecto. Sin embargo, puede ser útil listar cualquier restricción (limitación) del proyecto, como fechas límites, presupuestos máximos o arquitectura tecnológica.

Por lo general un analista de sistemas con mucha experiencia o un administrador de proyecto dirige esta tarea. La mayoría de los otros participantes se clasifican en general como PROPIETARIOS DEL SISTEMA. Esto incluye a los patrocinadores ejecutivos, a los administradores de más alto nivel que pagarán y apoyarán el proyecto. También incluye administradores de todas las unidades organizacionales que pueden ser impactados por el sistema y es posible que incluya a los administradores de sistemas de información. LOS USUARIOS DE SISTEMAS, DISEÑADORES DE SISTEMAS Y CONSTRUCTORES DE SISTEMAS por lo general no participan en esta tarea.

Como se muestra en la figura 4.6, una SOLICITUD DEL PROYECTO O ASIGNACIÓN dispara la tarea. Este disparador puede tomar una o varias formas alternativas. Puede ser tan simple como un memorando de autoridad de un comité de dirección de sistemas de información. O puede ser un memorando de un equipo o una unidad de negocios que solicita un desarrollo de sistemas. Algunas organizaciones requieren que todas las solicitudes de proyectos sean remitidas en formato de solicitud de servicio, como el de la figura 4.7.

**alcance** Límites de un proyecto: las áreas de un negocio que el proyecto podría atender (o no).

**FIGURA 4.7**

Solicitud de servicios de sistemas

<b>SoundStage Entertainment Club</b> <b>Information System Services</b> Phone: 494-0666 Fax: 494-0999 Internet: http://www.soundstage.com Intranet: http://www.soundstage.com/iss		<b>REQUEST FOR INFORMATION SYSTEM SERVICES</b>
<b>DATE OF REQUEST</b>	<b>SERVICE REQUESTED FOR DEPARTMENT(S)</b>	
January 9, 2003	Member Services, Warehouse, Shipping	
<b>SUBMITTED BY (key user contact)</b> <b>Name</b> Sarah Hartman <b>Title</b> Business Analyst, Member Services <b>Office</b> B035 <b>Phone</b> 494-0867		<b>EXECUTIVE SPONSOR (funding authority)</b> <b>Name</b> Galen Kirkhoff <b>Title</b> Vice President, Member Services <b>Office</b> G242 <b>Phone</b> 494-1242
<b>TYPE OF SERVICE REQUESTED:</b> <input type="checkbox"/> Information Strategy Planning <input checked="" type="checkbox"/> Business Process Analysis and Redesign <input checked="" type="checkbox"/> New Application Development <input type="checkbox"/> Other (please specify) _____		
<input type="checkbox"/> Existing Application Enhancement <input type="checkbox"/> Existing Application Maintenance (problem fix) <input type="checkbox"/> Not Sure		
<b>BRIEF STATEMENT OF PROBLEM, OPPORTUNITY, OR DIRECTIVE (attach additional documentation as necessary)</b> The information strategy planning group has targeted member services, marketing, and order fulfillment (inclusive of shipping) for business process redesign and integrated application development. Currently serviced by separate information systems, these areas are not well integrated to maximize efficient order services to our members. The current systems are not adaptable to our rapidly changing products and services. In some cases, separate systems exist for similar products and services. Some of these systems were inherited through mergers that expanded our products and services. There also exist several marketing opportunities to increase our presence to our members. One example includes Internet commerce services. Finally, the automatic identification system being developed for the warehouse must fully interoperate with member services.		
<b>BRIEF STATEMENT OF EXPECTED SOLUTION</b> We envision completely new and streamlined business processes that minimize the response time to member orders for products and services. An order shall not be considered fulfilled until it has been received by the member. The new system should provide for expanded club and member flexibility and adaptability of basic business products and services. We envision a system that extends to the desktop computers of both employees and members, with appropriate shared services provided across the network, consistent with the ISS distributed architecture. This is consistent with strategic plans to retire the AS/400 central computer and replace it with servers.		
<b>ACTION (ISS Office Use Only)</b> <input type="checkbox"/> Feasibility assessment approved <input checked="" type="checkbox"/> Feasibility assessment waived <input type="checkbox"/> Request delayed <input type="checkbox"/> Request rejected		
Assigned to <u>Sandra Shepherd</u> Approved Budget \$ <u>450,000</u> Start Date <u>ASAP</u> Deadline <u>ASAP</u> Backlogged until date: _____ Reason: _____		
<b>Authorized Signatures:</b> <u>Rebecca J. Todd</u> Chair, ISS Executive Steering Body		
<u>Galen Kirkhoff</u> Project Executive Sponsor		
FORM ISS-100-RFSS (Last revised December, 1999)		

El producto fundamental de esta tarea, la DEFINICIÓN PRELIMINAR DEL PROBLEMA, consiste en problemas, oportunidades y directrices que fueron identificadas. Las DEFINICIONES DE PROBLEMAS se almacenan en el repositorio para uso posterior en el proyecto. La figura 4.8 es un documento muestra que resume problemas, oportunidades y directrices en términos de:

- **Urgencia.** ¿En qué tiempo debe ser resuelto el problema o debe ser realizada la oportunidad o directriz? Una escala de calificación puede desarrollarse para responder de manera consistente a esta pregunta.
- **Visibilidad.** ¿A qué grado una solución o un sistema nuevo debe ser visible para los clientes o la administración ejecutiva? De nuevo, podría desarrollarse una escala de calificación para las respuestas.
- **Beneficios.** ¿Aproximadamente cuánto incrementaría una solución o un sistema nuevo los ingresos anuales o reduciría los costos anuales? Esto a menudo es incierto, pero si todos los participantes cooperan en esta estimación, demostrará ser bastante conservadora.



## Definición de problema

Proyecto:	Sistema de información de servicios a miembros	Administrador de proyecto:	Sandra Shepherd
Creado por:	Sandra Shepherd	Actualizado por última vez por:	Roberto Martínez
Fecha de creación:	9 de enero de 2003	Fecha de última actualización:	15 de enero de 2003

Definiciones breves de problema, oportunidad o directriz	Urgencia	Visibilidad	Beneficios anuales	Prioridad o lugar	Solución propuesta
1. El tiempo de respuesta del pedido medido desde el momento de recepción, hasta la entrega del mismo ha aumentado en un promedio de 15 días.	Tan pronto como sea posible	Alta	\$175 000	2	Nuevo desarrollo
2. Las recientes adquisiciones de Club de video de monitoreo privado y pantalla de juego enfatizarán más aún los requerimientos del sistema actual.	6 meses	Mediana	75 000	2	Nuevo desarrollo
3. Actualmente, tres distintos servicios de entrada de sistemas, audio, video y divisiones de juego. Cada sistema está diseñado para tener una interfaz con un sistema de almacenes distinto; por tanto, la intención de fusionar el inventario a un solo almacén se ha retrasado.	6 meses	Mediana	515 000	2	Nuevo desarrollo
4. Hay una falta general de acceso a la información administrativa y de toma de decisiones. Esto se torna difícil por la adquisición de dos sistemas de proceso de pedidos adicionales (de filtración privada y pantalla de juego).	12 meses	Baja	15 000	3	Después de que un sistema nuevo se desarrolla, es necesario proporcionar a los usuarios herramientas de reporte fáciles de aprender y utilizar
5. Actualmente existen inconsistencias de datos en los archivos de miembros y pedidos.	3 meses	Alta	35 000	1	Arreglo rápido; luego nuevo desarrollo
6. La filtración privada y los archivos de sistemas de pantalla de juegos son incompatibles con los equivalentes de SoundStage. Los problemas de datos de negocios incluyen inconsistencia de datos y falta de controles de edición de entrada.	6 meses	Mediana	Desconocidos	2	Nuevo desarrollo. Cuantificación adicional de beneficio podría incrementar la urgencia
7. Hay una oportunidad de abrir sistemas de pedidos en el Internet, pero la seguridad y el control son un tema.	12 meses	Baja	Desconocidos	4	Versión futura de un sistema recién desarrollado
8. El sistema de entrada de pedidos actual es incompatible con el siguiente sistema de identificación automática (código de barras) que se desarrolla para el almacén.	3 meses	Alta	65 000	1	Arreglo rápido; luego nuevo desarrollo

**FIGURA 4.8** Muestra de definiciones de problemas

- *Prioridad.* Con base en las respuestas anteriores, ¿cuáles son las prioridades acordadas entre todos para cada problema, oportunidad o directriz? Si el presupuesto o el programa se vuelven un problema, estas prioridades ayudarán a ajustar el alcance del proyecto.
- *Soluciones posibles (OPT).* En esta etapa inicial del proyecto, las soluciones posibles se expresan mejor en términos simples como *a)* salir de ello bastante bien, *b)* utilizar un arreglo rápido, *c)* hacer una mejora de simple a moderada del sistema existente, *d)* rediseñar el sistema existente o *e)* diseñar un nuevo sistema. Los participantes listados arriba para esta tarea son los adecuados para un análisis de alto nivel de estas opciones.

El marco de referencia PIECES que se presentó en el capítulo 3 puede utilizarse como marco de referencia para clasificar problemas, oportunidades, directrices y restricciones. Por ejemplo, el problema 1 en la figura 4.8 podría clasificarse de acuerdo con PIECES como P.B.: Desempeño, tiempos de respuesta. (Véase la figura 3.4 en el capítulo 3.) El problema 4 en la figura 4.8 podría clasificarse como I.A.2: Información, Salidas, Falta de información necesaria.

Las técnicas básicas utilizadas para completar esta tarea incluyen búsqueda de datos y reuniones con los PROPIETARIOS DEL SISTEMA. Estas técnicas se enseñan en el capítulo 5.

### > Tarea 1.2: Negociar el alcance base

El alcance define los límites del proyecto; aquellos aspectos del negocio que serán incluidos en el proyecto y los que no. El alcance puede cambiar durante el proyecto; sin embargo, el plan de proyecto inicial debe establecer el alcance preliminar o base. En consecuencia, si el alcance cambia en forma significativa, todas las partes tendrán una mejor apreciación de por qué cambian también el presupuesto y el programa. Esta tarea puede ocurrir en paralelo con la tarea anterior.

Una vez más, un analista de sistemas experto o un administrador de proyecto por lo general dirigen esta tarea. La mayor parte del resto de los participantes se clasifican de manera colectiva como PROPIETARIOS DEL SISTEMA. Esto incluye al patrocinador ejecutivo, los gerentes de todas las unidades organizacionales que pueden tener impactos por el sistema y tal vez los gerentes de sistemas de información. LOS USUARIOS DE SISTEMAS, DISEÑADORES DE SISTEMAS Y CONSTRUCTORES DE SISTEMAS en general no participan en esta tarea.

Como se muestra en la figura 4.6, esta tarea utiliza la DEFINICIÓN DE PROBLEMA PRELIMINAR producida por la tarea anterior. Debe tener sentido que esos problemas, oportunidades y directrices forman la base para definir el alcance. Las DEFINICIONES DEL ALCANCE DEL PROYECTO se suman al repositorio para su uso posterior. Estas definiciones también se documentan de manera formal como el producto de la tarea, DEFINICIÓN DEL PROBLEMA PRELIMINAR CON ALCANCE.

El alcance puede definirse con facilidad dentro del contexto de los componentes de su sistema de información. Por ejemplo, el alcance de un proyecto puede describirse en términos de:

- ¿Qué tipos de DATOS describen al sistema que se estudia? Por ejemplo, un sistema de información de ventas puede requerir datos acerca de cosas como CLIENTES, PEDIDOS, PRODUCTOS y REPRESENTANTES DE VENTAS.
- ¿Qué PROCESOS de negocios se incluyen en el sistema que se estudia? Por ejemplo, un sistema de información de ventas puede incluir procesos de negocios para ADMINISTRACIÓN DE CATÁLOGO, ADMINISTRACIÓN DE CLIENTES, INGRESO DE PEDIDOS, SATISFACCIÓN DE PEDIDOS, ADMINISTRACIÓN DE PEDIDOS y ADMINISTRACIÓN DE LA RELACIÓN CON LOS CLIENTES.
- ¿Cómo debe ser la INTERFAZ del sistema con los usuarios, locaciones y los demás sistemas? Por ejemplo, las interfaces potenciales de un sistema de información de ventas podría incluir CLIENTES, REPRESENTANTES DE VENTAS, EMPLEADOS Y GERENTES DE VENTAS, OFICINAS REGIONALES DE VENTAS y LOS SISTEMAS DE CUENTAS POR COBRAR y SISTEMAS DE INFORMACIÓN DE CONTROL DE INVENTARIOS.

Nótese que cada definición de alcance puede ser descrita como una simple lista. No “definimos” de manera necesaria los productos en la lista. Ni tampoco estamos muy preocupados por un análisis de requerimientos preciso. Y en definitiva, no nos interesan los pasos que toman mucho tiempo como la elaboración de modelos o de prototipos.

Una vez más, las técnicas primarias utilizadas para completar esta tarea son la identificación de hechos y las reuniones. Muchos analistas prefieren combinar esta tarea con la anterior y la siguiente para realizarlas dentro de una sola reunión.

### > Tarea 1.3: Evaluar el beneficio del proyecto base

Aquí es donde respondemos la pregunta: “¿vale la pena trabajar en este proyecto?”. En esta etapa inicial del proyecto, la pregunta en realidad podría reducirse a hacer una “mejor suposición”. ¿Solucionar los problemas, explotar las oportunidades o satisfacer las directrices devolverá el valor suficiente para superar los costos en los que incurriremos para desarrollar este sistema? Es imposible hacer un adecuado análisis de factibilidad con base en los hechos limitados que hemos reunido hasta el momento.

Una vez más, un analista de sistemas experto o un administrador de proyecto, por lo general dirigen esta tarea. Pero los PROPIETARIOS DEL SISTEMA, incluidos el patrocinador ejecutivo, los gerentes de unidades del negocio y los del sistema de información, deben tomar la decisión.

Como se muestra en la figura 4.6, la DEFINICIÓN PRELIMINAR DEL PROBLEMA CON EL ALCANCE dispara la tarea. Esto proporciona el nivel de información requerido para esta evaluación preliminar del valor. No hay un producto físico distinto a la decisión de SIGUE O NO SIGUE. En realidad hay varias decisiones alternativas. El proyecto puede aprobarse o cancelarse y el alcance del proyecto puede renegociarse (aumentar o disminuir). Es evidente que las tareas restantes en la fase de investigación preliminar son necesarias sólo si el proyecto ha sido considerado valioso y si tiene la aprobación para continuar.

### > Tarea 1.4: Desarrollar un programa y presupuesto base

Si el proyecto se ha considerado como valioso para continuar, ahora podemos planear la extensión del mismo. El proyecto inicial debe consistir al menos de lo siguiente:

- Un plan maestro preliminar que incluya programa y asignación de recursos para el proyecto completo. Este plan será actualizado al final de cada fase del proyecto. A veces se llama *plan base (baseline plan)*.
- Un plan y programa detallado para completar la siguiente fase del proyecto (la de análisis del problema).

La tarea es responsabilidad del *administrador del proyecto*. A la mayor parte de los administradores de proyectos les resulta útil incluir lo más posible del equipo del proyecto, incluidos los PROPIETARIOS DE SISTEMAS, USUARIOS, DISEÑADORES y CONSTRUCTORES. Como se mostró en la figura 4.6, esta tarea se dispara por la decisión de SIGUE O NO SIGUE para continuar el proyecto. Esta decisión representa un acuerdo consensuado acerca del alcance, los problemas, las oportunidades, las directrices y el valor del proyecto. (Este “valor” debe aún ser presentado y aprobado.) Las DEFINICIONES (ACOTADAS) DEL PROBLEMA SON la entrada fundamental del repositorio. El producto de esta tarea es EL PLAN Y PROGRAMA DEL PROYECTO BASE. La DEFINICIÓN DEL TRABAJO, el PROGRAMA DEL PROYECTO Y ASIGNACIÓN DE RECURSOS también se agregan al repositorio para una vigilancia continua y actualización, según resulte apropiado. El programa y los recursos se mantienen en forma normal en el repositorio como un archivo de software administrativo.

En la actualidad, las técnicas utilizadas para crear un plan de proyecto son soportadas por software de administración de proyectos como Microsoft *Project*.\*

### > Tarea 1.5: Comunicar el plan del proyecto

En la mayoría de las organizaciones, hay más proyectos potenciales que recursos para dar fondos y personal a esos proyectos. A menos que nuestro proyecto haya sido predefinido a ser de la más alta prioridad (por algún tipo de decisión táctica o proceso de planeación estratégica), debe ser presentado y defendido frente a un **comité de dirección** para su aprobación. La mayoría de las organizaciones utilizan un comité de dirección para aprobar y vigilar los proyectos y su progreso. La mayor parte de los integrantes de cualquier comité de dirección debe estar constituida por ejecutivos o administradores que no formen parte del área de los sistemas de información. Muchas organizaciones designan

---

**comité de dirección** Un comité de gerentes ejecutivos de negocios y sistemas que estudia y jerarquiza propuestas de proyectos que compiten entre sí, con el fin de determinar cuáles generarán más valor para la organización, de las cuales, algunas se aprobarán para que continúe el desarrollo de sistemas. También llamado *comité de dirección*.

\*Nota del R.T.: Al lector interesado en este tema se le recomienda consultar el PMBOK del PMI.

vicepresidentes para formar parte del comité de dirección. Otras organizaciones asignan el personal que reporta en forma directa a los vicepresidentes del comité de dirección. Algunas organizaciones utilizan dos comités de dirección, uno para vicepresidentes y uno para su personal directo. Los gerentes de sistemas de información trabajan en el comité de dirección sólo para responder preguntas y para comunicar prioridades de vuelta a los desarrolladores y a los administradores de proyecto.

Sin importar si un proyecto requiere o no de la aprobación de un comité de dirección, es de igual importancia lanzar y comunicar de manera formal el proyecto, las metas y el programa a la comunidad entera del negocio. Abrir las líneas de la comunicación es un paso importante para la investigación preliminar. Por esta razón, recomendamos las “mejores prácticas” para conducir un *evento de proyecto de salida (kickoff)* y crear un *sitio Web de proyecto de intranet*. La reunión de salida del proyecto está abierta a la comunidad entera de negocios, y no sólo a las unidades de negocios afectadas y al equipo de proyecto. El sitio Web de proyecto de intranet establece un portal de comunidad para todas las noticias no sensibles y la documentación relacionada con el proyecto.

De manera ideal, el patrocinador ejecutivo debe facilitar en forma conjunta la tarea con el administrador de proyecto elegido. La visibilidad del patrocinador ejecutivo establece una credibilidad instantánea y una prioridad para todos los que participan en la reunión de salida. Otros participantes en la reunión de salida deben incluir el equipo de proyecto completo, incluidos los PROPIETARIOS, USUARIOS, ANALISTAS, DISEÑADORES Y CONSTRUCTORES DE SISTEMAS asignados. Es preferible que la reunión de salida esté abierta a cualquier miembro del personal interesado de la comunidad de negocios. Esto crea conocimiento y hace un consenso entre los involucrados al tiempo que reduce tanto el volumen como las consecuencias del rumor y de la mala información. Para el componente de intranet, un Webmaster o un autor Web debe ser asignado al equipo de proyecto.

Como se muestra en la figura 4.6, esta tarea se dispara por el cumplimiento del PLAN Y PROGRAMA DEL PROYECTO BASE. Las DEFINICIONES DE PROBLEMA y el ALCANCE están disponibles en el repositorio. El producto es el ESQUEMA DEL PROYECTO y éste, por lo general, es un documento. Incluye diversos elementos que definen el proyecto en términos de participantes, problemas, oportunidades y directrices; alcance, metodología, definición del trabajo que se debe completar; productos, estándares de calidad, programa y presupuesto. El esquema del proyecto debe agregarse al sitio Web del proyecto para que todos lo veamos. Los elementos del esquema del proyecto también pueden ser presentados como diapositivas y boletines (por medio de software como *PowerPoint* de Microsoft) para incluirlos en el suceso de inicio del proyecto.

Las habilidades interpersonales y la comunicación son las claves para esta tarea. Éstas incluyen principios de persuasión, compra-venta, creación de contratos y hablar en público.

Con esto terminamos nuestro análisis de la fase de definición de alcance. En esta fase, los participantes podrían decidir que el proyecto no vale la pena para proponerse. También es posible que el comité de dirección pueda decidir que otros proyectos son más importantes. O el patrocinador ejecutivo podría no avalar el proyecto. En cada uno de estos casos, el proyecto se termina. Poco tiempo y esfuerzo se han utilizado. Por otro lado, con la bendición de todos los propietarios de sistemas y del comité de dirección, el proyecto podría continuar a la fase de análisis del problema.

## Fase de análisis del problema

Hay un viejo dicho que dice: “no trates de arreglarlo si no lo entiendes”. Esta declaración describe en forma adecuada la *fase de análisis del problema* del análisis de sistemas. Siempre hay un sistema actual o existente, sin importar a qué grado esté automatizado con tecnología de información. La fase de análisis del problema proporciona al analista una comprensión más completa de los problemas, oportunidades y/o directrices que dispararon el proyecto. La fase de análisis del problema responde a las preguntas, “¿vale la pena resolver estos problemas?” y “¿vale la pena construir un nuevo sistema?”. En otras metodologías, la fase de análisis de problemas puede ser conocida como *fase de estudio*, *estudio del sistema actual*, *fase de investigación detallada* o *fase de análisis de factibilidad*.

¿Alguna vez se puede pasar por alto la fase de análisis del problema? ¡Rara vez! Casi siempre se necesita un nivel de comprensión del sistema actual. Pero puede haber razo-

nes para acelerar la fase del análisis del problema. Primero, si el proyecto fue disparado por un plan estratégico o táctico, es probable que el valor del proyecto no esté en duda, la fase del análisis de problema se reduciría a una comprensión del sistema actual, no a analizarlo. Segundo, un proyecto puede ser iniciado por una directriz (tal como el cumplimiento con una instrucción y vencimiento gubernamental). De nuevo, en este caso el valor del proyecto no está en duda. Por último, algunas metodologías y organizaciones de manera deliberada consolidan el análisis del problema y las fases de análisis de requerimientos para acelerar el análisis de sistemas.

La meta de la fase del análisis del problema es estudiar y comprender el dominio del problema lo bastante bien para analizar a fondo sus problemas, oportunidades y restricciones. Algunas metodologías definen una comprensión muy detallada del sistema actual y lo documentan con sumo detalle mediante modelos de sistemas como diagramas de flujo de datos. En la actualidad, excepto cuando los procesos de negocios deben ser rediseñados, el esfuerzo requerido y el valor agregado por dichos modelos detallados se cuestionan y por lo general se ignoran. Por tanto, la versión actual de nuestra metodología hipotética *FAST* define un modelado de sistemas suficientemente amplio para refinar nuestra comprensión del alcance del proyecto, la definición del problema y la definición de un vocabulario común para el sistema.

El contexto para la fase del análisis del problema está sombreado en la figura 4.9. Nótese que la fase del análisis del problema trata en su mayor parte con los puntos de vista de los PROPIETARIOS y USUARIOS DEL SISTEMA acerca del sistema existente. Nótese que construimos sobre las listas creadas en la fase de investigación preliminar para analizar los componentes del CONOCIMIENTO, PROCESO y COMUNICACIONES del sistema existente. También nótese que determinamos la elaboración del mínimo modelo de sistema. Podemos aún utilizar el marco de referencia PIECES para analizar problemas, causas y efectos en cada componente del sistema.

La figura 4.10 es el diagrama de tareas para la fase de análisis de problemas. El producto e hito de la fase final es producir OBJETIVOS DE MEJORA DE SISTEMAS que aborden los problemas, oportunidades y directrices. Según se conozca el tamaño del sistema, su complejidad y el grado en que el proyecto es valioso, las tareas ilustradas pueden consumir de una a seis semanas. La mayoría de estas tareas pueden ser aceleradas por sesiones estilo JRP. La fase de análisis de problemas por lo general incluye las siguientes tareas:

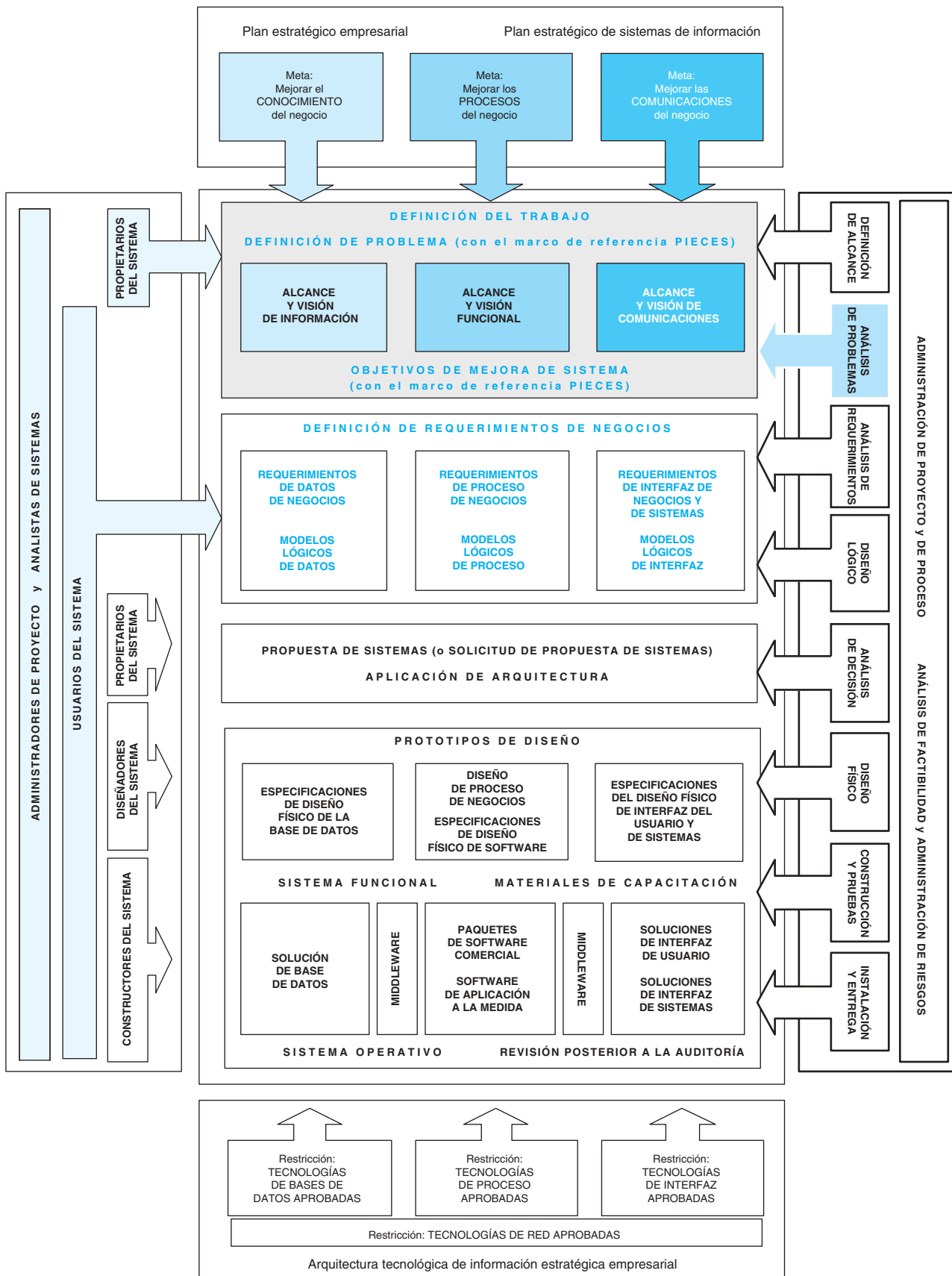
- 2.1 Entender el dominio del problema.
- 2.2 Analizar problemas y oportunidades.
- 2.3 Analizar procesos de negocios.
- 2.4 Establecer objetivos de mejora del sistema.
- 2.5 Actualizar o refinar el plan del proyecto.
- 2.6 Comunicar resultados y recomendaciones.

Ahora analicemos cada una de estas tareas con mayor detalle.

### > Tarea 2.1: Entender el dominio del problema

Durante la fase de análisis de problema, el *equipo* al inicio intenta conocer sobre el sistema actual. Cada PROPIETARIO, USUARIO y ANALISTA DE SISTEMAS aporta un nivel diferente de comprensión al sistema —detalle, vocabulario, percepciones y opiniones diferentes—. Un estudio bien conducido puede demostrar ser revelador para todas las partes, incluida la administración y los usuarios propios del sistema. Es importante estudiar y *entender el dominio del problema*, ese dominio en el que existen los problemas, oportunidades, directrices y restricciones del negocio.

Esta tarea será dirigida por el administrador del proyecto pero será facilitada por el analista de sistemas. Es común que un individuo desempeñe ambos papeles (como lo hace Sandra en el caso de SoundStage). Otros ANALISTAS DE SISTEMAS pueden también participar ya que realizan entrevistas, participan en juntas y documentan los resultados. Un estudio completo debe incluir PROPIETARIOS y USUARIOS DE SISTEMAS representativos de todas las unidades de negocios que serán apoyados o impactados por el sistema y el proyecto. Es muy importante que se incluyan suficientes usuarios para abarcar el alcance total del sistema que se estudia. En algunas organizaciones, uno o más usuarios experimentados son “prestados” al proyecto de tiempo completo como *analistas de sistemas*; sin embargo,



**FIGURA 4.9** Contexto de la fase de análisis de problemas del análisis de sistemas

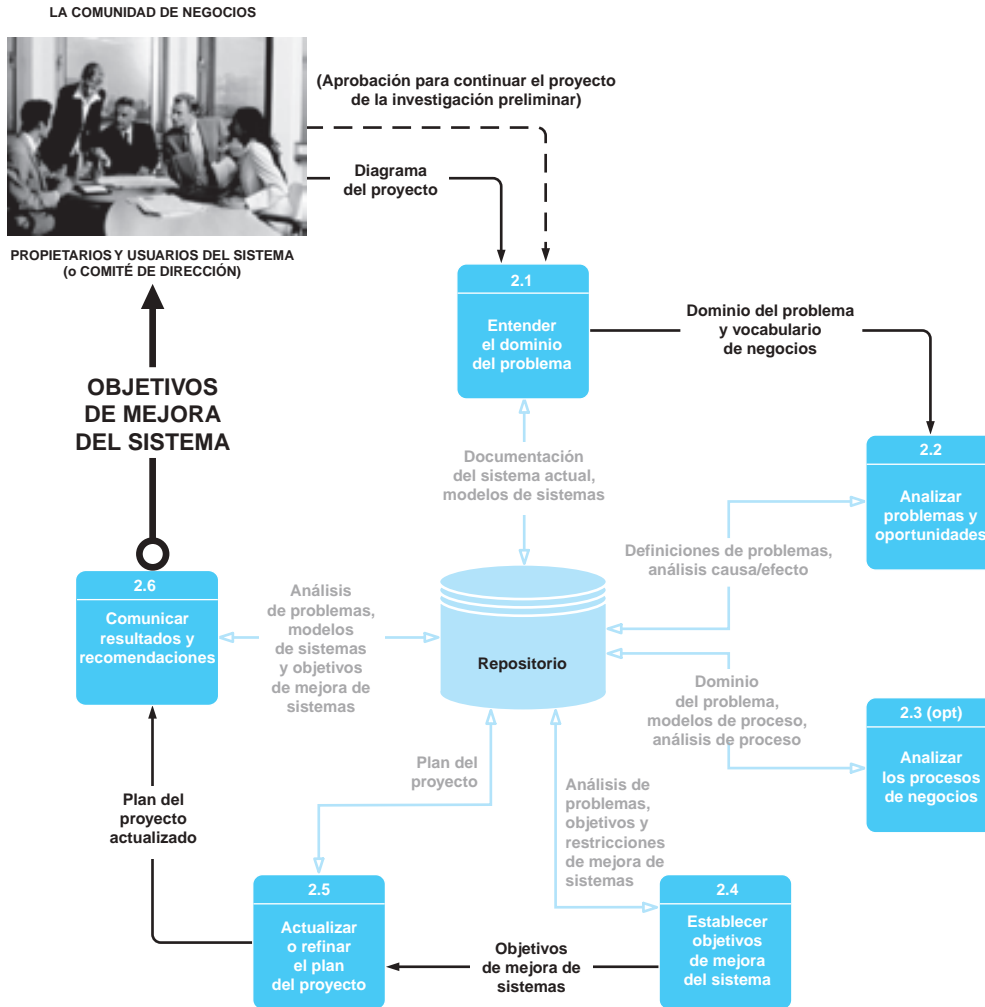


FIGURA 4.10

Tareas para la fase de análisis de problemas del análisis de sistemas

es raro que cualquier usuario pueda representar en su totalidad los intereses de todos los usuarios. Sin embargo, los analistas de sistemas pueden fungir como facilitadores para hacer que participen las personas correctas y sostener una comunicación eficaz de vuelta con las unidades del negocio y la administración. Los DISEÑADORES y LOS CONSTRUCTORES DE SISTEMAS rara vez participan en esta tarea a menos que sean entrevistados para determinar cualquier limitación técnica del sistema actual.

En la figura 4.10 esta tarea se dispara por APROBACIÓN PARA CONTINUAR EL PROYECTO; desde la fase de definición de alcance. (La línea punteada indica que esta aprobación es un suceso o disparador, no un flujo de datos o información.) La aprobación viene de los PROPIETARIOS DEL SISTEMA o de un comité de dirección. La entrada de información fundamental viene de los PROPIETARIOS DEL SISTEMA y cualquier DOCUMENTACIÓN DE SISTEMA ACTUAL que pueda existir en el repositorio y en las bibliotecas del programa para el sistema actual. La documentación del sistema actual no siempre existe. Y cuando existe, debe ser revisada con sumo cuidado para ver su actualidad; es notorio que la mayor parte de dicha documentación es obsoleta debido a que los programadores y analistas no siempre son diligentes para actualizarla cuando ocurren cambios a lo largo de la vida de un sistema.

Los productos de esta tarea son una comprensión del DOMINIO DEL PROBLEMA y del VOCABULARIO DE NEGOCIOS. Su comprensión del dominio del problema existente debe ser documentada para que pueda verificarse que se entiende bien. Hay diversas formas de documentar el dominio del problema. Es cierto, dibujar MODELOS DEL SISTEMA del que hay en la actualidad puede ayudar, pero eso puede llevar a un fenómeno llamado “parálisis de análisis” en el que el deseo de producir modelos perfectos se vuelve contraproducente para el cronograma. Otro

método podría ser utilizar los componentes de su sistema de información como marco de referencia para listar y definir el dominio del sistema:

- **CONOCIMIENTO.** Liste todas las “cosas” acerca de las cuales el sistema en la actualidad almacena datos (en archivos, bases de datos, formatos, etcétera). Defina cada cosa en términos de negocios. Por ejemplo, “un PEDIDO es una transacción de negocios en la que un cliente solicita adquirir productos”.  
Además, podríamos listar todos los informes producidos por el sistema actual y describir su propósito o uso. Por ejemplo, “el informe de pedidos abiertos describe todos los pedidos que no han sido surtidos una semana después de haber sido aprobados. El informe se utiliza para iniciar una administración de la relación con los clientes a través del contacto personal”.
- **PROCESOS.** Defina cada suceso de negocios para el cual una respuesta de negocios (proceso) es implementada en la actualidad. Por ejemplo, “un cliente coloca un nuevo pedido” o “un cliente solicita cambios a un pedido ya realizado” o “un cliente cancela un pedido”.
- **COMUNICACIONES.** Defina todas las ubicaciones que el sistema actual atiende y todos los usuarios en cada una de esas locaciones. Por ejemplo, “el sistema es utilizado en la actualidad en las oficinas de ventas regionales en San Diego, Dallas, St. Louis, Indianapolis, Atlanta y Manhattan. Cada oficina de ventas regionales tiene un gerente de ventas, un asistente del gerente de ventas, asistente administrativo y de 5 a 10 empleados de ventas, todos los cuales utilizan el sistema actual. Cada región también es la base de entre 5 y 30 representantes de ventas que viajan casi todos los días pero que cargan pedidos y otras transacciones todas las tardes”.  
Otra faceta de las interfaces son las interfaces del sistema; es decir, aquellas que existen entre el sistema de información actual y otros sistemas de información y aplicaciones de cómputo. Éstas pueden ser listadas de manera rápida y ser descritas por el personal de sistemas de información.

Por último, la metodología de desarrollo de sistemas de la organización y el plan de proyecto determinará qué tipos y nivel de documentación son los esperados.

El producto de vocabulario de negocios se abrevia con demasiada frecuencia. Entender el vocabulario de negocios del sistema es una forma excelente de comprender el sistema mismo. Construye puentes para salvar la brecha de comunicación que a menudo existe o se desarrolla entre los expertos de negocios y los de tecnología.

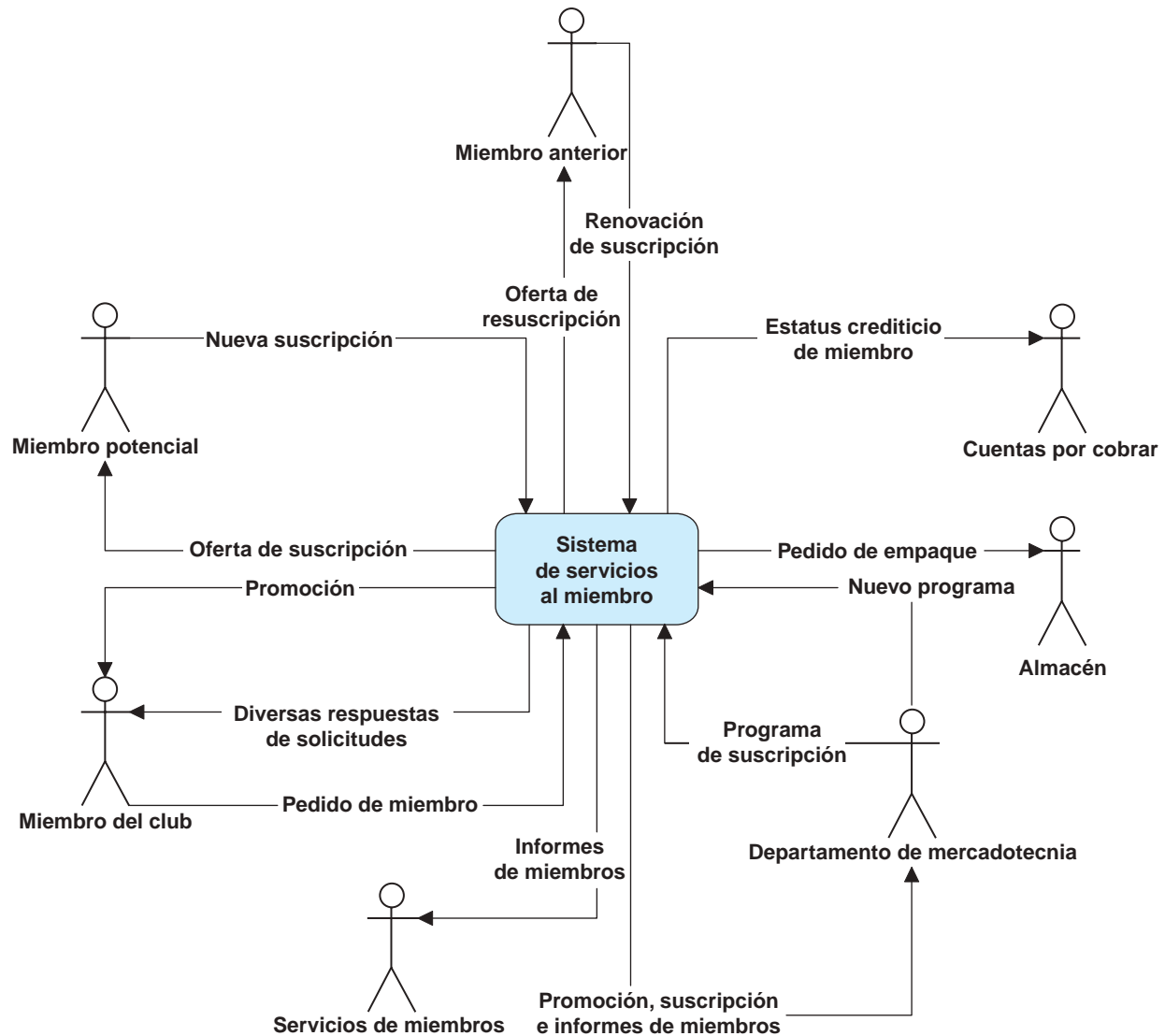
Si elige realizar **MODELOS DE SISTEMAS** durante esta tarea, sugerimos que si “desea aprender *algo*, no debe tratar de aprender *todo*; al menos no en esta tarea”. Para evitar una parálisis de análisis, sugerimos que los modelos de sistemas siguientes pueden ser apropiados:

- **CONOCIMIENTO.** Un modelo de datos de una página es muy útil para establecer vocabulario y reglas de negocios. La elaboración de modelos de datos se enseña en el capítulo 7.
- **PROCESOS.** En la actualidad, es muy aceptado que un diagrama de descomposición funcional de una o dos páginas debe ser suficiente para tener la idea del procesamiento del sistema actual. La elaboración de modelos de descomposición se enseña en el capítulo 8.
- **COMUNICACIONES.** Un diagrama de contexto de una página o los diagramas de caso de uso son muy útiles para ilustrar las entradas y salidas de un sistema con otras organizaciones, unidades de negocios y sistemas. Los diagramas de contexto se analizan más adelante. Los diagramas de casos de uso se enseñan en el capítulo 6.

Otras técnicas y habilidades son útiles para desarrollar una comprensión de un sistema existente. Es evidente que, las técnicas de búsqueda de resultados (enseñadas en el siguiente capítulo) son críticas para aprender acerca de cualquier sistema existente. También, las técnicas de planeación conjunta de requerimientos o JRP (enseñadas en el siguiente capítulo) pueden acelerar esta tarea. Por último, la capacidad de comunicar en forma clara a los usuarios lo que ha aprendido acerca de un sistema también es crucial.

**Diagrama de contexto** El propósito de un diagrama de contexto es analizar cómo el sistema interactúa con el mundo y especificar en términos generales las entradas y salidas del sistema. Los diagramas de contexto pueden dibujarse de diversas formas. En el capítulo 8 se presenta el formato tradicional, que fue hecho como el primer paso para dibujar diagramas de flujo de datos.





**FIGURA 4.11** Diagrama de contexto

En el capítulo 6 se muestra un formato diferente para un diagrama de contexto, el que está mostrado en la figura 4.11 utiliza un método híbrido. Utiliza símbolos de casos de uso ya que éstos se vuelven una herramienta por lo general aceptada de la fase de análisis de requerimientos.

El sistema mismo se muestra como una “caja negra” en medio del diagrama. No estamos listos para ver dentro de esa caja. Por ahora, sólo queremos ver cómo todos la utilizarán. Las figuras de líneas alrededor de la parte externa del diagrama son las personas, las organizaciones y los otros sistemas de información que interactuarán con el sistema. En los casos de uso, éstos se llaman actores y así podemos nombrarlos aquí. En los diagramas de flujo de datos tradicionales, se llaman agentes externos. En los capítulos 6 y 8 aprenderá que una vez que usted mira dentro de la caja del sistema, otros factores como el tiempo o aparatos como sensores pueden también ser actores o agentes externos. Pero en un diagrama de contexto rara vez se incluyen.

Las líneas indican las entradas (flechas que apuntan hacia el sistema) proporcionadas por los actores al sistema y las salidas (flechas que apuntan hacia los actores) creadas por el sistema. Cada entrada y salida se identifica con una frase propia que la describe.

Para construir un diagrama de contexto pregunte a los usuarios a qué transacciones de negocios debe responder el sistema; estas son las entradas. También pregunte a los

usuarios qué reportes, notificaciones y demás salidas deben ser producidas por el sistema. Un sistema puede tener muchos reportes o informes que pueden sobrecargar de manera rápida el diagrama; consolídelos como sea necesario para mantener legible el diagrama. Se analizarán en forma separada durante otras fases en el proceso.

Es cierto que no podríamos construir un sistema de información a partir de un diagrama de contexto. Pero es un primer paso sólido. Desde este diagrama simple, sabemos a qué entradas debe responder el sistema y qué salidas debe producir. En otras palabras, nos ayuda a entender el dominio del problema. Veremos en el capítulo 6 cómo detectar casos de uso a partir de un diagrama de contexto. Ese será el primer paso en abrir la “caja negra”. Seguimos los principios para el desarrollo de sistemas presentados en el capítulo 2: “utilice un método de solución de problemas” y “divida y vencerá”.

### > Tarea 2.2: Analizar problemas y oportunidades

Además de aprender acerca del sistema actual, el equipo del proyecto debe trabajar con los propietarios y los usuarios de sistemas para *analizar problemas y oportunidades*. Usted puede preguntarse, “¿no se identificaron con anterioridad los problemas y oportunidades, en la fase de investigación preliminar?” Sí, así fue. Pero esos problemas iniciales pueden ser sólo síntomas de otros, tal vez problemas no tan bien conocidos o entendidos por los usuarios. Además, en realidad no hemos analizado ninguno de esos problemas en el sentido clásico.

El análisis verdadero de problemas es una habilidad difícil de dominar, especialmente para los analistas de sistemas inexpertos. La experiencia sugiere que la mayoría de los analistas de sistemas (y muchos de los propietarios y usuarios de sistemas) tratan de solucionar problemas sin analizarlos de verdad. Ellos podrían definir un problema como este: “Necesitamos...” o “Queremos...”. Al hacer esto, definen el problema en términos de una solución. Los solucionadores de problemas más eficaces han aprendido a analizar el problema antes de afirmar una posible solución. Analizan cada problema percibido para sus **causas y efectos**. En la práctica, un efecto puede ser en realidad un síntoma de un problema arraigado en forma más profunda o más básica. Ese problema también debe ser analizado para encontrar sus causas y efectos y así sucesivamente hasta que llegue el momento en que las causas y los efectos no arrojen síntomas de otros problemas. El análisis de causa y efecto lleva a una verdadera comprensión de los problemas y puede conducir a soluciones más creativas y valiosas que no eran tan evidentes.

LOS ANALISTAS DE SISTEMAS facilitan esta tarea; sin embargo, todos los PROPIETARIOS y USUARIOS DE SISTEMAS deben participar de manera activa en el proceso del análisis de causa y efecto. Ellos son los expertos de dominio del problema. LOS DISEÑADORES y CONSTRUCTORES DE SISTEMAS por lo general no participan en este proceso a menos que sean llamados para analizar problemas técnicos que puedan existir en el sistema actual.

Como se muestra en la figura 4.10, la comprensión del equipo del DOMINIO DEL SISTEMA Y VOCABULARIO DE NEGOCIOS dispara esta tarea. Esta comprensión del dominio del problema es crucial debido a que los miembros del equipo no deben intentar analizar problemas a menos que entiendan el dominio en el que éstos ocurren. La otra entrada de información de esta tarea son las DEFINICIONES DEL PROBLEMA (de la fase de definición de alcance). Los productos de esta tarea son las DEFINICIONES DEL PROBLEMA (desde la fase de definición de alcance). Los productos de esta tarea son las DEFINICIONES DEL PROBLEMA actualizados y el ANÁLISIS DE CAUSA-EFECTO para cada problema y oportunidad. En la figura 4.12 se ilustra una forma de documentar un análisis de causa y efecto.

Una vez más, la identificación de hechos y las técnicas JRP son cruciales para esta tarea. Estas técnicas, así como el análisis de causa y efecto se enseñan en el siguiente capítulo.

### > Tarea 2.3: Analizar los procesos del negocio

Esta tarea es apropiada sólo para proyectos de *rediseño de procesos de negocios (BRP)* o proyectos de desarrollo de sistemas que se construyen sobre un significativo rediseño del proceso de negocios o que requieren de él. En dicho proyecto, al equipo se le pide examinar sus procesos de negocios en mucho mayor detalle para medir el valor agregado o sustraído por cada proceso en lo que se relacione con la organización total. El análisis de procesos de negocios puede estar influido por la política. Los propietarios y usuarios del

#### análisis de causa y efecto

Técnica en la que se estudian problemas para determinar sus causas y efectos.



## MATRIZ DE PROBLEMAS, OPORTUNIDADES, OBJETIVOS Y RESTRICCIONES

Proyecto:	Sistema de información de servicios a miembros	Administrador del proyecto:	Sandra Shepherd
Creado por:	Roberto Martínez	Actualizado la última vez por:	Roberto Martínez
Fecha de creación:	21 de enero de 2003	Actualizado por última vez el:	31 de enero de 2003
ANÁLISIS DE CAUSA Y EFECTO		OBJETIVOS DE MEJORA DEL SISTEMA	
Problema u oportunidad	Causas y efectos	Objetivo del sistema	Restricción de sistema
1. El tiempo de respuesta al pedido es inaceptable.	<ol style="list-style-type: none"> <li>La entrada de pedidos ha aumentado al tiempo que el número de empleados de pedidos ha disminuido. El tiempo para procesar un solo pedido se mantiene relativamente constante.</li> <li>El sistema es demasiado dependiente del teclado. Muchos de los mismos valores tienen clave para la mayoría de los pedidos. El resultado neto toma en cuenta (con el sistema actual) que cada pedido toma más tiempo del requerido para procesarse.</li> <li>La edición de datos es procesada por una AS/400. Mientras esa computadora va saturando su capacidad, las respuestas de edición de pedidos se van retrasando. Mientras los empleados tratan de trabajar más rápido para mantenerse al paso con el volumen de pedidos, el número de errores aumenta.</li> <li>El almacenamiento de tarjetas perforadas nunca fue diseñado para maximizar la eficiencia de la satisfacción de órdenes (pedidos). Cuando crecieron las operaciones del almacén, los retrasos en cuanto al llenado de pedidos fueron inevitables.</li> </ol>	<ol style="list-style-type: none"> <li>Disminuir el tiempo requerido para procesar un solo pedido en 30 por ciento.</li> <li>Eliminar la entrada de datos en el tablero en 50 por ciento para todos los pedidos.</li> <li>Para los pedidos pendientes, reducir en lo posible los golpes de teclado al reemplazarlos por objetos de señalar y dar clic en la pantalla de despliegue de la computadora.</li> <li>Mover la edición de datos de una computadora compartida al escritorio.</li> <li>Reemplazar las tarjetas perforadas existentes con un sistema de comunicación entre los servicios proporcionados y el almacén.</li> </ol>	<ol style="list-style-type: none"> <li>No habrá un aumento en la fuerza de trabajo de proceso de pedidos.</li> <li>Cualquier sistema desarrollado debe ser compatible con el estándar de escritorio Windows 95 existente.</li> <li>El nuevo sistema debe ser compatible con el sistema de identificación automático ya aprobado (para el código de barras).</li> </ol>

**FIGURA 4.12** Muestra de análisis de causa y efecto

sistema por igual pueden volverse muy defensivos acerca de sus procesos de negocios existentes. Los analistas que participan deben mantener el enfoque en los procesos y no en las personas que los realizan y recordar de manera constante a todos que la meta es identificar oportunidades para un cambio de negocios fundamental que beneficiará al negocio y a todos lo que están dentro de él.

Uno o más analistas de sistemas o de negocios facilitan la tarea. De manera ideal, los ANALISTAS deben ser experimentados, capacitados o certificados en los métodos BPR. Los únicos participantes restantes deben ser PROPIETARIOS y USUARIOS DE SISTEMAS. El análisis de proceso de negocios debe evitar cualquier tentación de enfocarse en soluciones de tecno-

logía de información hasta mucho después de que los procesos de negocios han sido rediseñados para una máxima eficiencia. Algunos analistas encuentran útil asumir la existencia de “personas perfectas” y “tecnología perfecta” que pueda hacer “posible” cualquier cosa. Preguntan, “si el mundo fuera perfecto, ¿necesitaríamos ese proceso?”

Como se describe en la figura 4.10, una tarea de análisis de proceso de negocios depende sólo de cierto conocimiento del DOMINIO DEL PROBLEMA (de la tarea 2.1). Los productos de esta tarea son MODELOS DE PROCESOS y ANÁLISIS DE PROCESOS “tal como son” del negocio. Los modelos de proceso pueden parecerse mucho a diagramas de flujo de datos (figura 4.2) excepto que se escriben en forma característica para mostrar 1) el volumen de flujo de datos a través de los procesos, 2) los tiempos de respuesta de cada proceso y 3) cualquier retraso o cuello de botella que ocurre en el sistema. Los datos del análisis de proceso proporcionan información adicional tal como *a)* el costo de cada proceso, *b)* el valor agregado de cada proceso y *c)* las consecuencias de eliminar u orientar el proceso. Con base en los modelos “tal como son” y sus análisis, el equipo desarrolla modelos “como serán” que rediseñan los procesos de negocios para eliminar la redundancia y la burocracia e incrementar la eficiencia y el servicio.

Diversas técnicas son aplicables a esta tarea. Una vez más, las técnicas de identificación de hechos y las reuniones del equipo (capítulo 5) son muy útiles. También, las técnicas de elaboración de modelos de procesos (capítulo 8) son críticas para el éxito del BPR.

### > Tarea 2.4: Establecer objetivos de mejora del sistema

Dada nuestra comprensión del alcance del sistema actual, los problemas y oportunidades, ahora podemos *establecer los objetivos de mejora del sistema*. El propósito de esta tarea es establecer el criterio en contra del cual cualquier mejora al sistema será medida y para identificar cualquier restricción que pueda limitar la flexibilidad en lograr esas mejoras. El criterio para el éxito debe ser medido en términos de **objetivos**. Los objetivos representan el primer intento por establecer expectativas para cualquier sistema nuevo. Además de identificar objetivos, debemos también identificar cualquier restricción conocida. Las **restricciones** colocan limitaciones o delimitaciones para lograr los objetivos. Los vencimientos, objetivos y tecnologías requeridas son ejemplos de restricciones.

LOS ANALISTAS DE SISTEMAS facilitan esta tarea. Otros participantes incluyen a los mismos PROPIETARIOS DE SISTEMAS y USUARIOS que han participado en otras tareas en esta fase de análisis de problemas. Nuevamente, no estamos aún preocupados por la tecnología; por tanto, los DISEÑADORES y LOS CONSTRUCTORES DE SISTEMAS no participan en esta tarea.

Esta tarea es disparada por el ANÁLISIS DE PROBLEMAS completado en las tareas 2.2 y 2.3. Para cada problema verificado y significativo, los analistas y los usuarios deben definir OBJETIVOS DE MEJORA DE SISTEMAS específicos. Deben también identificar cualquier RESTRICCIÓN que pueda limitarlos o evitar que logren los objetivos de mejora del sistema.

Los objetivos de mejora del sistema deben ser definiciones precisas y medibles del desempeño del negocio que definan las expectativas del nuevo sistema. Algunos ejemplos son:

- Reducir el número de cuentas incobrables de clientes en 50 por ciento dentro del siguiente año.
- Aumentar en 25 por ciento el número de solicitudes de préstamo que pueden ser procesadas en un turno de ocho horas.
- Disminuir en 50 por ciento el tiempo requerido para reprogramar un lote de producción cuando una estación de trabajo tiene un mal funcionamiento.

Lo siguiente es un ejemplo de un mal objetivo:

Crear un informe de cuentas morosas.

Éste es un mal objetivo porque afirma sólo un requerimiento y no un objetivo real. Ahora vamos a replantear ese objetivo:

Reducir pérdidas de crédito en 20 por ciento a través de la identificación temprana de cuentas morosas.

**objetivo** Una medición del éxito. Es algo que se espera lograr, si se tienen recursos suficientes.

**restricción** Algo que limita la flexibilidad en la definición de una solución según los objetivos que se tengan. En lo esencial, es imposible modificar las restricciones.

Esto nos brinda una mayor flexibilidad. Sí, el informe de cuentas morosas funcionaría. Pero una investigación de la morosidad de un cliente podría proporcionar una forma todavía mejor de lograr el mismo objetivo.

Los objetivos de mejora de sistemas pueden ser acotados por restricciones identificables. Las restricciones caen dentro de cuatro categorías, como se listan a continuación (con ejemplos):

- *Programa*: El nuevo sistema debe ser operativo para el 15 de abril.
- *Costo*: El nuevo sistema no puede costar más de 350 000 dólares.
- *Tecnología*: El nuevo sistema debe estar en línea o todos los nuevos sistemas deben utilizar el sistema de administración de base de datos DB2.
- *Política*: El nuevo sistema debe utilizar técnicas de inventario de doble declinación de balance.

Las últimas dos columnas de la figura 4.12 documentan los objetivos y restricciones de mejora de un sistema típico.

### > Tarea 2.5: Actualizar o refinar el plan del proyecto

Recuerde que el alcance del proyecto es un objetivo móvil. Con base en nuestro programa y presupuesto base de la fase de definición de alcance, el alcance puede haber crecido o disminuido en tamaño y complejidad. (¡El crecimiento es mucho más común!) Ahora que nos aproximamos a la terminación de la fase del análisis del problema, debemos reevaluar el alcance del proyecto y, en consecuencia, *actualizar o refinar el plan de proyecto*.

El administrador del proyecto, en conjunto con los PROPIETARIOS DEL SISTEMA y el equipo completo del proyecto, facilitan esta tarea. Los ANALISTAS DEL SISTEMA y LOS PROPIETARIOS DEL SISTEMA son los individuos fundamentales en esta tarea. Los analistas y los propietarios deben considerar la posibilidad de que no todos los objetivos pueden ser satisfechos por el nuevo sistema. ¿Por qué? El nuevo sistema puede ser más grande de lo esperado y puede requerirse reducir el alcance para cumplir con una fecha límite. En este caso, el propietario del sistema clasificará los objetivos en orden de importancia. Luego, si el alcance debe ser reducido, los objetivos de mayor prioridad le dirán al analista qué es lo más importante.

Como se muestra en la figura 4.10, esta tarea se dispara como resultado de la definición de los OBJETIVOS DE MEJORA DEL SISTEMA. El PLAN DE PROYECTO inicial es otra entrada fundamental y el PLAN DE PROYECTO ACTUALIZADO es la salida principal. El plan actualizado debe ahora incluir un plan detallado que se debe seguir en la fase de análisis de requerimientos.\*

### > Tarea 2.6: Comunicar resultados y propuestas

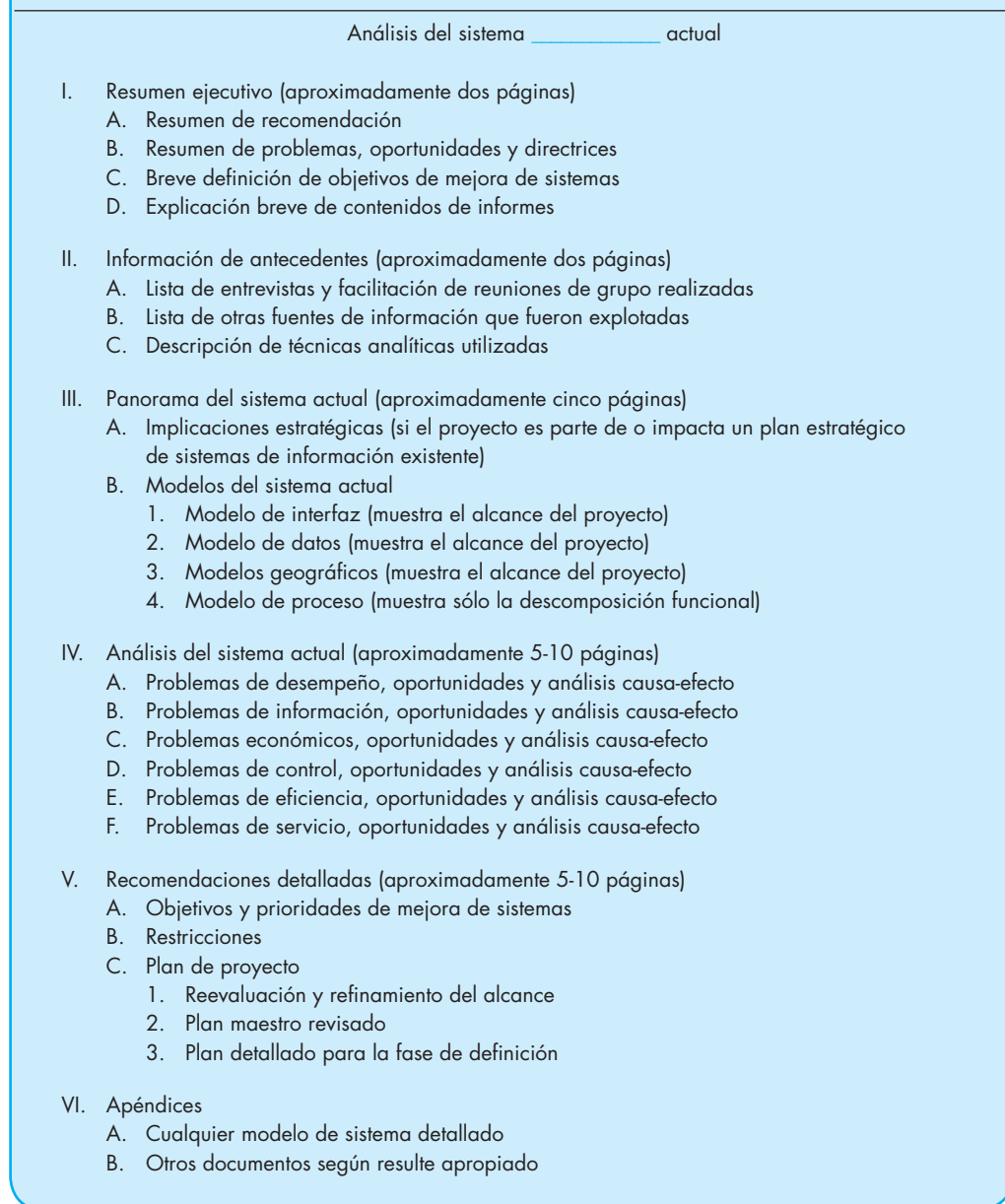
Al igual que con la fase de definición de alcance, la fase de análisis de problema concluye con una tarea de comunicación. Debemos *comunicar resultados y propuestas* a la comunidad del negocio. Otra reunión en la que los participantes deben incluir al equipo del proyecto completo, incluyendo PROPIETARIOS, USUARIOS, ANALISTAS, DISEÑADORES y CONSTRUCTORES DE SISTEMAS asignados. Y, como siempre, la reunión debe estar abierta a cualquier empleado interesado que forme parte de la comunidad del negocio. También, si se estableció un sitio web en la intranet para el proyecto, debe haberse mantenido a lo largo de la fase de análisis del problema para asegurar la comunicación continua del progreso del proyecto.

Esta tarea se dispara por la terminación del PLAN DEL PROYECTO ACTUALIZADO. Las entradas de información incluyen los ANÁLISIS DE PROBLEMAS, cualquier MODELO DE SISTEMAS, los OBJETIVOS DE MEJORA DE SISTEMAS, el principal producto de la fase de análisis de problemas. El formato puede ser un informe, una presentación verbal o una inspección por un auditor o grupo de compañeros (llamado *grupo de revisión*). En la figura 4.13 se muestra un esquema de un informe escrito.

Las habilidades interpersonales y de comunicaciones son básicas para esta tarea. Los analistas de sistemas deben tener la capacidad de escribir un informe de negocios formal y de hacer una presentación de negocios sin entrar en temas o alternativas técnicas.

\* Nota del R.T.: Las técnicas y los pasos para mejorar el plan de proyecto se enseñan en el PMBOK del PMI.

**FIGURA 4.13** Esquema para objetivos de mejora de sistemas e informe de recomendaciones



Esto concluye la fase del análisis del problema. Una de las siguientes decisiones debe realizarse luego de la conclusión de esta fase:

- Autorizar que el proyecto continúe, tal como está, a la fase de análisis de requerimientos.
- Ajustar el alcance, costo o programa para el proyecto y luego continuar con la fase de análisis de requerimientos.
- Cancelar el proyecto debido a: 1) falta de recursos para continuar el desarrollo del sistema, 2) darse cuenta de que los problemas y oportunidades simplemente no eran tan importantes como se había anticipado, o 3) comprensión de que no es probable que los beneficios del nuevo sistema excedan los costos.

Con cierto nivel de aprobación por parte de los PROPIETARIOS DEL SISTEMA, el proyecto puede ahora continuar hacia la fase de análisis de requerimientos.

## Fase de análisis de requerimientos

Muchos analistas inexpertos cometen un error crítico luego de completar la fase de análisis del problema. La tentación en ese punto es comenzar a considerar alternativas de solución, particularmente técnicas. Uno de los errores citados con mayor frecuencia en los nuevos sistemas de información se ilustra en la frase “seguro, el sistema funciona y técnicamente es impresionante, pero no hace lo que nosotros necesitamos”. La *fase de análisis de requerimientos* define los requerimientos de negocios para un sistema nuevo.

¿Notó la palabra clave en la oración citada? es “que” y no “cómo”! Los analistas con frecuencia están tan preocupados por la solución *técnica* que inadecuadamente definen los requerimientos de *negocios* para esa solución. La fase de análisis de requerimientos responde la pregunta, “¿Qué necesitan y desean los usuarios de un sistema nuevo?”

La fase de análisis de requerimientos es crítica para el éxito de cualquier sistema nuevo de información. En distintas metodologías la fase de análisis de requerimientos podría ser llamada *fase de definición* o *fase de diseño lógico*.

¿Se puede alguna vez pasar por alto la fase de análisis de requerimientos? ¡Definitivamente no! Los nuevos sistemas siempre serán evaluados, primero y sobre todo, acerca de si cumplen o no con los objetivos y requerimientos del negocio, ¡sin importar qué tan impresionante o compleja pueda ser la solución técnica!

Debe reconocerse que algunas metodologías integran las fases de análisis del problema y análisis de requerimientos en una sola fase.

Una vez más, sus componentes de sistemas de información (figura 4.14) pueden servir como un marco de referencia útil para documentar los requerimientos de sistemas de información. Nótese que seguimos ocupados con las perspectivas de los USUARIOS DEL SISTEMA. Los requerimientos pueden ser definidos en términos del marco laboral PIECES o en términos de los tipos de datos, procesos e interfaces que deben ser incluidos en el mismo.

En la figura 4.15 se ilustran las tareas típicas de la fase de análisis de requerimientos. El producto y la meta a alcanzar de la fase final permitirán crear una DEFINICIÓN DE LOS REQUERIMIENTOS DEL NEGOCIO que satisfaga los objetivos de mejora del sistema identificados en la fase anterior. Una de las primeras cosas que puede notar en este diagrama de tareas es que la mayoría de las tareas no se dan en forma secuencial como las de los diagramas de tareas anteriores. En lugar de eso, muchas de estas tareas ocurren en paralelo mientras el equipo trabaja hacia la meta de completar la definición de requerimientos. La fase de análisis de requerimientos generalmente incluye las siguientes tareas:

- 3.1 Identificar y expresar los requerimientos del sistema.
- 3.2 Priorizar los requerimientos de sistema.
- 3.3 Actualizar o refinar el plan de proyecto.
- 3.4 Comunicar la definición de requerimientos.

Ahora analicemos cada una de estas tareas con mayor detalle.

### > Tarea 3.1: Identificar y expresar los requerimientos del sistema

La tarea inicial de la fase de análisis de requerimientos es *identificar y expresar requerimientos*. Mientras que esto puede parecer como una tarea fácil o trivial, a menudo es la fuente de muchos errores, omisiones y conflictos. La base para esta tarea se estableció en la fase de análisis del problema cuando identificamos objetivos de mejora de sistemas. En forma mínima, esta tarea traduce estos objetivos en un esquema de **requerimientos funcionales y no funcionales** que se necesitarán para cumplir con los objetivos. Los requerimientos funcionales son frecuentemente identificados en términos de entradas, salidas, procesos y datos almacenados que son necesarios para satisfacer los objetivos de mejora del sistema. Ejemplos de requerimientos no funcionales incluyen desempeño (tiempo de desempeño y de respuesta); facilidad de aprendizaje y uso; presupuestos, costos y ahorros de costos; cronogramas y vencimientos; documentación y necesidades de capacitación; administración de la calidad y controles de auditoría interna y seguridad.

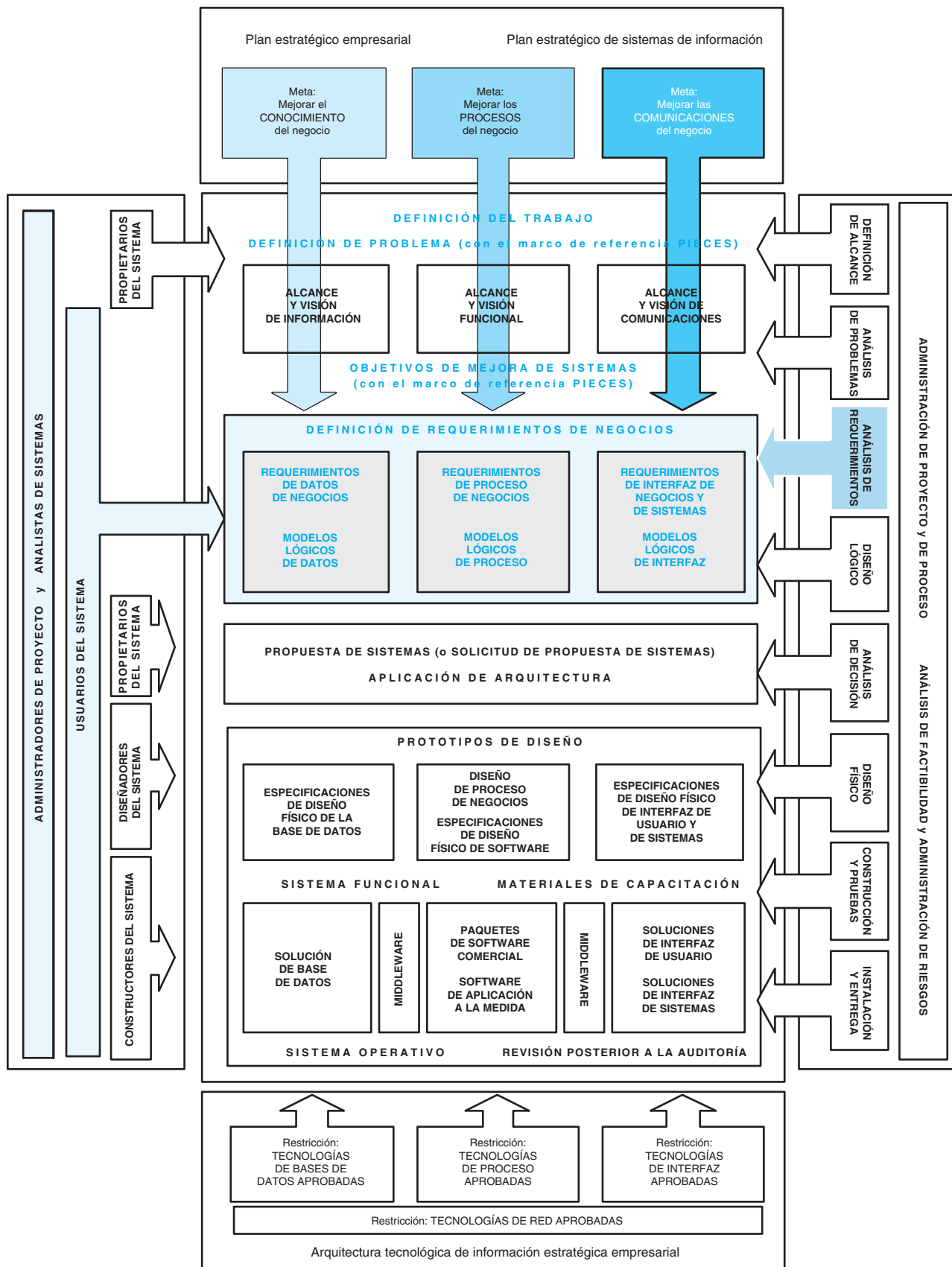
Rara vez esta tarea de definición identifica *todos* los requerimientos de negocios funcionales y no funcionales. Pero este esquema enmarcará su pensamiento mientras procede con tareas posteriores que agregarán nuevos requerimientos y detalles al mismo. Por tanto, ni la totalidad ni la perfección son una meta en esta tarea.

#### requerimiento funcional

Descripción de las actividades y servicios que debe brindar un sistema.

#### requerimiento no funcional

Descripción de otras características y restricciones que definen un sistema satisfactorio.



**FIGURA 4.14** Contexto de la fase de análisis de requerimientos del análisis de sistemas



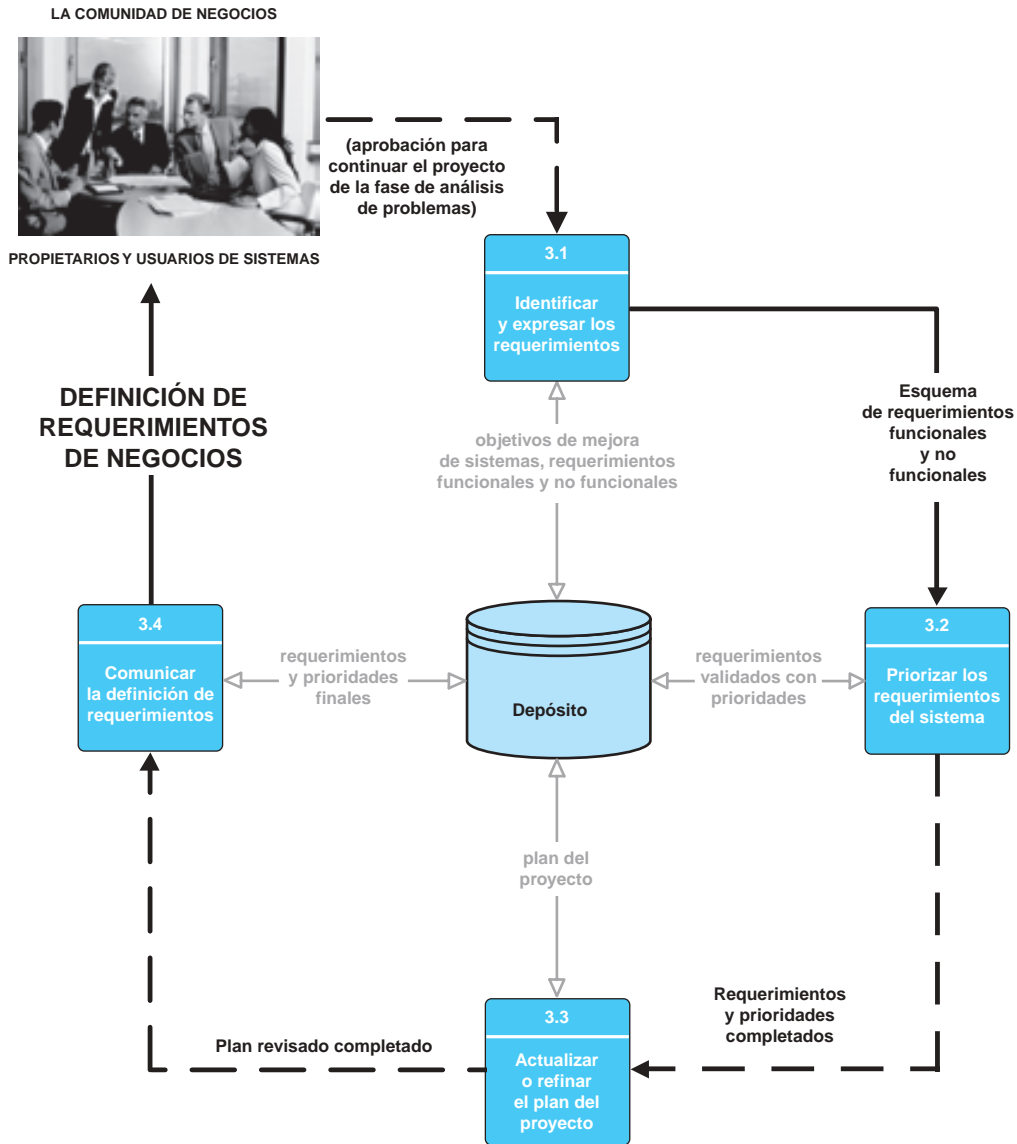


FIGURA 4.15

Tareas para la fase de análisis de requerimientos del análisis de sistemas

LOS ANALISTAS DEL SISTEMA facilitan la tarea. También documentan los resultados. Evidentemente, los USUARIOS DEL SISTEMA son la fuente principal de los requerimientos de negocios. Algunos PROPIETARIOS DEL SISTEMA pueden elegir participar en esta tarea ya que tuvieron un papel en la estructuración de los objetivos de mejora del sistema que guiarán la tarea. LOS DISEÑADORES y CONSTRUCTORES DE SISTEMAS no deben participar porque tienden a redirigir prematuramente el enfoque a la tecnología y a las soluciones técnicas.

Como se muestra en la figura 4.15, esta tarea (y fase) se dispara cuando se tiene la APROBACIÓN PARA CONTINUAR EL PROYECTO durante la FASE DE ANÁLISIS DEL PROBLEMA. La entrada fundamental son los OBJETIVOS DE MEJORA DEL SISTEMA definidos desde la fase de análisis del problema (vía repositorio). Desde luego, toda la información relevante obtenida durante la fase de análisis de problemas está disponible para su referencia según sea requerido.

El único producto a obtener en esta tarea son los REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES. Diversos formatos pueden funcionar. En su formato más simple, el esquema puede ser dividido en cuatro secciones lógicas: la lista original de los objetivos de mejora de sistemas, una sublista de *a)* entradas, *b)* procesos, *c)* salidas y *d)* datos almacenados necesarios para satisfacer el objetivo. Sin embargo, cada vez más, los analistas de sistemas expresan requerimientos funcionales mediante una herramienta de elaboración de modelos llamada **casos de uso**. Los casos de uso elaboran modelos de escenarios de negocios

**caso de uso** Escenario de negocios o evento respecto del cual el sistema debe proporcionar una respuesta definida. Los casos de uso evolucionaron a partir del análisis orientado a objetos; pero su utilización se ha vuelto común en muchos otros métodos de análisis y diseño de sistemas.

y sucesos que deben ser manejados por un sistema nuevo. Se presentan en el capítulo 6 y se utilizan a lo largo de este texto.

El marco de referencia PIECES que fue utilizado anteriormente para identificar problemas, oportunidades y restricciones también puede utilizarse como marco laboral para definir los requerimientos del borrador.

Diversas técnicas son aplicables a esta tarea. La planeación de requerimientos conjuntos (joint requirements planning, JRP) es la técnica preferida para delinear con rapidez los requerimientos del negocio. De manera alternativa, los analistas podrían utilizar otros métodos de identificación de hechos como encuestas y entrevistas. Tanto el JRP como la identificación de hechos se enseñan en el siguiente capítulo.

### > Tarea 3.2: Priorizar los requerimientos del sistema

Anteriormente comentamos que el éxito de un proyecto de desarrollo de sistemas puede ser medido en términos del grado en el que se satisfacen los requerimientos del negocio. Pero no todos los requerimientos se crean igual. Si un proyecto se retrasa en horario o sobre el presupuesto, puede ser útil para reconocer qué requerimientos son más importantes que otros. Por tanto, dados los requerimientos validados, los propietarios del sistema y los usuarios del sistema deben *priorizar los requerimientos del sistema*.

Otorgar prioridades a los requerimientos puede facilitarse por medio de una técnica popular llamada *timeboxing*. El *timeboxing* intenta dividir requerimientos en “partes” que puedan ser implementadas dentro de un periodo que no acabe con la paciencia del usuario y de la comunidad administrativa. El *timeboxing* obliga a que las prioridades sean definidas claramente.

LOS ANALISTAS DE SISTEMAS facilitan la tarea de clasificación de prioridades. LOS PROPIETARIOS Y USUARIOS DEL SISTEMA establecen las prioridades actuales. LOS DISEÑADORES y LOS CONSTRUCTORES de sistemas no participan en la tarea. La tarea es realizada cuando los REQUERIMIENTOS SON VALIDADOS. Debe resultar evidente que usted no pueda otorgar prioridades adecuadamente a un conjunto de requerimientos incompleto. El producto de esta tarea es obtener REQUERIMIENTOS CON PRIORIDADES. Las prioridades pueden ser clasificadas de acuerdo con su importancia relativa:

- Un *requerimiento obligatorio* es aquél que debe ser satisfecho por la mínima versión del sistema (versión 1.0.) El sistema es inútil sin él. ¡Tenga cuidado!, existe la tentación de etiquetar demasiados requerimientos como obligatorios. Un requerimiento obligatorio no puede ser clasificado porque es esencial para cualquier solución. De hecho, si un requerimiento obligatorio puede ser clasificado en orden de importancia, en realidad es un requerimiento deseable.
- Un *requerimiento deseable* es aquél que no es absolutamente esencial a la versión mínima del sistema (versión 1.0). Puede aún ser esencial para la visión de alguna versión futura. Los requerimientos deseables pueden y deben ser clasificados. El uso de números de versión como esquema de clasificación es una forma eficaz de comunicar y categorizar los requerimientos deseables.

### > Tarea 3.3: Actualizar o refinar el plan del proyecto

Nuevamente, recordemos que el alcance del proyecto es un objetivo móvil. Ahora que hemos identificado los requerimientos del sistema, debemos regresar y redefinir nuestra comprensión del alcance del proyecto y actualizar en consecuencia nuestro plan de proyecto. El equipo debe considerar la posibilidad de que el nuevo sistema pueda ser más grande de lo originalmente esperado. Si es así, en consecuencia, el equipo debe ajustar el programa, presupuesto o alcance. Debemos también asegurar la aprobación para que el proyecto continúe hacia la siguiente fase. (Puede que el trabajo ya se haya iniciado en las fases de diseño; sin embargo, las decisiones todavía requieren una revisión.)

El administrador del proyecto, en conjunto con los PROPIETARIOS DEL SISTEMA y el equipo de proyecto completo, facilitan esta tarea. Como siempre, el administrador del proyecto y los PROPIETARIOS DEL SISTEMA son los individuos clave en esta tarea. Deben considerar la posibilidad de que los requerimientos ahora excedan la visión original que se estableció para el proyecto y el nuevo sistema. Pueden tener que reducir el alcance para cumplir con un vencimiento o incrementar el presupuesto para que el trabajo se realice.

Como se muestra en la figura 4.15, esta tarea se realiza cuando los REQUERIMIENTOS SON PRIORIZADOS Y COMPLETADOS. EL PLAN DEL PROYECTO actualizado es la otra entrada clave y se actualiza en el repositorio cuando se considere apropiado.

---

**timeboxing** Técnica que entrega funcionalidad y requerimientos de sistemas de información mediante el control de versiones. El equipo de desarrollo selecciona el subconjunto más pequeño del sistema que al ser puesto en práctica por completo genera valor inmediato para los propietarios y usuarios del sistema. Se desarrolla ese subconjunto, de preferencia en seis a nueve meses o menos. En forma subsiguiente, se desarrollan versiones del sistema con valor añadido, en marcos cronológicos similares.



### > Tarea 3.4: Comunicar la definición de requerimientos

La comunicación es una tarea continua de la fase de análisis de requerimientos. A lo largo de la fase, debemos comunicar requerimientos y prioridades para la comunidad de negocios. Los usuarios y administradores con frecuencia abogarán por una consideración de requerimientos y prioridades. La comunicación es el proceso a través del cual se deben mediar las diferencias de opinión. El administrador del proyecto y el patrocinador ejecutivo conjuntamente deben facilitar esta tarea. Actualmente, una intranet o un portal de proyecto se utilizan con frecuencia para comunicar los requerimientos. Algunos sistemas permiten a los usuarios y administradores *acceder* a los documentos de requerimientos para asegurarse de ser notificados conforme ocurran los cambios. Las habilidades interpersonales, de comunicación y negociación son esenciales para esta tarea.

### > Manejo de requerimientos permanentes

La fase de análisis de requerimientos está ahora completa. ¿O no? Alguna vez fue popular congelar los requerimientos del negocio antes de empezar las fases de diseño de sistemas y las de construcción. Pero la economía actual se ha vuelto cada vez más acelerada. Los negocios se miden de acuerdo con su capacidad para adaptarse con rapidez a requerimientos y oportunidades constantemente cambiantes. Los sistemas de información no pueden ser menos cambiantes que el negocio mismo. Por tanto, el análisis de requerimientos realmente nunca termina. Mientras que hacemos una transición silenciosa hacia las fases restantes de nuestro proyecto, continúa una necesidad constante de manejar requerimientos a través del curso del proyecto y la vida del sistema.

El manejo de los requerimientos define un proceso para los propietarios, usuarios, analistas, diseñadores y constructores del sistema para enviar los cambios propuestos a los requerimientos de un sistema. El proceso especificó cómo se deben solicitar y documentar los cambios, cómo se registrarán y rastrearán y cuándo y cómo se evaluarán para saber su prioridad y la forma en que eventualmente serán satisfechos (si alguna vez sucede).

## Fase de diseño lógico

No todos los proyectos aceptan el desarrollo basado en modelos, pero la mayoría incluye un poco de elaboración de modelos de sistemas. Un diseño lógico documenta más aún los requerimientos de negocios por medio de los modelos de sistemas que ilustran las estructuras de datos, procesos de negocios, flujos de datos e interfaces de usuarios (cada vez más por medio de modelos de objetos, como se presentó anteriormente en el capítulo). En un sentido, validan los requerimientos establecidos en la fase anterior.

Una vez más, los componentes de sus sistemas de información (figura 4.16) pueden servir como un marco útil para documentar los requerimientos de sistemas de información. Nótese que todavía estamos ocupados por las perspectivas de los *USUARIOS DEL SISTEMA*. En esta fase, elaboramos diversos modelos de sistemas para documentar los requerimientos para un sistema nuevo y mejorado. Los modelos describen diversos aspectos de nuestros componentes. De manera alternativa, los prototipos podrían ser construidos para “identificar requerimientos”. Los prototipos de identificación fueron presentados anteriormente en el capítulo. Recuerde que a algunos prototipos se les puede aplicar una ingeniería inversa para convertirlos en modelos de sistemas.

En la figura 4.17 se ilustran las tareas típicas de la fase de diseño lógico. El producto y meta de la fase final es producir una *DEFINICIÓN DE REQUERIMIENTOS DEL NEGOCIO* que satisfagan los objetivos de mejora de sistemas identificados en la fase previa. Una de las primeras cosas que usted notará en este diagrama de tareas es que la mayoría de las tareas no son secuencias como en los diagramas de tareas anteriores. En lugar de eso, muchas de estas tareas ocurren en paralelo mientras el equipo trabaja hacia la meta de completar la definición de requerimientos.

La fase de diseño lógico generalmente incluye las siguientes tareas:

- 4.1a Requerimientos funcionales de estructura
- 4.1b Requerimientos funcionales del prototipo
- 4.2 Validar requerimientos funcionales
- 4.3 Definir casos de prueba de aceptación

Ahora analicemos cada una de estas tareas con mayor detalle.

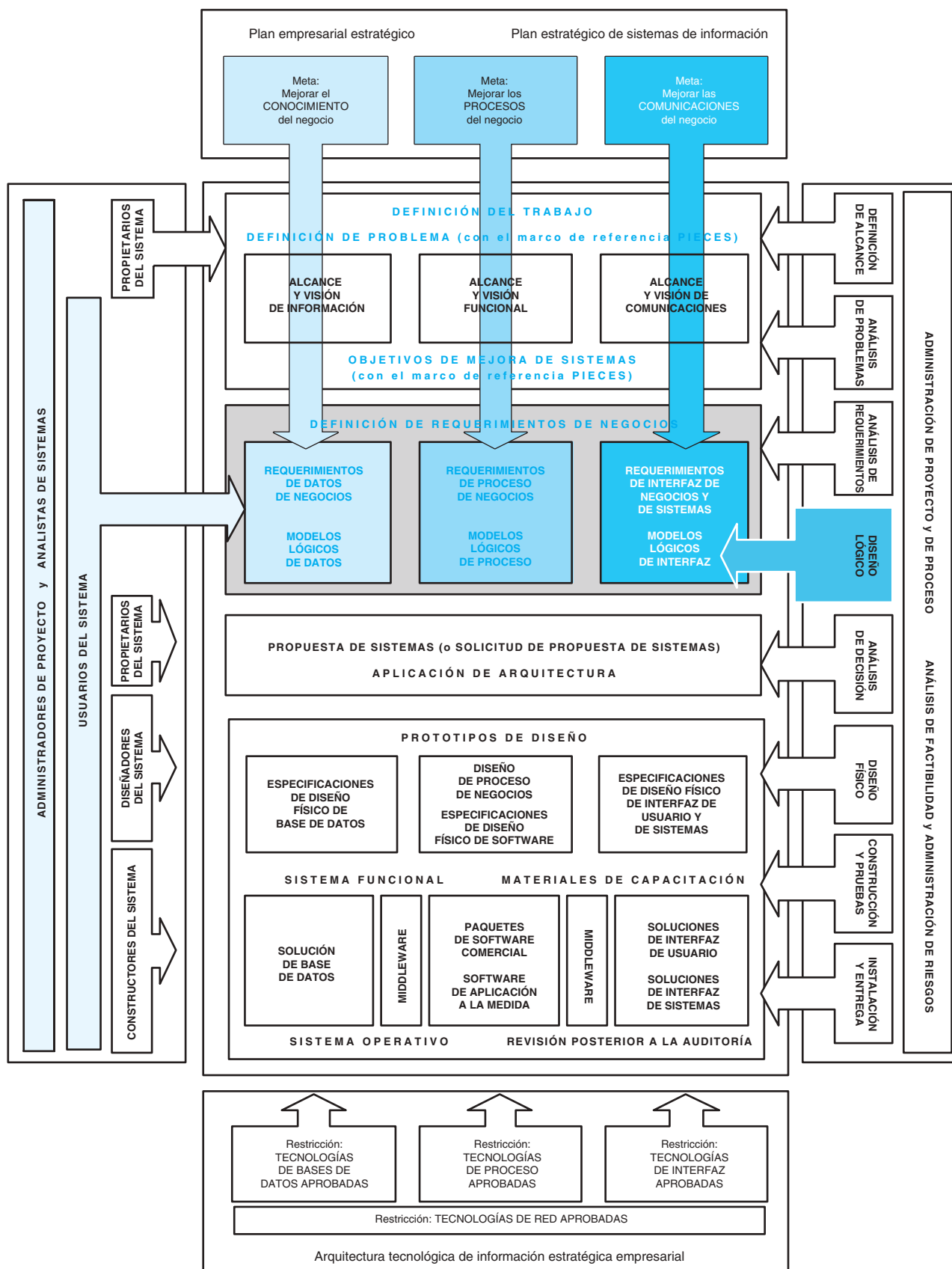


FIGURA 4.16 Contexto de la fase de diseño lógico del análisis de sistemas

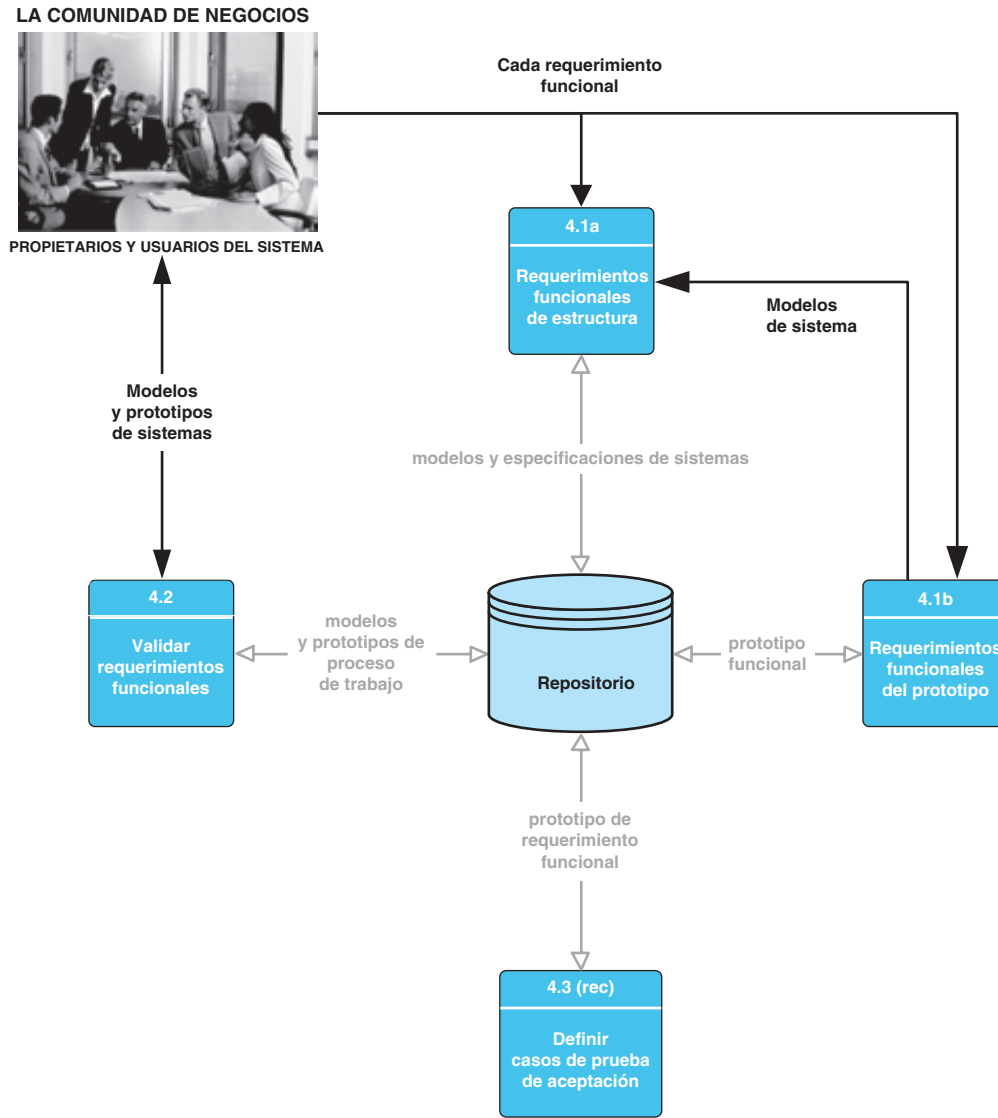


FIGURA 4.17

Tareas para la fase de diseño lógico del análisis de sistemas

### > Tarea 4.1a: Requerimientos funcionales de estructura

Un método para el diseño lógico es definir los *requerimientos funcionales de estructura*. Esto significa que, por medio de métodos acelerados, usted debe elaborar o actualizar uno o más modelos de sistemas para ilustrar el requerimiento funcional. Éstos pueden incluir cualquier combinación de datos, procesos y modelos de objetos que describan con precisión el negocio y los requerimientos de los usuarios (pero no una solución técnica). Los modelos de sistemas no están completos hasta que todos los requerimientos funcionales proporcionados han sido modelados. Los modelos con frecuencia se complementan con especificaciones lógicas detalladas que describen los atributos de datos, reglas y políticas de negocios.

Los analistas de sistemas facilitan la tarea. Ellos también documentan los resultados. Evidentemente, los USUARIOS DE SISTEMAS son la fuente primaria de detalles de hechos necesarios para elaborar los modelos. Como se muestra en la figura 4.17, esta tarea (y fase) es realizada por *cada* REQUERIMIENTO FUNCIONAL. Las salidas son los MODELOS DEL SISTEMAS Y ESPECIFICACIONES DETALLADAS reales. El nivel de detalle requerido depende de la metodología que se sigue. Los métodos acelerados generalmente requieren de documentación “únicamente esencial”. ¿Cuánto es esencial? Eso es discutible, pero los expertos hábiles en métodos sostienen que cada producto será esencial para las siguientes fases de diseño y programación. En este texto se le enseñará una diversidad de herramientas y técnicas de elaboración de modelos de sistemas para aplicarse a un diseño lógico.

### > Tarea 4.1b: Requerimientos funcionales del prototipo (alternativa)

La elaboración de prototipos es una alternativa (y a veces un requerimiento previo) para la elaboración de sistemas. Algunas veces los usuarios tienen dificultad para expresar los hechos necesarios para elaborar los modelos de sistemas adecuados. En tal caso, una alternativa o un enfoque complementario a la elaboración de modelos de sistemas es construir prototipos de identificación. La elaboración de prototipos normalmente se utiliza en la fase de análisis de requerimientos para construir entradas y salidas de ejemplos. Estas entradas y salidas ayudan a construir la base de datos y los programas para introducir y generar datos hacia y desde la base de datos. Aunque la elaboración del prototipo es opcional, frecuentemente se aplica a los proyectos de desarrollo de sistemas, en especial en casos donde los usuarios tengan dificultad en comunicar o visualizar sus requerimientos de negocio. La filosofía del prototipo afirma que los usuarios reconocerán sus requerimientos cuando los vean.

LOS CONSTRUCTORES DE SISTEMAS facilitan esta tarea de análisis. LOS ANALISTAS DE SISTEMAS documentan y analizan los resultados. Como siempre, los USUARIOS DE SISTEMAS son la fuente primaria de la entrada de hechos a la tarea. En la figura 4.17 se demuestra que esta tarea depende de uno o más REQUERIMIENTOS FUNCIONALES que han sido identificados por los usuarios. Los constructores y los analistas de sistemas responden construyendo PROTOTIPOS. Como se describió anteriormente en este capítulo, es posible aplicar una *ingeniería inversa* a algunos MODELOS DE SISTEMAS basándose en los prototipos de las bases de datos y las bibliotecas de programas.

### > Tarea 4.2: Validar requerimientos funcionales

Tanto LOS MODELOS DE SISTEMAS como LOS PROTOTIPOS son representaciones de los requerimientos de los usuarios. Éstos deben ser validados para que estén completos y correctos. LOS ANALISTAS DE SISTEMAS facilitan la tarea de asignación de prioridades al comprometer en forma interactiva a los usuarios del sistema para la identificación de errores, omisiones y aclaraciones.

### > Tarea 4.3: Definir casos de prueba de aceptación

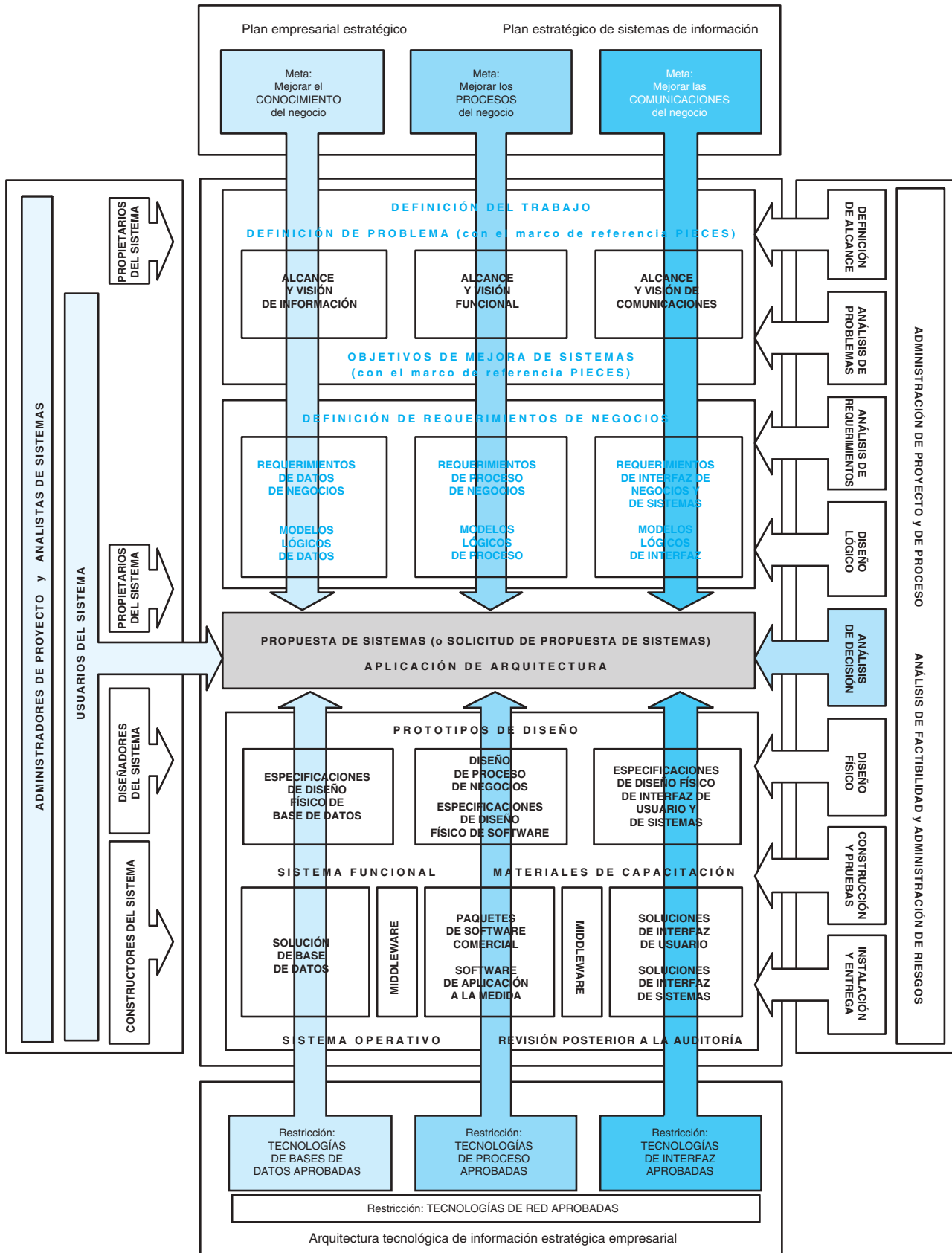
Aunque no es una tarea requerida, la mayoría de los expertos están de acuerdo en que no es demasiado adelantado comenzar a planear las pruebas del sistema. Los modelos y los prototipos de sistemas definen muy eficazmente los requerimientos de proceso, reglas de datos y reglas de negocios para el nuevo sistema. En consecuencia, estas especificaciones pueden ser utilizadas para definir CASOS DE PRUEBA que pueden finalmente ser utilizados para probar que los programas estén correctos. Tanto LOS ANALISTAS como LOS CONSTRUCTORES DE SISTEMAS pueden desempeñar esta tarea y validar los casos de pruebas con LOS USUARIOS DE SISTEMAS.

Recuerde que LOS OBJETIVOS DE MEJORA DE SISTEMAS fueron definidos anteriormente en el proyecto. Los casos de pruebas pueden ser definidos para probar estos objetivos también.

## Fase de análisis de decisión

Una vez que se tienen los requerimientos del negocio para un sistema de información mejorado, es posible abordar la implementación tecnológica del nuevo sistema (incluidas las alternativas basadas en computadora). El propósito de la *fase de análisis de decisión* es identificar las soluciones alternativas, analizarlas y recomendar un sistema objetivo que será diseñado, construido e implementado. Es muy posible que alguien ya haya detectado una posible solución técnica. Pero las soluciones alternativas, lo que es mejor, casi siempre existen. Durante la fase de análisis de decisión, es necesario que usted identifique opciones, las analice y luego venda la mejor solución con base en el análisis.

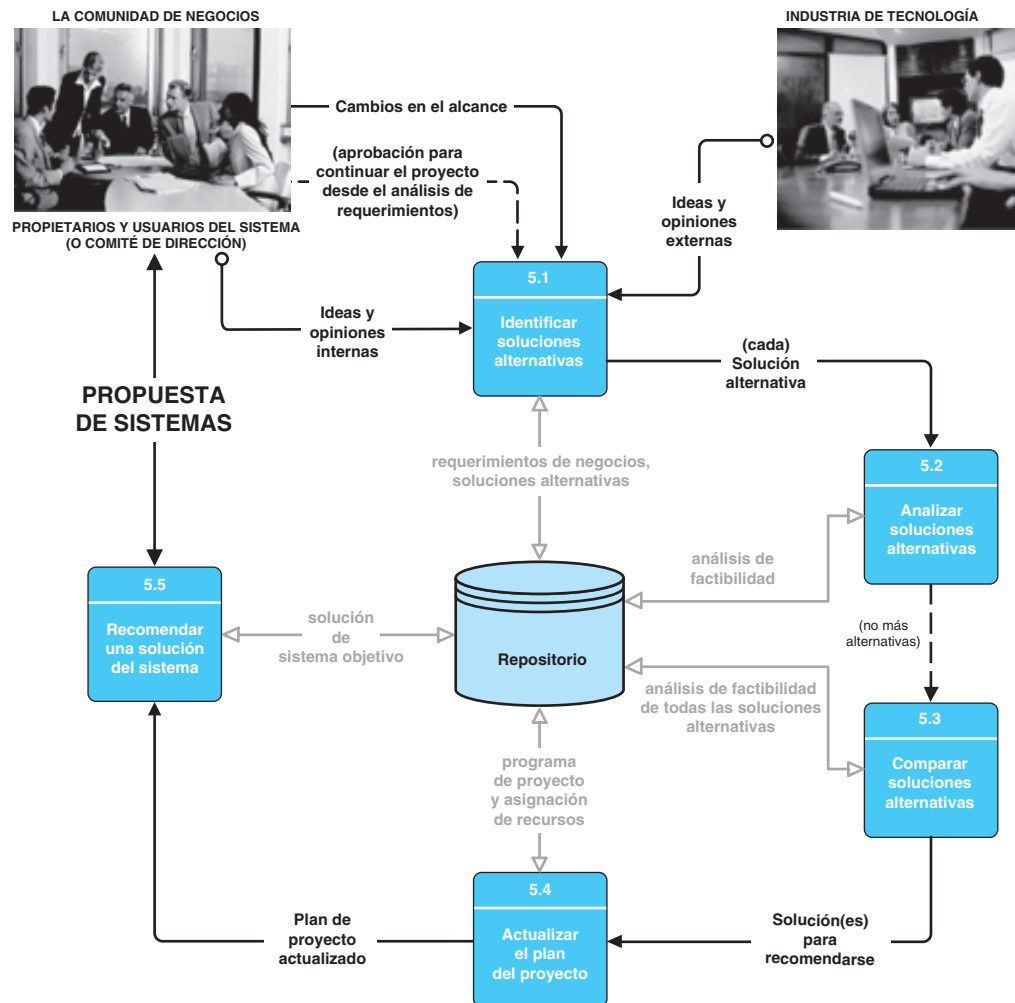
Una vez más, los componentes de sistemas de información (figura 4.18) pueden servir como un marco de referencia útil para la fase de análisis de decisión. Una de las primeras cosas que usted debe notar es que la tecnología de información y la arquitectura comienzan a influir en las decisiones que debemos tomar. En algunos casos, debemos trabajar con estándares. En otros casos, podemos intentar aplicar tecnología distinta o



**FIGURA 4.18** Contexto de la fase de análisis de decisión del análisis de sistemas

**FIGURA 4.19**

Tareas para la fase de análisis de decisión del análisis de sistemas



en surgimiento. Usted también debe notar que las perspectivas (puntos de vista) de los USUARIOS DEL SISTEMA están en transición con respecto a las de los DISEÑADORES DEL SISTEMA. Nuevamente, esto refleja nuestro interés por los negocios o la tecnología. Pero aún así, no estamos en el diseño. Los componentes indican nuestra meta de desarrollar una propuesta que cumpla con todos los requerimientos.

En la figura 4.19 se ilustran las tareas típicas de la fase de análisis de decisión. El producto y meta de la fase final es producir una PROPUESTA DEL SISTEMA que satisfaga los requerimientos de sistemas identificados en la fase previa. La fase de análisis de decisión normalmente incluye las siguientes tareas:

- 5.1 Identificar soluciones alternativas
- 5.2 Analizar soluciones alternativas
- 5.3 Comparar soluciones alternativas
- 5.4 Actualizar el plan del proyecto
- 5.5 Recomendar una solución del sistema

Ahora analicemos con mayor detalle cada una de estas tareas.

### > Tarea 5.1: Identificar soluciones alternativas

Dados los requerimientos de negocios establecidos en la fase de definición del análisis de sistemas, debemos primero identificar las soluciones alternativas. Algunas soluciones alternativas serán planteadas por las ideas y opiniones de diseño de los PROPIETARIOS y USUARIOS DE SISTEMAS. Otras vendrán de diversas fuentes incluyendo a los ANALISTAS DE SISTEMAS, DISE-



ÑADORES DE SISTEMAS, consultores técnicos y otros profesionales de sistemas de información. Y algunas opciones pueden estar limitadas por una arquitectura de tecnología aprobada, predefinida. Es la intención de esta tarea no evaluar las opciones, sino simplemente definir las posibles alternativas de solución que se van a considerar.

LOS ANALISTAS DE SISTEMAS facilitan esta tarea. LOS PROPIETARIOS y USUARIOS DE SISTEMAS normalmente no participan de manera directa en esta tarea, sino que pueden contribuir con ideas y opiniones que comienzan la tarea. Por ejemplo, un propietario o usuario puede haber leído un artículo o haber escuchado o sabido de la forma en que se implementó un sistema similar de un conocido o un competidor. En cualquier caso, es políticamente profundo el considerar las ideas. LOS DISEÑADORES y CONSTRUCTORES DE SISTEMAS como administradores de bases de datos, los administradores de redes y los arquitectos de tecnología y los programadores también son una fuente de ideas y opiniones.

Como se mostró en la figura 4.19, esta tarea es formalmente realizada cuando se otorga la APROBACIÓN PARA CONTINUAR EL PROYECTO DE LA FASE DE ANÁLISIS DE REQUERIMIENTOS. En realidad, las ideas y opiniones han sido generadas y capturadas desde la fase de investigación preliminar; es humano sugerir soluciones a lo largo de cualquier proceso de solución de problemas. Nótese que, además de venir del equipo de proyecto mismo, las IDEAS Y OPINIONES pueden ser generadas de fuentes internas y externas. Cada idea generada se considera como una SOLUCIÓN ALTERNATIVA para los REQUERIMIENTOS DEL NEGOCIO.

La cantidad de información que describe las características de cualquier solución opcional puede resultar abrumadora. Una matriz de alternativas, como la figura 4.20, es una herramienta útil para capturar, organizar y comparar eficazmente las características de las distintas opciones de solución.

Como ha sido el caso a lo largo de este capítulo, las técnicas de identificación de hechos y de facilitación de grupo como JRP son las técnicas principales utilizadas para investigar soluciones alternativas de sistemas. Las técnicas de identificación de hechos y de facilitación de grupo se enseñan en el siguiente capítulo. Asimismo, en el capítulo 9, “Análisis de factibilidad y propuesta del sistema”, se le enseñará la forma de generar y documentar las soluciones alternativas de sistemas en la matriz.

## > Tarea 5.2: Analizar soluciones alternativas

Debe analizarse la factibilidad de cada solución alternativa de sistemas. Esto puede ocurrir conforme se identifique cada posible solución o después de que todas las soluciones opcionales han sido identificadas. Los análisis de factibilidad no deben ser limitados a los costos y beneficios. La mayoría de los analistas evalúan soluciones contra al menos cuatro conjuntos de criterios:

- *Factibilidad técnica.* ¿Es la solución técnicamente práctica? ¿Nuestro personal tiene experiencia técnica para diseñar y construir esta solución?
- *Factibilidad operativa.* ¿La solución cumplirá con todos los requerimientos de los usuarios? ¿A qué grado? ¿Cómo cambiará la solución el ambiente de trabajo del usuario? ¿Cómo se sienten los usuarios acerca de dicha solución?
- *Factibilidad económica.* ¿La solución tiene un costo-beneficio?
- *Factibilidad de programa.* ¿Puede la solución ser diseñada e implementada dentro de un periodo aceptable?

Al completar esta tarea, los analistas y usuarios deben tener cuidado de no hacer comparaciones entre las diferentes soluciones. El análisis de factibilidad se realiza con cada solución alternativa sin importar la factibilidad de las demás soluciones. Este enfoque desalienta al analista y a los usuarios de tomar una decisión en forma prematura con respecto a una solución.

De nuevo, LOS ANALISTAS DE SISTEMAS facilitan la tarea. Normalmente los PROPIETARIOS DEL SISTEMA y LOS USUARIOS DEL SISTEMA analizan la factibilidad operativa, económica y de programa. LOS DISEÑADORES y CONSTRUCTORES DEL SISTEMA normalmente contribuyen al análisis y juegan un papel fundamental para analizar la factibilidad técnica.

En la figura 4.19 se muestra que la tarea es realizada cuando cada solución alternativa es analizada; sin embargo, es aceptable retrasar esta tarea hasta que todas las soluciones hayan sido identificadas. Las opiniones sobre los análisis de factibilidad provienen de diversos participantes en el equipo; sin embargo, es común que los expertos externos (e

**FIGURA 4.20** Matriz de soluciones alternativas del sistema

Características	Solución alternativa 1	Solución alternativa 2	Solución alternativa 3	Solución alternativa...
<b>Porción de sistema computarizado</b> Breve descripción de esa porción del sistema que sería computarizada en esta solución alternativa.	Paquete COTS Platinum Plus de Entertainment Software Solutions se adquiriría y sería personalizado para satisfacer la funcionalidad requerida para satisfacer los servicios del negocio.	Servicios y operaciones del almacén en relación con el llenado de pedidos.	Igual a la solución alternativa 2.	
<b>Beneficios</b> Breve descripción de los beneficios de negocios que se lograrían con esta solución alternativa.	Esta solución puede ser implementada rápidamente porque es comprada.	Respalda totalmente los procesos de negocios requeridos para SoundStage Inc. Más interacción, más eficiencia con las cuentas del negocio.	Igual a la solución alternativa 2.	
<b>Servidores y estaciones de trabajo</b> Descripción de los servidores y las estaciones de trabajo necesarias para respaldar esta solución alternativa.	La arquitectura técnica sugiere utilizar una computadora con procesador Pentium Pro, servidores de clase con MS Windows NT y CPU Pentium, estaciones de trabajo con MS Windows NT 4.0 (clientes).	Igual a la solución alternativa 1.	Igual a la solución alternativa 1.	
<b>Herramientas de software necesarias</b> Las herramientas de software necesarias para diseñar y construir la solución alternativa (por ejemplo, sistema de administración de base de datos, emuladores, sistemas operativos, lenguajes, etcétera). No son generalmente aplicables si los paquetes de aplicación de software se van a adquirir.	MS Visual C++ y MS Access para proporcionar el reporte y la integración.	MS Visual Basic 5.0 System Architect 3.1 Internet Explorer	MS Visual Basic 5.0 System Architect 3.1 Internet Explorer	
<b>Software de aplicación</b> Descripción del software que se va a adquirir, desarrollar, acceder o alguna combinación de estas técnicas.	Solución del paquete	Solución personalizada	Igual a la solución alternativa 2.	
<b>Método de proceso de datos</b> Generalmente alguna combinación del procesamiento en línea, por lotes, lotes diferidos, lotes remotos y tiempo real.	Cliente/servidor	Igual a la solución alternativa 1.	Igual a la solución alternativa 1.	
<b>Dispositivos e implicaciones de salida</b> Descripción de dispositivos de salida que se utilizarían, requerimientos de salida especiales (por ejemplo, red, formatos preimpresos, etcétera) y consideraciones de salida (por ejemplo, restricciones de tiempo).	(2) HP4MV impresoras láser del departamento (2) impresoras láser HP5SI	(2) HP4MV impresoras láser por departamento. (2) HP5SI LAN impresoras láser. (1) PRINTRONIX impresora de código de barras (incluye programas y controladores). Las páginas Web deben ser diseñadas para resolución VGA. Todas las pantallas internas serán diseñadas para resolución SVGA.	Igual a la solución alternativa 2.	
<b>Dispositivos e implicaciones de entrada</b> Descripción de métodos de entrada que se van a utilizar, dispositivos de entrada (tablero, ratón, etcétera), requerimientos especiales de entrada (por ejemplo, formatos nuevos o revisados de los cuales se ingresarán datos) y consideraciones de entrada (por ejemplo, tiempos de las entradas reales).	Tablero y ratón	Apple "Quick Take" cámara digital y software. (15) PSC Quickscan escáners de código de barras láser. (1) HP Scanjet 4C escáner flatbed. Teclado y ratón.	Igual a la solución alternativa 2.	
<b>Dispositivos e implicaciones de almacenamiento</b> Breves descripciones de qué datos deben ser almacenados, qué datos serían consultados de las tiendas existentes, qué medios de almacenamiento se utilizarían, cuánta capacidad de almacenamiento sería necesaria y cuántos datos serían organizados.	Servidor MS SQL DBMS con capacidad de 100 GB.	Solución alternativa 1	Igual a la solución alternativa 1.	

influencias) también proporcionen información al respecto. El análisis de factibilidad de cada solución alternativa se guarda en el repositorio para hacer una comparación posterior con las demás opciones de solución.

Nuevamente, las técnicas de identificación de hechos, desempeñan un papel importante en la tarea de análisis de sistemas. En consecuencia, la realización de un análisis de factibilidad para cada solución alternativa del sistema es esencial. Esa técnica se enseña en el capítulo 9, “Análisis de factibilidad y propuesta del sistema”.

### > Tarea 5.3: Comparar soluciones alternativas

Una vez que el análisis de factibilidad ha sido completado para cada solución alternativa, podemos compararlas y elegir una o más soluciones para recomendarlas a los PROPIETARIOS y a los USUARIOS DE SISTEMAS. En este punto, cualquier solución no factible normalmente se elimina de cualquier consideración posterior. Debido a que estamos buscando la solución más factible de las soluciones restantes, identificaremos y recomendaremos la solución que ofrezca la más completa combinación de factibilidad técnica, operativa, económica y de cronograma. Debe entenderse que durante la selección, es raro que se encuentre una solución que presente la factibilidad más operativa, técnica, económica y de cronograma.

Una vez más, los ANALISTAS DEL SISTEMA facilitan la tarea. Los DISEÑADORES y CONSTRUCTORES DEL SISTEMA deben estar disponibles para responder cualquier pregunta de factibilidad técnica. Pero finalmente, los PROPIETARIOS y los USUARIOS DEL SISTEMA deben estar facultados para dirigir el análisis final y la recomendación.

En la figura 4.19, esta tarea se lleva a cabo cuando se finaliza el análisis de factibilidad de todas las soluciones alternativas (NO MÁS SOLUCIONES ALTERNATIVAS). La información de entrada son TODOS LOS ANÁLISIS DE FACTIBILIDAD DE LAS SOLUCIONES ALTERNATIVAS. Una vez más, se puede utilizar una matriz para comunicar la gran cantidad de información que se tiene sobre las opciones de solución. La matriz de factibilidad de la figura 4.21 permite una comparación lado a lado de los diversos análisis de factibilidad que se tienen sobre las diferentes soluciones alternativas.

El producto de esta tarea es la(s) SOLUCIÓN(ES) QUE SE RECOMENDARÁ(N). Si se recomienda más de una solución, se deben establecer prioridades.

De nuevo, las técnicas de análisis de factibilidad (y la matriz) se enseñarán en el capítulo 9, “Análisis de factibilidad y propuesta de sistema”.

### > Tarea 5.4: Actualizar el plan del proyecto

Esperamos que se haya percatado del tema recurrente a lo largo de este capítulo. Continuamente actualizamos nuestro plan de proyecto al tiempo que aprendemos más acerca de un sistema, sus problemas, sus requerimientos y sus soluciones. Ajustamos el alcance en consecuencia. Por tanto, con base en nuestras soluciones recomendadas, debemos una vez más reevaluar el alcance del proyecto y, en consecuencia, *actualizar el plan del proyecto*.

El administrador del proyecto, en conjunto con los PROPIETARIOS DE SISTEMAS y el equipo completo del proyecto, facilitan esta tarea. Los ANALISTAS DE SISTEMAS y los PROPIETARIOS DE SISTEMAS son los individuos fundamentales en esta tarea. Pero como estamos en transición hacia el diseño de sistema técnico, necesitamos comenzar a incluir a los DISEÑADORES y CONSTRUCTORES DE SISTEMAS en las actualizaciones del plan del proyecto.

Como se muestra en la figura 4.19, esta tarea se dispara por la terminación de la(s) SOLUCIÓN(ES) QUE SE RECOMENDARÁ(N). El más reciente PROGRAMA DE PROYECTO Y ASIGNACIÓN DE RECURSOS debe ser revisado y actualizado. El PLAN DE PROYECTO ACTUALIZADO es el producto clave. El plan actualizado debe ahora incluir un plan detallado para la fase de diseño de sistemas que seguirá.\*

### > Tarea 5.5: Recomendar una solución del sistema

Al igual que con las fases de investigación preliminar y de análisis de problemas, la fase de análisis de decisión concluye con una tarea de comunicación. Debemos *recomendar una solución de sistema* para la comunidad del negocio.

\*Nota del R.T.: Para el lector interesado en el tema, las técnicas y pasos para actualizar el plan de proyecto se enseñan en el PMBOK del PMI.

**FIGURA 4.21** Matriz de soluciones alternativas del sistema

criterio de factibilidad	Peso	Solución alternativa 1	Solución alternativa 2	Solución alternativa 3	Solución alternativa...
<b>Factibilidad operativa</b> <b>Funcionalidad.</b> Descripción de a qué grado la solución alternativa beneficiaría a la organización y qué tan bien trabajaría el sistema. <b>Política.</b> Descripción de qué tan bien recibida sería esta solución por parte de la administración de usuarios, y el resto de la organización.	30%	Respalda únicamente los requerimientos de servicios ya que los procesos de negocios actuales tendrían que ser modificados para aprovechar la funcionalidad del software.  <b>Calificación: 60</b>	Respalda totalmente la funcionalidad requerida del usuario.  <b>Calificación: 100</b>	Igual a la solución alternativa 2.  <b>Calificación: 100</b>	
<b>Factibilidad técnica</b> <b>Tecnología.</b> Evaluación de la madurez, disponibilidad (o capacidad de adquirir) y necesidad de la tecnología de cómputo requerida para respaldar esta solución alternativa.  <b>Experiencia.</b> Evaluación de la experiencia técnica necesaria para desarrollar, operar y mantener la solución alternativa del sistema.	30%	La liberación actual del paquete Platinum Plus es la versión 1.0 y ha estado en el mercado por sólo seis semanas. La madurez del producto es un riesgo y el proveedor carga una cuota mensual adicional al monto del software por incluir el soporte técnico del mismo. Se requiere contratar o capacitar a un profesional que tenga experiencia en lenguaje C++ para realizar modificaciones para los requerimientos de integración del sistema. <b>Calificación: 50</b>	Aunque el personal técnico actual tiene sólo experiencia en Powerbuilder, los analistas senior que vieron la demostración y presentación de MS Visual Basic han acordado que la transición será simple y que encontrar programadores experimentados en VB será más fácil que encontrar programadores de Powerbuilder y a un costo mucho menor. MS Visual Basic 5.0 es una tecnología madura basada en el número de versión. <b>Calificación: 95</b>	Aunque el personal técnico actual se siente cómodo con Powerbuilder, la administración está preocupada con la reciente adquisición de Powerbuilder de Sybase Inc. MS SQL Server es un estándar actual de la compañía y compile con SYBASE en el mercado de los DBMS cliente/servidor. Debido a esto no tenemos garantía de que las versiones futuras de Powerbuilder "correrán bien" con nuestra versión actual de SQL Server. <b>Calificación: 60</b>	
<b>Factibilidad económica</b> <b>Costo para desarrollar:</b> <b>Periodo de retribución (descontado):</b> <b>Valor neto presente:</b> <b>Cálculos detallados:</b>	30%	Aproximadamente \$350 000  Aproximadamente 4.5 años Aproximadamente \$210 000 Véase el Anexo A <b>Calificación: 60</b>	Aproximadamente \$418 040  Aproximadamente 3.5 años Aproximadamente \$306 748 Véase el Anexo A <b>Calificación: 85</b>	Aproximadamente: \$400 000  Aproximadamente 3.3 años Aproximadamente \$325 500 Véase el Anexo A <b>Calificación: 90</b>	
<b>Factibilidad del programa</b> Evaluación de cuánto tomará diseñar e implementar la solución alternativa	10%	Menos de tres meses.  <b>Calificación: 95</b>	9 a 12 meses  <b>Calificación: 80</b>	9 meses  <b>Calificación: 85</b>	
<b>Clasificación</b>	<b>100%</b>	<b>60.5</b>	<b>92</b>	<b>83.5</b>	

El administrador de proyecto y el patrocinador ejecutivo deben realizar conjuntamente esta tarea. Otros participantes en la reunión deben incluir al equipo completo del proyecto, así como a los PROPIETARIOS, USUARIOS, ANALISTAS, DISEÑADORES y CONSTRUCTORES DE SISTEMAS asignados. Como siempre, la reunión debe estar abierta a cualquier empleado interesado de la comunidad del negocio. También, si se estableció un sitio Web de intranet para el proyecto, debe ser actualizado a lo largo de las fases de análisis de problemas para asegurar una comunicación continua del progreso del proyecto.

Esta tarea se realiza cuando se tiene el PLAN DEL PROYECTO ACTUALIZADO. La SOLUCIÓN DE SISTEMA OBJETIVO (de la tarea 4.3) se reforma para su presentación como una PROPUESTA DE SISTEMA. El formato puede ser un informe, una presentación verbal o una inspección por un auditor o un grupo de colegas (llamado *walkthrough*). En la figura 4.22, se muestra un esquema de un informe escrito.

## La siguiente generación: Análisis de sistemas

Pronosticar el futuro del análisis de sistemas no es fácil, pero lo intentaremos. La tecnología CASE mejorará y eso facilitará el modelado de requerimientos de sistemas. Primero, las herramientas CASE incluirán el modelado de objetos para respaldar las nuevas técnicas de análisis orientadas a objetos. Mientras que algunas herramientas CASE serán sólo orientadas a objetos, creemos que la demanda de otros tipos de soporte de modelado (por ejemplo, modelado de datos para bases de datos, modelado de proceso para BPR) darán un atractivo adicional a las herramientas completas CASE que puedan soportar muchos tipos de modelos. Segundo, la tecnología de ingeniería inversa en las herramientas CASE mejorará más nuestra capacidad de generar con mayor rapidez el modelo inicial de sistemas a partir de las bases de datos y los programas de aplicación existentes.

Mientras tanto, la tecnología RAD continuará permitiendo los métodos de análisis acelerado tales como la elaboración de prototipos. También esperamos que la tendencia de que las herramientas RAD y CASE interactúen a través de la ingeniería hacia adelante e inversa simplifiquen aún más la elaboración de modelos y de prototipos de identificación.

El análisis orientado a objetos eventualmente reemplazará el análisis estructurado y la ingeniería de información como las mejores prácticas para el análisis de sistemas. Este cambio puede no ocurrir con tanta rapidez como los partidarios de los objetos desearían, pero ocurrirá bastante rápido para una generación de analistas de sistemas que sean hábiles en los métodos más antiguos. Existe una gran oportunidad para que

los analistas jóvenes y talentosos dirijan la transición; sin embargo, las oportunidades profesionales seguirán siendo fuertes para los analistas que sepan que la elaboración de modelos de datos continuará en uso para el diseño de bases de datos. También, el renacimiento de la elaboración de modelos de proceso continuará mientras que los proyectos BPR sigan proliferando.

Asimismo pronosticamos que nuestras propuestas de sistemas serán cada vez más interesantes. Mientras que el Internet, el comercio electrónico y los negocios electrónicos continúen su dominio en nuestra economía, los analistas de sistemas pondrán nuevas alternativas a los viejos problemas. Se dará un cambio fundamental en los sistemas de negocios y de información para utilizar estas nuevas tecnologías.

¡Una cosa *no* cambiará! Seguiremos necesitando analistas de sistemas que entiendan la forma de investigar y analizar a profundidad los problemas del negocio y definir los requerimientos lógicos como un requisito previo al diseño de sistemas. Pero todos tendremos que hacer eso con una velocidad y precisión cada vez mayores para cumplir con los programas de desarrollo de sistemas requeridos en la rápida economía del mañana.

Las habilidades interpersonales y de comunicaciones son esenciales para esta tarea. Las habilidades personales como la capacidad de vender y de persuadir se vuelven importantes. (Muchas escuelas ofrecen cursos de expresión oral y de comunicaciones en estos temas.) Los analistas de sistemas deben ser capaces de escribir un informe de negocios formal y de hacer una presentación sin entrar en temas o alternativas técnicas.

Esto concluye la fase de análisis de decisión. Y también nuestra cobertura de análisis de sistemas.

### FIGURA 4.22 Esquema de una propuesta de sistema típica

- I. Introducción
  - A. Propósito del informe
  - B. Antecedentes del proyecto que lleva a este informe
  - C. Alcance del proyecto
  - D. Estructura del informe
- II. Herramientas y técnicas utilizadas
  - A. Solución generada
  - B. Análisis de factibilidad (costo-beneficio)
- III. Requerimientos de sistemas de información
- IV. Soluciones alternativas y análisis de factibilidad
- V. Recomendaciones
- VI. Apéndices

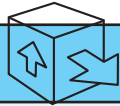
En este capítulo se proporcionó un panorama detallado de las fases de análisis de sistemas de un proyecto. Ahora está listo para aprender algunas de las habilidades de sistemas presentadas en este capítulo. Para la mayoría de los estudiantes, este sería el momento ideal para estudiar las técnicas de identificación de hechos que fueron identificados como críticos para casi cada fase y tarea descritos en este capítulo. En el capítulo 5 se enseñan estas habilidades. Se recomienda que usted lea el capítulo 6, “Modelado de requerimientos del sistema con los casos de uso”, antes de proceder a cualquiera de los capítulos de elaboración de modelos ya que los casos de uso son utilizados comúnmente para facilitar la actividad del modelado.

La secuencia de los capítulos de elaboración de modelos es flexible; sin embargo, personalmente preferimos y recomendamos que el capítulo 7, “Modelado y análisis de datos”, se estudie primero. Todos los sistemas de información incluyen bases de datos y la elaboración de modelos de datos es una habilidad esencial para el desarrollo de bases de datos. También, es más fácil sincronizar los modelos de datos iniciales con modelos de procesos posteriores que hacerlo al contrario. Su instructor podría preferir que usted primero estudie el capítulo 8, “Modelado de procesos”.

Si usted va directo al capítulo de elaboración de modelos de sistemas desde este capítulo, haga un compromiso de regresar al capítulo 5 para estudiar las técnicas de identificación de hechos. Sin importar qué tan bien domine la elaboración de modelos de sistemas, esa habilidad de modelado es completamente dependiente de su capacidad de identificar y reunir los hechos de negocios que subyacen en los modelos.

Para aquellos de ustedes que ya han terminado un curso de análisis de sistemas, este capítulo probablemente fue programado sólo como una revisión o contexto para el diseño de sistemas. Sugerimos que sólo revise los capítulos de modelado de sistemas y que proceda directamente al capítulo 10, “Diseño de sistemas”, el cual continuará donde termina este capítulo.

## Resumen



1. Formalmente, el análisis de sistemas es la diseción de un sistema en sus componentes. Como una fase de solución de problemas, precede al diseño de sistemas. Con relación al desarrollo de sistemas de información, el análisis de sistemas es la investigación preliminar de un proyecto propuesto, el estudio y el análisis de problemas del sistema existente, el análisis de requerimientos de los requerimientos de sistemas para el nuevo sistema y el análisis de decisión para que las soluciones alternativas satisfagan los requerimientos.
2. Los resultados del análisis de sistemas se almacenan en un repositorio para su uso en las fases y proyectos posteriores.
3. Existen diversas estrategias populares o en surgimiento para el análisis de sistemas. Estas técnicas pueden ser utilizadas en combinación entre ellas.
  - a) Las técnicas de análisis basadas en modelos enfatizan el dibujo de modelos de sistemas pictóricos que representan una realidad actual o una visión objetivo del sistema.
    - i) El análisis estructurado es una técnica que se enfoca en la elaboración de modelos de procesos.
    - ii) La ingeniería de información es una técnica que se enfoca en la elaboración de modelos de datos.
    - iii) El análisis orientado a objetos es una técnica que se enfoca en elaboración de modelos de objetos que encapsulan las preocupaciones de datos y procesos que actúan en esos datos.
  - b) Los enfoques de análisis acelerados enfatizan la construcción de modelos de sistemas de un sistema en un esfuerzo por acelerar el análisis de sistemas.
    - i) La elaboración de prototipos de identificación es una técnica que se enfoca en la construcción de subsistemas de pequeña escala, subsistemas funcionales para identificar los requerimientos.
    - ii) El análisis de arquitectura rápido intenta generar en forma automática los modelos de sistemas a partir de prototipos o de sistemas existentes. La generación automática de modelos requiere tecnología de ingeniería inversa.
  - c) Los enfoques basados en modelos y los de análisis de sistema acelerado dependen de las téc-

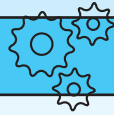
- nicas de identificación de requerimientos para identificar o extraer problemas y requerimientos de los propietarios y usuarios de sistemas.
- i)* La identificación de hechos es el proceso formal de utilizar investigación, entrevistas, cuestionarios, muestreo y otras técnicas para reunir información.
  - ii)* Las técnicas de planeación conjunta de requerimientos (JRP) utilizan talleres facilitados para reunir a todas las partes interesadas y acelerar el proceso de identificación de hechos.
- d)* El rediseño de proceso de negocios es una técnica que se enfoca en simplificar y dirigir los procesos de negocios fundamentales antes de aplicar la tecnología de información a esos procesos.
4. Cada fase de análisis de sistemas (investigación preliminar, análisis de problemas, análisis de requerimientos y análisis de decisión) puede ser entendida en el contexto de los componentes del sistema de información: CONOCIMIENTO, PROCESOS Y COMUNICACIONES.
  5. El propósito de la fase de investigación preliminar es determinar el valor del proyecto y crear un plan para completar esos proyectos considerados merecedores de un estudio y un análisis detallado. Para cumplir con la fase de investigación preliminar, el analista de sistemas trabajará con los propietarios y usuarios de sistemas para: *a)* Listar problemas, oportunidades y directrices; *b)* negociar el alcance preliminar; *c)* evaluar el valor del proyecto; *d)* planear el proyecto y *e)* presentar el proyecto a la comunidad de negocios. El producto de la fase de investigación preliminar es un diagrama del proyecto que debe ser aprobado por los propietarios de sistemas y/o un comité de toma de decisiones, comúnmente llamado comité de dirección.
  6. El propósito de la fase de análisis del problema es responder las preguntas, ¿vale la pena resolver los problemas? y ¿vale la pena construir un nuevo sistema? Para responder estas preguntas, la fase de análisis del problema analiza profundamente los supuestos problemas y las oportunidades identificadas primero en la fase de investigación preliminar. Para completar la fase del análisis del problema, el analista continuará trabajando con el propietario del sistema, usuarios de sistemas y demás administración y personal de sistemas de información. El analista de sistemas y los participantes adecuados *a)* estudiarán el dominio del problema; *b)* analizarán profundamente los problemas y oportunidades; *c)* opcionalmente, analizarán los procesos de negocios; *d)* establecerán objetivos y restricciones de mejora de sistemas; *e)* actualizarán el plan de proyecto y *f)* presentarán los resultados y recomendaciones. El producto obtenido de la fase de análisis de problemas es el conjunto de objetivos de mejora de sistemas.
  7. El propósito de la fase de análisis de requerimientos es identificar lo que deberá hacer el nuevo sistema sin la consideración de la tecnología; en otras palabras, definir los requerimientos de negocios para un nuevo sistema. Como en las fases de investigación preliminar y en la de análisis de problemas, el analista trabaja activamente con los usuarios y los propietarios de sistemas así como con otros profesionales de sistemas de información. Para completar la fase de análisis de requerimientos, el analista y los participantes adecuados *a)* definirán los requerimientos, *b)* analizarán los requerimientos funcionales por medio de la elaboración de modelos de sistemas o prototipos de identificación, *c)* rastrearán y completarán la definición de requerimientos, *d)* otorgarán prioridades a los requerimientos, y *e)* actualizarán el plan de proyecto y el alcance. El producto de la fase de análisis de requerimientos es la definición de requerimientos de negocios. Como los requerimientos son un objetivo móvil sin terminación, el análisis de requerimientos también incluye la tarea continua de administrar cambios a los requerimientos.
  8. El propósito de la fase de diseño lógico es documentar los requerimientos de negocios por medio de modelos de sistemas para el sistema propuesto. Estos modelos de sistemas pueden, según la metodología, ser cualquier combinación de los modelos de proceso, modelos de datos y modelos de objeto. Los modelos describen diversos aspectos de nuestros componentes. De manera alternativa, los prototipos podrían ser construidos para “identificar requerimientos”. Algunos prototipos de identificación pueden recibir una ingeniería inversa para convertirse en modelos de sistemas. El analista de sistemas y los participantes apropiados *a)* estructurarán o elaborarán un prototipo de requerimientos funcionales, *b)* validarán los requerimientos funcionales y *c)* definirán casos de aceptación de prueba. Estas tareas no son necesariamente secuenciales; pueden ocurrir en paralelo. El producto de la fase de diseño lógica es la definición de requerimientos de negocios.
  9. El propósito de la fase de análisis de decisión es encaminar el proyecto partiendo de las necesidades del negocio hacia las soluciones, al identificar, analizar y recomendar una solución técnica para el sistema. Para completar la fase de análisis de decisión, el analista y los participantes adecuados *a)* definirán las soluciones alternativas; *b)* analizarán las soluciones alternativas para factibilidad (técnica, operativa, económica y de cronograma); *c)* compararán las soluciones alternativas factibles para elegir una o más soluciones recomendadas; *d)* actualizarán el plan de proyecto con base en la solución recomendada; y *e)* presentarán y defenderán la solución objetivo. El producto de la fase de análisis de decisión es la propuesta del sistema.

## Preguntas de repaso



1. ¿Cuáles son los factores del negocio que impulsan el análisis de sistemas? Con base en estos factores, ¿qué deben abordar los análisis de sistemas?
2. ¿Qué es el análisis basado en modelos? ¿Por qué se utiliza? Proporcione varios ejemplos.
3. ¿Cuál es el mayor enfoque del análisis estructurado?
4. ¿Cuál es el mayor enfoque de la ingeniería de información?
5. ¿Por qué el análisis orientado a objetos se ha vuelto popular? ¿Qué problemas resuelve?
6. ¿Cuáles son las cinco fases del análisis de sistemas?
7. ¿Cuál es la meta de la fase de definición de alcance?
8. ¿Cuáles son las cinco tareas que usted hace en la fase de definición de alcance?
9. ¿Cuál es el gatillo para comunicar el plan del proyecto y a quiénes se comunicará? ¿Por qué es importante comunicar el plan de proyecto?
10. ¿Por qué muchos analistas de sistemas novatos fallan en analizar eficazmente los problemas? ¿Qué pueden hacer ellos para volverse más eficientes?
11. ¿Cuál es una herramienta popular utilizada para identificar y expresar los requerimientos funcionales de un sistema?
12. ¿Cuál es una técnica comúnmente utilizada para asignar prioridades a los requerimientos de sistemas?
13. ¿Cuándo podría utilizarse la elaboración de prototipos en lugar de la de modelos de sistemas para determinar los requerimientos funcionales?
14. ¿Por qué se necesita la fase de análisis de decisión?
15. ¿Cuáles son algunas formas para identificar las soluciones alternativas?

## Problemas y ejercicios



1. Hay muchos enfoques distintos para el análisis de sistemas. A pesar de estos enfoques distintos, ¿cuál es la definición universalmente aceptada del análisis de sistemas? ¿Cuál es la opinión general acerca de cuándo comienza el análisis de sistemas y cuándo termina? Como administrador de proyecto, ¿qué es importante saber acerca de la definición del análisis de sistemas y qué es importante asegurar en su organización en cuanto a la definición?
2. Como un analista de sistemas, usted estará expuesto a utilizar muchos enfoques distintos para el

análisis de sistemas a lo largo de su carrera. Es importante que entienda la base conceptual de cada tipo de método, sus diferencias, fortalezas y debilidades esenciales. Considere las diferencias en el análisis estructurado, la ingeniería de información y la elaboración de modelos de datos, así como el análisis orientado a objetos, todos los cuales representan un análisis basado en modelos y conteste la matriz que se presenta a continuación.

	CENTRICIDAD (datos, proceso, etcétera)	TIPO DE MODELOS UTILIZADOS	DIFERENCIAS ESENCIALES
ANÁLISIS ESTRUCTURADO			
INGENIERÍA DE INFORMACIÓN Y ELABORACIÓN DE MODELOS DE DATOS			
ANÁLISIS ORIENTADO A OBJETOS			



3. Los métodos de análisis de sistemas acelerados están basados en la premisa de que los prototipos pueden ayudar a revelar los requerimientos de negocios más importantes, mucho más rápido que los otros métodos. Describa los dos métodos más comúnmente utilizados para el análisis acelerado. ¿Qué hacen y cómo lo hacen? ¿Cuál es una de las críticas de la elaboración de prototipos? ¿Los métodos de análisis de sistemas acelerados reemplazan completamente a los métodos más formales, tales como un análisis estructurado?
4. Durante la fase de definición de alcance, ¿cuál es una pregunta que usted nunca debe perder de vista? ¿Y cómo la responde usted? ¿Cuáles cinco tareas deben ocurrir durante la fase de definición de alcance?
5. Usted es un analista de sistemas novato y está ansioso de probar sus capacidades en su primer proyecto. Usted está en una junta de análisis de problemas con los propietarios y los usuarios de sistemas y dice, “necesitamos hacer esto para resolver el problema”. ¿En qué trampa común está en peligro de caer? ¿Qué técnica podría utilizar para evitar esta trampa?
6. Su equipo de proyecto ha completado la fase de definición de alcance y está ahora en el punto en la fase de análisis de problemas para establecer los objetivos de mejora de sistemas. Como analista de sistemas en el equipo de proyecto, usted es el facilitador de una sesión de lluvia de ideas para definir los objetivos de mejora de sistemas. Como varios de los propietarios y usuarios del proyecto nunca han hecho esto antes, describa las características de los objetivos de mejora de sistemas adecuados y proporcione algunos ejemplos. Los miembros del equipo de proyecto sugieren los siguientes objetivos:
  - a) Reduzca el tiempo requerido para procesar el pedido.
  - b) El nuevo sistema debe utilizar Oracle para almacenar datos.
  - c) Las pantallas de entrada de datos deben ser rediseñadas para que sean más amigables al usuario.
  - d) La tasa de satisfacción del cliente con el proceso de pedido en línea debe aumentarse en un 10 por ciento.

¿Son estos ejemplos de objetivos de mejora de sistemas adecuados? ¿Por qué sí o por qué no? Si no, ¿cómo se pueden replantear? Frecuentemente, los objetivos tienen restricciones que están vinculadas a ellos; ¿cuál cree usted que sería la restricción que concuerda con estos dos objetivos?
7. Usted ha pasado ya por la fase de análisis de problemas del proyecto, y ahora comienza la fase de análisis de requerimientos. Durante la primera reunión acerca de los requerimientos del negocio, uno de los otros analistas en el equipo del proyecto pregunta a los usuarios de sistemas, “¿cómo debe satisfacer las necesidades del negocio el nuevo sistema?” ¿Qué error común comete el analista de sistemas? ¿Cuáles son con frecuencia las consecuencias de cometer este error?
8. ¿Cuál es la diferencia entre requerimientos funcionales y no funcionales y cuál es el propósito de clasificarlos en estas categorías? ¿Cuáles son dos formatos que un analista puede utilizar para documentar los requerimientos de sistema funcionales?
9. ¿Es importante priorizar los requerimientos de sistemas? y si es así, ¿cuándo debe hacerse esto? ¿Cuál es una técnica que puede ser utilizada y cuál es la diferencia entre requerimientos obligatorios y deseables? ¿Cuál es una forma de probar si un requerimiento es realmente obligatorio?
10. Una vez que se identifican los requerimientos del sistema y que se asignan las prioridades, ¿no deberían limitarse los requerimientos para prevenir un aumento del alcance o de las características? ¿El actualizar el plan del proyecto o permitir a los interesados continuar solicitando cambios sólo retrasará el diseño y la construcción del sistema e incluso hasta la terminación del proyecto mismo?
11. ¿Por qué los casos de aceptación de pruebas deben definirse durante la fase de diseño lógico? Después de todo, el diseño técnico no ha sido hecho aún, mucho menos la construcción del sistema. ¿No deberían esperar las actividades de prueba al menos hasta que la construcción esté en proceso?
12. ¿En qué difiere la fase de diseño lógico de la fase de análisis de requerimientos?
13. Digamos que usted forma parte del equipo de proyecto de un sistema que ha tenido muchos problemas durante la fase de análisis de requerimientos y que lleva varias semanas de retraso. El administrador del proyecto quiere tratar de recuperar el tiempo perdido al pasar por alto algunas tareas o abreviarlas en la fase de diseño lógico. El administrador del proyecto argumenta que, después de todo, ahora tienen una idea clara de los requerimientos, los diseñadores y los constructores son realmente experimentados y no necesitan el diseño lógico con el fin de hacer el diseño técnico. ¿Este es un método legítimo de ponerse al corriente con el cronograma de trabajo? ¿Cuáles son las posibles consecuencias?
14. Al identificar y definir las soluciones alternativas posibles, ¿cuáles son los papeles típicos de los diversos interesados que participan en el proyecto?
15. Usted es un analista de sistemas y se le ha pedido revisar la factibilidad del análisis y la evaluación de diversas soluciones de sistemas. ¿Qué conjuntos de criterios utilizaría normalmente? ¿Debe usted comparar las soluciones alternativas entre ellas en este punto? ¿Por qué sí o por qué no? ¿Cuál es el producto típico que resulta de esta tarea?

# Proyectos e investigación



1. Elija un sistema de información con el que esté familiarizado y que sienta debe ser mejorado, con base en sus experiencias como empleado, cliente, usuario de otro sistema o propietario de sistemas. Cambie de roles y de perspectivas conforme sea necesario realizar o responder lo siguiente:
  - a) Describa la naturaleza del sistema de información que ha elegido.
  - b) Describa a la organización que es dueña y mantiene el sistema de información.
  - c) Identifique los problemas y oportunidades de la línea de base, por la tarea 1.1.
  - d) Desarrolle una definición de problema preliminar, por medio del formato mostrado en la figura 4.8.
  
2. Asuma que usted es ahora un analista de sistemas en el proyecto descrito en la pregunta anterior. La administración ejecutiva estuvo realmente impresionada por su trabajo en la definición del problema. Como resultado, han dado el “visto bueno” al proyecto, se ha desarrollado el cronograma de trabajo y el de presupuesto, así mismo, el plan del proyecto ha sido aprobado por el comité de dirección ejecutivo. Como el analista de sistemas, ahora se le han asignado las siguientes tareas:
  - a) Desarrollar y documentar su comprensión sobre el dominio y vocabulario de negocios de este problema, utilizando el marco de referencia de componentes de sistemas de información de este libro, tal como se describe en la tarea 2.1.
  - b) Analizar los problemas y oportunidades por medio de un análisis de causa y efecto (tarea 2.2).
  - c) Analizar los procesos de negocios y desarrollar los modelos de proceso (tarea 2.3).
  - d) Establecer los objetivos de mejora de sistemas (tarea 2.4)
  - e) Preparar una matriz de problemas, oportunidades, objetivos y restricciones, utilizando la figura 4.12 como un ejemplo.
  
3. Comunicar los resultados y recomendaciones es la tarea final en la fase de análisis de problemas. Como analista de sistemas en el proyecto, usted tiene la tarea de preparar el informe de objetivos de mejora de sistemas y recomendaciones. Para este ejercicio, prepare sólo la porción de resumen ejecutivo del informe, al usar el formato mostrado en la figura 4.13. El comité de dirección ejecutivo utilizará este resumen para tomar sus decisiones en relación con las recomendaciones.
4. Hasta el momento, su trabajo eficiente en el proyecto ha continuado impresionando a la administración ejecutiva. Usted ha recibido un aumento de sueldo y se le ha pedido conducir la fase de análisis de requerimientos. Específicamente:
  - a) Identificar los requerimientos de sistemas y preparar un esquema de requerimientos funcionales y no funcionales, de acuerdo a la tarea 3.1. Como su organización utiliza la técnica de análisis estructurado y no la de modelo de casos, liste cada objetivo de mejora de sistemas, las entradas, los procesos, las salidas y los datos almacenados requeridos para cumplir con cada objetivo.
  - b) Asuma que los requerimientos que usted identificó en el paso anterior han sido validados. Asigne prioridades a los requerimientos de acuerdo con su importancia relativa y utilice el método descrito en la tarea 3.2.
5. Su trabajo ha ayudado a mantener el proyecto por delante del programa, así que la administración ejecutiva le da un par de semanas de vacaciones pagadas. Cuando usted regresa, el proyecto va hacia la fase de análisis de decisión. Su próxima tarea es identificar las soluciones alternativas.
  - a) Describa el proceso para identificar las soluciones alternativas. ¿Qué debe tener cuidado de *no* hacer en este punto?
  - b) Desarrolle una matriz de soluciones alternativas del sistema y utilice el formato de la figura 4.20 como ejemplo e incluya tres posibles soluciones.
6. Después de identificar soluciones alternativas, el siguiente paso es analizarlas.
  - a) Describa el proceso para analizar soluciones alternativas. ¿Qué es lo que *no* debe hacer el equipo del proyecto para completar esta tarea?
  - b) Desarrolle una matriz de análisis de factibilidad con base a las soluciones alternativas identificadas en la pregunta anterior y utilice el formato que se muestra en la figura 4.21 como un ejemplo. Determine los factores de peso para considerar cada solución.



## Casos breves

1. Usted es el director de información de un comercio importante. Recientemente, usted leyó “Spying on the Sales Floor” en el *Wall Street Journal* el 21 de diciembre de 2004. Usted observa que sus competidores utilizan la revisión y exploración de videos para analizar el comportamiento de los consumidores. ¿Debe adoptar su compañía esta herramienta (revisión de videos)? ¿Cuáles son las implicaciones estratégicas para su compañía que tiene el movimiento de sus competidores? ¿Qué oportunidades han sido creadas? ¿Amenazas?
2. Lea “Human Reengineering” de Cooper y Markus, en la *Sloan Management Review*, verano de 1995. En este artículo, Okuno trabaja para instituir una actitud positiva hacia el cambio. ¿Cómo hace él esto? Analice la importancia de la aceptación del cambio por parte de los empleados para alcanzar el éxito de la implementación de tecnología.
3. Refiérase al caso breve número 1. Usted, como director de información, cree que las ganancias por implementar la exploración de los videos en sus tiendas minoristas sobrepasarán cualquier percepción negativa de sus clientes. Su compañía es Baby’s R Us, una compañía filial de Toys R Us. Realice un estudio de factibilidad para esta inversión. Asegure incluir un listado de costos y beneficios intangibles, así como una razón para elegir una tasa de descuento. ¿Cuál es el retorno de inversión al utilizar la exploración de videos? Trate de elaborar su análisis usando como máximo 15 páginas.
4. Desarrolle un plan de proyecto y un estudio de factibilidad de programa para la inversión de la estrategia de revisión o exploración de videos en Baby’s R Us. Asegure incluir un diagrama de Gantt y PERT/CPM, así como un análisis claro de todas las tareas que deberán completarse.



## Ejercicios de equipo e individuales

1. ¿Con qué frecuencia cree usted que los temas legales juegan un papel importante en el éxito del proyecto? Piense en un ejemplo de un sistema de información potencialmente bueno o un programa que haya sido restringido o que no haya sido factible debido a los requerimientos legales.
2. Como equipo, realice una lluvia de ideas sobre formas de mejorar la aceptación del cambio por parte de los empleados que utilizan nuevos sistemas de información o procesos de negocios.
3. Piense un ejemplo de cuando la mejora de procesos del negocio es más apropiada que la reingeniería de proceso de negocios. Comparta su opinión con la clase.



## Lecturas recomendadas

*Application Development Trends* (periódico mensual). Natick, MA: Software Productivity Group, una compañía de ULLO International. Éste es nuestro periódico favorito de desarrollo de sistemas. Sigue al análisis de sistemas y a las estrategias de diseño, metodologías, CASE y otras tendencias relevantes. Visite su sitio Web en [www.adtmag.com](http://www.adtmag.com).

Grause, Donald C. y Gerald M. Weinberg. *Are Your Lights On? How to Figure Out What the Problem REALLY Is*, Nueva York: Dorset House Publishing, 1990. Aquí hay un título que realmente debe hacerlo pensar y el libro completo aborda uno de los aspectos menos publicados del análisis de sistemas: solución de problemas.

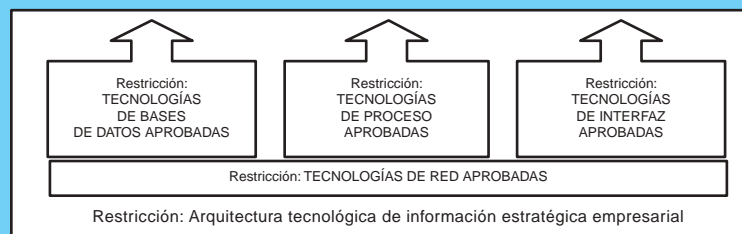
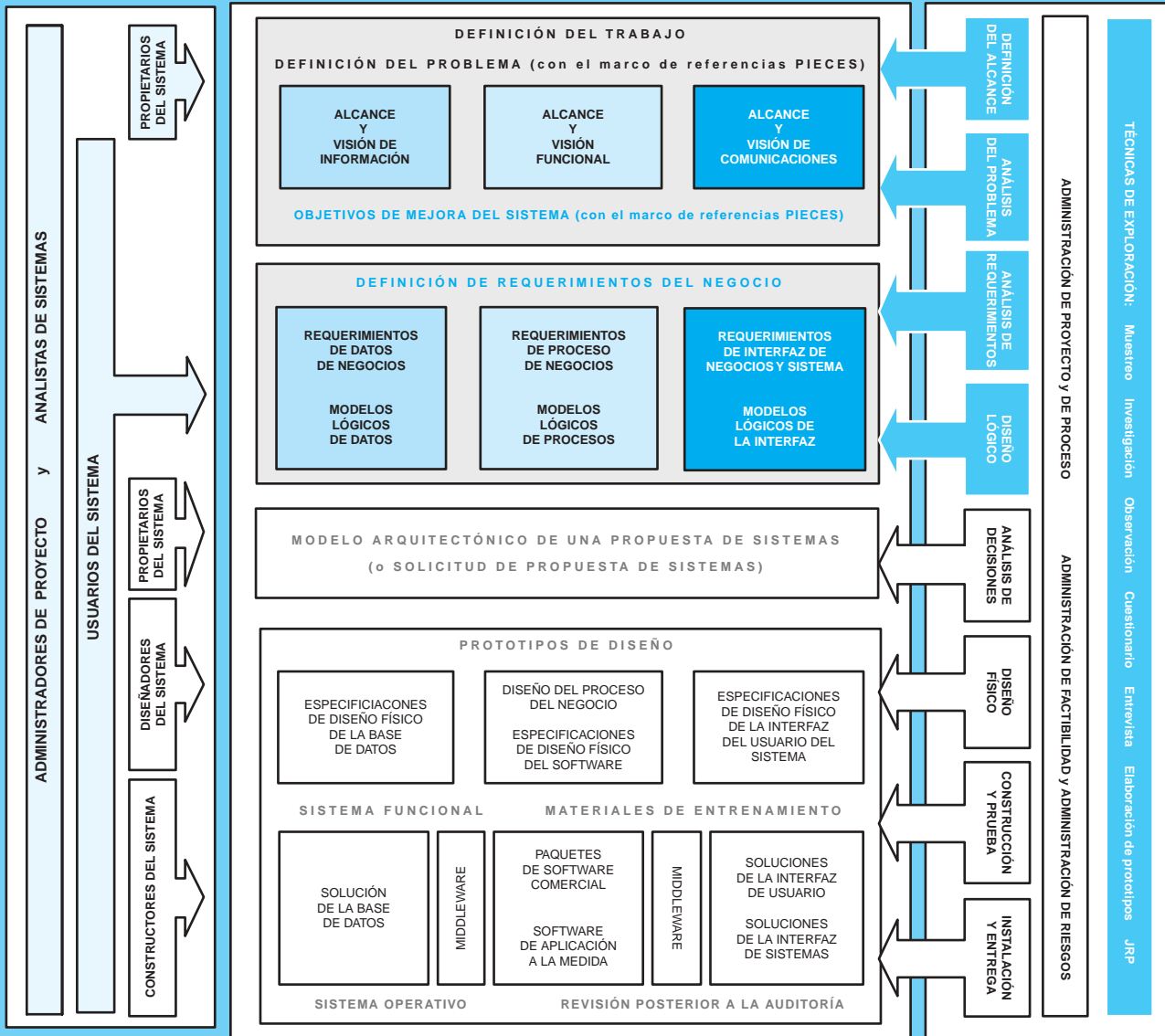
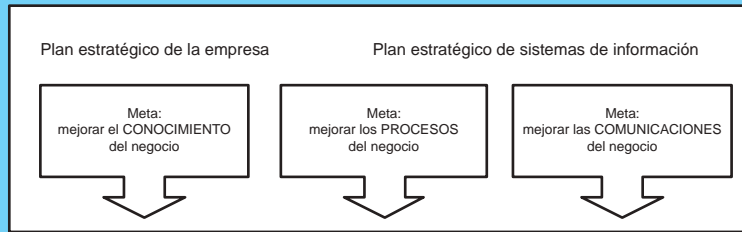
Hammer, Mike, “Reengineering Work: Don’t Automate, Obliterate”, *Harvard Business Review*, julio-agosto de 1990, pp. 104-111. El Dr. Hammer es un conocido experto en el rediseño de procesos de negocios. Este escrito inicial analiza algunos casos clásicos donde la técnica agregó valor en forma dramática al negocio.

Whetherbe, James, *Systems Analysis and Design: Best Practices*, 4a. ed., St. Paul, MN: West Publishing, 1994. Estamos en deuda con el Dr. Wetherbe por el marco de referencia PIECES.

Wood, Jane y Denise Silver, *Joint Application Design: How to Design Quality Systems in 40% Less Time*. Nueva York, John Wiley & Sons, 1989. Este libro proporciona una presentación profunda excelente del desarrollo de aplicación conjunta (JAD).

Yourdon, Edward, *Modern Structured Analysis*, Englewood Cliffs, NJ, Yourdon Press, 1989. Esta es una actualización al texto clásico de DeMarco acerca del mismo tema el cual define el estado actual de la práctica para el método de análisis estructurado.

Zachman, John, A., “A Framework for Information System Architecture”, *IBM Systems Journal*, 26, núm. 3, 1987. Este artículo presenta un marco de referencia conceptual popular de inspección de sistemas de información, así como el desarrollo de una arquitectura de información.



# 5

## Técnicas de exploración de hechos para identificación de los requerimientos

### Panorámica y objetivos del capítulo

Las técnicas efectivas para explorar hechos son cruciales en el desarrollo de los proyectos de sistemas. En este capítulo aprenderá sobre las técnicas para identificar y analizar requerimientos de sistemas de información. Usted aprenderá cómo usar varias técnicas de identificación para reunir información acerca de los problemas, oportunidades y directivas del sistema. Usted sabrá que ha comprendido las técnicas de exploración de hechos e identificación de requerimientos cuando pueda:

- Definir los requerimientos del sistema y diferenciar entre los requerimientos funcionales y los no funcionales.
- Comprender la actividad de análisis del problema y ser capaz de crear un diagrama Ishikawa (de espina de pescado) para ayudar a resolver el problema.
- Comprender el concepto de administración de requerimientos.
- Reconocer siete técnicas para explorar hechos y caracterizar las ventajas y desventajas de cada uno.
- Comprender seis guías para escuchar efectivamente
- Comprender qué preocupa por eso.
- Describir a los participantes típicos en una sesión o planeación conjunta de requerimientos (JRP) y describir sus roles.
- Completar el proceso de planeación con una sesión JRP, que incluya seleccionar y equipar el local, seleccionar a los participantes, y preparar una agenda para guiar la sesión JRP.
- Describir varios beneficios de usar JRP como una técnica de identificación de hechos.
- Describir una estrategia que explore hechos para sacar el mejor provecho de su tiempo con los usuarios finales.

## Introducción

Bob Martínez ha dedicado la mayor parte de la semana a leer. Comenzó con memorandos relacionados con el Sistema de Servicios a Miembros para comprender mejor el problema. Entonces revisó el manual de procedimientos de SoundStage para cualquier plan relacionado a los servicios a clientes y promociones. Estudió casi 100 formatos de solicitudes de miembros seleccionadas al azar, notando los tipos de información registrados en cada espacio y cuáles espacios fueron ocupados siempre, cuáles algunas veces y los que en ninguna ocasión fueron empleados. Él leyó la documentación del actual sistema de servicios a miembros de SoundStage. Revisó la información y los diagramas del anterior proyecto de desarrollo de sistemas de servicio tomando en cuenta las cosas que probablemente necesitarían ser cambiadas en el nuevo sistema. Fue un trabajo agotador. Pero al final realmente sintió que estaba empezando a comprender el sistema. Elaboró un reporte para Sandra, su jefa, acerca de los temas principales y las preguntas que necesitarían ser contestadas en la próxima reunión de planeación conjunta de requerimientos.

## Una introducción a la identificación de requerimientos

En el capítulo 3 analizamos varias fases del desarrollo de sistemas. Cada una es importante y necesaria para efectivamente diseñar, construir y, en última instancia, implementar un sistema que esté a la altura de las necesidades de los involucrados.\* Pero para desarrollar dicho sistema, debemos primero ser capaces de identificar correctamente, analizar y entender cuáles son los requerimientos de los usuarios o lo que los usuarios quieren que haga el sistema. El proceso y las técnicas que un analista de sistemas usa para identificar, analizar y entender requerimientos de sistema, son referidos como la **identificación de requerimientos**. Tal como ha sido sugerido en la página inicial del capítulo, la identificación de los requerimientos involucra principalmente a los analistas de sistemas que trabajan con los usuarios de sistemas y con los propietarios durante las primeras fases de desarrollo del sistema, con el fin de obtener una comprensión detallada de los requerimientos del negocio de un sistema de información.

¿Qué son los requerimientos del sistema? Los **requerimientos del sistema** especifican lo que el sistema de información deberá hacer o cuál propiedad o cualidad debe de tener éste. Los requerimientos del sistema que especifican lo que el sistema de información debe hacer son frecuentemente llamados **requerimientos funcionales**. Aquellos que especifican una propiedad o cualidad que el sistema debe tener con frecuencia son llamados **requerimientos no funcionales**.

El marco de referencia PIECES (tabla 5.1), presentado en el capítulo 3, proporciona una herramienta excelente para la clasificación de los requerimientos del sistema. La ventaja de clasificar los varios tipos de requerimientos es la habilidad de agrupar requerimientos para reportar, rastrear y validar los propósitos. Hacer eso ayuda a identificar los requerimientos que tal vez han sido pasados por alto.

Esencialmente, el propósito de la identificación de los requerimientos y su administración es identificar correctamente los requerimientos de CONOCIMIENTO, PROCESO y COMUNICACIÓN para los usuarios de un sistema nuevo. La falla de identificar correctamente los requerimientos del sistema puede dar como resultado en una o más de las siguientes:

- El sistema puede costar más de lo proyectado.
- El sistema puede ser entregado después de lo prometido.
- El sistema puede no estar a la altura de las expectativas de los usuarios, y esa insatisfacción puede originar que no lo usen.
- Una vez en producción, los costos de mantenimiento y mejora del sistema pueden ser excesivamente altos.
- El sistema puede ser poco confiable y tener la tendencia a fallar y tener mucho tiempo muerto.
- La reputación del equipo de TI en el grupo se mancha debido a cualquier falla, sin importar quién ha cometido el error, dicha falla será percibida como un error del equipo.

**identificación de requerimientos** Proceso y técnicas que usan los analistas de sistemas para identificar o extraer problemas de los sistemas y requerimientos de solución de la comunidad de usuarios.

**requerimiento del sistema** Algo que el sistema de información debe hacer o una propiedad que debe tener. También llamado un *requerimiento del negocio*.

**requerimiento funcional** Algo que debe hacer el sistema de información.

**requerimiento no funcional** Propiedad o cualidad que debe tener el sistema. Ejemplos incluyen seguridad, facilidad de uso, rendimiento, etcétera.

\* *Nota del R.T.:* En inglés se les llama *stakeholders* a las personas que serán afectadas de uno u otro modo por el sistema.

**TABLA 5.1** PIECES Clasificación de los requerimientos del sistema

Tipo de requerimiento no funcional	Explicación
Desempeño	<p>Los requerimientos de desempeño representan el desempeño que el sistema debe tener para satisfacer las necesidades de los usuarios.</p> <ul style="list-style-type: none"> <li>• ¿Cuál es el ritmo aceptable de producción?</li> <li>• ¿Cuál es el tiempo aceptable de respuesta?</li> </ul>
Información	<p>Los requerimientos de información representan la información que es útil al usuario en cuanto a contenido, tiempo, exactitud y formato.</p> <ul style="list-style-type: none"> <li>• ¿Cuáles son las entradas y salidas necesarias? ¿Cuándo deben suceder?</li> <li>• ¿Cuáles son los datos requeridos que deben almacenarse?</li> <li>• ¿Qué tan actualizada debe estar la información?</li> <li>• ¿Cuáles son las interfaces con los sistemas externos?</li> </ul>
Economía	<p>Los requerimientos de ahorro representan la necesidad de que el sistema reduzca costos o incremente ganancias.</p> <ul style="list-style-type: none"> <li>• ¿Cuáles son las áreas del sistema donde los costos deben reducirse?</li> <li>• ¿Cuánto deberían reducirse los costos o incrementarse las ganancias?</li> <li>• ¿Cuáles son los límites del presupuesto?</li> <li>• ¿Cuál es el cronograma del desarrollo?</li> </ul>
Control (y seguridad)	<p>Los requerimientos de control representan el ambiente en el cual el sistema debe operar, así como el tipo y grado de seguridad que debe alcanzarse.</p> <ul style="list-style-type: none"> <li>• ¿Debe controlarse el acceso al sistema o a la información?</li> <li>• ¿Cuáles son los requerimientos de privacidad?</li> <li>• La importancia de la información, ¿necesita de un manejo especial (respaldo, almacenaje fuera del lugar, etcétera)?</li> </ul>
Eficiencia	<p>Los requerimientos de eficiencia representan la capacidad del sistema para producir salidas con mínimo desperdicio.</p> <ul style="list-style-type: none"> <li>• ¿Hay pasos duplicados en el proceso que deban eliminarse?</li> <li>• ¿Hay formas de reducir el desperdicio por la manera en que el sistema usa sus recursos?</li> </ul>
Servicio	<p>Los requerimientos de servicio representan necesidades que funcionen para que el sistema sea confiable, flexible y expandible.</p> <ul style="list-style-type: none"> <li>• ¿Quiénes usarán el sistema, y dónde están localizados?</li> <li>• ¿Habrá distintos tipos de usuarios?</li> <li>• ¿Cuáles son los factores humanos adecuados?</li> <li>• ¿Qué dispositivos y materiales de entrenamiento deben incluirse en el sistema?</li> <li>• ¿Qué dispositivos y materiales de entrenamiento deben ser desarrollados y mantenidos separadamente del sistema, en programas y bases de datos de entrenamiento independientes basado en computadora (CBT)?</li> <li>• ¿Cuáles son los requerimientos de confiabilidad/disponibilidad?</li> <li>• ¿Cómo deberá ser empacado y distribuido el sistema?</li> <li>• ¿Qué documentación se necesita?</li> </ul>

**TABLA 5.2 Costos relativos para corregir un error**

Etapa en la cual se detecta el error	Relación de costos
Requerimientos	1
Diseño	3-6
Codificación	10
Desarrollo de pruebas	15-40
Pruebas de aceptación	30-70
Operación	40-1 000

El impacto en los costos puede ser impresionante. Considere por ejemplo la tabla 5.2, de Barry W. Boehm, un experto en economía de la tecnología de información.<sup>1</sup> Él estudió varios proyectos de desarrollo de software para determinar los costos de los errores en los requerimientos que fueron descubiertos más tarde en el proceso de desarrollo.

Basado en los hallazgos de Boehm, un requerimiento equivocado que no fue detectado y arreglado hasta la fase de la operación, costará 1 000 veces más de lo que costaría si fuera detectado y fijado en la fase de requerimientos. Por tanto, al definir los requerimientos del sistema, es muy importante que éstos obedezcan los siguientes lineamientos:

- *Consistentes*: Los requerimientos son no contradictorios o ambiguos.
- *Completos*: Los requerimientos describen todas las posibles entradas del sistema y las respuestas.
- *Factibles*: Los requerimientos pueden satisfacerse con los recursos disponibles y sus restricciones (el análisis de factibilidad se cubre en el capítulo 9).
- *Requeridos*: Los requerimientos se necesitan en realidad y cumplen con el objetivo del sistema.
- *Exactos*: Los requerimientos son expresados de manera correcta.
- *Rastreables*: Los requerimientos apuntan directamente hacia las funciones y características del sistema.
- *Verificables*: Los requerimientos se definen de forma en que pueden ser comprobados durante la prueba.

Este proceso puede ser difícil, frustrante y tomar mucho tiempo; por ello a menudo las empresas y los individuos toman atajos para ahorrar tiempo y dinero. Sin embargo, este criterio tan estrecho a menudo conduce a los problemas mencionados anteriormente. Ahora que comprendemos nuestro objetivo, analicemos el proceso.

## Proceso de identificación de requerimientos

El proceso de identificación de requerimientos consiste de las siguientes actividades:

- Identificación del problema y análisis.
- Identificación de los requerimientos.
- Documentación y análisis de los requerimientos.
- Administración de los requerimientos.

Ahora examinemos cada una de estas actividades en detalle:

### > Identificación del problema y análisis

Tal y como fue previamente asentado, los requerimientos resuelven problemas. Para que los analistas de sistemas sean exitosos, deberán tener habilidades para analizar problemas. Para

<sup>1</sup> Donald C. Gause y Gerald M. Weinberg, *Exploring Requirements: Quality before Design* (Nueva York: Dorset House Publishing, 1989), pp. 17-18.



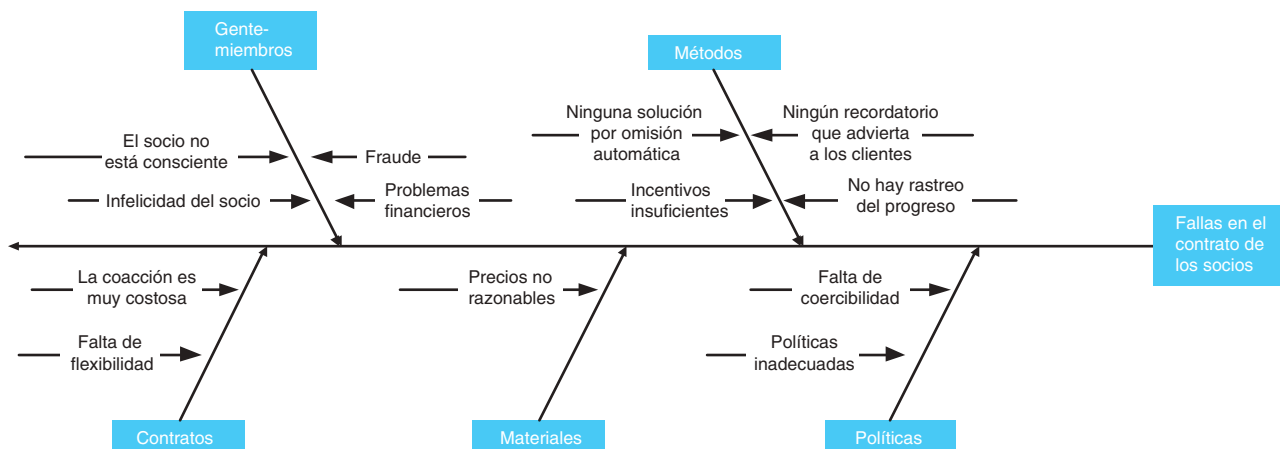
entender por completo el análisis de problemas, usemos el siguiente ejemplo: Una mujer lleva a su hijita al médico porque la niña está enferma. Lo primero que el médico intenta hacer es identificar el problema. La niña tiene dolor de oídos, fiebre y catarro. ¿Son éstos los problemas? La madre ha estado dando a la niña medicinas contra el dolor, pero la niña no se ha mejorado. Lo que sucede es que la madre está dando tratamiento para los síntomas y no para el problema real. Afortunadamente, el doctor está capacitado para analizar más allá. Después de examinar a la niña, el médico ha concluido que tiene una infección del oído, lo cual constituye la raíz de la causa de los síntomas. Ahora que el problema ha sido identificado y analizado, es tiempo de que el doctor ofrezca una solución. Normalmente, un antibiótico se prescribe para curar una infección de oído (otitis), pero para poder hacer eso, el doctor primero necesita determinar si existen contraindicaciones en la medicina que puede prescribir. ¿Qué edad tiene la niña y cuánto pesa? ¿La niña es alérgica a algún medicamento? ¿Puede tragar pastillas? Una vez que estos datos son conocidos, se puede recetar. Los analistas de sistemas aplican el mismo proceso de solución de problemas tal como lo usa un doctor, sin embargo en lugar de diagnosticar problemas médicos diagnostican problemas de sistemas.

Uno de los errores más comunes cometidos por analistas de sistemas inexpertos al tratar de analizar problemas es identificar un síntoma como un problema. Como resultado, pueden diseñar e implementar una solución que probablemente no resuelva el problema real o que pueda causar nuevos problemas. Una herramienta popular usada por equipos en desarrollo para identificar, analizar y resolver problemas, es un **diagrama Ishikawa**. El diagrama con forma de espina de pescado es el producto de Kaoru Ishikawa, pionero en la administración de procesos de calidad de los astilleros en Kawasaki, Japón. Por ello, se convirtió en uno de los padres fundadores de la administración moderna.

Al dibujar el diagrama de espina de pescado, se comienza con el nombre del problema que nos interesa. Éste se coloca del lado derecho del diagrama (o la *cabeza del pescado*). Las posibles causas del problema se dibujan como *espinas* fuera de la *columna vertebral principal*, cada una con una flecha que señale hacia la columna. Típicamente, estas “espinas” tienen cuatro categorías básicas: materiales, máquinas, mano de obra y métodos (las cuatro emes por materials, machines, manpower, y methods). Pueden adaptarse otros nombres al problema en cuestión. Otras categorías alternativas o adicionales incluyen posiciones, procedimientos, políticas y personas (las cuatro pes por places, procedures, policies, y people) o contexto, proveedores, sistemas y habilidades (las cuatro eses por surroundings, suppliers, systems, y skills).

La clave es tener de tres a seis categorías principales que abarquen todas las posibles áreas de causas. Comúnmente se llevan a cabo técnicas de lluvias de ideas (definidas posteriormente en este capítulo) para añadir causas a las espinas principales. Cuando se termina la lluvia de ideas, la espina de pescado ilustra una idea completa de todas las posibilidades acerca de lo que podría ser la causa de raíz para el problema encontrado. El equipo de desarrollo puede usar el diagrama para decidir y concordar en cuáles son las causas más probables del problema y cómo debería actuarse sobre ellas. La figura 5-1 es un ejemplo de un diagrama de espina de pescado que ilustra el problema de los miembros

**diagrama Ishikawa** Herramienta gráfica usada para identificar, explorar e ilustrar problemas, así como las causas y efectos de esos problemas. Es frecuente que se le llame diagrama de causas y efectos o diagrama de espina de pescado (porque se asemeja al esqueleto de un pez).



**FIGURA 5.1** Muestra de un diagrama de espina de pescado

de SoundStage que no cumplen con los contratos. En el diagrama, observe que el problema que debe resolverse está en el cuadro de la extrema derecha. Las cinco áreas que se han identificado como categorías de causas (Personas-miembros, Métodos, Contratos, Materiales, y Políticas) se listan en los cuadros arriba y abajo del *esqueleto de pescado* y se conectan con flechas (espinas) que apuntan a la columna vertebral del pescado. Las causas reales del problema para cada categoría se ilustran como flechas que apuntan hacia la flecha de la categoría (espinas).

## > Identificación de los requerimientos

Dada una comprensión de los problemas, el analista de sistemas puede comenzar a definir los requerimientos. Para que los analistas de sistemas de la actualidad tengan éxito en definir los requerimientos de los sistemas, deben tener habilidad en los métodos efectivos para recopilar información: la exploración. La **exploración** es una técnica que se usa a través del ciclo completo de desarrollo, pero es muy crítica en la fase de análisis de requerimientos. Una vez que se ha terminado la exploración, se usarán herramientas tales como los casos de uso (prácticos), los modelos de datos, los modelos de procesos, y los modelos de objetos para documentar los hechos, y se obtendrán conclusiones a partir de estos hechos. Usted va a aprender acerca de estas herramientas y también cómo documentar los requerimientos derivados de la exploración en los capítulos subsiguientes de este libro.

Los hechos están en el dominio de la aplicación de negocios y de los usuarios finales. Por tanto, el analista debe recolectar estos hechos con objeto de aplicar con efectividad las técnicas y las herramientas de la documentación. Durante la fase del análisis de sistemas, el analista aprende acerca del vocabulario, los problemas, las oportunidades, las restricciones, los requerimientos, y las prioridades de un negocio y de un sistema.

¿Qué tipo de hechos deben recolectarse? Ciertamente sería benéfico si tuviéramos un marco de referencia que nos ayude a determinar qué hechos es necesario recolectar, independientemente del proyecto en el cual estemos trabajando. Afortunadamente, tenemos este marco. Tal como resulta, los hechos que describen a cualquier sistema de información también corresponden muy bien a los bloques de construcción que se realzan en la página principal del capítulo. Observe que las técnicas de exploración se usan en el desarrollo inicial de sistemas para identificar los alcances y la visión de la información, la funcionalidad, y la comunicación, así como para identificar el proceso de conocimiento del negocio, y los requerimientos de comunicaciones del sistema.

## > Requerimientos de documentación y de análisis

Cuando el analista de sistemas realiza actividades de exploración, es importante que reúna o documente la información recopilada (es decir, los *requerimientos del borrador*) de una manera organizada, inteligible y significativa. Estos documentos iniciales van a suministrar la dirección de las técnicas de modelación que el analista de sistemas va a usar para analizar los requerimientos y determinar los requerimientos correctos del proyecto. Una vez que éstos han sido identificados, el analista de sistemas formaliza los requerimientos presentándolos en un documento que será revisado y aprobado por los usuarios.

**Documentación de los requerimientos del borrador** Los analistas de sistemas usan diferentes herramientas para documentar sus identificaciones iniciales en forma de borrador. Ellos escriben los *casos de uso* para describir las funciones del sistema desde la perspectiva de los usuarios externos y de una manera y con una terminología que el usuario entienda. Se usan *tablas de decisión* para documentar las complejas políticas de negocios de una organización y las reglas para tomar decisiones, y se usan *tablas de requerimientos* para documentar cada requerimiento específico. Cada una de estas herramientas se examina con más detalle posteriormente en el capítulo.

**Análisis de los requerimientos** Muy frecuentemente, las actividades de exploración producen requerimientos que entran en conflicto entre sí. Esto se debe a que los requerimientos son solicitados por muchas fuentes diferentes y cada persona tiene sus propias opiniones y deseos acerca de la funcionalidad y las características del nuevo sistema. El objetivo de la actividad del análisis de requerimientos es identificar y resolver los problemas con los requerimientos y alcanzar un consenso sobre cualesquiera modificaciones que satisfagan a los involucrados. El proceso tiene que ver con los requerimientos “iniciales” recopilados de los involucrados. Generalmente estos requerimientos

**exploración** Proceso formal del uso de la investigación, juntas, entrevistas, cuestionarios, muestreo, y otras técnicas para recabar información acerca de los problemas, los requerimientos, y las preferencias del sistema. También se llama *recopilación de información o recolección de datos*.

están incompletos y se documentan de una manera informal en instrumentos tales como los casos de uso, las tablas, y los reportes. El enfoque de esta etapa radica en alcanzar consensos sobre las necesidades de los involucrados; en otras palabras, el análisis debe responder a la pregunta: “¿Tenemos los requerimientos correctos del sistema para el proyecto?” Es inevitable que los requerimientos del borrador contengan muchos problemas, tales como:

- Requerimientos faltantes
- Requerimientos en conflicto
- Requerimientos infactibles
- Traslape de requerimientos
- Requerimientos ambiguos

Estos tipos de problemas de requerimientos son muy comunes en muchos de los documentos de requerimientos que se escriben actualmente. Si se dejan sin resolver, puede ser muy costoso repararlos posteriormente en el ciclo de desarrollo.

Anteriormente se mencionó que los involucrados deben concordar en los requerimientos resultantes del sistema: por tanto hay un proceso inevitable de negociación entre los involucrados durante el análisis. Si los involucrados múltiples proponen requerimientos que están en conflicto mutuo o si los requerimientos propuestos son demasiado ambiciosos, los involucrados deben negociar, frecuentemente bajo la guía del analista de sistemas, para concordar sobre cualesquiera modificaciones o simplificaciones de los requerimientos del sistema. También deben concordar sobre la condición crítica y la prioridad de los requerimientos. Esto es crucial para asegurar el éxito del esfuerzo de desarrollo.

Las actividades de exploración y del análisis de requerimientos están asociadas muy cercanamente entre sí y de hecho con frecuencia están entrelazadas. Si se encuentra que los requerimientos descubiertos durante el proceso de exploración son problemáticos, el analista puede seguir adelante y realizar actividades de análisis sobre los elementos seleccionados con objeto de resolver los problemas antes de seguir provocando necesidades y deseos adicionales del sistema.

Este capítulo se centra principalmente en el lado de los negocios de los requerimientos o, en otras palabras, los requerimientos lógicos, pero es importante observar que existen requerimientos técnicos adicionales que son de naturaleza física. Los ejemplos de requerimientos técnicos incluyen la especificación de un paquete de software o de una plataforma de hardware que se requieran. Estos tipos de requerimientos se estudiarán con profundidad en el capítulo 9.

**Formalización de los requerimientos** Generalmente los requerimientos del sistema se documentan de una manera formal para comunicarlos a los involucrados clave del sistema. Este documento sirve como contrato entre los propietarios del sistema y el equipo de desarrollo sobre lo que se va a suministrar en términos de un nuevo sistema. Así, puede someterse a muchas revisiones y repasos antes de que todos concuerden y autoricen el contenido. No hay un nombre o formato estándar para este documento. De hecho, muchas organizaciones usan nombres diferentes tales como declaración de requerimientos, especificación de requerimientos, definición de requerimientos, especificación funcional, y otros, y generalmente el formato se ajusta a las necesidades de la organización. Las compañías que proveen sistemas de información y software al gobierno de los Estados Unidos deben usar las convenciones de formato y terminología especificadas en el documento de estándares publicado por el gobierno MIL-STD-498.<sup>2</sup> Muchas organizaciones han creado sus propios estándares adaptados de MIL-STD-498 debido a que son muy completos y porque muchas personas ya están familiarizadas con ellos. En este libro usaremos el término **documento de definición de requerimientos**, y la figura 5.2 proporciona un ejemplo de uno. Por favor observe que este documento será consolidado con otra información de proyecto para formar la declaración de requerimientos, que es el producto final de la fase de análisis de requerimientos. Un documento de definición de requerimientos deberá consistir de lo siguiente:

---

#### documento de definición de requerimientos

Documento formal que comunica los requerimientos de un sistema propuesto a involucrados clave y sirve como un contrato del proyecto de sistemas. Sinónimo de declaración de requerimientos, especificación de requerimientos, y especificación funcional.

<sup>2</sup> MIL-STD-498 es un estándar que fusiona DOD-STD-2167A con DOD-STD-7935A para definir un conjunto de actividades y documentación adecuadas para el desarrollo de los sistemas de armas y de los sistemas automatizados de información.

**FIGURA 5.2**

Muestra de un esbozo de definición de requerimientos

**DOCUMENTO DE DEFINICIÓN DE REQUERIMIENTOS**

1. Introducción
    - 1.1. Propósito
    - 1.2. Fondo
    - 1.3. Alcance
    - 1.4. Definiciones, acrónimos y abreviaturas
    - 1.5. Referencias
  2. Descripción general del proyecto
    - 2.1. Requerimientos funcionales
  3. Requerimientos y restricciones
    - 3.1. Requerimientos funcionales
    - 3.2. Requerimientos no funcionales
  4. Conclusión
    - 4.1. Aspectos relevantes
- Apéndice (opcional)

- Las funciones y los servicios que el sistema debe proveer.
- Los requerimientos no funcionales, incluyendo los aspectos del sistema, las características, y los atributos.
- Las restricciones que limitan el desarrollo del sistema o bajo las cuales debe operar el sistema.
- Información acerca de otros sistemas con las cuales el sistema debe tener una interfase.

¿Quién va a leer el documento de definición de requerimientos? Probablemente este documento es el más leído y más referenciado de toda la documentación de proyecto. Los propietarios y los usuarios del sistema lo usan para especificar sus requerimientos y cambios que puedan surgir. Los gerentes lo usan para preparar los planes y las estimaciones de proyecto, y los desarrolladores para entender lo que se requiere y desarrollar pruebas para validar el sistema. Con esto en mente, es importante observar que los requerimientos se leen más frecuentemente que lo que se escriben. Por lo tanto, el tomarse tiempo para escribirlos correctamente, de manera concisa y clara no sólo va a ahorrar tiempo en términos del calendario, sino que también es más eficiente en costos y reduce el riesgo de errores costosos en los requerimientos. Por lo tanto, realizar la valoración de los requerimientos es un paso necesario para lograr ese objetivo. La validación de los requerimientos se realiza en un borrador final del documento de definición de requerimientos después que se han solicitado todas las entradas por parte de los propietarios y usuarios del sistema. El propósito de esta actividad es que el analista de sistemas se asegure de que los requerimientos se escriban correctamente. Ejemplos de errores que el analista de sistemas podría encontrar son:

- Modelos del sistema que contienen errores.
- Errores tipográficos o gramaticales.
- Conflicto de requerimientos.
- Requerimientos ambiguos o mal redactados.
- Falta de conformidad con los estándares de calidad requeridos para el documento.

### > Administración de los requerimientos

Durante la vida útil del proyecto es muy común que surjan nuevos requerimientos y que cambien los existentes después de haber aprobado un documento de definición de requerimientos. Algunos estudios han mostrado que durante la vida del proyecto cambiarán tanto como 50 por ciento o más de los requerimientos antes de que el sistema se ponga en producción. Obviamente que esto puede ser un gran dolor de cabeza para el equipo de

desarrollo. Para ayudar a aliviar los múltiples problemas asociados con los requerimientos que cambian, es necesario realizar una **administración de requerimientos**. Ésta abarca las políticas, los procedimientos, y los procesos que gobiernan el manejo del cambio de un requerimiento. En otras palabras, se especifica cómo debe presentarse una solicitud de cambio, cómo se analiza en cuanto el impacto sobre el alcance, el calendario, y el costo, cómo se aprueba o se rechaza, y cómo se implementa el cambio si se aprueba.

**administración de requerimientos** Proceso de administrar los cambios de los requerimientos.

## Técnicas de exploración

En esta sección presentamos siete técnicas comunes de exploración:

- Muestreo de la documentación, las formas y las bases de datos existentes.
- Investigación y visitas al sitio.
- Observación del ambiente de trabajo.
- Cuestionarios.
- Entrevistas.
- Propuestas de prototipos.
- Planeación conjunta de requerimientos.

Generalmente un analista aplica varias de estas técnicas durante un proyecto de sistemas individual. Para poder seleccionar la técnica más adecuada a usarse en cualquier situación dada, los analistas de sistemas necesitan aprender las ventajas y desventajas de cada una de las técnicas de exploración.

**La ética de la exploración** Durante la exploración, los analistas de sistemas frecuentemente encuentran o analizan información que es de naturaleza sensible. Podría ser un archivo de la estructura de precios de una compañía aeroespacial para una licitación de contrato o aun los perfiles de los empleados, incluyendo salarios, historial de su desempeño, historial médico, y planes de carrera. Los analistas deben tener mucho cuidado para proteger la seguridad y la privacidad de los hechos o datos que se les hayan confiado. Muchas personas y organizaciones en esta atmósfera altamente competitiva están buscando una “ventaja” para sacar ganancia. Los analistas de sistemas descuidados que dejan documentos delicados a la vista sobre sus escritorios, o que discuten en público los datos confidenciales podrían causar un gran daño a la organización o a las personas. Si este tipo de datos cayera en las manos equivocadas, el analista de sistemas podría perder el respeto, la credibilidad, o la confianza de los usuarios y de la gerencia. En algunos casos, el analista sería responsable de la intrusión en la privacidad de una persona y podría resultar legalmente responsable.

La mayoría de las corporaciones hacen todos los esfuerzos para asegurarse de que lleven los negocios de una manera ética porque la ley puede exigirselos. Ha habido muchos casos en los que las corporaciones han incurrido en fuertes multas por no llevar apropiadamente los negocios. A este efecto, muchas empresas requieren que sus empleados asistan a seminarios de entrenamiento anuales sobre la ética de la compañía, y refuerzan el aprendizaje exhibiendo banderas o letreros que contienen el código de conducta de la compañía y las declaraciones de ética en el lugar de trabajo en localidades muy visibles. Las políticas de ética de la compañía pueden estar en un formato de fotocopiado que se distribuye a todos los empleados, o en las páginas Web de la compañía, haciéndolas fácilmente accesibles a los empleados sin importar dónde se encuentren actualmente. Las políticas de ética documentan la conducta esperada y requerida. Las violaciones de estas políticas podrían conducir a una acción disciplinaria o aun a un despido. La ética juega un papel crucial en la exploración.

### > Muestreo de la documentación, los formatos y los archivos existentes

Al estudiar un sistema existente, los analistas de sistemas desarrollan una buena percepción del sistema mediante la consulta de la documentación, los formatos y los archivos existentes. Un buen analista siempre sabe obtener hechos de la documentación existente antes que de las personas.

**Recolección de hechos a partir de la documentación existente** ¿Qué tipo de documentos pueden enseñarle algo acerca de un sistema? El primero que el analista desea examinar es el organigrama, el cual sirve para identificar a los propietarios y usuarios individuales clave para un proyecto y sus relaciones de reporte. El analista también desea rastrear el historial que originó el proyecto. Para lograr esto, él deberá recolectar y revisar documentos que describan el problema. Éstos incluyen:

- Memorando interno de la oficina, estudios, minutas, notas en el buzón de sugerencias, quejas de los clientes, y reportes que documenten el área problema.
- Registros contables, revisiones del desempeño, de la medición del trabajo, y otros reportes operativos programados.
- Solicitudes de proyecto de los sistemas de información: pasados y presentes.

Además de los documentos que describen el problema, generalmente hay documentos que describen la función de negocios que está siendo estudiada o diseñada. Estos documentos pueden incluir:

- La declaración de la misión y el plan estratégico de la compañía.
- Los objetivos formales de las unidades de la organización que se están estudiando.
- Los manuales de políticas que pueden imponer restricciones sobre cualquier sistema propuesto.
- Los procedimientos operativos estándar (Standard operating procedures, SOP), las descripciones de puestos, o las instrucciones de tareas para las operaciones cotidianas específicas.
- Formas llenadas que representan las transacciones reales en los diferentes puntos del ciclo de procesamiento.
- Muestras de las bases de datos manuales y computarizadas.
- Muestras de las pantallas y los reportes manuales y computarizados.

También, los analistas frecuentemente revisan la documentación de estudios y diseños del sistema anterior realizados por los analistas y consultores anteriores. Esta documentación puede incluir:

- Diferentes tipos de diagramas de flujo y diagramas.
- Diccionarios o repositorios del proyecto.
- Documentación de diseño, tales como entradas, salidas, y bases de datos.
- Documentación de programa.
- Manuales de operación de las computadoras y manuales de entrenamiento.

Toda la documentación recolectada deberá analizarse para determinar si la información está o no actualizada. La documentación obsoleta no debe descartarse; sin embargo, los analistas deben tener en mente que será necesaria exploración adicional para verificar o actualizar los hechos recolectados. ¿Qué es lo que busca el analista en todo este material? Los elementos que puedan seleccionarse de estos documentos incluyen:

- Los síntomas y (posiblemente) las causas del problema.
- Qué personas en la organización entienden el problema.
- Las funciones de negocios que soportan al sistema presente.
- El tipo de datos que el sistema debe recolectar y reportar.
- Elementos en la documentación que el analista no entiende y que es necesario cubrir en entrevistas.

**Técnicas de muestreo de documentos y archivos** Debido a que sería impráctico estudiar todas las ocurrencias de todos los formatos o registros en un archivo o base de datos, normalmente los analistas de sistemas usan técnicas de **muestreo** para obtener una vista transversal suficientemente amplia para prever lo que puede pasar en el sistema. El analista de sistemas debe tratar de muestrear suficientes formatos que representen la naturaleza y la complejidad total de los datos. Los analistas con experiencia evitan las pifias de muestrear formatos en blanco: éstos dicen poco acerca de cómo se usa el formato en realidad, cuándo no se usa o con qué frecuencia se le da un mal uso. Cuando se estudian documentos o registros de una tabla de base de datos, los analistas deben estudiar suficientes muestras para identificar todas las condiciones y excepciones posibles de procesamiento. Pueden usarse técnicas de muestreo estadístico para determinar si el tamaño de la muestra es suficientemente grande para ser representativa de la población total de los registros o los documentos.

---

**muestreo** Proceso de recolectar una muestra representativa de documentos, formas y registros.

**TABLA 5.3** Tabla parcial de los factores de certeza

Certeza deseada	Factor de certeza
95%	1.960
90	1.645
80	1.281

Hay muchos factores y aspectos en el muestreo, y ésta es una buena razón para tomar un curso de introducción a la estadística. Una fórmula simple y confiable para determinar el tamaño de la muestra es:

$$\text{Tamaño de la muestra} = 0.25 \times (\text{factor de certidumbre/error aceptable})^2$$

El factor de certidumbre es un valor que simplemente puede buscarse en tablas estadísticas basándose en la certidumbre deseada de que la muestra seleccionada será representativa de la población total. Véase la tabla 5.3 para un ejemplo específico.

Suponga que un analista quiere tener una certeza del 90 por ciento de que una muestra de facturas no va a contener variaciones no muestreadas. El tamaño de muestra,  $TM$ , se calcula como sigue:

$$TM = 0.25(1.645/0.10)^2 = 68$$

El analista deberá muestrear 68 facturas para obtener la exactitud deseada. Si se desea un mayor nivel de certeza, se necesita un mayor número de facturas.

Si el analista sabe por experiencia que 1 de cada 10 facturas varía con respecto a la norma, entonces puede reemplazarse el heurístico 0.25 por  $p(1 - p)$  donde  $p$  es la proporción de facturas con variancias:

$$TM = p(1 - p)(1.645/0.10)^2$$

Con el uso de esta fórmula, el analista puede reducir el número de muestras requeridas para obtener la exactitud deseada:

$$TM = 0.10(1 - 0.10)(1.645/0.10)^2 = 25$$

¿Cómo se escogen las 25 facturas? Dos técnicas comúnmente usadas son el muestreo aleatorio y el muestreo estratificado. El **muestreo aleatorio** consiste en seleccionar los datos de la muestra en forma aleatoria o sin importar cómo se seleccionen. Por tanto, simplemente escogemos al azar 25 facturas basándonos en el tamaño de muestra calculado antes. El **muestreo estratificado** es un enfoque bien pensado y sistemático dirigido a reducir la variancia de los datos de la muestra. Para los archivos computarizados, el muestreo estratificado puede realizarse escribiendo un programa sencillo. Por ejemplo, suponga que nuestras facturas están en una base de datos que tiene un volumen de aproximadamente 250 000 facturas. Recuerde que nuestro tamaño de muestra debe incluir 25 facturas. Simplemente escribiremos un programa que imprima un registro de cada 10 000 (= 250 000/25). Para los documentos y archivos manuales, podríamos ejecutar un esquema similar.

### > Investigación y visitas al sitio

Una segunda técnica de exploración es investigar a fondo el dominio del problema. La mayoría de los problemas no son del todo únicos. Otras personas los han resuelto antes que nosotros. Muchas veces las organizaciones contactan o realizan visitas de sitio con compañías que saben que han experimentado antes problemas similares. Si esas compañías tienen “voluntad de compartir”, puede obtenerse información valiosa que puede ahorrar mucho tiempo y costo en el proceso de desarrollo.

Las revistas especializadas en computación y los libros de referencia son una buena fuente de información. Pueden suministrar información sobre cómo otros han resuelto problemas similares. Con los recientes avances en el ciberespacio, los analistas rara vez tienen que dejar sus escritorios para hacer una investigación.

---

**muestreo aleatorio** Técnica de muestreo que se caracteriza por no contar con ningún patrón o plan predeterminado para seleccionar los datos de la muestra.

---

**muestreo estratificado** Técnica sistemática de muestreo que intenta reducir la variancia de las estimaciones al dispersar el muestreo (por ejemplo, con una selección de los documentos o los registros mediante una fórmula) y evitar estimaciones muy altas o muy bajas.

La exploración del Internet y del intranet vía la computadora personal puede suministrar incontables cantidades de información.

Un tipo similar de investigación consiste en visitar a otras compañías o departamentos que han encarado problemas similares. Las membresías en sociedades profesionales tales como la Association for Information Technology Professionals (AITP) o la Association for Information Systems (AIS), entre otras, pueden suministrar una red de contactos útiles.

## > Observación del ambiente de trabajo

La observación es una de las técnicas más efectivas de recolección de datos para aprender acerca de un sistema. La **observación** consiste en que el analista de sistemas se convierte en un observador de las personas y de las actividades con objeto de aprender acerca del sistema. Frecuentemente se usa esta técnica cuando se cuestiona la validez de los datos recolectados mediante otros métodos o cuando la complejidad de ciertos aspectos del sistema impide obtener una clara explicación por parte de los usuarios finales.

**Recolección de hechos por observadores en el trabajo** Aun con un plan de observación bien concebido, el analista de sistemas no está seguro de que la observación tenga éxito. La siguiente historia, que aparece en un libro de Gerald M. Weinberg llamado *Rethinking Systems Analysis and Design*, nos da un ejemplo entretenido y al mismo tiempo excelente de algunas de las pifias de la observación.<sup>3</sup>

### La paradoja del ferrocarril

Aproximadamente a 30 millas de Ciudad Gótica se ubica la comunidad que hace viajes de ida y regreso del pueblo de los Suburbios. Cada mañana, miles de pueblerinos tomaban el Ferrocarril Central para trabajar en Ciudad Gótica. Cada tarde, el Ferrocarril Central los regresaba a sus esposas, hijos y perros que los esperaban.

El pueblo de los Suburbios era un suburbio próspero, y a muchas de las esposas les gustaba dejar a los niños y a los perros y pasar la tarde en Ciudad Gótica con sus parejas. Preferían iniciar su tarde de cena y de teatro visitando los fastuosos mercados en Ciudad Gótica. Pero había un problema. Para tener tiempo para ir de compras, una pueblerina tendría que salir de Ciudad Gótica a las 2:30 o 3:00 en la tarde. A esa hora, ningún tren del Ferrocarril Central hacía parada en el Pueblo de los Suburbios.

Algunos pueblerinos observaban que ciertamente un tren Central pasaba por su estación a las 2:30, pero no paraba. Decidieron solicitar al ferrocarril que el tren se programara para parar en el pueblo de los Suburbios. Rápidamente encontraron apoyo en su campaña de puerta en puerta para solicitar votos. Cuando se mandó la solicitud por correo, contenía 253 firmas. Aproximadamente tres semanas después, el comité solicitante recibió la siguiente carta del Ferrocarril Central:

Apreciable comité:

Gracias por su continuo interés en las operaciones del Ferrocarril Central. Nosotros tomamos en serio nuestro cometido de suministrar un servicio que dé respuesta a todas las personas que viven entre nuestras rutas, y apreciamos mucho la retroalimentación en todos los aspectos de nuestro negocio. En respuesta a su solicitud, nuestro representante de servicio al cliente visitó la estación del pueblo de los Suburbios en tres días diferentes, cada vez a las 2:30 de la tarde. Aunque él observó con mucho cuidado, *en ninguna de las tres ocasiones había pasajeros esperando por un tren dirigido al sur.*

Solamente podemos concluir que no hay una demanda real de una parada dirigida al sur a las 2:30, y por lo tanto debemos declinar lamentablemente su solicitud.

Atentamente,

Agente de servicio al cliente

Ferrocarril Central

**observación** Técnica de exploración en la cual el analista de sistemas participa u observa a una persona que realiza actividades para aprender acerca del sistema.

<sup>3</sup> Gerald M. Weinberg, *Rethinking Systems Analysis and Design*, pp. 23-24. Derechos reservados © 1988, 1982 por Gerald M. Weinberg. Reimpreso con autorización de Dorset House Publishing, 353 W. 12th St., Nueva York, NY 10014 (212-620-4053/800-DH-BOOKS/www.dorsethouse.com). Todos los derechos reservados.



¿Cuáles son las lecciones que se aprenden de la historia anterior? Una es que es necesario usar la técnica de exploración apropiada para el problema a la mano. La observación, en este caso, fue una elección incorrecta. ¿Por qué alguien estaría esperando por el tren de las 2:30 cuando toda la gente del pueblo sabía que el tren no para? Una segunda lección que debe aprenderse es verificar los resultados de la exploración con los usuarios. Basándose en la retroalimentación del usuario, usted puede identificar que es necesario intentar con otras técnicas de exploración para recabar información adicional. Nunca debe saltar a las conclusiones.

**Ventajas y desventajas de la observación** La observación puede ser una técnica de exploración muy útil y benéfica siempre que usted tenga la habilidad para observar todos los aspectos del trabajo que los usuarios están realizando y que el trabajo se esté realizando de la manera acostumbrada. Usted debe estar conciente de los pros y los contras de la técnica de observación. Las ventajas y las desventajas incluyen:

### Ventajas

- Los datos recabados basándose en la observación pueden ser muy confiables. Algunas veces se realizan observaciones para verificar la validez de los datos obtenidos directamente de las personas.
- El analista de sistemas puede ver exactamente lo que se está haciendo. Algunas veces es difícil explicar claramente con palabras las tareas complejas. A través de la observación, el analista de sistemas puede identificar las tareas que se han omitido o que se han descrito con inexactitud por otras técnicas de exploración. También, el analista puede obtener datos que describan el ambiente físico de la tarea (por ejemplo, la disposición física, el tránsito, la iluminación, el nivel de ruido).
- La observación es relativamente barata en comparación con otras técnicas de exploración. Generalmente otras técnicas requieren mayor tiempo de liberación del empleado y más gastos de copiado.
- La observación permite que el analista de sistemas haga mediciones del trabajo.

### Desventajas

- Ya que la gente generalmente se siente incómoda cuando está siendo vigilada, inconscientemente puede comportarse de una manera diferente que cuando está siendo observada.
- El trabajo que se esté observando tal vez no incluya el nivel de dificultad o de volumen normalmente experimentado durante ese tiempo.
- Algunas actividades de sistemas pueden tener lugar en horas estrambóticas, causando una inconveniencia de programación para el analista de sistemas.
- Las tareas que se observan están sujetas a diferentes tipos de interrupciones.
- Algunas tareas no siempre serán desempeñadas en la manera en que las observa el analista de sistemas. Por ejemplo, el analista de sistemas pudo haber observado cómo una compañía llenaba varias solicitudes de los clientes. Sin embargo, los procedimientos que el analista de sistemas observó pudieron haber sido los pasos usados para llenar varias solicitudes regulares de los clientes. Si cualquiera de estas solicitudes hubiera sido una solicitud especial (por ejemplo, de bienes que normalmente no se guardan en existencia), el analista de sistemas habría observado un conjunto diferente de procedimientos que se ejecutan.
- Si las personas han estado realizando tareas de una manera que viole los procedimientos estándar de operación, temporalmente pueden desempeñar su trabajo de manera correcta mientras que usted los observa. En otras palabras, la gente puede permitirle ver lo que ellos quieren que usted vea.

**Lineamientos de la observación** ¿Cómo obtiene hechos el analista de sistemas a través de la observación? ¿Se llega simplemente al sitio de observación y se comienza a

registrar todo lo que se ve? No. Antes deberá haber mucha preparación. El analista debe determinar cómo van en verdad a capturarse los datos. ¿Será necesario tener formatos especiales para el registro rápido de los datos? ¿Se molestarán las personas que están siendo observadas porque alguien las observe y registre sus acciones? ¿Cuándo deben observarse los periodos bajos, normales y altos de las operaciones de la tarea? El analista de sistemas debe identificar el momento ideal para observar un aspecto específico del sistema.

Un analista deberá planear observar un sitio cuando hay una carga de trabajo típica. Una vez que se ha observado una carga de trabajo típica, pueden hacerse observaciones durante los periodos pico para recopilar información para medir los efectos causados por el incremento de la carga. Como parte de la observación del analista, deberán obtenerse muestras de documentos o de formatos que usan quienes son observados.

Las técnicas de muestreo que se estudiaron anteriormente también son útiles para la observación. En este caso, la técnica se llama **muestreo del trabajo**; en ella puede realizarse un gran número de observaciones a intervalos aleatorios. Esta técnica es menos incómoda para las personas que están siendo observadas porque el periodo de observación no es continuo. Al usar el muestreo del trabajo, un analista necesita definir con anticipación las operaciones del trabajo que va a observar. Entonces es necesario calcular el tamaño de la muestra tal como se hizo para el muestreo de documentos y archivos. El analista deberá realizar muchas observaciones aleatorias, teniendo cuidado de observar las actividades en diferentes momentos del día. Contando el número de ocurrencias de cada operación durante las observaciones, el analista se dará cuenta de cómo pasan sus días los empleados.

Los siguientes lineamientos son clave para refinar las habilidades de observación:

- Determine el quién, qué, dónde, cuándo, porqué y cómo de la observación.
- Obtenga el permiso para observar por parte de los supervisores o los gerentes.
- Informe a quienes van a ser observados acerca del propósito de la observación.
- Mantenga un bajo perfil.
- Tome notas durante o inmediatamente después de la observación.
- Revise las notas de la observación con las personas apropiadas.
- No interrumpa a las personas que están trabajando.
- No se centre mucho en actividades triviales.
- No haga suposiciones.

**Viviendo el sistema** En este tipo de observación el analista de sistemas desempeña activamente el papel del usuario por un corto periodo. Ésta es una de las maneras más efectivas para aprender acerca de los problemas y los requerimientos del sistema. Al “ponerse los zapatos” del usuario, un analista de sistemas obtiene rápidamente una impresión de lo que experimenta el usuario y lo que tiene que hacer para realizar el trabajo. Este tipo de juego del papel le da al analista de sistemas una enseñanza de primera mano de los procesos y las funciones del negocio, así como de los problemas y retos asociados con éstos.

## > Cuestionarios

Otra técnica de exploración es conducir encuestas mediante **cuestionarios**. El cuestionario puede producirse en gran cantidad y distribuirse a los encuestados, quienes entonces pueden llenar el cuestionario cuando tengan tiempo. Los cuestionarios le permiten al analista recolectar hechos de un gran número de personas al tiempo que se mantienen respuestas uniformes. Al enfrentarse a una audiencia grande, ninguna otra técnica de exploración puede tabular los mismos hechos con la misma eficacia.

**Recolección de hechos mediante el uso de cuestionarios** Frecuentemente los analistas de sistemas han criticado el uso de los cuestionarios. Muchos analistas de sistemas objetan que las respuestas carecen de información confiable y útil. Sin embargo, los cuestionarios pueden ser un medio efectivo para recopilar los hechos, y muchas de estas críticas pueden atribuirse al uso inapropiado o poco efectivo de los cuestionarios. Antes de

### muestreo del trabajo

Técnica de exploración que consiste en tomar un gran número de observaciones para intervalos aleatorios.

**cuestionario** Documento que permite al analista recabar información y opiniones de los encuestados.

usar los cuestionarios, un analista deberá entender los pros y los contras asociados con su uso:

### Ventajas

- La mayoría de los cuestionarios pueden responderse rápidamente. La gente puede completar y devolver los cuestionarios con toda comodidad.
- Los cuestionarios son un medio relativamente barato de recopilar datos provenientes de un gran número de personas.
- Los cuestionarios permiten que las personas mantengan el anonimato. Por lo tanto, es más probable que las personas suministren los hechos reales en vez de decirle lo que piensan que su jefe querría que ellos hicieran.
- Las respuestas pueden tabularse y analizarse rápidamente.

### Desventajas

- Con frecuencia el número de encuestados es bajo.
- No existe garantía de que una persona responda o se expone a todas las preguntas.
- Los cuestionarios tienden a ser inflexibles. No hay oportunidad de que el analista de sistemas obtenga información voluntaria o parafrasee las preguntas que pudieron haber sido mal interpretadas.
- No es posible que el analista de sistemas observe y analice el lenguaje corporal del encuestado.
- No hay una oportunidad inmediata para aclarar una respuesta vaga o incompleta a cualquier pregunta.
- Los buenos cuestionarios son difíciles de preparar.

**Tipos de cuestionarios** Hay dos formatos para los cuestionarios: el formato libre y el formato fijo. Los **cuestionarios de formato libre** se diseñan para permitir a los usuarios que ejerciten más libertad o flexibilidad en sus respuestas a cada pregunta.

Aquí hay dos ejemplos de las preguntas con formato libre:

- ¿Qué reportes recibe actualmente y cómo los usa?
- ¿Hay algún problema con estos reportes (por ejemplo, son inexactos, la información es insuficiente, o son difíciles de leer, usar o ambas cosas)? Si es así, por favor explique.

Es obvio que las respuestas a estas preguntas pueden ser difíciles de tabular. También es posible que las respuestas de quienes responden no concuerden con las preguntas formuladas. Con objeto de asegurar respuestas útiles en los cuestionarios de formato libre, el analista deberá expresar las preguntas en oraciones simples y no usar palabras —tales como *bueno*— que puedan interpretarse en forma distinta por diferentes encuestados. El analista también deberá formular preguntas que puedan responderse con tres o menos oraciones. De otra manera, el cuestionario puede consumir más tiempo que el que el encuestado desea dedicar.

El segundo tipo de cuestionario es el de formato fijo. Los **cuestionarios de formato fijo** son más rígidos ya que requieren que el usuario seleccione una respuesta de un conjunto de respuestas posibles previamente definido. Dada cualquier pregunta, el encuestado debe escoger de las respuestas disponibles. Esto hace que los resultados sean mucho más fáciles de tabular. Por otro lado, el encuestado no puede suministrar información adicional que pudiera resultar valiosa.

Hay tres tipos de preguntas de formato fijo:

1. Para las *preguntas de opción múltiple*, se le dan al encuestado varias respuestas de las cuales escoger. Deberá decirse al encuestado si puede seleccionarse más de una respuesta. Algunas preguntas de opción múltiple permiten respuestas muy breves de formato libre cuando no son aplicables ninguna de las respuestas estándar. Los ejemplos de las preguntas de formato fijo de opción múltiple son:

¿Piensa usted que las órdenes atrasadas ocurren con demasiada frecuencia?  
 SÍ       NO

¿Es útil el reporte actual de cuentas por cobrar que usted recibe?  
 SÍ       NO

Si la respuesta es no, por favor explique.

---

### cuestionario de formato libre

Cuestionario diseñado para ofrecer al encuestado más laxitud en la respuesta. Se formula una pregunta, y el encuestado registra la respuesta en el espacio provisto después de la pregunta.

---

### cuestionario de formato fijo

Cuestionario que contiene preguntas que requieren la selección de una respuesta entre respuestas disponibles predefinidas.

2. Para las *preguntas de calificación*, se le da al encuestado un enunciado y se le pide que use las respuestas suministradas para emitir una opinión. Para evitar el provocar un sesgo, deberá haber un número igual de calificaciones positivas y negativas. El siguiente es un ejemplo de una pregunta de calificación con formato fijo:

La implementación de los descuentos en las cantidades causaría un aumento en las órdenes de los clientes.

- Definitivamente de acuerdo
- De acuerdo
- Sin opinión
- En desacuerdo
- Definitivamente en desacuerdo

3. Para las *preguntas de jerarquización*, se le dan al encuestado varias respuestas posibles, que deben jerarquizarse en orden de preferencia o de experiencia. Un ejemplo de una pregunta de jerarquización con formato fijo es:

Jerarquice las siguientes transacciones de acuerdo con el tiempo que usted ocupa en procesarlas:

- \_\_\_\_\_ órdenes de los clientes nuevos
- \_\_\_\_\_ cancelaciones de órdenes
- \_\_\_\_\_ órdenes modificadas
- \_\_\_\_\_ pagos

**Cómo desarrollar un cuestionario** Los buenos cuestionarios pueden ser difíciles de desarrollar. El siguiente procedimiento puede ser útil para el desarrollo de una encuesta efectiva:

1. Determine qué hechos y opiniones deben recolectarse y de quién debe obtenerlas. Si el número de personas es grande, considere el uso de un grupo de encuestados más pequeño seleccionado aleatoriamente.
2. Basándose en los hechos y en las opiniones buscadas, determine si las preguntas de formato libre o fijo darán las mejores respuestas. Frecuentemente se usa un formato combinado que permite la aclaración opcional de formato libre de las respuestas de formato fijo.
3. Escriba las preguntas. Examínelas en cuanto a errores de construcción y posibles malas interpretaciones. Asegúrese de que las preguntas no revelen su sesgo personal o sus opiniones. Revise las preguntas.
4. Ensaye las preguntas en una pequeña muestra de encuestados. Si sus encuestados tuvieran problemas con éstas o si las respuestas no fueran útiles, revise las preguntas.
5. Duplique y distribuya el cuestionario.

## > Entrevistas

**entrevista** Técnica de exploración mediante la cual el analista de sistemas recolecta información de las personas a través de la interacción cara a cara.

Generalmente se reconoce que la entrevista personal es la técnica de exploración más importante y de uso más frecuente. Las **entrevistas** personales consisten en preguntar los requerimientos a través de una interacción directa cara a cara. Las entrevistas pueden usarse para alcanzar alguno o todos los objetivos siguientes: indagar hechos, verificar hechos, aclarar hechos, generar entusiasmo, hacer que se involucre el usuario final, identificar los requerimientos, y solicitar ideas y opiniones. Hay dos papeles que se asumen en una entrevista. El analista de sistemas es el *entrevistador*, responsable de la organización y la conducción de la entrevista. El usuario del sistema o el propietario del sistema es el *entrevistado*, a quien se le pide que responda a una serie de preguntas.

Puede haber uno o más entrevistadores, entrevistados o ambos. En otras palabras, las entrevistas pueden conducirse de uno a uno o de muchos a muchos. Desafortunadamente, muchos analistas de sistemas son malos entrevistadores. En esta sección usted aprenderá a conducir entrevistas adecuadas.

**Recolección de hechos por los usuarios que hacen la entrevista** Las personas son el elemento más importante de un sistema de información. Más que cualquier otra cosa, la gente quiere estar en cosas activas. Ninguna otra técnica de exploración pone tanto énfasis en la gente como las entrevistas. Pero las personas tienen valores, prioridades, opiniones, motivaciones, y personalidades diferentes. Por lo tanto, para usar la técnica de la entrevista, un analista de sistemas debe poseer buenas habilidades de relaciones humanas para tratar con efectividad a diferentes tipos de personas. Al igual que con otras técnicas de exploración, la entrevista no es el mejor método para todas las situaciones; tiene sus ventajas y sus desventajas, que deben sopesarse con las de otras técnicas de exploración para cada situación de exploración:

#### Ventajas

- Las entrevistas dan al analista una oportunidad para motivar al entrevistado para que responda libre y abiertamente a las preguntas. Al establecer una armonía, el analista de sistemas puede darle al entrevistado una percepción de que está contribuyendo activamente al proyecto de sistemas.
- Las entrevistas permiten que el analista de sistemas intente obtener más retroalimentación del entrevistado.
- Las entrevistas permiten que el analista de sistemas adapte o parafrasee las preguntas para cada persona.
- Las entrevistas le dan al analista una oportunidad para observar la comunicación no oral del entrevistado. Un buen analista de sistemas puede ser capaz de obtener información al observar los movimientos corporales y las expresiones faciales del entrevistado así como al escuchar las respuestas orales a las preguntas.

#### Desventajas

- La entrevista es un enfoque de exploración que consume mucho tiempo, y por tanto es muy costosa.
- El éxito de las entrevistas depende mucho de las habilidades en relaciones humanas del analista de sistemas.
- Una entrevista puede ser impráctica debido a la ubicación del entrevistado.

**Tipos y técnicas de la entrevista** Hay dos tipos de entrevistas, la no estructurada y la estructurada. Las **entrevistas no estructuradas** se caracterizan porque contienen preguntas generales que permiten al entrevistado dirigir la conversación. Frecuentemente, este tipo de entrevista se sale del camino, y el analista debe estar preparado para redirigir la entrevista hacia el principal objetivo o tema. Por esta razón, las entrevistas no estructuradas generalmente no funcionan bien para el análisis y el diseño de sistemas. Las **entrevistas estructuradas** consisten en que el entrevistador formula preguntas específicas diseñadas para obtener información específica de parte del entrevistado. Dependiendo de las respuestas del entrevistado, el entrevistador dirigirá preguntas adicionales para obtener una aclaración o una ampliación. Algunas de estas preguntas pueden ser planeadas y otras, espontáneas.

Las entrevistas no estructuradas tienden a que se formulen **preguntas de respuesta abierta**. Este tipo de preguntas da a los entrevistados una gran libertad en sus respuestas. Un ejemplo de una pregunta de extremo abierto es: “¿Por qué no está satisfecho con el reporte de cuentas incobrables?” Las entrevistas estructuradas tienden a que se formulen más **preguntas de respuesta cerrada** que están diseñadas para suscitar respuestas cortas y directas de parte del entrevistado. Ejemplos de preguntas así son: “¿Está recibiendo el reporte de las cuentas incobrables a tiempo?” y “¿Contiene el reporte de cuentas incobrables información exacta?” Desde un punto de vista realista, la mayoría de las preguntas se sitúan entre los dos extremos.

---

#### entrevista no estructurada

Entrevista que se conduce solamente con un objetivo o tema general en mente y con pocas preguntas específicas, si es que las hay. El entrevistador cuenta con el entrevistado para proveer un marco y dirigir la conversación.

---

#### entrevista estructurada

Entrevista en la cual el entrevistador tiene un conjunto específico de preguntas para hacérselas al entrevistado.

---

#### pregunta de respuesta abierta

Pregunta que permite al entrevistado responder de cualquier manera que parezca apropiada.

---

#### pregunta de respuesta cerrada

Pregunta que restringe las respuestas ya sea a selecciones específicas o a respuestas cortas y directas.

## > Cómo conducir una entrevista

El éxito de un analista de sistemas depende, al menos parcialmente, de la capacidad para entrevistar. Una entrevista de éxito incluye la selección de las personas apropiadas para la entrevista, la preparación extensa para la entrevista, la conducción apropiada de la entrevista, y el seguimiento de la misma. Aquí examinamos cada uno de estos pasos con más detalle. Supongamos que el analista ha identificado la necesidad de una entrevista y ha determinado exactamente qué tipos de hechos y opiniones se necesitan.

**Selección de los entrevistados** El analista de sistemas debe entrevistar a los usuarios finales del sistema de información que se está estudiando. Un organigrama formal ayudará a identificar a estas personas y sus responsabilidades. Antes de la entrevista, el analista debe intentar aprender tanto como sea posible acerca de cada persona (sus fortalezas, sus temores, sus prejuicios y sus motivaciones) entonces la entrevista puede armarse para que considere las características de la persona.

El analista debe hacer una cita con el entrevistado y nunca simplemente aparecerse. La cita debe limitarse a un lapso de entre media hora y una hora. Entre mayor sea el nivel de dirección del entrevistado, deberá programarse menos tiempo. Si el entrevistado es un trabajador de oficina, de servicio o un obrero, el analista debe obtener el permiso del supervisor antes de programar la entrevista. También es importante asegurarse de que el lugar de la entrevista esté disponible durante el tiempo programado. Las entrevistas nunca deben conducirse en presencia de los compañeros de oficina del analista o de los colegas del entrevistado.

**Preparación para la entrevista** La preparación es la clave de una entrevista exitosa. Un entrevistado puede detectar fácilmente cuando un entrevistador no está preparado y puede resentir la falta de preparación porque se desperdicia tiempo valioso. Cuando se hace la cita, el entrevistado deberá ser notificado acerca del tema de la entrevista. Para asegurarse de que se cubran todos los aspectos pertinentes del tema, el analista debe preparar un *guión de entrevista*, el cual es una lista de verificación de preguntas específicas que el entrevistador le hará al entrevistado. El guión de entrevista también puede contener preguntas de seguimiento que se formularán solamente si las respuestas a otras preguntas justifican las respuestas adicionales. En la figura 5.3 se presenta una muestra de un guión de entrevista. Observe que la agenda se prepara cuidadosamente con el tiempo específico asignado a cada pregunta. También debe apartarse tiempo para formular preguntas de seguimiento y para redirigir la entrevista. Las preguntas deben seleccionarse y expresarse con cuidado. La mayoría de las preguntas comienzan con el tipo acostumbrado de formulación de quién, qué, cuándo, dónde, por qué y cuánto. Deben evitarse los siguientes tipos de preguntas:

- *Preguntas cargadas*, tales como “¿Tenemos que tener estas dos columnas en el reporte?” La pregunta transmite la opinión personal del entrevistado sobre el tema.
- *Preguntas con intención*, tales como “Usted no va a usar este CÓDIGO DE OPERADOR, ¿verdad?” La pregunta conduce a que el entrevistado responda, “No, por supuesto que no”, independientemente de su opinión real.
- *Preguntas sesgadas*, tales como “¿Cuántos códigos necesitamos para la CLASIFICACIÓN DE ALIMENTOS en el ARCHIVO DE INVENTARIOS? Pienso que 20 deberían de cubrirlo.” Estos tipos de preguntas sesgadas van a influir en el entrevistado.

Los entrevistadores siempre deben evitar preguntas amenazantes o críticas. El propósito de la entrevista es investigar, no evaluar ni criticar. Lineamientos adicionales para las preguntas incluyen:

- Use un lenguaje claro y conciso.
- No incluya su opinión como parte de la pregunta.
- Evite preguntas largas o complejas.
- Evite las preguntas amenazantes.
- No use “usted” cuando se refiera a un grupo de personas.

Entrevistado: Jeff Bentley, Gerente de cuentas por cobrar Fecha: 19 de enero de 2003 Hora: 1:30 p. m. Lugar: Sala 223, Edificio de administración Tema: Política actual de investigación de crédito		
Tiempo asignado	Pregunta u objetivo del administrador	Respuesta del entrevistado
1 a 2 min.	<b>Objetivo</b> Comienza la entrevista: <ul style="list-style-type: none"> <li>Nos presentamos</li> <li>Gracias Sr. Bentley por su valioso tiempo</li> <li>Enunciar el propósito de la entrevista: obtener una comprensión de las políticas existentes de investigación de crédito.</li> </ul>	
5 min.	<b>Pregunta 1</b> ¿Qué condiciones determinan si se aprueba una solicitud de crédito del cliente? <b>Seguimiento</b>	
5 min.	<b>Pregunta 2</b> ¿Cuáles son las posibles decisiones o acciones que podrían tomarse una vez que estas condiciones han sido evaluadas? <b>Seguimiento</b>	
3 min.	<b>Pregunta 3</b> ¿Cómo se notifica a los clientes cuando no se aprueba su solicitud de crédito? <b>Seguimiento</b>	
1 min.	<b>Pregunta 4</b> Después que se aprueba una nueva solicitud de crédito y se coloca en el archivo que contiene las solicitudes que pueden llenarse, un cliente puede pedir que se haga una modificación a la solicitud. ¿Tendría que pasar ésta nuevamente por la aprobación de crédito si el costo total de la nueva solicitud sobrepasa al costo original? <b>Seguimiento</b>	
1 min.	<b>Pregunta 5</b> ¿Quiénes son las personas que realizan las investigaciones de crédito? <b>Seguimiento</b>	
1 a 3 min.	<b>Pregunta 6</b> ¿Puedo obtener el permiso para hablar con estas personas para aprender específicamente cómo llevan a cabo el proceso de investigación de crédito? <b>Seguimiento</b> Si así es: ¿Cuál sería el momento apropiado para reunirme con cada uno de ellos?	
1 min.	<b>Objetivo</b> Termino de la entrevista: <ul style="list-style-type: none"> <li>Agradezca al Sr. Bentley por su cooperación y asegúrele que estará recibiendo una copia de lo que se obtuvo durante la entrevista.</li> </ul>	
21 minutos	Tiempo asignado para preguntas y objetivos	
9 minutos	Tiempo asignado para preguntas de seguimiento y redirección	
30 minutos	Tiempo asignado para la entrevista (1:30 p.m. a 2:00 p.m.)	
<b>Comentarios generales y notas:</b>		

**FIGURA 5.3** Muestra de un guión para entrevista

**Conducción de la entrevista** Respete el tiempo de su entrevistado. Arréglese para hacer la entrevista. Generalmente esto significa que usted se arreglará de manera diferente para entrevistar a los gerentes que para entrevistar a los trabajadores en el muelle de carga. Si la entrevista se celebra en una sala de reuniones en vez de la oficina del entrevistado, llegue temprano para asegurarse de que esté acondicionada apropiadamente.

Inicie la entrevista agradeciendo al entrevistado. Enuncie el propósito y la duración de la entrevista y cómo van a usarse los datos recabados. Entonces monitoree el tiempo de modo que cumpla su promesa.

Formule preguntas de seguimiento. Intente hasta que entienda los requerimientos del sistema. Pregunte especialmente acerca de las condiciones de excepción. Formule preguntas del tipo ¿qué pasa si?, tales como “¿Qué sucede si el cheque no pasa?” o “¿Qué pasa si un producto no está en existencia?”

Escuche con atención, observe al entrevistado, y tome notas acerca de sus respuestas verbales y no verbales. Es muy importante mantener la entrevista en su curso; esto implica anticipar la necesidad de adaptar la entrevista, si es necesario. Frecuentemente las preguntas pueden ser omitidas si han sido respondidas anteriormente o pueden eliminarse si se determina que son irrelevantes, basándose en respuestas anteriores.

Aquí hay un conjunto de reglas que debe seguir un entrevistador:

#### Haga

- Vístase adecuadamente.
- Sea cortés.
- Escuche cuidadosamente.
- Mantenga el control de la entrevista.
- Explore.
- Observe los gestos y la comunicación no oral.
- Sea paciente.
- Mantenga al entrevistado en calma.
- Mantenga su autocontrol.
- Termine a tiempo.

#### Evite

- Suponer que una respuesta esté terminada o que no lleva a ningún lado.
- Revelar pistas orales y no orales.
- Usar la jerga.
- Revelar sus sesgos personales.
- Hablar en lugar de escuchar.
- Suponer cualquier cosa acerca del tema o el entrevistado.
- Uso de la grabadora: una señal de habilidades deficientes para escuchar.

Concluya la entrevista expresando su aprecio y proveyendo respuestas a las preguntas interpuestas por el entrevistado. La conclusión es muy importante para mantener la armonía y la confianza con el entrevistado.

**Seguimiento de la entrevista** Para ayudar a mantener una buena armonía y confianza con los entrevistados, el entrevistador deberá mandarles un memorando que resuma la entrevista. Este documento deberá recordarles a los entrevistados sus contribuciones al proyecto de sistemas y permitirles la oportunidad de aclarar malas interpretaciones en que el entrevistador haya incurrido durante la entrevista. Además, a los entrevistados deberá dárseles la oportunidad de ofrecer información adicional que hayan omitido durante la entrevista.

**Cómo escuchar** Cuando la mayoría de la gente habla acerca de las habilidades de comunicación, piensan en hablar y escribir. Rara vez se menciona la habilidad de escuchar, pero puede ser la más importante durante el proceso de la entrevista. Con objeto de conducir una entrevista exitosa, el entrevistador debe diferenciar entre oír y escuchar: “Oír es reconocer que alguien está hablando, escuchar es entender lo que el orador quiere comunicar”.<sup>4</sup>

En realidad hemos estado condicionados durante toda la vida a no escuchar. Considere por ejemplo, como podemos ignorar a nuestros hermanos y hermanas que pelean mientras disfrutamos de nuestro CD favorito, cómo aprendemos a estudiar bloqueando nuestras distracciones tales como compañeros de cuarto ruidosos. Hemos aprendido a no escuchar, pero también podemos aprender cómo escuchar en forma efectiva y productiva.

<sup>4</sup> Thomas R. Gildersleeve, *Successful Data Processing Systems Analysis* (Englewood Cliffs, NJ: Prentice Hall, 1978), p. 93.



Al trabajar con usuarios que tratan de resolver sus problemas, los analistas pueden encontrar que hacer que los usuarios se comuniquen puede ser difícil. Los siguientes lineamientos pueden ayudar a abrir los canales de comunicación.

- *Llegue a la sesión con una actitud positiva.* El entrevistador debe aprovechar cualquier situación de la mejor manera, y verla como una experiencia divertida y placentera.
- *Haga que la otra persona se tranquilice.* El presentar una actitud agradable y animada puede ayudar a que la persona se relaje. El entrevistador debe comenzar hablando acerca de los intereses o las aficiones de la persona. Algunas veces el mostrar un interés en la vida personal del entrevistado puede servir para romper el hielo y hacer que la persona se relaje más.
- *Deje que la otra persona se entere que usted está escuchando.* El entrevistador siempre deberá mantener el contacto visual cuando escuche y asentir con la cabeza o emitir un “ajá” para reconocer lo que la persona está diciendo. Una buena postura y el inclinarse hacia delante le dirá al orador que el entrevistador está realmente interesado en lo que la persona está diciendo.
- *Haga preguntas.* El entrevistador deberá formular preguntas para asegurarse de que entiende claramente lo que la persona está diciendo o para aclarar un punto. Esto mostrará que el entrevistador está escuchando; también le dará a la otra persona la oportunidad de ampliar la respuesta.
- *No suponga nada.* Una de las peores cosas que puede hacer un entrevistador es actuar como si tuviera prisa. Por ejemplo, si un entrevistador supone lo que la otra persona va a decir y la interrumpe y termina la oración, posiblemente pierda lo que la persona trataba de decir e irrite al entrevistado. O si éste es interrumpido porque el entrevistador ya ha oído la información y piensa que no es aplicable al tema de la entrevista, podría perderse una pieza valiosa de información. ¡No suponga nada! El comentarista de la televisión Art Linkletter aprendió esta lección en su popular show de la televisión, *Kids Say the Darndest Things*, cuando le preguntó a un niño una cuestión filosófica:

En mi show una vez hice que un niño me dijera que él quería ser un piloto de aerolínea. Le pregunté qué haría si todos los motores se pararan en medio del Océano Pacífico. Él dijo: “Primero les diría a todos que se aprieten los cinturones de seguridad, y luego buscaría mi paracaídas y saltaría”.

Mientras que el auditorio se destornillaba de risa, centré mi atención en el joven para ver si se estaba pasando de listo. Las lágrimas que brotaban de sus ojos me alertaron de su pesar más que cualquier cosa que pudiera haber dicho, así es que le pregunté por qué dijo tal cosa. Su respuesta reveló la sólida lógica de un niño: “¡Voy a conseguir gasolina...Voy a regresar!”<sup>5</sup>

- *Tome notas.* El proceso de tomar notas sirve para dos propósitos. Primero, al grabatear notas breves mientras que la otra persona está hablando, usted le da la impresión de que lo que dice es suficientemente importante como para ser escrito. Segundo, las notas ayudan al entrevistador a recordar más tarde los puntos principales de la reunión.

**El lenguaje corporal y la proxemia** ¿Qué es el lenguaje de cuerpo, y por qué debería preocuparse un analista de sistemas acerca de esto durante el proceso de la entrevista? El **lenguaje corporal** es toda la información no verbal que una persona está comunicando. El lenguaje de cuerpo es una forma de comunicación no verbal que todos usamos y de la cual generalmente no estamos conscientes.

Los estudios han determinado un hecho desconcertante: del total de los sentimientos de una persona, sólo el 7 por ciento se comunica verbalmente (con palabras), mientras que el 38 por ciento se comunica por el tono de voz empleado y el 55 por ciento por las expresiones faciales y del cuerpo. Si usted solamente escucha las palabras de alguien, está perdiendo la mayor parte de lo que la persona tiene que decir.

---

**lenguaje corporal** Información no verbal que comunicamos.

<sup>5</sup> Donald Walton, *Are You Communicating? You Can't Manage without It*. (Nueva York: McGraw-Hill, 1989), p. 31.

Para esta discusión nos centraremos solamente en tres aspectos del lenguaje corporal: la expresión facial, el contacto visual, y la postura. La *expresión facial* implica que algunas veces usted puede entender cómo se siente una persona observando las expresiones en su rostro. Muchas emociones comunes son expresiones faciales fácilmente reconocibles asociadas con éstas. Sin embargo, la cara es una de las partes más controladas del cuerpo. Algunas personas que se dan cuenta de que sus expresiones con frecuencia revelan lo que están pensando, son muy buenas para disimularlas.

Otra forma de comunicación no verbal es el *contacto visual*. Éste es el aspecto menos controlado de la expresión facial. ¿Alguna vez ha hablado con alguien que no lo mire directamente? ¿Qué fue lo que sintió? Una falta continua del contacto visual puede indicar incertidumbre. Una mirada normal en general dura de tres a cinco segundos; sin embargo, el tiempo de contacto visual directo debe aumentar con la distancia. Es necesario que los analistas tengan cuidado de no usar un contacto visual excesivo con los usuarios que parecen sentirse amenazados de modo que no los intimiden más. El contacto visual directo puede causar sentimientos fuertes, ya sean positivos o negativos, en otras personas.

La *postura* es el aspecto menos controlado del cuerpo. Como tal, la postura corporal contiene mucha información para el analista astuto. Los miembros de un grupo que están de acuerdo tienden a exhibir la misma postura. Un buen analista vigilará los cambios de postura del auditorio que pudieran indicar ansiedad, desacuerdo, o aburrimiento. Normalmente un analista deberá mantener una posición corporal “abierta”, transmitiendo el acercamiento, la aceptación y la receptividad. En circunstancias especiales, el analista puede escoger usar un ángulo de confrontación con la cabeza hacia el frente o en un ángulo de 90 grados con respecto a otra persona con objeto de establecer control y dominación.

Además de la información comunicada por el lenguaje corporal, las personas también se comunican vía la proxemia. La **proxemia**, la relación entre las personas y el espacio alrededor de ellas, es un factor de comunicaciones que el analista experimentado puede controlar.

La gente todavía tiende a ser muy territorial con su espacio. Observe dónde se sientan sus compañeros de clase en alguno de sus cursos que no tienen pupitres asignados. O la próxima vez que usted participe en una conversación con alguien, muévase deliberadamente mucho más cerca o más lejos de la persona y vea lo que pasa. Un buen analista reconoce cuatro zonas espaciales:

- *Zona íntima*: menos de 1.5 pies.
- *Zona personal*: de 1.5 pies a 4 pies.
- *Zona social*: de 4 pies a 12 pies.
- *Zona pública*: más de 12 pies.

Ciertos tipos de comunicaciones tienen lugar solamente en alguna de estas zonas. Por ejemplo, un analista conduce la mayoría de las entrevistas con los usuarios del sistema en la zona personal. Pero puede ser necesario que el analista regrese a la zona social si el usuario muestra cualquier signo (lenguaje corporal) de estar incómodo. Algunas veces el contacto visual creciente puede compensar una distancia larga que no puede modificarse. Muchas personas usan las fronteras de la zona social como una distancia de “respeto”.

Hemos examinado algunas de las maneras informales en las que la gente comunica sus sentimientos y sus reacciones. Un buen analista usará toda la información disponible, no sólo la comunicación escrita o verbal de otros.

## > Elaboración de prototipos de identificación

Otro tipo de técnica de exploración es la elaboración de prototipos, la cual se introdujo en el capítulo 3 para usarse en el desarrollo rápido de aplicación (rapid application development, RAD). Como recordará, el concepto detrás de la elaboración de prototipos es la construcción de un pequeño modelo de trabajo de los requerimientos del usuario o un diseño propuesto para un sistema de información. Este tipo de elaboración de prototipo en general es una técnica de diseño, pero el enfoque puede aplicarse anteriormente al

---

**proxemia** Relación entre las personas y el espacio a su alrededor.



ciclo de vida de desarrollo del sistema para realizar el análisis de exploración y de requerimientos. El proceso de construcción de un prototipo para el propósito de identificar los requerimientos se denomina la **elaboración del prototipo de identificación**.

Frecuentemente se aplica la elaboración de prototipos de identificación a los proyectos de desarrollo de sistemas, en especial en los casos donde el equipo de desarrollo está tendiendo problemas para definir los requerimientos del sistema. La filosofía es que los usuarios van a reconocer sus requerimientos cuando los vean. Es importante que el prototipo se desarrolle rápidamente de modo que pueda usarse durante el proceso de desarrollo. En general, se construyen prototipos sólo en las áreas donde los requerimientos no se entienden claramente. Esto implica que puede excluirse mucha funcionalidad deseable e ignorarse el aseguramiento de la calidad. También, los requerimientos no funcionales tales como el desempeño y la confiabilidad pueden ser menos restrictivos que lo que sería para el producto final. Con frecuencia se usan tecnologías diferentes de las que se emplean para el software final en la construcción de los prototipos de identificación. En estos casos, lo más probable es que los prototipos se descarten cuando se termine el sistema. Este enfoque de “desechar” se usa principalmente para recabar información y desarrollar ideas para el concepto de sistema. Tal vez no se entiendan con claridad muchas áreas de un sistema propuesto, o algunos aspectos pueden ser un desafío técnico para los desarrolladores. La creación de prototipos de identificación permite a los desarrolladores así como a los usuarios entender mejor y refinar los aspectos implicados en el desarrollo del sistema. Esta técnica ayuda a minimizar el riesgo de entregar un sistema que no satisfaga las necesidades de los usuarios o que no pueda cumplir los requerimientos técnicos.

La elaboración de prototipos de identificación tiene sus ventajas y desventajas, lo que debe sopesarse con aquellas de otras técnicas de exploración para cada situación de exploración:

#### Ventajas

- Permite que los usuarios y los desarrolladores experimenten con el software y desarrollen una comprensión de cómo podría trabajar el sistema.
- Ayuda a determinar la factibilidad y la utilidad del sistema antes de incurrir en altos costos de desarrollo.
- Sirve como un mecanismo de entrenamiento para los usuarios.
- Ayuda a construir los planes y escenarios de prueba del sistema para usarse al último en el proceso de pruebas del sistema.
- Puede minimizar el tiempo invertido en la exploración y ayudar a definir requerimientos más estables y confiables.

#### Desventajas

- Puede ser necesario entrenar a los desarrolladores en el enfoque de elaboración de prototipos.
- Los usuarios pueden desarrollar expectativas poco realistas basándose en el desempeño, la confiabilidad, y las características del prototipo. Los prototipos solamente pueden simular la funcionalidad del sistema y su naturaleza es incompleta. Debe tenerse cuidado en educar a los usuarios acerca de este hecho para no desorientarlos.
- La elaboración de prototipos puede prolongar el programa de desarrollo y aumentar los costos de desarrollo.

### > Planeación conjunta de requerimientos

Muchas organizaciones están usando la sesión grupal de trabajo como un sustituto de entrevistas separadas y numerosas. Un ejemplo del enfoque de la sesión grupal de trabajo es la **planeación conjunta de requerimientos (joint requirements planning, JRP)**, en la cual se conducen reuniones de grupo altamente estructuradas con el objeto de identificar y analizar problemas y definir los requerimientos del sistema. Éstas y otras técnicas similares en general requieren de un extenso entrenamiento para trabajar como está pensado. Sin embargo, pueden reducir significativamente el tiempo invertido en la exploración en una o más fases del ciclo de vida. La JRP se está haciendo cada vez más común en la planeación de sistemas y en el análisis de sistemas para obtener un consenso del grupo sobre los problemas, objetivos, y requerimientos. En esta sección, usted aprenderá acerca

#### elaboración del prototipo de identificación

Acto de construir un representante a pequeña escala o modelo de trabajo de los requerimientos de los usuarios con objeto de identificar o verificar esos requerimientos.

#### planeación conjunta de requerimientos (JRP)

Proceso mediante el cual se conducen reuniones de grupo altamente estructuradas con el propósito de analizar problemas y definir requerimientos.

de los participantes en una sesión JRP y sus papeles. También estudiaremos qué hacer con la planeación y conducción de una sesión JRP, las herramientas y las técnicas que se usan durante una sesión de JRP, y los beneficios que se logran a través de JRP.

**Los participantes en JRP** Las sesiones de planeación conjunta de requerimientos incluyen una amplia variedad de participantes y de papeles. Se espera que cada participante asista y participe activamente en la sesión completa de JRP. Examinemos los diferentes tipos de personas que participan en una sesión típica de JRP y sus papeles:

- **Patrocinador:** Cualquier sesión exitosa de JRP requiere que una sola persona, llamada *patrocinador*, sirva como su campeón. Normalmente esta persona es un individuo que está en la dirección (*no* en la administración TI o IS) y que tiene autoridad sobre los diferentes departamentos y usuarios que van a participar en el proyecto de sistemas. El patrocinador da todo su apoyo al proyecto de sistemas al alentar a los usuarios designados a que participen en forma activa y por su propia voluntad en la sesión de JRP. Recordando el enfoque del “compromiso progresivo” al desarrollo de sistemas, es el patrocinador (propietario del sistema) quien toma las decisiones finales en relación con la dirección de avance o no avance del proyecto.

El patrocinador juega un papel visible durante una sesión de JRP al dar inicio a la junta introduciendo a los participantes. Frecuentemente, el patrocinador también hará comentarios finales sobre la sesión y trabaja íntimamente con el líder del JRP para planear la sesión al ayudar a identificar a las personas provenientes de la comunidad de usuarios que deberán asistir y al determinar la fecha y la ubicación de la sesión de JRP.

- **Facilitador:** Las sesiones de JRP incluyen a una persona que juega el papel de líder o facilitador. Generalmente, el *facilitador de JRP* es responsable de conducir todas las sesiones que se celebren para un proyecto de sistemas. Esta persona es alguien que tiene excelentes habilidades de comunicación, posee la capacidad para negociar y resolver conflictos de grupo, tiene un buen conocimiento del negocio, buenas habilidades para la organización, es imparcial con las decisiones que se van a enfrentar, y no reporta a ninguno de los participantes de la sesión de JRP.

Algunas veces es difícil encontrar una persona dentro de la compañía que posea todas estas cualidades. Entonces, con frecuencia las compañías deben suministrar un extenso entrenamiento para JRP o contratar un experto externo a la organización para cumplir con este papel. Muchos analistas de sistemas están entrenados para ser facilitadores de JRP.

El papel del facilitador de JRP es planear la sesión de JRP, conducir la sesión, y dar seguimiento a los resultados. Durante la sesión, el facilitador es responsable de encabezar la discusión, alentando a los asistentes para que participen activamente, resolviendo los conflictos clave que puedan surgir, y asegurándose de que se alcancen las metas y los objetivos de la reunión. La responsabilidad del facilitador de JRP es establecer las reglas de campo que se seguirán durante la reunión y asegurarse de que los participantes se adhieran a estas reglas.

- **Usuarios y gerentes:** La planeación conjunta de requerimientos incluye a varios participantes provenientes de los sectores de usuarios y gerencial de una organización a quienes se les concede tiempo de comisión de sus actividades cotidianas para dedicarse a una participación activa en las sesiones de JRP. Estos participantes son seleccionados por el patrocinador, quien debe ser cuidadoso para asegurarse de que cada persona tenga el conocimiento de los negocios para contribuir durante las sesiones de exploración. El patrocinador del proyecto debe ejercer su autoridad y estímulo para asegurarse de que estas personas se dedican a una participación activa.

Una sesión típica de JRP puede incluir cualquier número de personas desde uno relativamente pequeño de usuarios/gerentes hasta una docena o más. El papel de los usuarios durante una sesión de JRP es comunicar con efectividad las reglas y los requerimientos de negocios, revisar los prototipos de diseño, y tomar decisiones aceptables. El papel de los gerentes durante una sesión de JRP es aprobar los objetivos del proyecto, establecer las prioridades del mismo, aprobar los calendarios y los costos, y aprobar las necesidades de entrenamiento ya identificadas y los planes de

implementación. En general, se confía en que tanto los usuarios como los gerentes van a asegurarse que sus factores críticos de éxito están siendo considerados.

- **Secretario(s):** Una sesión de JRP también incluye uno o más *secretarios*, quienes son responsables de llevar el registro relativo a todo lo que se discuta en la reunión. Estos registros se publican y se distribuyen a los asistentes inmediatamente después de la reunión con objeto de conservar el impulso que ha sido establecido por la sesión de JRP y sus miembros. La necesidad de publicar rápidamente los registros se refleja por el hecho de que los secretarios están usando cada vez más las herramientas de CASE para capturar muchos hechos (datos documentados para el usuario y modelos de proceso) que se comunican durante una sesión de JRP. Entonces, es conveniente que los secretarios posean un profundo conocimiento del análisis y diseño de sistemas y que sean hábiles en el uso de las herramientas de CASE. Los analistas de sistemas frecuentemente juegan este papel.
- **Equipo de TI:** Una sesión de JRP también puede incluir varias personas de TI, quienes principalmente escuchan y toman notas en relación con los aspectos y los requerimientos voceados por los usuarios y los gerentes. Normalmente, el personal de TI no hace comentarios a menos que se le invite. En vez de ello, las preguntas o inquietudes que tengan en general son canalizadas al facilitador de JRP inmediatamente antes o después de la sesión de JRP. Es el facilitador de JRP quien inicia y facilita la discusión de los aspectos por los usuarios y los gerentes.

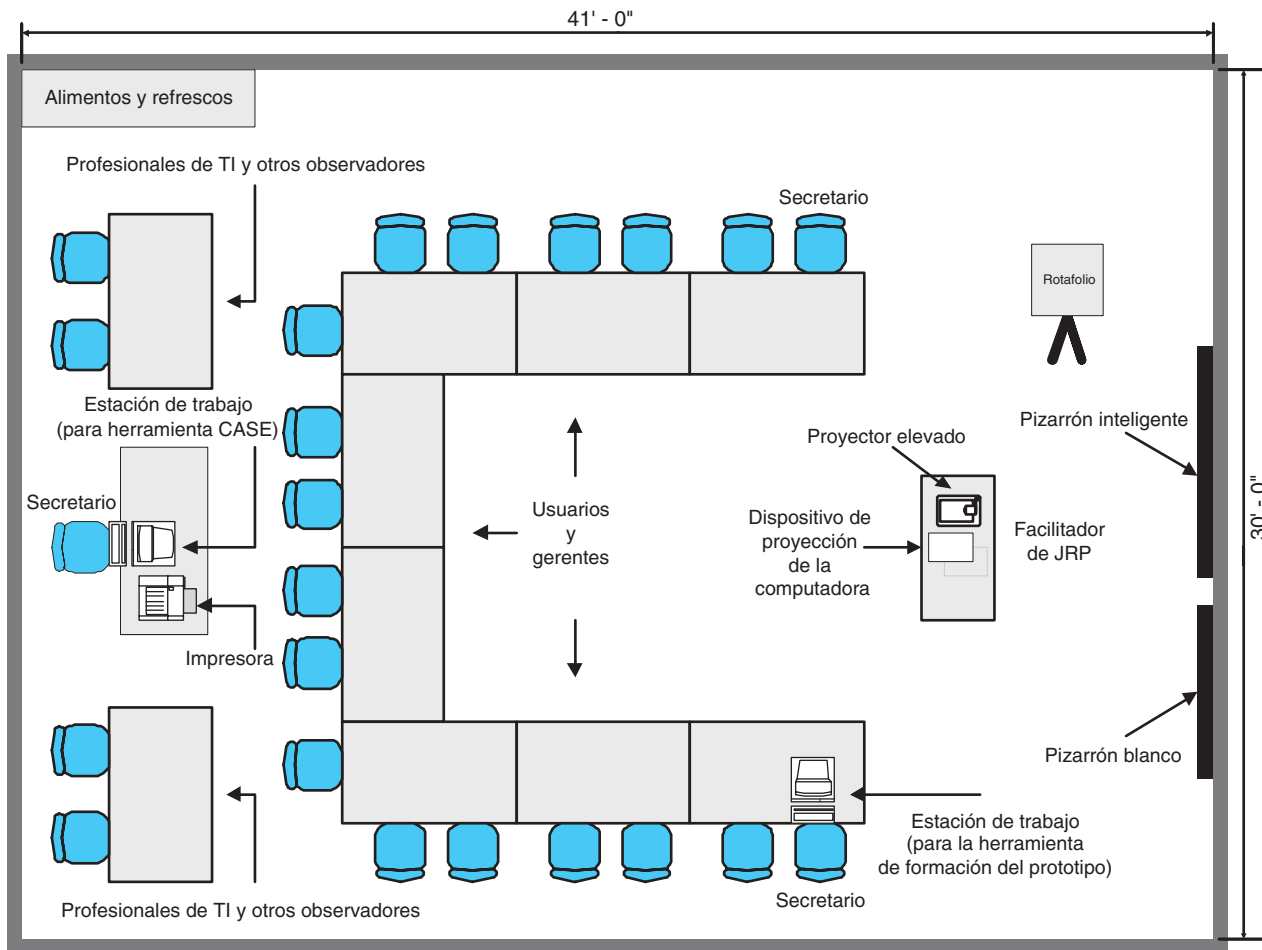
El equipo de TI en la sesión de JRP generalmente incluye miembros del equipo de proyecto, quienes pueden trabajar personalmente con el secretario para desarrollar modelos y otra documentación relacionada con los hechos comunicados durante la reunión. También puede llamarse a especialistas para que adquieran información con respecto a los aspectos técnicos e inquietudes especiales que puedan surgir. Si la situación lo justifica, el facilitador de JRP puede urgir a un profesional de TI para que encare el aspecto técnico.

**Cómo planear las sesiones de JRP** La mayoría de las sesiones de JRP abarcan de tres a cinco días y ocasionalmente duran hasta dos semanas. El éxito de cualquier sesión de JRP depende de la planeación apropiada y del desempeño efectivo del plan. Es necesaria alguna preparación mucho antes de realizar la sesión de JRP. Antes de planear una sesión de JRP, el analista debe trabajar íntimamente con el patrocinador ejecutivo para determinar el alcance del proyecto que se va a abordar a través de las sesiones de JRP. También es importante que se determinen los requerimientos de alto nivel y las expectativas de cada sesión de JRP. En general esto implica entrevistar a personas seleccionadas que son responsables de los departamentos o de las funciones que deben ser abordadas por el proyecto de sistemas. Finalmente, antes de planear la sesión de JRP, el analista debe asegurarse de que el patrocinador ejecutivo tenga deseos de dedicar gente, tiempo, y otros recursos a la sesión.

La planeación de una sesión de JRP implica tres pasos: selección de una ubicación para la sesión de JRP, selección de los participantes en JRP, y la preparación de una agenda que deberá seguirse durante la sesión de JRP. Examinemos cada uno de estos pasos de planeación en detalle:

1. **Selección de una ubicación para las sesiones de JRP:** Siempre que sea posible, las sesiones de JRP deberán conducirse lejos del lugar de trabajo de la compañía. La mayoría de los hoteles o las universidades locales tienen instalaciones diseñadas para atender las reuniones de grupo. Al celebrarse la sesión de JRP en una ubicación extra muros, los asistentes pueden concentrarse en los aspectos y actividades relacionados con la sesión de JRP y evitar interrupciones y distracciones que ocurrirían en su lugar de trabajo habitual. Independientemente de la ubicación de la sesión de JRP, deberá requerirse que todos los participantes asistan y deberá prohibírseles que regresen a su lugar de trabajo habitual.

Es común que una sesión de JRP requiera de varias salas. Se necesita una sala de conferencias en la cual pueda reunirse el grupo completo para abordar los aspectos de JRP. También, si la sesión de JRP incluye mucha gente, pueden ser necesarias varias salas más pequeñas para que se reúnan grupos separados de personas y que se adentren en la discusión de aspectos específicos.



**FIGURA 5.4** Disposición típica de una sala para sesiones de JRP

La sala de conferencias o sala principal de reuniones deberá alojar cómodamente a todos los asistentes. Deberá estar totalmente equipada con mesas, sillas y otros elementos que satisfagan las necesidades de todos los asistentes. La figura 5.4 ilustra una disposición típica de una sala para una sesión de JRP. Las ayudas visuales típicas para una sala de JRP deberán incluir un pizarrón blanco, un pizarrón inteligente o un pizarrón negro; uno o más rotafolios; y uno o más proyectores.

La sala también deberá incluir el equipo de cómputo necesario para los secretarios para registrar hechos y aspectos comunicados durante la sesión. La computadora deberá incluir paquetes de software para apoyar los diferentes tipos de registros o documentación que se vaya a capturar y publicar posteriormente por los secretarios. Este software puede incluir la herramienta CASE, un procesador de palabra, una hoja de cálculo, un paquete de presentación, un software para la elaboración de prototipos, una impresora, una copiadora (o un acceso rápido), y capacidad de proyección de cómputo. Como un lineamiento, el equipo de cómputo (excepto el que se use para la formación del prototipo) deberá ubicarse en la parte de atrás de la sala de modo que no interfiera o se convierta en una distracción para los participantes en el JRP. El foco de la sesión deberá ser la interacción personal de los participantes y no la tecnología.

Finalmente, la sala deberá estar equipada para teleconferencias de modo que puedan participar usuarios de localidades distantes; deberá incluir cuadernos de notas y lápices para los usuarios, los gerentes y otros asistentes. A estos últimos también deberá proporcionárseles etiquetas, tarjetas de ubicación, bocadillos, y bebidas para

que estén tan cómodos como sea posible. Las comodidades para los participantes son muy importantes ya que las sesiones de JRP son muy intensas y frecuentemente duran todo el día.

2. *Selección de los participantes de JRP:* Como se mencionó anteriormente, los participantes seleccionados incluyen al facilitador de JRP, el (los) secretario(s), y los representantes de la comunidad de usuarios. Los usuarios deberán ser personas clave que dominen el área de negocios. Desafortunadamente, con frecuencia los gerentes son muy dependientes de estas personas para llevar las áreas de negocios y se resisten a liberarlos de sus tareas. Así, el analista debe asegurarse de que la gerencia está comprometida con el proyecto de JRP y que quieren no solamente permitir sino también requieren que estas personas clave participen.

También pueden seleccionarse diferentes profesionales de TI para que participen en la sesión de JRP. Generalmente todas las personas de TI asignadas al equipo de proyecto participan en la sesión de JRP. También pueden seleccionarse otros especialistas de TI para abordar aspectos técnicos específicos alusivos al proyecto.

3. *Cómo preparar la agenda para una sesión de JRP:* La preparación es la clave para una sesión exitosa de JRP. El facilitador de JRP debe preparar documentación para informar brevemente a los participantes acerca del alcance y los objetivos de la sesión. Además, deberá prepararse una agenda para cada sesión de JRP y distribuirse antes de cada sesión. La agenda determina los aspectos que se van a discutir durante la sesión y el tiempo asignado a cada tema.

La agenda deberá contener tres partes: la introducción, el cuerpo y la conclusión. La introducción pretende comunicar las expectativas de la sesión, las reglas de campo, e influir o motivar a los asistentes para participar. El cuerpo pretende detallar los temas o aspectos que van a abordarse en la sesión de JRP. Finalmente, la conclusión representa el tiempo apartado para resumir la sesión del día y para recordar a los asistentes de los aspectos no solucionados (para desarrollarse posteriormente).

**Cómo conducir una sesión de JRP** La sesión de JRP comienza con comentarios iniciales, presentaciones, y un breve panorama de la agenda y de los objetivos de la sesión. El facilitador de JRP dirigirá la sesión siguiendo el guión preparado. Para conducir la sesión con éxito, como facilitador deberá seguir estos lineamientos:

- No se desvíe de la agenda en forma irrazonable.
- Manténgase dentro del programa (a los temas de la agenda se les asignan tiempos específicos).
- Asegúrese de que el secretario puede tomar notas (esto puede implicar que los usuarios y los gerentes vuelvan a enunciar sus comentarios más lentamente o con más claridad).
- Evite el uso de jerga técnica.
- Aplique habilidades para resolver conflictos.
- Permita recesos largos.
- Aliente el consenso de grupo.
- Aliente la participación del usuario y de la gerencia sin permitir que las personas dominen la sesión.
- Asegúrese de que los asistentes se adhieran a las reglas de campo establecidas para la sesión.

Uno de los objetivos de una sesión de JRP es generar posibles ideas para resolver un problema. Un enfoque es la lluvia de ideas. La **lluvia de ideas** consiste en alentar a los participantes para generar tantas ideas como sea posible, sin analizar la validez de las mismas.

La lluvia de ideas es una técnica formal que requiere disciplina. Deberán usarse los siguientes lineamientos para asegurar una lluvia de ideas efectiva:

1. Aísle a las personas apropiadas en un lugar que esté libre de distracciones e interrupciones.
2. Asegúrese de que todos entienden el propósito de la junta (generar ideas para resolver el problema) y céntrese en el (los) problema(s).

---

**Lluvia de ideas** Técnica para generar ideas al alentar a los participantes para que ofrezcan tantas ideas como sea posible en un corto tiempo sin ningún análisis hasta que se hayan agotado todas las ideas.

3. Asigne a una persona para que registre las ideas. Esta persona deberá usar un rotafolios, un pizarrón o un proyector elevado que pueda ver todo el grupo.
4. Recuérdeles a todos las reglas de la lluvia de ideas:
  - a) Ser espontáneo. Enunciar las ideas tan pronto como ocurran.
  - b) No se permite ninguna crítica, análisis, o evaluación de ninguna clase mientras que las ideas se están generando. Cualquier idea puede ser útil, aunque solamente sea para originar otra.
  - c) La meta es la cantidad de ideas, no necesariamente la calidad.
5. Dentro de un tiempo específico, los miembros del equipo expresan sus ideas tan rápido como puedan pensarlas.
6. Después que el grupo haya agotado sus ideas y todas hayan sido registradas, entonces, y solamente entonces, es que éstas pueden ser analizadas y evaluadas.
7. Refine, combine y amplifique las ideas que se generaron anteriormente.

Con un poco de práctica y atención a estas reglas, la lluvia de ideas puede ser una técnica muy efectiva para generar ideas para resolver problemas.

Como se mencionó anteriormente, el éxito de una sesión de JRP es muy dependiente de la planeación y de las habilidades del facilitador de JRP y de los secretarios. Estas habilidades mejoran a través del entrenamiento apropiado y la experiencia. Por lo tanto, las sesiones de JRP en general concluyen con un cuestionario de evaluación para que lo llenen los participantes. Las respuestas ayudarán a asegurar la posibilidad de los futuros éxitos de JRP.

El producto final de una sesión de JRP es comúnmente un documento formal escrito. En general este documento es creado por el facilitador de JRP y por los secretarios. Es esencial para confirmar las especificaciones acordadas durante la sesión por todos los participantes. Es obvio que el contenido y la organización de las especificaciones dependen de los objetivos de la sesión de JRP. El analista puede proveer un conjunto distinto de especificaciones a los diferentes participantes basándose en su papel: por ejemplo, un gerente puede recibir más que una versión resumida del documento suministrado a los usuarios participantes (especialmente en los casos en que los propietarios del sistema tengan una participación mínima real en la sesión de JRP).

**Los beneficios de JRP** La planeación conjunta de requerimientos ofrece muchos beneficios como un enfoque alternativo de exploración y desarrollo. Más y más compañías están comenzando a vislumbrar sus ventajas e incorporando el JRP en sus metodologías existentes. Una sesión de JRP conducida con efectividad ofrece los siguientes beneficios:

- JRP involucra activamente a los usuarios y a la gerencia en el proyecto de desarrollo (alentándolos a “apropiarse” del proyecto).
- JRP reduce el tiempo requerido para desarrollar sistemas. Esto se logra reemplazando las entrevistas tradicionales de uno a uno que consumen mucho tiempo de cada usuario y cada gerente con las reuniones de grupo. Estas últimas permiten obtener más fácilmente un consenso entre los usuarios y los gerentes, así como resolver la información y los requerimientos conflictivos.
- Cuando el JRP incorpora la elaboración de prototipos como un medio para confirmar los requerimientos y obtener las aprobaciones de diseño, se vislumbran los beneficios de la elaboración de prototipos.

El lograr una sesión exitosa de JRP depende del facilitador de JRP y de su habilidad para planear y facilitar la sesión de JRP.

## Una estrategia de exploración

Un analista necesita un método organizado para recabar hechos. Los analistas sin experiencia frecuentemente inician de inmediato las entrevistas. Ellos piensan, “Ve a la gente. ¡Ahí es donde están los hechos reales!” Este enfoque no reconoce un hecho importante de la vida: Las personas deben terminar su trabajo cotidiano. Usted podría pensar, “Pero yo pensaba que usted ha estado diciendo que el sistema es para personas y que la partici-



pación directa del usuario final en el desarrollo de sistemas es esencial. ¿No están ustedes contradiciéndose?

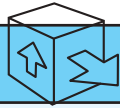
No del todo. El tiempo es oro. Desperdiciar el tiempo de los usuarios finales es desaprovechar el dinero de su compañía. Para aprovechar bien el tiempo transcurrido con usuarios finales, los analistas no deberían entrar directamente en las entrevistas. Los analistas primero deberían recolectar todos los hechos que puedan usando otros métodos. Considere la siguiente estrategia paso por paso:

1. Aprenda de documentos existentes, formas, informes, y archivos. Los analistas pueden aprender bastante sin tener contacto con las personas.
2. Si es apropiado, observe el sistema en acción.
3. Dados todos los hechos ya recabados, diseñe y distribuya cuestionarios para aclarar las cosas que no se entienden completamente.
4. Conduzca entrevistas (o las sesiones de trabajo en grupo). Como la mayor parte de los hechos pertinentes ya ha sido recolectada por los métodos de contacto de usuario bajo, las entrevistas pueden usarse para verificar y aclarar los asuntos y los problemas más difíciles. (Alternativamente, considere usar las técnicas JRP para reemplazar o complementar entrevistas.)
5. (Optativo.) Construya prototipos de identificación para cualquier requerimiento funcional que no sean comprendidos o para los requerimientos que necesitan ser validados.
6. Haga un seguimiento. Use técnicas de exploración apropiadas para verificar hechos (usualmente las entrevistas o la observación).

La estrategia no es sagrada. Aunque una estrategia de exploración debería ser desarrollada para cada fase pertinente de desarrollo de sistemas, cada proyecto es único. Algunas veces la observación y los cuestionarios pueden ser inadecuados. Pero la idea siempre debería ser coleccionar tantos hechos como sea posible antes de usar las entrevistas.

Este capítulo le inició en una gran variedad de técnicas para identificar los requerimientos de un sistema de información. La mayoría de metodologías de desarrollo de sistemas requieren algún nivel de documentación y el análisis de los requerimientos del sistema. A continuación, los capítulos restantes de esta parte presentan varias herramientas y técnicas de la documentación de sistemas que pueden usarse durante la fase de análisis del desarrollo de sistemas. La mayoría de ustedes proseguirán directamente al capítulo 6, “Modelado de requerimientos del sistema con los casos de uso”. Los modelos de casos de uso sirven como fundamento para el desarrollo de modelos subsiguientes para modelar los requerimientos adicionales de sistemas y se presentan en los capítulos 7 a 9.

## Resumen

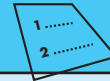


1. El proceso y las técnicas que un analista de sistemas usa para identificar, analizar, y entender los requerimientos de sistema son llamados identificación de requerimientos.
2. Los requerimientos de sistema especifican lo que el sistema de información debe hacer, o qué propiedad o cualidad debe tener el sistema.
3. El proceso de identificación de requerimientos consta de las siguientes actividades:
  - a) El análisis e identificación del problema.
  - b) La identificación de requerimientos.
  - c) La documentación y el análisis de los requerimientos.
  - d) La administración de los requerimientos.
4. La exploración de los hechos es una técnica que se usó a través del ciclo de desarrollo completo, pero es sumamente crítica en la fase de análisis de requerimientos.
5. Una herramienta popular usada por los equipos de desarrollo para identificar, analizar, y solucionar problemas es el diagrama Ishikawa.
6. La conducción de negocios de una manera ética es una práctica requerida, y los analistas necesitan ser más conscientes de las implicaciones de no ser éticos.
7. Hay siete técnicas de exploración comunes:
  - a) El muestreo de documentos y archivos existentes puede proveer muchos hechos y detalles sin que sea necesaria poca o ninguna comunicación personal directa. El analista deberá recolectar documentos históricos, manuales y formas de operaciones en los negocios, y documentos de sistemas de información.
  - b) La investigación es una técnica a menudo pasada por alto basada en el estudio de otras aplicaciones similares. Ahora se ha hecho más conveniente con la Internet y la Web (WWW). Las visitas del sitio Web son una forma especial de investigación.
  - c) La observación es una técnica de exploración en la cual el analista estudia a la gente haciendo su trabajo.
  - d) Se usan cuestionarios para recolectar hechos similares de un gran número de individuos.
  - e) Las entrevistas son lo más popular pero son la técnica de exploración de hechos que consume más tiempo. Al entrevistar, el analista se encuentra individualmente con las personas para recabar información.
    - i) Cuando la mayoría de la gente habla de habilidades de comunicación, piensan acerca de hablar y escribir. La de escuchar apenas se menciona, pero puede ser lo más importante, especialmente durante el proceso de entrevista.
    - ii) Los estudios de investigación han determinado un hecho sorprendente: Del total de los sentimientos de una persona, sólo siete por ciento se comunica oralmente (con palabras), mientras que 38 por ciento se comunica por el tono de voz usado y 55 por ciento por las expresiones facial y corporal. Si usted sólo escucha las palabras de alguien, pierde la mayor parte de lo que la persona tiene que decir. Los analistas de sistemas experimentados ponen mucha atención al lenguaje corporal y a la proxemia.
  - f) La elaboración de prototipos de identificación se aplica frecuentemente a los proyectos de desarrollo de sistemas, en especial en casos donde el equipo de desarrollo tiene problemas para definir los requerimientos del sistema. La filosofía es que los usuarios reconocerán sus requerimientos cuando los vean. Es importante que el prototipo sea desarrollado con rapidez a fin de que pueda ser usado durante el proceso de desarrollo.
  - g) Muchos analistas encuentran fallas en las entrevistas: las entrevistas por separado frecuentemente conducen a hechos, opiniones, y prioridades conflictivos. El resultado final es un gran número de entrevistas de seguimiento, reuniones grupales o ambas. Por esta razón, muchas organizaciones están usando una sesión de trabajo grupal conocida como sesión de planeación conjunta de requerimientos como un sustituto de la entrevista.
    - i) Las sesiones de planeación conjunta de requerimientos incluyen una gran variedad de participantes y los papeles. Se espera que cada participante asista y participe activamente durante toda la sesión JRP.
    - ii) Una sesión efectiva JRP incluye una planeación extensa. Planificar para una sesión JRP involucra tres pasos: la selección de un lugar para la sesión JRP, la selección de participantes en la JRP, y la preparación de un orden del día para ser seguido durante la sesión JRP.
8. Para ayudar a aliviar los muchos problemas asociados con el cambio de requerimientos, hay que realizar la administración de requerimientos. Ésta abarca las políticas, los procedimientos, y los procesos que gobiernan cómo se maneja un cambio de un requerimiento.
9. Porque “el tiempo es oro”, es sabio y práctico para el analista de sistemas usar una estrategia de exploración para maximizar el valor del tiempo ocupado con los usuarios finales.
  - a) Aprenda de documentos, formas, informes, y archivos existentes. Los analistas pueden aprender bastante sin tener contacto con las personas.

- b) Si es apropiado, observe el sistema en actividad.
- c) Dados todos los hechos ya recabados, diseñe y distribuya cuestionarios para aclarar las cosas que no son completamente comprendidas.
- d) Conduzca entrevistas (o las sesiones de trabajo grupal). Como la mayor parte de los hechos pertinentes ya ha sido recolectada por los métodos de contacto de usuario bajo, las entrevistas pueden usarse para verificar y aclarar los asuntos y los problemas más difíciles. (Alternativamente,

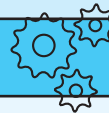
considere usar las técnicas JRP para reemplazar o complementar entrevistas.)

- e) (Optativo.) Construya prototipos de identificación para los requerimientos funcionales que no se comprendan o para los requerimientos que necesitan ser validados.
- f) Haga un seguimiento. Use técnicas de exploración apropiadas para verificar hechos (usualmente las entrevistas o la observación.)



## Preguntas de repaso

1. ¿Cuál es la importancia de conducir el proceso de identificación de requerimientos?
2. ¿Cuáles son las consecuencias posibles si usted fracasa al identificar correcta y completamente los requerimientos de sistema?
3. ¿Cuáles son algunos de los criterios considerados como críticos en la definición de los requerimientos de sistema?
4. ¿De qué actividades consta el proceso de identificación de requerimientos?
5. Describa brevemente el propósito y los componentes de un diagrama Ishikawa.
6. ¿Qué técnica se usa comúnmente en la fase de identificación de requerimientos? ¿Por qué es importante eso?
7. ¿Por qué es esencial el análisis de requerimientos?
8. Al recolectar los hechos de la documentación existente, ¿qué clase de documentos debería revisar el analista de sistema?
9. ¿Cuáles son algunas de las desventajas de recolectar hechos observando a los empleados en su ambiente de trabajo? ¿Cómo pueden manejar estas desventajas los analistas de sistemas?
10. ¿Cuáles son los tipos de cuestionarios para encuesta que los analistas de sistemas pueden utilizar para recopilar la información y las opiniones?
11. ¿Cuáles son algunas de las formas con las que usted puede ayudar a abrir las líneas de comunicación en una entrevista?
12. ¿Qué es la planeación conjunta de requerimientos (JRP)?
13. ¿Por qué la JRP se ha hecho popular?
14. ¿Por qué es tan importante el facilitador de JRP?
15. ¿Cuál es la preocupación principal al seleccionar un sitio para sesiones de JRP?



## Problemas y ejercicios

1. Usted maneja un proyecto que se pospuso dos veces porque su financiamiento fue desviado a proyectos de prioridad más alta. Los dueños del sistema no quieren que eso ocurra otra vez, así es que están llenos de ansiedad por iniciar el nuevo sistema y hacer que se construya tan rápido como sea posible. A usted lo presionan mucho para que no ocupe más de un par de días en la identificación de los requerimientos. Si hace falta cualquier cosa, le dicen, puede componerse más tarde. Usted realmente quiere complacerlos, pero se hace presente una pequeña voz de cautela. ¿Cuáles son las consecuencias y los costos potenciales de recorrer a la carrera el proceso de identificación de los requerimientos?
2. Usted ha aprendido la importancia de asegurarse de que los requerimientos están correctamente identificados. Si no, ¿cómo sabe usted cuándo tiene un requerimiento correcto; es decir, qué criterios debe cumplir cada requerimiento para ser considerado correcto?
3. ¿Qué error común comete a menudo un analista de sistemas nuevo al analizar un problema? ¿Cuáles son las consecuencias potenciales de este error? ¿Qué herramienta puede usarse para ayudar a evitar este problema?
4. Los desarrolladores de sistemas usan técnicas de exploración en cada fase de proyecto. ¿Es la exploración más importante durante la fase de análisis de requerimientos que para otras fases? ¿Por qué sí o por qué no?
5. ¿Qué asuntos éticos podrían surgir durante el proceso de exploración, y cómo deberían manejarse?
6. ¿Cuáles son algunas de las técnicas y herramientas comunes que un analista de sistemas puede usar para documentar los hallazgos iniciales? ¿Debería esperar el analista de sistemas que los requerimientos sean cabales y correctos en este punto? En caso

- de que no, ¿cuáles son los problemas comunes?  
¿Cuál debe ser el enfoque del equipo de proyecto en este punto?
7. ¿Cuál es el producto que se crea una vez que el análisis de requerimientos se termina? ¿Por qué se necesita este producto y qué es lo que incluye? ¿Quiénes son la audiencia, los usuarios (o ambos) de este producto, y por qué razones?
  8. Usted es un analista de sistemas en una compañía de desarrollo del software que ha sido contratada para hacer la fase de análisis de requerimientos para una organización grande. ¿Cuáles son las tres categorías de documentación existente que usted debería recolectar durante la identificación de los requerimientos? ¿Cuáles son algunos ejemplos de cada uno de estos tres tipos de documentación? ¿Con qué debería tener cuidado el analista de sistemas en la recolección de la documentación?
  9. Suponga que usted es un analista de sistemas en un proyecto que involucra modificar el proceso de orden de ventas. Como su compañía recibe aproximadamente 2 500 órdenes de ventas al día, ¿cuántas necesita usted muestrear si quiere la certeza de 95 por ciento de que tiene cubiertas todas las variaciones? ¿Qué ocurre si el número de órdenes de ventas por día fuera de 25 000?
  10. Las encuestas y los cuestionarios se usan frecuentemente para recolectar hechos. ¿Cuáles son las ventajas y las desventajas de los cuestionarios? ¿Cuándo escogería los cuestionarios de formato libre en vez de los cuestionarios del formato fijo? ¿Cuál es un método para determinar la efectividad de un cuestionario?
  11. ¿Cuáles son algunas de las razones para usar la planeación conjunta de requerimientos (JRP) como una técnica de exploración? ¿Cuál debería ser la base para seleccionar cuáles usuarios y gerentes participarán en la sesión JRP, y generalmente quién los selecciona? ¿Qué habilidades deberían poseer el facilitador y el secretario? ¿Cuál es el papel del equipo de TI (tecnología de la información) durante las sesiones de JRP? ¿Cuál es la duración típica de las sesiones de JRP?
  12. Provea al menos cinco de los factores críticos de éxito para las sesiones JRP.
  13. ¿Qué es lo que *no* debería hacer un analista al empezar la etapa de exploración de la identificación de requerimientos, sin importar cuán tentador sea?

## Proyectos e investigación



1. Los analistas de sistemas deben tener pericia en el análisis de problemas. Cuando los analistas de sistemas comienzan, a menudo encuentran difícil diferenciar los síntomas de los problemas, e identificar las causas reales del problema. Una herramienta que puede ayudar a los analistas a aprender a hacer esto es el diagrama Ishikawa, o de espina de pescado.
  - a) Encuentre y seleccione un problema que su empresa, escuela, u otra organización actualmente trata de resolver. Describa este problema.
  - b) Siga el proceso descrito en este capítulo y cree un diagrama Ishikawa.
  - c) ¿Con cuáles categorías comenzó usted en el diagrama, y que cuáles añadió durante el proceso?
  - d) ¿Coadyuvó este diagrama en encontrar la(s) causa(s) real(es) del problema? ¿La(s) causa(s) que usted originalmente pensó resultó(aron) ser verdadera(s), o algo diferente?
2. Observar el ambiente de trabajo es una técnica que antecede a la era de la informática, pero que todavía puede ser altamente efectivo. ¡Aunque no es aplicable para cada situación, el observar qué hacen las personas realmente y cómo lo hacen puede ser en algunos casos mucho más exacto que preguntarles! Seleccione un sistema (ya sea hipotético o verdadero) y haga lo siguiente:
  - a) Provea una visión general del sistema y qué trata usted de aprender del sistema para un proyecto.
    - b) Desarrolle un plan de observación de trabajo usando los lineamientos de este capítulo. El formato depende de usted, pero generalmente no debería necesitar más que 1 a 2 páginas.
    - c) Desarrolle un plan de muestreo del trabajo, y describa los procedimientos de muestreo que usted usará.
    - d) ¿Qué piensa acerca de este método comparado con otros métodos de exploración?
3. Usted es un analista de sistemas que trabaja en un proyecto para desarrollar una intranet para una organización grande con muchos miles de empleados trabajando en oficinas a todo lo largo de los Estados Unidos. Ésta será la primera intranet de la organización, y la gerencia ejecutiva la quiere para ayudar a aumentar eficiencia de los empleados y el compromiso con la empresa. Como parte de la exploración, es necesario recopilar información de los empleados con respecto al contenido de la intranet y la funcionalidad. Debido al tamaño y la distribución geográfica de la organización, así como las restricciones de tiempo del proyecto, el tiempo y los recursos son insuficientes para las entrevistas personales, así es que usted ha decidido que es necesario un cuestionario.
  - a) ¿Cuáles son los hechos y las opiniones que usted necesita recolectar?
  - b) ¿Deberá encuestarse a todos los empleados de la organización? ¿Por qué sí o por qué no? En

- caso de que no todos los empleados deban ser encuestados, ¿cómo seleccionaría usted a los que deban ser encuestados?
- c) ¿Qué formato piensa usted que funcionaría mejor para este cuestionario de encuesta? ¿Si fuera el formato fijo, qué tipo(s) de preguntas del formato fijo debería usar?
  - d) ¿Qué longitud debería tener el cuestionario de encuesta para obtener la información necesaria sin desanimar a los empleados para que lo llenen?
  - e) Cree el cuestionario de encuesta, usando los lineamientos para escribir preguntas dados en este capítulo.
4. Basado en las respuestas para su encuesta de la intranet, usted considera que sería de ayuda entrevistar alguien en otra organización que haya tenido experiencia en el desarrollo, mantenimiento o ambos de una intranet corporativa.
- a) ¿Qué tipo de entrevista piensa usted que sería más apropiado en esta situación: no estructurada o estructurada? ¿Por qué?
  - b) Haga una cita con el administrador de la intranet en su empresa u otra organización o escuela para discutir sus experiencias en el desarrollo, mantenimiento o ambos de una intranet. Describa la organización y su intranet.
  - c) Prepare un guión de entrevista usando el formato de la figura 5-3 por poner un ejemplo, asegurándose que las preguntas no tienen los problemas discutidos en este capítulo.
  - d) Conduzca la entrevista, y registre las respuestas.
  - e) ¿Qué es lo que usted piensa que funcionó bien en la entrevista, y qué haría usted en forma diferente la próxima vez?
5. El lenguaje corporal es una parte sumamente importante de la comunicación, como se describe en este libro. Los analistas necesitan darse cuenta no sólo de lo que está siendo comunicado a través del lenguaje corporal del entrevistado, sino también del impacto que su lenguaje corporal puede tener sobre el proceso de la entrevista. Haga una cita con varios compañeros de trabajo o condiscípulos para hacer una entrevista en relación con las características que les gustaría ver en una intranet; si es posible, seleccione a los entrevistados que usted conoce bien y a esos que usted no conoce. Prepárese para las entrevistas siguiendo los mismos pasos de la pregunta anterior.
- a) Describa a los entrevistados que usted seleccionó y las preguntas que usted formuló.
  - b) Durante cada entrevista, observe las expresiones faciales del entrevistado. ¿Qué observó usted? ¿Eran consistentes siempre las expresiones faciales con las respuestas?
  - c) Durante cada entrevista, observe el contacto visual. ¿Cuánto tiempo duró? Observe y describa lo que sucedió cuando usted hizo contacto visual durante más de tres a cinco segundos con el entrevistado.
  - d) Pruebe cambiar su zona espacial durante la entrevista. ¿Mostró el entrevistado algún síntoma de sentirse incómodo? ¿En qué momento ocurrió esto?
  - e) ¿Notó usted alguna diferencia en el lenguaje corporal entre quienes usted conoce bien y quienes no conoce?
  - f) ¿Qué fue lo más exitoso y lo menos exitoso que hizo al suscitar información?
6. Comúnmente los analistas tienen acceso a datos confidenciales o sensibles durante la fase de identificación de requerimientos de un proyecto, en particular durante la exploración. Los analistas necesitan darse cuenta de situaciones donde pueden infringir la ética profesional, ya sea por actos de comisión u omisión, y las posibles consecuencias. Busque en la Web y/o en periódicos de negocios comerciales en la biblioteca de su escuela artículos sobre incidentes que involucren brechas de ética profesional.
- a) ¿Qué artículos encontró usted?
  - b) ¿Cuál fue la naturaleza de cada uno de estos incidentes?
  - c) ¿Cuáles fueron las consecuencias?
  - d) ¿Cuál fue la responsabilidad personal del analista en cada incidente?
  - e) ¿Qué pudo haberse hecho al nivel de la organización, individual o ambos para evitar el incidente o reducir su gravedad?

## Casos breves



- En el capítulo 4, usted desarrolló estudios de factibilidad para un proyecto. Las evaluaciones económicas de factibilidad son afectadas significativamente por intangibles, cuyo valor se obtiene en parte por entrevistas y cuestionarios. Desarrolle preguntas de entrevista para determinar el valor para los empleados del trabajo a distancia.
  - Comience con preguntas no estructuradas planteadas a un grupo de empleados para determinar qué es lo que interesa a los empleados y cómo consideran el trabajo a distancia.
  - Una vez que usted sabe qué aspectos rodean a la percepción del empleado acerca del trabajo a distancia y por qué les podría agradar/desagradar, cree preguntas abiertas, pero que sean estructuradas, sobre esos asuntos, y entreviste a un segundo conjunto de trabajadores. ¿Por qué usamos dos grupos diferentes de empleados para este proceso?
- Desarrolle un cuestionario para distribuirse masivamente entre los empleados, basado en sus conclusiones de las entrevistas previas. ¿Por qué estamos nosotros completando el análisis con una encuesta anónima?
- Usted se encarga de desarrollar un sistema de inscripción de clase en línea nuevo para su escuela. Desarrolle un conjunto de preguntas de entrevista para determinar los asuntos y las necesidades de los estudiantes, el equipo de inscripción, y el profesorado para un sistema de inscripción en línea.
- Discuta el impacto que las preguntas sesgadas o principales pueden tener sobre un análisis. Cree una pregunta de entrevista sin sesgo y una pregunta capciosa o sesgada. Plantee cada una de esas preguntas a cinco personas. ¿Qué clase de respuestas obtuvo? ¿Eran las que usted esperaba?

## Ejercicios de equipo e individuales



- Cree un conjunto de preguntas de entrevistas con sesgo, capciosas o cargadas. Plántelas a otro estudiante en la clase. El otro estudiante, en lugar de contestar las preguntas, le deberá decir cómo está sesgado usted y qué respuesta anda usted buscando.
- Ejercicio de clase: Cree un conjunto de preguntas de entrevista que tengan el menor sesgo posible sobre un tema particular. Plantee las preguntas al grupo. Sin embargo, traiga puesta una playera, alfileres, etcétera, para conducir al grupo a que responda de una manera específica. Diviértase, y experimente con medios visuales, apoyos y cosas por el estilo.
- Se ha encontrado en estudios de investigación pasados que los empleados a quienes se les permite el trabajo a distancia, en realidad trabajan aproximadamente tres horas extras sin pago a la semana. Pero el trabajo a distancia es a menudo utilizado como una herramienta negociadora por un empleador: para que puedan optar por el trabajo a distancia, los empleados deben aceptar un sueldo inferior, típicamente 10 por ciento. ¿Qué piensa usted acerca de esto?

## Lecturas recomendadas



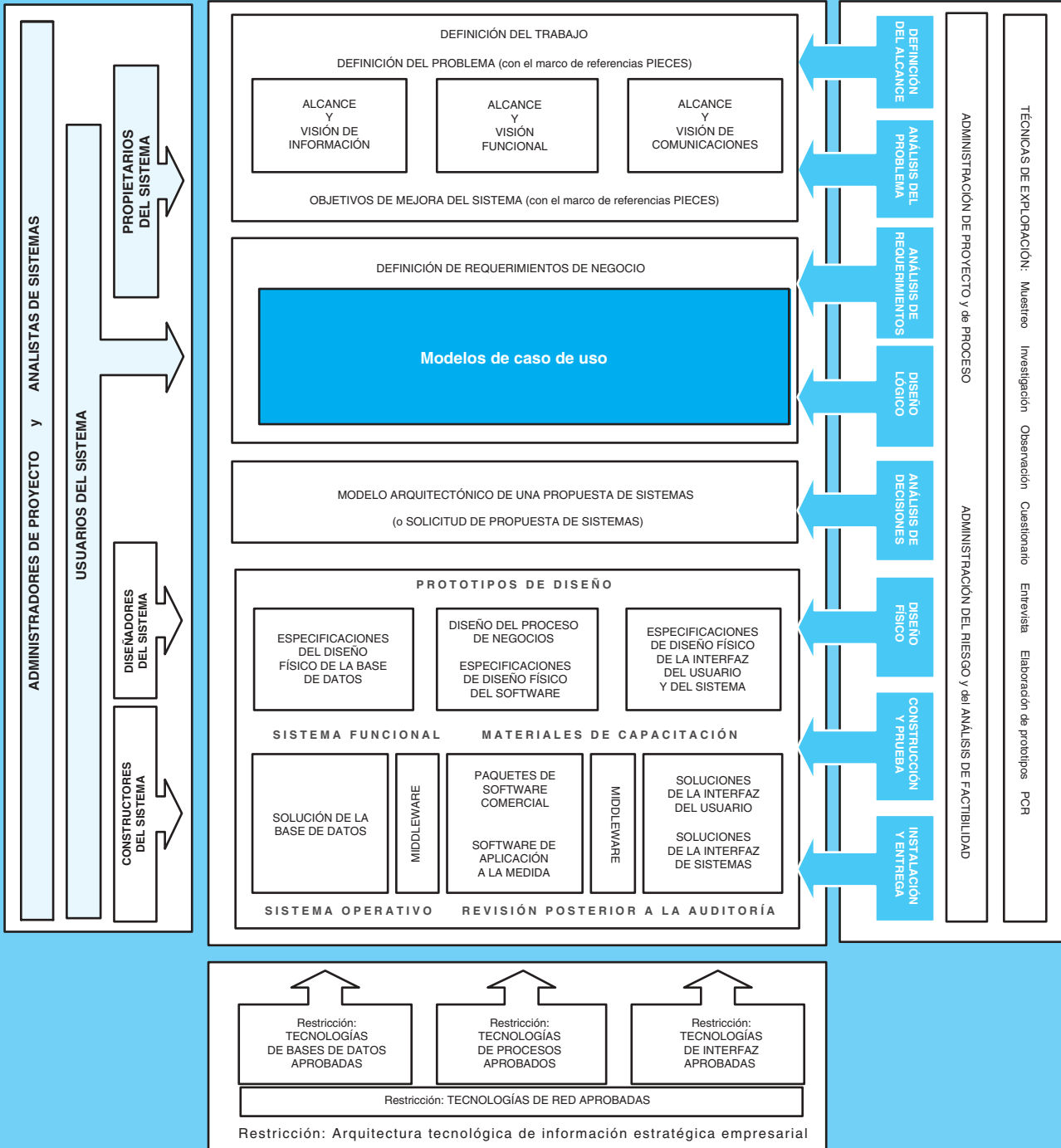
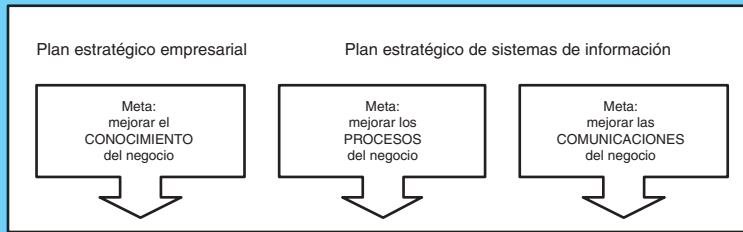
- Andrews, D. C. y N. S. Leventhal. *Fusion Integrating IE, CASE and JAD: A Handbook for Reengineering the Systems Organization*. Englewood Cliffs, NJ: Prentice Hall, 1993.
- Berdie, Douglas R. y John F. Anderson. *Questionnaires: Design and Use*. Metuchen, NJ: Scarecrow Press, 1974. Una guía práctica para la redacción de cuestionarios. Especialmente útil debido a su pequeño tamaño y sus ejemplos ilustrativos.
- Davis, William S. *Systems Analysis and Design*. Reading, MA: Addison-Wesley, 1983. Suministra sugerencias útiles para la preparación y conducción de entrevistas.
- Dejoie, Roy; George Fowler y David Paradice. *Ethical Issues in Information Systems*. Boston, MA: Boyd and Fraser, 1991. Se centra en el impacto de la tecnología de las computadoras sobre la

toma de decisiones ética en las organizaciones de negocios actuales.

- Fitzgerald, Jerry; Ardra F. Fitzgerald y Warren D. Stallings, Jr. *Fundamentals of Systems Analysis*, 2a. ed. New York: John Wiley & Sons, 1981. Un útil libro de estudio para el analista de sistemas. El capítulo 6, "Understanding the Existing System", hace una muy buena presentación de las técnicas de exploración en la fase de estudio.
- Gane, C. *Rapid Systems Development*. Nueva York: Rapid Systems Development, Inc., 1987. Este libro suministra una buena discusión sobre cómo conducir una reunión de grupo/entrevista.
- Gause, Donald C., y Gerald M. Weinberg. *Exploring Requirements: Quality before Design*. Nueva York: Dorset House Publishing,

1989. Un excelente libro que describe las técnicas de la exploración de requerimientos.
- Gildersleeve, Thomas R. *Successful Data Processing System Analysis*. Englewood Cliffs, NJ: Prentice Hall, 1978. El capítulo 4, "Interviewing in Systems Work", da una visión completa de la entrevista específicamente para el analista de sistemas. En este capítulo se presenta un guión de una entrevista de muestra y se le analiza.
- London, Keith R. *The People Side of Systems*. Nueva York: McGraw-Hill, 1976. El capítulo 5, "Investigation versus Inquisition", ofrece una visión muy buena orientada hacia la gente de la exploración, con un énfasis considerable en las entrevistas.
- Lord, Kenniston W., Jr. y James B. Steiner. *CDP Review Manual: A Data Processing Handbook*, 2a. ed. New York: Van Nostrand Reinhold, 1978. El capítulo 8, "Systems Analysis and Design", presenta una comparación comprensiva de los méritos y los deméritos de cada técnica de exploración. Este material está destinado a la preparación de los procesadores de datos para los exámenes del Certificado de Procesamiento de Datos (Certificate in Data Processing, CDP), uno de los cuales cubre el análisis y diseño de sistemas.
- Miller, Irwin y John F. Freund. *Probability and Statistics for Engineers*. Englewood Cliffs, NJ: Prentice Hall, 1965. Un libro introductorio a nivel universitario sobre probabilidad y estadística.
- Mitchell, Ian; Norman Parrington; Peter Dunne y John Moses. "Practical Prototyping, Part One", *Object Currents*, mayo de 1996. El primero de una serie de artículos en tres partes que explora la realización de prototipos y cómo obtener un beneficio de ello. La realización de prototipos es una parte integral de JRP.
- Robertson, Suzanne y James Robertson. *Mastering the Requirements Process*. Reading, MA: ACM Press/Addison-Wesley, 1999. Este libro contiene una cobertura a gran profundidad de los procedimientos paso a paso de la identificación de requerimientos.
- Salvendy, G., ed. *Handbook of Industrial Engineering*. Nueva York: John Wiley & Sons, 1974. Un manual comprensivo para ingenieros industriales; en cierto modo los analistas de sistemas son un tipo de ingeniero industrial. Una cobertura excelente del muestreo y de la medición del trabajo.
- Stewart, Charles J. y William B. Cash, Jr. *Interviewing: Principles and Practices*, 2a. ed. Dubuque, IA: Brown, 1978. Un popular libro a nivel universitario que provee una amplia exposición de las técnicas de la entrevista, muchas de las cuales son aplicables al análisis y diseño de sistemas.
- Walton, Donald. *Are You Communicating? You Can't Manage without It*. Nueva York: McGraw-Hill, 1989. Este libro es una guía fácil de usar sobre el proceso de las comunicaciones y es obligatorio para cualquiera que trabaje con las personas y las influya.
- Weinberg, Gerald M. *Rethinking Systems Analysis and Design*. Boston: Little, Brown and Company, 1982. Un libro creado para estimular una nueva manera de pensar.
- Wood, Jane y Denise Silver. *Joint Application Design*. Nueva York: John Wiley & Sons, 1989. Este libro provee una visión general completa de la técnica de diseño conjunto de aplicaciones de IBM.







# 6

## Modelado de requerimientos del sistema con los casos de uso

### Panorámica y objetivos del capítulo

En este capítulo usted aprenderá acerca de las herramientas y las técnicas necesarias para la elaboración de modelos de requerimientos de sistemas usando casos de uso. La captura y la documentación de los requerimientos del sistema han demostrado ser factores críticos para el resultado de un proyecto exitoso de desarrollo de requerimientos de sistemas. La documentación de los requerimientos desde la perspectiva de los usuarios de manera que puedan entender, promueve la participación del usuario, lo que aumenta en gran medida la probabilidad de éxito del proyecto. Sabrá que entiende la modelación de los casos de uso de los requerimientos cuando usted pueda:

- Describir los beneficios de la modelación de los casos de uso.
- Definir a los actores y a los casos de uso y ser capaz de identificarlos de los diagramas de contexto y de otras fuentes.
- Describir los cuatro tipos de actores.
- Describir las relaciones que pueden aparecer en un diagrama de caso de uso.
- Describir los pasos para preparar un modelo de caso de uso.
- Describir cómo construir un diagrama de modelo de caso de uso.
- Describir las diversas secciones de una narración de caso de uso y ser capaz de preparar una.
- Definir el propósito de la jerarquización de los casos de uso y de la matriz de prioridades así como el diagrama de dependencia de los casos de uso.

## Introducción

Después de la reunión de planeación conjunta de requerimientos (*joint requirements planning*, JRP), que se realizó como una tarea de la fase de análisis de requerimientos, el equipo del proyecto del sistema de Servicios para miembros SoundStage, ha construido una lista de casos de uso que especifica la totalidad de la funcionalidad requerida del sistema. Primero, cada caso de uso era simplemente una frase verbal (tal como “Colocar una nueva orden”) que describía algo que uno o más usuarios querían hacer con el sistema. Luego, cada caso de uso se documentaba con una narración que describía en detalle la interacción deseada entre el usuario y el sistema. Entonces Bob Martínez y otros analistas de sistemas sostuvieron una serie de entrevistas con los usuarios para verificar estas narraciones de casos de uso. Finalmente, ellos están analizando cuáles son los casos que tienen la más alta prioridad para el sistema. La jefa de Bob, Sandra, dice que ella va a identificar para ellos cuál funcionalidad tiene que incluirse en el primer ciclo de construcción del sistema. El plan es llevar a estos casos de uso de la más alta prioridad al diseño lógico y fases posteriores e implantar una versión 1.0 (que funcione) del sistema a tiempo y dentro del presupuesto.

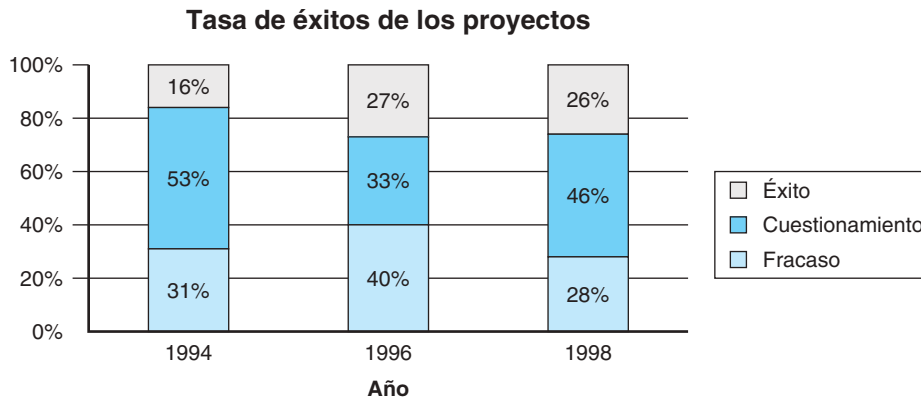
## Una introducción a la modelación de casos de uso

Uno de los retos fundamentales de importancia vital para cualquier equipo de desarrollo de sistemas de información, y especialmente para el analista de sistemas, es la capacidad para obtener, de los involucrados, los requerimientos correctos y necesarios del sistema, y especificarlos de manera inteligible para los involucrados de modo que estos requerimientos se verifiquen y se validen. De hecho, éste ha sido el caso durante muchos años, como lo escribió el distinguido autor Fred Brooks en su famoso artículo de 1987:

La parte individual más difícil de la construcción de un sistema de software es decidir con precisión qué construir. Ninguna otra parte del trabajo conceptual es tan difícil como establecer los requerimientos técnicos detallados, incluyendo todas las interfaces con las personas con las máquinas y con otros sistemas de software. Ningún otro trabajo invalida tanto al sistema resultante si se hace mal. Ninguna otra parte es más difícil de rectificar posteriormente.

El personal de tecnología de información siempre ha tenido problemas al tratar de especificar los requerimientos, especialmente los funcionales, de los usuarios. En el pasado hemos usado herramientas tales como los modelos de datos, los modelos de procesos, los prototipos, y las especificaciones de requerimientos que entendíamos y nos sentíamos cómodos con ellos, aunque eran difíciles de entender para cualquier usuario que no tuviera un entrenamiento en las prácticas de desarrollo de software. Debido a esto, muchos proyectos de desarrollo estaban, y todavía están, plagados con problemas de: menoscabo del alcance, sobregiro de costos y retraso del programa. Con frecuencia se desarrollan y se instalan sistemas que realmente no satisfacen las necesidades de los usuarios. Algunos se archivan y no se usan nunca, y un gran porcentaje se cancela aun antes de terminar el esfuerzo de desarrollo. Una compañía de investigación muy conocida, el Standish Group, estudió 23 000 aplicaciones de TI en 1994, 1996 y 1998.<sup>1</sup> Como se muestra en la figura 6.1, el estudio de 1998 encontró que solamente un poco más de un cuarto de los proyectos en 1998 tuvieron éxito (dentro del presupuesto, a tiempo, y cubría todas las necesidades). Más de un cuarto de ellos fracasó (se cancelaron antes de terminarse). Un poco más de la mitad fueron lo que Standish consideraba como cuestionables: el proyecto estaba terminado y operaba, pero se terminó ya sea con un sobregiro en el presupuesto, rebasando el tiempo estimado, o sin todas las características especificadas por los usuarios. La buena noticia que se refleja en estos estudios y en otros es que las prácticas y los métodos que estamos usando para desarrollar los sistemas de información están mejorando. La industria de desarrollo de software ha aprendido que con objeto de planear, analizar, diseñar,

<sup>1</sup> The Standish Group International, Inc., “CHAOS: A Recipe for Success” (versión electrónica), 1999. Obtenido el 5 de diciembre de 2002, de [www.pm2go.com/sample\\_research/chaos1998.pdf](http://www.pm2go.com/sample_research/chaos1998.pdf). El Standish Group es reconocido por su independiente investigación y análisis de la industria de tecnología de la información.

**FIGURA 6.1**

Tasas de éxito de los proyectos tal como lo reporta el grupo Standish

Fuente: The Standish Group International, Inc., "Chaos: A Recipe for Success" (versión electrónica), 1999, [www.pm2go.com/sample\\_research/chaos1998.pdf](http://www.pm2go.com/sample_research/chaos1998.pdf)

construir e implantar un sistema de información con éxito, el analista de sistemas debe entender primero las necesidades de los involucrados y las razones por las cuales debe desarrollarse el sistema; un concepto llamado **desarrollo centrado en el usuario**. Al centrarse en los usuarios del sistema, el analista puede concentrarse en cómo se va a usar el sistema y no en cómo va a construirse. La **modelación de casos de uso** es un enfoque que facilita el desarrollo centrado en el usuario.

La modelación de casos de uso tiene sus raíces en la modelación orientada a objetos, y usted aprenderá más acerca de cómo aplicar la modelación de casos de uso en el análisis orientado a objetos, pero también ha ganado popularidad en los ambientes de desarrollo que no son orientados a objetos. Usted aprenderá en los capítulos restantes de este libro cómo la modelación de casos de uso complementa al análisis tradicional de sistemas y a las herramientas de diseño tales como la modelación de datos y la modelación de procesos y al mismo tiempo, proporciona una base para las decisiones de arquitectura y las decisiones de diseño de interfaces.

La modelación de casos de uso fue concebida originalmente por el Dr. Ivan Jacobson en 1986 y adquirió popularidad tras la publicación de su libro, *Object-Oriented Software Engineering*, en 1992. El Dr. Jacobson usó la modelación de casos de uso como marco para su metodología de objetos; él la usó con mucho éxito para desarrollar los sistemas de información orientados a objetos. La modelación de casos de uso ha resultado ser una valiosa ayuda para enfrentar los retos de determinar cuál sistema se requiere hacer desde la perspectiva del usuario y del involucrado. En la actualidad se le reconoce ampliamente como la mejor práctica para la definición, la documentación y la comprensión de los requerimientos funcionales de un sistema de información.

El uso de la modelación de casos de uso facilita y alienta la participación del usuario, que es uno de los principales factores críticos de éxito para asegurar el éxito del proyecto. Además, la modelación de casos de uso proporciona los siguientes beneficios:

- Proporciona una herramienta para capturar los requerimientos funcionales.
- Ayuda a descomponer el alcance del sistema en piezas más manejables.
- Proporciona un medio de comunicación con los usuarios y con otros involucrados en relación con la funcionalidad del sistema. Los casos de uso presentan un lenguaje común que se entiende fácilmente por los diferentes involucrados.
- Proporciona un medio para identificar, asignar, rastrear, controlar y administrar las actividades de desarrollo de sistemas, especialmente un desarrollo por incrementos e iterativo.
- Proporciona una ayuda para estimar el alcance de proyecto, el esfuerzo a realizar y la programación.
- Proporciona una línea de base para pruebas en términos de la definición de los planes y casos de prueba.
- Proporciona una línea de base para los sistemas y manuales que ayudan al usuario así como para la documentación de desarrollo del sistema.
- Proporciona una herramienta para el seguimiento de los requerimientos.
- Proporciona un punto inicial para la identificación de los objetos o entidades de datos.
- Proporciona especificaciones funcionales para el diseño de las interfaces entre el usuario y el sistema.
- Proporciona un medio para definir los requisitos de acceso a la base de datos en términos de crear, cambiar, borrar y leer.
- Proporciona un marco para impulsar el proyecto de desarrollo de sistemas.

### desarrollo centrado en el usuario

Proceso de desarrollo de sistemas basado en la comprensión de las necesidades de los involucrados y las razones por las que deben desarrollarse los sistemas.

### modelación de casos de uso

Proceso de modelación de las funciones de un sistema en términos de eventos de negocios, quiénes iniciaron los eventos y cómo responde el sistema a estos eventos.

## Conceptos de sistemas en la modelación de casos de uso

### diagrama de casos de uso

Diagrama que ilustra las interacciones entre el sistema y los sistemas y usuarios externos. En otras palabras, describe gráficamente quién va a usar el sistema y de qué manera el usuario espera interactuar con el sistema.

### descomposición funcional

Acto de desarmar un sistema en subcomponentes.

### narración del caso de uso

Texto que describe el evento del negocio y la forma en que el usuario interactuará con el sistema para lograr la tarea.

**casos de uso** Una secuencia de pasos relacionados (un escenario), tanto automatizado como manual, con el propósito de completar una sola tarea del negocio.

Símbolo de caso de uso

### FIGURA 6.2

Un ejemplo de un diagrama de modelo de casos de uso

Existen dos elementos primordiales cuando se realiza la modelación de casos de uso. El primero es el **diagrama de casos de uso**, que ilustra gráficamente al sistema como una colección de casos, actores (usuarios) y sus relaciones. Este diagrama comunica a un alto nivel el alcance de los eventos de negocios que el sistema debe procesar. En la figura 6.2 se muestra un ejemplo de un diagrama de casos de uso. Muestra cada una de las funciones del sistema, o eventos del negocio (dibujados en las elipses), y a los actores, o usuarios del sistema, quienes interactúan con esas funciones. Como usted ve en la figura 6.2, los actores pueden situarse a ambos lados del conjunto de las figuras de casos de uso e interactuar con uno o más de los casos. El diagrama de casos de uso es muy sencillo. Pero con éste, comienza un importante proceso llamado **descomposición funcional**, el acto de descomponer un sistema en sus subcomponentes. Es imposible entender inmediatamente el sistema completo, pero es posible entender y especificar cada parte del mismo.

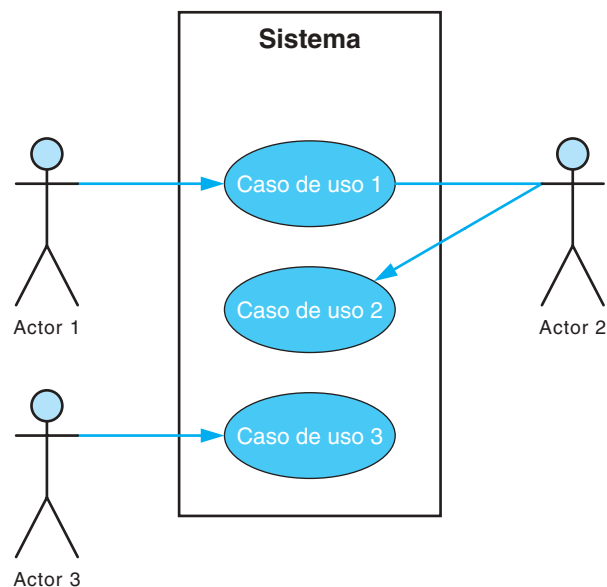
El segundo elemento, llamado la **narración del caso de uso**, describe los detalles de cada evento de negocios y especifica cómo interactúan los usuarios con el sistema durante ese evento. La narración del caso de uso se estudiará con detalle más adelante en el capítulo.

### > Los casos de uso

La modelación de los casos de uso identifica y describe las funciones del sistema mediante el uso de una herramienta llamada **casos de uso**. Dichos casos describen las funciones del sistema desde la perspectiva de los usuarios externos de una forma y con una terminología que ellos entienden. El alcanzar exacta y minuciosamente esto requiere un alto nivel de participación por parte del usuario, así como de un experto en la materia que sea versado en el proceso o evento de negocios.

Los casos de uso se representan gráficamente con una elipse horizontal con el nombre del caso que aparece encima, debajo o dentro de la elipse. Un caso de uso representa un objetivo individual del sistema y describe una secuencia de actividades y de interacciones del usuario para tratar de alcanzar el objetivo. La creación de los casos de uso ha probado ser una técnica excelente para entender y documentar mejor los requerimientos del sistema. Un caso de uso por sí solo no se considera como un requerimiento funcional, pero la historia (el escenario) que relata el caso de uso consiste en uno o más requerimientos.

Inicialmente, los casos de uso se definen durante la etapa de los requerimientos del ciclo de vida y se refinarán adicionalmente a lo largo del ciclo de vida. Durante la iden-



tificación de los requisitos, los casos de uso se emplean para capturar la esencia de los problemas de negocios y para modelar (a un alto nivel) la funcionalidad del sistema propuesto. Adicionalmente, son el punto inicial para la identificación de las entidades de datos (cubierto en el capítulo 7) o de los objetos del sistema (cubierto en el capítulo 9). Durante el análisis de requisitos, los casos de uso se refinan para modelar el uso del sistema con más detalle. En otras palabras, se actualizan para especificar lo que los usuarios están tratando de alcanzar y el porqué. Estos casos ayudan en la definición de los prototipos o interfaces del usuario. Durante el diseño los casos de uso se refinan para modelar cómo los usuarios usarán realmente el sistema con respecto a cualquier interfaz y a las restricciones del sistema. Estos tipos de casos de uso ayudan a identificar el comportamiento del objeto o del sistema y a diseñar las especificaciones de código y de interfaz, así como para servir para planear las pruebas del sistema. En la construcción, los casos de uso ayudan a los desarrolladores en la programación y en las pruebas. Estos casos también sirven como la línea base para preparar cualquier documentación del sistema y para el usuario, además de servir como herramientas para el entrenamiento de los usuarios. Y como los casos de uso contienen una enorme cantidad de detalle de la funcionalidad del sistema, éstos serán un recurso constante para validar al sistema.

## > Actores

Los casos de uso se inician o son generados por los usuarios externos llamados **actores**. Un actor inicia la actividad del sistema, un caso de uso, con el propósito de terminar alguna tarea de negocios que produzca algo con valor apreciable. Utilicemos el ejemplo de un estudiante universitario que se inscribe en los cursos del semestre de otoño. El actor sería el *estudiante*, y el evento de negocios, o caso de uso, sería *Inscribirse en Curso*. Un actor representa un papel desempeñado por un usuario que interactúa con el sistema y no significa que retrate a una persona o un puesto de trabajo. De hecho, un actor no tiene que ser humano. Puede ser una organización, otro sistema de información, un dispositivo externo tal como un sensor de calor, o aun el concepto de tiempo (que se estudiará un poco después). Un actor se representa gráficamente como una figura de línea rotulada con el nombre del papel que juega el actor.

Es importante observar que hay principalmente cuatro tipos de actores:

- **Actor primario de negocios:** el interesado que se beneficia principalmente de la ejecución de un caso de uso al recibir algo de valor medible u observable. El actor primario de negocios puede o no iniciar el evento de negocios. Por ejemplo, en el evento de negocios de un empleado que recibe un cheque como pago (algo con valor medible) del sistema de nómina cada viernes, el empleado no inicia el evento pero es el receptor primario del algo de valor.
- **Actor primario del sistema:** el involucrado que tiene una interfaz directa con el sistema para iniciar u ocasionar el evento de negocios o de sistema. Los actores primarios del sistema pueden interactuar con los actores primarios de negocios con el propósito de usar el sistema real. Ellos facilitan el evento a través del uso directo del sistema para beneficio del actor primario de negocios. Los ejemplos incluyen un dependiente de una tienda de abarrotes que selecciona los artículos para el cliente que compra abarrotes, una operadora de teléfonos que desproporciona información del directorio a un cliente, y un cajero de banco que procesa una transacción bancaria. El actor principal de negocios y el actor principal de sistema pueden ser la misma persona para eventos en los cuales el actor de negocios tiene una interfaz directa con el sistema; por ejemplo, una persona que reserva la renta de un automóvil a través de un sitio Web.
- **Actor externo servidor:** el involucrado que responde a una solicitud desde el caso de uso (por ejemplo, un buró de crédito que autoriza el pago mediante tarjeta de crédito).
- **Actor externo receptor:** el involucrado que no es el actor primario pero que recibe algo de valor medible u observable (salida) proveniente del caso de uso (por ejemplo, un almacén que recibe una orden de embalaje para preparar un flete después de que un cliente ha colocado una orden).

**actor** Cualquier cosa que necesite interactuar con el sistema para intercambiar información.



Símbolo de un actor

En muchos sistemas de información hay eventos de negocios ocasionados por el calendario o la hora del reloj. Considere los siguientes ejemplos:

- El sistema de facturación de una compañía de tarjetas de crédito genera automáticamente sus estados de cuenta en el quinto día de cada mes (fecha de facturación).
- Un banco concilia sus transacciones con cheques todos los días a las 5 p.m.
- Cada noche se genera automáticamente un reporte que lista cuáles cursos están cerrados a las inscripciones (ya no hay más lugares) y cuáles cursos todavía están abiertos.

**evento temporal** Evento del sistema que es activado por el tiempo.

Estos eventos son ejemplos de **eventos temporales**. ¿Quién sería el actor? Todos los eventos listados antes se ejecutaron (o se ocasionaron) automáticamente: cuando se cumplió cierta fecha o cierta hora. Por eso decimos que el actor de un evento temporal es el tiempo.

## > Relaciones

Una relación se ilustra como una línea entre dos símbolos en el diagrama de casos de uso. El significado de las relaciones puede diferir dependiendo de cómo se dibujen las líneas y qué tipos de símbolos conectan. En las siguientes secciones definiremos las diferentes relaciones que se encuentran en un diagrama de casos de uso.

**asociación** Relación entre un actor y un caso de uso en la que interactúan entre sí.

**Asociaciones** Existe una relación entre un actor y un caso de uso siempre que el caso describa una interacción entre éstos. A esta relación se le denomina **asociación**. Como se indica en la figura 6.3, una asociación se modela como una línea continua que conecta al actor y al caso de uso (1) indica que el caso fue iniciado por el actor en el otro extremo de la línea. Las asociaciones sin cabezas de flecha (2) indican una interacción entre el caso de uso y un actor externo servidor o receptor. Si un actor se asocia con un caso de uso, decimos que el actor se *comunica* con el caso. Las asociaciones pueden ser bidireccionales o unidireccionales.

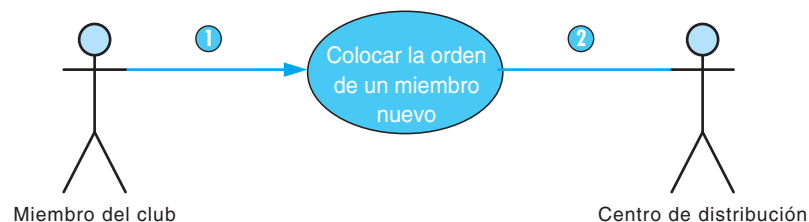
### caso de uso de extensión

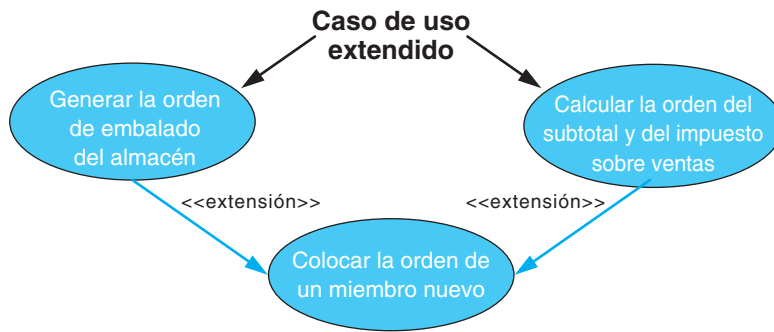
Un caso de uso que consiste en los pasos extraídos de otro más complejo para simplificar el caso original y, así, ampliar su funcionalidad. El caso de uso de extensión de hecho extiende la funcionalidad del caso de uso original.

**Extensión** Un caso de uso puede contener una funcionalidad compleja que consiste de varios pasos que hacen difícil entender a la lógica del caso. Con objeto de simplificar el caso de uso y hacer que se entienda más fácilmente, podemos extraer los pasos más complejos para formar su propio caso. El caso resultante se llama un **caso de uso de extensión** ya que extiende la funcionalidad del caso de uso original. La relación entre el caso de uso de extensión y el que se está extendiendo se llama una relación de *extensión*. Un caso de uso puede tener muchas relaciones de extensión, pero un caso de uso de extensión puede ser invocado solamente por el caso que se esté extendiendo. Como se ilustra en la figura 6.4, la relación de extensión se representa como una línea con cabeza de flecha (ya sea continua o segmentada) que comienza en el caso de uso de extensión y que apunta al caso de uso que se está extendiendo. Cada línea de relación de extensión se rotula como “<<extensión>>”. Generalmente los casos de uso de extensión no se identifican en la fase de requerimientos, sino en la de análisis.

**FIGURA 6.3**

Ejemplo de una relación de asociación





**FIGURA 6.4**

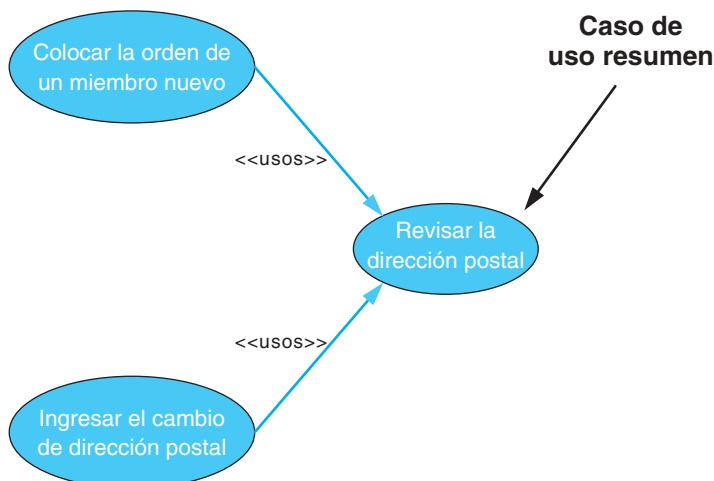
Ejemplo de una relación de extensión

**Usos (o inclusión)** Muy comúnmente, usted puede descubrir dos o más casos de uso que ejecuten pasos de funcionalidad idéntica. Lo mejor es extraer estos pasos comunes para formar un caso de uso separado que sea propio llamado un **caso de uso resumen**. Un caso de uso resumen representa una forma de “reuso” y es una herramienta excelente para reducir la redundancia entre los casos de uso. Un caso de uso resumen está disponible como referencia (o uso) para cualquier otro caso de uso que requiera su funcionalidad. La relación entre el caso de uso resumen y el caso de uso que lo usa se llama una relación de *uso* (algunas herramientas de la modelación de casos de uso lo denominan una relación de *inclusión*). La relación de uso que se presenta en la figura 6.5 se ilustra con una línea con cabeza de flecha (ya sea continua o segmentada) que comienza en el caso de uso oficial y que apunta al caso de uso que esté usando. Cada línea de relación de uso se rotula “<<uso>>”. Generalmente los casos de uso resúmenes no se identifican en la fase de requisitos, sino en la de análisis.

**caso de uso resumen** Un caso de uso que reduce la redundancia entre dos o más casos de uso al combinar los pasos comunes existentes en estos casos. Otro caso de uso *utiliza* o *incluye* el caso de uso resumen.

**Dependencia** Como administrador de proyecto o desarrollador líder, es de mucha ayuda saber cuáles casos de uso tienen una dependencia sobre otros casos de uso con objeto de determinar la secuencia en que es necesario desarrollar los casos de uso. Si usamos el negocio bancario como ejemplo, el caso de uso Hacer un retiro no puede ejecutarse hasta que haya ocurrido el caso de uso Abrir una cuenta bancaria. Debido a estas dependencias, el equipo de desarrollo muy probablemente escogerá desarrollar el caso de uso Abra una cuenta bancaria primero, en segundo lugar el caso de uso Haga un depósito, y en tercer lugar el caso de uso Haga un retiro para los propósitos de Condiciones de uso y pruebas. Un diagrama de casos de uso que modele las dependencias de caso de usos del sistema mediante el uso de la relación de **dependencia** proporciona un modelo que es una herramienta excelente para propósitos de planeación y de programación. La relación de dependencia tal como se presenta en la figura 6.6 se ilustra con una línea con cabeza de flecha

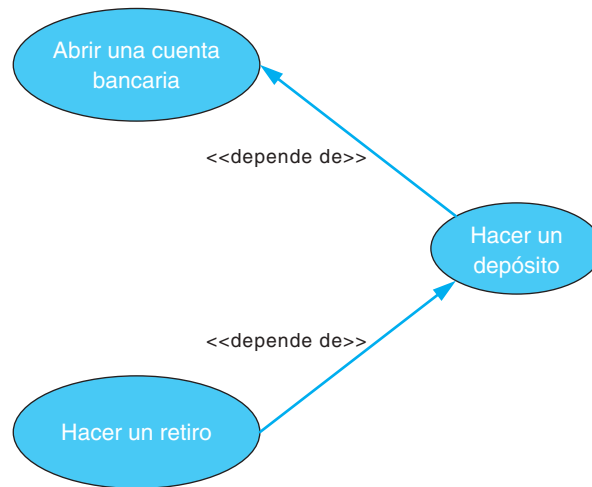
**dependencia** Relación entre casos de uso que indica que un caso de uso no puede realizarse hasta que se haya realizado el otro caso de uso.



**FIGURA 6.5**

Ejemplo de una relación de usos

**FIGURA 6.6**  
Ejemplo de una relación de dependencia

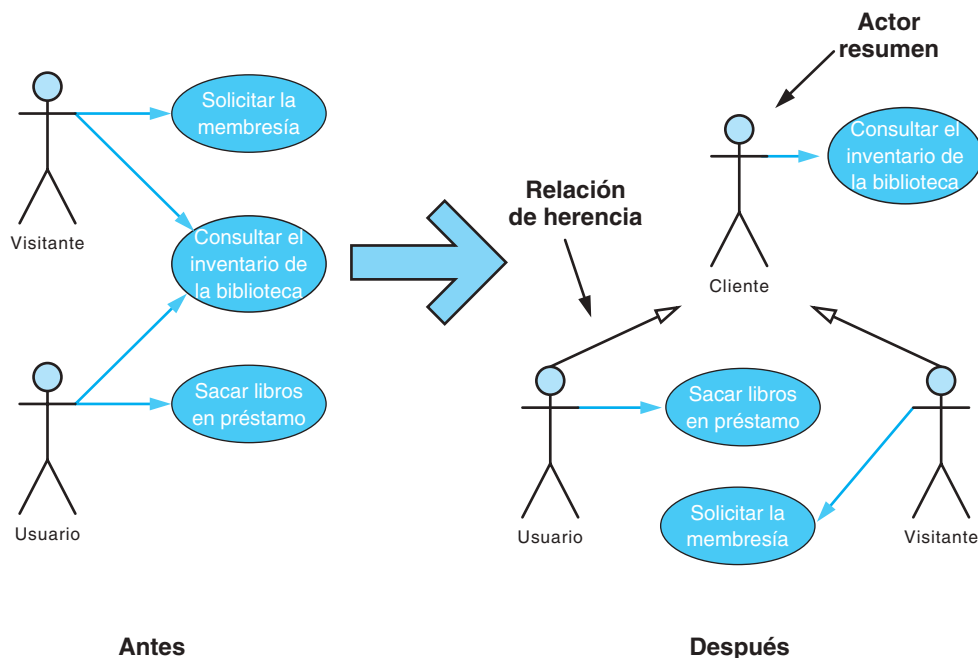


(ya sea continua o segmentada) que comienza en un caso de uso y que apunta al caso de uso del cual depende. La línea de relación de dependencia se rotula “<<depende de>>”.

**Herencia** Cuando dos o más actores comparten un comportamiento común (en otras palabras, pueden iniciar el mismo caso de uso) lo mejor es extrapolar este comportamiento común y asignarlo a un nuevo actor *resumen* con objeto de reducir la comunicación redundante con el sistema. Por ejemplo, un *usuario* de una biblioteca es un miembro que tiene una tarjeta que está autorizado a “Buscar en el inventario de la biblioteca” así como a “Sacar libros en préstamo” de la biblioteca. Como muchas bibliotecas son instituciones públicas, ellos dan la bienvenida a los *visitantes* para que usen sus servicios en el lugar tales como “Buscar en el inventario de la biblioteca” pero a los visitantes no se les permite la extensión de servicios (tal como “Sacar libros en préstamo”) que están reservados para los usuarios. Al crear un actor resumen llamado *cliente*, del cual van a heredar el *usuario* y el *visitante*, tenemos que modelar la relación solamente una vez iniciando el caso de uso Buscar en el inventario de la biblioteca. En el diagrama de caso de uso se ilustra la relación **herencia** con el tipo de flecha mostrada en la sección “Después” de la figura 6.7.

**herencia** En los casos de uso, relación entre actores creada para simplificar el dibujo cuando un actor resumen hereda las funciones de múltiples actores.

**FIGURA 6.7**  
Ejemplo de una relación de herencia



Antes

Después



## El proceso de la modelación de los casos de uso para los requerimientos

El objetivo de construir el modelo de casos de uso para los requerimientos es producir y analizar suficiente información de los requerimientos para preparar un modelo que comunique lo que se requiere desde la perspectiva del usuario pero que esté libre de los detalles específicos acerca de cómo va a construirse e implantarse el sistema. El seguimiento de este enfoque producirá después un diseño que es más robusto y que posiblemente sea menos impactado por el cambio. Pero para estimar y programar el proyecto con efectividad puede ser necesario que el modelo incluya “hipótesis de implantación de sistemas” preliminares para ayudar en esas actividades. Es crítico que el analista no se deslice a un estado de *parálisis del análisis* al preparar este modelo. La velocidad es la clave. No todos los hechos serán aprendidos durante esta fase del ciclo de vida, pero al utilizar un desarrollo iterativo y por incrementos, la metodología permite después la introducción de nuevos requerimientos en el proyecto sin impactar seriamente el despliegado de la solución final. Los pasos requeridos para producir este modelo son los siguientes:

1. Identificar a los actores de negocios.
2. Identificar los casos de uso para los requerimientos de negocios.
3. Construir un diagrama para el modelo de casos de uso.
4. Documentar las narraciones de casos de uso para los requerimientos de negocios.

### > Paso 1: Identificar a los actores de negocios

¿Por qué identificar primero a los actores? Al centrarse en los actores, usted puede concentrarse en cómo se usará el sistema y no en cómo se construirá. Además ayuda a refinar y definir aún más el alcance y las fronteras del sistema. Los actores también determinan que tan completos están los requerimientos del sistema.<sup>2</sup> Un beneficio de identificar a los actores primero es que al hacerlo se identifica a los candidatos que podemos entrevistar y observar posteriormente para terminar el esfuerzo de modelación de los casos de uso. Además, es posible usar a estas mismas personas para verificar y validar los casos de uso cuando terminan.

¿Dónde busca usted a los actores potenciales? Las siguientes referencias son fuentes excelentes:

- Un diagrama de contexto que identifique el alcance o las fronteras del sistema.
- La documentación del sistema y los manuales del usuario existentes.
- Minutas de las juntas y los talleres del proyecto.
- Documentos de los requerimientos, carta constitutiva del proyecto, o declaración del trabajo existente.

Al buscar actores haga las siguientes preguntas:

- ¿Quién o qué proporciona las entradas al sistema?
- ¿Quién o qué recibe las salidas del sistema?
- ¿Se requieren interfaces con otros sistemas?
- ¿Existen eventos que son originados automáticamente en un instante predeterminado?
- ¿Quién mantendrá la información en el sistema?

Los actores deberán nombrarse con un sustantivo o con una frase sustantiva.

Cuando usted identifique a un actor, cree el texto de una definición de ese actor de acuerdo con la perspectiva del usuario y usando sus términos. La figura 6.8 es una plantilla de un glosario de actores que puede usarse para documentar a los actores. Este ejemplo contiene un listado parcial de los actores del Sistema de Servicios para los Miembros de SoundStage.

<sup>2</sup> Frank Armour y Granville Miller, *Advance Use Case Modeling* (Boston: Addison-Wesley, 2001).

**FIGURA 6.8**

Lista parcial de los actores del Sistema de servicios para los miembros de SoundStage

**Glosario de actores**

Término	Sinónimo	Descripción
1. Miembro potencial		Una persona o corporación que ingresa una orden de suscripción con objeto de incorporarse al club.
2. Miembro del club	Miembro	Una persona o corporación que se adhiere al club mediante un convenio.
3. Ex miembro	Miembro inactivo	Un tipo de miembro que ha cumplido con la obligación del acuerdo, no ha colocado una orden en los últimos seis meses pero que todavía tiene derechos.
4. Mercadeo		Organización responsable de la creación de los programas de promoción y suscripción y de la generación de ventas para la compañía.
5. Servicios a los miembros		Organización responsable de suministrar un punto de contacto para los clientes de SoundStage Entertainment en términos de acuerdos y órdenes.
6. Centro de distribución	Almacén	Entidad que aloja y mantiene al inventario de productos de SoundStage Entertainment y que procesa los fletes y las devoluciones de los clientes.
7. Cuentas por cobrar		Organización responsable del procesamiento de los pagos y las facturas de los clientes así como de mantener la información de las cuentas de los clientes.
8. Tiempo		Concepto de actor responsable de ocasionar eventos temporales.

## > Paso 2: Identificar los casos de uso para los requerimientos de negocios

Un sistema de información típico puede constar de docenas de casos de uso. Durante el análisis de requerimientos tratamos de identificar y documentar solamente los más críticos, complejos e importantes, frecuentemente denominados casos de uso *esenciales* debido a consideraciones de tiempo y de costo. Un **caso de uso de requerimientos del negocio** captura las interacciones con el usuario de manera que esté libre de los detalles de la tecnología y de la implantación. Como un caso de uso describe cómo interactúa un actor en el mundo real con el sistema, una técnica excelente para encontrar los casos de uso de los requerimientos de negocios es examinar a los actores y cómo van a usar al sistema. Al buscar los casos de uso haga las siguientes preguntas:

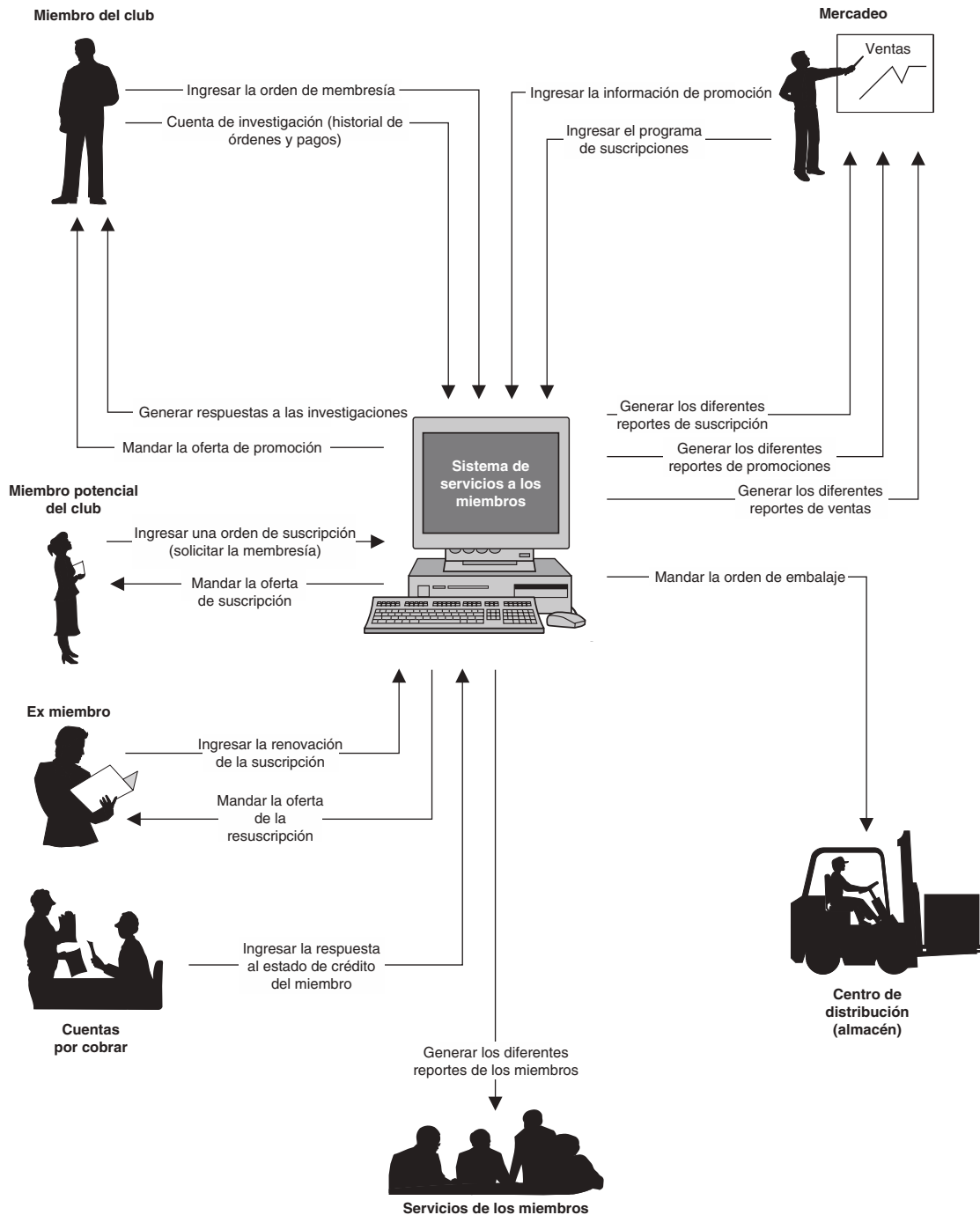
- ¿Cuáles son las principales tareas del actor?
- ¿Qué información necesita el actor del sistema?
- ¿Qué información proporciona el actor al sistema?
- ¿Necesita el sistema informar al actor de cambios o eventos que hayan ocurrido?
- ¿Necesita informar el actor al sistema de cambios o eventos que hayan ocurrido?

Nuevamente, un diagrama de contexto es una fuente excelente para encontrar casos de uso potenciales. Los diagramas de contexto se estudiaron en el capítulo 4. Vienen de la modelación tradicional de procesos (capítulo 8) pero son útiles aun para proyectos que adoptan un enfoque orientado a objetos. Examinemos el diagrama de contexto del Sistema de Servicios para Miembros de SoundStage en la figura 6.9. Podemos identificar los casos de uso potenciales al examinar el diagrama e identificar las entradas y salidas primarias del sistema y las partes externas que las ingresan y las reciben. Las entradas primarias que ocasionan los eventos de negocios (por ejemplo, *Ingresar la orden del miembro*) dentro de la organización se consideran casos de uso, y las partes externas que suministran estas entradas se consideran actores (por ejemplo, *Miembro del club*). Es importante observar que las entradas que son el resultado de peticiones al sistema no indican un

### caso de uso de requerimientos del negocio

Caso de uso creado durante el análisis de requerimientos para capturar la interacción entre un usuario y el sistema con independencia de los detalles de tecnología e implantación, también llamado un caso de uso esencial.

**Diagrama de contexto de los servicios para los miembros**



**FIGURA 6.9** Diagrama de contexto del Sistema de servicios para los miembros de SoundStage

caso de uso separado; y tal como una compañía de tarjetas de crédito que responde a una solicitud de autorización o, como se presenta en la figura 6.9, el actor *Cuentas por cobrar* que responde con *Información del nivel crediticio del miembro*.

Los casos de uso se denominan con un enunciado que especifica la meta del actor, tal como *Ingresar una orden de suscripción*. Los casos de uso que son eventos temporales,

generalmente se identifican como el resultado de analizar las salidas clave del sistema. Por ejemplo, cualquier salida que se genere basada en el tiempo o en una fecha, tales como los reportes mensuales o anuales, se considera como un caso de uso, y el actor, como usted recuerda, es el tiempo. En la figura 6.9 supongamos que uno de los diferentes reportes que recibe Servicios para los miembros es un *Reporte de conformidad por omisión a 10-30-60 días* que se genera automáticamente de manera cotidiana. Como la generación del reporte es originada por el tiempo se requiere un caso de uso para procesar el evento, y lo denominaríamos *Reporte de conformidad por omisión a 10-30-60 días generado rutinariamente*. Es importante observar que muchas veces los reportes individuales no se listan en un diagrama de contexto porque son demasiado numerosos y podrían saturar el diagrama y hacerlo difícil de leer. Es la responsabilidad del analista de sistemas investigar con los involucrados apropiados el tipo de salidas que reciben y sus características, en términos de volumen, frecuencia y mecanismo de accionamiento, con objeto de identificar a los “casos de uso ocultos”.

La figura 6.10 es una plantilla de un glosario de casos de uso que puede usarse para documentarlos. Este ejemplo contiene una lista parcial de casos de uso y de los actores del Sistema de servicios para los miembros de SoundStage identificados del diagrama de contexto así como de otras fuentes.

### > Paso 3: Construir el diagrama del modelo de casos de uso

Una vez que los casos de uso y los actores han sido identificados, puede usarse un diagrama de modelo de casos de uso para ilustrar gráficamente el alcance y las fronteras

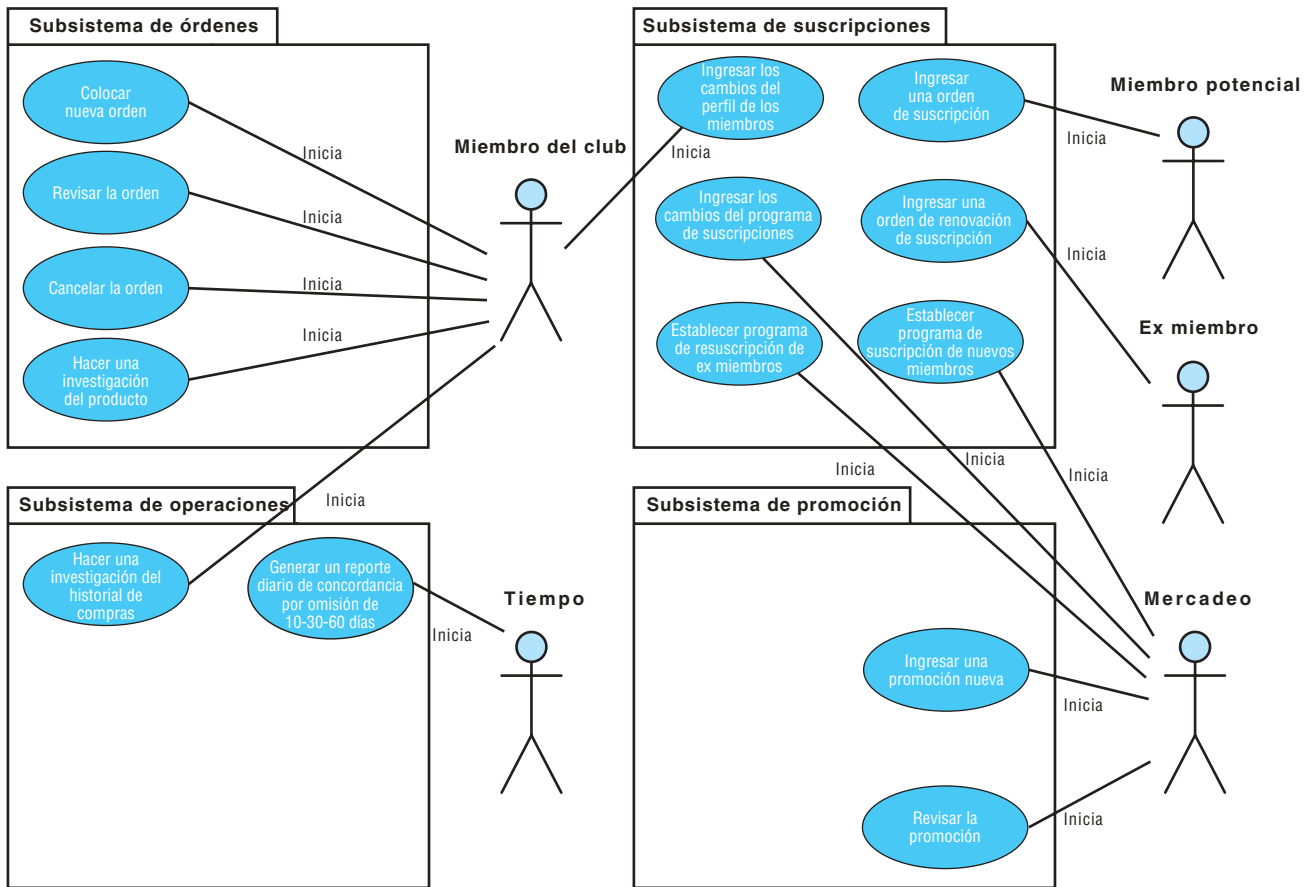
**FIGURA 6.10** Lista parcial de los casos de uso del Sistema de servicios para los miembros de SoundStage

Glosario de casos de uso		
Nombre del caso de uso	Descripción del caso de uso	Actores y papeles participantes
Ingresar una orden de suscripción	Este caso de uso describe el evento de un miembro potencial que solicita ingresar al club por suscripción. (“Tome un paquete de 12 CD por un chelín y concuerde en comprar cuatro más a precios regulares en menos de dos años.”)	<ul style="list-style-type: none"> <li>• Miembro potencial (negocio primario)</li> <li>• Centro de distribución (receptor externo)</li> </ul>
Ingresar una orden de renovación de suscripción	Este caso de uso describe el evento de un ex miembro que solicita reingresar al club por suscripción. (“Tome un paquete de 12 CD por un chelín y concuerde en comprar cuatro más a precios regulares en menos de dos años.”)	<ul style="list-style-type: none"> <li>• Ex miembro (negocio primario)</li> <li>• Centro de distribución (receptor externo)</li> </ul>
Ingresar los cambios del perfil del miembro	Este caso de uso describe el evento de un miembro del club que ingresa cambios de su perfil en cosas tales como la dirección postal, la dirección de e-mail, los códigos de privacidad y las preferencias de ordenado.	<ul style="list-style-type: none"> <li>• Miembro del club (negocio primario)</li> </ul>
Colocar una nueva orden	Este caso de uso describe el evento de un miembro del club que ingresa una orden para los productos de SoundStage.	<ul style="list-style-type: none"> <li>• Miembro del club (negocio primario)</li> <li>• Centro de distribución (receptor externo)</li> <li>• Cuentas por pagar/cuentas por cobrar (servidor externo)</li> </ul>
Revisar la orden	Este caso de uso describe el evento de un miembro del club que revisa una orden colocada previamente. (La orden no debe haber sido mandada.)	<ul style="list-style-type: none"> <li>• Miembro del club (negocio primario)</li> <li>• Centro de distribución (receptor externo)</li> <li>• Cuentas por pagar/cuentas por cobrar (servidor externo)</li> </ul>

**FIGURA 6.10** Conclusión

Cancelación de la orden	Este caso de uso describe el evento de un miembro del club que cancela una orden previamente colocada. (La orden no debe haberse mandado.)	<ul style="list-style-type: none"> <li>• Miembro del club (negocio primario)</li> <li>• Centro de distribución (receptor externo)</li> <li>• Cuentas por pagar/cuentas por cobrar (servidor externo)</li> </ul>
Hacer una investigación de producto	Este caso de uso describe el evento de un miembro del club que ve los productos para una posible compra. (Impulsado por el requerimiento de acceso a la Red.)	<ul style="list-style-type: none"> <li>• Miembro del club (negocio primario)</li> </ul>
Hacer una investigación del historial de la compra	Este caso de uso describe el evento de un miembro del club que ve su historial de compra. (Tiempo límite de tres años.)	<ul style="list-style-type: none"> <li>• Miembro del club (negocio primario)</li> </ul>
Establecer un nuevo programa de suscripción de miembros	Este caso de uso describe el evento del departamento de mercadeo que establece un nuevo plan de suscripción de membresía para atraer nuevos miembros.	<ul style="list-style-type: none"> <li>• Mercadeo (negocio primario)</li> </ul>
Ingresar los cambios del programa de suscripciones	Este caso de uso describe el evento del departamento de mercadeo que cambia el plan de suscripción para los miembros del club (por ejemplo, la extensión del periodo de cumplimiento).	<ul style="list-style-type: none"> <li>• Mercadeo (negocio primario)</li> </ul>
Establecer un programa de resuscripción de ex miembros	Este caso de uso describe el evento del departamento de mercadeo que establece un plan de resuscripción para atraer nuevamente a los ex miembros.	<ul style="list-style-type: none"> <li>• Mercadeo (negocio primario)</li> </ul>
Ingresar los cambios del perfil de los miembros	Este caso de uso describe el evento del departamento de mercadeo que establece un nuevo plan de promoción para tentar a los miembros activos e inactivos para que ordenen el producto. (Nota: Una promoción se caracteriza por anuncios específicos, generalmente nuevos, de que la compañía está tratando de vender a un precio especial. Estas promociones se integran en un catálogo que se manda [o se comunica] a todos los miembros.)	<ul style="list-style-type: none"> <li>• Mercadeo (negocio primario)</li> </ul>
Revisar la promoción	Este caso de uso describe el evento del departamento de mercadeo que revisa una promoción.	<ul style="list-style-type: none"> <li>• Mercadeo (negocio primario)</li> </ul>
Generar el reporte diario de conciliación de 10-30-60 días por omisión	Este caso de uso describe el evento de un reporte que se genera cotidianamente para listar a los miembros que no hayan cumplido con el acuerdo de comprar el número requerido de productos esbozados cuando se suscribieron. Este reporte está ordenado de acuerdo con el orden de los miembros que tienen 10 días de vencimiento, 30 días de vencimiento y 60 días de vencimiento.	<ul style="list-style-type: none"> <li>• Tiempo (actor iniciante)</li> <li>• Servicios a los miembros (receptor primario,* externo)</li> </ul>

\* Considerado primario porque recibe algo de valor medible.



**FIGURA 6.11** Diagrama del modelo de caso de uso del Sistema de servicios para los miembros de SoundStage

del sistema. El diagrama de casos de uso para los casos de uso listados en la figura 6.10 se muestra en la figura 6.11. Se creó usando el *System Architect* del Software Popkin y representa las relaciones entre los actores y los casos de uso. Además, los casos de uso han sido agrupados en subsistemas de negocios. Los subsistemas (símbolo de paquete de UML) representan las áreas funcionales lógicas de los procesos de negocios. La división del comportamiento del sistema en subsistemas es muy importante para entender la arquitectura del sistema y es clave para definir su estrategia de desarrollo; cuáles casos de uso serán desarrollados primero y por quién. Hemos rotulado las asociaciones entre los actores y los casos de uso como “iniciados” porque la herramienta no soportaba líneas con cabezas de flecha en ese momento. Tampoco incluimos a los actores proveedores y receptores externos debido a las limitaciones de espacio. La modelación de todos los casos de uso de un sistema específico puede requerir la creación de varios diagramas de modelos de casos de uso; como usted recuerda, un sistema puede contener docenas de casos de uso. En ese caso tal vez usted quiera crear un diagrama de modelo de casos de uso por separado para cada subsistema.

#### > Paso 4: Narraciones de los casos de uso para los requerimientos de documentos para los negocios

Cuando usted prepara las narraciones, es prudente documentarlas primero a un *alto nivel* para obtener rápidamente una comprensión de los eventos y de la magnitud del sistema. Entonces regrese a cada caso de uso y expándalo a una narración totalmente documentada de requerimientos de negocios. La figura 6.12 representa una narración de casos de uso de requerimientos para el caso de uso de Colocar una nueva orden en el Sistema de servicios

## Sistema de servicios para los miembros

Autor(es): \_\_\_\_\_ ①

Fecha: \_\_\_\_\_ ②

Versión: \_\_\_\_\_ ③

<b>Nombre del caso de uso:</b>	Colocar nueva orden ④	<b>Caso de uso del tipo de requerimientos de negocios:</b> <input checked="" type="checkbox"/> ⑤
<b>ID del caso de uso:</b>	MSS-BUC002.00 ⑥	
<b>Prioridad:</b>	Alta ⑦	
<b>Fuente:</b>	Requerimiento: MSS-R1.00 ⑧	
<b>Actor primario de negocios:</b>	Miembro del club ⑨	
<b>Otros actores participantes:</b>	<ul style="list-style-type: none"> <li>Almacén (receptor externo)</li> <li>Cuentas por cobrar (servidor externo) ⑩</li> </ul>	
<b>Otros involucrados interesados:</b>	<ul style="list-style-type: none"> <li>Mercadeo: interesados en las actividades de ventas con objeto de planear nuevas promociones.</li> <li>⑪ Suministro: interesados en las actividades de ventas con objeto de reponer el inventario.</li> <li>Administración: interesados en la actividad de las órdenes con objeto de evaluar el desempeño de la compañía y la satisfacción del cliente (miembro).</li> </ul>	
<b>Descripción:</b>	Este caso de uso describe el evento de un miembro del club que ingresa una nueva orden de productos de SoundStage. Una vez que se verifica que los productos están en existencia se manda al almacén una orden de embalaje para que se prepare el flete. Para cualquier producto que no esté en existencia, se crea una orden de devolución. Al completarse, al miembro se le manda una confirmación de la orden. ⑫	

**FIGURA 6.12** Narración del caso de uso Colocar una nueva orden en su versión de alto nivel

para los miembros. Observe que describe claramente el evento, que incluye los siguientes incisos:

- ① *Autor:* Los nombres de los individuos que contribuyeron a la escritura del caso de uso y que suministran un punto de contacto para cualquier persona que requiera información adicional acerca del caso de uso.
- ② *Fecha:* La fecha de la última modificación del caso de uso.
- ③ *Versión:* La versión actual del caso de uso (por ejemplo, 1.0).
- ④ *Nombre del caso de uso:* El nombre del caso de uso deberá representar la meta que el caso de uso está tratando de lograr. El nombre debe comenzar con un verbo (por ejemplo, Ingresar la Orden del nuevo miembro).
- ⑤ *Tipo de caso de uso:* Al realizar la modelación de los casos de uso se construyen primero los casos de uso de los requerimientos de negocios que se centren en la visión estratégica y en las metas de los diferentes involucrados. Este tipo de caso de uso está orientado hacia los negocios y refleja una vista de alto nivel del comportamiento deseado del sistema. Está libre de los detalles técnicos y puede incluir actividades manuales así como aquellas que se automatizarán. Los casos de uso de los requerimientos de negocios suministran una comprensión general del dominio y el alcance del problema pero no incluyen el detalle necesario para comunicar a los desarrolladores lo que el sistema debe hacer.
- ⑥ *ID del caso de uso:* Un identificador que distingue de manera única al caso de uso.
- ⑦ *Prioridad:* La prioridad comunica la importancia del caso de uso en términos de alta, mediana o baja.
- ⑧ *Fuente:* La fuente define a la entidad que originó la creación del caso de uso. Ésta podría ser un requerimiento, un documento específico, o un involucrado.
- ⑨ *Actor primario de negocios:* Es el involucrado que se beneficia en primer lugar del caso de uso al recibir algo de valor medible u observable.

- ⑩ *Otros actores participantes*: Otros actores que participan en el caso de uso para lograr su meta incluyen los actores que inician, los actores facilitadores, los actores servidores/receptores y los actores secundarios. Siempre incluya la manera en que el actor participa.
- ⑪ *Involucrado(s) interesado(s)*: Un involucrado es cualquier persona que tenga un aporte en el desarrollo y la operación del sistema de software. Un involucrado que tenga interés es una persona (diferente de un actor) que tiene un interés creado en la meta del caso de uso.
- ⑫ *Descripción*: Una descripción corta resumida que consiste en un par de oraciones que esbozan el propósito del caso de uso y sus actividades.

**Documentación del curso de los eventos del caso de uso** Para cada caso de uso de alto nivel ya identificado, ahora debemos expandirlo para incluir el curso típico de los eventos del caso de uso y los cursos alternos. El curso típico de los eventos del caso de uso es una descripción paso por paso que comienza con el actor que inicia el caso de uso y que continúa hasta el final del evento del negocio. En esta sección incluimos solamente los pasos principales que ocurren la mayoría de las veces (su curso típico). El curso alternativo documenta las excepciones o la bifurcación condicional del caso de uso. La figura 6.13 representa una narración de los requerimientos de un caso de uso para el caso de uso de Colocar una nueva orden del Sistema de Servicios para los Miembros. Observe que incluye los siguientes elementos adicionales:

- ① *Precondición*: Es una restricción del estado del sistema antes de la ejecución del caso de uso. En general, esto se refiere a otro caso de uso que debe ejecutarse previamente.
- ② *Disparador (generador)*: Es el evento que inició la ejecución del caso de uso. Frecuentemente, ésta es una acción física. Tal como un cliente que camina hasta el mostrador de ventas o un cheque que llega por el correo. El tiempo también puede disparar los casos de uso.
- ③ *Curso típico de los eventos*: Es la secuencia normal de las actividades realizadas por el(los) actor(es) y por el sistema con objeto de satisfacer la meta del caso de uso. Se incluyen las interacciones entre el sistema y el actor y las actividades realizadas por el sistema como respuesta a las interacciones. Observe que las acciones del actor se registran en la columna izquierda mientras que las acciones de los sistemas se registran en la columna derecha.
- ④ *Cursos alternos*: Los cursos alternos documentan los comportamientos del caso de uso si ocurre una excepción o una variación del curso típico. Esto puede suceder cuando ocurra un punto de decisión dentro del caso de uso o una excepción que requiera pasos adicionales fuera del alcance del curso típico.
- ⑤ *Conclusión*: Establece cuando termina con éxito el caso de uso; en otras palabras, cuando el actor primario recibe algo de valor medible.
- ⑥ *Postcondición*: Una postcondición es una restricción del estado del sistema después que el caso de uso ha sido ejecutado con éxito. Esto podría ser datos registrados en una base de datos o un recibo entregado a un cliente.
- ⑦ *Reglas de negocios*: Las reglas de negocios especifican políticas y procedimientos del negocio que el nuevo sistema debe obedecer. Esto podría incluir el cálculo de los cargos de embarque o las reglas para conceder plazos de crédito.
- ⑧ *Restricciones y especificaciones de la implantación*: Las restricciones y especificaciones de la implantación especifican requerimientos no funcionales que puedan impactar la realización del caso de uso y pueden ser de ayuda en cualquier planeación y alcance de la arquitectura. Los elementos que pueden incluirse son las especificaciones de seguridad, los requerimientos de la interfaz, etcétera.
- ⑨ *Hipótesis*: Cualesquiera hipótesis que formuló el creador al documentar el caso de uso.
- ⑩ *Aspectos abiertos*: Preguntas o aspectos que deben resolverse o investigarse antes de finalizar el caso de uso.



Sistema de servicios para los miembros

Autor(es): \_\_\_\_\_

Fecha: \_\_\_\_\_

Versión: \_\_\_\_\_

<b>Nombre del caso de uso:</b>	Colocar nueva orden	<b>Caso de uso del tipo de requerimientos de negocios:</b> <input checked="" type="checkbox"/>				
<b>ID del caso de uso:</b>	MSS-BUC002.00					
<b>Prioridad:</b>	Alta					
<b>Fuente:</b>	Requerimiento: MSS-R1.00					
<b>Actor primario de negocios:</b>	Miembro del club					
<b>Otros actores participantes:</b>	<ul style="list-style-type: none"> <li>Almacén (receptor externo)</li> <li>Cuentas por cobrar (servidor externo)</li> </ul>					
<b>Otros involucrados interesados:</b>	<ul style="list-style-type: none"> <li>Mercadeo: interesados en las actividades de ventas con objeto de planear nuevas promociones.</li> <li>Suministro: interesados en las actividades de ventas con objeto de reponer el inventario.</li> <li>Administración: interesados en la actividad de las órdenes con objeto de evaluar el desempeño de la compañía y la satisfacción del cliente (miembro).</li> </ul>					
<b>Descripción:</b>	Este caso de uso describe el evento de un miembro del club que ingresa una nueva orden de productos de SoundStage. Una vez que se verifica que los productos están en existencia, se manda al almacén una orden de embalaje para que se prepare el flete. Para cualquier producto que no esté en existencia se crea una orden de devolución. Al completarse, al miembro se le manda una confirmación de la orden.					
<b>Precondición:</b>	1 La parte (persona o compañía) que ingresa la orden debe ser miembro.					
<b>Ocasionador:</b>	2 Este caso de uso se inicia cuando se ingresa una nueva orden.					
<b>Curso típico de eventos:</b>	<table border="1"> <thead> <tr> <th>Acción del actor</th> <th>Respuesta del sistema</th> </tr> </thead> <tbody> <tr> <td>3 <b>Paso 1:</b> El miembro del club proporciona su información demográfica así como la información de las órdenes y de los pagos.</td> <td> <b>Paso 2:</b> El sistema responde verificando que se ha suministrado toda la información requerida.  <b>Paso 3:</b> El sistema verifica la información demográfica del miembro del club contra lo que se ha registrado anteriormente.  <b>Paso 4:</b> Para cada producto ordenado, el sistema valida la identidad del producto.  <b>Paso 5:</b> Para cada producto ordenado, el sistema verifica la disponibilidad del producto.  <b>Paso 6:</b> Para cada producto disponible, el sistema determina el precio que debe cobrarse al miembro del club.  <b>Paso 7:</b> Una vez que se procesan todos los productos ordenados, el sistema determina el costo total de la orden.  <b>Paso 8:</b> El sistema verifica el estado de la cuenta del miembro del club.  <b>Paso 9:</b> El sistema valida el pago del miembro del club si existe.  <b>Paso 10:</b> El sistema registra la información de la orden y luego libera la orden al centro de distribución apropiado (almacén) para llenarla.  <b>Paso 11:</b> Una vez que se procesa la orden, el sistema genera una confirmación de la orden y la manda al miembro del club.                 </td> </tr> </tbody> </table>	Acción del actor	Respuesta del sistema	3 <b>Paso 1:</b> El miembro del club proporciona su información demográfica así como la información de las órdenes y de los pagos.	<b>Paso 2:</b> El sistema responde verificando que se ha suministrado toda la información requerida. <b>Paso 3:</b> El sistema verifica la información demográfica del miembro del club contra lo que se ha registrado anteriormente. <b>Paso 4:</b> Para cada producto ordenado, el sistema valida la identidad del producto. <b>Paso 5:</b> Para cada producto ordenado, el sistema verifica la disponibilidad del producto. <b>Paso 6:</b> Para cada producto disponible, el sistema determina el precio que debe cobrarse al miembro del club. <b>Paso 7:</b> Una vez que se procesan todos los productos ordenados, el sistema determina el costo total de la orden. <b>Paso 8:</b> El sistema verifica el estado de la cuenta del miembro del club. <b>Paso 9:</b> El sistema valida el pago del miembro del club si existe. <b>Paso 10:</b> El sistema registra la información de la orden y luego libera la orden al centro de distribución apropiado (almacén) para llenarla. <b>Paso 11:</b> Una vez que se procesa la orden, el sistema genera una confirmación de la orden y la manda al miembro del club.	
Acción del actor	Respuesta del sistema					
3 <b>Paso 1:</b> El miembro del club proporciona su información demográfica así como la información de las órdenes y de los pagos.	<b>Paso 2:</b> El sistema responde verificando que se ha suministrado toda la información requerida. <b>Paso 3:</b> El sistema verifica la información demográfica del miembro del club contra lo que se ha registrado anteriormente. <b>Paso 4:</b> Para cada producto ordenado, el sistema valida la identidad del producto. <b>Paso 5:</b> Para cada producto ordenado, el sistema verifica la disponibilidad del producto. <b>Paso 6:</b> Para cada producto disponible, el sistema determina el precio que debe cobrarse al miembro del club. <b>Paso 7:</b> Una vez que se procesan todos los productos ordenados, el sistema determina el costo total de la orden. <b>Paso 8:</b> El sistema verifica el estado de la cuenta del miembro del club. <b>Paso 9:</b> El sistema valida el pago del miembro del club si existe. <b>Paso 10:</b> El sistema registra la información de la orden y luego libera la orden al centro de distribución apropiado (almacén) para llenarla. <b>Paso 11:</b> Una vez que se procesa la orden, el sistema genera una confirmación de la orden y la manda al miembro del club.					

**FIGURA 6.13** Versión expandida de la narración del caso de uso de Colocar una nueva orden

Los casos de uso de los requerimientos de negocios son herramientas excelentes porque describen los eventos que la organización debe procesar y responderles, pero les falta información con respecto a las interfaces y las actividades que están programadas para automatizarse por la tecnología de información. Posteriormente, en el capítulo 9, usted aprenderá cómo hacer evolucionar el caso de uso para incluir los detalles técnicos y de implantación.

<b>Cursos alternos:</b>	4	<p><b>Paso alternativo 2:</b> El miembro del club no ha suministrado toda la información necesaria para procesar la orden. Se notifica la discrepancia al miembro del club y se le urge a que vuelva a presentar la solicitud.</p> <p><b>Paso alternativo 3:</b> Si la información suministrada del miembro del club es diferente de lo que se registró anteriormente, verifique lo que está registrado actualmente, y luego actualice de acuerdo con esto la información del miembro del club.</p> <p><b>Paso alternativo 4:</b> Si la información de producto que suministró el miembro del club no concuerda con ninguno de los productos de SoundStage, notifique la discrepancia al miembro del club y solicite una aclaración.</p> <p><b>Paso alternativo 5:</b> Si no está disponible la cantidad ordenada del producto, se crea una orden de devolución.</p> <p><b>Paso alternativo 8:</b> Si el estado de la cuenta del miembro del club es que no tiene derechos, registre la información de la orden y póngala en estado de espera. Notifique el estado de la cuenta al miembro del club y la razón por la cual la orden está detenida. Finiquite el caso de uso.</p> <p><b>Paso alternativo 9:</b> Si el pago provisto por el miembro del club (tarjeta de crédito) no puede validarse, notifique al miembro del club y solicite un medio alterno de pago. Si el miembro del club no puede suministrar un medio alterno, cancele la orden y finiquite el caso de uso.</p>
<b>Conclusión:</b>	5	Este caso de uso concluye cuando el miembro del club recibe una confirmación de la orden.
<b>Postcondición:</b>	6	La orden ha sido registrada y si estaban disponibles los productos ordenados, éstos fueron liberados. Para cualquier producto no disponible se ha creado una orden de devolución.
<b>Reglas de negocios:</b>	7	<ul style="list-style-type: none"> <li>El miembro del club que responde a una promoción o un miembro que ejerce un crédito puede afectar el precio de cada artículo ordenado.</li> <li>Con las órdenes no se acepta efectivo ni cheques. Si llegan, serán regresados al miembro del club.</li> <li>Los productos se facturan al miembro del club solamente cuando han sido fletados.</li> </ul>
<b>Restricciones y especificaciones de implantación:</b>	8	<ul style="list-style-type: none"> <li>Debe suministrarse un GUI al socio de los Servicios para los miembros, y al miembro del club debe suministrarse una pantalla de la Red.</li> </ul>
<b>Hipótesis:</b>	9	La procuración de las órdenes de devolución será notificada mediante un reporte diario (caso de uso por separado).
<b>Aspectos abiertos:</b>	10	1. Necesidad de determinar cómo se asignan los centros de distribución.

**FIGURA 6.13** Conclusión

## Los casos de uso y la administración de proyectos

Como usted recordará, uno de los beneficios de la modelación de los casos de uso es que el modelo del caso de uso puede usarse para impulsar el esfuerzo completo de desarrollo de sistemas. Una vez que esté completo el modelo de caso de uso de los requerimientos de negocios, el administrador de proyectos o el analista de sistemas usa los casos de uso para los requerimientos de negocios para planear (estimar y programar) los ciclos de construcción del proyecto. Esto es especialmente crucial al aplicar el enfoque iterativo y de incrementos al desarrollo de software. A un ciclo de construcción, que consiste en las actividades del análisis, el diseño y la construcción de sistemas, se le asigna su alcance basándose en la importancia del caso de uso y en el tiempo que toma implantar el caso de uso. En otras palabras, se desarrollarán uno o más casos de uso para cada ciclo de construcción. Cuando un caso de uso es demasiado grande o complejo para terminarse en un ciclo de construcción, entonces inicialmente se implantará una versión simplificada, seguida por la versión completa en el siguiente ciclo de construcción. Para determinar la importancia de los casos de uso, el administrador de proyecto o el analista de sistemas terminarán la jerarquización de los casos de uso y la matriz de evaluación y construirá un diagrama de dependencia de casos de uso con entradas de los involucrados y del equipo de desarrollo. Usted aprenderá como usar estas herramientas en las siguientes secciones.

**matriz de jerarquía y prioridad de los casos de uso** Herramienta usada para evaluar los casos de uso y determinar su importancia.

### > Cómo jerarquizar y evaluar los casos de uso

En la mayoría de los proyectos de desarrollo de software, los casos de uso más importantes se desarrollan primero. Con objeto de determinar la prioridad de los casos de uso, el administrador de proyectos usa una herramienta llamada **matriz de jerarquía y priori-**

Nombre del caso de uso	Criterios de jerarquización, 1 a 5						Calificación total	Prioridad	Ciclo de construcción
	1	2	3	4	5	6			
Ingresar la orden de suscripción	5	5	5	4	5	5	29	Alto	1
Colocar la nueva orden	4	4	5	4	5	5	27	Alto	2
Hacer investigación de producto	1	1	1	1	1	1	6	Bajo	3
Establecer un programa de suscripción de nuevos miembros	4	5	5	3	5	5	27	Alto	1
Generar un reporte de conformidad por omisión diario de 10-30-60 días	1	1	1	1	1	1	6	Bajo	3
Revisar la orden	2	2	3	3	4	4	18	Medio	2

**FIGURA 6.14** Jerarquización y matriz de prioridad parciales de un caso de uso

**dad de los casos de uso.** Esta matriz se llena con las entradas de los involucrados y del equipo de desarrollo. Esta matriz, adaptada del trabajo de Craig Larman,<sup>3</sup> evalúa los casos de uso en una escala de 1 a 5 con respecto a seis criterios. Éstos son los siguientes:

1. Impacto significativo sobre el diseño de la arquitectura.
2. Fácil de implantar pero contiene una funcionalidad significativa.
3. Incluye funciones riesgosas, críticas con respecto al tiempo, o que son complejas.
4. Implica mucha investigación o tecnología nueva o de alto riesgo.
5. Incluye funciones primarias de negocios.
6. Aumentará las utilidades o disminuirá los costos.

Una vez que cada categoría ha sido calificada se anotan las calificaciones individuales, lo que conduce a la calificación final del caso de uso. A los casos de uso con las calificaciones más altas se les asignan la prioridad más alta y deben desarrollarse primero.

La figura 6.14 es una matriz parcial de prioridades y una jerarquización parcial de los casos de uso para el Sistema de servicios a los miembros. Basándose en los resultados del análisis, el caso de uso Ingresar la orden de suscripción debe desarrollarse primero. Pero no podemos estar seguros hasta que analicemos las dependencias del caso de uso.

### > Identificación de las dependencias de los casos de uso

Algunos casos de uso pueden depender de otros, con un caso de uso que dejando al sistema en un estado que es una precondition para otro caso de uso. Por ejemplo, una precondition para mandar una promoción del club es que la promoción debe crearse primero. Usamos un diagrama llamado el **diagrama de dependencia de casos de uso** para modelar estas dependencias. El diagrama de dependencia de casos de uso proporciona los siguientes beneficios:

- La ilustración gráfica de los eventos del sistema y de sus estados aumenta la comprensión de la funcionalidad del sistema.
- Ayuda a identificar los casos de uso faltantes. Un caso de uso con una precondition que no se satisfaga con la ejecución de cualquier otro caso de uso puede indicar un caso de uso faltante.
- Ayuda a facilitar la administración del proyecto al ilustrar cuáles son los casos de uso más críticos (que tienen el mayor número de dependencias) y entonces deben tener una prioridad más alta.

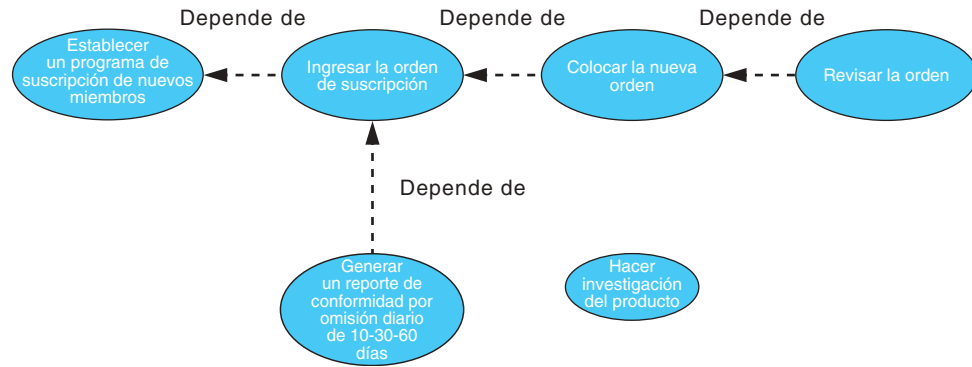
La figura 6.15 es el diagrama de dependencia de casos de uso para aquéllos listados en la figura 6.14. Los casos de uso que son dependientes entre sí se conectan con una

**diagrama de dependencia de casos de uso** Representación gráfica de las dependencias entre casos de uso.

<sup>3</sup> Craig Larman, *Applying UML Patterns* (Upper Saddle River, NJ: Prentice Hall, 1998).

**FIGURA 6.15**

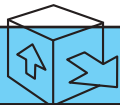
Ejemplo de un diagrama de dependencia de un caso de uso



línea segmentada rotulada “Depende de”. En la figura 6.15, el caso de uso Ingresar la orden de suscripción tiene una dependencia (precondición) con respecto al caso de uso Establecer un nuevo programa de suscripción de miembros. Debido a esta dependencia, el caso de uso Establecer un nuevo programa de suscripción de miembros debe desarrollarse primero aun cuando Ingresar la orden de suscripción tenga una calificación más alta como se refleja en la figura 6.14.

Este capítulo presentó una introducción a los casos de uso y cómo pueden usarse para documentar los requerimientos funcionales. Usted también ha aprendido que la modelación de casos de uso basada en los conceptos orientados a objetos es una herramienta complementaria excelente para las herramientas tradicionales del análisis y diseño de sistemas tales como la modelación de los procesos y la modelación de los datos. Muchos de ustedes proseguirán directamente al capítulo 7, “Modelado y análisis de datos”. Todos los sistemas de información incluyen bases de datos, y la modelación de datos es una habilidad esencial para el desarrollo de las bases de datos. También es más fácil sincronizar los modelos de datos con los modelos de procesos que al revés. Su profesor podría preferir que usted estudie primero el capítulo 8, “Modelado de procesos”. La modelación de procesos es una forma efectiva de analizar y documentar los requerimientos funcionales del sistema. En los cursos que siguen un enfoque orientado a objetos se recomienda revisar algún libro sobre análisis y modelación orientados a objetos con el uso de lenguaje unificado de modelación (UML).

## Resumen



1. Hay dos elementos primarios que participan al realizar la modelación de los casos de uso. El primero es el diagrama de caso de uso, que ilustra gráficamente al sistema como una colección de casos de uso, actores (usuarios) y sus relaciones. Los detalles de cada evento de negocios y de cómo interactúan los usuarios con el sistema se describen en el segundo elemento, llamado la narración del caso de uso, que es la descripción textual del evento de

negocios y cómo va a interactuar el usuario con el sistema para lograr la tarea.

2. La modelación de casos de uso utiliza dos construcciones: los actores y los casos de uso. Un actor representa cualquier cosa que necesite interactuar con el sistema para intercambiar información. Un actor es un usuario, un papel, que podría ser un sistema externo así como una persona. Un caso de uso es una secuencia de pasos relacionados

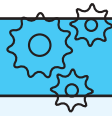
- (un escenario) tanto automatizada como manual, con el propósito de completar una tarea individual del negocio.
3. Hay principalmente cuatro tipos de actores:
    - a) *Actor primario de negocios*: El involucrado que se beneficia en primer lugar de la ejecución del caso de uso al recibir algo de valor medible u observable.
    - b) *Actor primario del sistema*: El involucrado que sostiene una interfaz directa con el sistema para iniciar u originar el evento de negocios o del sistema.
    - c) *Actor externo proveedor*: El involucrado que responde a una solicitud proveniente del caso de uso.
    - d) *Actor externo receptor*: El involucrado que no es el actor primario pero que recibe algo de valor medible u observable (salida) proveniente del caso de uso.
  4. Los eventos temporales son eventos de negocios que se realizan (o se originan) en forma automática; cuando se cumple cierta fecha o lapso. Debido a eso, decimos que el actor de un evento temporal es el tiempo.
  5. Una relación se ilustra como una línea entre dos símbolos en el diagrama de casos de uso.
    - a) Una asociación es una relación entre un actor y un caso de uso.
    - b) La relación entre el caso de uso de extensión y el caso de uso que se está extendiendo se llama una relación de extensión.
    - c) La relación entre el caso de uso resumen y el caso de uso que los usa se llama una relación de usos.
    - d) La relación de herencia ocurre cuando un actor hereda la capacidad para iniciar un caso de uso de otro.
    - e) La relación de dependencia indica una dependencia entre los casos de uso. En otras palabras, la precondition de un caso de uso depende de la postcondición de otro caso de uso.
  6. Los pasos requeridos para producir un modelo de casos de uso de requerimientos son los siguientes:
    - a) Identificar a los actores de negocios.
    - b) Identificar los casos de uso de requerimientos de negocios.
    - c) Construir el diagrama del modelo de casos de uso.
    - d) Documentar las narraciones de los casos de uso para los requerimientos de negocios.
  7. La matriz de prioridad y la jerarquización del caso de uso son las herramientas que usan los administradores de proyecto para priorizar y programar el desarrollo del caso de uso.



## Preguntas de repaso

1. ¿Qué es el desarrollo centrado en el usuario y por qué es crítico para el éxito del proceso de desarrollo de sistema?
2. ¿Cómo se relaciona la modelación de los casos de uso con el desarrollo centrado en el usuario?
3. Además de alentar la participación del usuario, la modelación de los casos de uso proporciona muchos otros beneficios; lístelos.
4. El modelado de caso de uso emplea dos elementos primarios: el diagrama de caso de uso y la narración de caso de uso. ¿Cómo se utilizan estos dos elementos y cuáles son sus diferencias?
5. Los diagramas de caso de uso constan de tres componentes. ¿Cuáles son estos tres componentes y cuál es su propósito?
6. ¿Cómo se usan los casos de uso durante el ciclo de vida completo del desarrollo del sistema?
7. De las cuatro categorías primarias de actores, ¿quién es el actor primario del sistema?
8. ¿Cuáles son los tipos diferentes de relaciones empleadas en un diagrama de caso de uso y cuál es su propósito?
9. ¿Cuál es el objetivo de construir el modelo de casos de uso de los requerimientos y qué pasos deben seguirse?
10. ¿Por qué es la identificación de los actores el primer paso en la modelación de los casos de uso?
11. ¿De qué deberíamos darnos cuenta cuando buscamos casos de uso de requerimientos de negocios?
12. ¿Cuál es el curso de eventos típico de un caso de uso?
13. ¿Por qué son esenciales la jerarquización y la evaluación de los casos de uso?
14. ¿Cuáles son los seis criterios en la jerarquización y en la matriz prioritaria?
15. ¿Cuál es el diagrama de dependencia de los casos de uso y por qué lo usamos?

# Problemas y ejercicios



1. De acuerdo con el autor Fred Brooks, ¿qué es lo más difícil de hacer en el desarrollo de sistemas? ¿Cómo ayuda la modelación de casos de uso en esta área?
2. En la modelación de casos de uso, ¿cuáles son los dos elementos principales que usa el analista de sistemas? Describa cada uno de estos elementos y explique su propósito.
3. ¿Qué debe tener siempre en mente un analista de sistemas al identificar y desarrollar los casos de uso en relación con su propósito? Como la exploración de requerimientos se completó previamente, ¿es en realidad necesario pasar mucho tiempo con los usuarios en este punto? ¿Simplemente qué debería representar un caso de uso? ¿Un caso de uso equivale a un requerimiento funcional?
4. ¿Durante qué parte del desarrollo del ciclo de vida de se definen por primera vez los casos de uso? ¿Cuándo se usan durante el ciclo de vida de desarrollo, y para qué propósito?
5. Relacione a los siguientes involucrados y usuarios externos con el actor correcto. ¿Qué es un evento temporal? ¿Quién o qué se considera como el actor en un evento temporal, y por qué?

## Los involucrados y los usuarios externos

- Servicio Postal de Estados Unidos
- Cerrojo con teclado
- Agente de autos de alquiler
- Gerente de ventas que genera el informe regional de ventas
- Administrador de ventas que recibe el informe regional de ventas
- Sistema automático de aspersor giratorio para regar el césped
- El conductor que compra gasolina con tarjeta de cajero automático
- El servicio de la autorización de préstamo bancario

## Actores

- Actor primario de negocios
- Actor primario del sistema
- Actor externo proveedor
- Actor externo receptor
- El tiempo

6. ¿Cuál es el tipo de relación para cada uno de los siguientes ejemplos?
  - La relación entre el caso de uso “Imprimir la forma” y varios otros casos de uso que implican la impresión de tipos diferentes de formas.
  - La relación entre un oficial de motocicleta y un dispositivo de mano para escribir citatorios.
  - La relación entre un cliente y un dependiente vendedor de una tienda que pueden interrogar

al sistema de inventario para ver si un artículo existe en inventario, y un actor creado específicamente para minimizar la comunicación duplicada de sistema.

- La relación entre el caso de uso “Calcular el promedio general” y el caso de uso prolongado “Crear un certificado de estudios”.
  - La relación entre el caso de uso “Enviar orden” y el caso de uso “Colocar una orden”.
7. Y&J Cookbooks es un negocio pequeño ficticio poseído y manejado por una pareja jubilada. Hasta este momento, Y&J Cookbooks han vendido sus libros sólo a través de pedidos por correo. Los dueños ahora quieren desarrollar un sistema en línea para vender libros de cocina raros y agotados a través de Internet. Los visitantes podrán hojear diferentes libros de cocina, pero tendrán que crear una cuenta del cliente antes de poder hacer una compra. Los pagos se aceptarán sólo en línea con una tarjeta de crédito reconocida, y se verificará la tarjeta de crédito antes de que la orden pueda ser aprobada. Basado en esta información, identifique a los actores principales del negocio.
  8. En la modelación de casos de uso, una vez que usted identifica a los actores de negocios, ¿qué perspectiva y qué lenguaje debería usar usted para definirlos? Use esa perspectiva y ese lenguaje para construir un glosario de actores usando la figura 6.8 como ejemplo.
  9. Un diagrama de contexto puede ayudar muchísimo para identificar los diferentes casos de uso. Prepare un diagrama de contexto de alto nivel para el sitio Web de Y&J Cookbooks, usando la figura 6.9 como ejemplo.
  10. El siguiente paso en la modelación de casos de uso para requerimientos es identificar los casos de uso de requerimientos de negocios. ¿Qué debería capturar cada caso de uso? ¿Qué técnica efectiva puede usted usar para identificar los casos de uso? ¿Qué preguntas puede hacer usted para identificar los casos de uso? ¿Cuál es la diferencia entre un caso de uso y un caso esencial de uso?
  11. El tercer paso en la modelación de casos de uso es construir los diagramas de modelo de caso de uso. Basado en el glosario de actores Y&J Cookbooks y en el diagrama de contexto, cree un diagrama de modelo de caso de uso de alto nivel, mostrando las interacciones entre el actor comprador/cliente y el sistema, incluyendo la búsqueda y el hojeador de libros, la compra y la operación de la cuenta del cliente. Asegúrese de demostrar las relaciones entre el actor y cada uno de estos casos de uso.
  12. El siguiente paso es crear narraciones de caso de uso para documentar los requerimientos de negocios. ¿Por qué la preparación de las narraciones generalmente se hace en un proceso de dos pasos y

cuáles son estos dos pasos? Basado en el diagrama de modelo de casos de uso de alto nivel precedente, cree una narración expandida, usando la figura 6.13 como ejemplo.

13. ¿Cuál es la relación de la modelación de los casos de uso con la gerencia del proyecto? ¿Por qué es importante? ¿Por qué se jerarquizan los casos y qué

herramienta se usa para jerarquizarlos? ¿Quién proporciona la entrada para jerarquizarlos? ¿Qué criterios se usan para la jerarquización? Explique por qué es necesario identificar las dependencias de los casos de uso y suministre un ejemplo. ¿Qué herramienta se usa para identificar las dependencias?



## Proyectos e investigación

1. Al principio del capítulo 6 hay una cita tomada de un artículo por Frederick P. Brooks Jr., quien es generalmente considerado como uno de los contribuyentes y autores más destacados del campo del desarrollo de administración de proyecto y de software. Navegue en la Web para encontrar este artículo y otros artículos de Fred Brooks, y acerca de él.
  - a) Al hacer su búsqueda, ¿cuántas referencias encontró usted acerca del autor?
  - b) Basado en la información presentada en los capítulos previos, explique la declaración de Brook que “la parte más difícil en la construcción de un sistema de software es decidir precisamente qué construir”.
  - c) ¿Cómo se llama el artículo en el cual Brooks hizo la declaración precedente, y cuál fue el tema principal del artículo?
  - d) ¿Cuál es el libro de Brook mejor conocido que está todavía en prensa y ampliamente leído décadas después de su publicación original? ¿Cuál fue el tema principal de este libro?
  - e) ¿Cuál considera usted que es la máxima contribución de Brooks hasta la fecha? ¿Por qué?
2. El Standish Group, que se mencionó en el capítulo 6, es un grupo de investigación independiente que estudia cambios y tendencias en la tecnología de la información. En 1994, el Standish Group publicó su Reporte CHAOS de avanzada, el cual “expone [expuso] el fracaso abrumador de proyectos de desarrollo de aplicaciones de tecnología de la información en el ambiente actual de sistemas de información para la administración”. Desde entonces, el Standish Group ha publicado actualizaciones periódicas de su informe original. Visite su sitio Web en [www.standishgroup.com](http://www.standishgroup.com) y siga los enlaces en su área de acceso al público, donde usted puede encontrar un resumen de su último informe de investigación de CHAOS, así como también el informe original de 1994.
  - a) ¿Qué criterios usa el Standish Group para determinar si un proyecto tuvo éxito, fue cuestionado, o fracasó?
  - b) ¿Basado en el último informe de investigación, cuál fue el porcentaje de proyectos que tuvieron éxito, fueron cuestionados, o fracasaron?
  - c) ¿Cómo se comparan estas últimas tasas con el éxito, el cuestionamiento y las tasas de fracaso de los proyectos mostradas en la figura 6.1 del libro de texto? ¿Cómo describiría usted las tendencias generales, si es que existen?
  - d) ¿Basado en su lectura y su experiencia, cuál(es) piensa usted que es(son) la(s) razón(es) de los cambios en las tasas de éxito, cuestionamiento, y fracasos de los proyectos?
  - e) Usted piensa que las metodologías actuales de administración de proyectos y de desarrollo de sistemas permanecerán esencialmente iguales pero seguirán siendo afinadas, ¿o avizora el surgimiento de metodologías radicalmente diferentes para administrar proyectos y sistemas de desarrollo en la siguiente década?
3. Seleccione un sistema de información usado en su organización o en su escuela. Entreviste a un analista de sistemas o un diseñador que esté familiarizado con el sistema. Basado en la información prevista, haga lo siguiente:
  - a) Describa el sistema de información y de organización que usted seleccionó.
  - b) Realice un diagrama de contexto del sistema.
  - c) Identifique a los actores de negocios.
  - d) Escriba un glosario de actores.
  - e) Identifique los casos de uso esenciales de requerimientos de negocios.
  - f) Escriba un glosario de casos de uso.
4. Basado en la información prevista referente al sistema que usted seleccionó en la pregunta precedente:
  - a) Construya un diagrama de modelo de caso de uso que incluya todos los subsistemas principales.
  - b) Prepare narraciones expandidas de caso de uso para tres de los casos de uso esenciales.
  - c) Prepare la jerarquización y la matriz prioritaria del caso de uso, luego úsela para jerarquizar y evaluar los casos de uso.
  - d) Identifique las dependencias del caso de uso.
  - e) Prepare un diagrama de dependencia del caso de uso.

5. Navegue en la Web o busque en las publicaciones profesionales en su biblioteca de la escuela, artículos de investigación sobre los desarrollos nuevos y emergentes en la modelación de casos de uso. Seleccione dos artículos, luego haga lo siguiente:
  - a) Proporcione la referencia bibliográfica de cada artículo. (Emplee el formato usado por su escuela.)
  - b) Elabore un resumen con sus propias palabras para cada artículo.
  - c) Compare y contraste las metodologías descritas en cada artículo. Describa cuál piensa usted que es más viable, significativo o ambos, y explique por qué.
6. Realice entrevistas con varios desarrolladores con respecto a sus puntos de vista sobre la modelación de los casos de uso. Si es posible, trate de encontrar desarrolladores de organizaciones diferentes o que tengan grados diferentes de experiencia o ambas cosas, así como también tipos diferentes de experiencia (es decir, un desarrollador que tenga experiencia principalmente como analista de sistemas, uno con experiencia como diseñador, y otro como constructor).
  - a) Describa a los desarrolladores que usted entrevistó en términos de su experiencia. Por ejemplo, ¿cuánto tiempo han trabajado en la tecnología de la información, cuál es su área de especialización, y qué tan familiarizados están con la modelación de casos de uso?
  - b) ¿Cuál es la naturaleza de su(s) organización(es)?
  - c) ¿Qué preguntas hizo usted?
  - d) ¿Cuáles son los puntos de vista de cada desarrollador con respecto a la importancia y el valor de la modelación de los casos de uso?
  - e) ¿Utilizan estos desarrolladores realmente la modelación de casos de uso en su organización actual? ¿Por qué sí o por qué no?
  - f) Si ellos fueran el CIO de su organización por un día, ¿cambiarían la arquitectura de tecnología de la información de su organización con respecto a la modelación de los casos de uso? Si es así, ¿cómo?
  - g) ¿Usando el modelo de madurez de capacidad, cómo evaluaría usted el nivel de madurez de su organización? ¿Por qué?
  - h) ¿Qué conclusiones (si las hay) puede obtener usted de las entrevistas estimando la aplicación práctica de la modelación de los casos de uso?
  - i) ¿Cuáles fueron sus puntos de vista con respecto a la importancia y el valor de la modelación de casos de uso antes de las entrevistas? ¿Cambiaron sus puntos de vista como resultado de las entrevistas? Si es así, ¿cómo cambiaron y por qué?

## Casos breves



1. En un caso breve del capítulo 5, usted entrevistó a los involucrados con respecto a un sistema de inscripciones en línea. En ese ejercicio, usted tenía que comprender los asuntos y necesidades que los involucrados tuvieran en cuanto al sistema. Revise sus conclusiones de esas entrevistas.
  - a) Revise otros sistemas de inscripción escolar. ¿Hay algún aspecto que usted piense que sería del agrado o desagrado de los involucrados? Tome notas y esboce ejemplos de pantallas de otros sistemas.
  - b) Realice una entrevista de seguimiento con los involucrados con quienes usted previamente habló y determine la funcionalidad específica y los requerimientos de facilidad de uso para su escuela.
2. Elabore una descripción de caso de uso para al menos uno de los requerimientos de funcionalidad que usted encontró en el problema previo. Siga el ejemplo mostrado en la figura 6.10.
3. Identifique a todos los actores del sistema de inscripción de la escuela. ¿Cuáles son los casos de uso que iniciará cada uno de ellos?
4. Usando su respuesta para el problema previo, dibuje un diagrama de caso de uso del sistema de inscripción de la escuela.

## Ejercicios de equipo e individuales



1. Mesa redonda: ¿Piensa usted que las personas son siempre absolutamente veraces en sus respuestas a las preguntas de las entrevistas?
2. Observe una película muda. Mesa redonda: ¿Cuál es la comunicación que tiene lugar además de las palabras?
3. Mesa redonda: Cuando usted determina los requerimientos de un sistema a través de un método tal como una entrevista, supone que la persona a quien entrevista y de quien recopila información, quiere que el sistema tenga éxito. ¿Es éste siempre el caso? ¿Cómo puede manejar usted la recopilación de requerimientos si no es éste el caso?



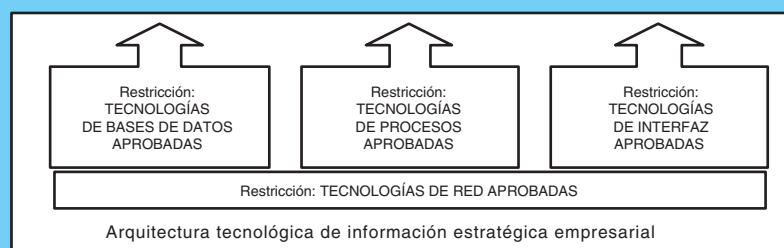
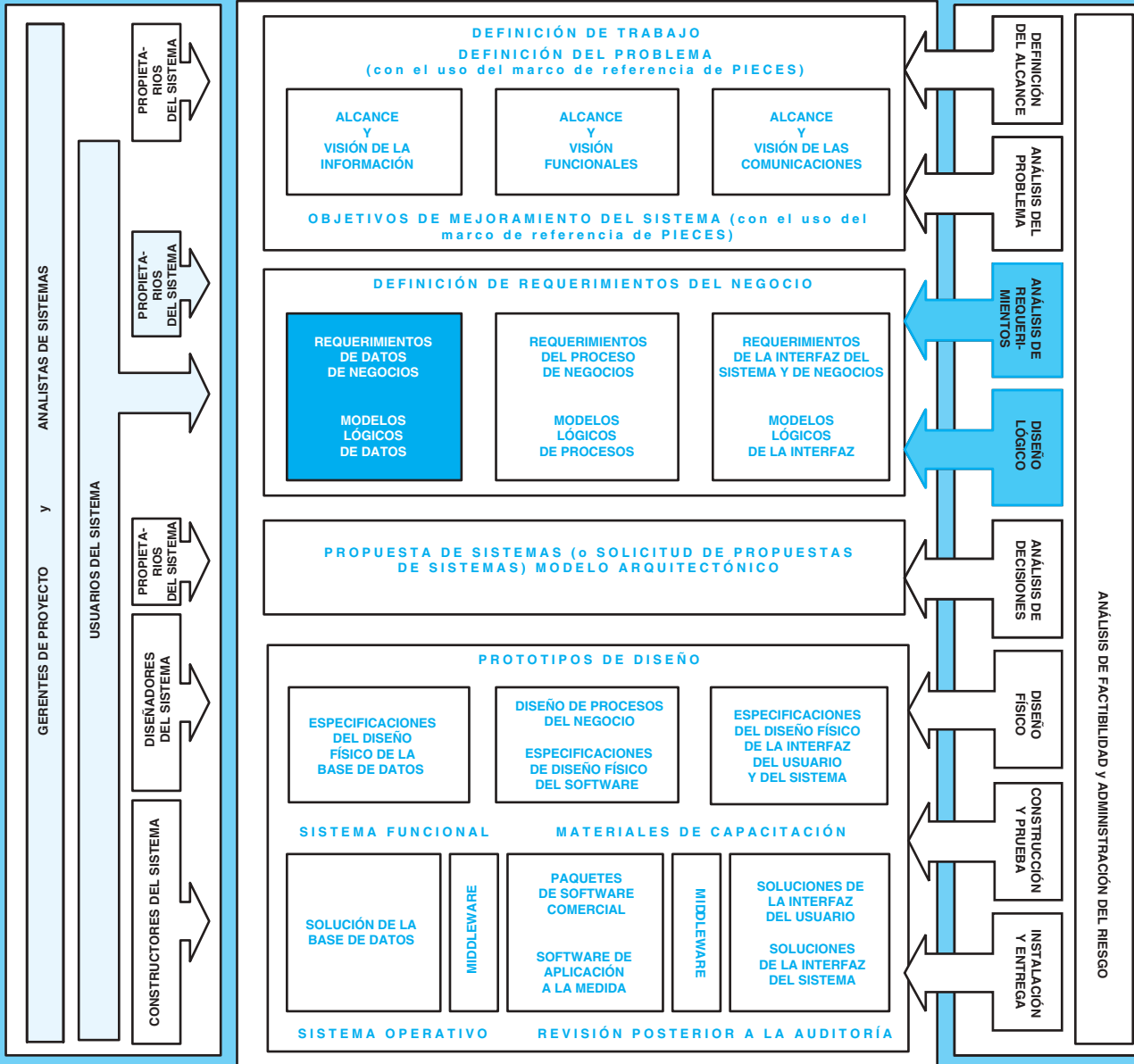
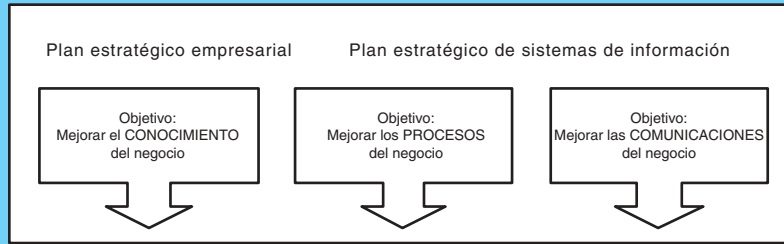


## Lecturas recomendadas

- Ambler, Scott W. *The Object Primer*. Nueva York: Cambridge University Press, 2001. Información muy buena acerca de la documentación de los casos de uso y su uso.
- Armour, Frank y Granville Miller. *Advance Use Case Modeling*. Boston: Addison-Wesley, 2001. Este libro presenta una excelente cobertura del proceso de la modelación de los casos de uso.
- Brooks, Jr., F.P., 1987. "No SilverBullet-Essence and Accidents of Software Engineering." *Computer* 20(4), 10-19 de abril. *proc. IFIP Congress*, Dublín, Irlanda, 1986.

- Jacobson, Ivar; Magnus Christerson; Patrik Jonsson; y Gunnar Overgaard. *Object-Oriented Software Engineering—A Use Case Driven Approach*. Workingham, Inglaterra: Addison-Wesley, 1992. Este libro presenta una cobertura detallada de cómo identificar y documentar los casos de uso.
- Larman, Craig. *Applying UML and Patterns*. Upper Saddle River, NJ: Prentice Hall, 1998. Este libro proporciona una vista panorámica comprensiva del proceso de la modelación de los casos de uso.





# 7

## Modelado y análisis de datos

### Panorámica y objetivos del capítulo

En este capítulo usted aprenderá cómo usar una popular herramienta de modelado de datos, los *diagramas de entidad relación*, para documentar los datos que deben capturarse y almacenarse en el sistema, independientemente de mostrar cómo se usan o se usarán los datos (es decir, muy aparte de las entradas, salidas y procesamientos específicos). También aprenderá acerca de una técnica de análisis de datos llamada *normalización* que se usa para asegurar que un modelo de datos es un “buen” modelo de datos. Usted sabrá que entiende el modelado de datos y los análisis de datos como herramientas y técnicas de análisis de sistemas cuando usted pueda:

- Definir el modelado de sistemas y diferenciar entre los modelos de sistemas lógicos y físicos.
- Definir el modelado de datos y explicar sus beneficios.
- Reconocer y entender los conceptos básicos y las construcciones de un modelo de datos.
- Leer e interpretar un modelo de datos de entidad relación.
- Explicar cuándo se construyen los modelos de datos durante un proyecto y dónde se almacenan los modelos.
- Descubrir las entidades y las relaciones.
- Construir un diagrama de contexto de entidad relación.
- Descubrir o inventar claves para las entidades y construir un diagrama basado en claves.
- Construir un diagrama de entidad relación con atribuciones completas y describir todas las estructuras de datos y atributos para el destinatario o enciclopedia.
- Normalizar un modelo de datos lógicos para retirar las impurezas que puedan hacer a una base de datos inestable, inflexible y no escalable.
- Describir una herramienta útil para el mapeo de requerimientos de datos en las localidades que opera el negocio.

## Introducción

A medida que el proyecto del sistema de servicios para los miembros SoundStage avanza del análisis de requerimientos al diseño lógico, la primera tarea, según su metodología, es analizar los requerimientos de datos para el nuevo sistema. Bob Martínez recuerda a su profesor favorito en la universidad que siempre dijo: “Obtengan los datos correctamente y el sistema podrá soportar todos los requerimientos presentes, incluso los requerimientos que los usuarios aún no visualizan; obtengan los datos en forma equivocada y será un lío tremendo cumplir con los requerimientos hoy, mañana y siempre.”

Bob disfrutó sus clases de bases de datos en la universidad y siempre obtuvo buenas calificaciones. Por supuesto, el sistema de servicios para los miembros es mayor y más detallado que cualquier proyecto de datos que él haya hecho en la escuela. Afortunadamente, él tiene la base de datos de la versión previa del sistema para comenzar, más las formas y los reportes del sistema previo, más las notas de las entrevistas del usuario, más las narrativas de casos de uso creadas durante la fase de análisis de requerimientos. Sandra le ha pedido a Bob que realice el primer intento de concentrar todo en un modelo lógico de datos. Él tiene toda la intención de impresionarla.

## ¿Qué es el modelado de datos?

**modelado de datos** Técnica para organizar y documentar los datos de un sistema. Algunas veces llamada *modelado de bases de datos*.

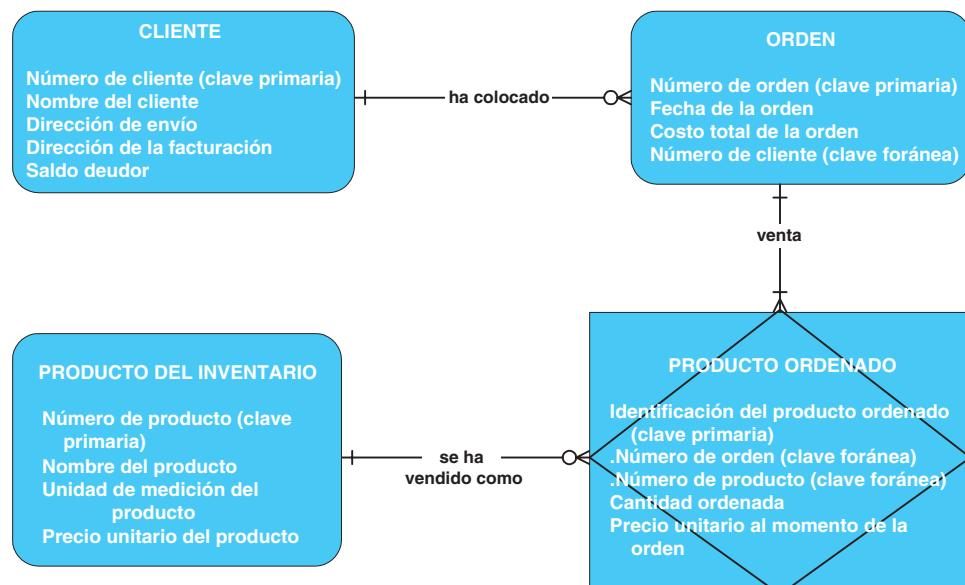
Los modelos de sistemas juegan un papel importante en el desarrollo de sistemas. Este capítulo presentará el **modelado de datos** como una técnica para definir los requerimientos del negocio de una base de datos. Algunas veces, al modelado de datos se le llama *modelado de base de datos* porque un modelo de datos a la larga se implementa como una base de datos.

La figura 7.1 es un ejemplo de un modelo simple de datos llamado *diagrama de entidad-relación*, o *ERD*. Este diagrama hace las siguientes aseveraciones de negocios:

- Necesitamos almacenar datos acerca de **CLIENTES**, **ÓRDENES** y **PRODUCTOS DE INVENTARIO**.
- Cada valor de **NÚMERO DE CLIENTE** identifica únicamente a un (y sólo un) **CLIENTE**. El valor del **NÚMERO DE ORDEN** identifica únicamente a una **ORDEN**. El valor del **NÚMERO DE PRODUCTO** identifica únicamente a un **PRODUCTO DE INVENTARIO**.
- Para cada **CLIENTE** necesitamos saber el **NOMBRE DEL CLIENTE**, la **DIRECCIÓN DE ENVÍO**, la **DIRECCIÓN PARA EL COBRO** y el **SALDO DEUDOR**. Para una **ORDEN** necesitamos saber la **FECHA DE LA ORDEN** y el **COSTO TOTAL DE LA ORDEN**. Para un **PRODUCTO DE INVENTARIO** necesitamos saber el **NOMBRE DEL PRODUCTO**, la **UNIDAD DE MEDICIÓN DEL PRODUCTO** y el **PRECIO UNITARIO DEL PRODUCTO**.

**FIGURA 7.1**

Un modelo de datos de entidad-relación



- Un CLIENTE ha colocado cero, una o más ÓRDENES actuales o recientes.
- Una ORDEN es colocada por exactamente un CLIENTE. El valor NÚMERO DE CLIENTE (como se registra en la ORDEN) identifica a ese CLIENTE.
- Una ORDEN vendió uno o más PRODUCTOS ORDENADOS. Así, una ORDEN debe contener al menos un PRODUCTO ORDENADO.
- Un PRODUCTO DE INVENTARIO pudo haber sido vendido como cero, uno o más PRODUCTOS ORDENADOS.
- Un PRODUCTO ORDENADO identifica a un solo PRODUCTO DE INVENTARIO en una sola ORDEN. El NÚMERO DE ORDEN (para un PRODUCTO ORDENADO) identifica la ORDEN, y el NÚMERO DEL PRODUCTO (para un PRODUCTO ORDENADO) identifica el PRODUCTO DE INVENTARIO. Conjuntamente, identifican a un PRODUCTO ORDENADO y solamente uno.
- Para cada PRODUCTO ORDENADO necesitamos saber la CANTIDAD ORDENADA y el PRECIO UNITARIO AL MOMENTO DE LA ORDEN.

Después de estudiar este capítulo, usted podrá leer y construir modelos de datos.

## Conceptos de sistemas para el modelado de datos

Hay varias notaciones para el modelado de datos. El modelo real frecuentemente se designa como un **diagrama de entidad relación** (*entity relationship diagram*, **ERD**) porque diseña los datos en términos de las entidades y las relaciones descritas por los datos. Hay varias notaciones para los ERD. La mayoría lleva el nombre de su inventor (por ejemplo, Chen, Martin, Bachman, Merise) o de acuerdo con un estándar publicado (por ejemplo, IDEF1X). Estos “lenguajes” de modelado de datos generalmente soportan los mismos conceptos y construcciones fundamentales. Hemos adoptado la notación de Martin (la ingeniería de la información) por su uso extendido y la herramienta de soporte CASE.

Exploremos algunos conceptos básicos que subyacen en todos los modelos de datos.

### > Las entidades

Todos los sistemas contienen datos, ¡generalmente montones de datos! Los datos describen “cosas”. Considere un sistema escolar. Éste incluye datos que describen elementos tales como estudiantes, maestros, cursos y aulas. Para cualquiera de estos elementos no es difícil imaginar una cierta cantidad de datos que describe cualquier instancia dada del mismo. Por ejemplo, los datos que describen a un estudiante específico podrían incluir nombre, dirección, número de teléfono, fecha de nacimiento, sexo, raza, especialidad (o carrera) y promedio general, por nombrar unos cuantos.

Necesitamos un concepto que represente en forma abstracta todas las instancias de un grupo de cosas semejantes. Llamamos a este concepto una entidad. Una **entidad** es algo acerca de lo cual el negocio necesita almacenar datos. En modelado de sistemas lo encontramos útil para asignar cada concepto abstracto a una forma. En este libro, una entidad se dibujará como un rectángulo con esquinas redondeadas (véase el margen). Esta forma representa todas las instancias de la entidad mencionada. Por ejemplo, la entidad ESTUDIANTE representa a todos los estudiantes en el sistema. Así, una entidad identifica las clases específicas de entidades y es distinguible de las otras entidades.

Las categorías de entidades (y los ejemplos) incluyen:

- Personas:** AGENCIA, CONTRATISTA, CLIENTE, DEPARTAMENTO, DIVISIÓN, EMPLEADO, INSTRUCTOR, ESTUDIANTE, PROVEEDOR. Observe que una clase de entidad de persona puede representar a los individuos, los grupos, o las organizaciones.
- Lugares:** REGIÓN DE VENTAS, EDIFICIO, CUARTO, OFICINA REGIONAL, CAMPUS.
- Objetos:** LIBRO, MÁQUINA, PARTE, PRODUCTO, MATERIA PRIMA, LICENCIA DEL SOFTWARE, PAQUETE DE SOFTWARE, HERRAMIENTA, MODELO DEL VEHÍCULO, VEHÍCULO. Una entidad del objeto puede representar objetos reales (tal como una licencia específica del software) o especificaciones para un tipo de objeto (tales como especificaciones para paquetes de software diferentes).
- Eventos:** SOLICITUD, PREMIO, CANCELACIÓN, CLASE, VUELO, FACTURA, ORDEN, INSCRIPCIÓN, RENOVACIÓN, REQUISICIÓN, RESERVACIÓN, VENTA, VIAJE.
- Conceptos:** CUENTA, BLOQUE DE TIEMPO, BONO, CURSO, FONDO, CAPACITACIÓN, SUMINISTROS.

### diagrama de entidad relación (*entity relationship diagram*, **ERD**)

Modelo de datos que utiliza varias notaciones para ilustrar los datos en términos de las entidades y relaciones descritas por esos datos.

**entidad** Clases de personas, lugares, objetos, eventos o conceptos acerca de las cuales necesitamos capturar y almacenar datos.

ESTUDIANTE

Una entidad

**instancia de entidad** Ocurrencia individual de una entidad.

**atributo** Propiedad o característica descriptiva de una entidad. Los sinónimos incluyen *elemento*, *propiedad* y *campo*.

**ESTUDIANTE**

- Nombre
- .Apellido paterno
- .Apellido materno
- .Nombre(s)
- Dirección
- .Nombre de la calle
- .Ciudad
- .Estado o provincia
- .País
- .Código Postal
- Teléfono
- .Código de área
- .Número de intercambio
- .Número dentro del intercambio
- Fecha de nacimiento
- Sexo
- Raza
- Especialidad
- Promedio general

Atributos y atributos compuestos

**atributo compuesto** Atributo que consta de otros atributos. Los sinónimos en los diferentes lenguajes de modelado de datos son numerosos: *atributo concatenado*, *atributo compuesto* y *estructura de datos*.

**tipo de datos** Propiedad de un atributo que define qué valores puede adoptar legítimamente el atributo.

**dominio** Propiedad de un atributo que define qué valores puede adoptar legítimamente el atributo.

Es importante distinguir entre una entidad y sus instancias. Una **instancia de entidad** es una ocurrencia individual de una entidad. Por ejemplo, la entidad ESTUDIANTE puede tener instancias múltiples: Mary, Joe, Mark, Susan, Cheryl, y así sucesivamente. En el modelado de los datos no nos preocupamos por estudiantes individuales porque reconocemos que cada estudiante está descrito por piezas similares de datos.

### > Los atributos

Si queremos almacenar datos acerca de una entidad, entonces necesitamos identificar qué piezas específicas de datos queremos almacenar acerca de cada instancia de una entidad dada. Llamamos **atributos** a estas piezas de datos. Como se observa al principio de esta sección, cada instancia de la entidad estudiante podría estar descrita por los siguientes atributos: nombre, dirección, número de teléfono, fecha de nacimiento, sexo, raza, especialidad, promedio general, y otros.

Nosotros ahora podemos extender nuestra abstracción gráfica de la entidad para incluir atributos registrando esos atributos dentro de la forma de entidad junto con el nombre (ver al margen).

Algunos atributos pueden agruparse lógicamente en superatributos, llamados **atributos compuestos**. Por ejemplo, el NOMBRE de un estudiante es de hecho un atributo compuesto que consta de APELLIDO PATERNO, APELLIDO MATERNO, NOMBRE(S). En el margen, mostramos una notación posible para los atributos compuestos. Observe que se pone un punto al principio de cada atributo primitivo que se incluye en el atributo complejo.

**Dominios** Al analizar un sistema, deberíamos definir esos valores para un atributo que son legítimos o que tienen sentido en los negocios. Los valores de cada atributo se definen en términos de tres propiedades: tipo de datos, dominio y por omisión.

El **tipo de datos** de un atributo define la clase de datos que pueden guardarse en ese atributo. La escritura de datos debería ser familiar a aquéllos de ustedes que hayan escrito programas de computadora; la declaración de tipos de variables es común en la mayoría de los lenguajes de programación. Para los propósitos del análisis de sistemas y la definición de los requerimientos del negocio, es útil declarar tipos lógicos de datos (no técnicos) para los atributos del negocio. Para esclarecer el tema usaremos los tipos de datos lógicos mostrados en la tabla 7.1.

El tipo de datos del atributo constriñe su dominio. El **dominio** de un atributo define qué valores puede adoptar en realidad el atributo. Finalmente, los diseñadores de sistemas deben usar tecnología para implementar los dominios de negocios de todos los atributos. La tabla 7.2 muestra cómo pueden expresarse los dominios lógicos para cada tipo de datos.

**TABLA 7.1 Tipos de datos lógicos representativos para atributos**

Tipo de dato lógico	Significado lógico de negocios
NÚMERO	Cualquier número, real o entero.
TEXTO	Una cadena de caracteres, incluyendo números. Cuando se incluyen números en un atributo de TEXTO, significa que no esperamos realizar operaciones aritméticas o comparaciones con esos números.
MEMO	Lo mismo que TEXTO pero con un tamaño indeterminado. Algunos sistemas de negocios requieren la capacidad de adjuntar notas potencialmente largas a un registro dado de base de datos.
FECHA	Cualquier fecha con cualquier formato.
TIEMPO	Cualquier hora con cualquier formato.
SÍ/NO	Un atributo que puede adoptar solamente uno de esos dos valores.
CONJUNTO DE VALORES	Un conjunto finito de valores. En la mayoría de los casos se establecería un esquema de códigos (por ejemplo, PI = alumno de primer ingreso, SA = alumno de segundo año, TA = alumno de tercer año, UA = alumno de último año, etc.).
IMAGEN	Cualquier ilustración o imagen.

**TABLA 7.2** Dominios lógicos representativos para tipos de datos lógicos

Tipo de datos	Dominio	Ejemplos
NÚMERO	Para los enteros, especifique el rango: {mínimo-máximo} Para los números reales, especifique el rango y la precisión: {mínimo.precisión-máximo.precisión}	{10-99}  {1.000-799.999}
TEXTO	TEXTO (tamaño máximo del atributo) <i>Generalmente los valores reales son infinitos; sin embargo, los usuarios pueden identificar ciertas restricciones en la narración.</i>	TEXTO (30)
MEMO	<i>No es aplicable. No hay restricciones lógicas de tamaño o contenido.</i>	<i>No es aplicable</i>
FECHA	Variación del formato MMDDYYYY. Para anotar al año 2000, no abrevie el año como YY. Los caracteres de formateado rara vez se almacenan; por lo tanto, no incluya guiones ni diagonales.	MMDDYYYY  MMYYYY YYYY
TIEMPO	Para los horarios AM/PM: HHMMT o Para horarios militares: HHMM	HHMMT  HHMM
SÍ/NO	{SÍ, NO}	{SÍ, NO} {ENCENDIDO, APAGADO}
CONJUNTO DE VALORES	{valor#1, valor#2,....valor#n} o  {tabla de códigos y significados}	{ALUMNO DE PRIMER INGRESO, ALUMNO DE SEGUNDO AÑO, ALUMNO DE TERCER AÑO, ALUMNO DE ÚLTIMO AÑO}  {PI = ALUMNO DE PRIMER INGRESO SA = ALUMNO DE SEGUNDO AÑO TA = ALUMNO DE TERCER AÑO UA = ALUMNO DE ÚLTIMO AÑO}
IMAGEN	<i>No es aplicable; sin embargo, cualesquiera de las características conocidas de las imágenes finalmente resultarán útiles para los diseñadores.</i>	<i>No es aplicable</i>

Finalmente, cada atributo debería tener un **valor lógico por omisión** que representa el valor de un atributo si su valor no es especificado por el usuario. La tabla 7.3 muestra valores por omisión posibles para un atributo. Observe que NOT NULL es una forma de especificar que cada instancia del atributo debe tener un valor, mientras que NULL es una forma de especificar que algunas instancias del atributo pueden ser optativas, o no tener un valor.

**Identificación** Una entidad tiene muchas instancias, quizá miles o millones. Allí existe una necesidad de identificar en forma única cada instancia basada en el valor de datos de uno o más atributos. Por tanto, cada entidad debe tener una **clave**. Por ejemplo, cada instancia de la entidad *estudiante* podría identificarse en forma única por el atributo *clave número de estudiante*. Dos estudiantes de ninguna manera pueden tener el mismo número de estudiante.

Algunas veces se requiere más de un atributo para identificar en forma única a una instancia de una entidad. Una clave consistente en un grupo de atributos se llama **clave concatenada**. Por ejemplo, cada instancia de entidad *DVD* en una tienda de video podría estar identificada en forma única por la concatenación de *NÚMERO DE TÍTULO* y el *NÚMERO DE*

**valor por omisión** Valor que se registrará si el usuario no especifica un valor.

**clave** Atributo o grupo de atributos que adopta un valor único para cada instancia de entidad. Algunas veces se llama un *identificador*.

**clave concatenada** Grupo de atributos que identifican en forma única a la instancia de una entidad. Sus sinónimos son *clave compuesta* y *clave combinada*.

TABLA 7.3 Valores permisibles por omisión de atributos

Valor por omisión	Interpretación	Ejemplos
Un valor legal del dominio (como se describe anteriormente)	En el caso de un atributo, si el usuario no especifica un valor, entonces use este valor.	0 1.00 FR
NINGUNO O NULO	En el caso de un atributo, si el usuario no especifica un valor, entonces déjelo en blanco.	NINGUNO NULO
REQUERIDO O NO NULO	En el caso de un atributo, requiere que el usuario ingrese un valor legal del dominio. (Esto se usa cuando ningún valor en el dominio es suficientemente común como para ser un valor por omisión y sin embargo debe ingresarse algún valor.)	REQUERIDO NO NULO

COPIA. El NÚMERO DE TÍTULO por sí mismo sería inadecuado porque la tienda puede poseer muchas copias de un solo título. El NÚMERO DE COPIA por sí mismo también sería inadecuado, ya que presumiblemente tenemos una *copia #1* para cada título que poseemos. Necesitamos que ambas piezas de datos identifiquen una cinta específica (por ejemplo, la *copia #7* de *La Guerra de las Galaxias: La venganza de los Sith*). En este libro le pondremos nombre al grupo, así como también a los atributos individuales. Por ejemplo, la clave concatenada para DVD se registraría como sigue:

ID DEL DVD  
.NÚMERO DE TÍTULO  
.NÚMERO DE COPIA

Frecuentemente, una entidad puede tener más de una clave. Por ejemplo, la entidad EMPLEADO puede identificarse de manera única por el NÚMERO DE SEGURIDAD SOCIAL, o el NÚMERO DE EMPLEADO asignado por la compañía, o la DIRECCIÓN DE CORREO ELECTRÓNICO. Cada uno de estos atributos se llama una *clave candidata*. La **clave candidata** es “candidata a convertirse en la clave primaria” de las instancias de una entidad. Algunas veces se le llama *identificador del candidato* (*candidate identifier*). (Una clave candidata puede ser un atributo individual o una clave concatenada.) Una **clave primaria** es la clave candidata que más comúnmente se usará para identificar en forma única una instancia de entidad individual. El valor por omisión de una clave primaria siempre es *NOT NULL*, porque si la clave no tiene valor, no puede servir el propósito de identificar la instancia de una entidad. Cualquier clave candidata que no esté seleccionada para convertirse en la clave primaria se llama **clave alterna**. Un sinónimo común es *clave secundaria*. En el margen mostramos nuestra notación de las claves primarias y alternas. Todas las claves candidatas deben ser primarias o alternas; por consiguiente, no usamos una notación separada para las claves candidatas. Todos los atributos que no son parte de la clave primaria se llaman atributos sin clave.

Algunas veces también es necesario identificar un subconjunto de instancias de una entidad en contraposición a una instancia sola. Por ejemplo, podemos requerir que una forma simple identifique a todos los estudiantes masculinos y todas las estudiantes femeninas. Un **criterio de subconjuntos** es un atributo (o atributo concatenado) cuyos valores finitos dividen a todas las instancias de entidad en subconjuntos útiles. Algunas veces esto se llama una *entrada inversa*. En nuestra entidad ESTUDIANTE, el atributo SEXO divide las instancias de ESTUDIANTE en dos subconjuntos: los estudiantes masculinos y los estudiantes femeninos. En general, los criterios de subconjuntos son útiles sólo cuando un atributo tiene un número finito (que significa “limitado”) de valores legítimos. Por ejemplo, el PROMEDIO GENERAL no sería un buen criterio de subconjuntos porque hay 999 valores posibles entre 0.00 y 10.00 para ese atributo.\* En el margen se muestra una notación para los criterios de subconjuntos.

## > Las relaciones

Conceptualmente, las entidades y los atributos no existen en forma aislada. Las cosas que representan interactúan y se impactan entre sí para soportar la misión del negocio.

\* Nota del R.T.: En Estados Unidos se ocupa la escala del GPA (*Grade Point Average*), que tiene 399 posibles valores entre 0.00 y 4.00.

**clave candidata** Una de varias claves que pueden servir como clave primaria de una entidad. También llamada *identificador de candidato*.

**clave primaria** Clave candidata que se usará muy comúnmente para identificar en forma única a una instancia de entidad individual.

**clave alterna** Clave candidata que no se selecciona para convertirse en clave primaria. Un sinónimo es *clave secundaria*.

**criterio(s) de subconjuntos** Atributo(s) cuyos valores finitos dividen a las instancias de la entidad en subconjuntos. Algunas veces llamados *entrada inversa*.



Es por ello que nosotros introducimos el concepto de una relación. Una **relación** es una asociación natural de negocios que existe entre una o más entidades. La relación puede representar un evento que enlaza a las entidades o meramente una afinidad lógica que existe entre las entidades. Considere, por ejemplo, las entidades ESTUDIANTE y PLAN DE ESTUDIOS. Podemos hacer las siguientes aseveraciones de negocios que enlazan a los estudiantes y los cursos:

- Un ESTUDIANTE SE REGISTRA EN UNO O MÁS PLANES DE ESTUDIOS.
- Un PLAN DE ESTUDIOS ESTÁ SIENDO ESTUDIADO POR cero, uno o más ESTUDIANTES.

Las expresiones subrayadas definen las relaciones de negocios que existen entre las dos entidades.

Gráficamente podemos ilustrar a esta asociación entre el ESTUDIANTE y el PLAN DE ESTUDIOS como se muestra en la figura 7.2. La línea de conexión representa una relación. Las expresiones describen la relación. Observe que todas las relaciones son implícitamente bidireccionales, lo que significa que pueden interpretarse en ambas direcciones (como lo sugieren las aseveraciones de negocios anteriores). Los métodos de modelado de datos pueden diferir en su forma de nombrar las relaciones: algunos incluyen ambas expresiones y los otros incluyen una sola expresión.

**Cardinalidad** La figura 7.2 también muestra la complejidad o el *grado* de cada relación. Por ejemplo, si sabemos leerla, la figura 7.2 puede contestar las siguientes preguntas:

- ¿Debe existir una instancia de ESTUDIANTE para cada instancia de PLAN DE ESTUDIOS? ¡No!
- ¿Debe existir una instancia de PLAN DE ESTUDIOS para cada instancia de ESTUDIANTE? ¡Sí!
- ¿Cuántas instancias de PLAN DE ESTUDIOS pueden existir para cada instancia de ESTUDIANTE? ¡Muchas!
- ¿Cuántas instancias de ESTUDIANTE pueden existir para cada instancia de PLAN DE ESTUDIOS? ¡Muchas!

Llamamos a este concepto *cardinalidad*. La **cardinalidad** define los números mínimo y máximo de ocurrencias de una entidad que pueden relacionarse con una ocurrencia individual de la otra entidad. Como todas las relaciones son bidireccionales, la cardinalidad debe definirse en ambas direcciones para cada relación. Una notación gráfica popular de la cardinalidad se muestra en la figura 7.3. Los símbolos de muestra de la cardinalidad se mostraron en la figura 7.2.

Conceptualmente, la cardinalidad nos dice las siguientes reglas acerca de las entidades de datos mostradas en la figura 7.2:

- Cuando insertamos una instancia ESTUDIANTE en la base de datos, debemos vincular (asociar) ese ESTUDIANTE con cuando menos una instancia de PLAN DE ESTUDIOS. En términos de negocios, “un estudiante no puede ser admitido sin declarar una especialidad”. (La mayoría de escuelas incluirían una instancia de PLAN DE ESTUDIOS llamada “indeciso” o “no declarado”).
- Un ESTUDIANTE puede estudiar más de un PLAN DE ESTUDIOS y debemos poder almacenar datos que indiquen todos los PLANES DE ESTUDIO para un ESTUDIANTE dado.
- Debemos insertar un PLAN DE ESTUDIOS antes de que podamos vincular (asociar) a los ESTUDIANTES con ese PLAN DE ESTUDIOS. Por esto es que un PLAN DE ESTUDIOS puede tener cero estudiantes: ningún estudiante ha sido admitido aún en ese PLAN DE ESTUDIOS.
- Una vez que un PLAN DE ESTUDIOS ha sido insertado en la base de datos, podemos vincular (asociar) a muchos ESTUDIANTES con ese PLAN DE ESTUDIOS.

**Grado** Otra medida de la complejidad de una relación de datos es su grado. El **grado** de una relación es el número de entidades que participan en la relación. Todas las relaciones que hemos explorado hasta ahora son *binarias* (grado = 2). En otras palabras, dos entidades diferentes participaron en la relación.

**ESTUDIANTE**

- Número de estudiante (clave primaria)
- Número de seguridad social (clave alterna)
- Nombre
  - .Apellido paterno
  - .Apellido materno
  - .Nombre(s)
- Dirección
  - .Dirección de la calle
  - .Ciudad
  - .Estado o provincia
  - .País
  - .Código Postal
- Teléfono
  - .Código de área
  - .Número de intercambio
  - .Número dentro del intercambio
- Fecha de nacimiento
- Sexo (criterio 1 de subconjuntos)
- Raza (criterio 2 de subconjuntos)
- Especialidad (criterio 3 de subconjuntos)
- Promedio general

**Claves y criterios de subconjuntos**

**relación** Asociación natural de negocios entre una o más entidades.

**cardinalidad** Números mínimo y máximo de ocurrencias de una entidad que pueden relacionarse con una ocurrencia individual de la otra entidad.

**grado** Número de entidades que participan en una relación.



**FIGURA 7.2**

Una relación (muchos a muchos)

**FIGURA 7.3**

Notaciones de cardinalidad

INTERPRETACIÓN DE LA CARDINALIDAD	INSTANCIAS MÍNIMAS	INSTANCIAS MÁXIMAS	NOTACIÓN GRÁFICA
Exactamente uno (uno y solamente uno)	1	1	
Cero o uno	0	1	
Uno o más	1	muchos (>1)	
Cero, uno o más	0	muchos (>1)	
Más de uno	>1	>1	

**relación recursiva** Relación que existe entre dos instancias de la misma entidad.

**entidad asociativa** Entidad que hereda su clave primaria de más de alguna otra entidad.

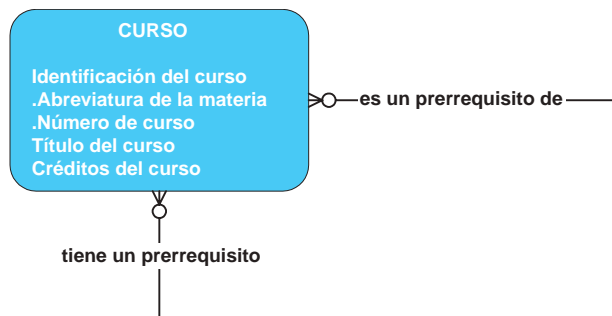
También pueden existir relaciones entre las diferentes instancias de la misma entidad. Llamamos a esto una **relación recursiva** (grado = 1). Por ejemplo, en su escuela un curso puede ser un prerrequisito para otros cursos. De modo semejante, un curso puede tener varios otros cursos como su prerrequisito. La figura 7.4 muestra esta relación recursiva de muchos a muchos.

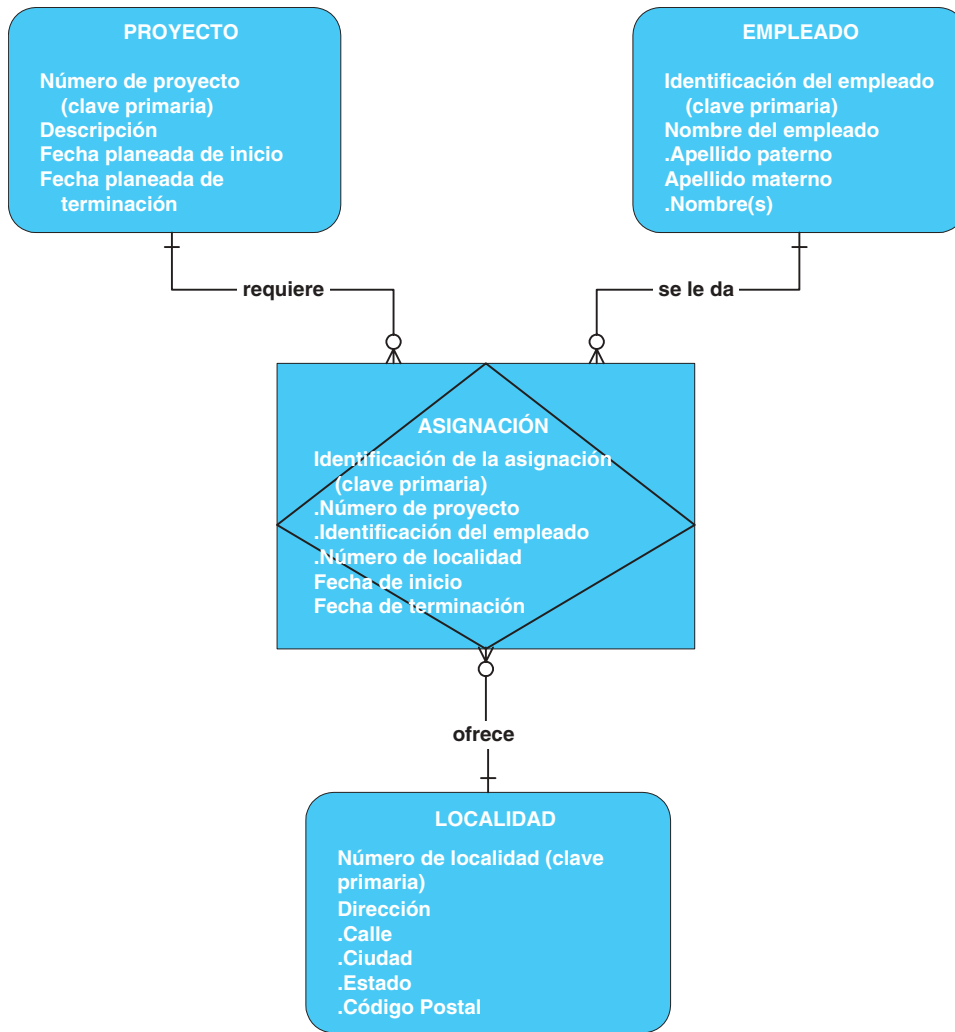
También pueden existir relaciones entre más de dos entidades diferentes. Algunas veces éstas se llaman relaciones *de orden N*. En la figura 7.5 se muestra un ejemplo de una *relación de tercer orden*, o *ternaria*. Se ilustra una relación *de orden N* con una construcción nueva de entidad designada *entidad asociativa*. Una **entidad asociativa** es una entidad que hereda su clave primaria de más de una entidad diferente (llamadas *padres*). Cada parte de esa clave concatenada señala a una y solamente una instancia de cada una de las entidades conectadas.

En la figura 7.5 la entidad asociativa ASIGNACIÓN (observe la forma característica) corresponde a un EMPLEADO, una POSICIÓN y un PROYECTO. Para cada instancia de ASIGNACIÓN, la

**FIGURA 7.4**

Una relación recursiva





**FIGURA 7.5**  
Una relación ternaria

clave indica cuál CÉDULA DE IDENTIDAD DEL EMPLEADO, cuál NÚMERO DE POSICIÓN y cuál NÚMERO DE PROYECTO se combinan para formar esa asignación.

Como se muestra también en la figura 7.5, una entidad asociativa puede describirse por sus propios atributos que no son claves. Además de la clave primaria, una ASIGNACIÓN se describe por los atributos FECHA INICIAL y FECHA FINAL. Si usted piensa acerca de eso, ninguno de estos atributos describen a un EMPLEADO, POSICIÓN o PROYECTO; éstos describen una instancia individual de la relación entre una instancia de cada una de esas entidades.

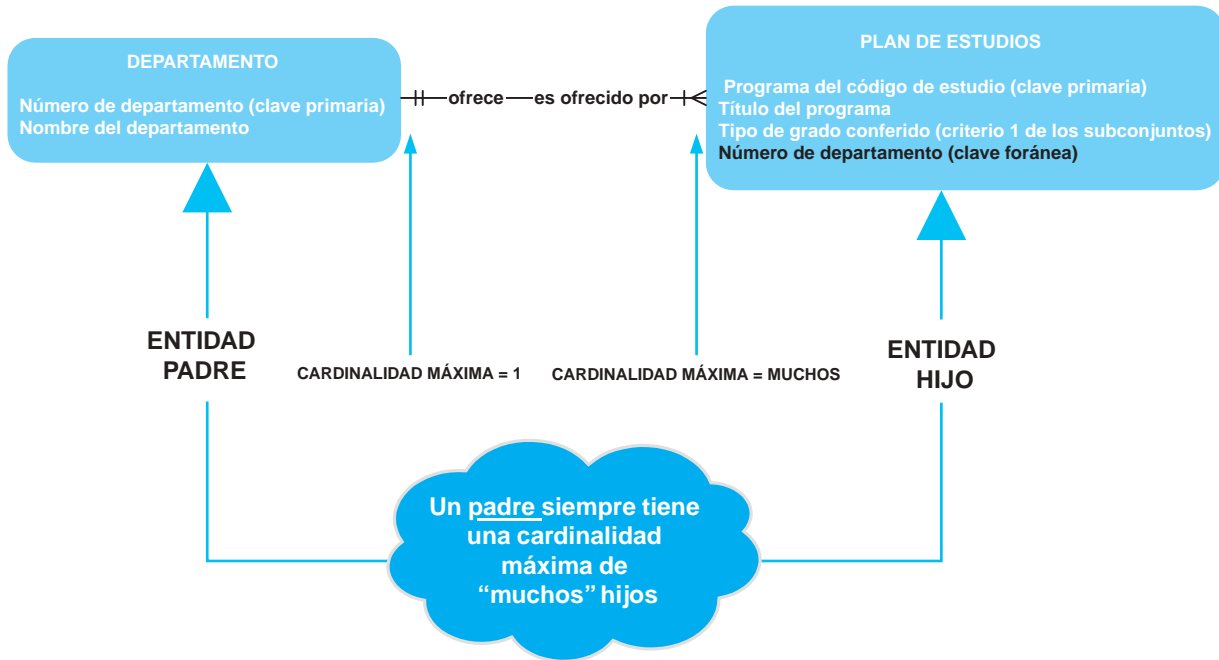
**Claves foráneas** Una relación significa que las instancias de una entidad se relacionan con las instancias de otra entidad. Deberíamos poder identificar esas instancias para cualquier entidad dada. La habilidad para identificar las instancias de entidad específicas relacionadas incluye establecer claves foráneas. Una **clave foránea** es una clave primaria de una entidad que se originó (se duplicó) en otra entidad para identificar las instancias de una relación. Una clave foránea (siempre en una **entidad hijo**) siempre corresponde a la clave primaria (en una **entidad padre**). En la figura 7.6(a) mostramos el concepto de claves foráneas con nuestro modelo simple de datos. Observe que la cardinalidad máxima para departamento es “uno”, mientras que la cardinalidad máxima para plan de estudios es “muchos”. En este caso, la entidad *padre* es el departamento y el plan de estudios es la entidad *hijo*. La clave primaria siempre es aportada por el padre al hijo como una clave foránea. Así, una instancia de plan de estudios ahora tiene una clave foránea nombre del departamento cuyo valor señala la instancia de departamento que le ofrece ese plan de estudios. (Las claves foráneas nunca son originadas del hijo al padre.)

**clave foránea** Clave primaria de una entidad que se usa en otra entidad para identificar las instancias de una relación.

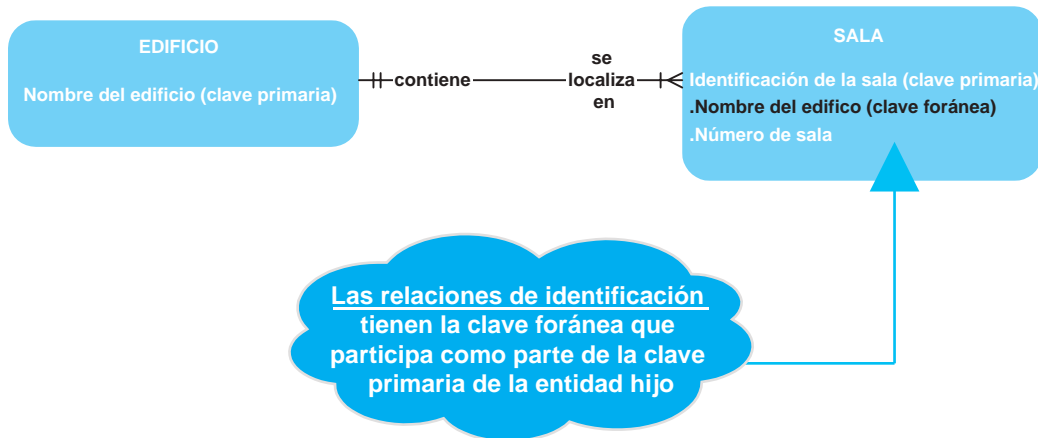
**entidad hijo** Entidad de dato que deriva uno o más atributos de otra entidad, llamada padre. En una relación uno a muchos el hijo es la entidad del lado de los “muchos”.

**entidad padre** Entidad de dato que aporta uno o más atributos a otra entidad, llamada hijo. En una relación uno a muchos el padre es la entidad en el lado de “uno”.

(a)



(b)

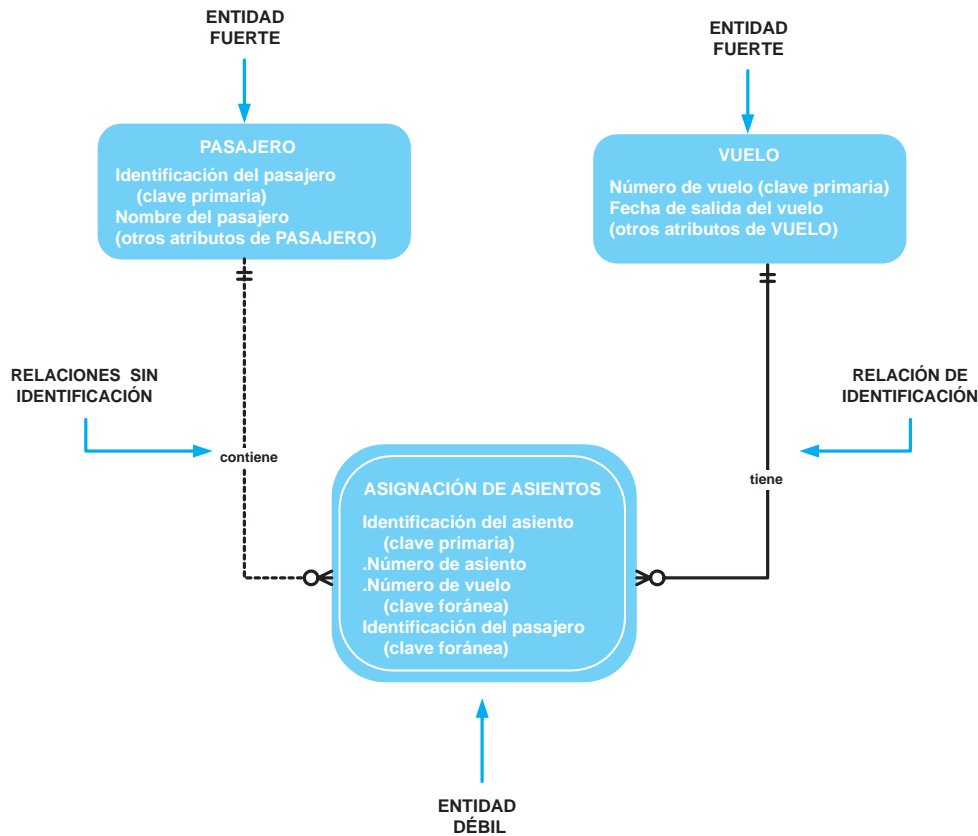


**FIGURA 7.6** Claves foráneas

**relación sin identificación** Relación en la cual cada una de las entidades participante tiene su propia clave primaria independiente.

En nuestro ejemplo, a la relación entre PLAN DE ESTUDIOS y DEPARTAMENTO se le denomina *relación sin identificación*. Las **relaciones sin identificación** son aquellas en las cuales cada una de las entidades participantes tiene su propia clave primaria independiente. En otras palabras, ninguno de los atributos de la clave primaria es compartido. Las entidades PLAN DE ESTUDIOS y DEPARTAMENTO también se denominan entidades *fuertes* o independientes porque ninguna de las dos depende de otra entidad para su identificación. Algunas veces, sin embargo, una clave foránea puede participar como parte de la clave primaria de la entidad hijo. Por ejemplo, en la figura 7.6(b) la entidad padre EDIFICIO aporta su clave a la entidad CUARTO. Así, NOMBRE DEL EDIFICIO sirve como una clave foránea para relacionar CUARTO y EDIFICIO y en conjunción con la CÉDULA DE IDENTIDAD DEL CUARTO para identificar en forma única una instancia dada de CUARTO. En esas situaciones, a la relación entre la

**FIGURA 7.7**  
Notaciones de una entidad débil y de una relación sin identificación



entidad padre y la entidad hijo se le llama *relación de identificación*. Las **relaciones de identificación** son aquellas en las cuales la entidad padre aporta su clave primaria para hacerse parte de la clave primaria de la entidad hijo. La entidad hijo de cualquier relación de identificación frecuentemente se llama una *entidad débil* porque su identificación depende de la existencia de la entidad padre.

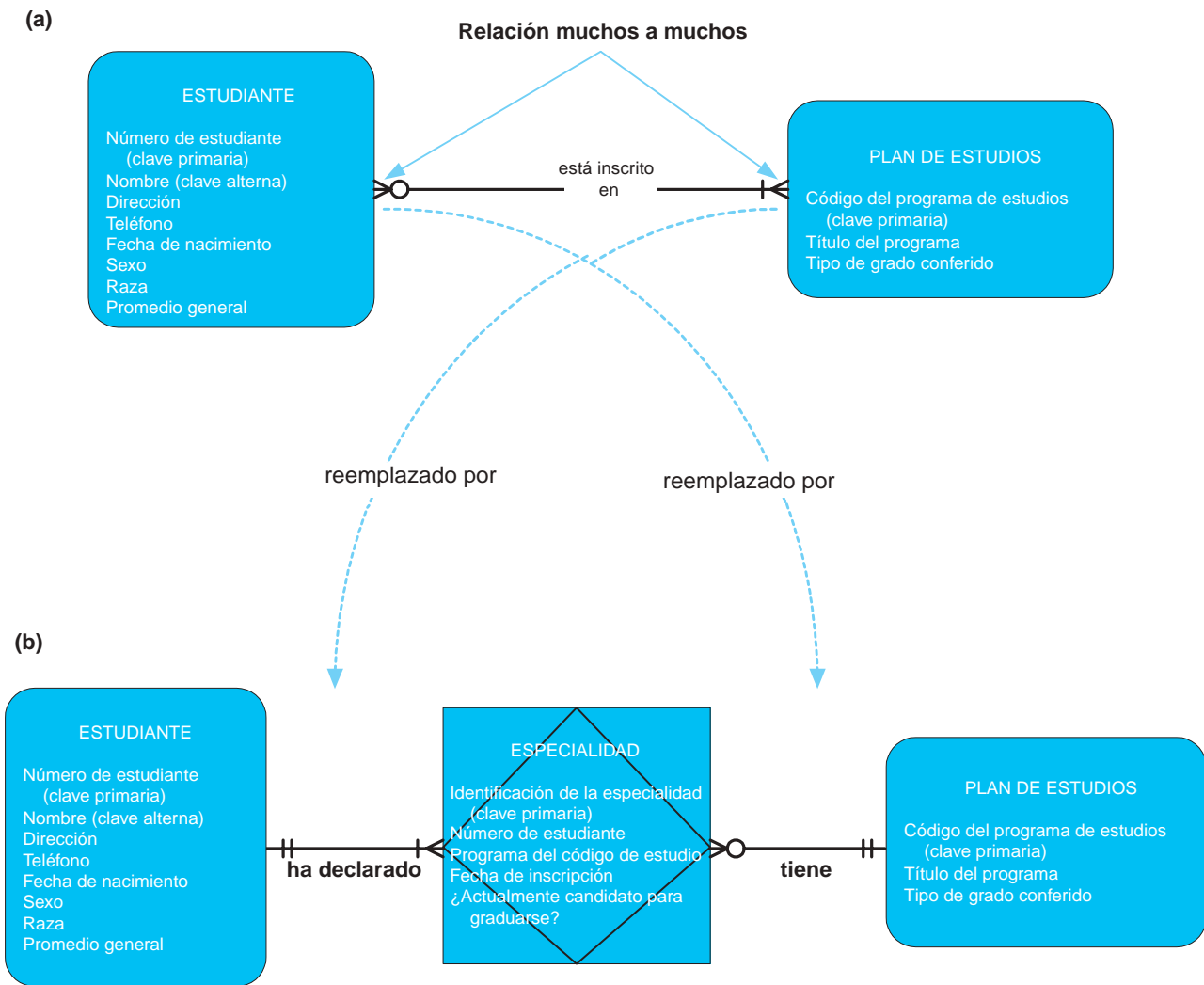
Las herramientas CASE y los métodos de modelado de datos más populares usan notaciones diferentes para distinguir entre relaciones de identificación y sin identificación y entre entidades fuertes y débiles. En la figura 7.7 usamos la notación de *línea segmentada* para representar la relación sin identificación entre PASAJERO y ASIGNACIÓN DE ASIENTOS. Ya que parte de la clave primaria de ASIGNACIÓN DE ASIENTOS es la clave foránea NÚMERO DE VUELO de la entidad padre VUELO, la relación es una relación de identificación y se representa usando una *línea continua*. Finalmente, la asignación de asientos es una entidad débil porque recibe la clave primaria de vuelo para componer su propia clave primaria. Una entidad débil se representa usando un símbolo compuesto por un *rectángulo redondeado dentro de un rectángulo redondeado*.

NOTA: Para reforzar los conceptos anteriores de las relaciones de identificación y sin identificación y de las entidades fuertes contra entidades débiles, y para ser consistente con los métodos de modelado de datos más populares y con las herramientas CASE más ampliamente usadas, los autores usan las notaciones de modelado en todos los ejemplos subsiguientes de modelado de datos presentados en el libro.

¿Qué ocurre si usted no puede diferenciar entre padre e hijo? Por ejemplo, en la figura 7.8(a) se ve que un PLAN DE ESTUDIOS enrola cero, uno o más ESTUDIANTES. Al mismo tiempo, vemos que un ESTUDIANTE se inscribe en uno o más PLANES DE ESTUDIO. La cardinalidad máxima por ambos lados es “muchos”. Entonces, ¿cuál es el padre y cuál es el hijo? ¡Usted no lo puede decir! A esto se le llama una *relación no específica*. Una **relación no específica** (o *relación de muchos para muchos*) es una en la cual se asocian muchas instancias de una entidad con muchas instancias de otra entidad. Tales relaciones son adecuadas sólo para los modelos preliminares de datos y deberán resolverse tan rápido como sea posible.

**relación de identificación** Relación donde muchas instancias de una entidad se asocian con muchas instancias de otra entidad. También llamada relación muchos a muchos.

**relación no específica** Relación donde muchas instancias de una entidad se asocian con muchas instancias de otra entidad. También llamada *relación de muchos a muchos*.

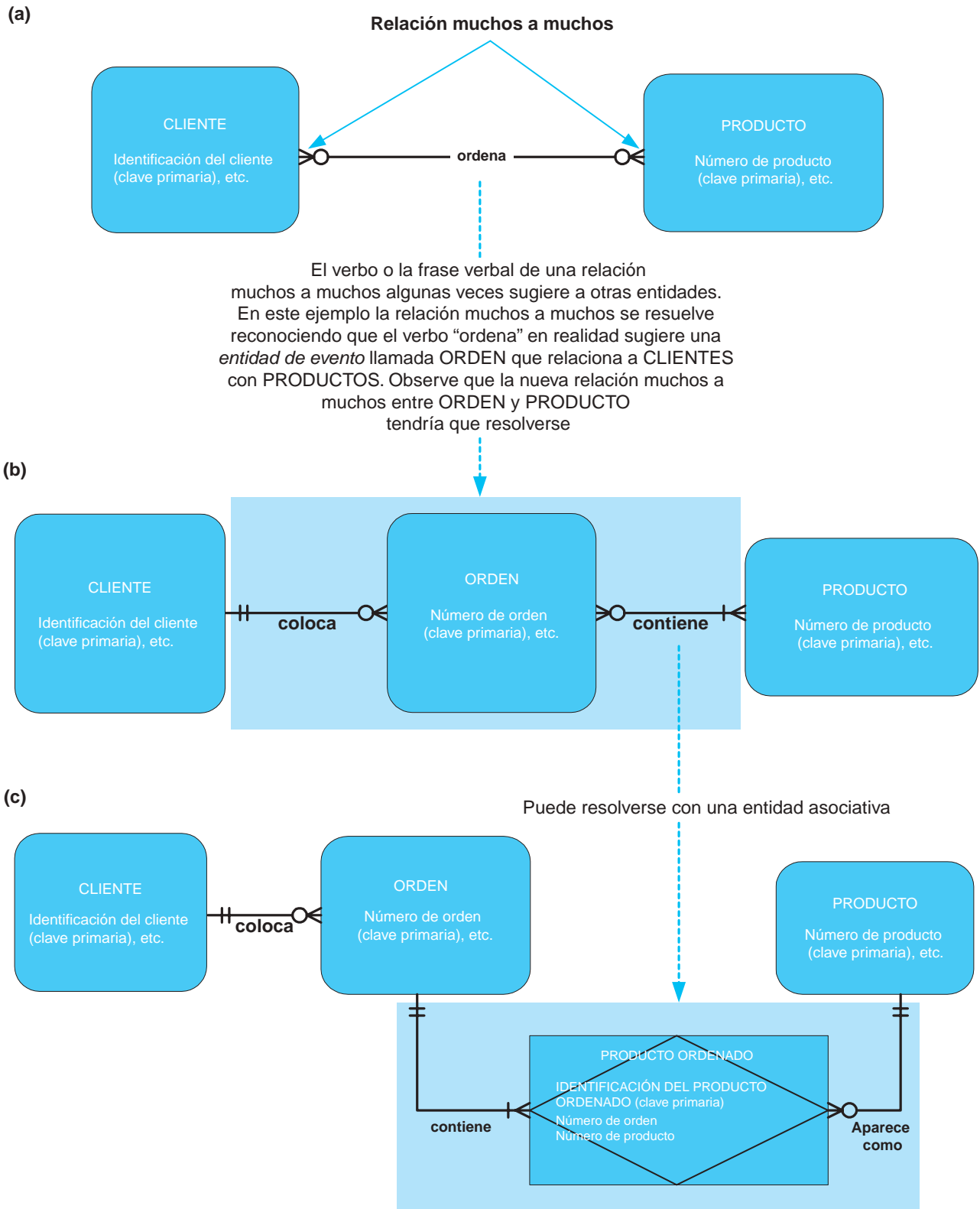


**FIGURA 7.8** Transformación de las relaciones no específicas con una entidad asociativa

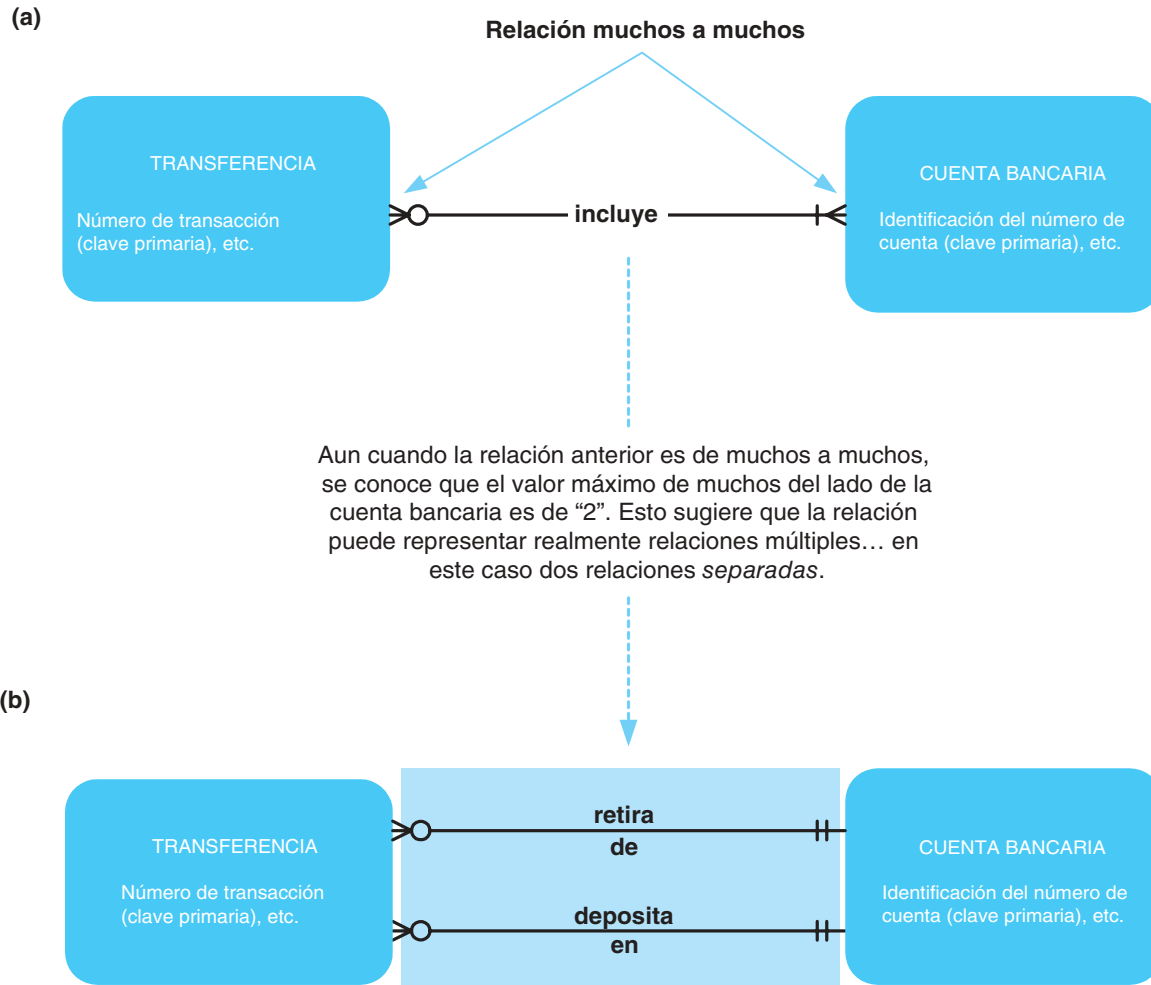
Muchas relaciones no específicas pueden resolverse en un par de relaciones uno a muchos. Como se ilustra en la figura 7.8(b), cada entidad se convierte en un padre. Una *entidad* nueva, *asociativa*, es introducida como el hijo de cada padre. En la figura 7.8(b), cada instancia de ESPECIALIDAD representa la inscripción de un ESTUDIANTE en un PLAN DE ESTUDIOS. Si un estudiante lleva dos especialidades, ese estudiante tendrá dos instancias de la entidad ESPECIALIDAD.

Estudie cuidadosamente la figura 7.8(b). Para las entidades asociativas, la cardinalidad del hijo al padre siempre es uno y solamente uno. Eso tiene sentido porque una instancia de ESPECIALIDAD debe corresponder a uno y solamente a un ESTUDIANTE y a uno y solamente a un PLAN DE ESTUDIOS. La cardinalidad de padre a hijo depende de la regla de negocios. En nuestro ejemplo, un ESTUDIANTE debe declarar una o más ESPECIALIDADES. Inversamente, un PLAN DE ESTUDIOS está siendo estudiado por cero, una o más ESPECIALIDADES; tal vez es nueva y nadie ha sido admitido todavía. Una entidad asociativa también puede describirse por sus propios atributos que no son claves (tales como la FECHA DE INSCRIPCIÓN y ¿CANDIDATO ACTUAL PARA GRADUARSE?). Finalmente, las entidades asociativas heredan las claves primarias de los padres; así, todas las entidades asociativas son entidades débiles.

No todas las relaciones no específicas pueden y deben resolverse automáticamente como se describe antes. Ocasionalmente las relaciones no específicas resultan del fracaso del analista para identificar la existencia de otras entidades. Por ejemplo, examine la relación entre CLIENTE y PRODUCTO en la figura 7.9(a). Reconozca que la relación “órdenes”



**FIGURA 7.9** Transformación de las relaciones no específicas mediante el reconocimiento de una entidad fundamental de negocios



**FIGURA 7.10** Transformación de las relaciones no específicas mediante el reconocimiento de relaciones separadas

entre CLIENTE y PRODUCTO sugiere un evento acerca del cual un usuario pudiese querer almacenar datos. Ese evento representa una entidad de evento llamada ORDEN diseñada en la figura 7.9(b). En realidad, CLIENTE y PRODUCTO no tienen una relación natural y directa como se bosquejó en la figura 7.9(a). Más bien se relacionan indirectamente mediante una ORDEN. Así, nuestra relación muchos a muchos fue reemplazada por relaciones separadas entre CLIENTE, ORDEN y PRODUCTO. Observe que la relación entre ORDEN y PRODUCTO es una relación muchos a muchos. Esa relación necesitaría resolverse reemplazándola con una entidad asociativa y dos relaciones uno a muchos, como se ilustra en la figura 7.9(c).

Finalmente, algunas relaciones poco específicas pueden resolverse introduciendo relaciones separadas. Examine la relación muchos a muchos entre la TRANSFERENCIA y la CUENTA BANCARIA mostrada en la figura 7.10(a). Aun cuando es verdad que una transacción de TRANSFERENCIA incluye a muchas CUENTAS DE BANCO y una CUENTA DE BANCO puede estar involucrada en muchas transacciones de TRANSFERENCIA, ¡debemos tener cuidado! Algunas veces las notaciones de modelado de datos nos pueden desorientar. Técnicamente, una transacción individual de TRANSFERENCIA involucra a dos CUENTAS DE BANCO. Cuando conocemos el número máximo específico de ocurrencias de una relación de muchos a muchos, a menudo esto sugiere que nuestra relación original es débil o demasiado general. Observe en la figura 7.10(b) que nuestra relación “involucra” fue reemplazada por dos relaciones uno a muchos separadas que describen más exactamente las relaciones de negocios entre una TRANSFERENCIA y las CUENTAS DE BANCO.



**Generalización** La mayoría de la gente asocia el concepto de generalización con las modernas técnicas orientadas a objetos. En realidad, los conceptos han sido aplicados por los modeladores de datos durante largos años. La generalización es un enfoque que trata de descubrir y sacar provecho de los aspectos comunes entre las entidades. La **generalización** es una técnica a través de la cual los atributos que son comunes para varios tipos de una entidad se agrupan en su propia entidad. Considere, por ejemplo, una escuela típica. Una escuela inscribe a los estudiantes y emplea a los empleados (en una universidad, una persona podría ser ambos). Hay varios atributos que son comunes para ambas entidades; por ejemplo, nombre, sexo, raza, estado civil, y posiblemente incluso una clave basada en el número de seguridad social. Podríamos consolidar estos atributos comunes en un supertipo de entidad designado persona. Un **supertipo** de entidad es una entidad cuyas instancias almacenan atributos que son comunes para uno o más subtipos de entidad.

El supertipo de entidad tendrá una o más relaciones *uno a uno* para los *subtipos* de entidad. Estas relaciones algunas veces se llaman relaciones “es un” (o “fue un,” o “podría ser un”) porque cada instancia del supertipo “es también una” instancia de uno o más subtipos. Un **subtipo** de entidad es una entidad cuyas instancias heredan algunos atributos comunes de un supertipo de entidad y luego suma otros atributos que son únicos para una instancia del subtipo. En nuestro ejemplo, “una PERSONA es un empleado, o un estudiante, o ambos”. La parte superior media de la figura 7.11 ilustra esta generalización ❶ como una jerarquía. Observe que los subtipos ESTUDIANTE y EMPLEADO han heredado atributos de PERSONA, además de añadir los suyos propios.

Al extender la metáfora, vemos que una entidad puede ser tanto un supertipo como un subtipo. Regresando a la figura 7.11 vemos que un ESTUDIANTE (que era un subtipo de PERSONA) tiene sus propios subtipos. En el diagrama vemos que un ESTUDIANTE es ❷ ya sea un PROSPECTO, o un ESTUDIANTE ACTUAL, o un ALUMNO TRUNCO (que desistió sin haber terminado), y ❸ un ESTUDIANTE podría ser un EX ALUMNO. Estos subtipos adicionales heredan todos los atributos de ESTUDIANTE así como también los de PERSONA. Finalmente, observe que todos los subtipos son entidades débiles.

A través de la herencia, el concepto de generalización en los modelos de datos nos permite reducir el número de atributos mediante el uso compartido cuidadoso de atributos comunes. Los subtipos no sólo heredan los atributos, sino también los tipos de datos, los dominios y las acciones por omisión de esos atributos. Esto puede ampliar grandemente la consistencia con la cual tratamos los atributos que se aplican a muchas entidades diferentes (por ejemplo, fechas, nombres, direcciones, vigencia, etc.).

Además de heredar atributos, los subtipos también heredan relaciones con otras entidades. Por ejemplo, todos los EMPLEADOS y todos los ESTUDIANTES heredan la relación ❹ entre PERSONA y DIRECCIÓN. Pero solamente los EMPLEADOS heredan la relación ❺ con CONTRATOS. Y sólo un EX ALUMNO puede relacionarse con ❻ un GRADO OTORGADO.

---

**generalización** Concepto según el cual los atributos que son comunes a varios tipos de entidad se agrupan en su propia entidad.

---

**supertipo** Entidad cuyas instancias almacenan atributos que son comunes a uno o más subtipos de entidad.

---

**subtipo** Entidad cuyas instancias pueden heredar atributos comunes provenientes del supertipo de su entidad.

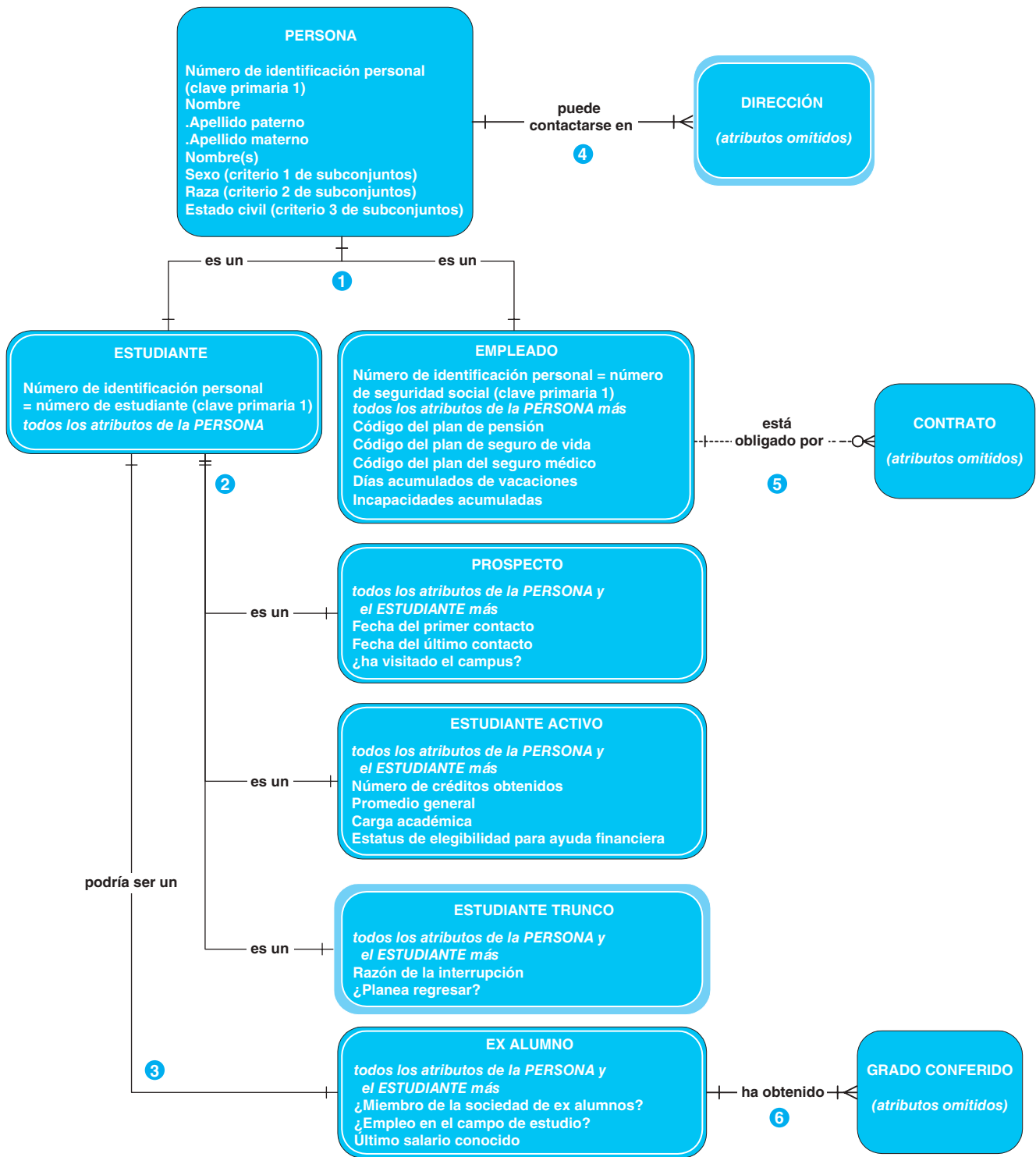
## El proceso del modelado lógico de datos

Ahora que usted entiende los conceptos básicos de los modelos de datos podemos examinar el proceso de modelado de datos. ¿Cuándo lo hace usted? ¿Cuántos modelos de datos pueden trazarse? ¿Qué tecnología existe para soportar el proceso?

El modelado de datos puede realizarse durante varios tipos de proyectos y en fases múltiples de proyectos. Los modelos de datos son progresivos; no hay un modelo “final” de datos para un negocio o una aplicación. En lugar de eso, un modelo de datos debe considerarse como un documento vivo que cambiará en respuesta a un negocio cambiante. Lo ideal es que los modelos de datos se almacenen en un repositorio a fin de que puedan recuperarse, expandirse y editarse con el paso del tiempo. Examinemos cómo el modelado de datos puede entrar en juego durante la planeación y el análisis de sistemas.

### > El modelado estratégico de datos

Muchas organizaciones seleccionan proyectos de aplicación de desarrollo basados en los planes estratégicos de sistemas de información. La planeación estratégica es un proyecto separado. Este proyecto produce un plan estratégico de sistemas de información que de-



**FIGURA 7.11** Una jerarquía de generalización

fine una vista global y la arquitectura de los sistemas de información. Casi siempre esta arquitectura incluye un *modelo de datos de la empresa*. La ingeniería de la información es una metodología que abarca este enfoque.

Un modelo de datos de la empresa típicamente identifica sólo lo fundamental de las entidades. Las entidades están por lo general definidas (como en un diccionario), pero no están descritas en términos de claves o atributos. El modelo de datos de la empresa puede

o no incluir las relaciones (dependiendo de los estándares de la metodología planificadora y del nivel de detalle deseado por gerencia ejecutiva). Si se incluyen las relaciones, muchas de ellas no serán específicas (un concepto presentado anteriormente en el capítulo).

¿Cómo afecta un modelo de datos de una empresa al subsiguiente desarrollo de aplicaciones? Parte de la estrategia del plan de información identifica los proyectos de desarrollo de aplicaciones y los prioriza de acuerdo con cualesquiera criterios de administración que considere apropiados. A medida que estos proyectos comienzan, los subconjuntos apropiados de la arquitectura de sistemas de información, incluyendo un subconjunto del modelo de datos de la empresa, se suministran al equipo de desarrollo de aplicaciones como un punto de partida.

El modelo de datos de la empresa generalmente se almacena en un repositorio corporativo. Cuando comienza el proyecto de desarrollo de aplicaciones, el subconjunto del modelo de datos de la empresa (así como los otros modelos) es exportado del repositorio corporativo a un repositorio de proyecto. Una vez que el equipo de proyecto completa el análisis y diseño de los sistemas, los modelos de datos expandidos y afinados son importados de vuelta al repositorio corporativo.

## > El modelado de datos durante el análisis de sistemas

En el análisis de los sistemas y en este capítulo, enfocaremos la atención en el modelado *lógico* de los datos como una parte del análisis de los sistemas. El modelo de datos para un sistema de información individual generalmente se llama **modelo de datos de la aplicación**.

Los modelos de datos raramente se construyen durante la fase de definición del alcance del análisis de sistemas. La corta duración de esa fase los hace imprácticos. Si existe un modelo de datos de la empresa, podría recuperarse el subconjunto de ese modelo que sea aplicable al proyecto y revisarse como parte del requerimiento de fase para establecer un contexto. Alternativamente, el equipo del proyecto podría identificar una lista simple de entidades, las cosas acerca de las cuales los miembros del equipo piensan que el sistema tendrá que capturar y almacenar datos.

Por desgracia, el modelado de datos raramente se asocia con la fase de análisis del problema del análisis de sistemas. Algunos analistas prefieren trazar modelos de procesos (capítulo 8) para documentar el sistema actual, pero muchos analistas informan que los modelos de datos son muy superiores por las siguientes razones:

- Los modelos de datos ayudan a los analistas a identificar rápidamente el vocabulario de negocios más completamente que los modelos de procesos.
- Los modelos de datos casi siempre se construyen más rápidamente que los modelos de procesos.
- Un modelo completo de datos puede caber en una sola hoja de papel. Los modelos de procesos a menudo requieren docenas de hojas de papel.
- Los modeladores de procesos frecuentemente y con demasiada facilidad se enredan en detalles innecesarios.
- Los modelos de datos para los sistemas existentes y propuestos son mucho más similares que los modelos de procesos para sistemas existentes y propuestos. Consecuentemente, hay menos trabajo que descartar a medida que usted se mueve hacia fases posteriores.

¡Estamos de acuerdo! Un modelo de fases de análisis de problemas incluye sólo entidades y relaciones, pero ningún atributo; se le llama **modelo de datos de contexto**. El intento es afinar nuestra comprensión del alcance, no entrar en detalles minuciosos acerca de las entidades y reglas de negocios. Muchas relaciones pueden ser no específicas.

Muchas herramientas automatizadas proveen la habilidad para leer los archivos de sistema y las bases de datos existentes y traducirlos en modelos “físicos” de datos. Estos modelos físicos de datos luego pueden transformarse en su modelo “lógico” equivalente de datos. Esta capacidad de traducción beneficia tanto al análisis del problema como a las fases de análisis de requerimientos.

El análisis de requerimientos resulta en un modelo lógico de datos que se desarrolla por etapas como sigue:

1. Empezamos por construir el *modelo de datos de contexto* para establecer el alcance del proyecto. Si ya se desarrolló un modelo de datos de contexto durante el análisis del problema, ese modelo puede revisarse para reflejar los requerimientos nuevos y el alcance del proyecto.

---

**modelo de datos de la aplicación** Modelo de datos para un sistema de información individual completo.

---

**modelo de datos de contexto** Modelo de datos que incluye entidades y relaciones pero no atributos.

**modelo de datos basado en claves** Modelo de datos que incluye entidades y relaciones con cardinalidades precisas que transforman las relaciones no específicas en entidades asociativas, y que también incluyen claves primarias y alternas.

**modelo integral de datos** Modelo de datos que incluye todas las entidades, los atributos, las relaciones, los criterios de subconjuntos y las cardinalidades precisas.

- Después, se traza un **modelo de datos basado en claves**. Este modelo eliminará las relaciones no específicas, agregará las entidades asociativas, e incluirá claves primarias y alternas. El modelo basado en claves también incluirá las cardinalidades precisas y cualesquiera jerarquías de generalización.
- En seguida, se construirá un **modelo integral de datos**. El modelo integral de datos incluye todos los atributos descriptivos restantes y todos los criterios de subconjuntos. Cada atributo se define en el repositorio con tipos de datos, dominios y acciones por omisión (en lo que algunas veces se llama un *modelo de datos totalmente descrito*).
- El modelo de datos terminado se analiza con respecto a la adaptabilidad y la flexibilidad a través de un proceso llamado *normalización*. El modelo final analizado se llama *modelo normalizado de datos*.

Este modelo de requerimientos de datos requiere un esfuerzo del equipo que incluye a los analistas de sistemas, usuarios, gerentes y analistas de datos. Un administrador de datos a menudo define estándares para todos los modelos de datos y aprueba estos.

Al final, durante la fase de análisis de decisiones, el modelo de datos se usará para tomar las decisiones sobre la implementación: la mejor manera de implementar los requerimientos con la tecnología de la base de datos. En la práctica, esta decisión ya pudo haber sido estandarizada como parte de una arquitectura de la base de datos. Por ejemplo, SoundStage ha definido como estándar dos sistemas de administración de base de datos: Microsoft *Access* para las bases de datos personal y de trabajo de grupo, y el *Servidor SQL* de Microsoft para las bases de datos de la empresa.

Por último, los modelos de datos no pueden construirse sin hechos apropiados e información suministrada por la comunidad de usuarios. Estos hechos pueden recolectarse a través de varias técnicas, tales como el muestreo de formas y archivos existentes, la investigación de sistemas similares, encuestas de los usuarios y de la gerencia, y las entrevistas de usuarios y la gerencia. El método más rápido para coleccionar hechos e información y la construcción y verificación simultánea de los modelos de datos es la planeación conjunta de requerimientos (JRP). En la técnica JRP se organiza cuidadosamente una reunión de grupo para recolectar los hechos, construir y verificar los modelos; generalmente en una o dos sesiones de día completo. Las técnicas de investigación de los hechos y de recopilación de información se presentaron en el capítulo 5. La tabla 7.4 resume algunas preguntas que pueden ser útiles para la investigación de los hechos y la recopilación de información en relación con el modelado de datos.

## > Mirando hacia delante en el diseño de sistemas

Durante el diseño de sistemas, el modelado lógico de los datos se transformará en un modelo físico de datos (llamado *esquema de la base de datos*) para el sistema de administración de base de datos elegido. Este modelo reflejará las limitaciones y capacidades técnicas de esa tecnología de bases de datos, así como también los requerimientos de mejora del desempeño propuestos por el administrador de la base de datos. Cualquier otra discusión adicional del diseño de bases de datos se difiere hasta el capítulo 12.

## > Las herramientas automatizadas para el modelado de datos

Los modelos de datos se almacenan en un repositorio. En cierto sentido, el modelo de datos está formado por **metadatos**; es decir, información acerca de los datos del negocio. La tecnología CASE (*Computer-Aided System Engineering*, Ingeniería de Software Asistida por Computadora), presentada en el capítulo 3, provee el repositorio para almacenar el modelo de datos y sus descripciones detalladas. La mayoría de los productos de CASE soportan el modelado de datos asistido por computadora y el diseño de la base de datos. Algunos productos de CASE (como *ERwin*, de Logic Works) sólo soportan el modelado de datos y el diseño de la base de datos. La herramienta CASE evita el trabajo pesado del trazo y mantenimiento de estos modelos y sus detalles subyacentes.

Al usar un producto CASE, usted fácilmente puede crear modelos de datos profesionales, accesibles sin el uso de papel, lápiz, gomas y plantillas. Los modelos pueden modificarse fácilmente para reflejar correcciones y los cambios propuestos por usuarios finales; y ¡usted no tiene que volver a empezar! También, la mayoría de productos CASE proveen herramientas analíticas poderosas que pueden comprobar sus modelos en cuanto a errores inadvertidos, de integridad y consistencia. Algunos productos de CASE hasta pueden

**metadatos** Datos acerca de los datos.

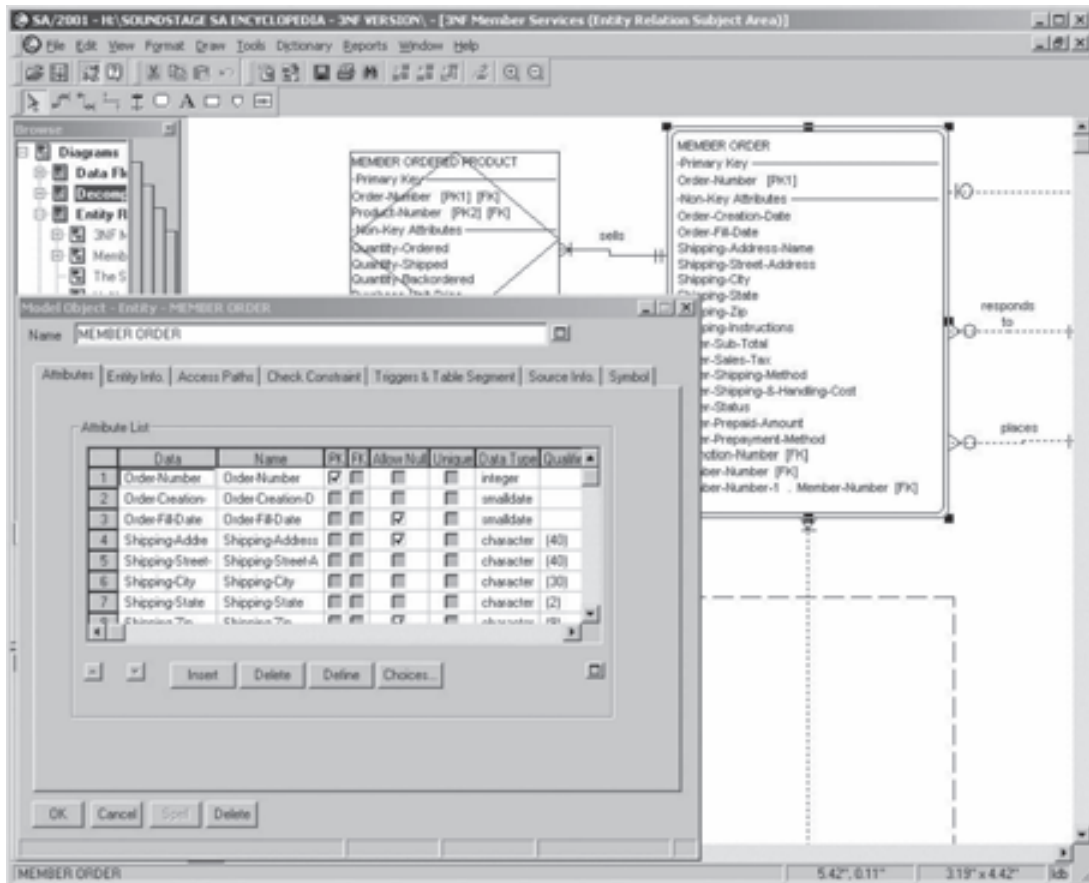
**TABLA 7.4 JRP y preguntas de entrevista para el modelado de datos**

Propósito	Ejemplos de preguntas
Descubrir las entidades del sistema	¿Cuáles son los sujetos de los negocios? En otras palabras: ¿Qué tipos de personas, organizaciones, unidades organizativas, lugares, cosas, materiales o eventos se usan o interactúan con este sistema y qué datos deben capturarse o mantenerse? ¿Cuántas instancias existen de cada sujeto?
Descubrir las claves de la entidad	¿Qué característica única (o características) distingue(n) a una instancia de cada sujeto de otras instancias del mismo sujeto? ¿Existe algún plan para cambiar este esquema de identificación en el futuro?
Descubrir los criterios al establecer los subconjuntos de la entidad	¿Hay algunas características de un sujeto que dividan a todas las instancias del sujeto en subconjuntos útiles? ¿Existe algún subconjunto de los sujetos anteriores para el cual usted no tenga una manera conveniente de agrupar las instancias?
Descubrir los atributos y los dominios	¿Qué características describen a cada sujeto? Para cada una de esas características: 1) ¿Qué tipo de datos se almacenan? 2) ¿Quién es responsable de definir los valores legítimos para los datos? 3) ¿Cuáles son los valores legítimos de los datos? 4) ¿Se requiere algún valor? 5) ¿Existe algún valor por omisión que deba asignarse si usted no lo especifica de otro modo?
Descubrir las necesidades de seguridad y de control	¿Hay algunas restricciones sobre quién puede ver o usar los datos? ¿A quién se le permite crear los datos? ¿A quién se le permite actualizar los datos? ¿A quién se le permite borrar los datos?
Descubrir las necesidades de cronometrado de los datos	¿Qué tan frecuentemente cambian los datos? ¿Para qué periodo tienen valor los datos para el negocio? ¿Por cuanto tiempo debemos conservar los datos? ¿Necesita datos o tendencias históricos?
Descubrir las jerarquías de generalización	¿Son iguales todas las instancias de cada sujeto? Es decir, ¿hay tipos especiales de cada sujeto que se describan o manejen en forma diferente? ¿Puede consolidarse cualquier parte de los datos para compartirse?
Descubrir las relaciones y los grados	¿Qué eventos ocurren que impliquen asociaciones entre sujetos? ¿Qué actividades o transacciones de negocios incluyen el manejo o cambio de los datos acerca de varios sujetos diferentes del mismo tipo o un tipo diferente?
Descubrir las cardinalidades	¿Se maneja de la misma manera cada actividad o evento del negocio, o hay circunstancias especiales? ¿Puede ocurrir un evento con solamente algunos de los sujetos asociados, o deben intervenir todos los sujetos?

ayudarle a analizar los modelos de datos en cuanto a consistencia, integridad y flexibilidad. Los ahorros potenciales de tiempo y calidad pueden ser sustanciales.

Como se mencionó anteriormente, algunas herramientas de CASE soportan la ingeniería inversa de las estructuras existentes de archivos y de bases de datos en los modelos de datos. Los modelos de datos resultantes representan modelos “físicos” de datos que pueden revisarse y someterse a un proceso de reingeniería para crear nuevos archivos o bases de datos, o puede traducirse a su modelo “lógico” equivalente. El modelo lógico de datos podría entonces editarse y someterse a un proceso de ingeniería hacia delante para convertirse en un modelo físico revisado de datos, y subsiguientemente en una implementación de archivo o de bases de datos.

Las herramientas CASE ciertamente tienen sus limitaciones. No todas las convenciones de modelado de datos son soportadas por todos los productos CASE. Y las diferentes herramientas de CASE adoptan notaciones ligeramente diferentes para los mismos métodos que



**FIGURA 7.12** Captura en pantalla de la herramienta CASE del System Architect

modelan datos. Por consiguiente, es muy probable que cualquier producto CASE dado pueda obligar a una compañía a adaptar sus símbolos de la metodología de modelado de datos o enfocarlo, a fin de que sea trabajada dentro de las limitaciones de la herramienta CASE.

Todos los modelos de datos SoundStage en la siguiente sección de este capítulo fueron creados con Popkin Systems y la herramienta del Software de CASE, *System Architect 2001*. Para el estudio del caso, le proveemos las salidas impresas exactamente como salieron de nuestras impresoras. Las únicas modificaciones fueron las viñetas que llaman la atención para aspectos específicos de interés en las salidas impresas. Las entidades, los atributos y las relaciones en los modelos de datos SoundStage fueron todos automáticamente catalogados en el repositorio de proyecto (al cual System Architect llama enciclopedia) del *System Architect*. La figura 7.12 ilustra algunas de las pantallas del *System Architect* tal como se usan para el modelado de datos.

## Cómo construir modelos de datos

Ahora usted sabe bastante acerca de modelos de datos para leerlos e interpretarlos. Pero como analista de sistemas o usuario final informado, usted debe aprender a construirlos. Usaremos el proyecto SoundStage Entertainment Club para enseñarle cómo construir modelos de datos.

NOTA: Este ejemplo le enseña a trazar el modelo de datos a partir de apuntes. En realidad, usted siempre debería buscar un modelo existente de datos. Si tales modelos existen, la administración de datos o grupo de la gerencia de datos generalmente los mantienen. Alternativamente, usted podría diseñar un modelo de datos a partir de la base de datos existente.

## > El descubrimiento de las entidades

La primera tarea en el modelado de datos es relativamente fácil. Usted necesita descubrir a las entidades fundamentales en el sistema, que son o podrían ser descritas por los datos. Usted no deberá restringir su examen a las entidades sobre las cuales los usuarios finales saben que quieren almacenar datos. Hay varias técnicas que pueden usarse para identificar las entidades:

- Durante las entrevistas o las sesiones JRP con los dueños y usuarios del sistema, preste atención a las palabras clave en su debate. Por ejemplo, durante una entrevista con una persona que discuta el ambiente de negocio y las actividades de SoundStage, un usuario puede afirmar: “tenemos que seguir la pista a todos nuestros miembros y sus acuerdos comprometidos”. Observe que las palabras clave en esta declaración son MIEMBROS y acuerdos. ¡Ambos son entidades!
- Durante las entrevistas o las sesiones JRP, específicamente pida a los dueños del sistema y a los usuarios que identifiquen cosas acerca de las cuales les gustaría capturar, almacenar y producir información. Estas cosas a menudo representan entidades que deben bosquejarse en el modelo de datos.
- Otra técnica para identificar las entidades es estudiar las formas, los archivos e informes existentes. Algunas formas identifican entidades de eventos. Los ejemplos incluyen ÓRDENES, SOLICITUDES, PAGOS, REPOSITORIOS, y así sucesivamente. Pero la mayoría de estas mismas formas también contienen datos que describen otras entidades. Considere una forma de inscripción usada en el sistema de inscripción escolar de su escuela. Una INSCRIPCIÓN es por sí misma una entidad de evento. Pero la forma promedio de inscripción también contiene datos que describen otras entidades, tales como el ESTUDIANTE (una persona), los CURSOS (que son conceptos), los INSTRUCTORES (otras personas), el CONSEJERO (otra persona más), las DIVISIONES (otro concepto), y así sucesivamente. El estudio de los archivos de computadora, de las bases de datos o de las salidas del sistema informatizado de inscripción también podría descubrir estas mismas entidades.
- Si las narrativas de caso de uso (práctico) se han escrito durante la fase de análisis de requerimientos, pueden ser una fuente de atributos de datos y de las entidades. Escudriñe cada narrativa de caso práctico buscando sustantivos. Cada sustantivo es una entidad o atributo de datos potencial. Usted tendrá que realizar la manipulación de la lista resultante de sustantivos porque no todos ellos serán atributos o entidades. Algunos serán referencias para los usuarios u otros sistemas de información. Algunos serán referencias a cosas que son parte de la interfaz del usuario, no los datos. Algunos serán sinónimos de otros atributos o las entidades en otro sitio de la lista, y usted no querría duplicarlos.
- La tecnología también le puede ayudar a identificar las entidades. Algunas herramientas de CASE pueden usarse para diseñar modelos físicos de datos a partir de archivos y bases de datos existentes. Generalmente el analista debe limpiar el modelo resultante reemplazando los nombres físicos, los códigos y los comentarios con sus equivalentes lógicos, amigables con los negocios.

Aun cuando estas técnicas pueden resultar útiles para identificar las entidades, ocasionalmente le gustarán bromas. Una verificación de calidad simple y rápida puede eliminar las entidades falsas. Pida a su usuario que especifique el número de instancias de cada entidad. Una entidad verdadera tiene múltiples instancias; ¡docenas, centenares, miles, o más! Si no, la entidad no existe.

A medida que las entidades son descubiertas déles nombres simples, significativos, orientados a los negocios. Las entidades deberán nombrarse con sustantivos que describan a la persona, al acontecimiento, lugar, objeto o cosa acerca de la cual queremos almacenar datos. No trate de abreviar o usar acrónimos. Los nombres deberán ser en singular a fin de distinguir el concepto lógico de la entidad de las instancias reales de la entidad. Los nombres pueden incluir cláusulas o adjetivos apropiados para mejor describir a la entidad: por ejemplo, una ORDEN DEL CLIENTE externamente generada debe ser distinguida de una ORDEN DE EXISTENCIAS internamente generada.

Defina cada entidad en términos de negocios. No en términos técnicos, ni como “datos acerca de...” Intente esto: use un diccionario para crear una definición en borrador y luego hágala a la medida para el negocio actual. Sus nombres y definiciones de la entidad deberán establecer un glosario inicial de terminología de negocios que les servirá tanto a usted como a los usuarios y analistas futuros por años.

**TABLA 7.5** Entidades fundamentales para el proyecto SoundStage

Nombre de la entidad	Definición de negocios
ACUERDO	Un contrato mediante el cual un miembro acepta comprar un cierto número de productos un cierto tiempo. Después de cumplir ese acuerdo, el miembro se hace elegible para créditos de bonificación que pueden redimirse por productos gratis o con descuento.
MIEMBRO	Un miembro activo de uno o más clubes. Nota: Un objetivo clave del sistema es volver a reclutar a miembros inactivos en vez de borrarlos.
ORDEN DE UN MIEMBRO	Una orden generada por un miembro como parte de una promoción mensual, o una orden iniciada por un miembro. Nota: El sistema actual solamente soporta órdenes generadas a partir de las promociones; sin embargo, a las órdenes iniciadas por los clientes se les ha dado una alta prioridad como una opción adicional del sistema propuesto.
TRANSACCIÓN	Un evento del negocio al cual debe responder el sistema de servicios al miembro.
PRODUCTO	Un producto inventariado disponible para su promoción y venta a otros miembros. Nota: Los objetivos de la mejora del sistema incluyen: 1) la compatibilidad con un nuevo sistema de código de barras que está siendo desarrollado para el almacén y 2) la adaptabilidad a una mezcla de productos que cambia rápidamente.
PROMOCIÓN	Un evento mensual o trimestral en el cual se ofrece a los miembros ofertas especiales de productos.

Nuestra gerencia y usuarios de SoundStage inicialmente identificaron las entidades listadas en la tabla 7.5. Observe cómo las definiciones contribuyen a establecer el vocabulario del sistema.

### > El modelo de datos de contexto

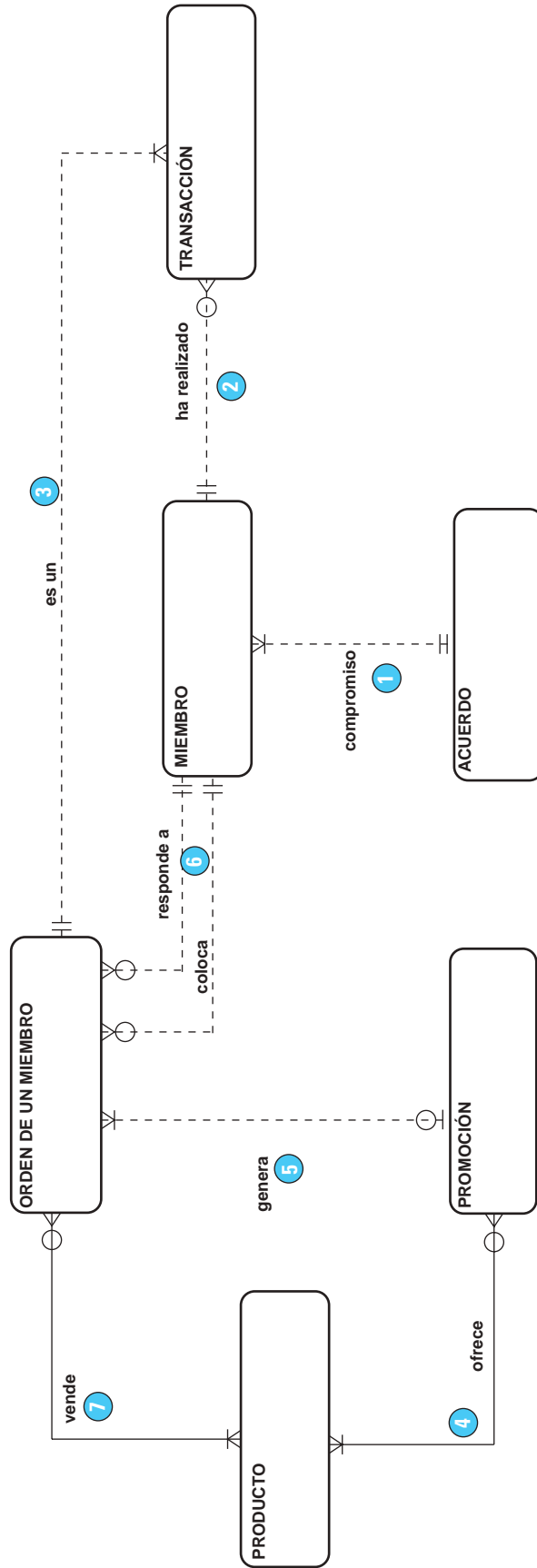
La siguiente tarea en el modelado de datos es construir el modelo de datos de contexto. Este modelo deberá incluir las entidades fundamentales de negocios que fueron previamente descubiertas, así como sus relaciones naturales.

Las relaciones deberán nombrarse con expresiones que, cuando se combinen con los nombres de entidad, formen oraciones o afirmaciones simples de negocios. Algunas herramientas de CASE, como el *System Architect*, le permiten nombrar las relaciones en ambas direcciones. De otra manera, siempre nombre la relación de padre a hijo.

Hemos completado esta tarea en la figura 7.13. Ésta representa un modelo de datos creado en el *System Architect*. Una vez que empezamos a mapear los atributos pueden salir a la superficie relaciones y entidades nuevas. Los siguientes números establecen referencias para los mismos números de la figura 7.13. El ERD comunica lo siguiente:

- 1 Un ACUERDO obliga a uno o más MIEMBROS. Aun cuando las relaciones pueden nombrarse en sólo una dirección (del padre al hijo), la otra dirección está implícita. Por ejemplo, es implícito que un MIEMBROS está obligado a uno y solamente un ACUERDO.
- 2 Un MIEMBROS ha realizado cero, una o más TRANSACCIONES. Implícitamente, una TRANSACCIÓN dada fue realizada por uno y solamente un MIEMBRO.
- 3 La ORDEN de un MIEMBROS es una TRANSACCIÓN. De hecho, una ORDEN DEL MIEMBRO dada puede corresponder a muchas TRANSACCIONES (por ejemplo, una orden de un miembro nuevo, una orden cancelada de un miembro, una orden cambiada de un miembro, etcétera). Pero una TRANSACCIÓN dada puede o no puede representar una ORDEN de un MIEMBRO.





**FIGURA 7.13** Modelo de datos de contexto de SoundStage

- 4 Una PROMOCIÓN ofrece uno o más PRODUCTOS. Implícitamente, un PRODUCTO se ofrece en cero, una o más PROMOCIONES. Por ejemplo, un CD de música que atrae tanto a audiencias de country/western como de rock ligero podría presentarse en la promoción para ambos. Como los productos exceden en número grandemente a las promociones, la mayoría de los productos nunca se presenta en una promoción.
- 5 Una PROMOCIÓN genera muchas ÓRDENES de los MIEMBROS. Implícitamente, una ORDEN de un MIEMBRO se genera para cero o una PROMOCIÓN. ¿Por qué cero? En el sistema nuevo, un miembro podrá iniciar su propia orden.
- 6 Es permisible que exista más de una relación entre las mismas dos entidades si las relaciones separadas comunican eventos o asociaciones diferentes de negocios. Así, un MIEMBRO responde a cero, una o más ÓRDENES de un MIEMBRO. Esta relación soporta las órdenes iniciadas por los miembros. Un MIEMBRO coloca cero, una o más ÓRDENES de un MIEMBRO. Esta relación soporta las órdenes iniciadas por los miembros. En ambos casos, una ORDEN de un MIEMBRO es colocada por (le responde) exactamente un MIEMBRO. Aunque no lo necesitamos para esta relación doble, algunas herramientas CASE (incluyendo el *System Architect*) proveen un símbolo para registrar las relaciones Booleanas (tales como CONDICIONAL, INCLUYENTE). Así, para cualesquiera dos relaciones, podría usarse un símbolo Booleano para establecer que las instancias de las relaciones deben ser mutuamente excluyentes (= INCLUYENTE) o mutuamente contingentes (= CONDICIONAL).
- 7 La ORDEN de un MIEMBRO vende uno o más PRODUCTOS. Implícitamente, un PRODUCTO se vende sobre cero, una o más ÓRDENES de un MIEMBRO. Observe que ésta es una relación no específica, lo cual se resolverá más tarde.

Si usted leyó con todo cuidado cada uno de los incisos anteriores, probablemente aprendió mucho acerca del sistema SoundStage. Los modelos de datos se han hecho cada vez más populares como una herramienta para describir el contexto del negocio para los proyectos de sistemas.

## > El modelo de datos basado en claves

La siguiente tarea es identificar las claves de cada entidad. Se sugieren los siguientes lineamientos para las claves:<sup>1</sup>

1. El valor de una clave no deberá cambiar durante la duración de vida de cada instancia de entidad. Por ejemplo, el NOMBRE sería una clave deficiente, ya que el apellido de una persona pudo cambiar por el matrimonio o por divorciarse.
2. El valor de una clave no puede ser nulo.
3. Deben instalarse controles para asegurar que el valor de una clave es válido. Esto puede lograrse definiendo con precisión el dominio y usando los controles de validación del sistema de administración de base de datos para implementar ese dominio.
4. Algunos expertos (Bruce) sugieren que usted evite las **claves inteligentes**. Una clave inteligente es un código del negocio cuya estructura comunica datos acerca de una instancia de entidad (como su clasificación, el tamaño u otras propiedades). Un código es un grupo de caracteres y/o dígitos que identifican y describen algo en el sistema de negocios. Algunos expertos argumentan que porque esas características pueden cambiar se viola la regla 1 anteriormente citada.

Disentimos. Los códigos del negocio pueden otorgar un valor a la organización porque pueden ser procesados rápidamente por humanos sin la asistencia de una computadora.

- a. Hay varios tipos de códigos. Pueden combinarse para formar medios efectivos para la identificación de la instancia de entidad.
  - (1) Los *códigos de serie* asignan números generados secuencialmente a las instancias de entidad. Muchos sistemas de administración de base de datos pueden generar y restringir los códigos de serie a los requerimientos de un negocio.
  - (2) Los *códigos de bloque* son similares a los códigos seriales, excepto que los números del bloque están divididos en grupos que tienen algún significado de

---

**clave inteligente** Código de negocios cuya estructura comunica datos acerca de una instancia de entidad.

<sup>1</sup> Adaptado de Thomas A. Bruce, *Designing Quality Databases with IDEF1X Information Models*. Derechos de autor © 1992 por Thomas A. Bruce. Reproducido con autorización de Dorset House Publishing, 353 W. 12th, Nueva York, NY 10014 (212-620-4053/1-800-DH-BOOKS/www.dorsethouse.com). Todos los derechos reservados.

negocios. Por ejemplo, un proveedor de la televisión de satélite podría asignar 100-199 como canales de PAGO POR EVENTO, 200-299 como canales de cable, 300-399 como canales de DEPORTES, 400-499 como canales de PROGRAMACIÓN PARA ADULTOS, 500-599 como canales de SOLAMENTE MÚSICA, 600-699 como canales de JUEGOS INTERACTIVOS, 700-799 como canales de INTERNET, 800-899 como canales de cable PREMIUM, y 900-999 como canales de PELÍCULAS Y EVENTOS PREMIUM.

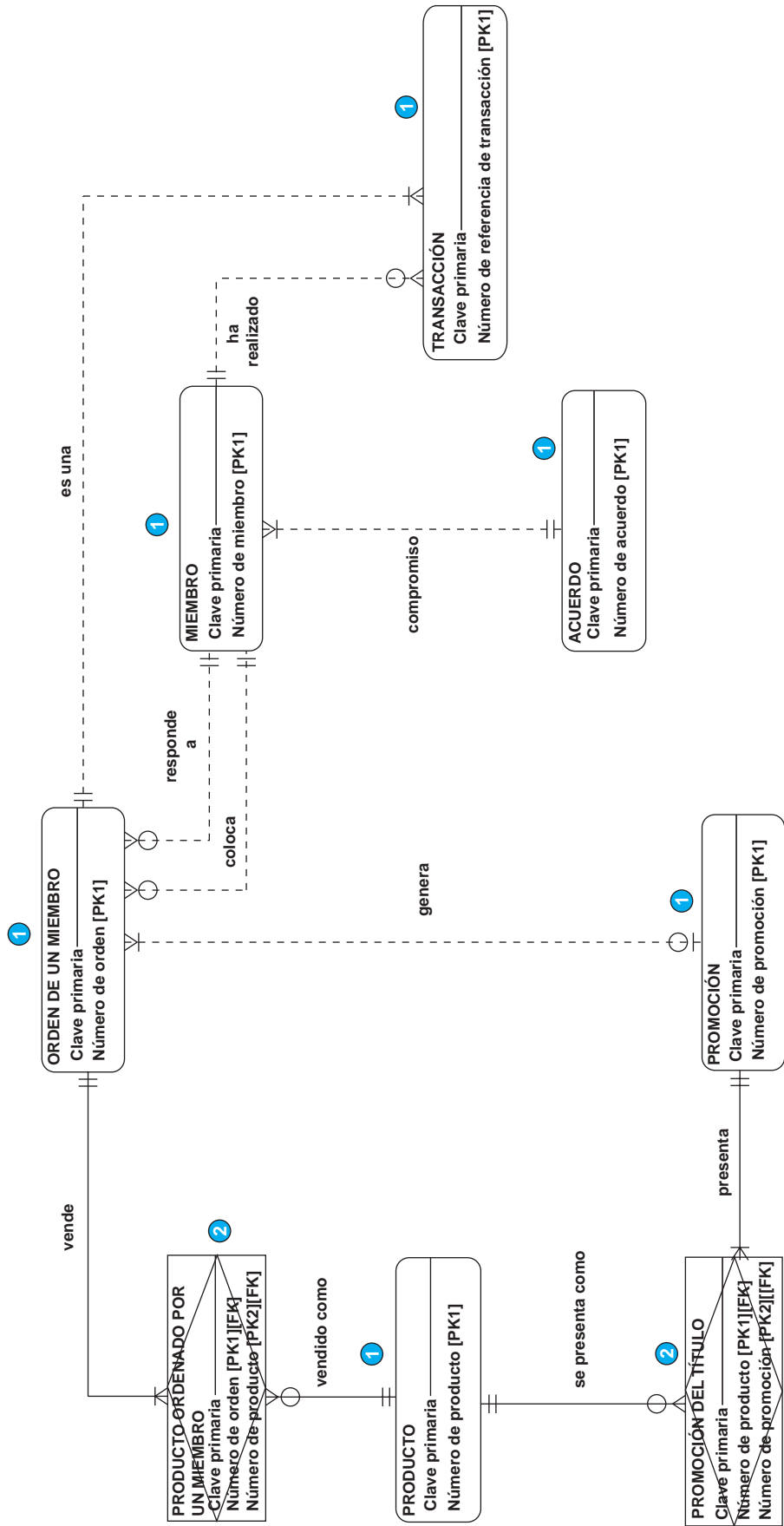
- (3) Los *códigos alfabéticos* usan combinaciones finitas de letras (y posiblemente números) para describir instancias de entidad. Por ejemplo, cada ESTADO tiene un código alfabético único de dos caracteres. Los códigos alfabéticos generalmente deben combinarse con códigos de serie o de bloque para identificar de manera única las instancias de la mayoría de entidades.
  - (4) En los *códigos de posición significativa*, cada dígito o grupo de dígitos describe una característica mensurable o identificable de la instancia de entidad. Los códigos de dígitos significativos se usan frecuentemente para codificar partidas del inventario. Los códigos que usted ve en llantas y bombillas (focos) son ejemplos de códigos significativos de posición. Nos dicen sobre características como el tamaño de la llanta y la capacidad en vatios, respectivamente.
  - (5) Los *códigos jerárquicos* proveen una interpretación de arriba a abajo para una instancia de entidad. Cada artículo cifrado se factoriza en grupos, subgrupos y así sucesivamente. Por ejemplo, podríamos codificar posiciones del empleado como sigue:
    - El primer dígito identifica clasificación (oficinista, profesor, etc.).
    - El segundo y tercer dígitos indican el nivel dentro de la clasificación.
    - El cuarto y quinto dígitos indican el calendario de empleo.
- b. Se sugieren los siguientes lineamientos al crear un esquema de codificación de negocios:
- (1) Los códigos deberían ser extensibles para permitir el crecimiento.
  - (2) El código completo debe dar como resultado un valor único para cada instancia de entidad.
  - (3) Los códigos deberán ser bastante grandes para describir las características distintivas pero suficientemente pequeños como para ser interpretados por personas *sin una computadora*.
  - (4) Los códigos deberán ser convenientes. Una instancia nueva deberá ser fácil de crear.
5. Considere inventar una clave representativa en lugar de sustituir las largas claves concatenadas de entidades independientes. Esta sugerencia no es práctica para entidades asociativas, porque cada parte de la clave concatenada es una clave foránea que debe corresponder precisamente a la clave primaria de su entidad padre.

La figura 7.14 es el modelo de datos basado en claves para el proyecto SoundStage. Observe que la clave primaria se especifica para cada entidad.

- 1 Muchas entidades tienen una clave primaria simple, de un solo atributo.
- 2 Resolvimos la relación no específica entre ORDEN de un MIEMBRO y PRODUCTO introduciendo la entidad asociativa PRODUCTO ORDENADO por un MIEMBRO. Cada instancia asociativa de entidad representa un producto para la orden de un miembro. Las entidades padre contribuyeron sus propias claves primarias para contener la clave concatenada de la entidad asociativa. El *System Architect* coloca a un “PK1” junto al NÚMERO DE ORDEN para señalar que es “la parte uno” de la clave primaria concatenada y un “PK2” junto al NÚMERO DEL PRODUCTO para señalar que es “la parte dos” de la clave concatenada. También observe que cada atributo en esa clave concatenada, por sí mismo, es una clave foránea que apunta de regreso a la instancia correcta de entidad padre.

Asimismo, la relación no específica entre PRODUCTO y PROMOCIÓN se resolvió usando una entidad asociativa, PROMOCIÓN DEL TÍTULO, que también hereda las claves de las entidades padre.

Al desarrollar este modelo, cuide un par de cosas. Si usted no puede definir claves para una entidad, puede ser que la entidad realmente no exista; es decir, las ocurrencias múltiples de la así llamada entidad no existen. Por tanto, la asignación de claves es una buena verificación de la calidad antes de atribuir completamente el modelo de datos. Además, si dos o más entidades tienen claves idénticas, son con toda certeza la misma entidad.



**FIGURA 7.14** El modelo de datos de SoundStage basado en claves

## > Las jerarquías generalizadas

En este momento sería útil identificar cualesquiera jerarquías de generalización en el dominio de los negocios. El proyecto SoundStage, al principio de este capítulo, identificó al menos una estructura de supertipo/subtipo. Las discusiones siguientes revelaron una jerarquía de generalización. Así, nuestro modelo basado en claves fue revisado como se muestra en la figura 7.15. Tuvimos que diseñar el modelo de una manera algo diferente debido a la jerarquía; sin embargo, las relaciones y claves que se definieron previamente han sido retenidas. Llamamos su atención a lo siguiente:

- ❶ La herramienta CASE SoundStage automáticamente dibuja una caja con línea segmentada arrojada alrededor de una jerarquía de generalización.
- ❷ Los subtipos heredan las claves de los supertipos.
- ❸ Desconectamos la PROMOCIÓN de PRODUCTO como se mostró anteriormente y la reconectamos al subtipo TÍTULO. Esto se hizo para establecer con exactitud la regla de negocios de que la MERCANCÍA nunca es exhibida en una PROMOCIÓN, solamente los TÍTULOS.

## > El modelo integral de datos

Puede parecer una tarea trivial identificar los atributos de datos restantes; sin embargo, los analistas no familiarizados con el modelado de datos frecuentemente encuentran problemas. Para lograr esta tarea, usted debe tener una comprensión completa de los atributos de datos para el sistema. Estos hechos pueden descubrirse usando enfoques de arriba hacia abajo (como la lluvia de ideas) o enfoques de abajo hacia arriba (como el muestreo de formas y archivos). Si existe un modelo de datos de la empresa, algunos (quizá muchos) de los atributos tal vez ya hayan sido identificados y registrados en un repositorio.

Se ofrecen los siguientes lineamientos para la atribución:

- Muchas organizaciones tienen estándares para asignar nombres y abreviaciones aprobadas. El administrador de datos generalmente mantiene tales estándares.
- Escoja cuidadosamente los nombres de los atributos. Muchos atributos comparten nombres de base comunes tales como NOMBRE, DIRECCIÓN, FECHA. A menos que los atributos puedan generalizarse en un supertipo, es mejor dar a cada variación un nombre único, como:

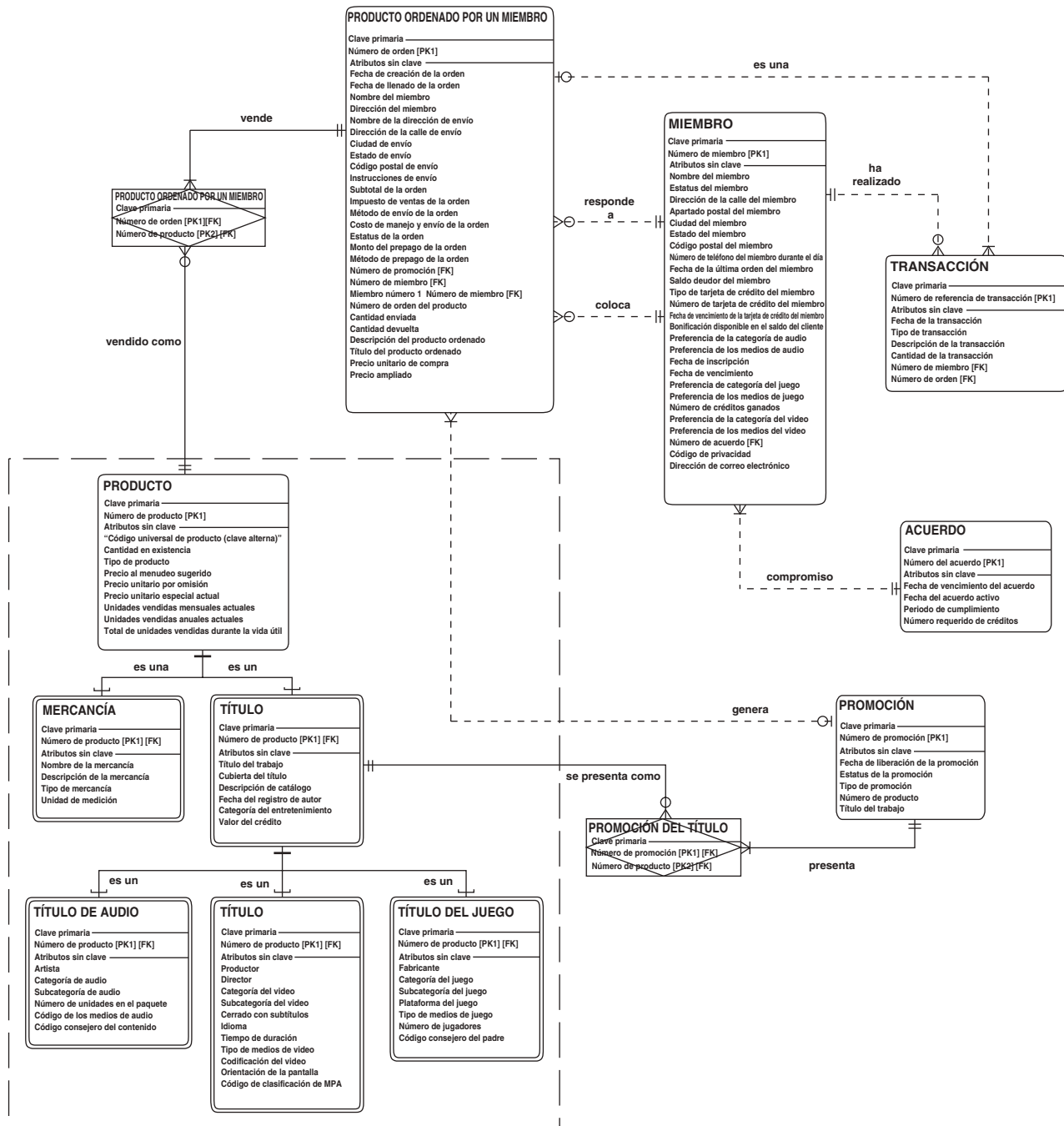
NOMBRE DEL CLIENTE	DIRECCIÓN DEL CLIENTE	FECHA DE ORDEN
NOMBRE DEL PROVEEDOR	DIRECCIÓN DEL PROVEEDOR	FECHA DE FACTURA
NOMBRE DEL EMPLEADO	DIRECCIÓN DEL EMPLEADO	FECHA DE VUELO

También recuerde que un proyecto no está aislado de otros proyectos, pasados o futuros. Los nombres deben ser distinguibles entre los proyectos.

Algunas organizaciones mantienen plantillas reusables, globales para estos atributos de base comunes. Esto promueve tipos coherentes de datos, dominios y acciones por omisión entre todas las aplicaciones.

- Los nombres de los atributos físicos en las formas y reportes existentes frecuentemente se abrevian para ahorrar espacio. Los nombres de los atributos lógicos deberán ser más evidentes; por ejemplo, traducir el atributo CCE del formato de orden de pedido a su equivalente lógico: CANTIDAD A COBRAR CONTRA ENTREGA; traducir CO a CANTIDAD ORDENADA, y así sucesivamente.
- Muchos atributos adoptan sólo valores SÍ o NO. Intente nombrar estos atributos como preguntas. Por ejemplo, el de nombre del atributo ¿CANDIDATO PARA GRADUARSE?, sugiere que los valores son SÍ y NO.
  - Cada atributo deberá mapearse a sólo una entidad. Si un atributo verdaderamente describe entidades diferentes, probablemente contiene varios atributos diferentes. Dé a cada uno un nombre único.
- Las claves foráneas son la excepción para la regla de no redundancia; identifican las instancias asociadas de entidades relacionadas.





**FIGURA 7.16** El modelo integral de datos SoundStage

- El dominio de un atributo no deberá basarse en la lógica. Por ejemplo, en el caso SoundStage aprendimos que los valores de MEDIOS dependen del tipo de producto. Si el tipo de producto es un video, los medios podrían ser una película VHS, una película de 8 mm, un disco de láser, o un DVD. Si el tipo del producto es audio, los medios podrían ser una cinta de casete, CD o MD. La mejor solución sería asignar atributos separados a cada dominio: MEDIOS AUDIOVISUALES y MEDIOS DE VIDEO.

La figura 7.16 suministra el mapeo de atributos de datos a las entidades para la fase de definición de nuestro proyecto de sistemas SoundStage. Mientras el modelo totalmente atribuido identifica todos los atributos que deben capturarse y almacenarse en nuestra





## > El análisis de datos

La técnica usada para mejorar un modelo de datos como preparación para el diseño de la base de datos se llama *análisis de datos*. El **análisis de datos** es un proceso que prepara un modelo de datos para su implementación como una base de datos simple, no redundante, flexible y adaptable. La técnica específica se llama *normalización*. La **normalización** es una técnica de análisis de datos que organiza los atributos de datos de modo que se agrupan para formar entidades no redundantes, estables, flexibles y adaptables. La normalización es una técnica de tres pasos que coloca el modelo de datos en la primera forma normal, la segunda forma normal y la tercera forma normal.<sup>2</sup> No se enrede con la terminología: es más fácil de lo que parece. Por ahora, establezcamos una comprensión inicial de estos tres formatos:

- Dicho simplemente, una entidad está en la **primera forma normal (1FN)** si no hay atributos que puedan tener más de un valor para una instancia individual de la entidad. Cualesquiera atributos que puedan tener valores múltiples realmente describen una entidad separada, posiblemente una entidad y relación.
- Una entidad está en la **segunda forma normal (2FN)** si está ya en 1FN y si los valores de todos los atributos de la clave no primaria son dependientes de la clave totalmente primaria; no sólo de parte de ésta. Cualesquiera atributos sin clave que dependen sólo de parte de la clave primaria deberán moverse a cualquier entidad donde esa clave parcial es de hecho la clave completa. Esto puede requerir crear una relación y una entidad nuevas en el modelo.
- Una entidad está en **tercera forma normal (3FN)** si está ya en 2FN y si los valores de sus atributos de la clave no primaria no dependen de cualesquiera otros atributos de la clave no primaria. Cualesquiera atributos sin clave que dependen de otros atributos sin clave deben moverse o borrarse. Otra vez, las entidades y las relaciones nuevas tendrían que añadirse al modelo de datos.

## > Ejemplo de normalización

Hay numerosos enfoques de la normalización. Hemos elegido presentar un enfoque no teórico y no matemático. Dejaremos la teoría, el álgebra relacional y las implicaciones detalladas para los cursos de base de datos y los libros de texto.

Como siempre, usaremos el estudio de caso de SoundStage para demostrar los pasos. Empecemos haciendo referencia al modelo integral de datos que fue desarrollado anteriormente (véase la figura 7.16). ¿Es un modelo de datos normalizado? No. Identifiquemos los problemas y recorramos los pasos para normalizar nuestro modelo de datos.

**Primera Forma Normal** El primer paso en el análisis de datos es colocar a cada entidad dentro de 1FN. En la figura 7.16, ¿cuáles entidades no están en 1FN?

Usted debería encontrar dos: ORDEN DE UN MIEMBRO y PROMOCIÓN. Cada uno contiene un *grupo repetitivo*, es decir, un grupo de atributos que pueden tener valores múltiples para una sola instancia de la entidad {denotada por los corchetes}. Estos atributos se repiten muchas veces “como un grupo”. Considere, por ejemplo, la entidad ORDEN de un MIEMBRO. Una sola ORDEN de un MIEMBRO puede contener muchos productos; por consiguiente, los atributos NÚMERO DEL PRODUCTO ORDENADO, DESCRIPCIÓN DEL PRODUCTO ORDENADO, TÍTULO DEL PRODUCTO ORDENADO, CANTIDAD ORDENADA, CANTIDAD ENVIADA, CANTIDAD DEVUELTA, PRECIO UNITARIO DE COMPRA, y el PRECIO EXTENDIDO, pueden (y probablemente lo hacen) repetirse para cada instancia de la ORDEN de un MIEMBRO.

De modo semejante, como una PROMOCIÓN puede presentar más que un TÍTULO del PRODUCTO, los atributos NÚMERO del PRODUCTO y TÍTULO DEL TRABAJO pueden repetirse. ¿Cómo fijamos estas anomalías en nuestro modelo?

Las figuras 7.17 y 7.18 muestran cómo colocar estas dos entidades dentro de 1FN. La entidad original se bosqueja en el lado izquierdo de la página. Las entidades de 1FN están en el lado derecho de la página. Cada figura muestra cómo la normalización cambió el

---

**análisis de datos** Técnica que se usa para mejorar un modelo de datos para su implementación como una base de datos.

---

**normalización** Técnica de análisis de datos que organiza los datos en grupos para formar entidades no redundantes, estables, flexibles y adaptables.

---

**primera forma normal (1FN)** Entidad cuyos atributos no tienen más que un valor para una instancia individual de esa entidad.

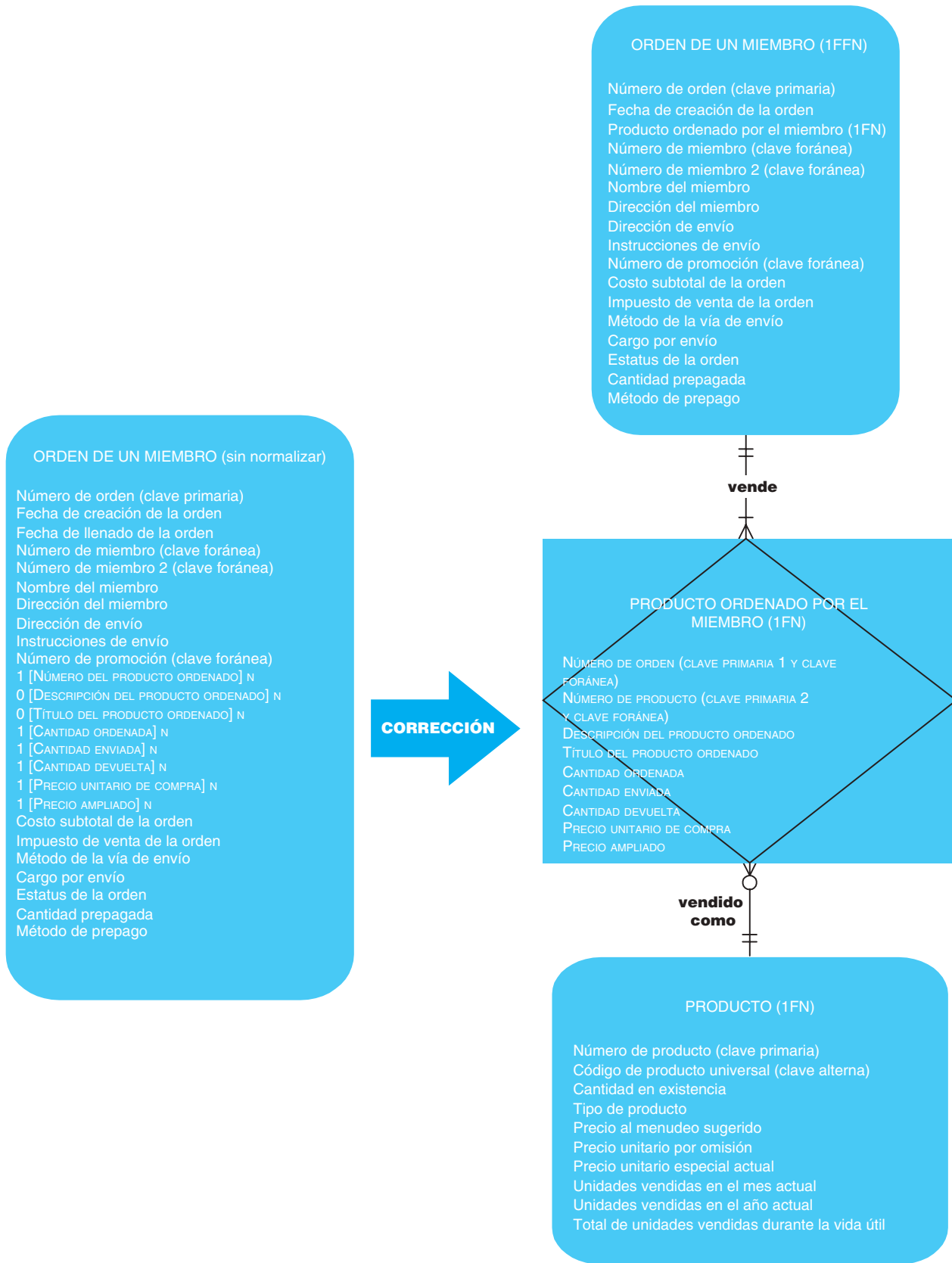
---

**segunda forma normal (2FN)** Entidad cuyos atributos de clave no primaria dependen de la clave primaria completa.

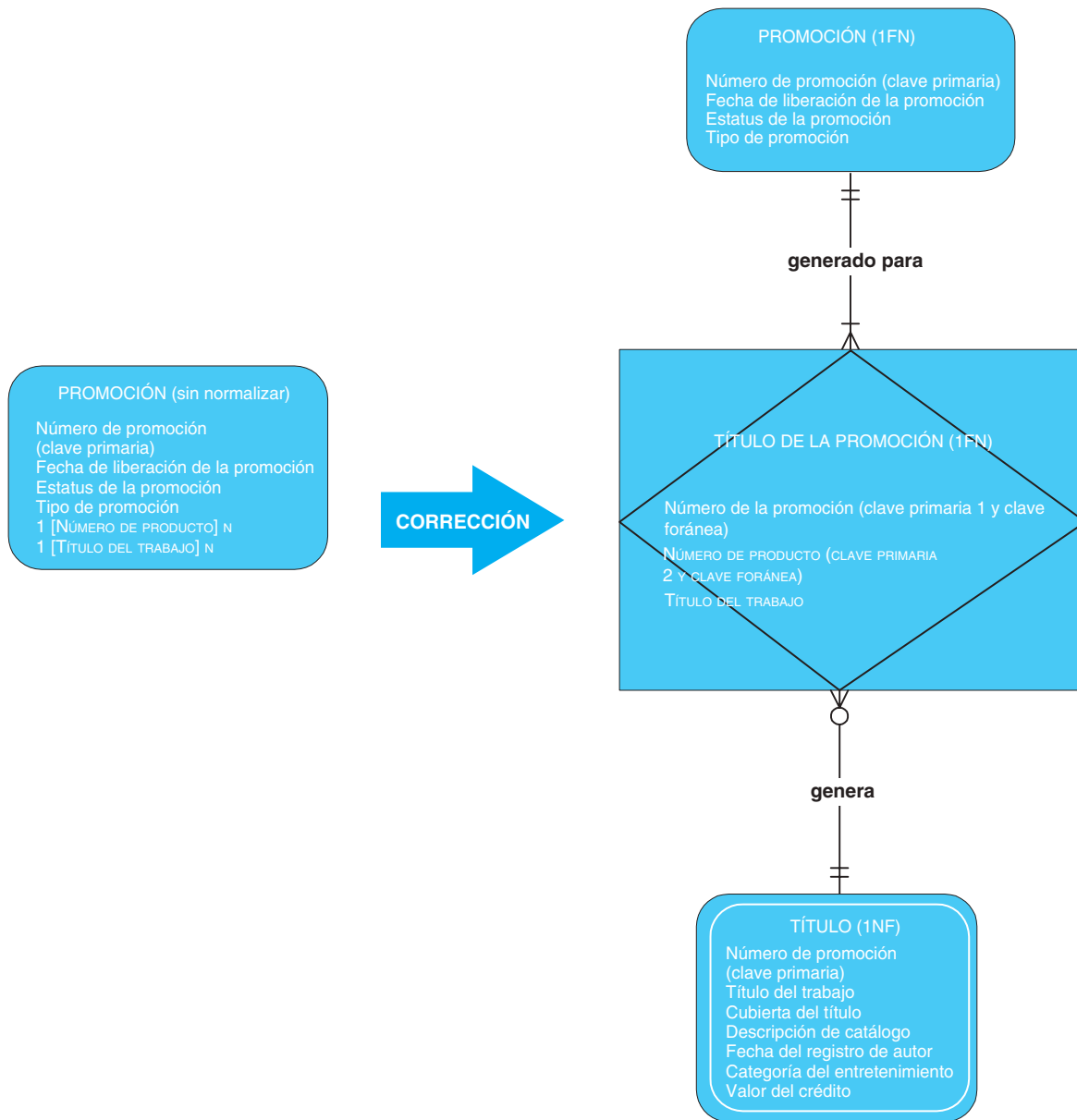
---

**tercera forma normal (3FN)** Entidad cuyos atributos clave no primarios no dependen de ningunos otros atributos con clave primaria.

<sup>2</sup> Los expertos en bases de datos han identificado formas normales adicionales. La tercera forma normal elimina la mayoría de las anomalías de los datos. Dejamos la discusión de las formas normales avanzadas a los libros y cursos sobre bases de datos.



**FIGURA 7.17** Primera forma normal (1FN)



**FIGURA 7.18** Primera forma normal (1FN)

modelo de datos y las asignaciones de atributos. Para su conveniencia, los atributos que son afectados están en negritas y en letras versalitas (mayúsculas pequeñas).

En la figura 7.17, primero retiramos los atributos que pueden tener más de un valor para una instancia de la entidad ORDEN de un MIEMBRO. Solamente eso coloca a ORDEN de un MIEMBRO en 1FN. ¿Pero qué hacemos con los atributos retirados? No los podemos retirar totalmente del modelo: ¡son parte de los requerimientos de negocios! Por consiguiente, movimos el grupo entero de atributos a una entidad nueva, el PRODUCTO ORDENADO POR UN MIEMBRO. Cada instancia de estos atributos describe a un PRODUCTO en una sola ORDEN de un MIEMBRO. Así, si una orden específica contiene cinco PRODUCTOS, habrá cinco instancias de la entidad PRODUCTO ORDENADO POR EL MIEMBRO. Cada instancia de entidad tiene sólo un valor para cada atributo; por consiguiente, la entidad nueva está también en la primera forma normal.

Otro ejemplo de 1FN se muestra en la figura 7.18 para la entidad PROMOCIÓN. Como antes, movimos los atributos repetitivos a una entidad diferente, PROMOCIÓN DEL TÍTULO.

Todas las demás entidades están ya en 1FN porque no contienen ningunos grupos repetitivos.

**Segunda Forma Normal** El siguiente paso del análisis de datos es colocar a las entidades dentro de 2FN. Recuerde que hace falta que usted ya haya colocado todas las entidades dentro de 1FN. También recuerde que el 2FN busca un atributo cuyo valor es determinado sólo por parte de la clave primaria; no la clave completa concatenada. Consecuentemente, las entidades que tienen una clave primaria de un solo atributo ya están en 2FN. Eso se encarga de producto (y sus subtipos), orden de un miembro, miembro, promoción, acuerdo y transacción. Así, necesitamos comprobar sólo esas entidades que tienen una clave concatenada: producto ordenado por un miembro y promoción del título.

Primero verifiquemos la entidad PRODUCTO ORDENADO POR EL MIEMBRO. La mayor parte de los atributos dependen de la clave primaria completa. Por ejemplo, la CANTIDAD ORDENADA tiene poco sentido a menos que usted tenga *tanto* un NÚMERO DE ORDEN *como* un NÚMERO DEL PRODUCTO. ¡Piense acerca de eso! Por sí mismo, el NÚMERO DE ORDEN es inadecuado, ya que la orden podría tener tantas cantidades ordenadas como haya productos en la orden. De modo semejante, por sí mismo, el NÚMERO DEL PRODUCTO es inadecuado, ya que el mismo producto podría aparecer en muchas órdenes. Así, la CANTIDAD ORDENADA requiere ambas partes de la clave y depende de la clave completa. Lo mismo podría decirse de CANTIDAD ENVIADA, CANTIDAD DEVUELTA, PRECIO UNITARIO DE COMPRA y PRECIO AMPLIADO.

Pero, ¿qué hay acerca de la DESCRIPCIÓN DEL PRODUCTO ORDENADO y del TÍTULO DEL PRODUCTO ORDENADO? ¿Necesitamos nosotros en realidad el NÚMERO DE ORDEN para determinar el valor de cualquiera de los dos? ¡No! En lugar de eso, los valores de estos atributos dependen sólo del valor del NÚMERO DEL PRODUCTO. Así, los atributos no dependen de la clave completa; hemos descubierto una anomalía de *dependencia parcial* que debe arreglarse. ¿Cómo arreglamos este tipo de error de normalización?

Refiérase a la figura 7.19 en la siguiente página. Para arreglar el problema, simplemente movemos los atributos sin clave, la DESCRIPCIÓN DEL PRODUCTO ORDENADO y el TÍTULO DEL PRODUCTO ORDENADO, a una entidad que tiene sólo el NÚMERO DEL PRODUCTO como su clave. Si es necesario, tendríamos que crear esta entidad, pero ya existe la entidad PRODUCTO con esa clave. Pero debemos tener cuidado porque el PRODUCTO es un supertipo. En la inspección de los subtipos descubrimos que los atributos están ya en las entidades MERCANCÍA y TÍTULO, si bien bajo un sinónimo. Así, realmente no tuvimos que mover los atributos de la entidad del PRODUCTO ORDENADO POR UN MIEMBRO, simplemente los borramos como atributos redundantes.

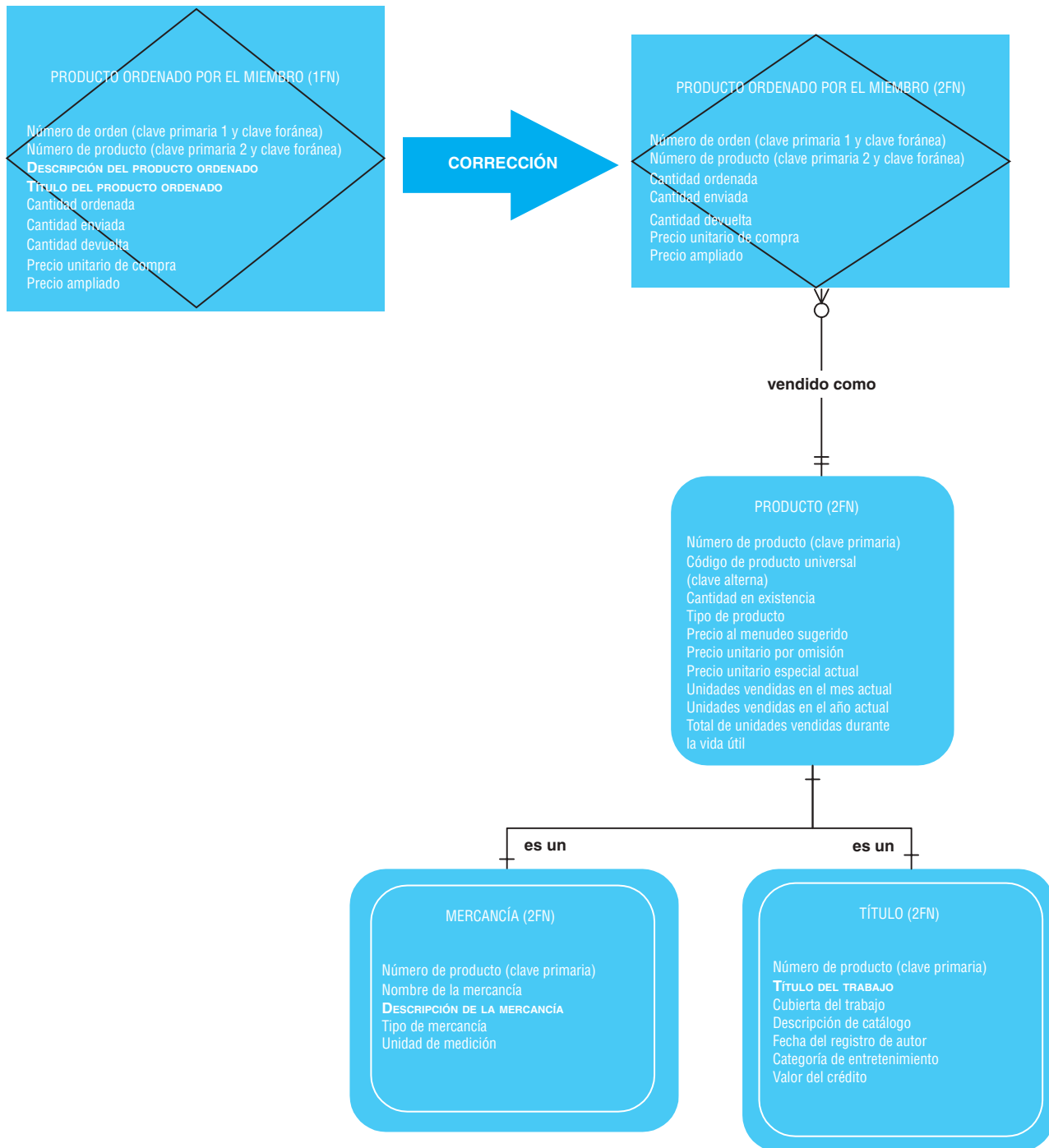
Entonces, examinemos la entidad de PROMOCIÓN DEL TÍTULO. La clave concatenada es la combinación del NÚMERO DE PROMOCIÓN con el NÚMERO DEL PRODUCTO. El TÍTULO DE TRABAJO depende de la parte NÚMERO DEL PRODUCTO de la clave concatenada. Así, el TÍTULO DE TRABAJO se retira de la PROMOCIÓN DEL TÍTULO (véase la figura 7.20). Observe que el TÍTULO DE TRABAJO ya existía en la entidad del TÍTULO, que tiene un número del producto como su clave primaria.

**Tercera Forma Normal** Podemos simplificar aún más nuestras entidades colocándolas dentro de 3FN. Se requiere que las entidades estén en 2FN antes de iniciar el análisis de 3FN. El tercer análisis en forma normal contempla dos tipos de problemas, los *datos derivados* y las *dependencias transitivas*. En ambos casos el error fundamental es que los atributos sin clave dependen de otros atributos sin clave.

El primer tipo de análisis 3FN es fácil: examina cada entidad para los atributos derivados. Los **atributos derivados** son aquellos cuyos valores pueden calcularse a partir de otros atributos o derivarse a través de la lógica de los valores de otros atributos. Si usted lo piensa bien, almacenar un atributo derivado tiene poco sentido. Primero, desperdicia espacio de almacenamiento en disco. Segundo, complica lo que deberían ser actualizaciones simples. ¿Por qué? Cada vez que usted cambia los atributos de base debe recordar volver a realizar el cálculo y también cambiar su resultado.

Por ejemplo, examine la entidad del PRODUCTO ORDENADO por el MIEMBRO en la figura 7.21. El atributo PRECIO AMPLIADO se calcula multiplicando la CANTIDAD ORDENADA por el PRECIO UNITARIO DE COMPRA. Así, el PRECIO AMPLIADO (un atributo sin clave) no depende de la clave

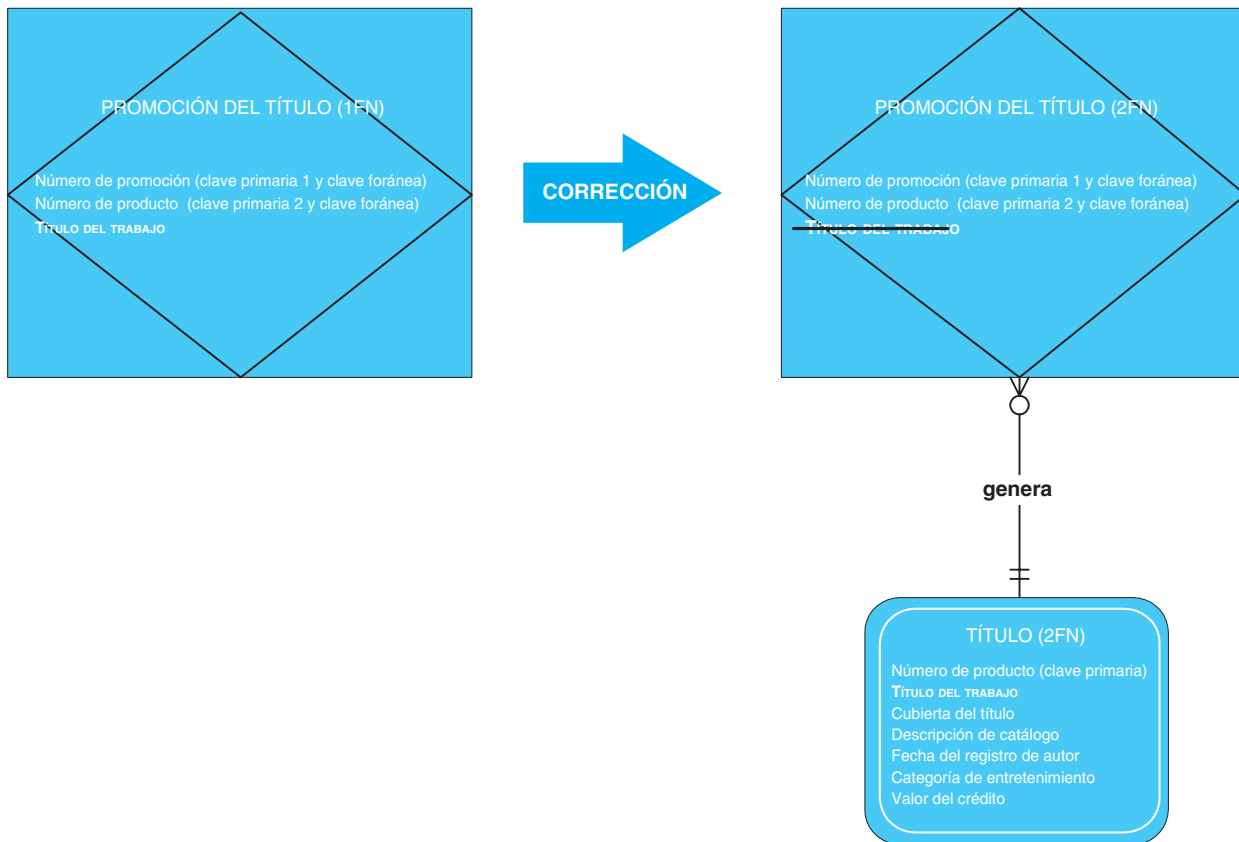
**atributo derivado** Atributo cuyo valor puede calcularse de otros atributos o derivarse de los valores de otros atributos.



**FIGURA 7.19** Segunda forma normal (2FN)

primaria tanto como depende de los atributos sin clave, la CANTIDAD ORDENADA y el PRECIO UNITARIO DE COMPRA. Así, corregimos la entidad suprimiendo el PRECIO AMPLIADO.

Suena simple, ¿no es cierto? Pues bien, ¡no siempre! Hay desacuerdo de hasta dónde puede llegar con esta regla. Algunos expertos argumentan que la regla debería aplicarse sólo dentro de una entidad única. Así, estos expertos no suprimirían un atributo derivado, si los atributos requiriesen que la derivación se asignara a entidades *diferentes*. Estamos de acuerdo basándonos en el argumento de que un atributo derivado que involucra a las entidades múltiples presenta un mayor peligro para la falta de consistencia de los datos

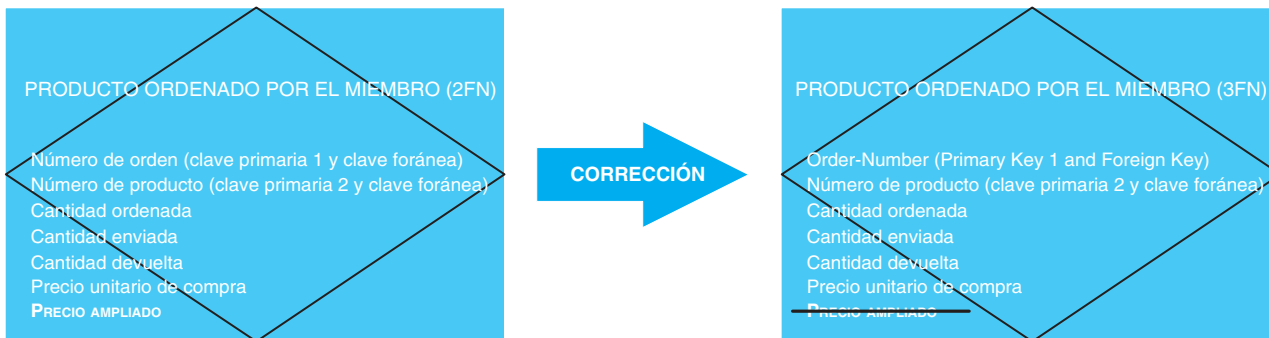


**FIGURA 7.20** Segunda forma normal (2FN)

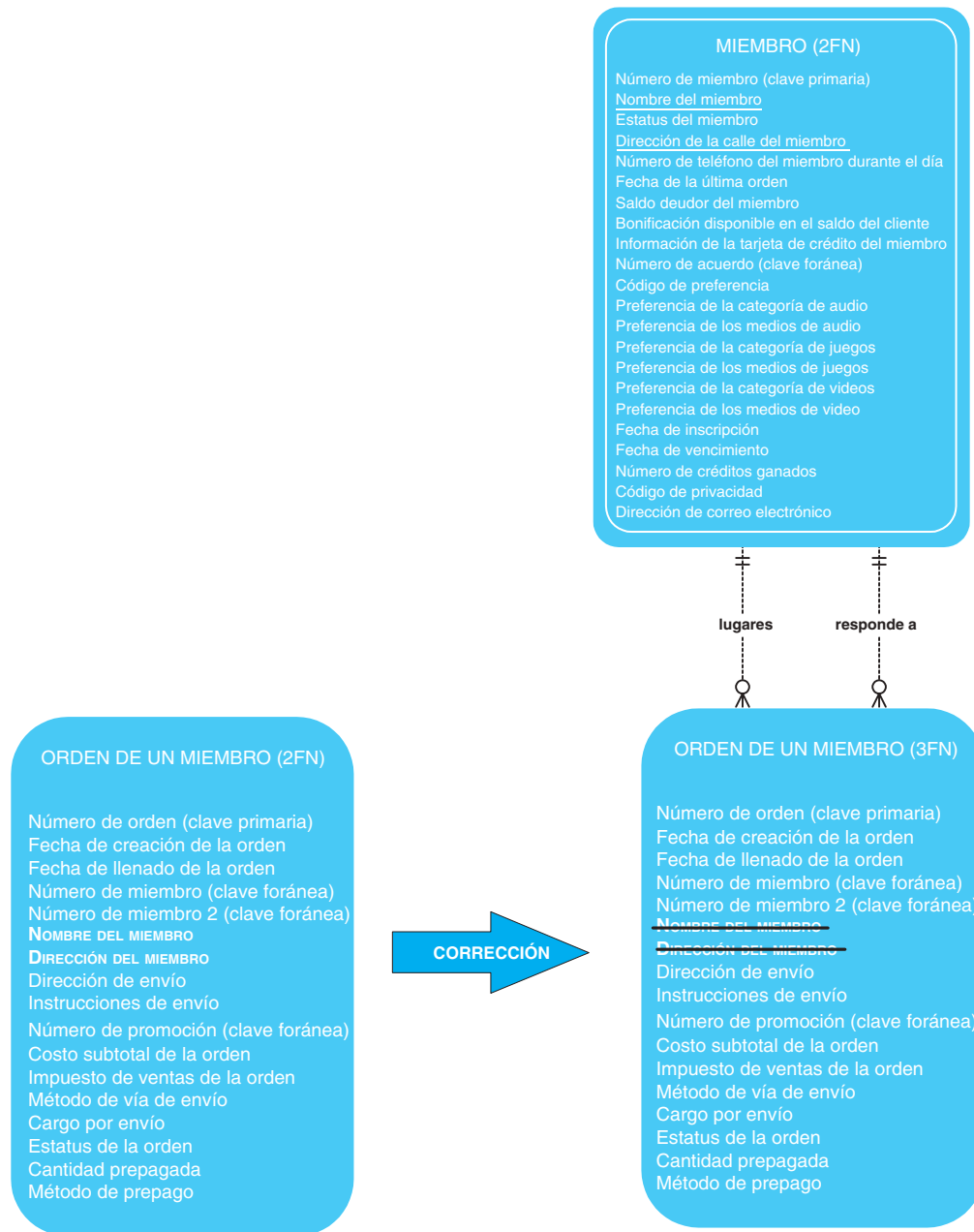
causada al actualizar un atributo en una entidad y olvidando subsiguientemente actualizar el atributo derivado en otra entidad.

**dependencia transitiva** Cuando el valor de un atributo que no es clave depende del valor de otro atributo que no es clave por otro medio que no sea la derivación.

Otra forma de análisis del 3FN revisa en busca de *dependencias transitivas*. Una **dependencia transitiva** existe cuando un atributo sin clave depende de otro atributo sin clave (diferente de la derivación). Este error generalmente señala que una entidad sin descubrir está todavía incrustada dentro de la entidad del problema. Si no se corrige tal condición, puede causar problemas futuros de adaptabilidad y flexibilidad, si un requerimiento nuevo finalmente requiere que implementemos esa entidad sin descubrir como una tabla separada de la base de datos.



**FIGURA 7.21** Tercera forma normal (3FN)



**FIGURA 7.22** Tercera forma normal (3FN)

El análisis transitivo se realiza sólo en entidades que no tienen una clave concatenada. En nuestro ejemplo, esto incluye PRODUCTO, ORDEN DE UN MIEMBRO, PROMOCIÓN, ACUERDO, MIEMBRO y TRANSACCIÓN. Para la entidad PRODUCTO, todos los atributos sin clave dependen de la clave primaria y sólo la clave primaria. Así, el PRODUCTO está ya en la tercera forma normal. Un análisis similar de PROMOCIÓN, ACUERDO y TRANSACCIÓN revelan que también están en la tercera forma normal.

Pero examine la entidad ORDEN DEL MIEMBRO de la figura 7.22. En particular, los atributos NOMBRE DEL MIEMBRO y DIRECCIÓN DEL MIEMBRO. ¿Estos atributos dependen de la clave primaria, NÚMERO DE ORDEN? ¡No! La clave primaria NÚMERO DE ORDEN de ningún modo determina el valor de NOMBRE DEL MIEMBRO y de DIRECCIÓN DEL MIEMBRO. Por otra parte, los valores de NOMBRE DEL MIEMBRO y de la DIRECCIÓN DEL MIEMBRO dependen del valor de otra clave no *primaria* en la entidad: NÚMERO DEL MIEMBRO.

¿Cómo arreglamos este problema? El NOMBRE DEL MIEMBRO y la DIRECCIÓN DEL MIEMBRO necesitan moverse de la entidad ORDEN DEL MIEMBRO a una entidad cuya clave primaria es simplemente NÚMERO DEL MIEMBRO. Si es necesario, crearíamos esa entidad, pero en nuestro caso tenemos ya una entidad MIEMBRO con la clave primaria requerida. Tal como resulta, no necesitamos realmente mover los atributos del problema porque ya están asignados a la entidad MIEMBRO. Nosotros, sin embargo, observamos que la DIRECCIÓN DEL MIEMBRO era un sinónimo de la DIRECCIÓN DE LA CALLE DEL MIEMBRO. Elegimos conservar el término más reciente en MIEMBRO.

Existen varias formas normales más allá de 3FN. Cada forma normal sucesiva hace al modelo de datos más simple, menos redundante y más flexible. Sin embargo, los analistas de sistemas (y la mayoría de expertos en bases de datos) raramente llevan los modelos de datos más allá de 3FN. Consecuentemente, dejaremos el debate adicional sobre la normalización a los libros de texto sobre bases de datos.

Las primeras ocasiones en que usted normaliza un modelo de datos, el proceso aparecerá lento y tedioso. Sin embargo, con el tiempo y la práctica, se vuelve rápido y rutinario. Muchos modeladores experimentados reducen significativamente el esfuerzo y tiempo de modelado al hacer la normalización durante la asignación de atributos (pueden hacer la normalización en el momento en que desarrollan el modelo integral de datos). Siempre puede ayudar recordar el siguiente estribillo (se ignora la fuente), que adecuadamente resume la primera, la segunda y la tercera formas normales:

Se dice que una entidad está en la tercera forma normal si cada atributo de clave no primaria depende de la clave primaria, la clave primaria completa, y nada más que la clave primaria.

**La simplificación por inspección** La normalización es un proceso bastante mecánico. Pero depende de nombrar consistencias en el modelo original de datos (antes de la normalización). Si varios analistas trabajan en una aplicación común, no es raro crear problemas que no serán contemplados por la normalización. Estos problemas se resuelven de la mejor manera a través de la simplificación por inspección, un proceso en donde una entidad de datos en 3FN se simplifica aún más por esfuerzos tales como encarar la redundancia sutil de los datos.

El modelo de datos final, normalizado, se replantea en la figura 7.23 en la siguiente página.

**El soporte CASE para la normalización** Muchas herramientas CASE aducen que soportan los conceptos de normalización. Leen el modelo de datos y tratan de aislar los errores posibles de normalización. Al examinarlas detenidamente, la mayoría de las herramientas CASE pueden normalizar sólo hasta la primera forma normal. Logran esto en una de dos formas. Buscan relaciones muchos a muchos y transforman esas relaciones en entidades asociativas. O buscan atributos específicamente descritos, como que tengan valores múltiples para una instancia de entidad individual. (Podría argumentarse que el analista debería haber reconocido eso como un error del 1FN y no debería haber descrito los atributos como tales.)

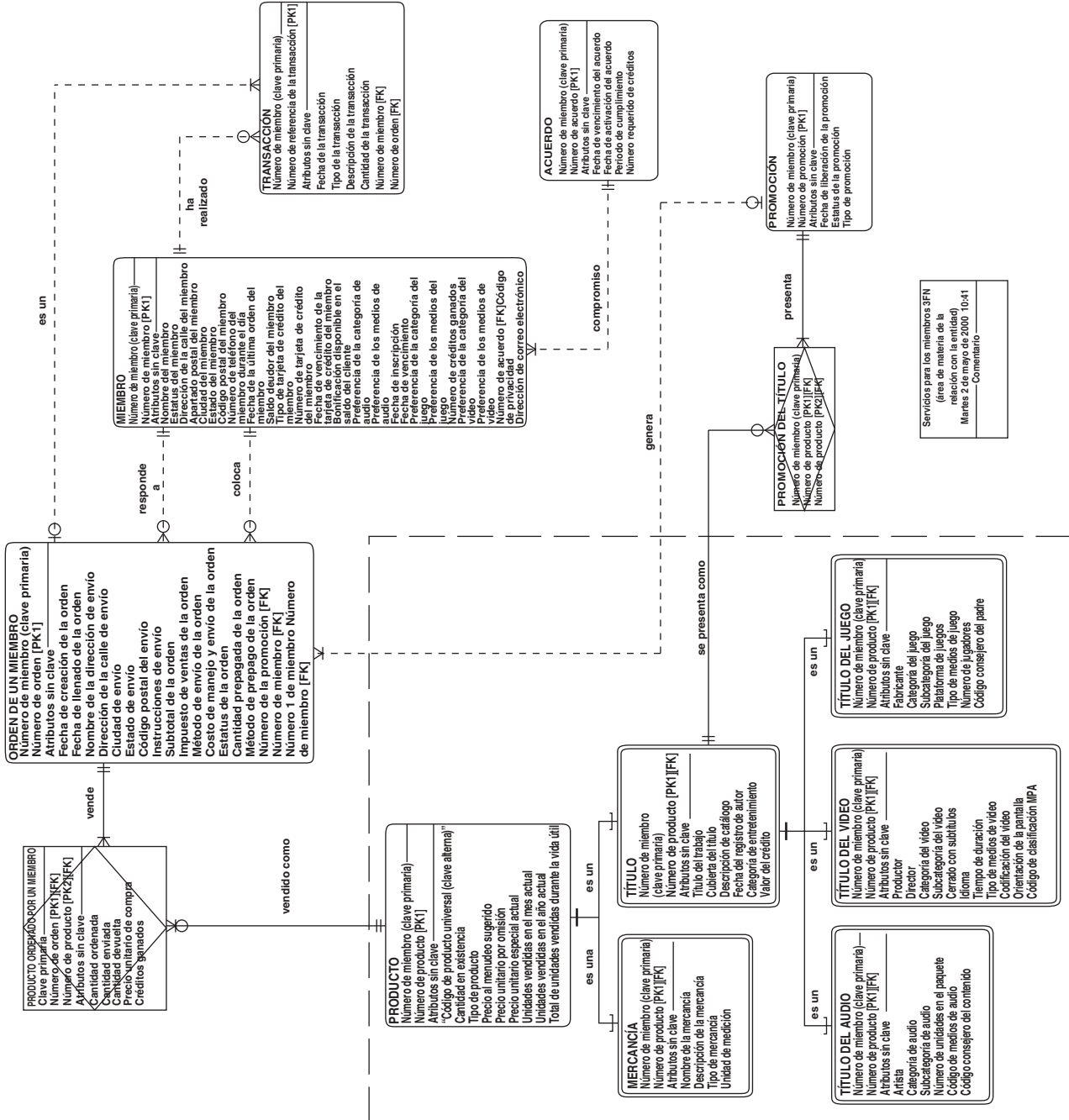
Es en extremo difícil para una herramienta CASE identificar los errores de la segunda y tercera formas normales. Eso requeriría que la herramienta CASE tuviera la inteligencia para reconocer dependencias parciales y transitivas. En realidad, tales dependencias pueden ser descubiertas sólo a través de exámenes menores que los de rutina por los analistas de sistemas o los expertos en bases de datos.

## Cómo mapear los requerimientos de datos a las localidades

Aun cuando un modelo lógico de datos es efectivo para describir qué datos deben guardarse para un sistema nuevo, no comunica los requerimientos sobre la base de una localidad de negocios en operación. Necesitamos identificar qué datos y derechos de acceso se necesitan en cuáles posiciones. Específicamente, podríamos formular las siguientes preguntas de negocios:

- ¿Cuáles subconjuntos de las entidades y los atributos son necesarios para realizar el trabajo en cada posición?
- ¿Qué nivel de acceso se requiere?





**FIGURA 7.23** Modelo de datos lógicos de SoundStage en la tercera forma normal (3FN)



Entidad.Atributo	Ubicación	Ciudad	Marketing	Publicidad	Almacén	Ventas	A/R	Boston	Ventas	Almacén	San Francisco	Ventas	San Diego	Almacén
<b>Cliente</b>	INDV					TDS	TDS		SC	SC		SC		SC
.Número de cliente	L				L	CLAB	L		CLAB	L		CLAB		L
.Nombre del cliente	LA				L	CLAB	L		CLAB	L		CLAB		L
.Dirección del cliente	LA				L	CLAB	L		CLAB	L		CLAB		L
.Clasificación de crédito del cliente	X					L	LA		L			L		
.Saldo deudor del cliente	L					L	LA		L			L		
<b>Orden</b>	INDV		TDS		SC	TDS			SC	SC		SC		SC
.Número de orden	ELB		L	CLAB	L	CLAB	L		CLAB	L		CLAB		L
.Fecha de la orden	ELB		L	CLAB	L	CLAB	L		CLAB	L		CLAB		L
.Cantidad de la orden	ELB		L	CLAB		CLAB	L		CLAB	L		CLAB		L
<b>Producto ordenado</b>	INDV		TDS		SC	TDS			SC	SC		SC		SC
.Cantidad ordenada	EAB		L	CLAB	L	CLAB	L		CLAB			CLAB		
.Precio unitario del artículo ordenado	EAB		L	CLAB		CLAB	L		CLAB			CLAB		
<b>Producto</b>	TDS		TDS	TDS	TDS	TDS			TDS	TDS		TDS		TDS
.Número del producto	L		CLAB	L	L	L			L	L		L		L
.Nombre del producto	L		CLAB	L	L	L			L	L		L		L
.Descripción del producto	L		CLAB	LA	L	L			L	L		L		L
.Unidad de medición del producto	L		CLAB	L	L	L			L	L		L		L
.Precio unitario actual del producto	L		CLAB	L		L			L	L		L		L
.Cantidad en la mano del producto	X				LA	L			L	LA		L		LA
	INDV = individual			TDS = TODOS		SC = Subconjunto	X = sin acceso							
	E = entregar	C = crear		L = leer		A = actualizar	B = borrar							

**FIGURA 7.24** Matriz CLAB de datos de posición

- ¿Puede *crear* la posición instancias de la entidad?
- ¿Puede *leer* la posición instancias de la entidad?
- ¿Puede *suprimir* la posición instancias de la entidad?
- ¿Puede *actualizar* la posición instancias existentes de la entidad?

**matriz CLAB de datos de posición** Matriz que se usa para mapear los requerimientos de datos en las posiciones.

Los analistas de sistemas han encontrado útil definir estos requerimientos lógicos en forma de una *matriz CLAB de datos de posición*. Una **matriz CLAB de datos de posición** es una tabla en la cual las filas indican las entidades (y los atributos posibles); las columnas indican posiciones, y las celdas (la intersección de filas y columnas) documentan niveles de acceso, donde C = crear, L = leer, A = actualizar o modificar, y B = borrar. La figura 7.24 ilustra una matriz CLAB de datos de posición. La decisión de incluir o no incluir atributos se basa en si es necesario que las posiciones se restrinjan en lo que se refiere a cuáles atributos se puede tener acceso. La figura 7.24 también muestra la habilidad para documentar que una posición requiere acceso sólo para un subconjunto (llamado SC) de instancias de entidad. Por ejemplo, cada oficina de ventas necesita tener acceso sólo a los clientes en su región.

En algunas metodologías y herramientas CASE, usted puede definir *vistas* de los modelos de datos para cada posición. Una vista incluye sólo las entidades y atributos a los que se puede tener acceso para una posición individual. Si se definen las vistas, también deben mantenerse en sincronía con el modelo maestro de datos. (La mayoría de las herramientas CASE hacen esto automáticamente.)

La mayoría de ustedes irán directamente al capítulo 8, “Modelado de procesos”. Mientras que el modelado de datos se ocupa de los datos independientemente de cómo se capturan y se usan esos datos (datos estáticos), el modelado de procesos muestra cómo se capturarán y se usarán los datos (datos dinámicos).

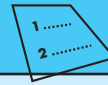
Si usted quiere aprender inmediatamente cómo implementar los modelos de datos como bases de datos, deberá revisar o leer el capítulo 12, “Diseño de bases de datos”. En ese capítulo los modelos lógicos de datos se transforman en esquemas físicos de bases de datos. Con las herramientas CASE, el código para crear la base de datos puede generarse automáticamente.



## Resumen

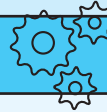
1. El modelado de datos es una técnica para organizar y documentar los DATOS de un sistema. Algunas veces se llama al modelado de datos “modelado de bases de datos”, porque un modelo de datos generalmente se implementa como una base de datos.
2. Hay varias notaciones para el modelado de datos. Frecuentemente, al modelo real se le llama ERD (*Entity Relationship Diagram*, Diagrama de Entidad Relación) porque bosqueja los datos en términos de las entidades y relaciones descritas por los datos.
3. Una entidad es algo acerca de lo cual el negocio necesita almacenar datos. Las clases de entidades incluyen las personas, los lugares, los objetos, los eventos y los conceptos.
4. Una instancia de entidad es una ocurrencia individual de una clase de entidad.
5. Las piezas de datos que queremos almacenar sobre cada instancia de una entidad dada se llaman atributos. Un atributo es una característica o propiedad descriptiva de la entidad. Algunos atributos pueden agruparse lógicamente en superatributos llamados atributos compuestos.
6. Al analizar un sistema, debemos definir los valores de un atributo que sean legítimos o que tengan sentido en los negocios. Los valores de cada atributo se definen en términos de tres propiedades: tipo de dato, dominio del dato y datos por omisión:
  - a) Los tipos de datos determinan qué clase de datos pueden almacenarse en el atributo.
  - b) El dominio de un atributo define qué valores puede adoptar legítimamente un atributo.
  - c) El valor por omisión de un atributo es el valor que será registrado si no es especificado por el usuario.
7. Cada entidad debe tener un identificador o clave. Una clave es un atributo, o un grupo de atributos, que asume un valor único para cada instancia de entidad.
  - a) Un grupo de atributos que identifica en forma única a una instancia de una entidad se llama clave concatenada.
  - b) Una clave candidato es un “candidato para convertirse en el identificador primario” de las instancias de una entidad.
  - c) Una clave primaria es la clave candidata que más comúnmente se usará para identificar en forma única una instancia de entidad individual.

- d) Cualquier clave candidata que no se seleccione para convertirse en clave primaria se designa clave alterna.
- e) Algunas veces también es necesario identificar un subconjunto de instancias en contraposición con una instancia individual. Un criterio de subconjuntos es un atributo (o atributo concatenado) cuyos valores finitos dividen a todas las instancias de entidad en subconjuntos útiles.
8. Una relación es una asociación natural de negocios que existe entre uno o más entidades. La relación puede representar un evento que enlaza a las entidades o meramente una afinidad lógica que existe entre las entidades. Todas las relaciones son implícitamente bidireccionales, es decir, pueden interpretarse en ambas direcciones.
  9. La cardinalidad define los números mínimo y máximo de las ocurrencias de una entidad para una ocurrencia individual de la entidad relacionada. Y como todas las relaciones son bidireccionales, la cardinalidad debe definirse en ambas direcciones para cada relación.
  10. El grado de una relación es el número de clases de entidad que participan en la relación. No todas las relaciones son binarias. Algunas relaciones pueden ser relaciones recursivas, en donde existe una relación entre las diferentes instancias de la misma entidad. Las relaciones también pueden existir entre más de dos entidades diferentes, como en el caso de una relación de tercer orden o ternaria.
  11. Una entidad asociativa es una entidad que hereda su clave primaria de más que alguna otra entidad (los padres). Cada parte de esa clave concatenada apunta a una instancia y solamente una de cada una de las entidades de conexión.
  12. Una clave foránea es una clave primaria de una entidad que aporta (se duplica) a otra entidad para identificar las instancias de una relación. Una clave foránea (siempre en una entidad hijo) siempre corresponde a la clave primaria (en una entidad padre).
  13. Las relaciones sin identificación son aquéllas en las que cada una de las entidades participantes tiene su propia clave primaria independiente, de las cuales ninguno de los atributos de la clave primaria es compartido. Las entidades en una relación sin identificación se denominan entidades *fuertes* o independientes porque ninguna de ellas depende de ninguna otra entidad para su identificación. Las relaciones de identificación son aquéllas en las cuales la entidad padre aporta su clave primaria para formar parte de la clave primaria de la entidad hijo. La entidad hijo de cualquier relación de identificación se llama entidad *débil* porque su identificación depende de la existencia de la entidad padre.
  14. Una relación no específica (o relación de muchos a muchos) es una en la cual se asocian muchas instancias de una entidad con muchas instancias de otra entidad. Tales relaciones son adecuadas sólo para los modelos preliminares de datos y deberán resolverse tan rápido como sea posible.
  15. La generalización es un enfoque que trata de descubrir y sacar provecho de las comunales entre entidades. Es una técnica en donde los atributos se agrupan para formar supertipos y subtipos de entidades.
    - a) Un supertipo de entidad es una entidad cuyas instancias almacenan atributos que son comunes para uno o más subtipos de entidad.
    - b) Un subtipo de entidad es una entidad cuyas instancias heredan algunos atributos comunes de un supertipo de entidad y luego añade otros atributos que son únicos para una instancia del subtipo.
  16. Un modelo lógico de datos se desarrolla en las siguientes etapas:
    - a) Las entidades se descubren y se definen.
    - b) Se construye un modelo de datos de contexto. Un modelo de datos de contexto contiene sólo entidades y relaciones de negocios identificadas por los dueños del sistema y los usuarios.
    - c) Se construye un modelo de datos basado en claves. El modelo basado en claves elimina las relaciones no específicas y añade las entidades asociativas. A todas las entidades en el modelo se les dan claves.
    - d) Se construye un modelo totalmente atribuido. Este modelo muestra todos los atributos que deben guardarse en el sistema.
    - e) Se construye un modelo totalmente descrito. Cada atributo se define en el diccionario y se describe en términos de propiedades como el dominio y la seguridad.
    - f) Entonces se analiza el modelo de datos completados con respecto a la adaptabilidad y la flexibilidad a través de un proceso llamado normalización. El modelo final analizado se llama modelo de datos de la tercera forma normal.
  17. Un modelo lógico de datos no comunica requerimientos de datos sobre una base operativa de negocios de posición. Los analistas de sistemas lo han encontrado útil para definir estos requerimientos en la forma de una matriz CLAB de datos de posición.



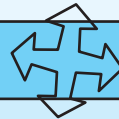
## Preguntas de repaso

1. ¿Cuál es la diferencia entre los modelos lógicos y los físicos?
2. ¿Por qué es importante crear un modelo independiente de implementación de un sistema?
3. ¿Por qué es necesario crear un modelo dependiente de implementación de un sistema?
4. ¿Qué es una entidad? ¿Cuáles son las instancias de entidad?
5. Una *relación* es una asociación natural de negocios entre entidades. ¿Cuál es la relación entre el estudiante y el maestro? ¿Depende de cuántas clases puede tomar un estudiante, o de cuántas clases un maestro puede enseñar?
6. ¿Qué es la cardinalidad? Dé un ejemplo.



## Problemas y ejercicios

1. ¿Qué es un dominio razonable para el atributo de datos del apellido de un estudiante?
2. ¿Qué valor por omisión escogería usted para el apellido de un estudiante?
3. ¿Qué valor por omisión escogería usted para el género?
4. La tabla de estudiantes con la que usted está trabajando contiene los atributos: la IDENTIDAD DEL ESTUDIANTE, el NOMBRE, el NÚMERO DE TELÉFONO y la especialidad. Normalice para 3NF.
5. ¿Qué atributos tendría usted en una tabla para describir una película?
6. Una relación de muchos a muchos (también designada una relación no específica) puede y generalmente debe transformarse en un par de relaciones uno a muchos con una entidad asociativa. ¿Cuándo no es éste el caso?
7. Dé un ejemplo de una relación muchos a muchos. Resuelva mediante el uso de una entidad o una entidad asociativa. ¿Cuál usó? ¿Por qué?
8. Describa cada una de las primeras tres formas normales. Dé un ejemplo de cada una.
9. Un cliente va a una zapatería y compra varios pares de zapatos. Diagrame esta relación.
10. Dé un ejemplo de cada una de las relaciones: ternaria, de identificación y sin identificación.
11. A primera vista, el modelado de datos parece no requerir mucha creatividad. ¿Por qué es esto incorrecto?
12. ¿Le puede dar una base de datos bien diseñada una ventaja estratégica a un negocio? ¿Cómo?



## Proyectos e investigación

1. Vaya a la biblioteca de la escuela. Pida al bibliotecario en el escritorio de circulación que imprima una copia de la información que tienen sobre usted. ¿Qué tipos de datos se guardan? ¿Hay algo que le asombre o que parezca irrelevante para sacar libros en préstamo? Si es así, pregunte por qué se colecta la información.
2. Cree una lista de atributos para la entidad estudiante de la información que usted encontró en el problema previo. Normalice para la tercera forma normal.
3. Vaya a una tienda de comestibles y haga una compra. ¿Qué tipo de datos mantendría un buen sistema de información sobre una transacción? ¿Qué hace un buen sistema de información por un negocio?
4. Compare su respuesta de la pregunta anteriormente citada (la tienda de comestibles) con la de al menos otro estudiante. ¿En qué fueron diferentes sus respuestas?
5. ¿Qué aspectos legales y de privacidad están relacionados con las bases de datos usadas por las tiendas de comestibles? Vaya al <http://www.findlaw.com> e indague algunos casos jurídicos recientes sobre el tema. Presente su trabajo en un ensayo corto (de cinco páginas) a su clase.
6. ¿Cómo pueden las bases de datos y la información en bases de datos ser usadas por los negocios para una ventaja estratégica? En un pequeño grupo, vaya a un negocio de su elección. Haga una lluvia de ideas para crear una base de datos que le ofrezca a esa compañía ya sea una solución a un problema existente o que explote una oportunidad existente en el negocio. Recuerde ser creativo.

## Casos breves



1. Considere una tienda de comestibles en línea ficticia llamada Wow Munchies. Ésta es una franquicia nacional, compuesta por mercadeo, contabilidad, fletes y departamentos de servicio al cliente. La directora general ha decidido actualizar la base de datos correspondiente a la red de la tienda para recolectar información pertinente de transacciones y compras de los diferentes departamentos. Ella está indecisa acerca de qué software o hardware soportará esta base de datos.

- Paso 1. ¿Cómo determinará usted qué datos son importantes para su recolección y mantenimiento en la base de datos nueva? Sea específico.
- Paso 2. Cree encuestas, cuestionarios y cosas por el estilo, y adminístrelas al personal apropiado en los departamentos afectados. Revise las formas que cada departamento usa.
- Paso 3. ¿Qué clase de respuestas obtuvo? Vaya a una tienda de comestibles en línea y vea qué datos están siendo recolectados por las compañías “rivales”. ¿Omitió usted algo? ¿Requirió alguna de las respuestas que obtuvo en una reunión secundaria con el personal del departamento? Si es así, revise sus encuestas, sus preguntas y su reentrevista.
- Paso 4. Utilice los métodos que esbozó en la pregunta anterior y evalúe las entidades y atributos que necesitará incluir en sus datos.
- Paso 5. Esboce las relaciones y la cardinalidad entre las entidades. ¿Qué tipo de modelo de datos está usted usando? ¿La implementación es específica o no? ¿Por qué?

Paso 6. Revise su modelo de datos de modo que no haya relaciones muchos a muchos, y el modelo se normalice para la tercera forma normal.

Paso 7. Someta todos los cuestionarios, encuestas, formas y respuestas a su profesor, junto con su borrador final del modelo de datos. Incluya una breve explicación en lo que se refiere a cómo derivó usted sus entidades y atributos, y las relaciones y cualesquiera suposiciones o limitaciones que su modelo de datos pueda tener.

2. Investigue a una agencia de renta de carros y cree un modelo de datos para su base de datos para las rentas de carros. ¿Qué departamentos son afectados por el alquiler de un coche? Revise cualesquiera formas públicamente disponibles y cree las encuestas y entrevistas necesarias para ayudarle a determinar lo que su base de datos debería contener. Asegúrese de normalizar para la tercera forma normal y resolver cualquier relación muchos a muchos. Entregue su modelo de datos y todos los documentos de soporte a su profesor.
3. Considere a la mafia. Suponga que los grupos del crimen organizado mantienen bases de datos para evadir la captura y para manejar sus negocios. ¿Qué información conservarían? ¿Por qué?
4. ¿Qué aspectos legales, éticos y de privacidad se asocian con las bases de datos que se usan en el combate al crimen? Indáguelo y presente un ensayo corto (diez páginas o menos) a su grupo.

## Ejercicios de equipo e individuales



1. La administración de proyectos en un equipo geográfica y culturalmente disperso es difícil. Asigne a cada miembro del equipo un país con un huso horario diferente e idiomas diferentes. Suponga que todos los miembros comparten la misma lengua, aunque ese idioma no será la lengua materna para todos los miembros. Los países de ejemplo son: Estados Unidos, India, Israel, China, Pakistán, Irán, Francia, Perú y Japón.
2. Ejercicio individual: Se dice que los límites de nuestra creatividad somos nosotros mismos y nuestras experiencias. Es decir, las mismas cosas que nos hacen ser quienes somos también nos constriñen. ¿Qué significa ser creativo? ¿Cómo se vuelve usted creativo y cómo piensa libremente?
3. Ejercicio individual o de equipo: Someta a un proceso de reingeniería un proceso de la vida común. ¿Si el “cielo es el límite”, cómo cambiaría usted este proceso? Identifique cada uno de los pasos en el proceso antiguo y luego cada paso en el nuevo proceso (por ejemplo, ponerse maquillaje, limpiar la casa, ir a la tienda de comestibles, asistir a clase). Lleve sus notas al grupo para una mesa redonda.



## Lecturas recomendadas

Bruce, Thomas A. *Designing Quality Databases with IDEF1X Information Models*. Nueva York: Dorset House Publishing, 1992. En realidad lo usamos como libro de texto para nuestro curso de análisis y diseño de bases de datos. IDEF1X es una sintaxis rica y estandarizada para el modelado de datos (que Bruce llama modelado de la información). El lenguaje gráfico se ve diferente, pero comunica los mismos conceptos de sistemas presentados en nuestro libro. El lenguaje está soportado por cuando menos dos herramientas CASE: *ERwin* de Trabajos Lógicos y *System Architect* de Popkin. El libro incluye dos estudios de caso.

Hay, David C. *Data Model Patterns: Conventions of Thought*. New York: Dorset House Publishing, 1996. ¡Qué buena novela! Comienza con la premisa de que la mayoría de modelos de datos de negocios se derivan de algunos patrones básicos repetitivos. A los vendedores de CASE, ¿qué les parecería incluir estos patrones en las herramientas de CASE como plantillas reutilizables?

Martin, James, y Clive Finkelstein. *Information Engineering*, 3 vols. Nueva York: Savant Institute, 1981. La ingeniería de la informa-

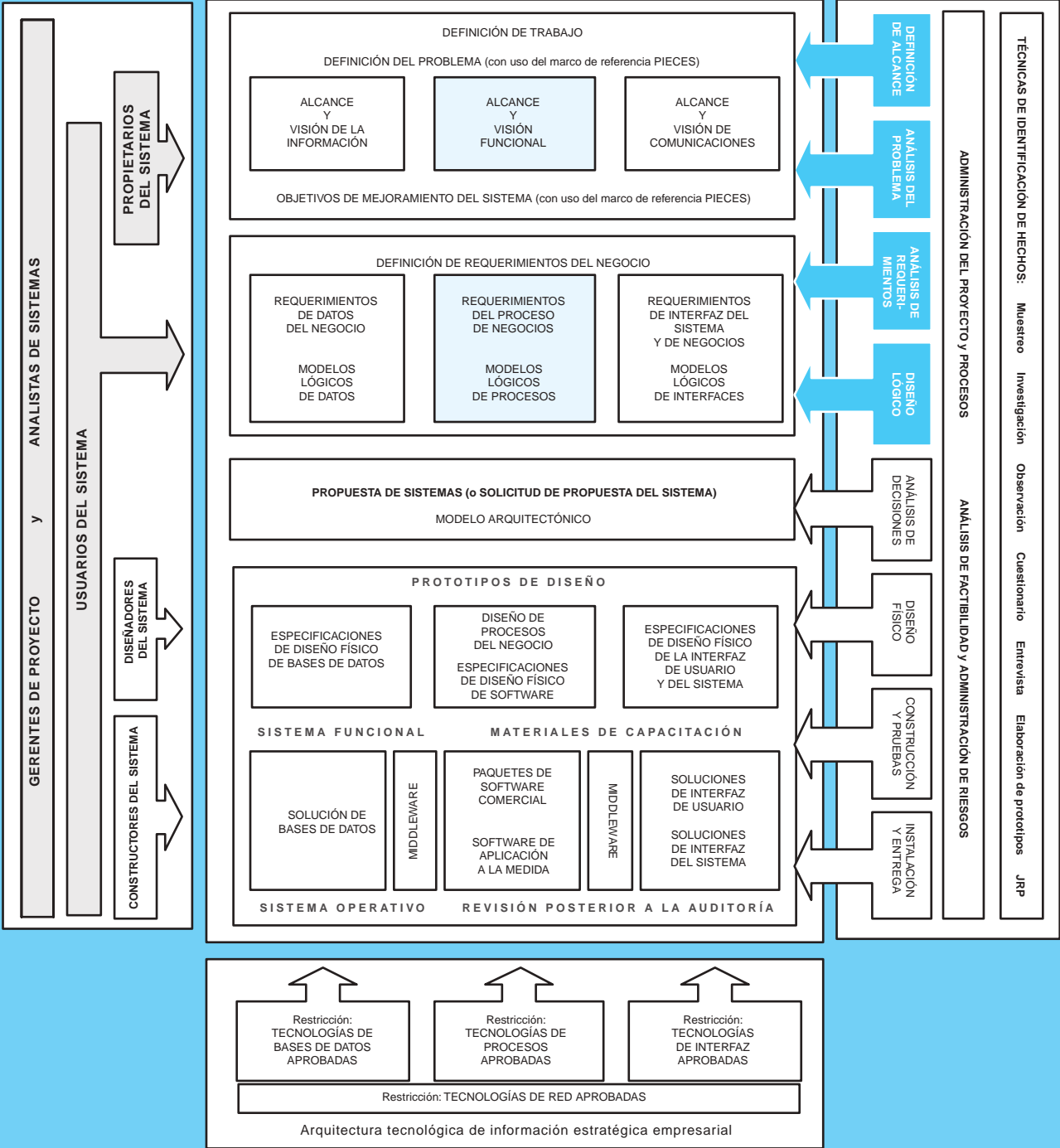
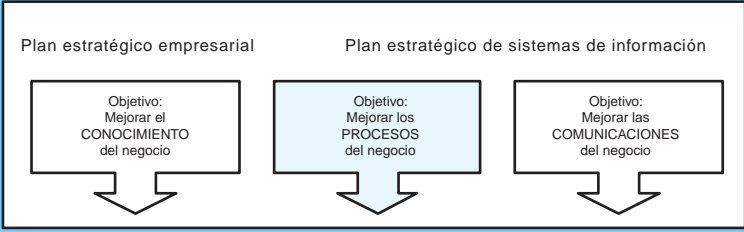
ción es una metodología formal orientada a los lenguajes de cuarta generación sobre las bases de datos. El lenguaje gráfico de modelado de datos es virtualmente idéntico al nuestro. El modelado de datos se estudia en los volúmenes I y II.

Reingruber, Michael, y William Gregory. *The Data Modeling Handbook*. Nueva York: John Wiley & Sons, 1994. Éste es un libro excelente sobre el modelado de datos y es especialmente útil para asegurar la calidad y la exactitud de los modelos de datos.

Schlaer, Sally, y Stephen J. Mellor. *Object-Oriented Systems Analysis: Modeling the World in Data*. Englewood Cliffs, NJ: Yourdon Press, 1988. ¡No se preocupen por el título! Hoy “Object-oriented” (orientado a objetos) significa algo diferente de lo que significaba en 1988, pero el libro todavía es uno de los más accesibles en el tema de modelado de datos.

Teorey, Toby J. *Database Modeling & Design: The Fundamental Principles*, 2a. ed., San Francisco: Morgan Kaufman Publishers, 1994. Este libro es un poco más conceptual que los otros en la lista, pero provee ideas útiles sobre la práctica del modelado de datos.







# 8

## Modelado de procesos

### Panorámica y objetivos del capítulo

En este capítulo usted aprenderá cómo dibujar diagramas de flujo de datos, un modelo popular que documenta los procesos del sistema y sus flujos de datos. Usted habrá aprendido sobre la modelación de procesos como una herramienta del análisis de sistemas cuando pueda:

- Definir la modelación de los sistemas y diferenciar entre los modelos lógicos y físicos de los sistemas.
- Definir la modelación de los procesos y explicar sus beneficios.
- Reconocer y entender los conceptos y elementos básicos de un modelo de procesos.
- Leer e interpretar un diagrama de flujo de datos.
- Explicar cuándo construir los modelos de procesos y dónde almacenarlos.
- Construir un diagrama de contexto para ilustrar las interfaces del sistema con su ambiente.
- Identificar los casos de uso y los eventos de negocios externos y temporales de un sistema.
- Realizar la división y organización de los eventos en un diagrama de descomposición funcional.
- Dibujar diagramas de eventos y luego fusionar estos diagramas de eventos con los diagramas de sistemas.
- Dibujar diagramas simples de flujos de datos y describir flujos y procesos elementales de datos en términos de las estructuras de datos y la lógica del procedimiento (inglés estructurado y tablas de decisión), respectivamente.
- Documentar la distribución de los procesos relacionados con las ubicaciones (sitios donde se usará el sistema dentro de la empresa).
- Sincronizar los datos y los modelos de procesos usando una matriz CLAB.

## Introducción

Bob Martínez ha estado trabajando en el sistema SoundStage Member Services por varias semanas. Él entiende el sistema bastante bien, pero todavía es fácil caer en confusiones y olvidar detalles.

“El problema”, dijo Bob a su jefa, Sandra Shepherd, “es que el sistema es demasiado grande para guardarlo todo en tu cabeza a la vez”.

“Me alegra que hayas dicho eso”, contestó Sandra, “porque tu siguiente asignación es descomponer el sistema en partes que podrás entender fácilmente. Cada parte se llama proceso, y tendrás que definir las entradas y salidas de ese proceso, además de quién o de dónde proviene cada entrada y hacia quién o hacia dónde va cada salida. Y, a propósito, necesitarás especificar la lógica del proceso”.

“Yo pensé que los casos uso hacían todo eso”, contestó Bob.

“No son lo suficientemente específicos”, afirmó Sandra. “¿Te gustaría turnar un proyecto a un programador localizado al otro extremo del país sin nada más de especificaciones que lo que tenga un caso práctico? ¿Quién sabe con lo que terminarías? Bienvenido al mundo de la modelación de los procesos, Bob. Pon manos a la obra.”

## Una introducción a la modelación de procesos

En el capítulo 4 se le introdujo a las actividades de análisis de sistemas que requerían del dibujo de modelos de sistemas. Los modelos de sistemas juegan un papel importante en el desarrollo de los sistemas. Como analista de sistemas o usuario, usted constantemente se ocupará de problemas no estructurados. Una forma de estructurar tales problemas es dibujar modelos. Un **modelo** es una representación de la realidad. Así como un cuadro vale mil palabras, la mayoría de los modelos de sistemas son representaciones pictóricas de la realidad. Los modelos pueden construirse para los sistemas existentes como una forma de entender mejor esos sistemas o para los sistemas propuestos como una forma de documentar los requisitos del negocio o los diseños técnicos. Un concepto importante es la distinción entre modelos lógicos y físicos.

Los **modelos lógicos** muestran *lo que* un sistema es o lo que hace. Son *independientes* de su implementación; es decir, bosquejan el sistema siendo independiente de cualquier implementación técnica. Como tales, los modelos lógicos ilustran la *esencia* del sistema. Los sinónimos populares incluyen *modelo elemental*, *modelo conceptual* y *modelo de negocios*. Los **modelos físicos** muestran no sólo *lo que* un sistema es o lo que hace sino que también *cómo* se implementa el sistema física y técnicamente. Son *dependientes* en la implementación porque reflejan la selección de la tecnología y las limitaciones de esas selecciones de tecnología. Son sinónimos *modelo de implementación* y *modelo técnico*.

Los analistas de sistemas hace mucho tiempo han reconocido el valor de separar los aspectos de negocios y los técnicos. Por esto es que usan modelos de sistemas lógicos para definir los requisitos de negocios y modelos de sistemas físicos para esquematizar diseños técnicos. Las actividades del análisis de sistemas tienden a enfocar la atención en los modelos de sistemas lógicos por las siguientes razones:

- Los modelos lógicos eliminan los sesgos que son el resultado de la forma en que se implementa el sistema actual, o la forma en que cualquier otra persona piensa que el sistema podría estar implementado. Así, vencemos el síndrome de: “siempre lo hemos hecho de esta forma”. Consecuentemente, los modelos lógicos alientan la creatividad.
- Los modelos lógicos reducen el riesgo de omitir los requisitos de negocios porque estamos demasiado preocupados con los detalles técnicos. Puede ser caro corregir estos errores después de que se implementa el sistema. Al separar lo que el sistema debe hacer de cómo lo hará el sistema, podemos analizar mejor los requisitos de totalidad, exactitud y consistencia.
- Los modelos lógicos nos permiten comunicarnos con los usuarios finales con lenguajes que no son técnicos o menos técnicos. Así, no perdemos requisitos en la jerga técnica de la disciplina de cómputo.

**modelo** Representación pictórica de la realidad.

**modelo lógico** Representación pictórica no técnica que ilustra lo que es un sistema o lo que hace. Los sinónimos son modelo elemental, modelo conceptual y modelo de negocios.

**modelo físico** Representación pictórica técnica que ilustra lo que es un sistema o lo que hace y cómo se implementa el sistema. Los sinónimos son modelo de implementación y modelo técnico.

En este capítulo nos centraremos exclusivamente en la modelación de procesos *lógicos* durante el análisis de los sistemas. El **modelado de procesos** es una técnica para organizar y documentar la estructura y el flujo de datos a través de los PROCESOS de un sistema y/o la lógica, las políticas y los procedimientos a ser implementados por los PROCESOS de un sistema. En el contexto de los bloques de construcción de un sistema de información (vea la página principal del sitio Web al principio del capítulo), se usan modelos *lógicos* de proceso para documentar el enfoque de PROCESO de un sistema de información desde la perspectiva de los dueños y de los usuarios del sistema (la intersección de la columna de PROCESO con los renglones del dueño del sistema y del usuario del sistema). También observe que un tipo especial de modelo de proceso, llamado *diagrama de contexto*, ilustra el enfoque de COMUNICACIÓN desde la perspectiva de los dueños y los usuarios del sistema.

La modelación de procesos se originó en los métodos clásicos de ingeniería del software; por consiguiente, usted pudo haber encontrado tipos diversos de modelos de proceso, tales como gráficas de estructura de programa, diagramas de flujo lógicos, o tablas de decisión en un curso de aplicación de programación. En este capítulo enfocaremos la atención en un modelo de procesos de análisis de sistemas: los *diagramas de flujo de datos*.

Un **diagrama de flujo de datos (DFD)** es una herramienta que esquematiza el flujo de datos a través de un sistema y el trabajo o el procesamiento realizado por el mismo. Los sinónimos incluyen *diagrama de burbuja*, *gráfica de transformación* y *modelo de procesos*. También introduciremos una herramienta de planificación de DFD llamada *diagramas de descomposición*. Finalmente, estudiaremos los *diagramas de contexto*, un modelo análogo al proceso que realmente ilustra las interfaces del sistema con los negocios y con el mundo exterior, incluyendo otros sistemas de información.<sup>1</sup>

En la figura 8.1 se ilustra un diagrama simple de flujo de datos. En la fase de diseño, parte de estos procesos de negocios podría implementarse como software de computadora (ya sea desarrollado internamente o comprado de un vendedor del software). Si usted examina este diagrama de flujo de datos, lo debería encontrar fácil de leer, aun antes de que termine este capítulo; ésa siempre ha sido la ventaja de los DFD. Hay sólo tres símbolos y una conexión:

- Los rectángulos redondeados representan los *procesos* o el trabajo por hacerse. Observe que se ilustran en el color de PROCESO de su marco de sistema de información.
- Los cuadrados representan a los *agentes externos*: el límite del sistema. Observe que se ilustran en la INTERFAZ de su marco de sistema de información.
- Las cajas con un extremo abierto representan el *almacenamiento de datos*, algunas veces llamado archivos o bases de datos. Si usted ya ha leído el capítulo 7, este almacenamiento de datos corresponde a todas las instancias de una sola entidad en un modelo de datos. Consecuentemente, han sido ilustradas con el color de DATOS de su marco de sistemas de información.\*
- Las flechas representan los *flujos de datos*, o entradas y salidas, hacia y desde los procesos.

Algunas veces hay una tendencia a confundir los diagramas de flujo de datos con los diagramas de flujo porque el diseño de programa frecuentemente incluye el uso de estos últimos. Sin embargo, los diagramas de flujo de datos son muy diferentes. Resumamos las diferencias.

- Los procesos en un diagrama de flujo de datos pueden operar en paralelo. Así, varios procesos podrían estar ejecutándose o trabajando simultáneamente. Esto es consistente con la forma en que trabajan los negocios. Por otra parte, los procesos en los diagramas de flujo pueden ejecutarse sólo uno a la vez.
- Los diagramas de flujo de datos muestran el flujo de datos a través del sistema. Sus flechas representan las trayectorias por las cuales los datos pueden fluir. Comúnmente no se muestran los circuitos y las ramificaciones. Por otra parte, los diagramas de flujo muestran la secuencia de los procesos o de las operaciones en un algoritmo o programa. Sus flechas representan los punteros para el siguiente proceso u operación. Esto puede incluir circuitos y ramificaciones.

<sup>1</sup> En el análisis estructurado clásico se considera que los diagramas de contexto son otro tipo de modelo de procesos. Pero en el análisis orientado a objetos, ilustran el alcance y las interfaces. En esta edición hemos escogido la última definición.

\* *Nota del autor:* Hay varios conjuntos de símbolos competitivos de los DFD. A lo largo de este capítulo, los autores han escogido usar la notación de Gane y Sarson debido a su amplia popularidad.

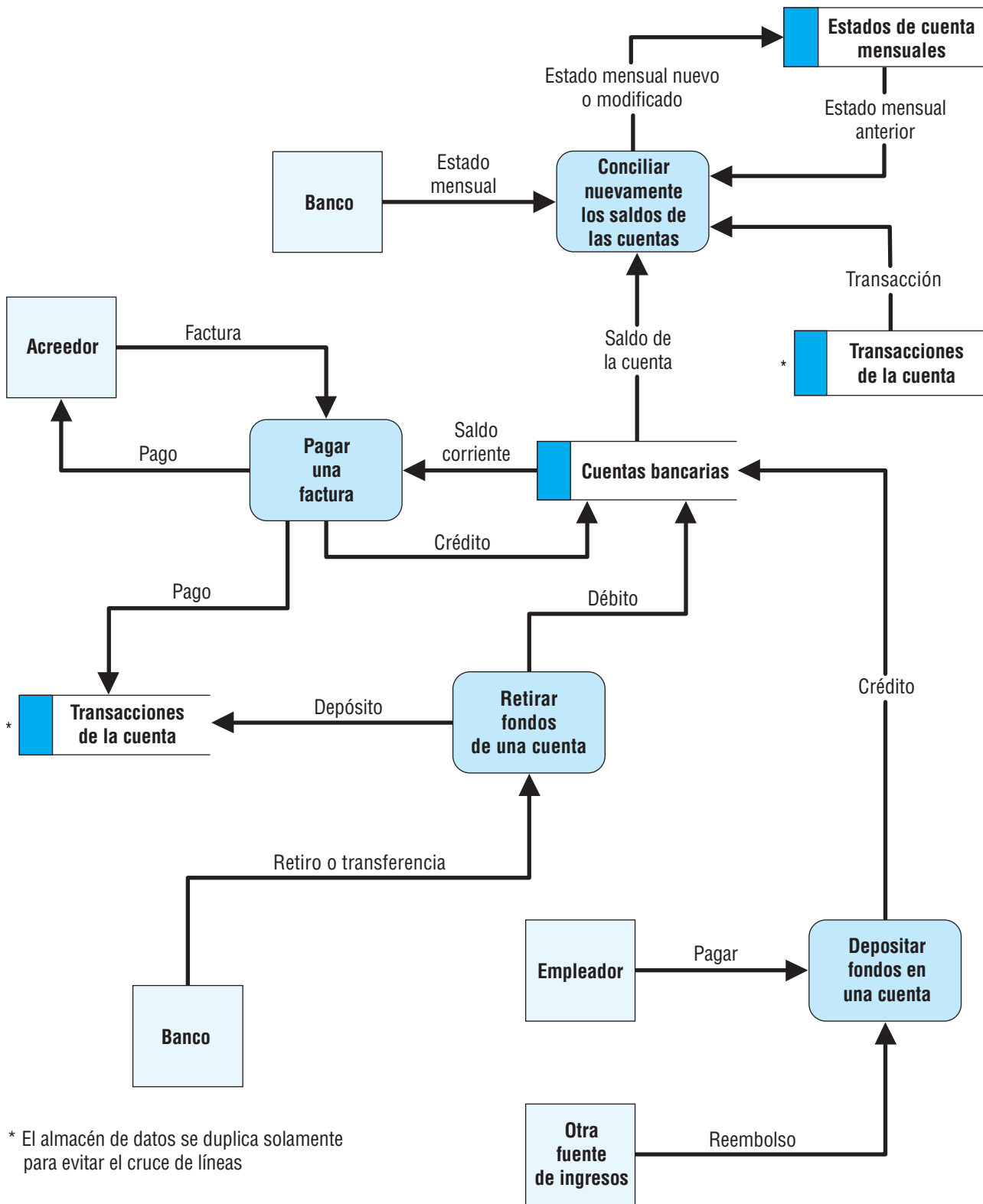
## modelado de procesos

Técnica que se usa para organizar y documentar los procesos de un sistema.

## diagrama de flujo de

**datos (DFD)** Modelo de procesos que se usa para ilustrar el flujo de datos a través de un sistema y el trabajo o procesamiento realizado por el sistema. Los sinónimos son diagrama de burbuja, gráfica de transformación, y modelo de procesos.





**FIGURA 8.1** Diagrama simple de flujo de datos

- Los diagramas de flujo de datos pueden mostrar procesos que tienen un cronometraje radicalmente diferente. Por ejemplo, un DFD individual podría incluir procesos que ocurren cada hora, diariamente, semanalmente, anualmente y a petición. Esto no ocurre en los diagramas de flujo.

Los diagramas de flujo de datos han sido populares por más de 20 años, pero el interés en los DFD se ha renovado recientemente por su aplicabilidad en el *rediseño de los procesos de negocios* (BPR, por sus siglas en inglés). Como los negocios han caído en la cuenta de que la mayoría de sistemas de procesamiento de datos han automatizado procesos de negocios obsoletos, ineficientes y burocráticos, hay un interés renovado en aerodinamizar los procesos de negocios. Esto se logra modelándolos primero con el objeto de analizarlos, rediseñarlos y/o mejorarlos. Subsecuentemente, la tecnología de la información puede aplicarse a los procesos de negocios mejorados en formas creativas que maximizan el valor devuelto al negocio. Estudiaremos nuevamente esta tendencia al final del capítulo.

## Los conceptos de sistemas para la modelación de procesos

### > Agentes externos

Todos los sistemas de información responden a los acontecimientos y las condiciones en el ambiente del sistema. El ambiente de un sistema de información incluye a los *agentes externos* que forman la frontera del sistema y definen lugares donde el sistema tiene una interfaz con su ambiente. Un **agente externo** define a una persona, una unidad de organización, otro sistema u otra organización que se sitúa fuera del alcance del proyecto pero que interactúa con el sistema en estudio. Los agentes externos proveen las entradas netas al sistema y reciben las salidas netas de un sistema. Un sinónimo común es *entidad externa* (no debe confundirlo con la *entidad de datos* que se introduce en el capítulo 7).

El término *externo* significa “fuera del sistema que está siendo analizado o diseñado”. En la práctica, un agente externo realmente puede estar fuera del negocio (tales como organismos gubernamentales, clientes, proveedores y contratistas), o puede estar dentro del negocio pero fuera del alcance del proyecto y del sistema (tales como otros departamentos, otras funciones de negocios y otros sistemas de información internos). Un agente externo se representa por un cuadrado en el diagrama de flujo de datos. El equivalente DeMarco/Yourdon es un rectángulo (vea en el margen).

Es importante reconocer que el trabajo y las actividades ocurren dentro del agente externo, pero que se dice que el trabajo y esas actividades están fuera del “alcance” y no están sujetas a cambios. Así, los flujos de datos entre su sistema y estas fronteras no deberán causar un cambio sustantivo al trabajo o a las actividades realizadas por los agentes externos.

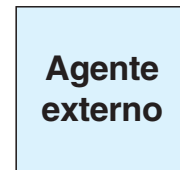
Los agentes externos de un sistema de información raramente están fijos. Como el alcance de proyecto y las metas cambian, el alcance de un sistema de información puede aumentar o decrecer. Si el sistema del alcance crece, puede consumir parte de los agentes externos originales; en otras palabras, lo que antes se consideraba fuera del sistema ahora se considera dentro del sistema (*como procesos nuevos*).

De modo semejante, si el alcance del sistema decrece (debido a restricciones del presupuesto o del programa), los procesos que una vez se consideraron dentro del sistema pueden convertirse en agentes externos.

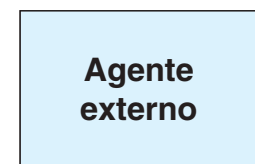
Los agentes externos en un diagrama lógico de flujo de datos pueden incluir a las personas, las unidades de negocios, otros sistemas internos con los cuales un sistema debe interactuar, y las organizaciones externas. Su inclusión en el DFD lógico significa que su sistema interactúa con estos agentes. Casi siempre son algunos de los siguientes:

- Una oficina, departamento, división o individuo dentro de su compañía que provee entradas netas a ese sistema, reciben salidas netas de ese sistema o ambos.
- Una organización, agencia o individuo que está fuera de su compañía pero que provee entradas netas o recibe salidas netas del sistema. Los ejemplos incluyen a los CLIENTES, LOS PROVEEDORES, LOS CONTRATISTAS, LOS BANCOS y LOS ORGANISMOS GUBERNAMENTALES.

**agente externo** Persona foránea, unidad de organización, sistema u organización que interactúa con un sistema. También llamado *entidad externa*.



Forma de Gane y Sarson



Forma de DeMarco/Yourdon

Símbolos para agente externo

- Otro negocio u otro sistema de información (tal vez, aunque no necesariamente, basado en computadora) que esté separado del sistema pero con el cual debe interactuar. Se está volviendo común crear interfaces con los sistemas de información de otros negocios.
- Los usuarios finales o los gerentes de un sistema. En este caso, el usuario o el gerente puede fungir como una fuente básica de datos que sirva de entrada a un sistema, o bien, un destino básico que funja como salida para ser el producto de un sistema.

Los agentes externos deberán nombrarse con sustantivos descriptivos, singulares, tales como REGISTRADOR, el PROVEEDOR, SISTEMA DE FABRICACIÓN o el SISTEMA DE INFORMACIÓN FINANCIERA. Los agentes externos representan sistemas *físicos* fijos; por consiguiente, pueden tener acrónimos o nombres físicos; aun en un DFD lógico. Por ejemplo, un agente externo que representa el sistema de información de la administración financiera de nuestra escuela sería designado FMIS (Financial Management Information System). Si un agente externo describe a un individuo, recomendamos títulos de trabajo o nombres de puestos en lugar de los nombres propios (por ejemplo, use el EJECUTIVO DE CUENTA, no *María López*).

Para evitar el cruce de líneas de flujo de datos en un DFD se permite duplicar agentes externos en DFDs. Pero como regla general, los agentes externos deberán ubicarse en los perímetros de la página, consistente con su definición como una frontera de sistema.

## > Los almacenes de datos

La mayoría de los sistemas de información captan datos para su uso posterior. Los datos se guardan en un almacén de datos, el último símbolo en un diagrama de flujo de datos. Se representa por la caja con extremo abierto (vea en el margen). Un **almacén de datos** es un “inventario” de información. Sus sinónimos son *archivo* y *base de datos* (aunque esos términos están demasiados orientados a la implementación para la modelación elemental de procesos). Si los flujos de datos son *datos en movimiento*, considere los almacenes de datos como *datos estáticos*.

Idealmente, los almacenes básicos de datos deberán describir “cosas” acerca de las cuales el negocio desea almacenar. Estas cosas incluyen:

**Personas:** AGENCIA, CONTRATISTA, CLIENTE, DEPARTAMENTO, DIVISIÓN, EMPLEADO, INSTRUCTOR, OFICINA, ESTUDIANTE, PROVEEDOR. Observe que una entidad o persona puede representar bien sea individuos, grupos u organizaciones.

**Lugares:** REGIÓN DE VENTAS, EDIFICIO, SALA, OFICINA REGIONAL, CAMPUS.

**Objetos:** LIBRO, MÁQUINA, PARTE, PRODUCTO, MATERIA PRIMA, LICENCIA DEL SOFTWARE, PAQUETE DE SOFTWARE, HERRAMIENTA, MODELO DE VEHÍCULO, VEHÍCULO. Una entidad puede representar objetos reales (como la LICENCIA del SOFTWARE) o especificaciones para un tipo de objeto (tal como PAQUETE DE SOFTWARE).

**Eventos:** SOLICITUD, CONCESIÓN, CANCELACIÓN, CLASE, VUELO, FACTURA, ORDEN, INSCRIPCIÓN, RENOVACIÓN, REQUISICIÓN, RESERVACIÓN, VENTA, VIAJE.

**Conceptos:** CUENTA, BLOQUE DE TIEMPO, ACCIÓN, CURSO, FONDO, CAPACITACIÓN, EXISTENCIAS.

**NOTA:** Si la lista anterior se ve familiar, deberá parecerse a: un almacén de datos que representa *todas las ocurrencias* de una entidad de datos, lo cual se define en el capítulo 7 como algo acerca de la información que queremos almacenar. Como tal, el almacén de datos representa la sincronización del modelo de procesos de un sistema con su modelo de datos.

Si se hace la modelación de datos antes que la modelación del proceso, la identificación de la mayoría de los almacenes de datos se simplifica por la siguiente regla:

Debería haber un almacén de datos para cada entidad de datos en un diagrama de entidad-relación. (Incluimos almacenes de datos asociativos y de entidad débil en nuestros modelos.)

Si, por otra parte, la modelación de procesos se hace antes de la modelación de datos, el descubrimiento de los almacenes de datos tiende a ser más arbitrario. En ese caso, nuestra mejor recomendación es identificar las implementaciones existentes de archivos o de almacenes de datos (por ejemplo, archivos de computadora y bases de datos, gabinetes de archivos, libros de registro, catálogos) y luego los renombra para reflejar las “cosas” lógicas

**almacén de datos** Datos almacenados destinados a un uso posterior. Los sinónimos son archivo y base de datos.



Forma de Gane y Sarson

**Almacén de datos**

Forma de DeMarco/Yourdon

Símbolos para almacén de datos

acerca de las cuales almacenan datos. Consistente con las estrategias de ingeniería de la información, recomendamos que los modelos de datos precedan a los modelos de proceso.

Generalmente, los almacenes de datos deberían nombrarse como el plural de la entidad correspondiente de un modelo de datos. Así, si el modelo de datos incluye una entidad llamada CLIENTE, los modelos de procesos incluirán un almacén de datos llamado CLIENTES. Esto tiene sentido porque el almacén de datos, por definición, almacena todas las instancias de la entidad. Evite términos físicos como *archivo*, *base de datos*, *gabinete de archivos*, *carpeta de archivos* y cosas por el estilo.

Como fue el caso con las fronteras, es permisible duplicar almacenes de datos en un DFD para evitar el cruzamiento de los flujos de datos. La duplicación deberá minimizarse.

### > Conceptos de procesos

Recuerde del capítulo 2 que un bloque constructivo fundamental de los sistemas de información son los procesos. Todos los sistemas de información incluyen procesos, usualmente muchos de ellos. Los procesos de sistemas de información responden a las condiciones y eventos de los negocios transformando datos (otro bloque constructivo) en información útil. Los procesos de modelación nos ayudan a entender las interacciones con el ambiente del sistema, otros sistemas y otros procesos.

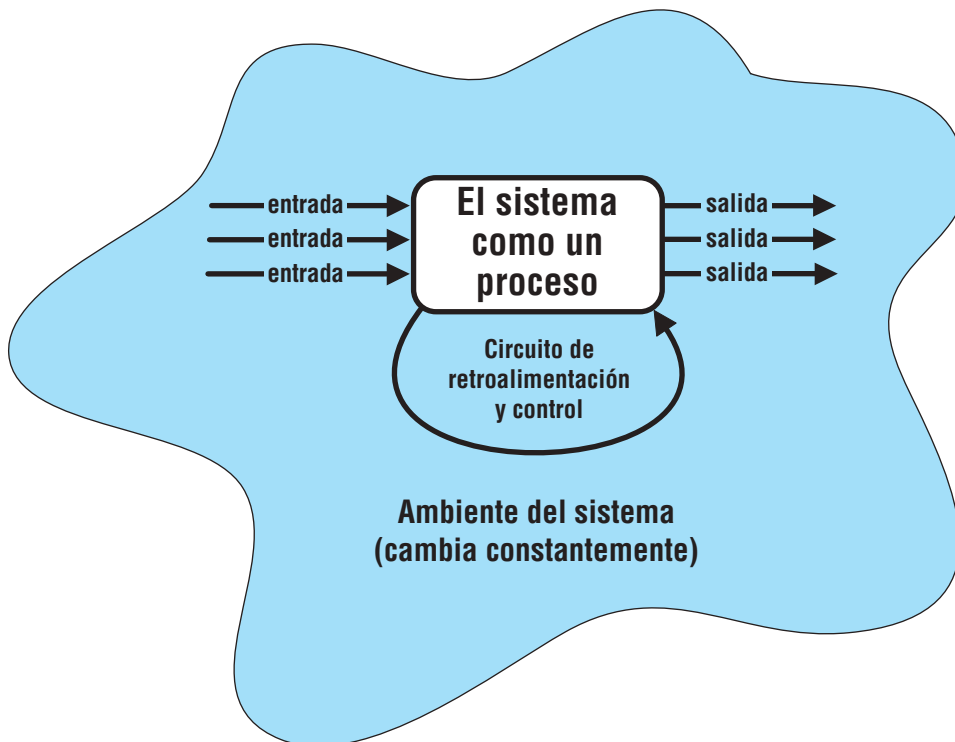
**Un sistema Es un Proceso** Hemos usado la palabra *sistema* a lo largo de todo este libro para describir casi cualquier arreglo ordenado de ideas o conceptos. Las personas hablan de sistemas educativos, sistemas de cómputo, sistemas de administración, sistemas de negocios y, claro está, los sistemas de información. De acuerdo con lo más antiguo y más simple de todos los modelos de sistemas, un sistema *es* un proceso.

En el análisis de los sistemas, los modelos se usan para ver o presentar un sistema. Como se muestra en la figura 8.2, el modelo de proceso más simple de un sistema se basa en entradas, salidas y el sistema mismo, visto como un proceso. El símbolo de proceso define la demarcación del sistema. El sistema está dentro del límite; el ambiente está fuera de ese límite. El sistema intercambia entradas y salidas con su ambiente. Ya que el ambiente siempre cambia, los sistemas bien diseñados tienen una retroalimentación y un lazo de control para permitir que el sistema se adapte a sí mismo para cambiar condiciones.

Considere un negocio como un sistema. Opera dentro de un ambiente que incluye a los clientes, proveedores, competidores, otras industrias y el gobierno. Sus entradas incluyen materiales, servicios, empleados nuevos, equipo nuevo, instalaciones, dinero y

Nombre del proceso

Forma de Gane & Sarson; se usa a lo largo de este libro



**FIGURA 8.2**  
El modelo clásico de procesos de un sistema

órdenes (por nombrar sólo unos cuantos). Sus salidas incluyen productos y/o servicios, materiales de desperdicio, el equipo retirado, los anteriores empleados y el dinero (los pagos). Monitorea su ambiente para hacer cambios necesarios en su línea de productos, servicios, procedimientos de operación, los competidores y la economía.

Un rectángulo redondeado (notación de Gane y Sarson) se usa a todo lo largo de este capítulo para representar un proceso (vea en el margen). Un **proceso** es trabajo que se realiza en, o como respuesta a, condiciones o flujos de datos entrantes. Un sinónimo es *transformar*. Las diferentes notaciones de modelación de proceso usan un círculo (la notación DeMarco/Yourdon) o un rectángulo (la notación SSADM/IDEFO). La elección a menudo depende de su metodología y de las características de la herramienta CASE.

Aunque los procesos pueden ser realizados por personas, departamentos, robots, máquinas o computadoras, otra vez queremos enfocar la atención en *qué* trabajo o *qué* acción está siendo realizada (el proceso *lógico*), no en *quién* o *qué* está haciendo ese trabajo o actividad (el proceso *físico*). Por ejemplo, en la figura 8.1 incluimos el proceso lógico RETIRAR FONDOS DE UNA CUENTA. No indicamos cómo se haría esto. Intuitivamente, podemos pensar acerca de varias implementaciones físicas tales como usar un ATM, utilizar el servicio de autopago de un banco, o realmente entrar en el banco.

**Descomposición de procesos** Generalmente, un sistema complejo es demasiado difícil de entender, sobre todo cuando es visto como un todo (lo que significa *como un proceso individual*). Por consiguiente, en el análisis de sistemas separamos un sistema en sus subsistemas o componentes, los cuales se descomponen a su vez en subsistemas más pequeños, hasta que hayamos identificado los subconjuntos manejables del sistema global (véase la figura 8.3). Llamamos a esta técnica *descomposición*. La **descomposición** es el acto de separar el sistema en sus subsistemas, componentes, procesos y subprocessos. Cada nivel de *abstracción* revela mayor o menor detalle (tal como se desea) acerca del sistema global o un subconjunto de ese sistema. Usted ya ha aplicado la descomposición de varias maneras. La mayoría de ustedes ha *desglosado* tareas; ésta es una forma de descomposición. Muchos de ustedes han fragmentado un programa de computadora de tamaño mediano en uno grande que comprenda subprogramas que podrían desarrollarse y probarse independientemente antes de que sean integrados. Esto es también una descomposición.

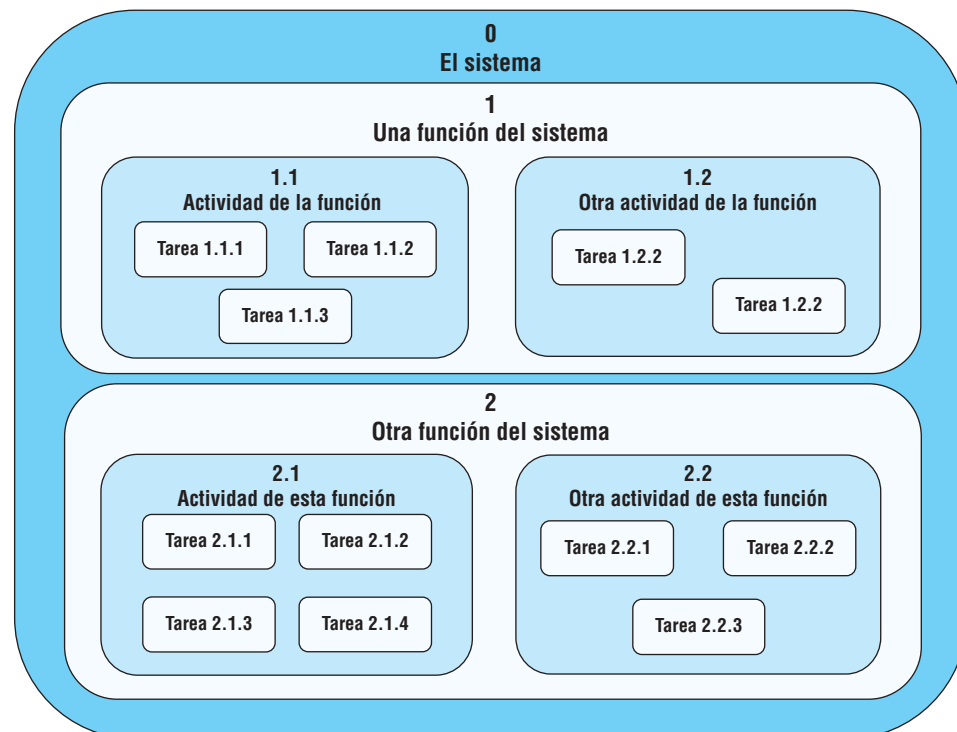
En el análisis de sistemas, la descomposición le permite fragmentar un sistema en los subsistemas lógicos de procesos para la comunicación, análisis y diseño mejorados. Un diagrama parecido a la figura 8.3 puede ser un poco difícil de construir al tratar con

**proceso** Trabajo realizado por un sistema como respuesta a los flujos de datos o condiciones entrantes. Un sinónimo es *transformar*.

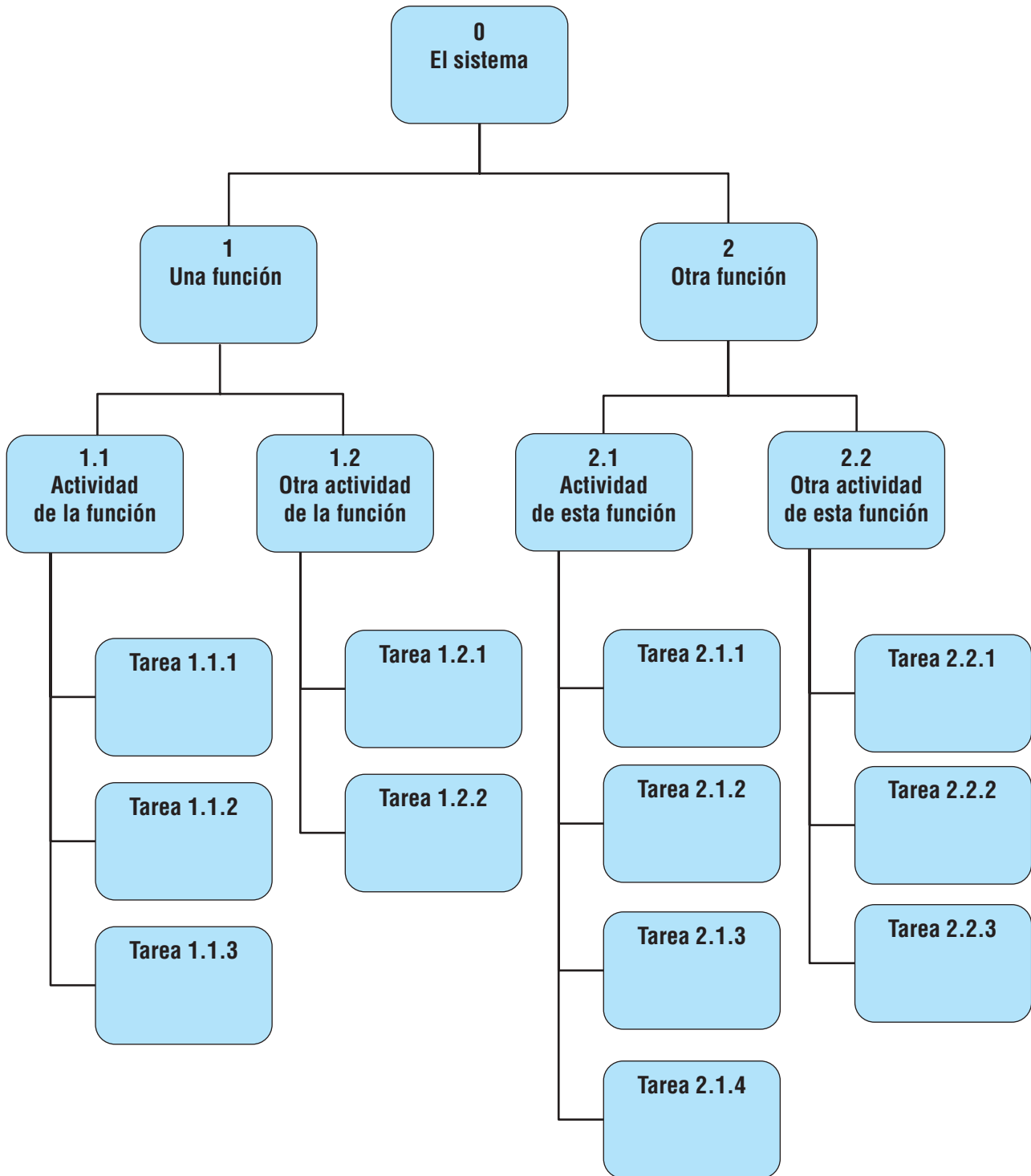
**descomposición** Acto de desarmar un sistema en subcomponentes.

### FIGURA 8.3

Un sistema consta de muchos subsistemas y procesos







**FIGURA 8.4** Diagrama simple de flujo de datos

cualquier Sistema, excepto con los más pequeños. La figura 8.4 muestra una descripción alternativa que está soportada por muchas herramientas CASE y metodologías de desarrollo. Se le llama *diagrama de descomposición*. Lo usaremos frecuentemente en este capítulo. Un **diagrama de descomposición**, también llamado *gráfica de jerarquía*, demuestra la descomposición funcional de arriba abajo y la estructura de un sistema. Un diagrama de descomposición es sobre todo una herramienta de planificación para

**diagrama de descomposición** Herramienta que se usa para ilustrar la descomposición de un sistema. También llamado *gráfica de jerarquía*.

modelos de procesos más detallados, llamados diagramas de flujo de datos. Para estos se deben aplicar las siguientes reglas:

- Cada proceso en un diagrama de descomposición puede ser un *proceso padre*, un *proceso hijo* (de un padre) o ambos.
- Un padre *debe* tener dos o más hijos; un hijo único no tiene sentido porque eso no revelaría ningún detalle adicional acerca del sistema.
- En la mayoría de los estándares de descomposición de diagramas, un hijo puede tener sólo un padre.
- Finalmente, un hijo de un padre puede ser el padre de sus propios hijos.

Las mitades superior e inferior del diagrama de descomposición en la figura 8.4 demuestran dos estilos para diseñar los procesos y las conexiones. Usted puede usar cualquiera de los dos o ambos, como sea necesario, para presentar un modelo sin dificultades. Algunos modelos pueden requerir páginas múltiples para mayor claridad.

Las conexiones en un diagrama de descomposición no contienen puntas de flecha porque el diagrama tiene como objetivo mostrar la *estructura*, no el *flujo*. Tampoco se nombran las conexiones. Implícitamente todas ellas tienen el mismo nombre (CONSISTENTE EN) ya que la suma de los procesos hijos para un proceso padre es *igual* al proceso del padre.

**Convenciones y procesos lógicos** Los procesos *lógicos* son trabajo o acciones que deben realizarse sin importar *cómo* implemente usted el sistema. Cada proceso lógico es (o será) implementado como uno o más procesos físicos que pueden incluir trabajo realizado por las personas, por robots o por máquinas, o trabajo realizado por el software de la computadora. Sin embargo, no tiene importancia qué implementación se use, porque los procesos lógicos sólo deberán señalar qué hay trabajo que debe hacerse.

El nombramiento de las convenciones para los procesos lógicos depende de dónde se sitúa el proceso en el diagrama de flujo de datos/diagrama de descomposición y el tipo de proceso esquematizado. Hay tres tipos de procesos lógicos: *funciones*, *eventos* y *procesos elementales*.

Una **función** es un conjunto de actividades relacionadas con el negocio. Una función no tiene principio o fin; únicamente realiza su trabajo continuamente según se necesite. Por ejemplo, un sistema de manufactura puede incluir las siguientes funciones (subsistemas): PRODUCCIÓN, PLANEACIÓN, PROGRAMACIÓN DE LA PRODUCCIÓN, ADMINISTRACIÓN DE MATERIALES, CONTROL DE LA PRODUCCIÓN, ADMINISTRACIÓN DE LA CALIDAD Y CONTROL DE INVENTARIOS. Cada una de estas funciones puede constar de docenas o centenares de procesos más discretos para soportar tareas y actividades específicas. Los nombres de función son sustantivos que describen la función entera. Los ejemplos adicionales son ENTRADA DE ORDEN, ADMINISTRACIÓN DE ÓRDENES, REPORTES DE VENTAS, RELACIONES DE LOS CLIENTES, Y DEVOLUCIONES Y REEMBOLSOS.

Un **evento** es una unidad de trabajo lógica que debe completarse como un todo. Un evento se desencadena por una entrada discreta y se completa cuando el proceso ha respondido con salidas apropiadas. Los eventos algunas veces se llaman *transacciones*. Las funciones constan de procesos que responden a los acontecimientos. Por ejemplo, la función de ADMINISTRACIÓN DE MATERIALES puede responder a los siguientes eventos: PRUEBA DE LA CALIDAD DEL MATERIAL, EXISTENCIA DE MATERIALES NUEVOS, DISPOSICIÓN DE MATERIALES DAÑADOS, DISPOSICIÓN DE MATERIALES ECHADOS A PERDER, REQUISICIÓN DE MATERIALES PARA LA PRODUCCIÓN, DEVOLUCIÓN DE MATERIALES NO USADOS EN LA PRODUCCIÓN, ORDENAR MATERIALES NUEVOS, etc. Cada uno de estos eventos tiene un gatillo y respuesta que pueden definirse por sus entradas y salidas.

Con el uso de técnicas de análisis estructurado *moderno* como los recomendados por McMenamin, Palmer, Yourdon y los Robertson (vea las lecturas recomendadas al final del capítulo), los analistas descomponen las funciones del sistema en eventos de negocios. Cada acontecimiento de negocios se representa por un proceso individual que responderá a ese acontecimiento. Los nombres de proceso de los eventos tienden a ser muy generales. Adoptaremos la convención de nombrar los procesos de eventos como sigue: EL PROCESO \_\_\_\_\_, RESPONDE A \_\_\_\_\_, O GENERA \_\_\_\_\_, donde el espacio vacío sería el nombre del evento (o su entrada correspondiente). Los nombres de procesos de eventos de muestra son PROCESO DE LA ORDEN DEL CLIENTE, PROCESO DE CAMBIO DE LA ORDEN DEL CLIENTE, PROCESO DEL CAMBIO DE LA DIRECCIÓN DEL CLIENTE, RESPONDER A LAS QUEJAS DEL CLIENTE, RESPONDER A LA AVERIGUACIÓN DE LAS ÓRDENES, RESPUESTA A LA VERIFICACIÓN DE

---

**función** Conjunto de actividades relacionadas y continuas de un negocio.

---

**evento** Unidad lógica de trabajo que debe completarse como un todo. Algunas veces llamado *transacción*.

PRECIOS DE LOS PRODUCTOS, GENERACIÓN DE LOS REPORTES DE LAS DEVOLUCIONES, GENERACIÓN DEL ESTADO DE CUENTA DEL CLIENTE Y GENERACIÓN DE FACTURAS.

Un proceso de evento puede descomponerse aún más en procesos elementales que ilustran en detalle cómo el sistema debe responder a un evento. Los **procesos elementales** son actividades discretas, detalladas o tareas requeridas para completar la respuesta a un evento. En otras palabras, son el nivel más bajo de detalle esquematizado en un modelo de proceso. Un sinónimo común podría ser *proceso primitivo*. Los procesos elementales deberán nombrarse con un verbo de acción fuerte seguido por una cláusula del objeto que describe sobre qué (o para qué) se realiza un trabajo. Los ejemplos de nombres elementales de proceso son VALIDAR LA IDENTIFICACIÓN DE LOS CLIENTES, VALIDAR EL NÚMERO ORDENADO DE PRODUCTO, VERIFICAR LA DISPONIBILIDAD DEL PRODUCTO, CALCULAR EL COSTO DE LA ORDEN, COMPROBAR EL CRÉDITO DEL CLIENTE, CLASIFICAR LAS DEVOLUCIONES, OBTENER LA DIRECCIÓN DEL CLIENTE, ACTUALIZAR LA DIRECCIÓN DEL CLIENTE, AGREGAR A UN CLIENTE NUEVO Y SUPRIMIR AL CLIENTE.

Los modelos de procesos *lógicos* omiten cualquier proceso que no haga nada más que mover o encaminar los datos, dejando así los datos sin cambio. Así, usted deberá omitir cualquier proceso que le corresponda a una secretaria u oficinista, quienes reciben y simplemente remiten diferentes documentos a la siguiente posición de proceso. Por último, usted deberá quedarse únicamente con procesos lógicos que:

- *Realicen cálculos* (calcular el promedio general).
- *Tomen decisiones* (determinar la disponibilidad de los productos ordenados).
- *Ordenen, filtren o de otra manera resuman los datos* (identificar las facturas atrasadas).
- *Organicen los datos en información útil* (generar un reporte o contestar una pregunta).
- *Desencadenen otros procesos* (encender el horno o instruir a un robot).
- *Usen datos almacenados* (crear, leer, actualizar o borrar un registro).

Tenga el cuidado de evitar tres errores mecánicos comunes con los procesos (ilustrado en la figura 8.5):

- El proceso 3.1.2 tiene entradas pero ninguna salida. Llamamos a lo que se conoce como un *agujero negro*, porque los datos entran en el proceso y luego desaparecen. En la mayoría de los casos, el modelador simplemente olvidó incluir la salida.
- El proceso 3.1.3 tiene salidas pero ninguna entrada. En este caso, probablemente los flujos de entrada fueron olvidados.
- En el proceso 3.1.1 las entradas son insuficientes para producir la salida. Llamamos a esto un *agujero gris*. Hay varias causas posibles, incluyendo 1) un proceso mal nombrado, 2) entradas y/o salidas mal nombradas, o 3) hechos incompletos. Los agujeros grises son los errores más comunes y los más bochornosos. Una vez que se entregan al programador, los flujos de datos de entrada a un proceso (para implementarse como un programa) deben ser suficientes para producir los flujos de datos de salida.

## > Flujos de datos

Los procesos responden a las entradas y generan salidas. De esta forma, como mínimo, todos los procesos tienen al menos un *flujo de datos* de entrada y otro de salida. Los flujos de datos son las comunicaciones entre procesos y el ambiente del sistema. Examinemos algunos de los conceptos y convenciones básicas de los flujos de datos.

**Datos en movimiento** Un flujo de datos es *información en movimiento*. El flujo de datos entre un sistema y su ambiente o entre dos procesos dentro de un sistema es *comunicación*. Estudiemos esta forma de comunicación.

Un **flujo de datos** representa una entrada de datos a un proceso o la salida de datos (o la información) de un proceso. Un flujo de datos se usa también para representar la creación, lectura, supresión o actualización de datos en un archivo o base de datos (designados como *almacén de datos* en el DFD). Piense acerca de un flujo de datos como una carretera por la cual transitan paquetes de composición conocida. El nombre implica la clase de datos que pueden transitar por esa carretera. Esta carretera se esquematiza como una línea continua con flecha (vea en el margen).

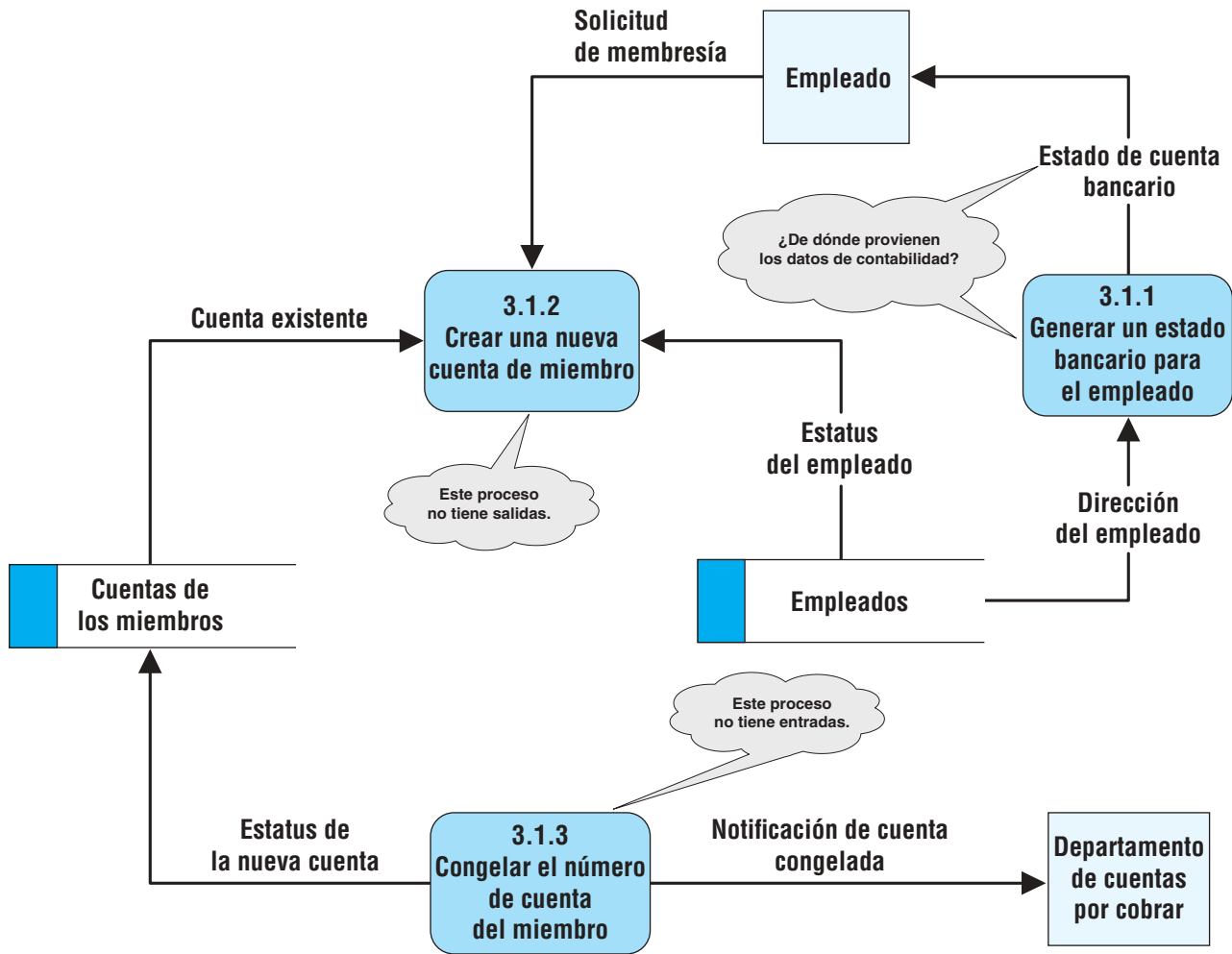
El concepto de *paquete* es crítico. Los datos que deberán viajar conjuntamente deberán mostrarse como un flujo de datos individual, sin importar cuántos documentos *físicos* podrían incluirse. El concepto de paquete se ilustra en la figura 8.6, la cual muestra las formas correctas e incorrectas para presentar un paquete lógico de flujo de datos.

**proceso elemental** Actividad o tarea discreta o detallada que se requiere para completar la respuesta a un evento. También llamado *proceso primitivo*.

**flujo de datos** Datos que entran o salen de un proceso.

Nombre del flujo de datos →

Símbolo de flujo de datos

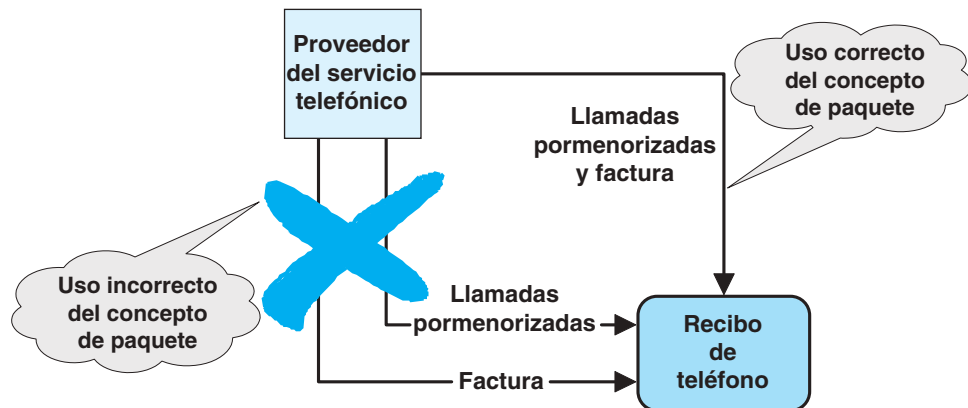


**FIGURA 8.5** Errores comunes en los diagramas de flujo de datos

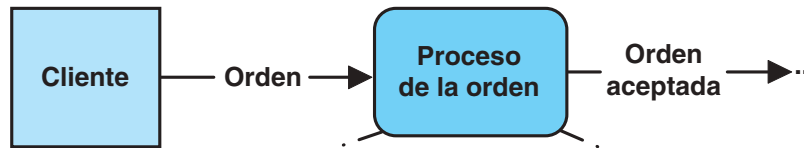
**flujo de datos compuesto** Flujo de datos que consta de otros flujos de datos.

El concepto de *composición conocida* es igualmente importante. Un flujo de datos está compuesto ya sea de atributos de datos reales (también llamados *estructuras de datos*; abundaremos acerca de ellos posteriormente) u otros flujos de datos. Un **flujo de datos compuesto** es un flujo de datos que consta de otros flujos de datos. Se usan para combinar flujos de datos similares en diagramas de flujo de datos de alto nivel para simplificar la lectura de esos diagramas. Por ejemplo, en la figura 8.7(a), un DFD de alto nivel consolida todos los tipos de órdenes en un flujo de datos compuesto designado

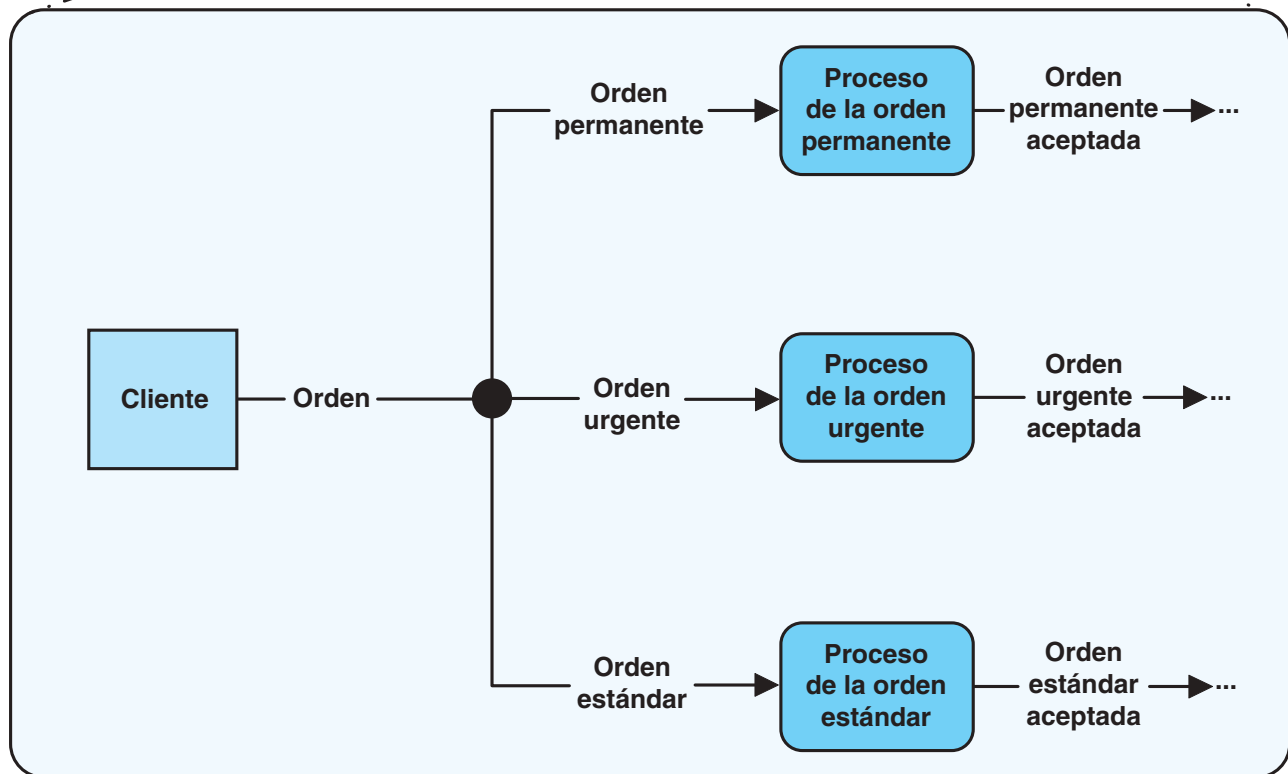
**FIGURA 8.6**  
El concepto de paquete de flujo de datos



(a) DFD de alto nivel



(b) DFD más detallado

**FIGURA 8.7** Flujos de datos elementales y compuestos

ORDEN. En la figura 8.7(b), un diagrama de flujo de datos más detallado muestra tipos específicos de órdenes: ORDEN VIGENTE, ORDEN URGENTE y ORDEN ESTÁNDAR. Estas órdenes diferentes requieren un procesamiento algo diferente. (El círculo pequeño y negro se llama *conexión*. Señala que cualquier ORDEN dada es una instancia de sólo uno de los tipos de orden.)

Otro uso común de flujos de datos compuestos es consolidar todos los informes y respuestas de averiguación en uno o dos flujos complejos. Hay dos razones para esto. Primero, estas salidas pueden ser muy numerosas. Segundo, muchos sistemas modernos proveen informes extensos creados por el usuario y averiguaciones que no serán predecibles antes de la implementación y uso del sistema.

Algunos métodos de diagramación de flujo de datos también reconocen los *flujos sin datos* designados como *flujos de control*. Un **flujo de control** representa una condición o un evento sin datos que desencadena un proceso. Considérelolo como una condición para ser monitoreada mientras el sistema trabaja. Cuando el sistema se percata de que la condición cumple con algún estado predeterminado, comienza el proceso para el cual se alimenta. El ejemplo clásico de un sistema de información es el *tiempo*. Por ejemplo, un proceso de generación de reportes puede desencadenarse por el evento temporal FIN DE MES. En

**flujo de control** Condición o evento sin datos que desencadena un proceso.

Nombre del flujo de control ▷

Símbolo del  
flujo de control

los sistemas de tiempo real, los flujos de control a menudo representan condiciones de tiempo real como la TEMPERATURA y la ALTITUD. En la mayoría de metodologías que distinguen entre los flujos de datos y de control, el flujo de control se esquematiza como una línea segmentada con flecha (vea en el margen).

Típicamente, los analistas de sistemas de información se han ocupado en su mayor parte de flujos de datos; sin embargo, a medida que los sistemas de información se integran más con los sistemas de tiempo real (tales como los procesos de fabricación y la fabricación de computadoras integradas), se hace necesario distinguir el concepto de flujos de control.

**Convenciones y flujos de datos lógicos** Mientras que reconocemos que los flujos de datos pueden implementarse de varias maneras (por ejemplo, llamadas por teléfono, formatos de negocios, códigos de barras, memoranda, reportes, pantallas de la computadora y comunicaciones de computadora a computadora), tenemos interés sólo en los flujos de datos *lógicos*. Así, tenemos sólo interés en que el flujo sea necesario (no en cómo implementaremos ese flujo). Los nombres de flujo de datos deberían desalentar un compromiso prematuro con cualquier implementación posible.

Los nombres de flujo de datos deberán ser sustantivos descriptivos y frases sustantivadas que son singulares, en contraposición al plural (ORDEN; no ÓRDENES). No queremos dar por hecho que las ocurrencias del flujo deban implementarse como un proceso de *lotes* (*batch*) *físico*.

Los nombres del flujo de datos también deberán ser únicos. Use adjetivos y adverbios para ayudar a describir cómo el procesamiento ha cambiado un flujo de datos. Por ejemplo, si una entrada para un proceso se llama ORDEN, la salida no deberá llamarse ORDEN. Podría llamarse ORDEN VÁLIDA, ORDEN APROBADA, ORDEN CON PRODUCTOS VÁLIDOS, ORDEN CON CRÉDITO APROBADO, o cualquier otro nombre más descriptivo que refleje lo que el proceso hizo con la orden original.

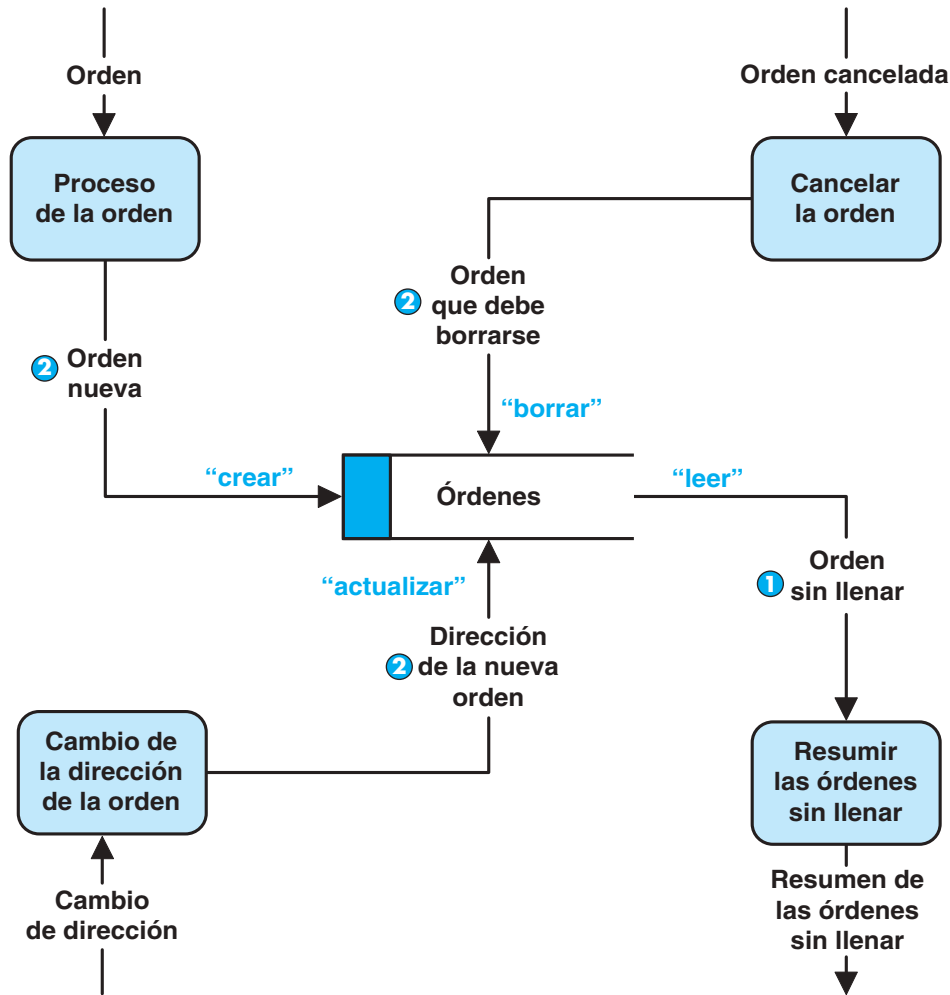
Los flujos de datos lógicos hacia y desde los almacenes de datos requieren consideraciones especiales de nombrado (vea la figura 8.8). (Los nombres del almacén de datos son plurales y las viñetas numeradas corresponden a la nota de la figura.)

- Sólo se muestra el flujo *neto* de datos. En forma intuitiva, usted puede darse cuenta de que tiene que obtener un registro para actualizarlo o suprimirlo. Pero a menos que los datos sean necesarios para algún otro propósito (por ejemplo, un cálculo o una decisión), no se muestra la acción “leer”. Esto mantiene al diagrama sin obstrucciones.
- ① Un flujo de datos que va de un almacén de datos a un proceso indica que deben “leerse” datos para algún propósito específico. El nombre del flujo de datos deberá indicar claramente qué datos deben ser leídos. Esto se muestra en la figura 8.8.
- ② Un flujo de datos que va de un proceso a un almacén de datos indica que deben crearse, suprimirse o actualizarse datos en/desde ese almacén de datos. Nuevamente, como se muestra en la figura 8.8, estos flujos de datos deberán nombrarse claramente para reflejar la acción específica realizada (como CLIENTE NUEVO, CLIENTE PARA SER SUPRIDO, o DIRECCIÓN ACTUALIZADA DE ORDEN).

Observe que los nombres sugieren las acciones clásicas que pueden realizarse en un archivo, a saber, CREAR, LEER, ACTUALIZAR y BORRAR (CLAB). En un DFD verdadero, realmente no registraríamos estos nombres de acción en el diagrama.

Ningún flujo de datos deberá carecer de nombre. Los flujos de datos anónimos son frecuentemente el resultado de la etapa de pensamiento del diagrama de flujo (paso 1, paso 2, etc.). Si usted no le puede dar al flujo de datos un nombre razonable, probablemente no exista. Siendo consistentes con nuestra meta de modelado *lógico*, los nombres de flujo de datos deberán describir el flujo de datos sin describir cómo es el flujo o cómo podría implementarse. Suponga, por ejemplo, que los usuarios finales explican su sistema como sigue: “Llenamos la Forma 23 por triplicado y la enviamos a...” El nombre lógico para la “forma 23” el flujo de datos podría ser SOLICITUD DE CURSO. Este nombre lógico elimina los sesgos físicos de la implementación; la idea de que debemos usar una *forma de papel* y la noción que debemos usar copias al carbón. Finalmente, esto nos liberará para considerar otras alternativas físicas, como las respuestas telefónicas digitales, las pantallas de inscripción en línea o aun las páginas de Internet de *e-business*.

Finalmente, todos los flujos de datos deben comenzar y/o acabar en un proceso, porque los flujos de datos son las entradas y las salidas de un proceso. Consecuentemente, todos los flujos de datos en el lado izquierdo de la figura 8.9 son ilegales. Los diagramas corregidos se muestran del lado correcto.

**FIGURA 8.8**

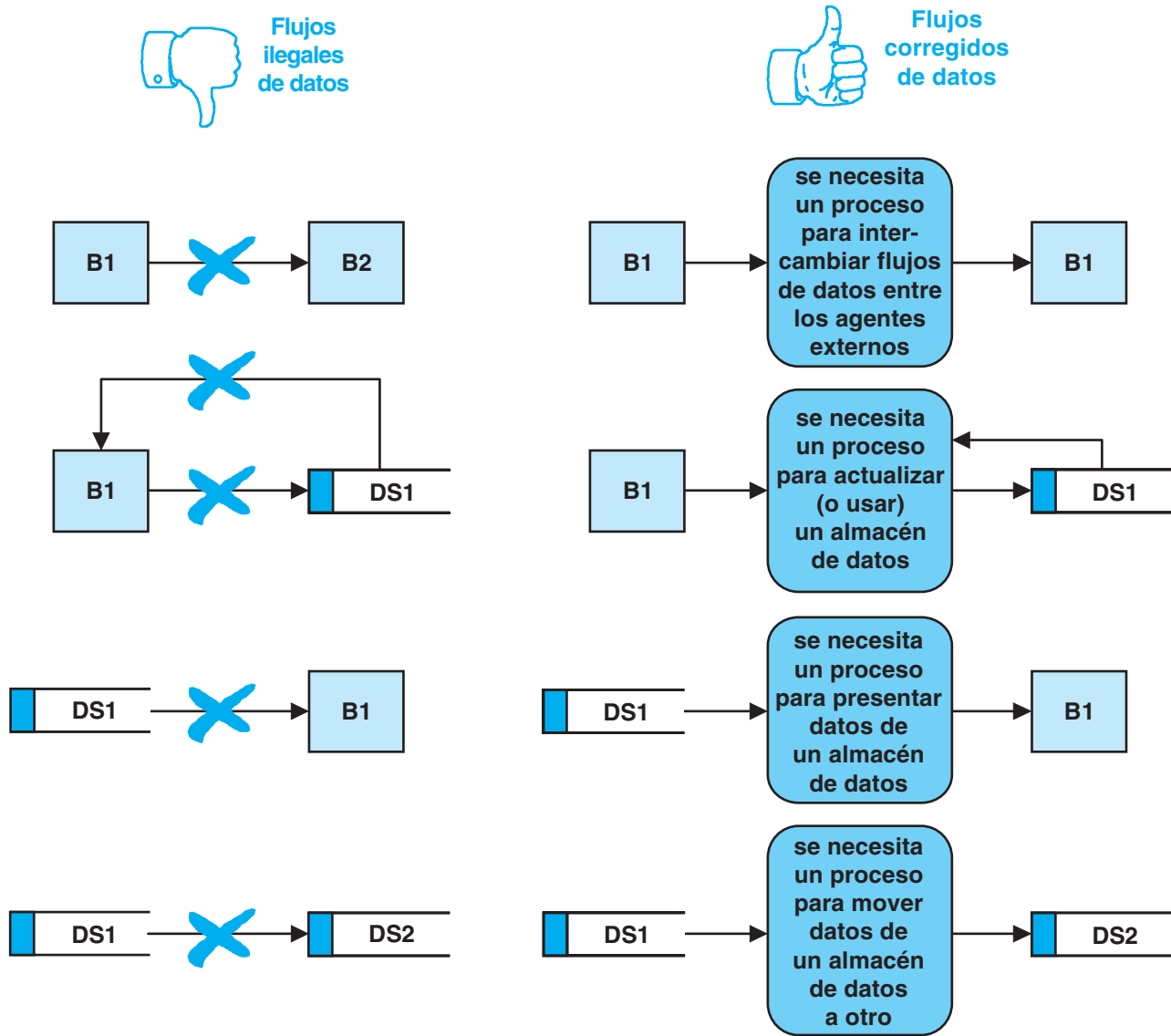
Flujos de datos hacia y desde los almacenes de datos

**Conservación del flujo de datos** Por largos años hemos tratado de mejorar los procesos de negocios automatizándolos. Eso no siempre ha funcionado o marchado bien porque los procesos de negocios se diseñaron para procesar flujos de datos en una era anterior a la computación. Considere el formato comercial común. Es frecuente ver la forma dividida en secciones que se diseñan para audiencias diferentes. El primer destinatario completa su parte de la forma, el siguiente destinatario completa su parte, y así sucesivamente. En ciertos puntos de esta secuencia de proceso, una copia de la forma aun podría separarse y enviarse a otro destinatario, que inicia una forma nueva de partes múltiples que requiere la transcripción de muchos de los mismos datos provenientes de la forma inicial. En nuestra propia universidad hemos visto ejemplos donde el diseño deficiente de la forma requiere que los mismos datos se capturen una docena de veces.

Ahora, si el flujo de datos actuales se computariza basándose en las formas y procesos de negocios actuales, los programas de computadora resultantes meramente automatizarán estas ineficiencias. ¡Esto es precisamente lo que ha ocurrido en la mayoría de los negocios! Actualmente, un énfasis nuevo en el *rediseño de los procesos de negocios* alienta a la gerencia, los usuarios y los analistas de sistemas a identificar y eliminar estas ineficiencias antes de diseñar cualquier sistema de información nuevo. Podemos soportar esta tendencia en los diagramas *lógicos* de flujo de datos practicando la conservación de datos. La **conservación de datos**, algunas veces designada como "inanición de los procesos", requiere que un flujo de datos contenga sólo los datos que son verdaderamente necesarios para el proceso receptor. Al asegurar que los procesos reciben sólo datos que realmente necesitan, simplificamos la interfaz entre esos procesos. Para practicar la conservación de datos, debemos definir con precisión la composición de datos de cada flujo de datos (no compuesto). La composición de datos se expresa en forma de *estructuras de datos*.

#### conservación de datos

Práctica de asegurar que un flujo de datos contenga solamente los datos necesarios para el proceso receptor.



**FIGURA 8.9** Flujos ilegales de datos

**atributo de datos** La unidad fundamental de datos que tiene significado para los usuarios y la empresa.

**estructura de datos** Disposición específica de atributos de datos que definen a una instancia individual de un flujo de datos.

**Estructuras de datos** Finalmente, un flujo de datos contiene artículos de datos designados *atributos*. Un **atributo de datos** es la pieza mínima de información que tiene significado para los usuarios finales y el negocio. (Esta definición también se aplica a los *atributos* tal como se presentaron en el capítulo 7.) Los atributos de muestra para la orden de flujo de datos podrían incluir número de orden, fecha de orden, número del cliente, dirección de envío (que consta de atributos tales como dirección de la calle, ciudad y código postal), número de productos ordenados, cantidad(es) ordenada(s), etc. Observe que algunos atributos ocurren una vez para cada instancia de orden, mientras los otros pueden ocurrir varias veces para una sola instancia de orden.

Los atributos de datos que comprenden un flujo de datos se organizan en **estructuras de datos**. Los flujos de datos pueden describirse en términos de los siguientes tipos de estructuras de datos:

- Una *secuencia* o grupo de atributos de datos que ocurren uno tras otro.
- La *selección* de uno o más atributos de un conjunto de atributos.
- La *repetición* de uno o más atributos.



**FIGURA 8.10** Estructura de datos para un flujo de datos

Estructura de datos	Interpretación en inglés
<p>ORDEN =            NÚMERO DE ORDEN +            FECHA DE LA ORDEN +            [NÚMERO PERSONAL DEL CUENTE,              NÚMERO DE LA CUENTA              CORPORATIVA]            DIRECCIÓN DE ENVÍO = DIRECCIÓN +            (DIRECCIÓN DE COBRO = DIRECCIÓN) +            1 {NÚMERO DE PRODUCTO +              DESCRIPCIÓN DE PRODUCTO +              CANTIDAD ORDENADA +              PRECIO DEL PRODUCTO +              FUENTE DEL PRECIO DEL PRODUCTO +              PRECIO AMPLIADO} N +            SUMA DE LOS PRECIOS AMPLIADOS +            CANTIDAD PREPAGADA            (NÚMERO DE LA TARJETA DE CRÉDITO +              FECHA DE VENCIMIENTO)            (NÚMERO DE COTIZACIÓN)</p> <p>DIRECCIÓN =            (APARTADO POSTAL) +            DIRECCIÓN DE LA CALLE +            CIUDAD +            [MUNICIPIO EN EL ESTADO] +            (PAÍS) +            CÓDIGO POSTAL</p>	<p>Un ejemplo de orden consta de:            NÚMERO DE LA ORDEN y            FECHA DE LA ORDEN y            Ya sea NÚMERO PERSONAL DEL CLIENTE              o NÚMERO DE LA CUENTA CORPORATIVA            y DIRECCIÓN DE ENVÍO (que equivale a            DIRECCIÓN)            y en forma optativa: DIRECCIÓN DE COBRO (que            equivale a DIRECCIÓN)            y uno o más ejemplos de:              NÚMERO DE PRODUCTO y              DESCRIPCIÓN DEL PRODUCTO y              CANTIDAD ORDENADA y              PRECIO DEL PRODUCTO y              FUENTE DEL PRECIO DEL PRODUCTO y PRECIO              AMPLIADO            y SUMA DE LOS PRECIOS AMPLIADOS y            CANTIDAD PREPAGADA y            Opcionalmente: tanto el NÚMERO DE LA TARJETA            DE CRÉDITO como la FECHA DE VENCIMIENTO            y opcionalmente: NÚMERO DE COTIZACIÓN</p> <p>Un ejemplo de dirección consta de:            opcionalmente: APARTADO POSTAL y DIRECCIÓN            DE LA CALLE Y CIUDAD y            ya sea ESTADO O MUNICIPIO            y opcionalmente: PAÍS            y CÓDIGO POSTAL</p>

La notación más común para las estructuras de datos es una notación algebraica booleana, que es requerida por muchas herramientas CASE. Otras herramientas y metodologías CASE soportan notaciones que están patentadas pero que en esencia son equivalentes. Una estructura de datos de muestra para la ORDEN de flujo de datos se presenta en la figura 8.10. Esta notación algebraica usa los siguientes símbolos:

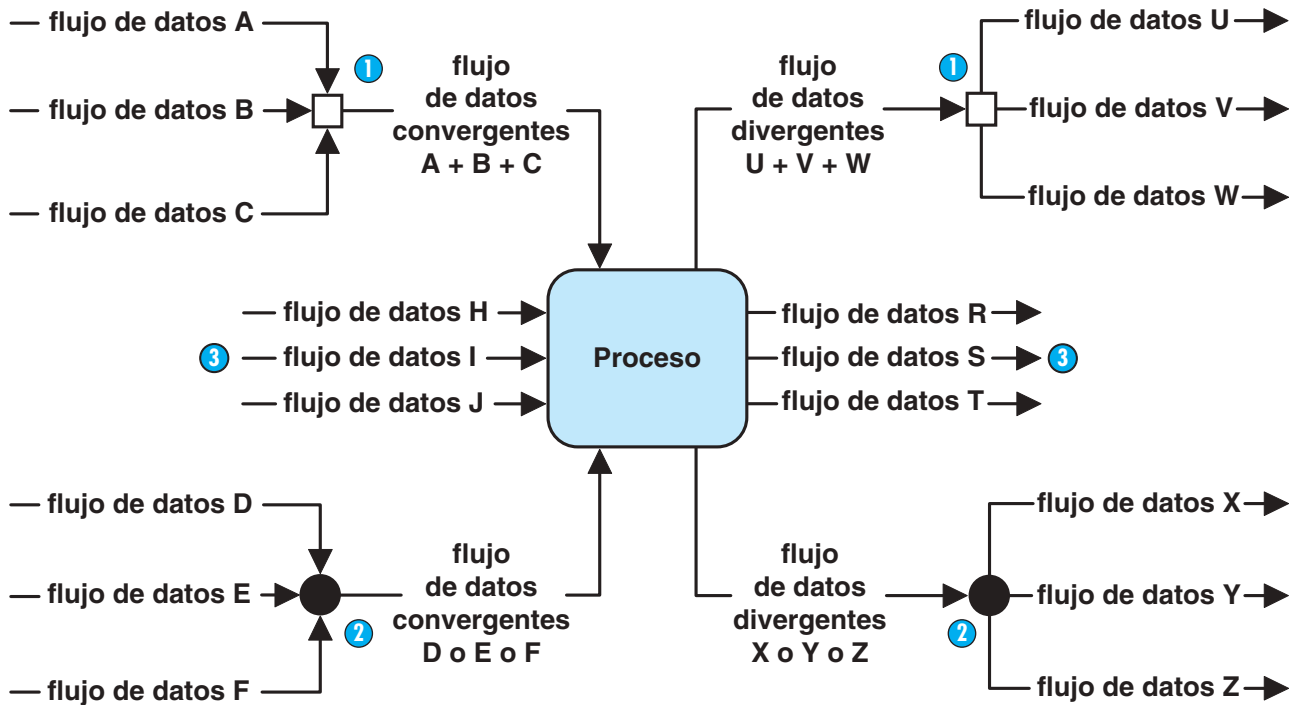
- = significa “consistente en” o “está compuesta de”.
- + significa “y” y designa a una *secuencia*.
- [...] significa que “solamente uno de los atributos dentro de los corchetes puede estar presente”; designa la *selección*. Los atributos en los corchetes están separados por comas.
- {...} significa que los atributos en las llaves pueden ocurrir muchas veces para una instancia del flujo de datos; designa la *repetición*. Los atributos dentro de las llaves se separan con comas.
- (...) significa que el(los) atributo(s) en los paréntesis son opcionales (no hay valor) para algunas instancias del flujo de datos.

En nuestra experiencia, todos los flujos de datos pueden describirse en términos de estas construcciones fundamentales. La figura 8.11 demuestra cada uno de las construcciones fundamentales usando ejemplos. Regresando a la figura 8.10 observe que las construcciones se combinan para describir el contenido de datos del flujo de datos.

**FIGURA 8.11** Construcciones de estructuras de datos

<p><b>Estructura de datos</b></p>	<p><b>Formato por ejemplo</b> (la parte relevante está en <b>negritas</b>)</p>	<p>Interpretación (la parte relevante está en <b>negritas</b>)</p>
<p><b>Secuencia de atributos.</b> La secuencia de la estructura de datos indica uno o más atributos que pueden (o deben) incluirse en un flujo de datos.</p>	<p>DECLARACIÓN DE SALARIO E IMPUESTOS =  <b>NÚMERO DE IDENTIFICACIÓN DEL CAUSANTE</b> +  <b>NOMBRE DEL CAUSANTE</b> +  <b>DIRECCIÓN DEL CAUSANTE</b> +  <b>SALARIOS, PROPINAS, Y COMPENSACIÓN</b> +  <b>IMPUESTO FEDERAL RETENIDO</b> + ...</p>	<p>Un ejemplo de DECLARACIÓN DE SALARIO E IMPUESTOS consta de:  <b>NÚMERO DE IDENTIFICACIÓN DEL CAUSANTE Y</b>  <b>NOMBRE DEL CAUSANTE Y</b>  <b>DIRECCIÓN DEL CAUSANTE Y</b>  <b>SALARIOS, PROPINAS Y COMPENSACIÓN Y</b>  <b>RETENCIÓN DE IMPUESTO FEDERAL Y...</b></p>
<p><b>Selección de atributos.</b> La selección de la estructura de datos le permite mostrar situaciones donde diferentes conjuntos de atributos describen instancias diferentes del flujo de datos.</p>	<p>ORDEN =  <b>(NÚMERO PERSONAL DEL CLIENTE, NÚMERO DE LA CUENTA CORPORATIVA)</b> +          FECHA DE LA ORDEN + ...</p>	<p>Un ejemplo de ORDEN consta de:  <b>Ya sea NÚMERO PERSONAL DEL CLIENTE o NÚMERO DE LA CUENTA CORPORATIVA;</b> y          FECHA DE LA ORDEN y...</p>
<p><b>Repetición de atributos.</b> La estructura de datos repetitiva se usa para marcar un atributo de datos o un grupo de atributos de datos que pueden (o deben) repetirse un número especificado de veces para una sola instancia del flujo de datos.          El número mínimo de repeticiones generalmente es cero o uno.          El número máximo de repeticiones puede especificarse como "n" que implica "muchos" donde el número real de instancias varía para cada instancia del flujo de datos.</p>	<p>RECLAMACIÓN =          NÚMERO DE PÓLIZA +          NOMBRE DEL TENEDOR DE LA PÓLIZA +  <b>O{NOMBRE DEL DEPENDIENTE +</b>  <b>RELACIÓN DEL DEPENDIENTE} N +</b>  <b>1 {DESCRIPCIÓN DEL GASTO +</b>  <b>PROVEEDOR DEL SERVICIO +</b>  <b>CANTIDAD DEL GASTO} N</b></p>	<p>Un ejemplo de RECLAMACIÓN consta en:          NÚMERO DE PÓLIZA          NOMBRE DEL TENEDOR DE LA PÓLIZA          DIRECCIÓN DEL TENEDOR DE LA PÓLIZA  <b>cero o más ejemplos de:</b>  <b>NOMBRE DEL DEPENDIENTE Y</b>  <b>RELACIÓN DEL DEPENDIENTE Y</b>  <b>uno o más ejemplos de:</b>  <b>DESCRIPCIÓN DEL GASTO Y</b>  <b>PROVEEDOR DEL SERVICIO Y</b>  <b>CUENTA DE GASTOS</b></p>
<p><b>Atributos opcionales.</b> La notación opcional indica que un atributo, o grupo de atributos en una secuencia o una selección de estructura de datos, tal vez no se incluya en todas las instancias de un flujo de datos.          Nota: Para la estructura de datos repetitiva, un mínimo de "cero" es lo mismo que hacer "optativo" al grupo repetitivo completo.</p>	<p>RECLAMACIÓN =          NÚMERO DE PÓLIZA +          NOMBRE DEL TENEDOR DE LA PÓLIZA +          DIRECCIÓN DEL TENEDOR DE LA PÓLIZA +  <b>(NOMBRE DEL CÓNYUGE +</b>  <b>FECHA DE NACIMIENTO) + ...</b></p>	<p>Un ejemplo de RECLAMACIÓN consta de:          NÚMERO DE LA PÓLIZA Y          NOMBRE DEL TENEDOR DE LA PÓLIZA Y          DIRECCIÓN DEL TENEDOR DE LA PÓLIZA Y  <b>opcionalmente, NOMBRE DEL CÓNYUGE Y</b>  <b>FECHA DE NACIMIENTO Y...</b></p>
<p><b>Atributos reusablees.</b> Para grupos de atributos que están contenidos en muchos flujos de datos, es conveniente crear una estructura de datos separada que pueda reusarse en otras estructuras de datos.</p>	<p>FECHA =          MES +          DÍA +          AÑO</p>	<p>Entonces, las estructuras reusablees pueden incluirse en otras estructuras de flujo de datos como sigue:          ORDEN = NÚMERO DE ORDEN... + FECHA          FACTURA = NÚMERO DE FACTURA... + FECHA          PAGO = NÚMERO DE CLIENTE ... + FECHA</p>





**FIGURA 8.12** Flujos de datos divergentes y convergentes

La importancia de definir las estructuras de datos para cada flujo de datos debería ser evidente: ¡usted está definiendo los requisitos de datos de negocios para cada entrada y salida! Estos requisitos deben determinarse antes de que cualquier proceso pueda implementarse como un programa de computadora. Esta notación estándar provee una manera simple pero efectiva para comunicarse entre usuarios finales y programadores.

**Dominios** Un atributo es una pieza de información. Al analizar un sistema, tiene sentido que definamos esos valores para un atributo de modo que sean legítimos, o que tengan sentido. Los valores para cada atributo se definen en términos de dos propiedades: tipo de datos y dominio. El **tipo de datos** para un atributo define qué clase de datos pueden almacenarse en ese atributo, mientras que el **dominio** de un atributo define qué valores puede adoptar legítimamente un atributo. Los conceptos de tipo de datos y de dominio se introdujeron en el capítulo 7. Vea esa discusión y las tablas 7.1 y 7.2 para una descripción más amplia del tipo de datos y del dominio.

**Flujos de datos divergentes y convergentes** Algunas veces es útil esquematizar los flujos de datos divergentes o convergentes en un diagrama de flujo de datos. Los **flujos de datos divergentes** señalan que todo o parte de un flujo de datos individual se encaminan a diferentes destinos.<sup>2</sup> Un **flujo de datos convergente** es el que incorpora flujos de datos múltiples en un flujo de datos individual. Los flujos de datos convergentes señalan que los flujos de datos provenientes de diferentes fuentes pueden (deben) agruparse como un paquete individual para el procesamiento subsiguiente.

Los flujos de datos divergentes y convergentes se esquematizan como se muestra en la figura 8.12. Observe que no incluimos un proceso para “encaminar” los flujos. Éstos simplemente divergen o convergen en un flujo común. Las siguientes notaciones, que no están soportadas por todas las herramientas de CASE, se usan en este libro:

**tipo de dato** Clase de datos que pueden almacenarse en un atributo.

**dominio** Valores legítimos de un atributo.

**flujo de datos divergente** Un flujo de datos que consiste de otros flujos de datos.

**flujo de datos convergente** La fusión de varios flujos de datos en uno solo.

<sup>2</sup> Algunos expertos sugieren que los flujos de datos divergentes deberán usarse solamente cuando todos los datos del flujo se encaminen a todos los destinos. Preferimos la definición clásica de DeMarco que permite que todo el flujo o parte de éste se encamine a diferentes procesos.

- ① El cuadrado pequeño significa “y”. Esto quiere decir que cada vez que el proceso se realiza, debe alimentar (u obtener) todos los flujos de datos divergentes o convergentes. (*Algunas notaciones DFD simplemente colocan un + entre los flujos de datos.*)
- ② El empalme pequeño de color negro significa “excluyente o”. Esto quiere decir que cada vez que el proceso se realiza, debe alimentar (u obtener) sólo uno de los flujos de datos divergentes o convergentes. (*Algunas notaciones DFD simplemente colocan un \* entre los flujos de datos.*)
- ③ A falta de un flujo de datos divergente o convergente, el lector debería asumir uno “incluyente o”. Esto quiere decir que cada vez que el proceso se realiza puede alimentar a cualquiera o a todos los flujos de datos esquematizados.

Con las reglas anteriormente citadas puede esquematizarse lo más complejo de los procesos de negocios y las combinaciones de flujo de datos.

## El proceso de modelación lógica de procesos

Ahora que usted entiende los conceptos básicos de modelos de proceso podemos examinar cómo construir un modelo de proceso. ¿Cuándo lo hace usted? ¿Cuántos modelos de proceso pueden esquematizarse? ¿Qué tecnología existe para soportar el desarrollo de los modelos de proceso?

### > Planeación estratégica de sistemas

Muchas organizaciones seleccionan proyectos de desarrollo de aplicación basados en los planes estratégicos de sistemas de información. La planificación estratégica es un proyecto separado que produce un plan de estrategia de sistemas de información que define una visión global y la arquitectura para los sistemas de información. Esta arquitectura con frecuencia incluye un *modelo de proceso de la empresa*. (El plan por lo general tiene otros componentes arquitectónicos que no son importantes para este debate.)

Un modelo de proceso de la empresa por lo común identifica sólo áreas y funciones de negocios. Los eventos y los procesos detallados raramente se examinan. Las áreas de negocios y las funciones se identifican y se mapean a otros modelos de la empresa, como el modelo de datos de la empresa (capítulo 7). Las áreas de negocios y las funciones se priorizan subsecuentemente en proyectos de desarrollo de aplicación. Las prioridades por lo general se basan en qué áreas de negocios, funciones y aplicaciones de soporte le devolverán el mayor valor al negocio como un todo.

Un modelo de proceso de la empresa se almacena en un depósito corporativo. Posteriormente, a medida que se inician los proyectos de desarrollo de aplicación, los subconjuntos del proceso de la empresa se exportan a los equipos de proyecto para servir como punto de partida para construir modelos de proceso más detallados (incluyendo los diagramas de flujo de datos). Una vez que el equipo de proyecto completa el análisis y el diseño de los sistemas, los modelos de proceso expandidos y refinados se devuelven al depósito corporativo.

### > Modelado de procesos para el rediseño de los procesos del negocio

El rediseño de un proceso de negocios (BPR, por sus siglas en inglés) se ha discutido varias veces en este libro y en este capítulo. Recuerde que los proyectos BPR analizan los procesos de negocios y luego los rediseñan para eliminar ineficiencias y burocracias antes de cualquier (re)aplicación de información tecnológica. Para rediseñar los procesos de negocios primero debemos estudiar los procesos existentes. Los modelos de proceso juegan un papel integral en BPR.

Cada metodología BPR recomienda sus propias notaciones de modelo de proceso y documentación. La mayor parte de los modelos son un cruce entre diagramas de flujo de datos y diagramas de flujo. Los diagramas tienen tendencia a ser muy *físicos* porque el equipo BPR está tratando de aislar las idiosincrasias de implementación que causan ineficiencia y reducen valor. Los diagramas de flujo de datos/diagramas de flujo de BPR pue-

den incluir información y símbolos nuevos para ilustrar la sincronización, la producción, los retrasos, los costos y el valor. Dados estos datos adicionales, el equipo BPR intenta entonces simplificar los procesos y los flujos de datos en un esfuerzo para maximizar la eficiencia y devolver el mayor valor a la organización.

Las oportunidades para el uso eficiente de la tecnología de información también pueden registrarse en los diagramas físicos. Si es así, el diagrama BPR se convierte en una entrada para el análisis de los sistemas (que se describe en seguida).

## > Modelado de procesos durante el análisis de los sistemas

En el análisis de los sistemas y en este capítulo, enfocamos la atención en la modelación de procesos *lógicos* como una parte del análisis de requisitos de negocios. En su marco de sistemas de información, los modelos de procesos lógicos tienen un enfoque del proceso y una perspectiva de dueño del sistema y/o usuario del sistema. Se construyen típicamente como entregables de la fase de análisis de requisitos de un proyecto. Mientras los modelos lógicos de proceso no tengan que ver con los detalles de implementación o tecnología, pueden construirse (a través de *la ingeniería inversa*) a partir del software de aplicación existente, pero esta tecnología es mucho menos madura y confiable que la tecnología correspondiente a la ingeniería inversa de *datos*.

En el apogeo de las metodologías originales de análisis estructurado, la modelación de procesos también se realizaba en la fase de análisis de problemas del análisis de los sistemas. Los analistas construían un *modelo físico de procesos del sistema actual*, un *modelo lógico del sistema actual* y un *modelo lógico del sistema objetivo*. Cada modelo se construía de arriba abajo: desde modelos muy generales hasta modelos muy detallados. Aun cuando conceptualmente es sólido, este enfoque condujo a la sobresaturación de la modelación y a demoras importantes del proyecto, tanto que aun el gurú de técnicas estructuradas Ed Yourdon lo llamó “la parálisis del análisis”.

En la actualidad, la mayoría de las estrategias modernas de análisis estructurado se enfocan exclusivamente en el desarrollo del *modelo lógico del sistema objetivo*. En lugar de estar construido ya sea de arriba abajo o de abajo hacia arriba, se organiza según una estrategia de sentido común llamada *partición de eventos*. La **partición de eventos** divide un sistema en subsistemas basado en eventos de negocios y en las respuestas a estos eventos. Esta estrategia para la modelación de procesos impulsada por los eventos se ilustra en la figura 8.13 y se describe como sigue:

- 1 Se construye un **diagrama de flujo de datos de contexto** del sistema para establecer el alcance *inicial* del proyecto. Este diagrama simple de flujo de datos de una página muestra solamente las interfaces principales del sistema con su ambiente.
- 2 Se dibuja un **diagrama de descomposición funcional** para separar el sistema en los subsistemas y/o las funciones lógicas. (Este paso se omite para los sistemas muy pequeños.)
- 3 Se compila una **lista de respuesta a los eventos** o de **casos de uso** para identificar y confirmar los eventos de negocios para los cuales el sistema debe proveer una respuesta. La lista también describirá las respuestas requeridas o posibles para cada evento.
- 4 Un proceso, designado un **manejador o administrador de eventos**, se añade al diagrama de descomposición para cada evento. El diagrama de descomposición ahora sirve de contorno para el sistema.
- 5 En forma opcional, un **diagrama de eventos** se construye y se valida para cada evento. Este diagrama simple de flujo de datos muestra sólo el manejador o administrador de acontecimiento y las entradas y las salidas para cada evento.
- 6 Se construyen uno o más **diagramas del sistema** mancomunando los diagramas de eventos. Estos diagramas de flujo de datos muestran el “panorama” del sistema.
- 7 Los **diagramas elementales** se construyen para los procesos de eventos que requieren detalles adicionales de procesamiento. Estos diagramas de flujo de datos muestran todos los procesos elementales, los almacenes de datos y flujos de datos para los eventos individuales. 8 La lógica de cada proceso elemental y 9 la estructura de datos de cada flujo de datos elementales se describen usando las herramientas descritas anteriormente en el capítulo.

---

### partición de eventos

Estrategia de análisis estructurado en la cual un sistema se divide en subsistemas en base a los eventos de negocios y en las respuestas a esos eventos.

---

### diagrama de flujo de datos de contexto

Diagrama que muestra al sistema como una “caja negra” y sus principales interfaces como su ambiente.

---

### diagrama de descomposición funcional

Diagrama que divide al sistema en funciones y subsistemas lógicos.

---

### lista de respuestas a los eventos/casos de uso

Lista de los eventos de negocios para los cuales el sistema debe proveer una respuesta. Similar a una lista de casos de uso.

---

### manejador de eventos

Proceso que manipula un evento dado en la lista de eventos-respuestas.

---

### diagrama de eventos

Un diagrama de flujo de datos para manejar un solo evento, así como los agentes y almacenes de datos que proporcionan entradas o reciben salidas.

---

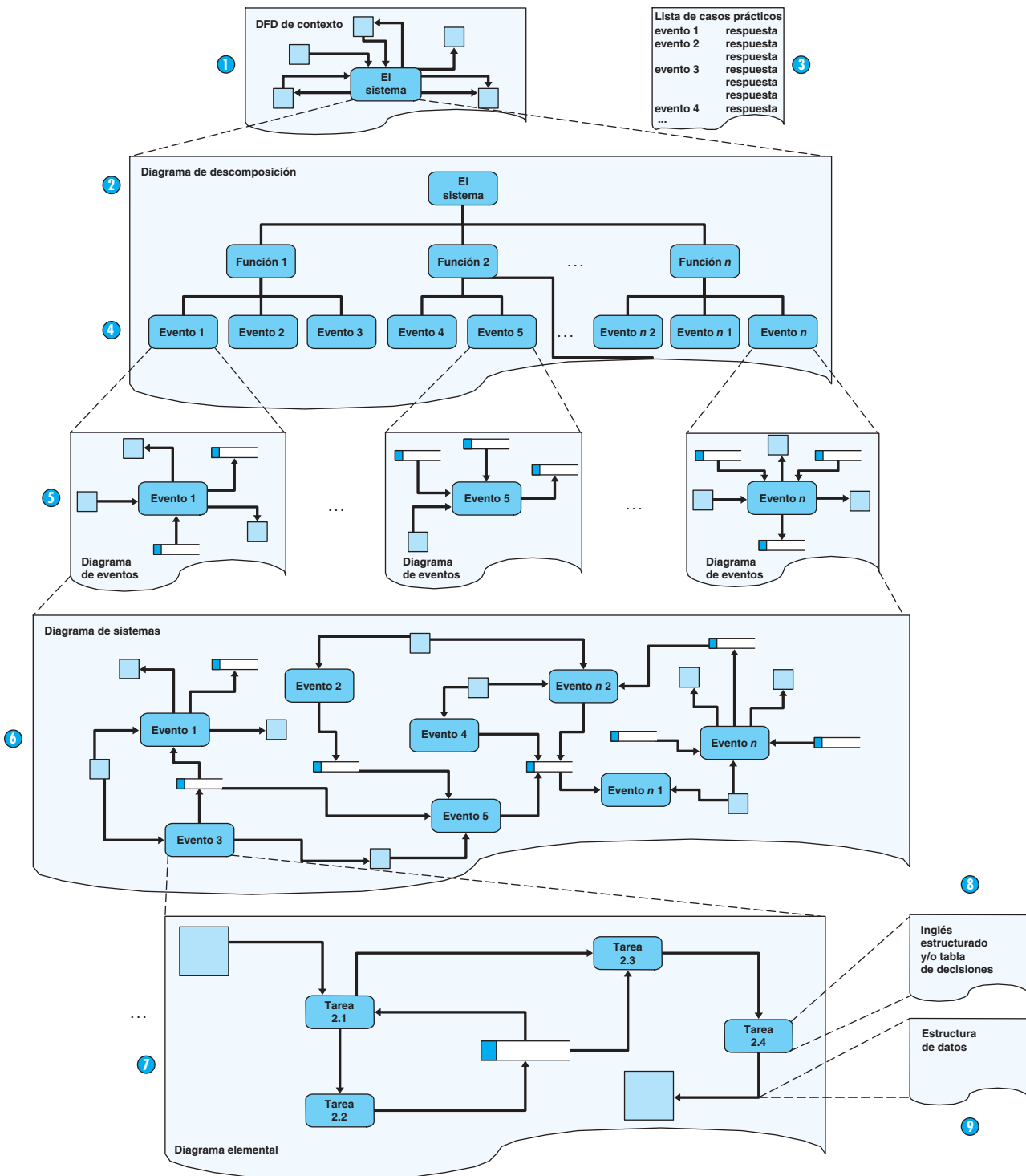
### diagrama del sistema

Diagrama de flujo de datos que fusiona los diagramas de eventos de todo un sistema o parte del él.

---

### diagrama elemental

Diagrama de flujo de datos que ilustra los procesos elementales, los almacenes de datos y los flujos de datos de un evento individual.



**FIGURA 8.13** Estrategia de modelado de procesos impulsados por los eventos

Los modelos de procesos anteriormente citados documentan colectivamente todos los requisitos de procesamiento de negocios para un sistema. Demostraremos la técnica en nuestro estudio del caso SoundStage.

El modelo lógico de procesos del análisis de sistemas describe los requisitos del procesamiento de negocios del sistema, no soluciones técnicas. Recuerde del capítulo 4 que el propósito de la fase de análisis de decisión debe determinar la mejor forma de implementar esos requisitos con tecnología. En la práctica, esta decisión ya pudo haber sido

estandarizada como parte de una arquitectura de aplicación. Por ejemplo, la arquitectura de aplicación de SoundStage requiere que el equipo de desarrollo determine si puede comprarse un sistema aceptable. En caso de que no, la arquitectura de aplicación actual especifica que el software construido internamente sea escrito ya sea en *Visual Basic .NET* o *C++* de la Corporación Microsoft.

## > Mirando hacia delante al diseño de sistemas

Durante el diseño de sistemas, el modelo lógico de procesos se transformará en un modelo físico de procesos (llamado *esquema de aplicación*) para la arquitectura técnica escogida. Este modelo reflejará las limitaciones y capacidades técnicas de la tecnología escogida. Cualquier discusión adicional del proceso físico/diseño de aplicación se difiere hasta el capítulo 11.

## > Descubrimiento de hechos y recopilación de información para el modelado de procesos

Los modelos de proceso no pueden construirse sin hechos e información apropiados, tal como los provee la comunidad de usuarios. Estos hechos pueden recolectarse a través de varias técnicas, tales como la toma de muestras de formas y archivos existentes, la investigación de sistemas similares, encuestas y entrevistas de usuarios y de la gerencia. El método más rápido para recolectar los hechos y la información en forma simultánea para construir y verificar los modelos de procesos es la *Planeación Conjunta de Requisitos* (Joint Requirements Planning, JRP). JRP utiliza un grupo de encuentro que facilita la recolección de los hechos, construcción y verificación de los modelos; generalmente en una o dos sesiones de un día completo.

El descubrimiento de los hechos, la recopilación de información y las técnicas JRP se exploraron en el capítulo 5.

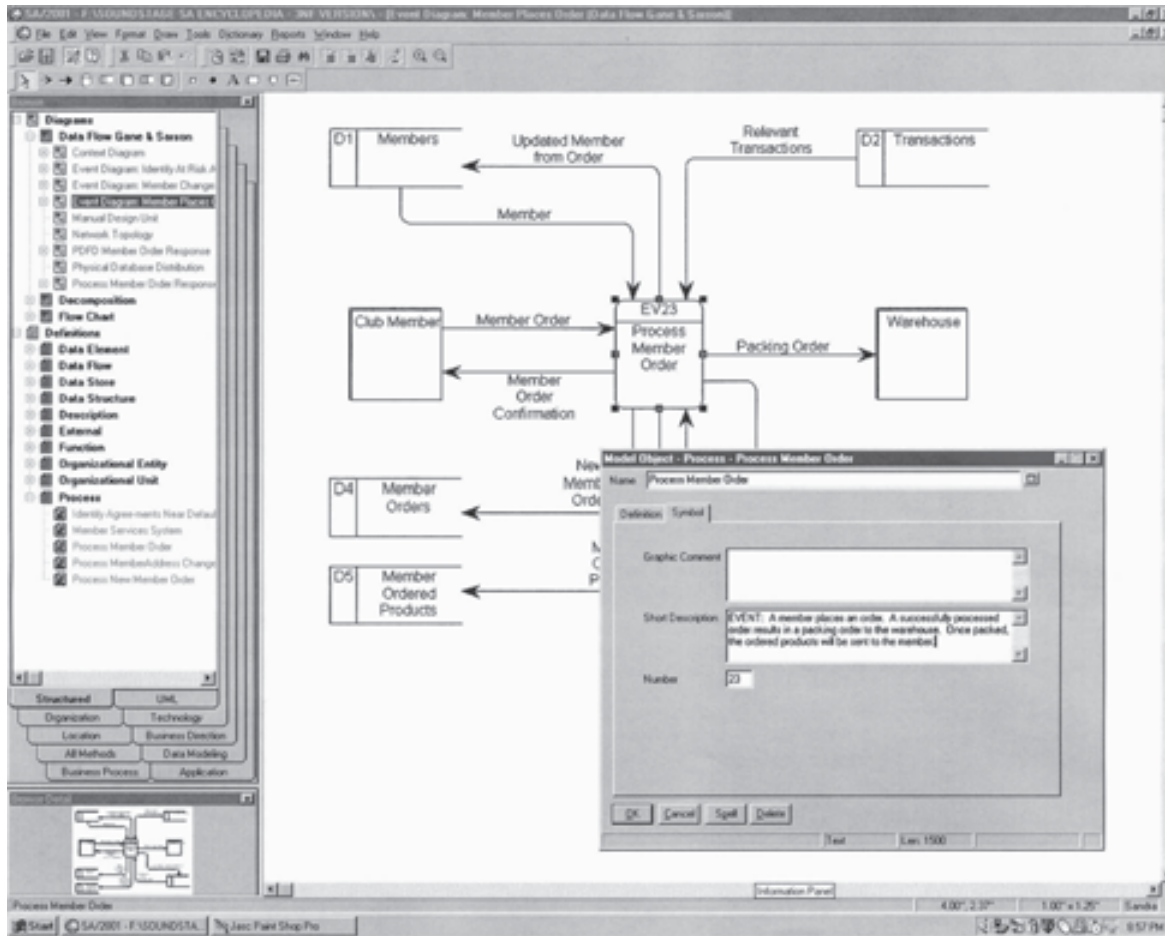
## > La ingeniería de sistemas asistida por computadora (CASE) para la modelación de procesos

Al igual que todos los modelos de sistemas, los modelos de procesos se almacenan en un depósito. La tecnología de la ingeniería de sistemas asistida por computadora, introducida en el capítulo 3, provee el depósito para almacenar el modelo de proceso y sus descripciones detalladas. La mayoría de los productos de CASE soportan la modelación de procesos asistida por computadora. La mayoría soporta los diagramas de descomposición y los diagramas de flujo de datos. Algunas soportan extensiones para el análisis y rediseño de los procesos de negocios.

Al usar un producto CASE, usted puede crear fácilmente modelos de procesos profesionales, legibles sin usar papel, lápiz, borrador y plantillas. Los modelos pueden modificarse en forma sencilla para reflejar las correcciones y los cambios propuestos por los usuarios finales. Además, la mayoría de los productos de CASE proveen herramientas analíticas poderosas que pueden comprobar sus modelos en cuanto a errores mecánicos, la totalidad y la consistencia. Algunos productos de CASE aun le pueden ayudar a analizar el modelo de los datos en cuanto a la consistencia, totalidad y flexibilidad. Los ahorros de tiempo potenciales y la calidad son sustanciales.

Las herramientas de CASE ciertamente tienen sus limitaciones. No todas las convenciones de los modelos de proceso son soportadas por todos los productos de CASE. Por consiguiente, cualquier producto CASE puede obligar a la compañía a adaptar sus símbolos o su enfoque de la metodología de modelación de procesos, de modo que se pueda trabajar dentro de las limitaciones de la herramienta de CASE.

Todos los modelos de procesos de SoundStage de la siguiente sección de este capítulo fueron creados con la herramienta CASE de Popkin llamada, *System Architect 2001*. Para el estudio de caso, le proveemos de las salidas impresas exactamente como salieron de nuestras impresoras. Las únicas modificaciones del artista fueron las viñetas ocasionales que llaman su atención para elementos específicos de interés en las salidas impresas. Todos los procesos, los flujos de datos, los almacenes de datos y las fronteras en los modelos de procesos de SoundStage fueron automáticamente catalogados en el repositorio del proyecto de *System Architect* (al cual le llama enciclopedia). La figura 8.14 ilustra algunas pantallas del *System Architect* tal como se usan para el modelado de datos.



**FIGURA 8.14** CASE para el modelado de procesos (usando el *System Architect 2001* de Popkin Software and Systems)

## Cómo construir modelos de procesos

Como un analista de sistemas o un usuario final informado, usted debe aprender a dibujar diagramas de descomposición y de flujo de datos para modelar los requisitos de los procesos de negocios. Usaremos el proyecto SoundStage Entertainment Club para enseñarle cómo dibujar estos modelos de procesos.

Spongamos que las fases preliminares de investigación y de análisis de problemas del proyecto han sido completadas y el equipo del proyecto entiende las fortalezas, las debilidades, las limitaciones, los problemas, las oportunidades y las restricciones del sistema actual. El equipo también ya ha construido el modelo de datos (en el capítulo 7) para documentar los requisitos de los datos de negocios para el sistema nuevo. Los miembros del equipo ahora construirán los modelos correspondientes de proceso.

### > El diagrama de flujo de datos de contexto

Primero necesitamos documentar el alcance inicial del proyecto. Todos los proyectos tienen un alcance, el cual define qué aspecto del negocio deberá soportar un sistema o aplicación y cómo el sistema que se está modelando debe interactuar con otros sistemas y el negocio como un todo. En su marco de sistemas de información, el alcance se define como el enfoque de la comunicación desde la perspectiva de los dueños del



SISTEMA. ESTÁ DOCUMENTADO CON UN **diagrama de flujo de datos de contexto**. Ya que el alcance de cualquier proyecto siempre está sujeto al cambio, el diagrama de contexto está también sujeto al cambio constante. Un sinónimo es *modelo del medio ambiente* [Yourdon, 1990].

Sugerimos la siguiente estrategia para documentar la demarcación del sistema y el alcance:

1. Piense acerca del sistema como un envase para distinguir el interior del exterior. Ignore los funcionamientos interiores del envase. Algunas veces a esto se le llama el pensamiento de la “caja negra”.
2. Pregunte a sus usuarios finales a qué transacciones de negocios debe responder un sistema. Éstas son las *entradas netas* al sistema. Para cada entrada neta determine su fuente. Las fuentes se convertirán en *agentes externos* en el diagrama de flujo de datos de contexto.
3. Pregunte a sus usuarios finales qué respuestas deben ser producidas por el sistema. Éstas son las salidas netas del sistema. Para cada salida neta determine su destino. Los destinos también se convertirán en *agentes externos*. Los requisitos de reportes y averiguaciones pueden obstaculizar el diagrama rápidamente. Considere consolidarlos en flujos de datos complejos.
4. Identifique cualesquiera almacenes de datos *externos*. Muchos sistemas requieren acceso a los archivos o las bases de datos de otros sistemas. Pueden usar los datos en esos archivos o bases de datos. Algunas veces pueden actualizar ciertos datos en esos archivos y bases de datos. Pero generalmente no se les permite cambiar la estructura de esos archivos y bases de datos; por consiguiente, están fuera del alcance de proyecto.
5. Dibuje un diagrama de contexto a partir de toda la información precedente.

Si usted trata de incluir todas las entradas y las salidas entre un sistema y el resto del negocio y el mundo exterior, un diagrama típico de flujo de datos de contexto podría mostrar hasta 50 o más flujos de datos. Tal diagrama tendría un valor de comunicación bajo, si lo tuviera. Por consiguiente, sugerimos que usted muestre sólo aquellos flujos de datos que representen el principal objetivo o las entradas y salidas más importantes del sistema. Difiera los flujos de datos menos comunes para que los DFD más detallados se dibujen posteriormente.

El diagrama de flujo de datos de contexto contiene a un proceso y solamente uno (vea la figura 8.15). Algunas veces, este proceso se identifica con el número “0”; sin embargo, nuestra herramienta CASE no permitía esto. Los agentes externos se dibujan alrededor del perímetro. Los flujos de datos definen las interacciones de un sistema con las fronteras y con los almacenes externos de datos.

Como se muestra en el diagrama de flujo de datos de contexto, el propósito principal del sistema es procesar las SUSCRIPCIONES NUEVAS en respuesta a las OFERTAS DE SUSCRIPCIÓN, crear PROMOCIONES NUEVAS para productos, y responder a las ÓRDENES de un MIEMBRO enviando ÓRDENES de EMBALAJE al almacén para que se llenen. (Observe que hicimos que todos los nombres de los flujos de datos fueran singulares.) La gerencia también ha enfatizado la necesidad de REPORTE DIVERSOS. Finalmente, las extensiones de Web para este sistema requieren que el sistema provea a los miembros con RESPUESTAS DIVERSAS de AVERIGUACIÓN en relación con las órdenes y las cuentas.

## > El diagrama de descomposición funcional

Recuerde que un diagrama de descomposición muestra la descomposición funcional de arriba abajo o la estructura de un sistema. También nos provee de los inicios de un contorno para dibujar nuestros diagramas de flujo de datos.

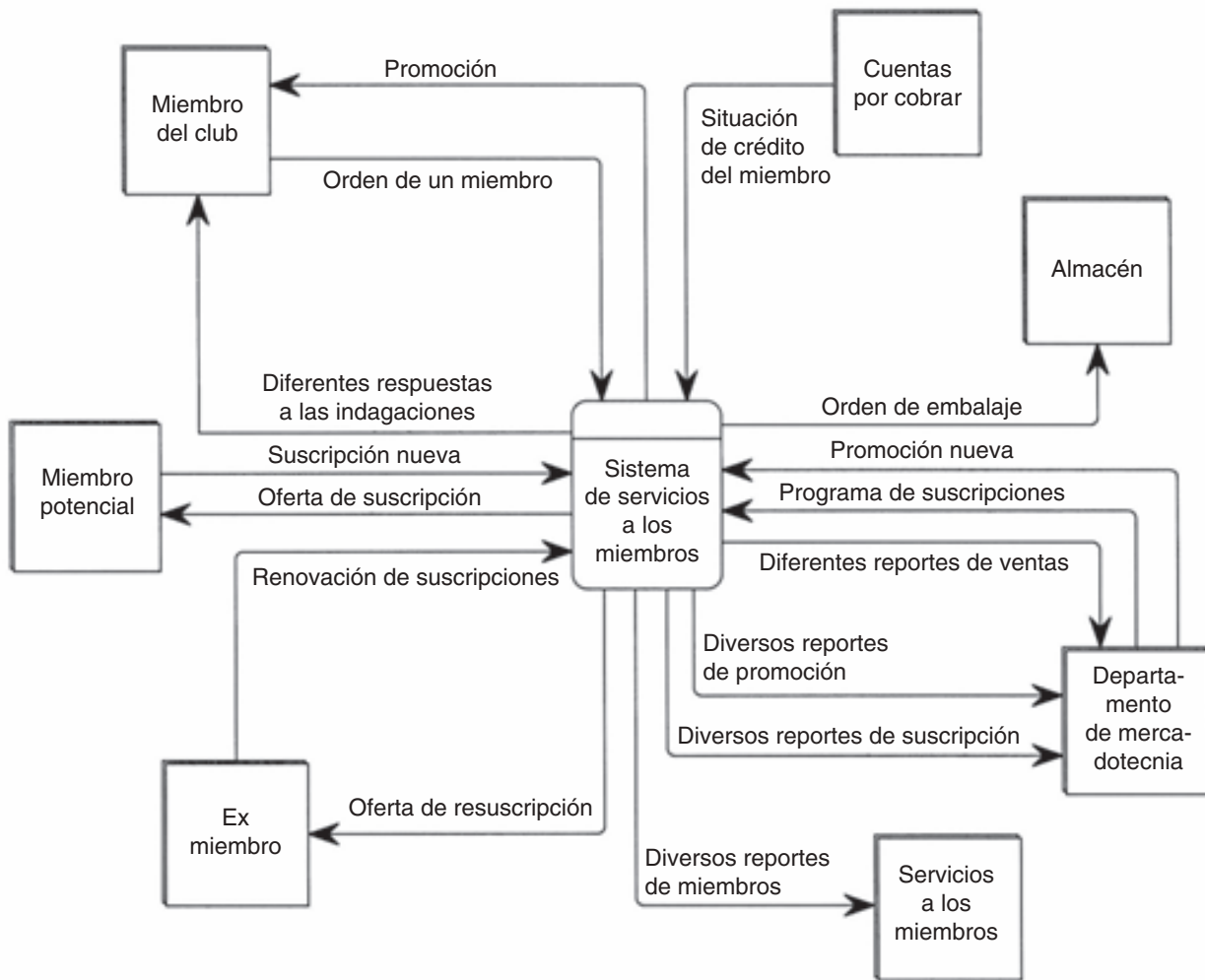
La figura 8.16 es el diagrama de descomposición funcional del proyecto SoundStage. Estudiemos este diagrama. Primero, observe que los procesos se esquematizan como rectángulos, no rectángulos redondeados. Ésta es meramente una limitación de la implementación de nuestra herramienta de CASE de los diagramas de descomposición; usted también puede tener que adaptarse a su herramienta de CASE.

## diagrama de flujo de

## datos de contexto

Modelo de procesos usado para documentar el alcance de un sistema. También llamado modelo del medio ambiente.





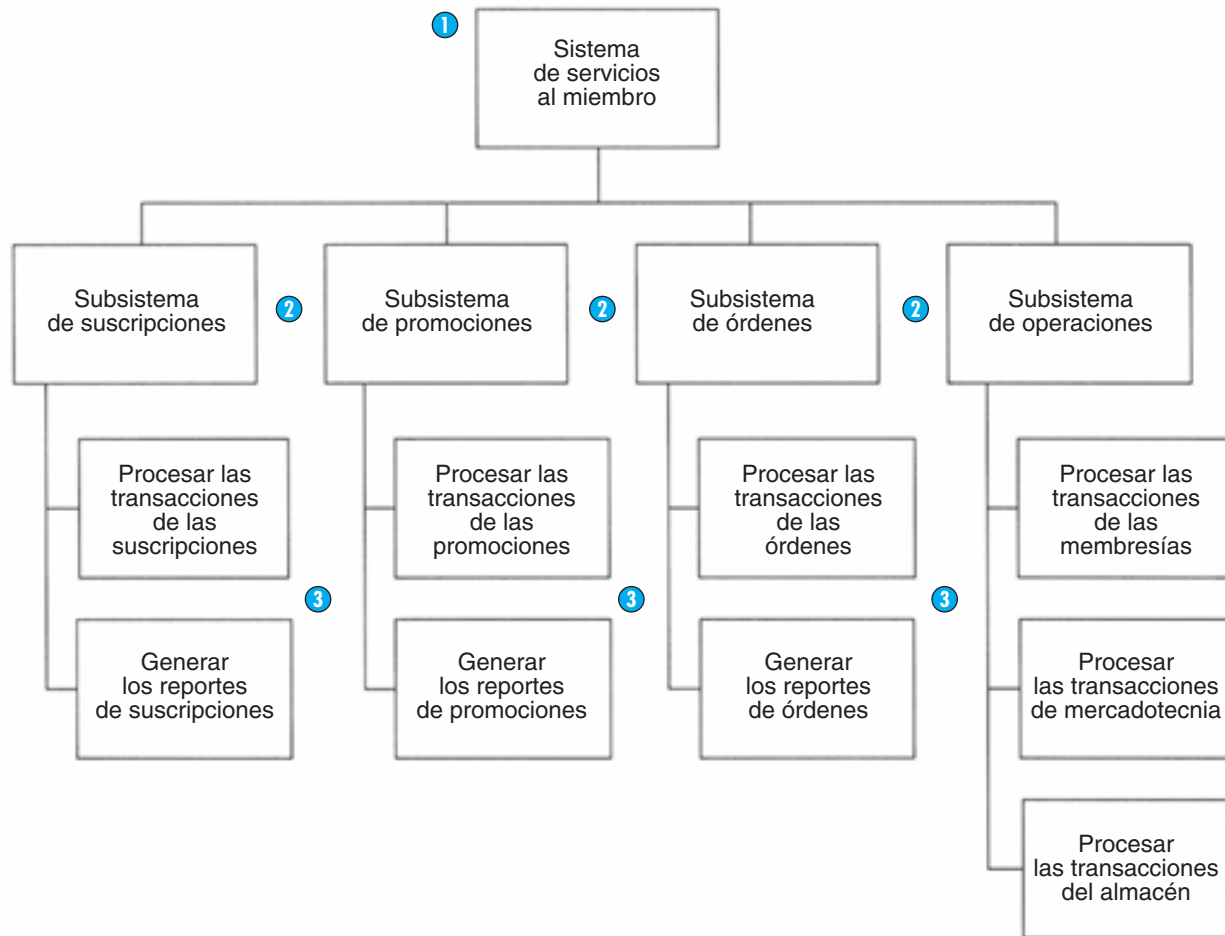
**FIGURA 8.15** Diagrama de flujo de datos de contexto (creado con el *System Architect 2001*)

La siguiente es una discusión elemento por elemento del diagrama de descomposición. Los números en un círculo corresponden a puntos específicos de interés en el diagrama.

- ① El proceso raíz corresponde al sistema entero.
- ② Inicialmente, el sistema se factoriza en subsistemas y/o funciones. Estos subsistemas y estas funciones no necesariamente corresponden a las unidades de organización en un organigrama. Progresivamente, los analistas y los usuarios reciben instrucciones de ignorar las fronteras organizativas y construir sistemas funcionales cruzados que aerodinamizan el procesamiento y los datos compartidos.
- ③ Nos gusta separar los aspectos operacionales y de reporte de un sistema. Así, dividimos cada subsistema consecuentemente. Más tarde, si esta estructura no tuviera sentido, la podemos cambiar.

Los sistemas más grandes primero deben ser divididos en subsistemas y funciones. No hay límite al número de procesos hijo para un proceso padre. Muchos autores solían recomendar un máximo de cinco para nueve procesos por padre, pero cualquier límite es demasiado artificial. ¡En lugar de eso, estructure el sistema de modo que tenga sentido para el negocio!

La factorización de un proceso padre en un proceso hijo individual no tiene sentido. No proveería ningún detalle adicional. Por consiguiente, si un proceso debe dividirse, deberá hacerlo en al menos dos procesos hijo.



**FIGURA 8.16** Diagrama de descomposición funcional (creado con el *System Architect 2001*)

### > Lista de respuesta a los eventos o casos de uso

Después de construir el diagrama de descomposición, determinamos a qué acontecimientos de negocios debe responder el sistema y cuáles respuestas son apropiadas. Los eventos no son difíciles de encontrar. Algunas de las entradas en el diagrama de contexto se asocian con eventos, pero el diagrama de contexto raramente los muestra todos. Esencialmente, los hay de tres tipos:

- Los *acontecimientos externos* se nombran así porque son iniciados por agentes externos. Cuando estos eventos ocurren, se produce un flujo de datos de entrada para el sistema. Por ejemplo, el evento EL CLIENTE COLOCA UNA NUEVA ORDEN se reconoce en forma de la ORDEN de flujo de datos de entrada del agente externo CLIENTE.
- Los *acontecimientos temporales* desencadenan procesos con base en el tiempo, o algo que meramente ocurre. Cuando estos acontecimientos suceden, tiene lugar un *flujo de control* de entrada. Los ejemplos de eventos temporales incluyen el TIEMPO PARA RECORDARLES A LOS CLIENTES QUE PAGUEN LAS FACTURAS PASADAS o el FIN DE MES.
- Los *eventos de estado* desencadenan procesos basados en el cambio de un sistema de un estado o condición a otro. Al igual que los eventos temporales, los eventos de estado se ilustrarán como un *flujo de control* de entrada.

Los sistemas de información generalmente responden en su mayor parte a los eventos externos y temporales. Los eventos de estado por lo regular se asocian con sistemas de tiempo real tales como el control del elevador o del robot.

**caso de uso** Herramienta de análisis para encontrar e identificar los eventos y las respuestas de negocios.

Uno de los enfoques más populares y exitosos para encontrar e identificar acontecimientos y respuestas es una técnica llamada *casos de uso* (capítulo 6) desarrollada por el doctor Ivar Jacobson. Esta técnica está arraigada en el análisis orientado a objetos, pero es fácilmente adaptable a la diagramación de análisis estructurado y de flujo de datos. El análisis del **caso de uso** es el proceso de identificar y modelar los eventos de negocios, quién los inició y cómo les responde el sistema.

Los casos prácticos identifican y describen los procesos de sistema necesarios desde la perspectiva de los usuarios. Cada caso práctico es iniciado por los usuarios o los sistemas externos llamados *actores*. Un actor es cualquier cosa que necesita interactuar con el sistema para intercambiar información y así es análogo a los agentes externos en DFDs.

El diagrama de flujo de datos de contexto identifica a los actores clave como *agentes externos*. También identifica a *algunos* de los casos prácticos. La palabra clave es “algunos”. Recuerde que el diagrama de contexto muestra sólo las entradas y salidas principales de un sistema. Casi siempre hay más entradas y salidas que las que se esquematizan; generalmente muchas más. Algunas de las entradas y salidas esquematizadas son realmente mezclas de muchos tipos y variaciones de esas entradas y salidas (por ejemplo, los “reportes diversos” en nuestro diagrama de contexto). Además, el diagrama de contexto no puede ilustrar las muchas entradas y salidas de excepción como los errores, averiguaciones y seguimientos.

Una forma de expandir los casos de uso es entrevistar a los agentes externos (los actores) esquematizados en el diagrama. Los agentes 1) pueden identificar los eventos (casos de uso) para los cuales creen que el sistema tiene que proveer una respuesta, y 2) identificar a otros actores (los agentes externos nuevos) que no se mostraron originalmente en el diagrama de contexto.

Otra forma de identificar los casos de uso (eventos) es estudiar el modelo de datos, asumiendo que se desarrolló un modelo de datos antes de dibujar los diagramas de flujos de datos, y estudiar la historia de vida de cada entidad en ese modelo de datos. Las instancias de estas entidades deben ser creadas, actualizadas y finalmente suprimidas. Los eventos o los casos de uso desencadenan estas acciones en la entidad. No es difícil hacer que los usuarios hablen de los eventos que podrían crear, actualizar y suprimir las instancias de entidad. Después de todo, ellos viven estos eventos diariamente. Este acercamiento se usó para construir la lista del caso de uso del proyecto SoundStage.

Una tabla parcial de casos de uso se ilustra en la figura 8.17. Para cada caso práctico, usted encontrará:

- El actor que inicia el evento (que se convertirá en un agente externo en nuestros DFD).
- El evento (que será manipulado por un proceso en nuestros DFD).
- La entrada o gatillo (que se convertirá en un flujo de datos o de control en nuestros DFD).
- Todas las salidas y las respuestas (que también se convertirán en flujos de datos en nuestros DFD). Observe que usamos paréntesis para denotar eventos temporales.
- Las salidas (pero tenga cuidado de no implicar la implementación). Cuando usamos el término *reporte* no necesariamente nos referimos a un documento basado en papeles. Observe que nuestras respuestas incluyen cambios a datos almacenados acerca de entidades del modelo de datos. Éstos incluyen la creación de instancias nuevas de la entidad, actualizar las instancias existentes de la entidad y suprimir las instancias de la entidad.

El número de casos de uso para un sistema generalmente es muy grande. Esto es necesario para asegurar que los diseñadores de sistemas respondan a todos los eventos de negocios. Como un paso final, considere la asignación de cada evento a uno de los subsistemas y las funciones identificadas en el diagrama de descomposición (dibujado en el paso previo).

## > Los diagramas de descomposición de eventos

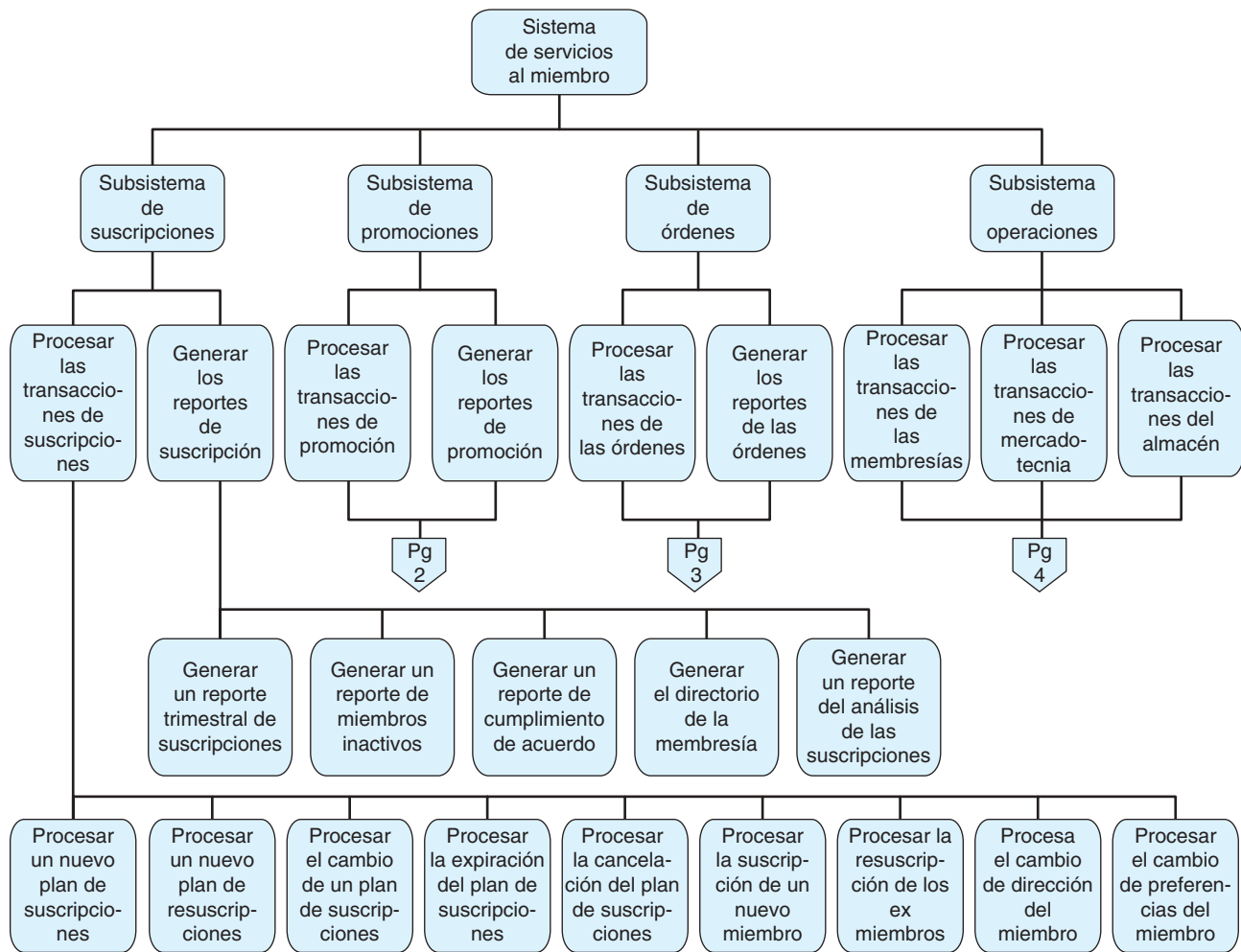
Para subdividir aún más nuestras funciones en el diagrama de descomposición, simplemente sumamos los procesos de manejo de eventos (uno por cada caso práctico) a la descomposición (vea la figura 8.18). Si el diagrama entero de descomposición no cabe en una sola página, adicione páginas separadas para los subsistemas o las funciones. El proceso raíz en una página siguiente debería duplicarse de una página anterior para proveer una

**FIGURA 8.17** Tabla parcial de casos de uso

Actor/agente externo	Evento (o caso práctico)	Desencadenador	Respuestas
Mercadotecnia	Establece un nuevo plan de suscripción de membresías para reclutar nuevos miembros.	NUEVO PROGRAMA DE SUSCRIPCIÓN DE MIEMBROS	Generar una CONFIRMACIÓN DEL PLAN DE SUSCRIPCIÓN. Crear un ACUERDO en la base de datos.
Mercadotecnia	Establece un nuevo plan de resuscripción de membresías para atraer a los ex miembros.	PROGRAMA DE RESUSCRIPCIÓN DE EX MIEMBROS	Generar una CONFIRMACIÓN DEL PLAN DE SUSCRIPCIÓN. Crear un ACUERDO en la base de datos.
Mercadotecnia	Cambia del plan de suscripción para miembros vigentes (por ejemplo, ampliar el periodo de cumplimiento).	CAMBIO DEL PLAN DE SUSCRIPCIÓN	Generar CONFIRMACIÓN DE CAMBIO DE ACUERDO. Actualizar el ACUERDO en la base de datos.
(tiempo)	Expira el plan de suscripción.	(fecha actual)	Generar CONFIRMACIÓN DE CAMBIO DE ACUERDO. Borrar lógicamente (vaciar) el ACUERDO en la base de datos.
Mercadotecnia	Cancela un plan de suscripción antes de la fecha de expiración planeada.	CANCELACIÓN DEL PLAN DE SUSCRIPCIÓN	Generar CONFIRMACIÓN DE CAMBIO DE ACUERDO. Borrar lógicamente (vaciar) el ACUERDO en la base de datos.
Miembro	Se adhiere al club mediante una suscripción. ("Llévese cualquiera de 12 CDs por un dólar y compre 4 más a precios regulares en dos años.")	SUSCRIPCIÓN NUEVA	Generar la CONFIRMACIÓN DE LA ACTUALIZACIÓN DEL DIRECTORIO DE MIEMBROS. Crear un MIEMBRO en la base de datos. Crear primero la ORDEN DEL MIEMBRO y los PRODUCTOS ORDENADOS POR EL MIEMBRO en la base de datos.
Miembro	Cambio de dirección (incluyendo correo electrónico y código de privacidad).	CAMBIO DE DIRECCIÓN	Generar la CONFIRMACIÓN DE LA ACTUALIZACIÓN DEL DIRECTORIO DE MIEMBROS. Actualizar al MIEMBRO en la base de datos.
Cuentas por cobrar	Cambia la situación de crédito del miembro.	CAMBIO DEL ESTATUS DE CRÉDITO	Generar la CONFIRMACIÓN DE LA ACTUALIZACIÓN DEL DIRECTORIO DE CRÉDITOS. Actualizar al MIEMBRO en la base de datos.

**FIGURA 8.17** (Conclusión)

Actor/agente externo	Evento (o caso práctico)	Desencadenador	Respuestas
(tiempo)	90 días después que el área de mercadotecnia decida ya no vender más el producto.	(fecha actual)	Generar la CONFIRMACIÓN DEL CAMBIO DEL CATÁLOGO Borrar lógicamente (desactivar) el PRODUCTO en la base de datos.
Miembro	Quiere recoger productos para una posible compra. (El requerimiento lógico está impulsado por la visión del acceso a la información basado en la Red.)	AVERIGUACIÓN DE PRODUCTO	Generar una DESCRIPCIÓN DE CATÁLOGO.
Miembro	Coloca la orden.	ORDEN DE UN NUEVO MIEMBRO	Generar una CONFIRMACIÓN DE LA ORDEN DE UN MIEMBRO. Crear la ORDEN DE UN MIEMBRO y el PRODUCTO ORDENADO POR UN MIEMBRO en la base de datos.
Miembro	Revisa la orden.	PETICIÓN DE CAMBIO DE LA ORDEN DEL CLIENTE	Generar una CONFIRMACIÓN DE LA ORDEN DE UN MIEMBRO. Actualizar la ORDEN DEL MIEMBRO y/o los PRODUCTOS ORDENADOS POR EL MIEMBRO en la base de datos.
Miembro	Cancela la orden.	CANCELACIÓN DE LA ORDEN DE UN MIEMBRO	Generar una CONFIRMACIÓN DE LA ORDEN DE UN MIEMBRO. Borrar lógicamente la ORDEN DEL MIEMBRO y los PRODUCTOS ORDENADOS POR EL MIEMBRO en la base de datos.
(tiempo)	90 días después de la orden.	(fecha actual)	Borrar físicamente la ORDEN DEL MIEMBRO y los PRODUCTOS ORDENADOS POR EL MIEMBRO en la base de datos.
Miembro	Indaga acerca de su historial de compra (límite de tiempo de 3 años).	AVERIGUACIÓN DE LA COMPRA DEL MIEMBRO	Generar el HISTORIAL DE LAS COMPRAS DEL MIEMBRO.
(cada uno) Club	(final del mes).	(fecha actual)	Generar el ANÁLISIS MENSUAL DE VENTAS. Generar un ANÁLISIS MENSUAL DE EXCEPCIÓN DE ACUERDOS DE LOS MIEMBROS. Generar un REPORTE DE ANÁLISIS DE LAS MEMBRESÍAS



**FIGURA 8.18** Diagrama parcial de descomposición de eventos (creado con el *System Architect 2001*)

referencia cruzada. La figura 8.18 muestra sólo los procesos de eventos para el subsistema de membresías. Los eventos para las funciones de promociones y órdenes estarían en páginas separadas.

No hay necesidad de dividir el diagrama de descomposición más allá de los eventos y los reportes. Eso sería como delinear hasta las frases o párrafos finales en un papel. El diagrama de descomposición, tal como está construido, servirá como un buen contorno para los diagramas posteriores de flujo de datos.

### > Los diagramas de eventos

Al usar nuestro diagrama de descomposición como un esquema podemos dibujar un diagrama de eventos para cada proceso de eventos. Éste es un paso optativo, pero útil. Un **diagrama de eventos** es un diagrama de contexto para un evento individual. Demuestra las entradas, las salidas y las interacciones del almacén de datos para el evento. Al dibujar un diagrama de eventos para cada proceso, los usuarios no se abrumarán por el tamaño global del sistema. Pueden examinar cada caso de uso como su propio diagrama de contexto.

Antes de dibujar cualquier diagrama de eventos, usted puede encontrar útil tener una lista de todos los almacenes de datos disponibles. Ya que SoundStage completó el modelo de datos para este proyecto, los miembros del equipo simplemente crearon una lista de cada nombre de entidad en ese modelo (vea en el margen en la página siguiente). Es útil revisar la definición y los atributos para cada almacén de entidad /datos en la lista.

#### diagrama de eventos

Diagrama de flujo de datos que muestra el contexto de un evento individual.

### ALMACENES DE DATOS (ENTIDADES)

ACUERDOS  
MIEMBROS  
ÓRDENES DE LOS MIEMBROS  
PRODUCTOS ORDENADOS  
POR LOS MIEMBROS  
PRODUCTOS  
PROMOCIONES  
PROMOCIONES TITULARES

La mayoría de diagramas de eventos contiene un proceso individual: el mismo proceso que se nombró para manejar el evento en el diagrama de descomposición. Para cada evento, ilustre lo siguiente:

- Las entradas y sus fuentes. Las fuentes se esquematizan como agentes externos. La estructura de datos de cada entrada deberá registrarse en el depósito.
- Las salidas y sus destinos. Los destinos son esquematizados como agentes externos. La estructura de datos para cada salida deberá registrarse en el depósito.
- Para cualquier almacén de datos deberán “leerse” sus registros y añadirse al diagrama de eventos. Los flujos de datos deberán añadirse y nombrarse para reflejar qué dato es leído por el proceso.
- Cualquier almacén de datos en los cuales deban crearse, suprimirse o actualizarse registros deberán incluirse en el diagrama de eventos. Los flujos de datos de los almacenes de datos deberán nombrarse para reflejar la naturaleza de la actualización.

La simplicidad de la diagramación de eventos hace de la técnica una herramienta poderosa de comunicación entre los usuarios y los profesionales técnicos.

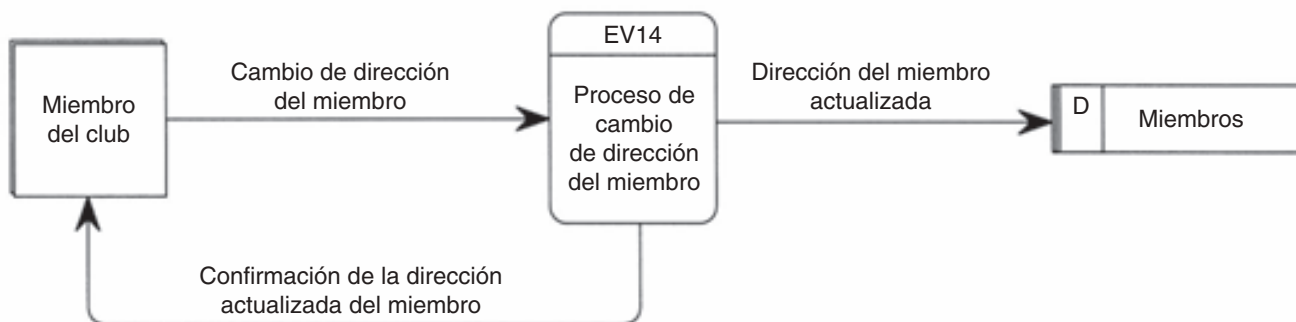
Un conjunto completo de diagramas de eventos para el estudio de caso SoundStage duplicaría la longitud de este capítulo sin añadir valor educativo sustantivo. Así, demostraremos el modelo con tres ejemplos simples.

La figura 8.19 ilustra un diagrama simple de eventos para un evento externo. La mayoría de los sistemas tienen muchos de estos diagramas simples de eventos porque todos los sistemas deben suministrar un mantenimiento de rutina de los almacenes de datos.

La figura 8.20 esquematiza un evento externo algo más complejo, uno para la transacción de negocios ORDEN DEL MIEMBRO. Observe que las transacciones de negocios tienden a usar y actualizar más almacenes de datos y más interacciones con los agentes externos.

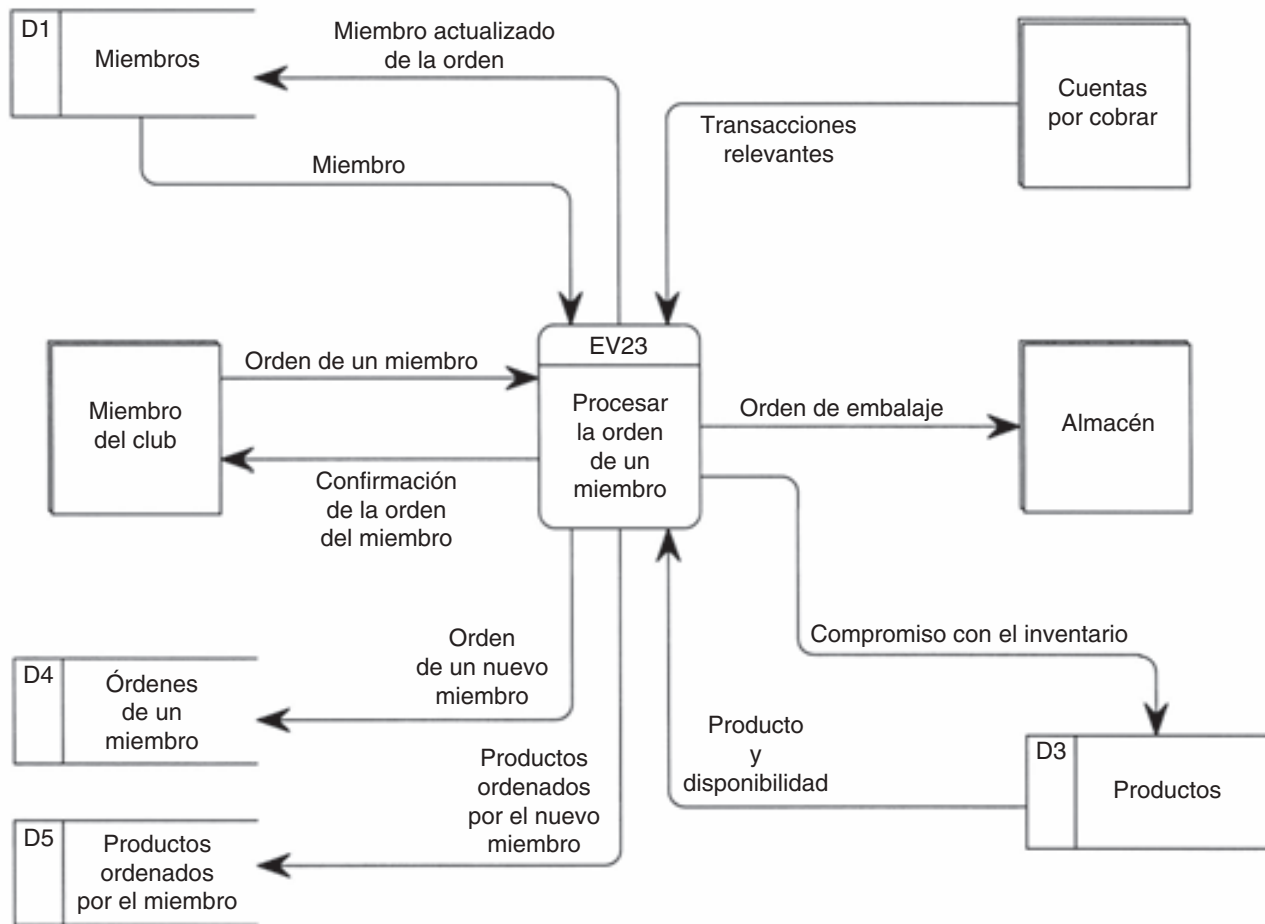
¿Puede tener un diagrama de eventos más de un proceso? La respuesta es tal vez. Algunos procesos de eventos pueden desencadenar otros procesos de eventos. En este caso, deberá mostrarse la combinación de eventos en un diagrama de eventos individual. De acuerdo con nuestra experiencia, la mayoría de diagramas de eventos tienen un proceso. Un diagrama de eventos ocasional puede tener dos o quizá tres procesos. Si el número de procesos excede de tres, usted probablemente dibuja lo que se llama un *diagrama de actividad* (prematuramente), no un diagrama de eventos; en otras palabras, usted se ocupa demasiado de los detalles. La mayoría de los procesos de eventos no se comunican directamente entre sí. En lugar de eso, se comunican a través de almacenes compartidos de datos. Esto le permite a cada proceso de eventos cumplir con su trabajo sin preocupaciones acerca de vigilar otros procesos.

La figura 8.21 muestra un diagrama de eventos para un evento temporal. Agreguemos una entidad externa CALENDARIO O TIEMPO para servir como fuente para este flujo de control.



**FIGURA 8.19** Diagrama simple de eventos externos (creado con el *System Architect 2001*)





**FIGURA 8.20** Diagrama más complejo de eventos externos (creado con el *System Architect 2001*)

Cada proceso de evento deberá describirse al depósito de CASE con las siguientes propiedades:

- La oración del evento: para la perspectiva de negocios.
- Los requisitos de producción: el volumen de entradas para algún lapso.
- Los requisitos de tiempo de respuesta: con qué rapidez debe ser manejado el evento típico.
- La seguridad, la auditoría y los requisitos de control.
- Los requisitos de archivos (desde una perspectiva de negocios).

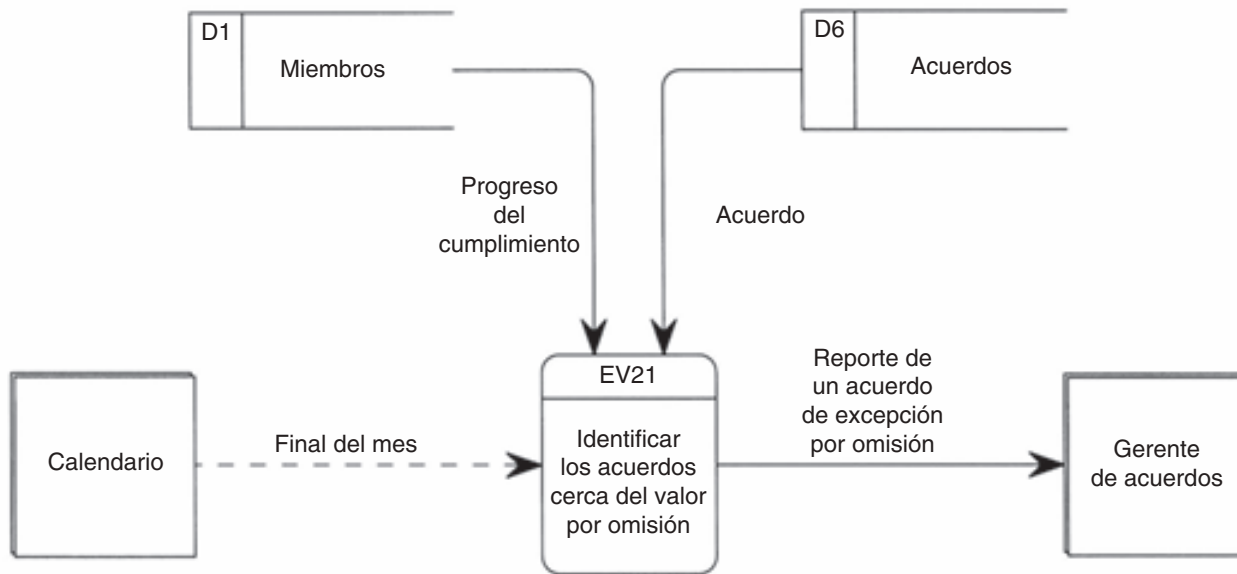
Por ejemplo, considere el diagrama de eventos en la figura 8.19: “Un miembro propone un cambio de dirección.”

- Ocurren 25 veces al mes.
- Debería ser procesado dentro de 15 días.
- Debe proteger privacidad de direcciones a menos que el miembro autorice liberación.
- Deberá retener un registro semipermanente de algún tipo.

Todas las propiedades anteriores pueden añadirse a las descripciones asociadas con los procesos, flujos de datos y almacenes de datos apropiados en el modelo.

### > El(los) diagrama(s) de sistema

El diagrama de eventos sirve como un contexto significativo para ayudar a los usuarios a validar la exactitud de cada evento para el cual el sistema debe proveer una respuesta.



**FIGURA 8.21** Diagrama de eventos temporales (creado con el *System Architect 2001*)

Pero estos eventos no existen de manera aislada. Colectivamente definen sistemas y subsistemas. Es, por consiguiente, útil construir uno o más diagramas de sistema que muestren todos los eventos en el sistema o subsistema.

Se dice que el diagrama del sistema se “expande” del proceso individual que creamos en el diagrama original de contexto (figura 8.15). El diagrama de sistema muestra bien sea 1) todos los eventos del sistema en un diagrama individual, o 2) todos los eventos para un subsistema individual en un diagrama individual. Dependiendo del tamaño del sistema, un diagrama individual puede ser demasiado grande.

Aun cuando el proyecto SoundStage es de tamaño moderado, todavía responde a demasiados eventos como para apretujar todos esos procesos en un solo diagrama. En lugar de eso, Bob Martínez eligió dibujar un diagrama de subsistema para cada uno de los subsistemas principales. La figura 8.22 muestra el diagrama de subsistema para el SUBSISTEMA de ÓRDENES. Consolida todos los eventos de transacciones y de escritura de reportes para ese subsistema en un solo diagrama. (Los eventos de reportar pueden omitirse o consolidarse en combinaciones si el diagrama está demasiado saturado.) Observe que el diagrama de sistemas demuestra cómo los procesos de eventos se comunican usando almacenes compartidos de datos.

Si es necesario, y después de dibujar los cuatro diagramas de subsistemas para este proyecto, Bob pudo haber dibujado un diagrama de sistema que ilustre sólo las interacciones entre esos cuatro subsistemas. Ésta es una reliquia de la estrategia original de diagramación de flujos de datos de arriba abajo de la metodología original de análisis estructurado. En la práctica, este diagrama de nivel más alto requiere tanta consolidación de flujos y almacenes de datos que su valor de comunicación es dudoso. Para Bob, ésta fue tarea laboriosa pero no productiva, y su tiempo fue mejor empleado en el siguiente conjunto de diagramas de flujo de datos.

Ahora tenemos un conjunto de diagramas de eventos (uno por cada evento de negocios) y uno o más diagramas de sistema/subsistema. Los procesos del diagrama de eventos se fusionan con los diagramas de sistemas. Es muy importante que cada uno de los flujos de datos, almacenes de datos y agentes externos que se ilustraron en los diagramas de eventos se representen en los diagramas de sistemas. Observe que duplicamos los almacenes de datos y los agentes externos para minimizar el cruce de líneas. La mayoría de las herramientas CASE incluyen ayuda para revisar el balance de errores.

Antes de dejar este tema deberíamos introducir el concepto de balance. El **balance** es la sincronización de los diagramas de flujo de datos en niveles diferentes de detalle para conservar la consistencia y totalidad de los modelos. El balance es una técnica de comprobación de calidad, la cual implica que si usted expande un proceso hasta otro DFD para revelar más

**balance** Concepto que requiere que los diagramas de flujo de datos con niveles diferentes de detalle reflejen consistencia y totalidad.

detalle, debe incluir los mismos flujos y almacenes de datos en el diagrama hijo que incluyó en el proceso inicial del diagrama padre (o sus equivalentes lógicos).

### > Los diagramas elementales

Algunos procesos de eventos en el diagrama de sistemas pueden expandirse hasta un diagrama elemental de flujo de datos para revelar más detalle. Esto es especialmente cierto con los procesos de transacción de negocios más complejos (por ejemplo, procesamiento de órdenes). Otros eventos, tales como la generación de reportes, son lo suficientemente simples como para no requerir más expansión.

Los procesos de eventos con diagramas de eventos más complejos deberán expandirse a un diagrama elemental de flujo de datos más detallado, tal como se ilustra en la figura 8.23. Este DFD elemental muestra los requisitos detallados del procesamiento para el evento. Este DFD muestra varios procesos elementales *para* el proceso del evento. Cada proceso elemental es cohesivo, es decir, hace solamente una única cosa. En una diagramación elemental es permisible tener flujos que conecten los procesos elementales.

Cuando Bob dibujó este diagrama elemental de flujo de datos, tuvo que sumar flujos de datos nuevos entre los procesos. Al hacer eso, él trató de practicar una buena conservación de datos, asegurándose de que cada proceso tuviera sólo los datos que verdaderamente necesitaba. La estructura de datos para cada flujo de datos tuvo que describirse en su depósito de herramienta CASE. También observe que él usó conexiones de flujo de datos para dividir y fusionar los flujos de datos apropiados al diagrama.

Observe que el DFD contiene algunos flujos de datos nuevos de excepción que *no* fueron introducidos en la figura 8.22. No debería ser difícil imaginar una estructura de programa de computadora al examinar a este DFD.

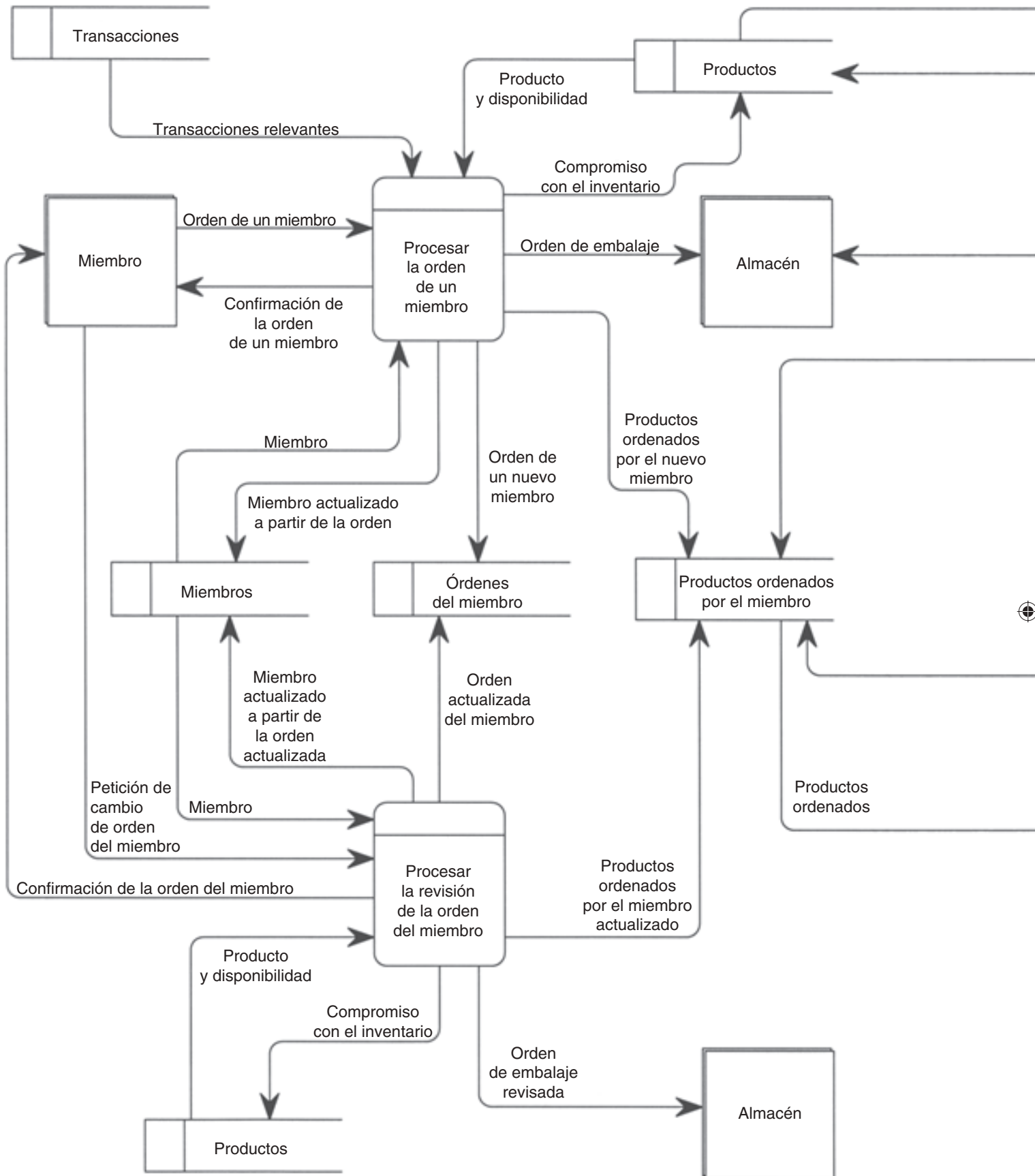
La combinación del diagrama de contexto, el diagrama de sistemas, el diagrama de eventos y los diagramas elementales completan nuestros modelos de proceso. Colectivamente, éste *es* el modelo de proceso. Un modelo de proceso bien diseñado y completo puede comunicar eficazmente los requisitos de negocios entre los usuarios finales y los programadores de computadoras, eliminando mucha de la confusión que a menudo ocurre en el diseño de sistemas, la programación y la implementación.

### > Completando la especificación

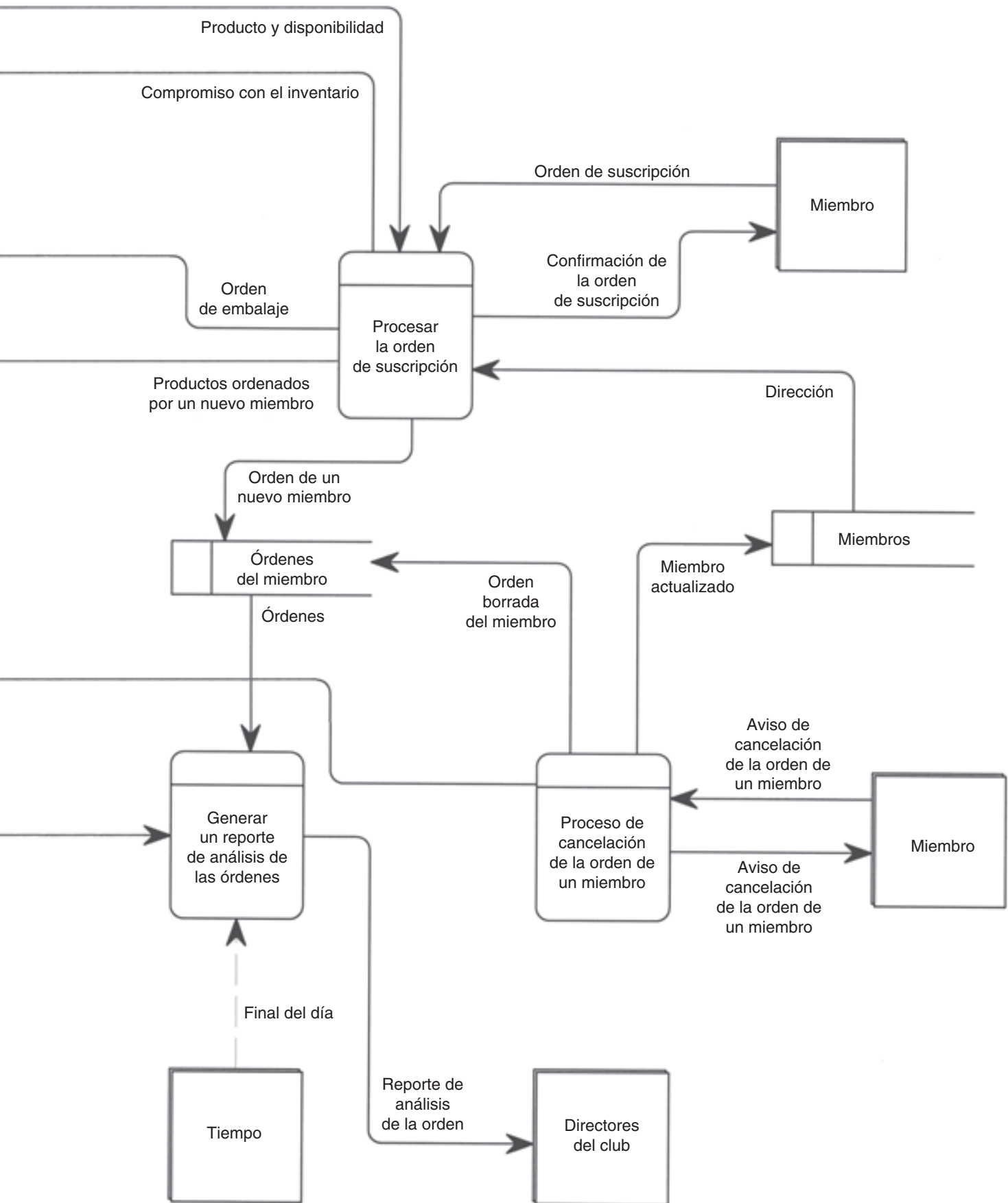
Los diagramas de flujo de datos son completos. ¿A dónde va usted desde aquí? Eso depende de su elección de metodología. Si usted practica la metodología de análisis estructurado puro (de la cual se derivó la diagramación de flujo de datos), debe completar la especificación. Para hacerlo, cada flujo de datos, almacén de datos y proceso *elemental* (lo que significa que ya no puede expandirse más hasta un DFD más detallado) debe describirse a la enciclopedia o diccionario de datos. Las herramientas CASE proveen facilidades para tales descripciones.

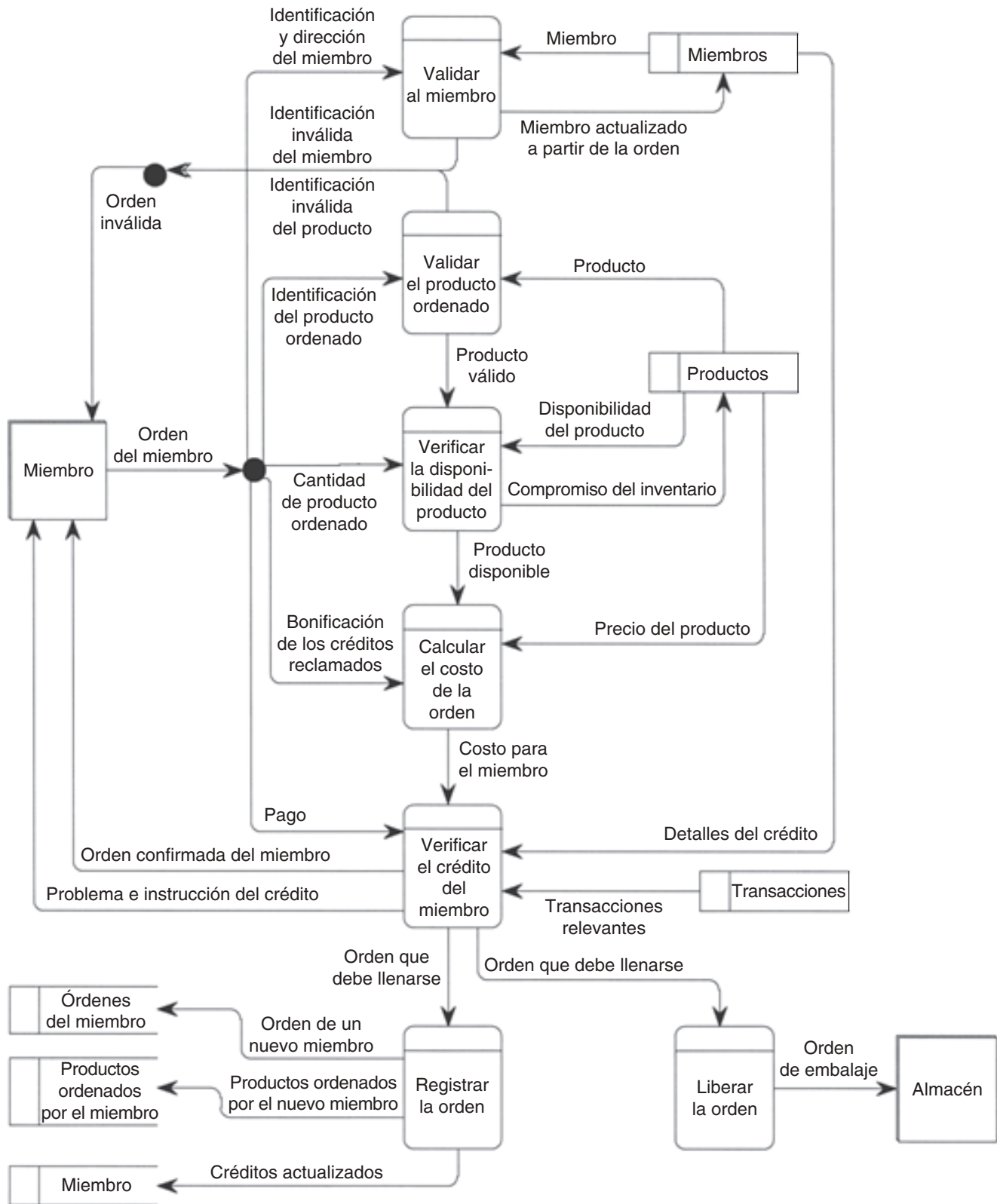
Los flujos de datos están compuestos por estructuras de datos, como se explicó anteriormente en este capítulo. La figura 8.24 demuestra cómo el *System Architect 2001*, la herramienta CASE SoundStage, puede usarse para describir un flujo de datos. Observe que esta herramienta CASE usa una notación algebraica para las estructuras de datos, tal como se describen en este capítulo. Finalmente, cada elemento de datos o atributo también deberá estar descrito en el diccionario de datos para especificar el tipo de datos, el dominio y el valor por omisión (como se describió en el capítulo 7). Los almacenes de datos corresponden a todas las instancias de una entidad de datos de nuestro modelo de datos. De esta forma, se describe mejor cada entidad y sus atributos correspondientes en el diccionario de datos, tal como se enseñó en el capítulo 7. A algunos analistas les gusta trasladar el contenido de cada almacén de datos a una estructura de datos relacional parecida a la que se usa en la figura 8.24 para describir los flujos de datos. Consideramos que ésta es una tarea laboriosa; que las descripciones de entidad del modelo de datos describen el contenido de un almacén de datos. Además, la definición de las estructuras de datos para los almacenes de datos podría conducir a errores de sincronización entre los datos y los modelos de procesos; si usted hiciera algún cambio a una entidad en el modelo de datos, se vería forzado a recordar hacer esos mismos cambios en la estructura de datos correspondiente del almacén de datos. Esto requiere demasiado esfuerzo (a menos que usted tenga una herramienta CASE capaz de hacerlo automáticamente).

**Lógica del proceso** Los diagramas de descomposición y los diagramas de flujo de datos resultarán ser herramientas muy efectivas para identificar procesos, pero no son buenas

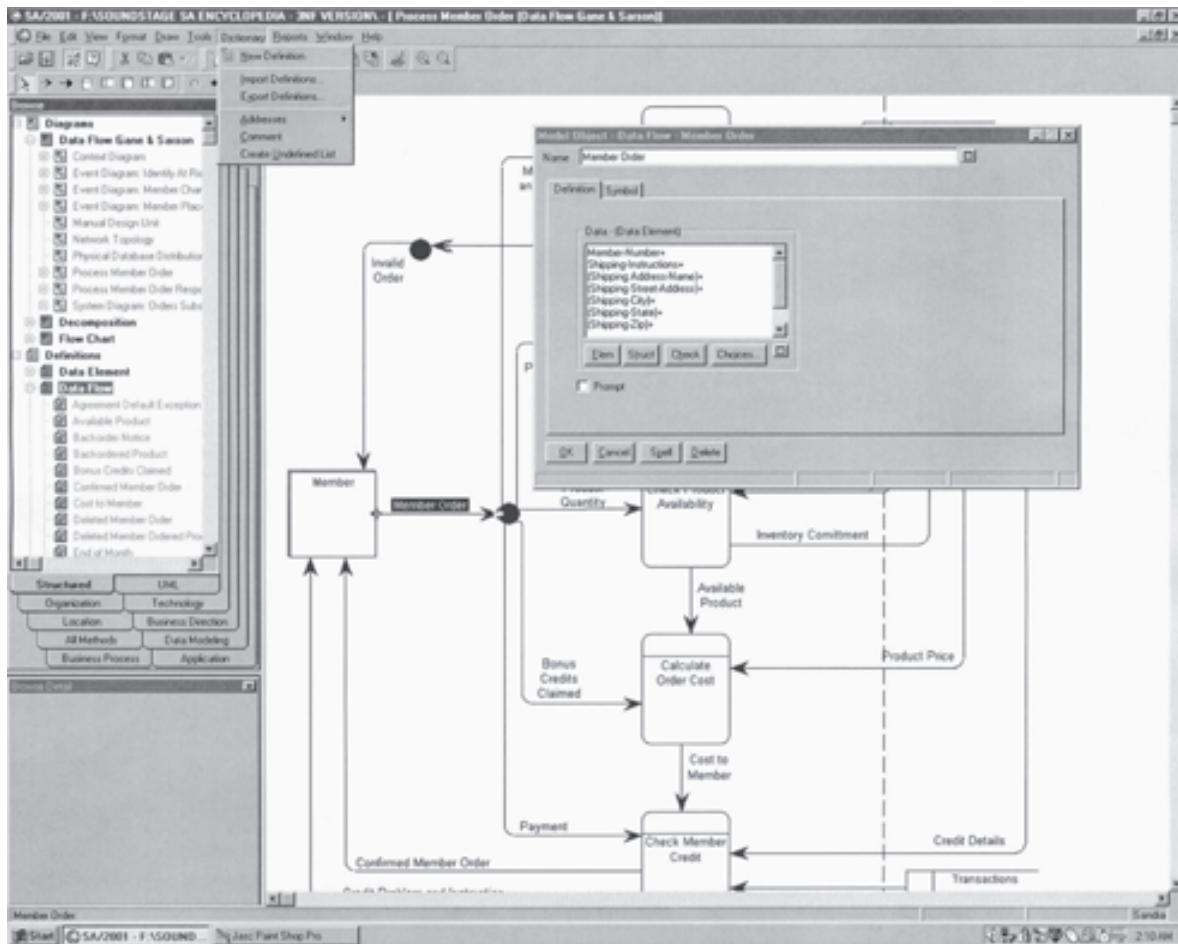


**FIGURA 8.22** Diagrama de sistemas (creado con el *System Architect 2001*)





**FIGURA 8.23** Diagrama elemental (creado con el *System Architect 2001*)



**FIGURA 8.24** Flujo de datos (creado con el *System Architect 2001*)

para mostrar la lógica dentro de esos procesos. Finalmente, necesitaremos especificar *instrucciones* detalladas para los procesos elementales en un diagrama de flujo de datos. Considero, por ejemplo, un proceso elemental llamado verificar el crédito del cliente. Por sí mismo, el proceso nombrado es insuficiente para explicar la lógica necesaria para comprobar el crédito del cliente. Necesitamos una forma efectiva de modelar la lógica de un proceso elemental. Idealmente, nuestro modelo lógico debería igualmente ser efectivo para comunicarse con los usuarios (quienes deben verificar la exactitud de negocios de la lógica) y los programadores (quienes tendrían que implementar la lógica de negocios en un lenguaje de programación).

Podemos descartar los diagramas de flujo. Aun cuando ellos ciertamente modelan la lógica del proceso, la mayoría de los usuarios finales tienden a intimidarse mucho por ellos. Lo mismo sería cierto del pseudocódigo y otras herramientas populares de lógica de programación. También podemos descartar al inglés natural. Con demasiada frecuencia es impreciso y frecuentemente sujeto a la interpretación (y la mala interpretación). La figura 8.25 resume algunos problemas comunes encontrados por quienes tratan de usar el inglés natural como un idioma procedural.

Para ocuparse de este problema, requerimos una herramienta que concilie algunas ventajas del inglés natural con algo del rigor de las herramientas lógicas de programación. El **inglés estructurado** es un idioma y una sintaxis que se basa en las fortalezas relativas de la programación estructurada y el inglés natural, para especificar la lógica subyacente de procesos elementales en los modelos de procesos (como los *diagramas de flujo de datos*). Un ejemplo de inglés estructurado se muestra en la figura 8.26. (Los números y letras al principio de cada declaración son optativos. A algunos usuarios finales les gustan porque remueven aún más la “apariencia y la percepción” de la especificación.)

**inglés estructurado** Sintaxis de lenguaje para especificar la lógica de un proceso.

### FIGURA 8.25 Problemas con el inglés natural como lenguaje de especificación de procedimientos

- Muchos de nosotros no escribimos bien, y además tendemos a no cuestionar nuestras habilidades de redacción.
- ¡Muchos de nosotros tenemos demasiados estudios! Frecuentemente es difícil para una persona con muchos estudios comunicarse con una audiencia que tal vez no haya tenido las mismas oportunidades de estudiar. Por ejemplo, el egresado promedio de universidad (incluyendo la mayoría de los analistas) tiene un vocabulario de trabajo de 10 000 a 20 000 palabras; por otro lado, el egresado promedio que no asistió a la universidad tiene un vocabulario de trabajo de aproximadamente 5 000 palabras.
- Algunos de nosotros escribimos todo como si fuera un programa. Si los procedimientos de negocios requiriesen esta precisión, escribiríamos todo en un lenguaje de programación.
- Con demasiada frecuencia, permitimos que la jerga y los acrónimos de la computación dominen nuestro lenguaje.
- Frecuentemente las declaraciones en lenguaje coloquial tienen un alcance excesivo o confuso. ¿Cómo desarrollaría usted este procedimiento? “Si los clientes cruzan la puerta y no quieren retirar dinero de su cuenta o depositar dinero en su cuenta o hacer un pago a un préstamo, mándelos al departamento del fideicomiso. ¿Significa esto que la única vez que usted no deberá mandar al cliente al departamento de fideicomiso es cuando él o ella desean hacer las tres transacciones? ¿O significa que si el cliente no desea realizar cuando menos una de las tres transacciones, ese cliente no deberá ser enviado al departamento de fideicomiso?”
- Nosotros tendemos a usar demasiado las oraciones compuestas. Considere el siguiente procedimiento: “Retire los tornillos que fijan la tapa del contacto en la pared. Retire la tapa del contacto. Desconecte cada uno de los alambres del contacto, pero primero asegúrese que la energía eléctrica del contacto ha sido desconectada”. ¡Una persona desprevenida podría tratar de desconectar los alambres antes de desconectar la energía!
- Muchas palabras tienen definiciones múltiples.
- Numerosas declaraciones usan adjetivos imprecisos. Por ejemplo, un funcionario de préstamos le pide a un maestro que certifique que el estudiante tiene un buen desempeño académico. ¿Qué es “buen”?
- Las instrucciones condicionadas pueden ser imprecisas. Por ejemplo, si afirmamos que “todos los solicitantes menores de 19 años deben tener permiso de sus padres”, ¿queremos decir menos de 19 o al menos 19?
- Las condiciones compuestas tienden a surgir en el lenguaje coloquial. Por ejemplo, si la aprobación de crédito está en función de varias condiciones (clasificación del crédito, límite de crédito, ventas anuales para el cliente en cuestión) entonces las diferentes combinaciones de estos factores pueden conducir a diferentes decisiones. A medida que aumenta el número de condiciones y de posibles combinaciones, el procedimiento se hace más y más tedioso y más difícil de escribir.

Fuente: Adaptado de Leslie Matthies: *The New Playscript Procedure* (Stamford, CT: Office Publications, Inc., 1977).

El inglés estructurado *no* es un pseudocódigo. No se preocupa por las declaraciones, inicialización, enlace y asuntos técnicos de este tipo. Sin embargo, ciertamente pide pres-tadas algunas de las construcciones lógicas de la *programación estructurada* para vencer la falta de estructura y la precisión en el idioma inglés. Piense acerca de eso como el matrimonio de idioma inglés natural con la sintaxis de la programación estructurada.

### FIGURA 8.26

Uso del inglés estructurado para documentar un proceso elemental

1. Para cada número de cliente en los datos ingrese CLIENTES:
  - a. Para cada PRÉSTAMO en los datos ingrese PRÉSTAMOS que concuerde con el NÚMERO DEL CLIENTE anterior:
    - 1) Mantenga un total móvil del NÚMERO DE PRÉSTAMOS para el NÚMERO DE CLIENTE.
    - 2) Mantenga un total móvil del CAPITAL PRINCIPAL DEL PRÉSTAMO para el NÚMERO DEL CLIENTE.
    - 3) Mantenga un total móvil del SALDO ACTUAL DEL PRÉSTAMO para el NÚMERO DEL CLIENTE.
    - 4) Mantenga un total móvil de las CANTIDADES VENCIDAS para el NÚMERO DEL CLIENTE.
  - b. Si el TOTAL DE CANTIDADES VENCIDAS para el NÚMERO DEL CLIENTE es mayor que 100.00 entonces
    - 1) Escriba el NÚMERO DEL CLIENTE y los datos en el flujo de datos PRÉSTAMOS EN PELIGRO.

O bien

    - 1) Excluya el NÚMERO DEL CLIENTE y los datos del flujo de datos PRÉSTAMOS EN PELIGRO.



La estructura global de una especificación de inglés estructurado se construye usando las construcciones fundamentales que han gobernado la programación estructurada durante casi tres decenios. Estas construcciones (resumidas en la figura 8.27) son:

- Una *secuencia* de oraciones simples, declarativas, una después de otra. No se aconsejan las oraciones compuestas porque frecuentemente crean ambigüedad. Cada oración usa verbos fuertes de acción tales como GET, FIND, RECORD, CREATE, READ, UPDATE, DELETE, CALCULATE, WRITE, SORT, MERGE, o cualquier otra cosa reconocible o comprensible para los usuarios. Puede incluirse una fórmula como parte de una oración (por ejemplo, CALCULATE GROSS PAY = HOURS WORKED × HOURLY WAGE).
- Una *estructura condicional* o *de decisión* indica que un proceso debe realizar pasos diferentes bajo condiciones bien especificadas. Hay dos variaciones (y una salida) en esta construcción.
  - La construcción de IF-THEN-ELSE implica que si la condición es cierta, deberán seguirse un conjunto de pasos, si la condición es falsa, deberá realizarse un conjunto diferente de pasos. Los pasos que se llevarán a cabo son generalmente una *secuencia* de una o más oraciones tal como se describe antes.
  - La construcción CASE es usada cuando hay más de dos conjuntos de pasos para escoger. Nuevamente, estos pasos constan de dichas declaraciones *secuenciales*. La construcción de caso es una sustitución elegante para una construcción IF-THEN-ELSE IF-THEN-THEN ELSE IF-THEN... (que es muy intrincada para el usuario común).
  - Para la lógica basada en las condiciones múltiples y las combinaciones de condiciones (que los programadores llaman una *declaración IF anidada*), las *tablas de decisión* son una herramienta de modelación de lógica mucho más elegante. Las tablas de decisión se introducirán en poco tiempo.
- Una estructura de *iteración*, o de *repetición*, especifica que un conjunto de pasos deberá repetirse basándose en alguna condición establecida. Hay dos variaciones en esta construcción:
  - La construcción DO-WHILE señala que ciertos pasos se repetirán cero, una o más veces basándose en el valor de la condición establecida. Observe que estos pasos no pueden ejecutarse en absoluto si la condición no es cierta cuando la condición se prueba por primera vez.
  - La construcción REPEAT-UNTIL señala que ciertos pasos se repiten una o más veces basándose en el valor de la condición establecida. Observe que un conjunto de pasos REPEAT-UNTIL debe ejecutarse al menos una vez, a diferencia del conjunto DO-WHILE de acciones.

Adicionalmente, el inglés estructurado coloca las siguientes restricciones en la lógica de proceso:

- Sólo pueden usarse verbos imperativos.
- Sólo pueden usarse nombres que han sido definidos en el diccionario del proyecto. Estos nombres pueden incluir aquéllos de los flujos de datos, almacenes de datos, entidades (de los modelos de datos; véase el capítulo 7), los atributos (los campos de datos o propiedades especificados contenidos en un flujo de datos, almacén de datos, o una entidad) y los dominios (los valores legales especificados para los atributos).
- Las fórmulas deberán enunciarse claramente usando notaciones matemáticas apropiadas. En resumen, usted puede usar cualquier notación reconocible para los usuarios. Asegúrese de que cada operando en una fórmula alimente al proceso en un flujo de datos o una constante definida.
- Los adjetivos y los adverbios indefinidos (la palabra *good*, por ejemplo) no se permiten, a menos que se definan claramente en el diccionario del proyecto como valores legales para los atributos de datos.
- El bloqueo y la sangría se usan para marcar el inicio y final de las construcciones y para magnificar la legibilidad. (Algunos autores y modelos alientan al uso de verbos especiales como ENDIF, ENDCASE, ENDDO y ENDREPEAT para terminar construcciones. Nos desagrada esta práctica porque le da al inglés estructurado un aspecto y una percepción de pseudocódigo o de programación demasiado acentuados.)
- En caso de duda, la legibilidad del usuario siempre deberá tener prioridad sobre las preferencias del programador.

Estructuras de procedimiento del inglés estructurado

Construcción	Plantilla de muestras																																			
<p><b>Secuencia de pasos</b> – Ejecutar incondicionalmente una secuencia de pasos.</p>	<p>[Paso 1] [Paso 2] ... [Paso n]</p>																																			
<p><b>Pasos de condición simple</b> – Si la condición especificada es verdad, entonces realice el primer conjunto de pasos. De otra manera, realice el segundo conjunto de pasos.</p> <p>Use esta construcción si la condición tiene solamente dos valores posibles.</p> <p>(Nota: El segundo conjunto de condiciones es opcional.)</p>	<p><b>If</b> [condición de verdad] <b>then</b>     [secuencia de pasos u otros pasos condicionales] <b>else</b>     [secuencia de pasos u otros pasos condicionales] <b>End-If</b></p>																																			
<p><b>Pasos de condición compleja</b> – Pruebe el valor de la condición y realice el conjunto apropiado de pasos.</p> <p>Use esta construcción si la condición tiene más de dos valores.</p>	<p><b>Haga lo siguiente basándose en</b> [condición]: <b>Caso 1: If</b> [condición] = [valor] then     [secuencia de pasos u otros pasos condicionales] <b>Caso 2: If</b> [condición] = [valor] then     [secuencia de pasos u otros pasos condicionales] ... <b>Caso n: If</b> [condición] = [valor] then     [secuencia de pasos u otros pasos condicionales] <b>End-Case</b></p>																																			
<p><b>Condiciones múltiples</b> – Pruebe el valor de las condiciones múltiples para determinar el conjunto correcto de pasos.</p> <p>Use una tabla de decisiones en lugar de las construcciones de inglés estructurado if-then-else para simplificar la presentación de la lógica compleja que incluye las combinaciones de las condiciones.</p> <p><i>Una tabla de decisiones es una presentación tabular de lógica compleja en la cual las filas representan condiciones y posibles acciones y las columnas indican cuáles combinaciones de condiciones conducen a acciones específicas.</i></p>	<table border="1"> <thead> <tr> <th>TABLA DE DECISIONES</th> <th>Regla</th> <th>Regla</th> <th>Regla</th> <th>Regla</th> </tr> </thead> <tbody> <tr> <td>[Condición]</td> <td>valor</td> <td>valor</td> <td>valor</td> <td>valor</td> </tr> <tr> <td>[Condición]</td> <td>valor</td> <td>valor</td> <td>valor</td> <td>valor</td> </tr> <tr> <td>[Condición]</td> <td>valor</td> <td>valor</td> <td>valor</td> <td>valor</td> </tr> <tr> <td>[secuencia de pasos u otros pasos condicionales]</td> <td>X</td> <td></td> <td></td> <td></td> </tr> <tr> <td>[secuencia de pasos u otros pasos condicionales]</td> <td></td> <td>X</td> <td>X</td> <td></td> </tr> <tr> <td>[secuencia de pasos u otros pasos condicionales]</td> <td></td> <td></td> <td></td> <td>X</td> </tr> </tbody> </table> <p>Aunque no sea una construcción de inglés estructurado, una tabla de decisiones puede tener un nombre, y referenciarse dentro de un procedimiento de inglés estructurado.</p>	TABLA DE DECISIONES	Regla	Regla	Regla	Regla	[Condición]	valor	valor	valor	valor	[Condición]	valor	valor	valor	valor	[Condición]	valor	valor	valor	valor	[secuencia de pasos u otros pasos condicionales]	X				[secuencia de pasos u otros pasos condicionales]		X	X		[secuencia de pasos u otros pasos condicionales]				X
TABLA DE DECISIONES	Regla	Regla	Regla	Regla																																
[Condición]	valor	valor	valor	valor																																
[Condición]	valor	valor	valor	valor																																
[Condición]	valor	valor	valor	valor																																
[secuencia de pasos u otros pasos condicionales]	X																																			
[secuencia de pasos u otros pasos condicionales]		X	X																																	
[secuencia de pasos u otros pasos condicionales]				X																																
<p><b>Iteración uno a muchos</b> – Repita el conjunto de pasos hasta que la condición sea falsa.</p> <p>Use esta construcción si el conjunto de pasos <u>debe</u> realizarse cuando menos una vez, independientemente del valor inicial de la condición.</p>	<p><b>Repita lo siguiente hasta</b> [condición de verdad]:     [secuencia de pasos u otros pasos condicionales] <b>End-Repeat</b></p>																																			
<p><b>Iteración cero a muchos</b> – Repita el conjunto de pasos hasta que la condición sea falsa.</p> <p>Use esta construcción si el conjunto de pasos se basa condicionalmente en el valor inicial de la condición.</p>	<p><b>Do while</b> [condición de verdad]:     [secuencia de pasos u otros pasos condicionales] <b>End-Do</b></p> <p style="text-align: center;">- OR -</p> <p><b>For</b> [condición de verdad]:     [secuencia de pasos u otros pasos condicionales] <b>End-For</b></p>																																			

**FIGURA 8.27** Construcciones de inglés estructurado

El inglés estructurado deberá ser lo suficientemente preciso para especificar con claridad el procedimiento de negocios requerido para un programador o un usuario. Pero eso no deberá ser tan inflexible como para que un programador o un usuario pase las horas discutiendo sobre la sintaxis.

Muchos procesos son gobernados por combinaciones complejas de condiciones que no se expresan con facilidad con el inglés estructurado. Esto se encuentra más comúnmente en las políticas de negocios. Una **política** es un conjunto de reglas que gobierna algún proceso en el negocio.

En la mayoría de las compañías, las políticas son la base para la toma de decisiones. Por ejemplo, una compañía de tarjetas de crédito debe enviar las facturas a los tenedores de tarjeta según las políticas diversas que se apegan a las restricciones impuestas por los gobiernos estatales y federales (las tasas máximas de interés y los pagos mínimos, por ejemplo). Las políticas constan de *reglas* que a menudo podrían traducirse a los programas de computadora si los usuarios y los analistas de sistemas pudieran comunicar exactamente esas reglas al programador de computadoras.

Hay formas para formalizar la especificación de las políticas y otras combinaciones complejas de condiciones. Una herramienta lógica de modelación de este tipo es una **tabla de decisiones**. Mientras que las personas que están poco familiarizadas con ellas tienden a evitarlas, las tablas de decisión son muy útiles para especificar políticas complejas y reglas de toma de decisiones. La figura 8.28 ilustra los tres componentes de una tabla de decisiones simple:

**política** Conjunto de reglas que rige la manera de realizar un proceso.

**tabla de decisiones** Forma tabular de presentación que especifica un conjunto de condiciones y sus acciones correspondientes.

- Los *talones de condición* (las filas superiores) describen las condiciones o los factores que afectarán la decisión o la política.

### UNA DECLARACIÓN SIMPLE DE POLÍTICA

VERIFICAR LA TARJETA DE IDENTIFICACIÓN PARA COBRAR

Un cliente con derecho a cobrar cheques puede cobrar cheques personales de hasta 75 dólares y cheques de nómina de compañías previamente autorizadas por *LMART*. Esta tarjeta se emite de acuerdo con los términos y condiciones de la solicitud y está sujeta a cambios sin aviso. Esta tarjeta es propiedad de *LMART* y deberá entregarse cuando lo solicite *LMART*.

FIRMA Charles C. Parker, Jr.

EXPIRA 31 de mayo de 2008

### TABLA EQUIVALENTE DE DECISIONES DE POLÍTICAS

Condiciones y acciones		Regla 1	Regla 2	Regla 3	Regla 4	
{	Talones de condiciones	C1: Tipo de cheque	personal	nómina	personal	nómina
	C2: cantidad de cheque menor o igual a 75 dólares	sí	no importa	no	no importa	
	C3: Compañía acreditada por <i>LMART</i>	no importa	sí	no importa	no	
{	Talones de acciones	A1: Pagar el cheque	X	X		
	A2: No pagar el cheque			X	X	

} Reglas

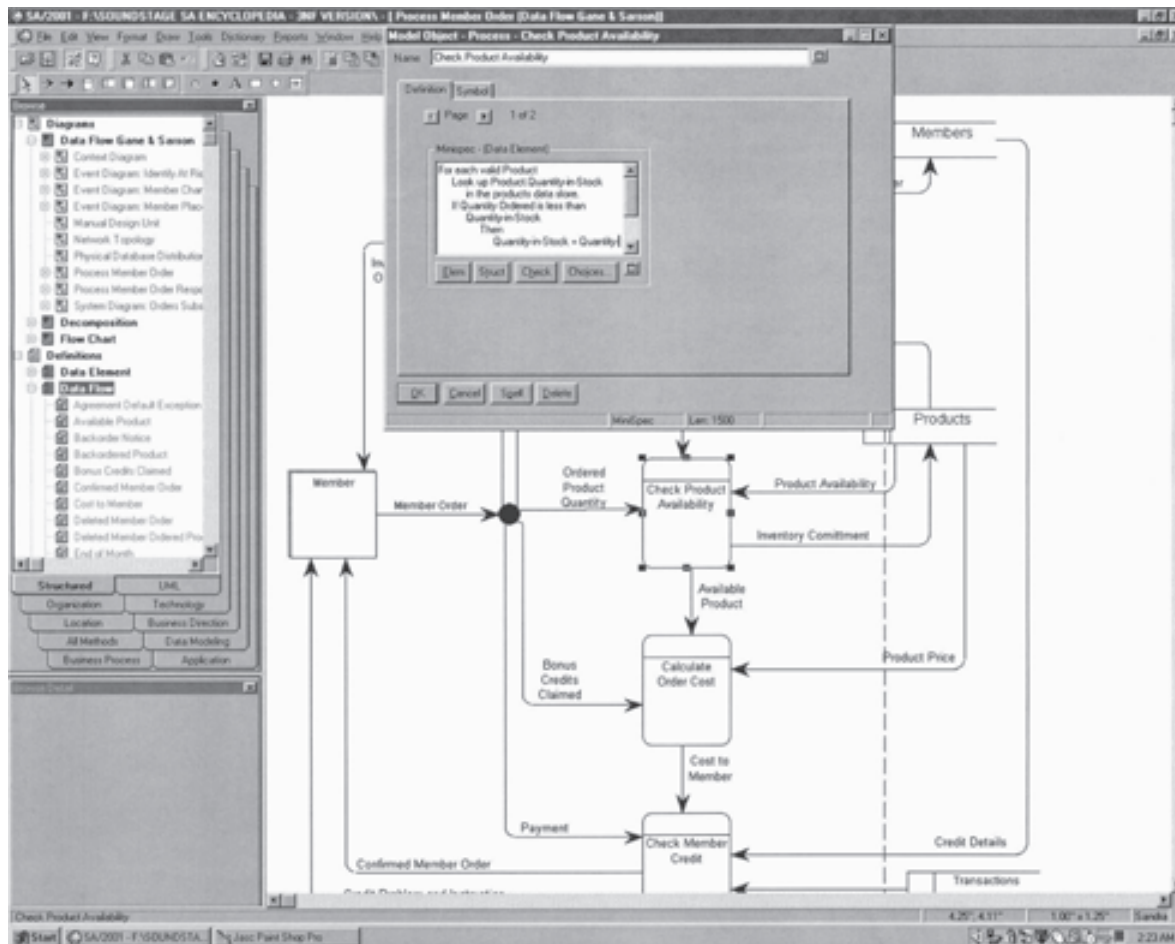
**FIGURA 8.28** Ejemplo de tabla de decisiones

- Los *talones de acción* (las filas inferiores) describen, en forma de declaraciones, las acciones posibles de política o las decisiones.
- Las *reglas* (las columnas) describen cuáles acciones deben tomarse bajo una combinación específica de condiciones.

La figura esquematiza una política de pago de cheques que aparece en la parte de atrás de una tarjeta de pago de cheques para una tienda de comestibles. Esta misma política ha sido documentada con una tabla de decisiones. Tres condiciones afectan la decisión para cobrar cheques: el tipo de cheque, si la cantidad del cheque excede el límite máximo, y si la compañía que emitió el cheque está acreditada por la tienda; las acciones (las decisiones) son ya sea pagar el cheque o rehusarse a pagar en efectivo el cheque. Observe que cada combinación de condiciones define una regla que da como resultado una acción, denotada por una letra x.

Tanto las tablas de decisión como el inglés estructurado pueden describir un proceso elemental individual. Por ejemplo, una declaración legítima en una especificación de inglés estructurado podría leerse como DETERMINE SI EL CHEQUE SE PAGA USANDO LA TABLA DE DECISIONES, POLÍTICA DE PAGOS DE CHEQUES DE LMART.

Los procesos elementales pueden describirse mediante el inglés estructurado y/o las tablas de decisión. Ya que son “elementales”, deberán describirse en una página o menos de cualquiera de las dos herramientas. La figura 8.29 demuestra cómo el *System Architect 2001* puede usarse para describir un proceso elemental. Al igual que muchas herramientas CASE, el *System Architect* no soporta la construcción de la tabla de decisiones. Por fortuna, las tablas de decisión se construyen fácilmente usando las características de la tabla en la mayoría de los procesadores de texto y las hojas de cálculo.



**FIGURA 8.29** Proceso elemental (creado con el *System Architect 2001*)

## Sincronización de los modelos de sistemas

Los datos y los modelos de proceso representan visiones diferentes del mismo sistema, pero estas visiones están relacionadas. Los modeladores necesitan sincronizar las visiones diferentes para asegurar consistencia y totalidad de la especificación total de sistema. En esta sección revisaremos los conceptos básicos de sincronización para los datos y los modelos de proceso.

### > Sincronización de los modelos de datos y de procesos

El enlace entre los modelos de datos y los procesos es casi universalmente aceptado por todas las metodologías principales. En resumen, debería existir un almacén de datos en los modelos de proceso para cada entidad de dicho modelo. Algunas metodologías exigen a las entidades asociativas de este requisito, pero creemos que es más simple (y más consistente) aplicar la regla a todas las entidades del modelo de datos.

La figura 8.30 ilustra una *matriz CLAB típica de datos a procesos*. La decisión de incluir o no incluir atributos se basa en si los procesos necesitan restringirse en relación a cuáles atributos pueden tener acceso.

Matriz CLAB de datos a procesos

Entidad. Atributo	Procesar la solicitud del cliente	Procesar la solicitud de crédito del cliente	Procesar el cambio de dirección del cliente	Procesar el cambio interno del crédito del cliente	Procesar la orden del nuevo cliente	Procesar la cancelación de la orden del cliente	Procesar el cambio del cliente a orden por pagar	Procesar el cambio interno a la orden del cliente	Procesar la adición de un nuevo producto	Procesar el retiro del producto del mercado	Procesar el cambio del precio del producto	Procesar el cambio a la especificación del producto	Procesar el ajuste de inventario del producto
Cliente	C	C			L	L	L	L					
.Número de cliente	C	C			L	L	L	L					
.Nombre del cliente	C	C	A		L		L	L					
.Dirección del cliente	C	C	A		LA		LA	LA					
.Clasificación de crédito del cliente		C		A	L		L	L					
.Saldo deudor del cliente					LA	A	L	L					
Orden					C	B	LA	LA					
.Número de orden					C		L	L					
.Fecha de la orden					C		A	A					
.Cantidad de la orden					C		A	A					
Producto ordenado					C	B	CLAB	CLAB		LA			
.Cantidad ordenada					C		CLAB	CLAB					
.Precio unitario del artículo ordenado					C		CLAB	CLAB					
Producto					L	L	L	L	C	B	LA	LA	LA
.Número de producto					L	L	L	L	C			L	
.Nombre del producto					L		L	L	C			LA	
.Descripción del producto					L		L	L	C			LA	
.Unidad de medición del producto					L		L	L	C		LA	LA	
.Precio unitario actual del producto					L		L	L			A		
.Cantidad de producto en la mano					LA	A	LA	LA					LA

C = crear    L = leer    A = actualizar    B = borrar

**FIGURA 8.30** Ejemplo de matriz CLAB de datos a procesos

La comprobación de calidad de la sincronización se enuncia como sigue:

Cada entidad deberá tener *al menos* una C, una L, una A y una entrada B para la totalidad del sistema. En caso de que no, probablemente se omitieron uno o más procesos de eventos de los modelos de procesos. Más importante aún, los usuarios y la gerencia deberán validar que se han incluido todos los comandos posibles create, read, update y delete.

La matriz provee una verificación simple de la calidad que es más simple de leer que los modelos de datos o de proceso. Por supuesto, cualesquiera errores y omisiones deberán registrarse tanto en la matriz como en los modelos de proceso y de datos correspondientes para asegurar la sincronización correcta.

### > Distribución de procesos

Los modelos de proceso ilustran el trabajo esencial para ser realizado por el sistema como un todo. Sin embargo, los procesos deben distribuirse entre todas las ubicaciones donde el trabajo deba realizarse. Algún trabajo puede ser único para una ubicación. Otro trabajo puede realizarse en ubicaciones múltiples. Antes de que diseñemos el sistema de información, debemos identificar y documentar los procesos que deben realizarse en dichas ubicaciones. Esto puede lograrse a través de una *matriz de asociación entre procesos y ubicaciones*. Una **matriz de asociación entre procesos y ubicaciones** es una tabla en la cual las filas indican procesos (eventos o procesos elementales), las columnas indican ubicaciones y las celdas (la intersección de filas y columnas) documentan qué procesos deben realizarse y en cuáles ubicaciones. La figura 8.31 ilustra una matriz típica de asociación de procesos a ubicaciones. Una vez que se valide la exactitud, el diseñador de sistemas la usará para determinar cuáles procesos deberían implementarse en forma central o local.

Algunas metodologías y herramientas CASE pueden soportar visiones del modelo de proceso que son apropiados para una ubicación. Si es así, estas visiones (los subconjuntos de los modelos de proceso) deben mantenerse en sincronía con los modelos maestros de proceso del sistema como un todo.

**matriz de asociación entre procesos y ubicaciones** Tabla usada para documentar los procesos y la ubicación en la que se deben ejecutar.

**Matriz de asociación entre procesos y ubicaciones**

Proceso	Clientes	Kansas City	Mercadotecnia	Publicidad	Almacén	Ventas	Cuentas por cobrar	Boston	Ventas	Almacén	San Francisco	Ventas	San Diego	Almacén
Procesar la solicitud del cliente	X					X			X			X		
Procesar la solicitud de crédito del cliente	X						X							
Procesar el cambio de dirección del cliente	X					X			X			X		
Procesar el cambio de crédito interno del cliente							X							
Procesar la orden del nuevo cliente	X					X			X			X		
Procesar la cancelación de la orden del cliente	X					X			X			X		
Procesar el cambio de cliente a orden por pagar	X					X			X			X		
Procesar el cambio interno a la orden del cliente						X			X			X		
Procesar la adición de un nuevo producto			X											
Procesar el retiro del producto del mercado			X											
Procesar el cambio de precio del producto			X											
Procesar el cambio de la especificación del producto			X	X										
Procesar el ajuste del inventario del producto					X					X				X

**FIGURA 8.31** Ejemplo de matriz de asociación entre procesos y ubicaciones a posición

Si usted está tomando un enfoque tradicional, proseguirá directamente al capítulo 9, “Análisis de factibilidad y propuesta del sistema”. Una vez que se han mostrado los modelos de procesos y de datos que describen los requisitos lógicos del sistema, el capítulo 9 examinará los métodos y las técnicas para identificar las soluciones *físicas* alternativas que satisfagan los requisitos lógicos, los aspectos técnicos para analizar la factibilidad de cada una de estas soluciones, y los enfoques para presentar la solución que usted considere como más factible. Esta propuesta de sistema culmina el análisis de sistemas en nuestra metodología *FAST*.

Si usted está interesado en cómo vamos a usar los DFD lógicos en este capítulo durante el diseño de sistemas, tal vez quiera inspeccionar el capítulo 11, “Modelado y arquitectura de la aplicación”. En ese capítulo, le enseñaremos cómo transformar los diagramas de flujo de datos lógicos en diagramas de flujo de datos físicos que modelen la arquitectura de tecnología de un sistema que va a diseñarse y a implementarse.



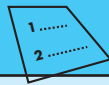
## Resumen

1. Construimos modelos lógicos para entender mejor los dominios y requisitos de los problemas de negocios.
2. La modelación de procesos es una técnica para organizar y documentar los requisitos de proceso y el diseño de un sistema. Este capítulo enfocó la atención en un modelo de proceso llamado diagrama de flujo de datos, que esquematiza el flujo de datos a través de los procesos de un sistema.
3. Los agentes externos son entidades que están fuera del alcance de un sistema y del proyecto pero que proveen entradas netas o salidas netas de un sistema. Como tal, forman las fronteras del sistema.
4. Los almacenes de datos presentan los archivos de datos que el sistema va a usar y mantener. Un almacén de datos en un modelo de proceso corresponde a todas las instancias de una entidad dentro de un modelo de datos.
5. Un sistema es un proceso. Un proceso es un trabajo que se realiza, o una respuesta a entradas y condiciones.
6. Tal como los sistemas pueden descomponerse recursivamente en subsistemas, los procesos pueden descomponerse recursivamente en subprocesos. Un diagrama de descomposición muestra la descomposición funcional de un sistema en procesos y subprocesos. Es una herramienta de planificación para los siguientes diagramas de flujo de datos.
7. Los procesos lógicos muestran el trabajo esencial para ser realizado por un sistema sin mostrar cómo serán implementados los procesos. Hay tres tipos de procesos lógicos: las funciones (el nivel muy alto), los eventos (nivel intermedio de detalle) y los procesos elementales (muy detallado).
8. Los procesos elementales se describen con más profundidad por la lógica procedural. El inglés estructurado es una herramienta para expresar esta lógica procedural. Se deriva de las construcciones de la lógica estructurada de programación unidas con el inglés natural.
9. Los procesos elementales complejos pueden estar descritos por políticas que se expresan en las tablas de decisión, que demuestran combinaciones complejas de condiciones que dan como resultado acciones específicas.
10. Los flujos de datos son las entradas y las salidas de los procesos. También ilustran accesos a los almacenes de datos y actualizaciones.
11. Todos los flujos de datos constan ya sea de otros flujos o de estructuras discretas de datos que incluyen atributos descriptivos. Un flujo de datos deberá contener sólo la cantidad de datos necesarios para un proceso; a esto se le llama conservación de datos.
12. La modelación de procesos puede usarse en diferentes tipos de proyectos, incluyendo el re-

diseño de los procesos de negocios y el desarrollo de aplicaciones. Para los proyectos de desarrollo de aplicación, este capítulo enseñó una estrategia de diagramación de flujos de datos impulsados por los eventos tal como sigue:

- a) Dibuje un diagrama de flujo de datos de contexto que muestre cómo el sistema tiene interfaces con otros sistemas, las organizaciones de negocios y las externas.
  - b) Dibuje un diagrama de descomposición funcional que muestre los subsistemas y/o funciones claves que integran al sistema.
  - c) Cree una lista de eventos que identifique los eventos externos y temporales para los cuales el sistema debe proveer una respuesta. Los eventos externos son desencadenados por los agentes externos de un sistema. Los eventos temporales son desencadenados por el transcurso del tiempo.
  - d) Actualice el diagrama de descomposición para incluir procesos para manejar los eventos (un proceso por evento).
- e) Para cada evento, dibuje un diagrama de eventos que muestre sus interacciones con entidades externas, almacenes de datos y, en ocasiones, con otros disparadores de otros acontecimientos.
  - f) Combine los diagramas de eventos en uno o más diagramas del sistema.
  - g) Para cada evento en el diagrama de sistemas, descríbalos como un proceso elemental usando inglés estructurado o bien expándalo a un diagrama elemental de flujo de datos que incluya los procesos elementales que deben describirse subsecuentemente, ya sea por el inglés estructurado o por las tablas de decisión, o por ambos. Cuando los procesos se expanden a diagramas de flujo de datos para revelar mayor detalle es importante mantener la consistencia entre los diferentes tipos de diagramas, a lo cual se le llama sincronización.
13. La mayoría de las herramientas de ingeniería de software asistida por computadora soportan la diagramación de descomposición y la diagramación de flujo de datos.

## Preguntas de repaso



1. ¿Qué es un modelo lógico y cuáles son sus sinónimos comunes?
2. ¿Por qué los modelos lógicos son herramientas valiosas para los analistas de sistemas?
3. ¿Qué es un diagrama de flujo de datos y cuáles son sus sinónimos comunes?
4. ¿En qué difiere un diagrama de flujo de datos de un diagrama de flujo?
5. ¿Por qué se considera que un sistema es un proceso?
6. ¿Qué es la descomposición y por qué es necesaria? ¿Cuál es la herramienta que se usa para esquematizar la descomposición de un sistema?
7. ¿Cuáles son los tres tipos de procesos lógicos?
8. ¿Cuáles son los errores mecánicos comunes cuando se esquematizan los procesos en un diagrama de flujo de datos y en otros modelos de proceso?
9. ¿Qué es el inglés estructurado y por qué se usa al construir la lógica de proceso?
10. ¿Cuáles son las convenciones para otorgar nombres de los flujos de datos lógicos?
11. ¿Qué es la conservación de datos y por qué se necesita?
12. ¿Cuáles son los agentes externos y por qué pueden cambiar los agentes externos de un sistema de información?
13. ¿Cuáles son algunos ejemplos de la modelación impulsada por eventos que se usa en el análisis de sistemas?
14. ¿Qué modelo de procesos se usa para documentar el alcance para un sistema de información, y qué se esquematiza en este modelo de procesos?
15. ¿Por qué es importante sincronizar los modelos de datos y de proceso?

## Problemas y ejercicios



1. Usted funge como asistente estudiantil para una firma de Ingeniería y es pagado por hora. Cada quincena, usted devuelve una hoja de asistencia a su supervisor, y tres días hábiles después, su cheque de salario se deposita directamente en su cuenta de cheques. Liste los diferentes objetos o entidades, procesos lógicos, flujos de datos y almacenes de datos que intervienen a partir del tiempo que usted entrega su hoja de asistencia.
2. Coteje los términos en la primera columna con las definiciones o los ejemplos en la segunda columna:
 

<ol style="list-style-type: none"> <li>1. Inglés estructurado</li> <li>2. Proceso</li> <li>3. Modelo lógico</li> <li>4. Proceso elemental</li> </ol>	<ol style="list-style-type: none"> <li>A. Dividir un sistema en sus componentes</li> <li>B. La unidad lógica de trabajo que debe completarse como un todo</li> <li>C. Herramienta para la modelación lógica</li> <li>D. Conjunto de actividades de negocios que están relacionadas y en curso</li> </ol>
--	--



- |                            |   |
|----------------------------|---|
| 5. Política                | E. Técnica para organizar y documentar los procesos de un sistema                           |
| 6. DFD                     | F. Lenguaje de especificación de procedimientos   |
| 7. Tabla de decisiones     | G. Ilustración de lo que un sistema es o lo que hace  |
| 8. Descomposición          | H. Ilustración de un flujo de datos de sistema  |
| 9. Evento                  | I. Ilustración de la descomposición de sistemas   |
| 10. Modelo físico          | J. Trabajo realizado por el sistema en respuesta a los flujos de datos entrantes            |
| 11. Función                | K. Actividad /tarea separada, detallada necesaria para completar la respuesta a los eventos |
| 12. Gráfica de jerarquía   | L. Reglas de terminación de procesos  |
| 13. Modelación de procesos | M. Ilustración de lo que un sistema es o lo que hace, y cómo se implementa                  |

3. En un diagrama de descomposición, ¿cómo muestra usted un hijo para un padre y cómo muestra usted más de un padre para un hijo? ¿Por qué las conexiones en un diagrama de descomposición no muestran puntas de flecha, como la mayoría de los otros diagramas? ¿Por qué las conexiones no tienen nombre?
4. Considere los carriles para transporte colectivo voluntario en Sacramento, California. Entre las horas 6:00 a.m. a 10:00 a.m. y 3:00 p.m. a 7:00 p.m., lunes a viernes, se restringen a vehículos de pasajeros con dos o más personas de cualquier edad, motocicletas y vehículos híbridos (gas/electricidad) con una o más personas. Para todos los demás vehículos o condiciones, el conductor está sujeto a una multa. Fuera de esos lapsos de tiempo, no hay restricciones en lo que se refiere a su uso. Basándose en esta información, escriba una tabla de decisiones de política para el uso de los oficiales de patrullaje de carreteras.
5. Usted trabaja en la oficina del cuartel general de la división de investigación de una delegación de policía y está desarrollando un sistema automatizado de rastreo de casos para la oficina del cuartel general para reemplazar el sistema manual actual. Los casos son abiertos al público cuando se recibe una forma de petición de investigación de otras divisiones en su delegación; ninguno de los casos se inicia internamente. Se crea una carpeta nueva de caso, que contiene cualquier información de registro criminal basada en la verificación de diversas bases de datos de justicia criminal, y luego se manda a la oficina de investigación de campo apropiada. Cuando

el caso se completa, el cuartel general recibe un informe de investigación de la oficina de campo, el caso se cierra, y una copia del reporte completo de investigación se envía a la oficina generadora. Cada semana se manda a cada una de las oficinas del campo un listado que muestra los casos abiertos, completados y en marcha. ¿Cuáles son algunas de las estrategias que usted podría usar para establecer el alcance y las fronteras del sistema?

6. Basado en la pregunta anterior, ¿qué determinó usted como las partes del sistema?:
  - a) Entradas netas
  - b) Salidas netas
  - c) Agentes externos
  - d) Almacenes externos
7. Ahora que usted tiene sus entradas netas, salidas netas, agentes externos y almacenes externos, dibuje un diagrama de flujo de datos de contexto.
8. Ahora usted está modelando la lógica de cada proceso elemental para el sistema que rastrea los casos y ha decidido escribirla en inglés estructurado para comunicarse eficazmente con los usuarios y los programadores. Para el reporte de listado de casos abiertos/cerrados/en curso para la oficina de campo, escriba una declaración en inglés estructurado para documentar el proceso de mantener un total móvil de casos abiertos, cerrados y en marcha. Adicionalmente, usted quiere añadirle otra columna al reporte mostrando el número de casos todavía en progreso con una antigüedad mayor a seis meses.
9. Ahora usted está listo para crear el diagrama de descomposición funcional para el sistema de rastreo de casos. ¿Cuál es el proceso raíz para el diagrama de descomposición de funciones? ¿Qué subsistemas incluiría usted típicamente? ¿Qué procesos mostraría usted y a que subsistema pertenecerían? Use esta información para crear un diagrama de descomposición de funciones.
10. Su siguiente paso, después de dibujar el diagrama de descomposición, es crear la lista de las respuestas a los eventos o de casos prácticos. Deberá haber un caso práctico para cada evento iniciado por un agente externo. Los eventos temporales también deberán mostrarse. Para cada agente externo deberá haber al menos un caso práctico. Inicie una tabla parcial de casos prácticos que incluya los eventos de OTRA DIVISIÓN ENVÍA UNA PETICIÓN DE INVESTIGACIÓN Y LA OFICINA DE CAMPO ENVÍA UN REPORTE COMPLETADO DE INVESTIGACIÓN. También incluya el evento GENERAR UN REPORTE DE LISTADO DE CASOS ABIERTOS/CERRADOS/EN CURSO.
11. Un diagrama de eventos equivale a un diagrama de contexto para un evento. El diagrama de eventos incluye las entradas, las salidas y las interacciones de almacén de datos relacionadas con ese evento específico. Su propósito es ayudar a los usuarios a que se centren en un evento individual sin abrumarse o confundirse por un cuadro del sistema entero. Seleccione uno de los eventos del sistema de rastreo de casos y dibuje un diagrama de eventos.

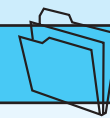
12. Haga coincidir las definiciones o los ejemplos de la primera columna con los términos de la segunda columna:
- |   |                              |   |                         |
|---|------------------------------|---|-------------------------|
| A. Segmento de datos significativos más pequeño                   | 1. Dominio                   | K. Símbolo que dado el flujo de datos es instancia de sólo un tipo              | 11. Estructura de datos |
| B. Combinación de flujos de datos que son similares               | 2. Conexión                  | L. Datos en reposo  | 12. Tipo de datos       |
| C. Expresado en forma de estructuras de datos                     | 3. Agente externo            | M. El sistema se descompone en subsistemas basándose en los eventos de negocios | 13. Almacén de datos    |
| D. Condición a ser monitoreada                                    | 4. Partición de eventos      |   |                         |
| E. Valores legítimos de un atributo                               | 5. Flujo de datos compuestos |   |                         |
| F. “Inanición de los procesos”                                    | 6. Flujo de control          |   |                         |
| G. Salida de datos de un proceso o entrada al proceso             | 7. Composición de datos      |   |                         |
| H. Clase de datos que pueden almacenarse en un atributo           | 8. Atributo de datos         |   |                         |
| I. Arreglo de atributos de datos que comprenden un flujo de datos | 9. Conservación de datos     |   |                         |
| J. Entidad exterior que interactúa con un sistema                 | 10. Flujo de datos           |   |                         |
13. Aunque los datos y los modelos de proceso esquematizan el mismo sistema con puntos de vista diferentes, los diseñadores de sistema deben sincronizar estos puntos de vista diferentes para asegurarse de que sus modelos son consistentes y completos. Una forma de asegurar esto es a través de una matriz CLAB de datos a procesos. Seleccione un sistema con el cual usted esté familiarizado. Identifique al menos tres de las entidades usadas en ese sistema y sus atributos. Después, identifique los procesos asociados con esas entidades. Luego construya una matriz CLAB de datos a procesos, usando la figura 8.30 como su plantilla. Como una verificación de calidad, muéstrele su matriz a alguien más que esté familiarizado con el sistema y haga que la revisen en cuanto a completud y corrección.

## Proyectos e investigación



- Suponga que usted comienza a trabajar en un proyecto para una organización que nunca ha usado ninguna herramienta o técnica de modelación para diseñar un sistema. (Sí, es difícil de imaginar, pero eso existe.) Su gerente se resiste a cambiar la forma en que siempre se han hecho las cosas. Escriba un ensayo de una o dos hojas (o una presentación en PowerPoint como alternativa) acerca de por qué la modelación de sistemas vale el tiempo y recursos involucrados.
- El libro de texto usa las notaciones de modelación de procesos de Gane y Sarson y las compara con las notaciones usadas por las metodologías de modelación de procesos de DeMarco/Yourdon y de SSADM/IDEF0. Indague al menos dos de estos u otro proceso de metodologías de modelación, luego compárelos y contrástelos.
  - ¿Qué otro proceso de metodologías de modelación encontró usted?
  - ¿Cuáles, si es que las hay, son las diferencias significativas en sus metodologías de modelación de procesos, aparte de aquéllas en sus métodos de notación?
  - ¿Cuáles son sus similitudes?
  - ¿Qué método de notación usa su organización?
- Si usted recibió instrucciones de recomendar una de estas metodologías para su organización, ¿cuál escogería? ¿Por qué?
- Hasta hace poco, las revistas y publicaciones estaban disponibles sólo en versiones impresas. Los editores publicistas ahora le ofrecen un número creciente de publicaciones ya sea en la versión impresa tradicional o en un formato digital que puede descargarse en la Red. Considere los procesos involucrados en estos dos métodos y:
  - Cree un diagrama de flujo de datos de alto nivel que describa los métodos tradicionales típicos de renovación de una suscripción por correo a la versión impresa de una revista.
  - Cree otro diagrama de flujo de datos de alto nivel que describa los procesos para renovar una suscripción por la vía de Internet a la versión del formato digital de una revista.
  - ¿Cuáles son, si hay algunas, las diferencias esenciales entre los dos diagramas?
  - Basándose en los diagramas de flujo de datos, ¿cuál formato es más eficiente para renovar una suscripción de la revista? ¿Piensa usted que lo mismo es verdad desde la perspectiva del sus-

- criptor? ¿Qué hay acerca de la perspectiva del editor publicista?
- e) ¿Qué hay acerca de recibir y leer la revista? Desde su propia perspectiva, ¿cuáles son las ventajas y desventajas de una revista publicada en formato digital contra la versión impresa tradicional?
4. En 1978, Tom DeMarco escribió lo que se considera ser el texto clásico en la metodología del análisis de sistemas estructurados: *Structured Analysis and System Specification*. James Wetherbe es considerado por los autores del libro de texto como “uno de los defensores más fuertes de los conceptos de sistemas y del pensamiento en sistemas como parte de la disciplina de análisis y diseño de sistemas”, y ha escrito numerosos artículos y libros sobre “el pensamiento en sistemas”. Edward Yourdon es otro líder ampliamente reconocido en el diseño de sistemas y se le recuerda por formalizar la metodología de enfoques impulsada por eventos en su libro de 1989, *Modern Structured Analysis*. Haga una búsqueda en Internet de sus sitios Web, si los hay, y de artículos y/o libros recientes de estos tres líderes en el análisis y diseño de sistemas.
- a) ¿Qué artículos y/o sitios de Web encontró usted?  
 b) Describa una parte de su trabajo más reciente.  
 c) Compare y contraste a cada uno de estos autores en términos de su perspectiva del análisis y el diseño de sistemas; específicamente, ¿ve usted sus enfoques como complementarios u opuestos?  
 d) ¿Qué tendencias o cambios emergentes en el análisis y diseño de sistemas prevén?
- e) ¿Considera usted que todavía representan a los pensadores líderes en el análisis y diseño de sistemas? ¿Por qué?
5. Considere los diferentes sistemas de información usados en su escuela u organización. Encuentre un sistema con documentación incompleta y/o obsoleta (esto no debería costar trabajo en la mayoría de organizaciones?). Actualice y complete la documentación para este sistema usando diagramas de flujos de datos, estructuras de datos y otros modelos de procesos descritos en este capítulo. Use el método de notación de Gane y Sarson, a menos que su organización soporte un método diferente de notación.
6. En la conclusión de este capítulo, el libro de texto menciona que la modelación de los datos y procesos finalmente puede caer en desuso debido al uso y popularidad creciente del análisis y modelado orientado a objetos con UML. Indague artículos discutiendo este tema en su biblioteca de la escuela y/o Internet.
- a) ¿Qué artículos encontró usted?  
 b) ¿Cuáles fueron las posiciones de los autores sobre este tema?  
 c) Compare y contraste los artículos que usted encontró; ¿cuál fue el que tuvo el caso más convincente?
- A medida que usted escudriña en su bola de cristal, ¿cuál metodología de modelación piensa usted que será la más ampliamente usada dentro de 10 años? ¿Predice usted que será alguna de estas dos metodologías o una completamente diferente? ¿Por qué?



## Casos breves

1. Vaya a una compañía pequeña de su elección. ¿Qué hace el negocio? Escriba un ensayo de una o dos páginas describiendo el negocio y su sistema existente. Luego dibuje un diagrama nivelado en contexto y un diagrama nivelado en sistema para el sistema existente. ¿Ve usted ineficiencias o debilidades en el sistema actual? Descríbalas.
2. En el caso previo, usted documentó (en un nivel alto) un sistema de información existente en un negocio de su elección. Ahora describa el sistema que *usted* piensa es apropiado para este negocio. Considere la eficiencia, el flujo de información de un departamento para otro y así sucesivamente. ¿Hay información pertinente que el sistema previo no utilizó? ¿Cómo le puede ofrecer su sistema nuevo una ventaja de negocio? Documente esta ventaja en un ensayo de dos páginas, así como en un diagrama de nivel de contexto y de sistema.
3. En el capítulo 2 usted se ocupó del siguiente problema: Los departamentos de servicio del gobierno están profundamente agobiados por la cantidad de datos que sujetan y procesan. Entreviste a alguien de un departamento de servicios y redacte un ensayo pequeño. Los ejemplos de los departamentos de servicio en donde se deben examinar rápidamente cantidades inmensas de datos son de los que tratamos, por ejemplo, de personas desaparecidas, servicios protectores de los niños, DMV y rastreo de personas a prueba después de un crimen. Usted deberá incluir, pero no deberá limitarse a, temas tales como:
- ¿Cuál es el trabajo del departamento (o de la persona)?
  - ¿Qué tipo de datos recolectan y analizan?
  - ¿Qué tipo de análisis realizan con los datos?
  - ¿Cuánta información recolectan y de quiénes y qué programas usan?

En este ejercicio, utilice la información que usted recogió de su entrevista y esquematice un DFD completo del sistema de información *existente* para ese departamento. Usted puede necesitar regresar y reentrevistar a su contacto u obtener formas, informes y pronto, a fin de que usted tenga un cuadro completo del flujo de datos.

4. En el caso previo, usted documentó el sistema de información existente de un organismo oficial.

Ahora piense acerca de cómo *debería ser* el flujo de información. Discuta en un ensayo corto el flujo existente de información y la clase de información que usted piensa debería estar en el sistema. ¿Hay una diferencia grande entre lo que usted piensa que deberían tener y lo que ellos tienen?

## Ejercicios de equipo e individuales



1. Debate de mesa redonda personal/clase. Se ha dicho que la información es poder y que la separación entre las naciones económicamente ricas, así como individuos, compañías y cosas por el estilo, y aquellos que no lo son (económicamente ricos), es el uso y el control de la información. ¿Cuánto vale la información? ¿Cómo afecta a la viabilidad y la competitividad de naciones o compañías? ¿Cómo afecta su acceso y uso a la información a su habilidad para tener éxito en un ambiente relacionado con el trabajo?
2. Debate de mesa redonda personal/clase. Suponga que una compañía de software ha sido contratada para crear un paquete de software muy complicado y de vanguardia para una compañía. A la compañía le gustaría poseer los derechos (incluyendo modificación y venta) de ese software, pero la compañía de software también quiere los derechos del mismo. Encuentre un ejemplo de esta situación en la vida real. ¿Quién piensa usted que *debería* poseer el software? ¿Por qué? ¿A quién apoya la ley? ¿Por qué?

## Lecturas recomendadas



- Copi, I. R. *Introduction to Logic*. Nueva York: Macmillan, 1972. Copi provee varias ilustraciones de solución de problemas y ejercicios que ayudan al estudio de la lógica. El problema del chip para póquer en nuestro ejercicio se adaptó de uno de los ejercicios de razonamiento de Copi.
- DeMarco, Tom. *Structured Analysis and System Specification*. Englewood Cliffs, NJ: Prentice Hall, 1978. Éste es el libro clásico sobre la metodología del análisis de los sistemas estructurados, el cual se fundamenta fuertemente en el uso de los diagramas de flujo de datos. La progresión a través de 1) *DFDs de los sistemas físicos actuales*, 2) *DFDs de los sistemas lógicos actuales*, 3) *DFDs de los sistemas lógicos objetivos*, y 4) *DFDs de los sistemas físicos objetivos* ya casi no se practica, pero la esencia del trabajo pionero de DeMarco persiste en el análisis estructurado impulsado por eventos. DeMarco creó las notaciones lógica y de estructura de datos que se usan en este libro.
- Gildersleeve, T. R. *Successful Data Processing Systems Analysis*. Englewood Cliffs, NJ: Prentice Hall, 1978. La primera edición de este libro incluye un capítulo completo sobre la construcción de las tablas de decisiones. Gildersleeve hace un excelente trabajo al demostrar cómo las descripciones de los procesos narrativos se traducen en elementos de condición y acción en las tablas de decisiones. Desafortunadamente, el capítulo se suprimió de la segunda edición.
- Harmon, Paul, y Mark Watson. *Understanding UML: The Developers Guide*. San Francisco: Morgan Kaufman Publishers, 1998. Este libro hace un excelente trabajo al introducir los casos de uso.
- Martin, James, y Carma McClure. *Action Diagrams: Towards Clearly Specified Programs*. Englewood Cliffs, NJ: Prentice Hall, 1986. Este libro describe la gramática formal del inglés estructurado que alienta la progresión natural de un proceso (programa) del inglés estructurado al código. Los diagramas de acción tienen soporte directo en algunas herramientas de CASE.
- Matthies, Leslie H. *The New Playscript Procedure*. Stamford, CT: Office Publications, 1977. Este libro provee una explicación completa y proporciona ejemplos de las debilidades de la lengua inglesa como herramienta para especificar los procedimientos de negocios.
- McMenamin, Stephen M., y John F. Palmer. *Essential Systems Analysis*. New York: Yourdon Press, 1984. Este fue el primer libro en sugerir la partición de eventos como una estrategia formal para mejorar el análisis estructurado. El libro reforzó la diferencia entre los modelos de procesos lógicos y físicos y la importancia creciente de los modelos lógicos (a los cuales llamaron modelos *elementales*).
- Robertson, James, y Suzanne Robertson. *Complete Systems Analysis* (vols. 1 y 2). Nueva York: Dorset House Publishing, 1994. Éste es el libro más actualizado y entendible sobre el enfoque impulsado por eventos del análisis estructurado, aun cuando pensamos que pone demasiado énfasis en los modelos físicos y de sistemas actuales más que el libro de Yourdon descrito en seguida.
- Notas del seminario para *Process Modeling Techniques*. Atlanta: Structured Solutions, Inc., 1991. Probablemente usted no

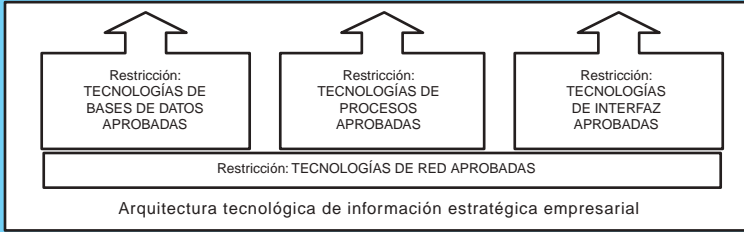
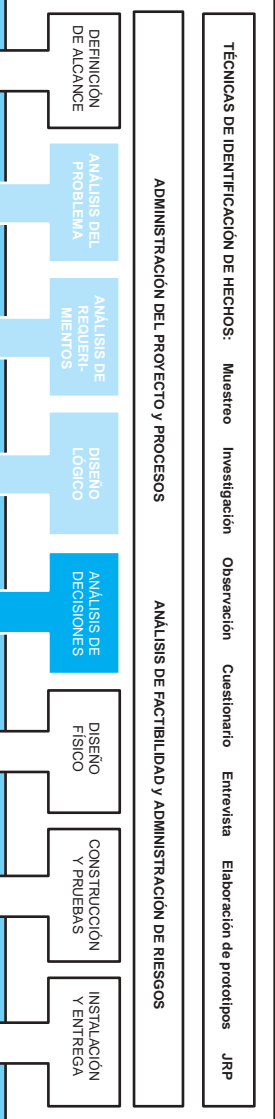
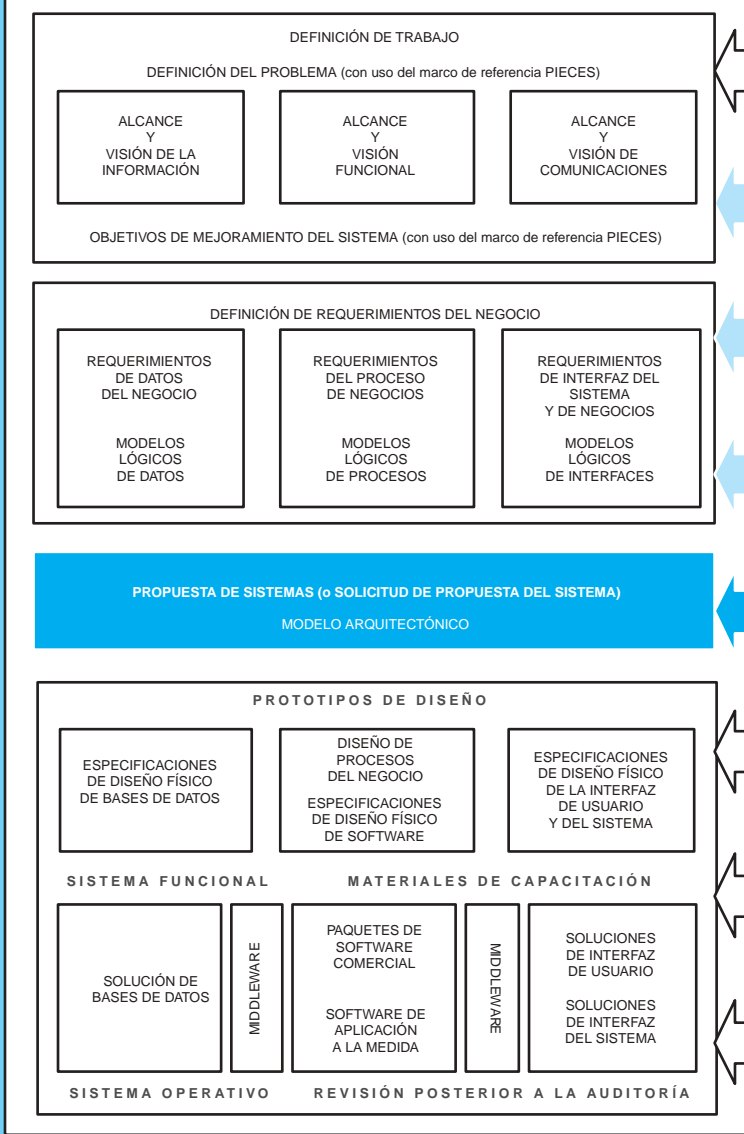
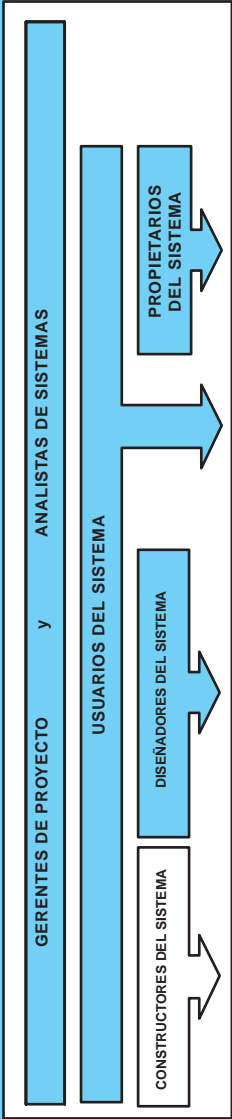
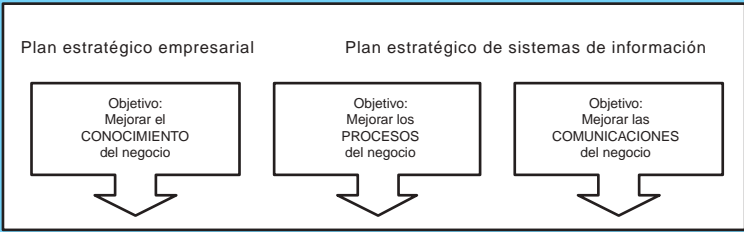
pueda obtener una copia de estas notas, pero queríamos darles un reconocimiento a los instructores del curso de metodología *AD/Method* que estimuló nuestro pensamiento y motivó nuestra separación de las técnicas clásicas del análisis estructurado hacia las técnicas impulsadas por eventos del análisis estructurado que se enseñan en este capítulo. Structured Solutions fue adquirido por Protelicess, Inc.

Wetherbe, James, y Nicholas P. Vatarli. *Systems Analysis and Design: Best Practices*, 4a. ed. St. Paul, MN: West Publishing, 1994. Jim Wetherbe siempre ha sido uno de los más ardientes partidarios de los conceptos de sistemas y del pensamiento de sistemas como parte de la disciplina del análisis y el diseño

de sistemas. Jim ha modelado muchas mentes, incluyendo la nuestra. Los autores proveen un bonito capítulo sobre los conceptos de sistemas en este libro; y el resto del libro es de lectura obligada para aquellos de ustedes que realmente quieran aprender el “pensamiento de los sistemas”.

Yourdon, Edward. *Modern Structured Analysis*. Englewood Cliffs, NJ: Yourdon Press, 1989. Éste fue el primer libro de actualidad en abandonar el énfasis excesivo del análisis estructurado clásico sobre los modelos de sistemas físicos actuales y en formalizar el enfoque impulsado por los eventos de McMenamin and Palmer.





# Análisis de factibilidad y propuesta del sistema

## Panorámica y objetivos del capítulo

Un buen analista de sistemas evalúa soluciones alternas antes de proponer cambios. En este capítulo, usted aprenderá cómo analizar y documentar esas alternativas con base en cuatro criterios de factibilidad: operativos, técnicos, económicos y de calendario. Además, estudiará cómo elaborar una propuesta de sistema en la forma de un informe escrito y una presentación formal. Reconocerá que ha entendido el análisis de factibilidad y las habilidades de recomendaciones que necesitan los analistas de sistemas cuando pueda:

- Identificar los puntos de comprobación de factibilidad en el ciclo de vida del sistema.
- Identificar soluciones de sistema alternas.
- Definir y describir seis tipos de factibilidad y sus criterios respectivos.
- Realizar diversos análisis de costo-beneficio mediante costos y beneficios ajustados en el tiempo.
- Redactar informes de propuestas de sistemas adecuados para diversos auditorios.
- Planear una presentación formal para los propietarios y usuarios del sistema.

## Introducción

A medida que avanza el análisis del proyecto de sistemas de servicios a miembros de SoundStage, Bob Martínez está cada vez más emocionado. El programador que hay en él quisiera brincar e iniciar la codificación del sistema de información. Sin embargo, Sandra, su jefa, hizo que investigara las soluciones de paquetes existentes en el mercado. Resultaron muy caras. Entonces, Sandra calculó los costos de mano de obra de la programación en la compañía. Bob se dio cuenta de que las soluciones de paquetes eran comparativamente baratas y se podrían instalar mucho más rápidamente. Todavía sería necesario programar, ya que las soluciones de paquetes tendrían que adaptarse para satisfacer todos los requerimientos.

La decisión final de elección de la solución debía tomarla el comité de dirección que supervisaba el proyecto. Sandra dijo que los ejecutivos del comité de dirección estaban muy conscientes de las cuestiones presupuestarias. Revisarían las cifras y aprobarían la continuación del proyecto sólo si se acompañaba de un rendimiento sobre la inversión apropiado. Bob tendría una pequeña participación en la presentación de la propuesta del sistema. Ensayó y estudió los hechos para tener la certeza de estar listo para cualquier pregunta. No quería estropear la presentación. Le sorprendió darse cuenta de que ahora estaba contento de que en algunos cursos de la universidad le hubieran pedido ponerse traje y elaborar una presentación formal.

## Análisis de factibilidad y la propuesta de sistema

En el mundo de negocios actual, es cada vez más evidente que los analistas deben aprender a pensar como gerentes del negocio.

Las aplicaciones de computadoras se están extendiendo a un ritmo sin precedentes. Ahora, más que nunca, los administradores esperan que los sistemas de información sean rentables. La información es una inversión de capital significativa que debe justificarse, de igual modo que el área de marketing debe justificar un nuevo producto, y la de manufactura, una nueva planta o equipo. Ahora se pide a los analistas, más que antes, responder preguntas como las siguientes: ¿se recuperará la inversión? ¿Existen otras inversiones que generarían un rendimiento sobre la inversión todavía mayor?

Este capítulo trata sobre los problemas del análisis de factibilidad de interés para los analistas de sistemas y usuarios de los sistemas de información. También enfatiza la importancia de elaborar recomendaciones para los administradores en la forma de una propuesta de sistema, que es un informe escrito formal o una presentación oral. Como se muestra en la página de inicio del capítulo, el análisis de factibilidad es apropiado para las fases de análisis de sistemas, si bien reviste importancia especial en la fase de análisis de decisiones. La propuesta de sistema representa el producto y muestra la solución técnica de CONOCIMIENTO, PROCESO y COMUNICACIÓN.

### > Análisis de factibilidad: una estrategia de compromiso creciente

Empecemos por definir formalmente la factibilidad y el análisis de factibilidad. La **factibilidad** es una medición de cuán beneficioso o práctico será el desarrollo de un sistema de información para la organización. El **análisis de factibilidad** es el proceso con el que se mide la factibilidad misma.

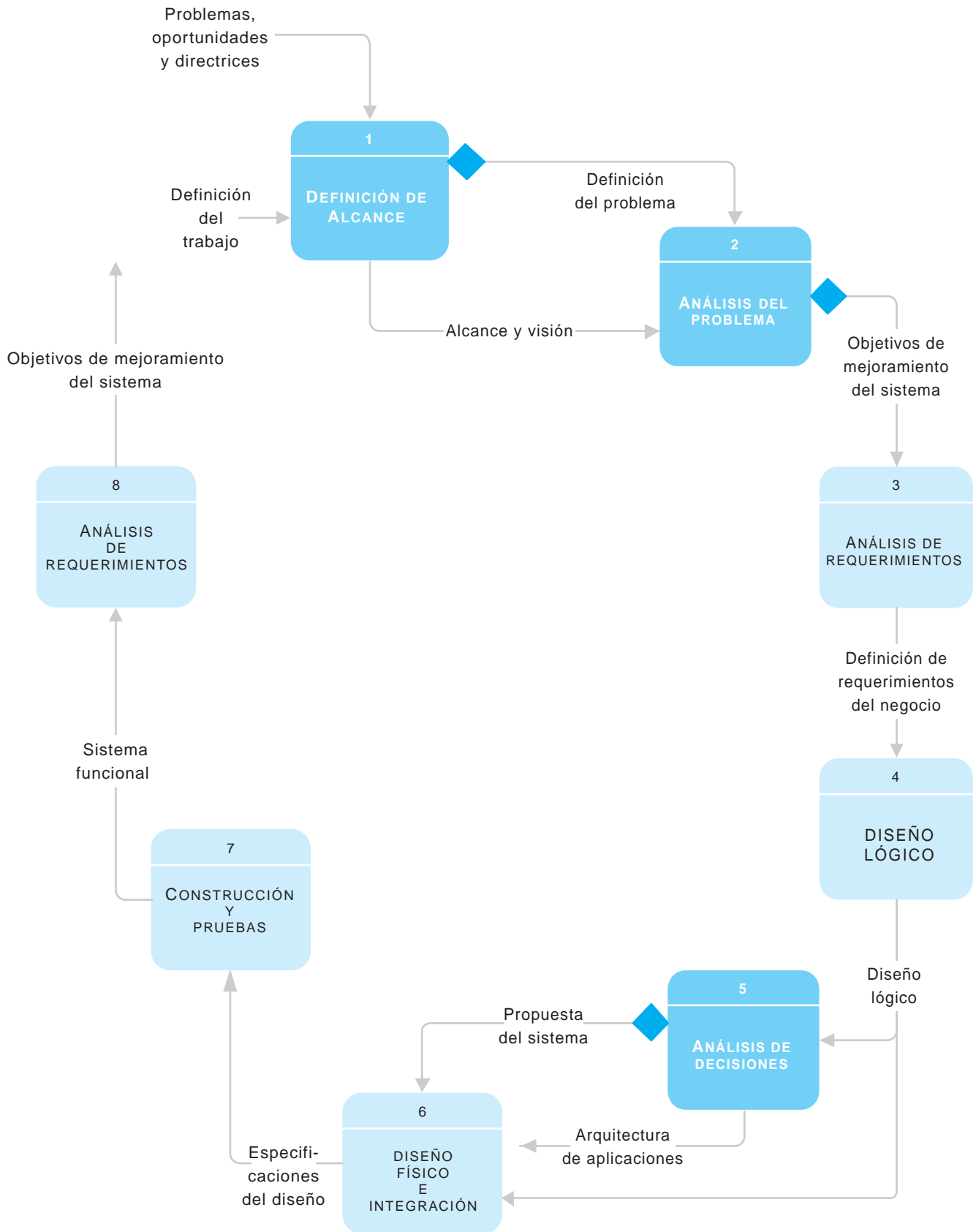
La factibilidad debe cuantificarse durante todo el ciclo de vida. En los primeros capítulos llamamos a esto una estrategia de *compromiso creciente* con la factibilidad. El alcance y la complejidad de un proyecto aparentemente factible pueden cambiar después del análisis completo de los problemas y oportunidades iniciales o de que se ha diseñado el sistema. Así, un proyecto que es factible en un momento dado puede dejar de serlo más adelante.

La figura 9.1 muestra los puntos de comprobación de factibilidad durante las fases de análisis de sistemas de nuestro ciclo de vida. Los puntos de comprobación se representan con rombos. Éstos indican que debe efectuarse una reevaluación de factibilidad y revisión administrativa al final de la fase previa (y antes de comenzar la que sigue). Un proyecto se podría cancelar o modificar en cualquier control, sin importar los recursos que se le hayan dedicado.

**factibilidad** Medición del grado en que un sistema de información será beneficioso o práctico para una organización.

**análisis de factibilidad** Proceso con el que se mide y evalúa la factibilidad.





**FIGURA 9.1** Puntos de comprobación de factibilidad durante el análisis de sistemas

La idea de cancelar un proyecto es algo que frecuentemente resulta difícil enfrentar. Una inclinación natural sería justificar la continuación del proyecto con base en el tiempo y dinero ya dedicados a él. Sin embargo, un principio administrativo fundamental es nunca dedicar dinero bueno al malo, sino reducir las pérdidas y pasar a un proyecto más factible. Decidirse por la cancelación no significa que sean insignificantes los costos ya erogados. Tarde o temprano deben recuperarse los costos para que se considere exitosa la inversión. Examinemos brevemente los puntos de comprobación de la figura 9.1.

### > Análisis de sistemas: control de la definición del alcance

El primer análisis de factibilidad se realiza durante la fase de definición del alcance. En esta etapa inicial del proyecto, la factibilidad pocas veces es más que una medición de la urgencia del problema y una primera estimación de los costos de desarrollo. Responde a la pregunta: ¿acaso los problemas (u oportunidades) justifican el costo de un estudio detallado y el análisis del sistema actual? Siendo realistas, la factibilidad no se puede medir con exactitud hasta que se entiendan mejor los problemas (y oportunidades) y requerimientos.

Luego de estimar los beneficios de resolver los problemas y las oportunidades, los analistas estiman los costos de desarrollar el sistema que se espera. Los analistas experimentados aumentan sistemáticamente esos costos en 50 a 100% (o más), ya que la experiencia les indica que los problemas pocas veces están bien definidos y que es frecuente que los requerimientos de los usuarios se subestimen.

### > Análisis de sistemas: control del análisis del problema

El siguiente control ocurre después de un estudio más detallado de análisis del problema del sistema actual. Puesto que se entienden más a fondo los problemas, el analista puede estimar con mayor aproximación los costos de desarrollo y los beneficios que se obtendrán de un nuevo sistema. El valor mínimo de la solución de un problema es igual al costo de ese problema. Por ejemplo, si los costos de administración de inventarios exceden en 35 000 dólares los límites aceptables, el valor mínimo de un sistema de información aceptable sería de 35 000 dólares. Se espera que el sistema mejorado genere resultados mejores que esos; pero al menos debe tener ese valor mínimo.

En este punto, los costos de desarrollo son todavía conjeturas. Los analistas aún deben definir plenamente los requerimientos de los usuarios o especificar una solución de diseño a esos requerimientos.

Si las estimaciones de costos aumentan significativamente entre la fase de investigación preliminar y la de análisis del problema, es probable que ello se relacione con el alcance. Este último tiende a incrementarse en numerosos proyectos. Si pone en riesgo la factibilidad podría ser necesario reducirlo.

### > Diseño de sistemas: control de análisis de decisiones

La fase de análisis de decisiones es una actividad importante del análisis de factibilidad, ya que en ella se muestra gráficamente una de las muchas opciones posibles como meta del diseño de sistemas.

Los problemas y requerimientos se deben conocer a estas alturas. Durante la fase de análisis de decisiones, se definen soluciones alternas en cuanto a métodos de entrada/salida y de almacenamiento de datos, necesidades de hardware y software de computadora, métodos de procesamiento e implicaciones de personal. La lista siguiente muestra una gama característica de opciones que puede evaluar el analista.

- No hacer nada. Dejar el sistema actual tal y como está. Sin importar la opinión de los administradores al respecto, esta opción se debe considerar y analizar como una inicial, contra la cual se pueden y deben valorar todas las demás.
- Empezar la reingeniería de los procesos de negocios (manuales), no la de los procesos computarizados. Esto podría consistir en agilizar las actividades, disminuir la duplicación y las tareas innecesarias, reorganizar la distribución de las oficinas y eliminar los formularios y procesos redundantes e innecesarios, entre otros cambios.
- Mejorar los procesos computarizados existentes.
- Comprar una aplicación de paquete.
- Diseñar y construir un nuevo sistema computarizado.

Después de definir estas opciones se analiza la factibilidad operativa, técnica, económica y de calendario. Luego se recomienda una alternativa a los propietarios del sistema para su aprobación y como base del diseño general y detallado.

## Seis pruebas de factibilidad

Hasta ahora se han definido la factibilidad y el análisis de factibilidad, además de identificar los controles de la factibilidad misma durante el análisis de sistemas. La factibilidad puede considerarse desde perspectivas diversas. Las siguientes son seis categorías de pruebas de factibilidad:

- La *factibilidad operativa* es una medición del grado en que una solución satisface los requerimientos de sistema identificados para resolver los problemas y aprovechar las oportunidades que se visualizan en relación con el sistema.
- La *factibilidad cultural (o política)* es una medición del sentir de las personas acerca de una solución y el grado de su aceptación en un ambiente organizacional dado.
- La *factibilidad técnica* es una medición de la naturaleza práctica de una solución técnica específica y la disponibilidad de recursos técnicos y experiencia para ponerla en práctica y mantenerla.
- La *factibilidad de calendario* es una medición de cuán razonable sea el calendario del proyecto.
- La *factibilidad económica* mide la efectividad en costos de un proyecto o solución.
- La *factibilidad legal* es una medición del grado en que se puede poner en práctica una solución dentro de las obligaciones legales y contractuales existentes.

En realidad, son pocos los sistemas que no son viables. En vez de ello, diversas opciones de solución tienden a ser más o menos factibles que otras. A continuación se analizan los cuatro criterios de factibilidad recién mencionados.

### > Factibilidad operativa

La **factibilidad operativa** es una medición del grado en que un sistema propuesto resuelve los problemas y aprovecha las oportunidades identificadas durante las fases de definición del alcance del análisis del problema, así como el grado en que satisface los requerimientos de sistema identificados en la fase de análisis de requerimientos. La factibilidad operativa también se plantea si, dado lo que se sabe acerca del problema y el costo de la solución, todavía vale la pena intentar la resolución del problema. El marco de referencia PIECES (capítulo 3) puede usarse como base para analizar la urgencia de un problema o la efectividad de una solución.

### > Factibilidad cultural (o política)

Se relaciona con la factibilidad operativa. Sin embargo, mientras que esta última versa sobre el grado en que la solución satisface los requerimientos del sistema, la **factibilidad cultural (o política)** es la forma en que los usuarios finales se sienten respecto del sistema propuesto. Podría decirse que con la factibilidad operativa se evalúa si un sistema *puede* funcionar o no, mientras que con la factibilidad cultural/política se plantea si el sistema *funcionará* en un ambiente organizacional dado.

En esta era de la información, el conocimiento es poder. Resulta común que un sistema de información modifique la estructura de cómo se canaliza y controla la información, lo que hasta cierto punto cambia la estructura de poder organizacional. Algunos usuarios y administradores pueden sentirse amenazados y luchar contra la puesta en práctica del sistema.

Debe tenerse en cuenta también que la cultura de las organizaciones es cada vez más multicultural. Es posible que se hayan fusionado empleados y divisiones de diversas compañías, con perspectivas muy distintas acerca de cómo se debe estructurar el trabajo y qué deben hacer y no hacer los sistemas de información. En el caso de las organizaciones internacionales, el sistema de información también debe recibir la aceptación en diversas culturas nacionales. Este tema se analiza con las preguntas siguientes:

- ¿Apoyan los directivos al sistema?
- ¿Cómo se sentirán los usuarios finales acerca de su función o papel en el nuevo sistema?

#### factibilidad operativa

Medición del grado en que una solución satisface los requerimientos del sistema identificados para resolver problemas y aprovechar las oportunidades que se visualizan para el sistema.

#### factibilidad cultural (o política)

Medición del grado en que la solución será aceptada en un ambiente organizacional dado.

- ¿Se resistirán los usuarios finales o administradores a usar el nuevo sistema? ¿Es posible superar este problema? En caso afirmativo, ¿cómo?
- ¿Cómo se modificará el ambiente de trabajo de los usuarios finales? ¿Se podrán o querrán adaptar los usuarios finales y administradores al cambio?

## > Factibilidad técnica

**factibilidad técnica** Medición del grado en que una solución técnica es práctica y de la disponibilidad de recursos técnicos y experiencia.

Hoy es poco lo que resulta técnicamente imposible. Por ende, con la **factibilidad técnica** se considera lo práctico y razonable. La factibilidad técnica responde a tres preguntas principales:

1. ¿Es práctica la tecnología o solución propuesta?
2. ¿Se cuenta actualmente con la tecnología necesaria?
3. ¿Se tiene la experiencia técnica requerida?

**¿Es práctica la tecnología o solución propuesta?** Lo normal es que esté disponible la tecnología para cualquier solución definida. La pregunta es si la tecnología tiene madurez suficiente para su fácil aplicación al problema en cuestión. Algunas empresas prefieren usar tecnología de punta, mientras que muchas otras optan por aplicar tecnología madura y comprobada. Esta última tiene una base de clientes más amplia para obtener asesoría acerca de problemas y mejoras.

**¿Se cuenta actualmente con la tecnología necesaria?** En el supuesto de que es práctica la tecnología que requiere la solución, hay que preguntarse: ¿está disponible en nuestra fuente de sistemas de información? Si lo está, el paso siguiente es plantearse si se tiene la capacidad necesaria. Por ejemplo, ¿se podrán manejar los informes y formularios que requiera el nuevo sistema con la impresora actual?

Si la respuesta a cualquiera de esas preguntas es no, lo siguiente es plantearse si se puede obtener la tecnología. Es factible que la tecnología sea práctica y esté disponible, además de ser necesaria. Pero podría ser simplemente que no se pueda costear por el momento. Aunque este argumento raya en lo concerniente a la factibilidad económica, en realidad es factibilidad técnica. Si no se puede costear la tecnología, ¿entonces la alternativa que requiere esa tecnología es impráctica y técnicamente inviable!

**¿Se tiene la experiencia técnica requerida?** Esta consideración de la factibilidad técnica se suele pasar por alto durante el análisis de factibilidad. Inclusive si una compañía cuenta con la tecnología, eso no significa que posea la capacidad necesaria para aplicarla correctamente. Por ejemplo, una compañía podría tener un Sistema de Administración de Bases de Datos (DBMS, por sus siglas en inglés). Sin embargo, los analistas y programadores disponibles para el proyecto podrían no conocer ese DBMS en grado suficiente para aplicarlo correctamente. Cierto, todos los profesionales de sistemas de la información pueden aprender nuevas tecnologías; pero la curva de aprendizaje tiene efecto negativo en la factibilidad técnica del proyecto, específicamente, en su calendario.

## > Factibilidad del calendario

**factibilidad del calendario** Medición del grado en que un calendario de proyecto es razonable.

Si se cuenta con la experiencia técnica, ¿son razonables las fechas límite del proyecto?; es decir, ¿cuál es la **factibilidad del calendario** del proyecto? Algunos proyectos se inician con fechas límite específicas. Es necesario determinar si tales fechas son imperativas o deseables. Por ejemplo, un proyecto de desarrollo de un sistema para satisfacer nuevos reglamentos gubernamentales de presentación de informes podría tener una fecha límite que coincide con la fecha en la que deban presentarse los nuevos informes. Las multas relacionadas con el no cumplimiento de esa fecha haría imperativo alcanzarla. Si las fechas límite son deseables, en lugar de imperativas, el analista puede proponer un calendario alternativo.

Es preferible (a menos que la fecha límite sea imperativa) entregar un sistema de información que funciona correctamente dos meses después que proporcionar a tiempo un sistema de información inútil y propenso a errores. Si bien no cumplir con las fechas límite puede generar problemas, el desarrollo de sistemas inadecuados puede ser desastroso. Es una elección entre el menor de dos males.

## > Factibilidad económica

Lo que en última instancia interesa en muchos proyectos es su **factibilidad económica**. Durante las fases iniciales del proyecto, el análisis de esta factibilidad equivale a poco más que juzgar si los posibles beneficios de solucionar el problema valen la pena o no. Resulta prácticamente imposible estimar costos en dicha etapa, ya que no se han identificado los requerimientos de los usuarios finales ni las soluciones técnicas alternas. Sin embargo, tan pronto se tengan identificados esos requerimientos y soluciones, el analista puede ponderar los costos y beneficios de cada alternativa, lo que se denomina análisis de costo-beneficio. Este último es tema de estudio más adelante en el capítulo.

## > Factibilidad legal

Los sistemas de información tienen efectos legales. En primer término, existen restricciones de copyright. En cualquier sistema que incluya componentes adquiridos, hay que tener la certeza de que no se transgredan los acuerdos de la licencia. Por ejemplo, ello significa instalar únicamente copias con licencia. Sin embargo, los acuerdos de licencia y la protección contra copias también pueden restringir la forma de integrar los datos y procesos con otras partes del sistema. Si se trabaja con programadores independientes por contrato, se tienen que definir anticipadamente la propiedad del código fuente del programa y los acuerdos de confidencialidad.

Los contratos colectivos de trabajo son otra posible restricción a los sistemas de información, por la forma en que se paga a los trabajadores y la manera de vigilar su trabajo. Deben satisfacerse los requerimientos legales de informes financieros. Los requerimientos de sistemas para compartir datos con socios comerciales podrían ir inclusive contra las leyes antimonopolios. Por último, hoy los sistemas de información son de alcance internacional. Algunos países estipulan dónde se deben guardar y procesar los datos sobre empleados locales y transacciones locales. Las naciones difieren en cuanto al número de horas que componen la semana laboral y el tiempo de descanso para el almuerzo o comida de los empleados.

## > Resultado final

Hasta este punto se ha analizado el hecho de que cualquier solución alternativa puede evaluarse según seis criterios de factibilidad: operativos, culturales/políticos, técnicos, de calendario, económicos y legales. ¿Cómo elige el analista la mejor solución? No es fácil hacerlo. Los aspectos operativos y económicos suelen estar en conflicto. Por ejemplo, la solución óptima desde el punto de vista operativo para los usuarios finales también puede ser la más costosa y, con ello, la de menor factibilidad económica. La decisión final se toma sólo después de reunirse con los usuarios finales, revisar los datos y elegir la mejor alternativa en general.

---

**factibilidad económica** Medición de la relación costo-beneficio de un proyecto o solución.

---

**factibilidad legal** Medición del grado en que una solución se puede implantar dentro del marco de las obligaciones legales y contractuales existentes.

## Técnicas de análisis de costo-beneficio

Se ha definido la factibilidad económica como un análisis de costo-beneficio. ¿Cómo se estiman los costos y beneficios? ¿Cómo se comparan esos costos y beneficios para determinar la factibilidad económica? Muchas facultades (universidades) ofrecen cursos completos sobre estos temas: cursos de administración financiera, análisis de decisiones financieras y análisis y economía de ingeniería. Este apartado presenta una panorámica de esas técnicas.

## > ¿Cuánto costará el sistema?

Los costos caen en dos categorías. Existen algunos relacionados con el desarrollo del sistema y otros correspondientes a la operación del sistema mismo. Los primeros pueden calcularse desde el comienzo del proyecto y se deben refinar al final de cada fase del mismo. Los segundos son estimables sólo después de haber definido algunas soluciones

específicas basadas en la computadora. A continuación se estudian más de cerca los costos de los sistemas de información.

Los costos de desarrollar un sistema de información pueden clasificarse según la fase en que ocurren. Los de desarrollo del sistema usualmente son costos de una sola vez, que no vuelven a ocurrir, sino hasta después de completar el proyecto. Muchas organizaciones tienen categorías estándar de costos que deben evaluarse. A falta de esas categorías, la lista siguiente debe ser útil:

- *Costos de personal.* Los sueldos de analistas de sistemas, programadores, consultores, personal de introducción de datos, operadores de computadoras, secretarías, etc., que trabajan en el proyecto, componen los costos de personal. Puesto que muchas de estas personas dedican tiempo a diversos proyectos, sus sueldos se deben prorratear para reflejar el tiempo dedicado a cada uno de ellos.
- *Uso de computadoras.* Es el tiempo de computadora usado en una o más de las actividades siguientes: programación, pruebas, conversión, procesamiento de texto, mantenimiento del diccionario del proyecto, elaboración de prototipos, cargado de nuevos archivos de datos, etc. Si un centro de cómputo realiza cargos por uso de los recursos de cómputo, como el almacenamiento en disco o la impresión de informes, debe estimarse su costo.
- *Capacitación.* Los cursos de capacitación pueden acompañarse de gastos si es necesario capacitar al personal de cómputo o usuarios finales. Los cursos de capacitación en paquete se pueden cargar sobre la base de una tarifa fija por sitio, una cuota por estudiante (por ejemplo, 395 dólares por estudiante) o una tarifa por hora (como 75 dólares por hora de clase).
- *Costos de insumos, duplicación y equipo.*
- *Costos de equipo y software de cómputo nuevos.*

Un ejemplo de costos de desarrollo de una solución típica se ilustra en la figura 9.2. Cuando los analistas estiman los costos de desarrollo es importante asignar partidas a la posibilidad de que se incurra en costos del sistema después de que entre en operación. Los beneficios durante su ciclo de vida deben hacer que se recuperen los costos de desarrollo y de operación. A diferencia de los costos de desarrollo del sistema, los de operación tienden a recurrir durante el ciclo de vida del propio sistema. Estos costos de operación pueden clasificarse como fijos o variables.

Los **costos fijos** ocurren a intervalos regulares y con una tasa relativamente fija. Sus ejemplos abarcan:

- Pagos de arrendamientos y de licencias de software.
- Sueldos prorrateados de operadores de sistemas de información y personal de soporte (si bien los sueldos tienden a aumentar, su incremento es gradual y no suele cambiar radicalmente de un mes a otro).

Los **costos variables** ocurren en proporción a algún factor de uso. Sus ejemplos abarcan:

- Costos de uso de computadoras (por ejemplo, tiempo de uso de CPU, equipo de uso de conexión de terminales, uso de espacio de almacenamiento), que pueden variar con la carga de trabajo.
- Insumos (por ejemplo, formularios impresos, papel de impresora, tarjetas perforadas, disquetes, cintas magnéticas y otros), que también varían con la carga de trabajo.
- Costos generales y de administración prorrateados (como los de servicios públicos, mantenimiento y teléfono), que pueden asignarse durante la vida útil del sistema con técnicas estándar de contabilidad de costos.

Ejemplos de estimaciones de costos de operación de una solución se ilustran en la figura 9.2.

## > ¿Cuáles beneficios proporcionará el sistema?

Los beneficios normalmente aumentan las utilidades o disminuyen los costos, ambas características muy deseables en un nuevo sistema de información. En la medida posible, los beneficios se deben cuantificar en términos monetarios, además de clasificarlos como tangibles o intangibles.

Los **beneficios tangibles** son los de fácil cuantificación. Suelen medirse como ahorros mensuales o anuales, o como utilidades para la empresa. Por ejemplo, considere el contexto siguiente:

---

**costo fijo** Costo que ocurre a intervalos regulares y con tasa relativamente fija o estable.

---

**costo variable** Costo que ocurre en proporción a algún factor de uso.

---

**beneficio tangible** Un beneficio que puede ser fácilmente cuantificado.

### Costos estimados para la alternativa del sistema cliente-servidor

#### COSTOS DE DESARROLLO

##### Personal:

2	Analistas de sistemas (400 horas a \$50.00/h)	\$40 000
4	Programadores/Analistas (250 horas a \$35.00/h)	\$35 000
1	Diseñador de GUI (200 horas a \$40.00/h)	\$8 000
1	Especialista en telecomunicaciones (50 horas a \$50.00/h)	\$2 500
1	Arquitecto de sistemas (100 horas a \$50.00/h)	\$5 000
1	Especialista en bases de datos (15 horas a \$45.00/h)	\$675
1	Bibliotecario de sistemas (250 horas a \$15.00/h)	\$3 750

##### Gastos:

4	Registros de capacitación en Smalltalk (\$3 500.00/estudiante)	\$14 000
---	--	----------

##### Hardware y software nuevos:

1	Servidor de desarrollo	\$18 700
1	Software de servidor (sistema operativo y otros)	\$1 500
1	Software de servidor de DBMS	\$7 500
7	Software de cliente de DBMS (\$950.00 por cliente)	\$6 650

##### Costos de desarrollo totales:

\$143 275
-----------

#### COSTOS OPERATIVOS ANUALES PROYECTADOS

##### Personal:

2	Programadores/Analistas (125 horas a \$35.00/h)	\$8 750
1	Bibliotecario de sistemas (20 horas a \$15.00/h)	\$300

##### Gastos:

1	Contrato de mantenimiento de servidor	\$995
1	Contrato de mantenimiento de software de DBMS de servidor	\$525
	Formularios preimpresos (15 000/años a unos \$0.22/formulario)	\$3 300

##### Costos anuales proyectados totales:

\$13 870
----------

**FIGURA 9.2** Costos de una solución de sistemas propuesta

Mientras se procesan solicitudes de vivienda estudiantil, se descubre una redundancia considerable de los datos escritos y archivados. El análisis revela que los mismos datos se escriben siete veces, lo que requiere un promedio de 44 minutos adicionales de trabajo de oficina por cada solicitud. La oficina procesa 1 500 solicitudes anuales. Eso significa un total de 66 000 minutos o 1 100 horas anuales de trabajo redundante. Si el sueldo de una secretaria es de 15 dólares por hora, el costo de este problema y el beneficio de solucionarlo es de 16 500 dólares por año.

En forma alterna, los beneficios tangibles podrían medirse como ahorros en costos unitarios o utilidades. Por ejemplo, un esquema alternativo de evaluación del inventario podría reducir el costo de administración de existencias en 0.32 dólares por unidad. Algunos ejemplos de beneficios tangibles se enumeran al margen de esta página.

Otros son **beneficios intangibles**, es decir, los que se considera difícil o imposible de cuantificar. Salvo que estos beneficios por lo menos se identifiquen, es del todo posible que muchos proyectos no sean factibles. Los ejemplos de beneficios intangibles se muestran en la lista al margen de la página siguiente.

#### BENEFICIOS TANGIBLES

- Menos errores de procesamiento
- Aumento de la producción
- Disminución del tiempo de respuesta
- Eliminación de pasos de tareas
- Aumento de las ventas
- Reducción de pérdidas de créditos
- Disminución de gastos

**BENEFICIOS INTANGIBLES**

Mejoramiento del crédito mercantil

Mejoramiento del ánimo de los empleados

Mejor servicio a la comunidad

Mejor toma de decisiones

**beneficio intangible** Un beneficio que se considera difícil o imposible de cuantificar.

Desafortunadamente, si un beneficio no puede ser cuantificado, es difícil aceptar la validez del análisis de costo-beneficio correspondiente, que se basa en datos incompletos. Ciertos analistas ponen en tela de juicio la existencia de beneficios intangibles. Afirman que todos los beneficios son cuantificables, si bien algunos simplemente son más difíciles de cuantificar. Por ejemplo, suponga que se señala el mejoramiento del crédito mercantil como un posible beneficio intangible. ¿Es posible cuantificarlo? Podría intentar el análisis siguiente:

1. ¿Cuál es el resultado del rechazo de un cliente? El cliente coloca menos pedidos o ninguno en absoluto.
2. ¿Hasta qué grado se reducen los pedidos de cliente? Aunque a un usuario puede serle difícil cuantificar de manera específica este efecto, podría intentar que el usuario final calcule las posibilidades (o podría inventar una estimación a la que el usuario final reaccione). Por ejemplo:
  - a) Existen probabilidades de 50% (0.50) de que un cliente habitual coloque algunos pedidos —menos de 10% del total de sus pedidos— con competidores para poner a prueba su funcionamiento.
  - b) Existen probabilidades de 20% (0.20) de que el cliente habitual coloque hasta la mitad de sus pedidos (0.50) con competidores, en particular si el surtido de sus pedidos ha sido históricamente lento.
  - c) Existe un 10% (0.10) de probabilidades de que un cliente habitual coloque un pedido con nosotros sólo como último recurso. Eso reduciría nuestro negocio normal con el cliente a 10% de su volumen actual (pérdida de 90% o 0.90).
  - d) Existe un 5% (0.05) de probabilidades de que un cliente habitual decida no colocar con nosotros ningún pedido en absoluto (pérdida de 100% o 1.00).
3. La pérdida de negocio estimada se podría calcular como sigue:

$$\begin{aligned}
 \text{Pérdida} &= 0.50 \times (\text{pérdida de 0.10 del negocio}) \\
 &\quad + 0.20 \times (\text{pérdida de 0.50 del negocio}) \\
 &\quad + 0.10 \times (\text{pérdida de 0.90 del negocio}) \\
 &\quad + 0.50 \times (\text{pérdida de 1.00 del negocio}) \\
 &= 0.29 \\
 &= 29\% \text{ de estimación estadística de pérdida de negocio}
 \end{aligned}$$

4. Si un cliente en promedio compra 40 000 dólares por año, cabe esperar la pérdida de 29% u 11 600 dólares de esas compras. Si se tienen 500 clientes podría esperarse que la pérdida total sea de 5 800 000 dólares.
5. Presente este análisis a los administradores y úselo como punto de partida para cuantificar el beneficio.

### > ¿Es rentable el sistema propuesto?

Existen tres técnicas muy usadas para evaluar la factibilidad económica, también llamada *rentabilidad*: análisis de recuperación de la inversión, rendimiento sobre la inversión y el valor presente neto.

En la elección de una u otra técnica debe considerarse el auditorio que las usará. Casi todos los administradores que estudian en facultades de administración están familiarizados con las tres técnicas. Un concepto que se aplica a las tres es el ajuste de costos y beneficios para reflejar el valor del dinero en el tiempo.

**Valor del dinero en el tiempo** Un concepto que comparten las tres técnicas es el **valor del dinero en el tiempo**, es decir, el hecho de que una unidad monetaria vale hoy más que dentro de un año. Se le podría invertir ahora y, gracias a los intereses acumulados, se tendría más dinero al cabo de un año. Así pues, cualquier persona preferiría tenerla hoy, no luego de un año. Es por eso que los acreedores quieren que se les paguen con prontitud sus facturas, no pueden invertir el dinero que no tienen. El mismo principio puede aplicarse a los costos y beneficios *antes* que realizar el análisis de costo-beneficio.



Algunos costos de un sistema se acumulan después de ponerlo en práctica. Además, todos los beneficios del nuevo sistema se acumulan en el futuro. Antes de emprender el análisis de costo-beneficio, estos costos deben llevarse a unidades monetarias actuales. Un ejemplo debe aclarar este concepto.

Suponga que obtendremos un beneficio de 20 000 dólares al cabo de dos años. ¿Cuál es el valor presente de ese beneficio? Si el rendimiento actual sobre inversiones es cercano a 10%, una inversión de 16 528 dólares permitiría tener 20 000 dólares en dos años (más adelante se muestra el cálculo respectivo). Por lo tanto, el valor presente del beneficio estimado es de 16 528 dólares, es decir que preferiríamos tener 16 528 dólares hoy en lugar de la promesa de recibir 20 000 dólares luego de dos años.

Es frecuente que se comparen entre sí proyectos que tienen ciclos de vida diferentes, de modo que las técnicas de análisis del valor del dinero en el tiempo se han convertido en los métodos de análisis de costo-beneficio que prefieren muchos administradores. Al ajustar los costos y beneficios en relación con el tiempo, es posible mejorar las técnicas de análisis de costo-beneficio siguientes.

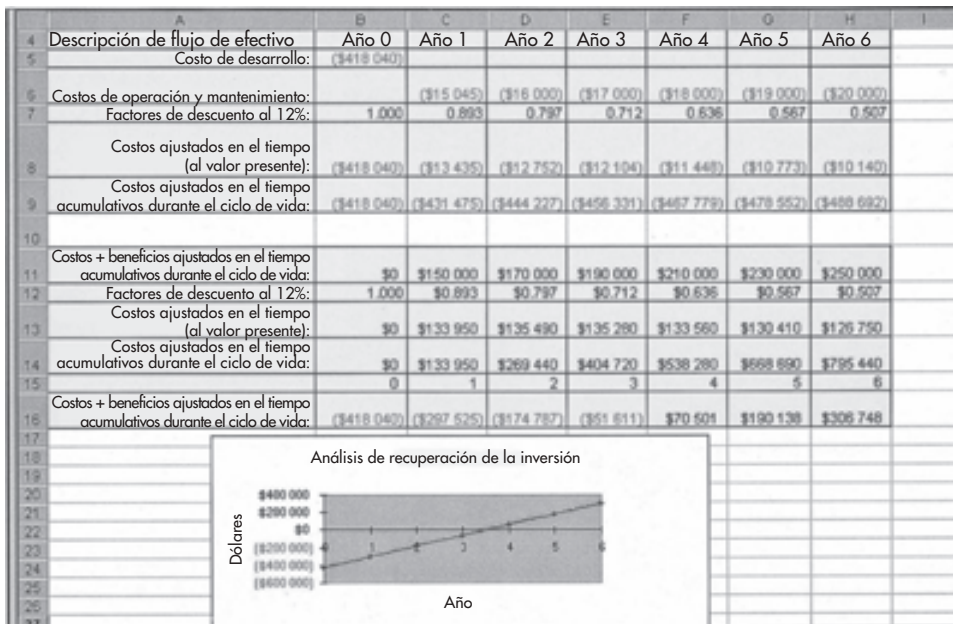
**Análisis de la recuperación de la inversión** El análisis de la recuperación de la inversión es un método sencillo y muy usado para determinar cuándo una inversión genera beneficios que permiten recuperarla. Se incurre en costos de desarrollo de sistemas mucho antes de que empiecen a acumularse los beneficios, de modo que se requiere algún tiempo para que estos últimos superen a los costos. Después de su puesta en práctica, se incurre en gastos de operación adicionales, que también deben recuperarse. El análisis de recuperación de la inversión determina cuánto tiempo debe transcurrir antes de que los beneficios acumulados sean mayores que los costos acumulados y corrientes. Este lapso se llama **periodo de recuperación de la inversión**.

En la figura 9.3 se observa un sistema de información que se desarrollará con costo de 418 040 dólares. Los costos de operación métodos estimados de cada uno de los seis años siguientes también se incluyen en la tabla. Asimismo, se muestran los beneficios netos estimados durante los mismos seis años de operación. ¿Cuál es el periodo de recuperación de la inversión?

Primero, es necesario ajustar los costos y beneficios en relación con el valor del dinero en el tiempo (es decir, ajustarlos a unidades monetarias actuales). Eso se hace como sigue: el valor presente de una unidad monetaria en el año  $n$  depende de lo que suele llamarse **tasa de descuento**. Esta última es un porcentaje similar a las tasas de intereses que se obtienen sobre una cuenta bancaria. En muchos casos, la tasa de descuento para una empresa es el **costo de oportunidad** de la posibilidad de invertir el dinero en otros proyectos, lo que abarca inclusive las inversiones en el mercado bursátil, fondos de mercado de dinero, bonos, etc. En forma alterna, una tasa de descuento podría representar lo

**análisis de la recuperación de la inversión** Técnica para determinar si una inversión se recuperará y cuándo lo hará.

**periodo de recuperación de la inversión** Periodo que transcurre antes de que los beneficios acumulados superen a los costos acumulados.



**FIGURA 9.3**  
Análisis de la recuperación de la inversión de un proyecto

**FIGURA 9.4** Tabla parcial del valor presente de un dólar

Periodos	8%	9%	10%	11%	12%	13%	14%
1	0.926	0.917	0.909	0.901	0.893	0.885	0.877
2	0.857	0.842	0.826	0.812	0.797	0.783	0.769
3	0.794	0.772	0.751	0.731	0.712	0.693	0.675
4	0.735	0.708	0.683	0.659	0.636	0.613	0.592
5	0.681	0.650	0.621	0.593	0.567	0.543	0.519
6	0.630	0.596	0.564	0.535	0.507	0.480	0.456
7	0.583	0.547	0.513	0.482	0.452	0.425	0.400
8	0.540	0.502	0.467	0.434	0.404	0.376	0.351

que la compañía considere un rendimiento aceptable sobre sus inversiones, dato que se podría indagar con el director financiero o contralor.

Suponga que la tasa de descuento de la compañía de ejemplo es de 12%. El valor en el presente, o **valor presente**, de una unidad monetaria en cualquier momento futuro se puede calcular con la fórmula siguiente:

$$VA_n = 1/(1 + i)^n$$

donde  $VA_n$  es el valor presente de 1 dólar dentro de  $n$  años e  $i$  es la tasa de descuento. Así pues, el valor presente de una unidad monetaria al cabo de dos años es:

$$VA_2 = 1/(1 + 0.12)^2 = 0.797$$

Mencionamos antes que una unidad monetaria actual vale más que una que se recibirá dentro de un año. Sin embargo, pareciera que su valor es menor. Se trata de una ilusión. El valor presente se interpreta como sigue. Tener \$0.797 hoy es mejor que tenerlos dentro de dos años. ¿Cuánto mejor? Exactamente \$0.203, ya que los \$0.797 se convertirán en 1 dólar en dos años (sobre la base de una tasa de descuento de 12%).

A fin de determinar el valor presente de cualquier costo o beneficio en el año 2, basta multiplicar 0.797 por el costo o beneficio estimado. Por ejemplo, el gasto de operación estimado en el año 2 es de 16 000 dólares. El valor presente de este gasto es  $16\,000 \times 0.797$ , es decir, 12 752 dólares (redondeado). Por fortuna, es innecesario calcular los factores de descuento. Existen tablas similares a la tabla parcial de la figura 9.4, en las que se muestra el valor presente de una unidad monetaria con periodos y tasas de descuento distintas. Basta multiplicar este número por el costo o beneficio estimado para obtener el valor presente de este último. Versiones más detalladas de esa tabla forman parte de libros de contabilidad y finanzas, así como de las funciones de programas de hojas de cálculo.

Mejor aún: muchos programas de hojas de cálculo incluyen funciones para el cálculo del valor presente de cualquier flujo de efectivo, trátase de un costo o de un beneficio. Todos los ejemplos de este modelo se realizaron con Microsoft Excel. Algunas tablas se pueden preparar con *Lotus 1-2-3*. La ventaja de las hojas de cálculo es que una vez configuradas las filas, columnas y funciones basta introducir los costos y beneficios, y la hoja de cálculo descuenta las cifras a su valor presente. (De hecho, también se puede programar una hoja de cálculo para que lleve a cabo el análisis de costo-beneficio.)

En la figura 9.3, advierta que se muestran todos los costos y beneficios del ejemplo en su valor presente. Observe también que la tasa de descuento del año 0 es 1.000. ¿Por qué? El valor presente de una unidad monetaria en el año 0 es exactamente 1 dólar. En otras palabras, si se tiene esa unidad monetaria hoy, su valor es exactamente 1 dólar.

Una vez descontados los costos y beneficios, es posible completar el análisis de recuperación de la inversión. Observe los costos y beneficios acumulativos durante el ciclo de vida. Los costos aumentan gradualmente a lo largo del periodo de seis años, conforme se incurre en costos de operación. Sin embargo, advierta también que los beneficios se acumulan con un ritmo mucho mayor.

**valor presente** Valor que tiene actualmente una unidad monetaria que se recibirá en cualquier momento del futuro.

A lo largo del ciclo de vida, superarán a los costos entre los años 3 y 4. La representación gráfica de los costos y beneficios acumulativos durante el ciclo de vida, ajustados en relación con el tiempo, permite estimar que el punto de equilibrio (cuando costos + beneficios = 0) ocurrirá aproximadamente 3.5 años después de que el sistema empiece a funcionar.

¿Es este sistema de información una buena o mala inversión? Ello depende de varios factores. Muchas compañías cuentan con directrices sobre el periodo de recuperación de todas sus inversiones. A falta de ellas, es necesario determinar una razonable para hacerlo con el periodo de recuperación. Suponga que la directriz indica que todas las inversiones deben tener un periodo de recuperación de cuatro años o menos. Puesto que en el ejemplo es de 3.5 años, la inversión es buena. Si fuera mayor de cuatro años, sería una mala inversión.

Advierta que también puede realizar el análisis de recuperación de la inversión sin ajustar los costos y beneficios en relación con el tiempo. No obstante, el resultado sería un periodo de 2.8 años, aparentemente más atractivo que el de 3.5 años calculado. Así pues, el periodo de recuperación no ajustado en el tiempo tiende a ser excesivamente optimista y engañoso.

**Análisis del rendimiento sobre la inversión** El **análisis del rendimiento sobre la inversión (ROI, por sus siglas en inglés)** es una técnica en la que se compara la rentabilidad durante el ciclo de vida de soluciones o proyectos alternos. El ROI de una solución o proyecto es una tasa porcentual que mide la relación entre lo que una empresa obtiene de una inversión y el monto invertido. El ROI durante el ciclo de vida de un proyecto o solución potencial se calcula como sigue:

$$\text{ROI durante el ciclo de vida} = (\text{beneficios estimados sobre el ciclo de vida} - \text{costos estimados durante el ciclo de vida}) / \text{costos estimados durante el ciclo de vida}$$

Calculemos el ROI durante el ciclo de vida de la misma solución de sistemas usada en el análisis del periodo de recuperación. Una vez más, deben ajustarse todos los costos y beneficios en el tiempo, con respecto al periodo de seis años. Los costos y beneficios ajustados en el tiempo se presentan en las filas 9 y 16 de la figura 9.3. La resta de los beneficios menos los costos, ambos estimados en relación con el ciclo de vida, es:

$$\$795\,440 - \$488\,692 = \$306\,748$$

Por tanto, el ROI durante el ciclo de vida es:

$$\text{ROI durante el ciclo de vida} = \$306\,748 / \$488\,692 = 0.628 = 63\%$$

Este ROI corresponde al ciclo de vida, *no* a un año. Una sencilla división entre la vida estimada del sistema (63/6) indica un ROI promedio de 10.5% anual. Esta solución se puede comparar con otras alternas. La que tenga el ROI más alto es la mejor. Sin embargo, al igual que en el periodo de recuperación de la inversión, la empresa puede establecer un ROI mínimo aceptable para todas sus inversiones. Si ninguna de las soluciones alternas llega a ese mínimo aceptable o lo excede, entonces ninguna de las alternativas es económicamente factible. Una vez más, las hojas de cálculo simplifican mucho el análisis del ROI, gracias a sus funciones de análisis financiero integradas.

Al igual que en el análisis del periodo de recuperación, se podría haber calculado el ROI sin el ajuste en el tiempo de los costos y beneficios. Sin embargo, ello habría generado un ROI desorientador de 129.4% durante el ciclo de vida o 21.6% anual. Por consiguiente, de nuevo se recomienda el ajuste en relación con el tiempo de todos los costos y beneficios a unidades monetarias actuales.

**Valor presente neto** Muchos administradores consideran que el **valor presente neto** de una alternativa de inversión es la técnica de análisis de costo-beneficio preferible, en particular los que tienen estudios de administración amplios. Una vez más, se determinan inicialmente los costos y beneficios en cada año de la vida del sistema. Y una vez más, es necesario ajustar todos los costos y beneficios a su valor monetario actual.

La figura 9.5 muestra la técnica del valor presente neto. Los costos se representan con flujos de efectivo negativos, y los beneficios, con flujos de efectivo positivos. Todos los costos y beneficios del ejemplo se llevaron a su valor presente. Advierta de nuevo que la tasa de descuento del año 0 (usado para acumular todos los costos de desarrollo) es de 1.000, ya que el valor presente de una unidad monetaria en el año 0 es exactamente 1 dólar.

### análisis del rendimiento sobre la inversión (ROI)

Técnica en la que se compara la rentabilidad de soluciones alternas durante el ciclo de vida del sistema.

**valor presente neto** Técnica de análisis que compara los costos y beneficios anuales descontados de diferentes soluciones alternativas.

**FIGURA 9.5**

Análisis de valor presente neto de un proyecto

Análisis de valor presente neto de la alternativa del sistema cliente-servidor							
(Importes redondeados al \$1 más cercano)							
Descripción de flujo de efectivo	Año 0	Año 1	Año 2	Año 3	Año 4	Año 5	Año 6
Costos de desarrollo:							
	(\$418 040)						
Costos de operación y mantenimiento:							
		(\$15 045)	(\$16 000)	(\$17 000)	(\$18 000)	(\$19 000)	(\$20 000)
Factores de descuento al 12%:	1.000	0.893	0.797	0.712	0.636	0.567	0.507
Valor presente del costo anual:	(\$418 040)	(\$13 435)	(\$12 752)	(\$12 104)	(\$11 449)	(\$10 773)	(\$10 140)
Valor presente total de costos durante el ciclo de vida:							(\$436 692)
Beneficios derivados de la operación del nuevo sistema							
	\$0	\$150 000	\$170 000	\$190 000	\$210 000	\$230 000	\$250 000
Factores de descuento al 12%:	1.000	\$0.893	\$0.797	\$0.712	\$0.636	\$0.567	\$0.507
Valor presente de los beneficios anuales:	\$0	\$133 950	\$135 490	\$135 280	\$133 560	\$130 410	\$126 750
Valor presente total de los beneficios durante el ciclo de vida:							\$795 440
VALOR PRESENTE NETO DE ESTA ALTERNATIVA:							\$306 748

Después de descontar todos los costos y beneficios, se resta la suma de los costos descontados de la suma de los beneficios descontados para determinar el valor presente neto. Si el resultado es positivo, la inversión es buena, y si es negativo, mala. Al comparar soluciones o proyectos múltiples, el mejor es el que tenga el valor presente neto positivo más alto. (¡Esto funciona inclusive cuando las alternativas tienen ciclos de vida distintos!) En el ejemplo, la solución evaluada tiene un valor presente neto de 306 748 dólares. Ello significa que invertir dicha cantidad al 12% durante seis años permite obtener la misma utilidad que si se pone en práctica la solución de sistemas de información. Se trata de una buena inversión, siempre y cuando ninguna otra alternativa tenga un valor presente neto mayor de 306 748 dólares.

Una vez más, las hojas de cálculo pueden simplificar en gran medida el análisis de valor neto presente mediante sus funciones incorporadas de análisis financiero.

## Análisis de factibilidad de las soluciones alternativas del sistema

Durante la fase de análisis de decisiones del análisis de sistemas, el analista de sistemas identifica las soluciones alternativas del sistema y luego las analiza en cuanto a su factibilidad. En este capítulo se estudian los criterios y técnicas de análisis correspondientes. En la presente sección se evalúa un par de técnicas de documentación que pueden mejorar considerablemente la comparación de soluciones alternativas del sistema. En ambas se usa un formato matricial. Los autores han observado que estas matrices son útiles para presentar las alternativas y recomendaciones a los administradores.

### > Matriz de soluciones alternativas del sistema

La primera matriz permite comparar soluciones alternativas del sistema en base a diversas características. La **matriz de soluciones alternativas del sistema** sirve para documentar las similitudes y diferencias entre dichos sistemas, si bien no brinda análisis alguno.

Las columnas de la matriz representan soluciones alternativas. Los analistas experimentados siempre consideran varias opciones de puesta en práctica. Al menos una de esas opciones debe ser el sistema existente, ya que sirve como base para la comparación de alternativas.

Las filas de la matriz representan las características que diferencian a las soluciones alternativas. Para los fines de este libro, algunas de las características se basan en los componentes de los sistemas de información. El desglose es como sigue:

- **Interesados.** Identificar la manera en que el sistema interactúa con las personas y con otros sistemas.
- **Conocimiento.** Identificar el modo en que se implantan los almacenes de datos (por ejemplo, archivos convencionales, bases de datos relacionales y otras estructuras de

**matriz de soluciones alternativas del sistema** Herramienta usada para documentar similitudes y diferencias entre soluciones alternativas del sistema.

**FIGURA 9.6** Plantilla de la matriz de soluciones alternativas del sistema

	Nombre de la alternativa 1	Nombre de la alternativa 2	Nombre de la alternativa 3
Interesados			
Conocimiento			
Procesos			
Comunicaciones			

bases de datos), cómo se capturan las entradas (por ejemplo, en línea, por lotes, etc.) y la manera en que se generan las salidas (por ejemplo, conforme a un horario, por demanda, impresas, en pantalla, etc.).

- *Procesos.* Identificar cómo se modifican los procesos de negocios (manualmente) y cómo se implantan los procesos de cómputo. En relación con los segundos, se tienen numerosas opciones, entre ellas en línea o por lote, o bien con paquetería o software creado en la empresa.
- *Comunicaciones.* Identificar el modo en que se distribuyen los procesos y datos. Una vez más podrían considerarse varias alternativas, por ejemplo, las soluciones centralizada, descentralizada, distribuida (duplicada) o cooperativa (cliente/servidor). Las estrategias y tipos de distribución de red son temas del capítulo 11.

En las celdas de la matriz se documentan las características que ayudan a que los lectores entiendan las diferencias entre las opciones. La estructura básica de la matriz se ilustra en la figura 9.6.

Antes de considerar cualquier posible solución, deben tenerse en cuenta las restricciones a que estén sujetas las soluciones. Dichas limitantes asumen la forma de decisiones arquitectónicas cuyo fin es lograr orden y constancia en las aplicaciones. Por ejemplo, una arquitectura de tecnología podría restringir las soluciones a bases de datos relacionales o a redes cliente/servidor.

Son varias las estrategias de identificación de soluciones alternativas, entre ellas:

- *Reconocer las ideas y opiniones de los usuarios.* Los usuarios suelen recomendar soluciones manuales o relacionadas con la tecnología a lo largo de un proyecto de sistemas. Se las debe considerar.
- *Consulta de las normas metodológicas y arquitectónicas.* Las normas metodológicas y arquitectónicas de desarrollo de muchas organizaciones suelen regir la manera en que se seleccionan las soluciones tecnológicas, así como la tecnología o tecnologías que se incluyen.
- *Lluvia de ideas de posibles soluciones.* La lluvia de ideas (*brainstorming*) es una técnica efectiva para identificar posibles soluciones. Resulta especialmente efectiva cuando se utiliza una estrategia o marco de referencia organizados, como los componentes de IS u otras características de IS. La lluvia de ideas debe abarcar soluciones que representen la compra, construcción y combinaciones de compra y construcción.
- *Búsqueda de referencias.* El analista debe solicitar ideas y opiniones de otras personas y organizaciones que han implantado sistemas similares.
- *Lectura de revistas y otras publicaciones periódicas apropiadas.* Este tipo de literatura puede incluir anuncios y artículos concernientes a estrategias de automatización, sus éxitos, fracasos y tecnologías.

Los miembros del equipo de desarrollo podrían usar una combinación de las estrategias antes detalladas para obtener varias posibles soluciones alternativas del sistema.

La figura 9.7 contiene una matriz de ejemplo parcialmente escrita con soluciones alternativas del sistema, en la que se enumeran tres de las cinco opciones. En ella, la matriz

**FIGURA 9.7** Ejemplo de matriz de soluciones alternativas del sistema

Características	Alternativa 1	Alternativa 2	Alternativa 3	Alternativa ...
<p><b>Porción computarizada del sistema</b></p> <p>Breve descripción de la porción del sistema que sería computarizada (automatizada) en esta alternativa.</p>	Se compraría el paquete COTS Platinum Plus de Entertainment Software Solutions y se adaptaría para satisfacer la funcionalidad que requiere servicios a miembros.	Operaciones de servicios a miembros y almacén relacionadas con el surtido de pedidos.	Lo mismo que la alternativa 2.	
<p><b>Beneficios</b></p> <p>Breve descripción de los beneficios del negocio que se obtendrían con esta alternativa.</p>	Esta solución se puede implantar rápidamente por ser adquirida.	Brinda soporte completo a los procesos del negocio que requieren los usuarios de SoundStage, Inc. Además, interacción más eficiente con las cuentas de los miembros.	Lo mismo que la alternativa 2.	
<p><b>Servidores y estaciones de trabajo</b></p> <p>Descripción de los servidores y estaciones de trabajo necesarios para esta alternativa.</p>	En lo técnico, la arquitectura requiere servidores y estaciones de trabajo (clientes) Pentium III, MS Windows 2000.	Lo mismo que la alternativa 1.	Lo mismo que la alternativa 1.	
<p><b>Herramientas de software necesarias</b></p> <p>Herramientas de software que se requerirían para el diseño y construcción de la solución alternativa del sistema (p. ej., sistema de administración de bases de datos, emuladores, sistemas operativos, lenguajes). En general, es inaplicable si se compran paquetes de software de aplicaciones.</p>	MS Visual C++ y MS Access para la personalización del paquete con el fin de proporcionar integración y redacción de informes.	MS Visual Basic 5.0 System Architect 2001 Internet Explorer.	MS Visual Basic 5.0 System Architect 2001 Internet Explorer.	
<p><b>Software de aplicaciones</b></p> <p>Descripción del software que se comprará, construirá, accederá, o una combinación de estas técnicas.</p>	Solución de software de paquete.	Solución personalizada.	Lo mismo que la alternativa 2.	
<p><b>Método de procesamiento de datos</b></p> <p>En general, alguna combinación de procesamiento en línea, por lotes, por lotes diferido, por lotes remoto y en tiempo real.</p>	Cliente/Servidor.	Lo mismo que la alternativa 1.	Lo mismo que la alternativa 1.	
<p><b>Dispositivos de salida e implicaciones</b></p> <p>Descripción de los métodos de salida que se usarán, requerimientos de salida especiales (como redes, formularios preimpresos, etc.) y consideraciones de salida (como las restricciones de tiempo).</p>	2 impresoras láser departamentales HP4MV 2 impresoras láser en LAN HP5SI.	2 impresoras láser departamentales HP4MV 2 impresoras láser en LAN HP5SI 1 impresora de código de barras PRINTRONIX (incluye software y controladores) Las páginas web deben diseñarse para resolución VGA. Todas las pantallas internas se diseñarán con resolución SVGA.	Lo mismo que la alternativa 2.	
<p><b>Dispositivos de entrada e implicaciones</b></p> <p>Descripción de los métodos de entrada que se usarán, dispositivos de entrada (como el teclado o ratón), requerimientos de entrada especiales (como formularios nuevos o modificados en los que se introducirán los datos) y consideraciones de entrada (como el horario de las entradas reales).</p>	Teclado y ratón.	Cámara digital y software "Quick Take" de Apple 15 escáneres de código de barras láser PSC Quickscan 1 escáner de cama plana HP Scanjet 4C Teclado y ratón	Lo mismo que la alternativa 2.	
<p><b>Dispositivos de almacenamiento e implicaciones</b></p> <p>Descripciones breves de los datos que se almacenarán, los que se recuperarán de almacenes de datos existentes, los medios de almacenamiento que se usarán, la capacidad de almacenamiento necesaria y la forma de organizar los datos.</p>	MS SQL Server DBMS con un arreglo de discos de 100 Gb de capacidad.	Lo mismo que la alternativa 1.	Lo mismo que la alternativa 1.	

**FIGURA 9.8** Plantilla de matriz de análisis de factibilidad de soluciones

	Ponderación	Alternativa 1	Alternativa 2	Alternativa 3
Descripción				
Factibilidad operativa				
Factibilidad cultural				
Factibilidad técnica				
Factibilidad económica				
Factibilidad de calendario				
Factibilidad legal				
Calificación ponderada				

sirve para tener una panorámica de las características relativas a la porción del sistema que se computarizará (automatizará), los beneficios de negocios y las herramientas o aplicaciones de software necesarias. En las páginas siguientes se brindan detalles adicionales sobre otras características, como las mencionadas previamente. Dos columnas pueden ser similares, salvo por su contenido en una o dos celdas. Se usarían varias páginas si se consideraran más de tres alternativas. Es posible duplicar una plantilla de tabla de un procesador de texto para crear una matriz de soluciones alternativas del sistema.

### > Matriz de análisis de factibilidad de soluciones

La segunda matriz complementa a la de solución de alternativas del sistema con un análisis y clasificación de las mismas. Se llama **matriz de análisis de factibilidad de soluciones**.

Las columnas de la matriz corresponden a las mismas soluciones alternativas incluidas en la matriz de soluciones alternativas del sistema. Algunas filas corresponden a los criterios de factibilidad descritos en el capítulo. Se añaden filas para describir la solución general y una clasificación de las alternativas. El formato general se muestra en la figura 9.8.

Las celdas contienen notas de la evaluación de factibilidad de cada opción. Puede asignarse una clasificación o calificación por cada criterio (en cuanto a factibilidad operativa se clasificaría a las alternativas con los números 1, 2, 3, etc.) a cada celda. Después de la clasificación o calificación de todas las opciones respecto de todos los criterios, se registra una clasificación o calificación final en la última fila. No todos los criterios de factibilidad son necesariamente de igual importancia, por lo que antes de asignar la calificación final se pueden eliminar las soluciones alternativas que no son viables en relación con cualquier criterio. En realidad, eso ocurre con suma frecuencia.

Una matriz de análisis de factibilidad de soluciones terminada se presenta en la figura 9.9. En ella se proporciona la evaluación de factibilidad de cada solución alternativa. En este ejemplo se registra directamente una calificación en la celda para la evaluación del criterio de factibilidad de cada opción. Estas ponderaciones permiten cuantificar el análisis. Sin embargo, debe tenerse en cuenta la necesidad de eliminar cualquier solución que no sea del todo viable respecto de algún criterio. Por ejemplo, no podría considerarse una solución cuya puesta en práctica requiriese transgredir los contratos con proveedores.

---

**matriz de análisis de factibilidad de soluciones** Herramienta usada para clasificar las soluciones alternativas del sistema.

**FIGURA 9.9** Ejemplo de matriz de análisis de factibilidad de soluciones

	<b>Wt</b>	<b>Alternativa 1</b>	<b>Alternativa 2</b>	<b>Alternativa 3</b>
Descripción		Compra de paquete comercial listo para uso, para servicios a miembros.	Construir una nueva aplicación en la compañía con el VB. NET estándar de la empresa y la base de datos SQL Server	Programar a la medida la aplicación actual con Powerbuilder.
Factibilidad operativa	15%	Soporta únicamente los requerimientos de servicios a miembros. Tendrían que modificarse los procesos de negocios actuales para aprovechar la funcionalidad del software. Además, preocupa la seguridad del sistema. <b>Calificación: 60</b>	Soporta plenamente la funcionalidad requerida de usuarios. <b>Calificación: 100</b>	Soporta plenamente la funcionalidad requerida del usuario. <b>Calificación: 100</b>
Factibilidad cultural	15%	Posible resistencia de los usuarios a la interfaz de usuario no estándar del paquete que se propone adquirir. <b>Calificación: 70</b>	No se prevén problemas. <b>Calificación: 100</b>	No se prevén problemas. <b>Calificación: 100</b>
Factibilidad técnica	20%	La liberación de la producción actual del paquete Platinum Plus es la versión 1.0 y ha estado en el mercado apenas seis semanas. La madurez del producto es un riesgo y la compañía cobra una cuota mensual adicional por soporte técnico. Se requiere contratar personal con experiencia en Java J2EE o capacitarlo para que realice modificaciones orientadas a los requerimientos de integración. <b>Calificación: 50</b>	La solución requiere escribir una aplicación en VB. NET. Aunque el personal técnico actual sólo tiene experiencia con Powerbuilder, debe ser relativamente sencillo encontrar programadores con experiencia en VB. NET. <b>Calificación: 95</b>	Aunque el personal técnico actual se siente a gusto con Powerbuilder, a los administradores les preocupa la adquisición de Powerbuilder de Sybase Inc. MS SQL Server es la norma actual de la compañía para base de datos y compite con el DBMS de Sybase. No se tiene garantía de que las versiones futuras de Powerbuilder se "comporten adecuadamente" con nuestra versión actual de SQL Server. <b>Calificación: 60</b>
Factibilidad económica	30%	Costo de desarrollo: Aprox. \$350 000 Recuperación de la inversión: Aprox. 4.5 años Valor presente neto: Aprox. \$210 000 Cálculos detallados: Vea el anexo A <b>Calificación: 60</b>	Aprox. \$418 000 Aprox. 3.5 años Aprox. \$307 000 Vea el anexo A <b>Calificación: 85</b>	Aprox. \$400 000 Aprox. 3.3 años Aprox. \$325 000 Vea el anexo A <b>Calificación: 90</b>
Factibilidad de calendario	10%	Menos de tres meses <b>Calificación: 95</b>	9-12 meses <b>Calificación: 80</b>	9 meses <b>Calificación: 85</b>
Factibilidad legal	10%	No se prevén problemas <b>Calificación: 100</b>	No se prevén problemas <b>Calificación: 100</b>	No se prevén problemas <b>Calificación: 100</b>
Calificación ponderada	100%	67	92.5	87.5



## La propuesta del sistema

Recuerde, del capítulo 4, que la fase de análisis de decisiones comprende identificar las soluciones alternativas, analizarlas, compararlas y luego seleccionar la que sea mejor en general, para recomendar por último una solución. Ha aprendido ya cómo realizar las tres primeras tareas. A continuación se estudia la manera de recomendar una solución.

Recomendar una solución incluye generar una **propuesta del sistema**. Este producto usualmente es un informe formal por escrito o una presentación verbal, las cuales se dirigen a los usuarios y propietarios del sistema. Por lo tanto, el analista de sistemas debe ser capaz de redactar un documento formal y hacer una presentación de negocios sin entrar en detalles técnicos o alternativas. En los párrafos siguientes se analizan algunos conceptos importantes de los informes escritos y presentaciones.

**propuesta del sistema** Informe o presentación de una solución recomendada.

### > Informe escrito

El informe escrito es el método de que más utilizan los analistas para comunicarse con los usuarios de sistemas. Tienden a generar informes largos y voluminosos, de aspecto impresionante. Aunque en ocasiones son necesarios, lo más frecuente es que no sea así. Si un administrador recibe un informe técnico de 300 páginas, es factible que lo hojee, sin leerlo, y puede tener casi la certeza de que no lo estudiará cuidadosamente.

**Extensión del informe escrito** El método del ensayo y error ha enseñado a los autores la extensión apropiada de un informe. Las siguientes son directrices generales para limitarla:

- Dirigido a administradores de nivel ejecutivo: una o dos páginas.
- Dirigido a administradores de nivel medio: tres a cinco páginas.
- Dirigido a supervisores: menos de 10 páginas.
- Dirigido a personal de oficina: menos de 50 páginas.

Es posible organizar un informe más largo de manera que incluya subinformes para administradores de los diversos niveles. Estos subinformes usualmente se incluyen como las primeras secciones del informe y lo resumen, enfocados en la información trascendente.

**Organización del informe escrito** Existe una estructura general para organizar cualquier informe. Todo informe consiste en elementos primarios y secundarios. Los *elementos primarios* abarcan la información real que pretende comunicar el informe. Son ejemplo de ello la introducción y la conclusión.

Si bien los elementos primarios muestran la información real, todos los informes también incluyen *elementos secundarios*, que articulan el informe de modo que el lector pueda identificar fácilmente el informe mismo y sus elementos primarios. Además, dan aspecto profesional al documento.

Como se indica en la figura 9.10, los elementos primarios pueden organizarse en dos formatos: factual y administrativo. El *formato factual* es el tradicional y se adecua mejor

**FIGURA 9.10** Formatos para informes escritos

Formato factual	Formato administrativo
I. Introducción	I. Introducción
II. Métodos y procedimientos	II. Conclusiones y recomendaciones
III. Hechos y detalles	III. Resumen y discusión de hechos y detalles
IV. Análisis y discusión de hechos y detalles	IV. Métodos y procedimientos
V. Recomendaciones	V. Conclusión final
VI. Conclusiones	VI. Apéndices con hechos y detalles

**FIGURA 9.11** Elementos secundarios de un informe escrito**Carta de presentación****Página de título****Contenido****Lista de figuras, ilustraciones y tablas****Sinopsis o resumen ejecutivo***(Los elementos primarios —que conforman el cuerpo del informe en los formatos factual o administrativo— se describen en esta sección del informe.)***Apéndices**

a lectores interesados en hechos y detalles, así como en las conclusiones. Es el formato que se usaría para especificar los requerimientos y las descripciones de diseño detalladas a los usuarios del sistema. Sin embargo, es inapropiado para muchos administradores y ejecutivos.

El *formato administrativo* es uno moderno y orientado a resultados, que prefieren muchos administradores y ejecutivos. Se diseña para lectores interesados en resultados, no en hechos. Presenta en primer término las conclusiones o recomendaciones. Cualquier lector puede leerlo directamente hasta llegar al punto en el que el nivel de detalles excede su interés.

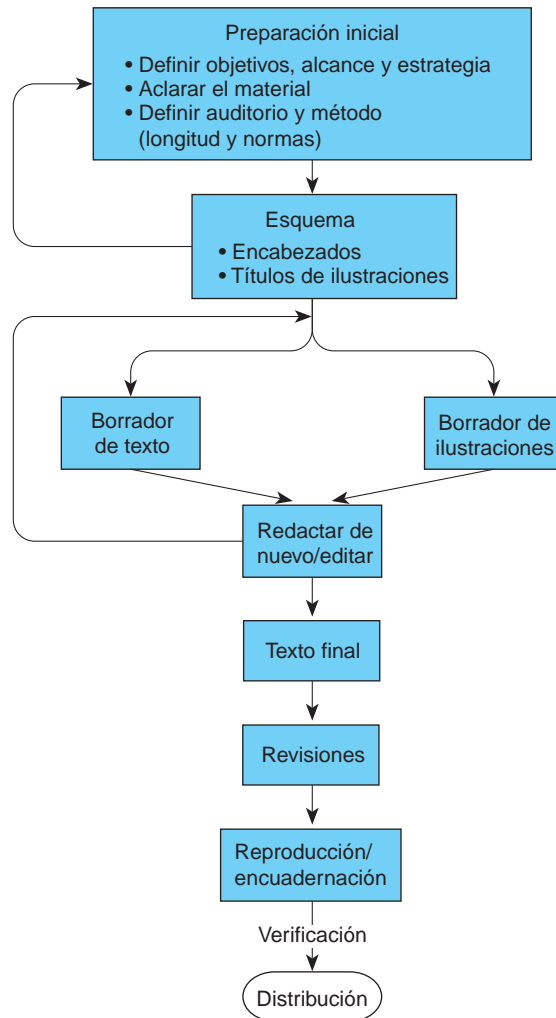
Los dos formatos tienen elementos comunes. La *introducción* debe abarcar cuatro componentes: propósito del informe, definición del problema, alcance del proyecto y explicación narrativa del contenido del informe. La *sección de métodos y procedimientos* debe explicar brevemente cómo se desarrolló la información contenida en el informe; por ejemplo, de qué manera se realizó un estudio o como se diseñará un nuevo sistema. La parte más voluminosa del informe es la *sección de hechos*. Esta última debe recibir un nombre que indique el tipo de datos de hechos que se presenta (por ejemplo, “Descripción del sistema existente”, “Análisis de soluciones alternativas” o “Especificaciones de diseño”). En las *conclusiones* se debe resumir brevemente el reporte y corroborar la definición del problema, de los resultados y las recomendaciones.

La figura 9.11 muestra los elementos secundarios del informe y su relación con los elementos primarios. Muchos de esos elementos se explican por sí solos. Aquí únicamente se comentan con brevedad los que no se explican a sí mismos. Ningún informe se debe distribuir sin una *carta de presentación* al receptor. Esta carta debe ser claramente visible, en lugar de estar oculta en el interior del informe. La carta de presentación señala las acciones necesarias en relación con el informe. Además, llama la atención acerca de características del proyecto o informe que merecen atención especial y es el documento apropiado para reconocer la ayuda recibida de diversas personas.

La *sinopsis* o *resumen ejecutivo* es de una o dos páginas y, como su nombre lo indica, resume todo el informe. Ayuda a que los lectores decidan si el informe contiene la información que necesitan conocer. También sirve como informe sumario del más alto nivel. Casi todos los administradores leen este tipo de resúmenes. Muchos continúan con la lectura, posiblemente con omisión de los hechos detallados y apéndices.

**Redacción del informe** La figura 9.12 muestra el procedimiento adecuado para redactar un informe formal. He aquí algunas directrices que deben seguirse:

- *Cada párrafo debe comunicar una sola idea.* Los párrafos deben fluir adecuadamente, de uno a otro. Su estructura deficiente casi siempre se relaciona con problemas de esquematización.
- *Las oraciones no deben ser excesivamente complejas.* La longitud promedio de las oraciones no ha de exceder de 20 palabras. Diversos estudios indican que las oraciones de mayor extensión son difíciles de leer y entender.
- *Redactar en voz activa.* La voz pasiva se vuelve aburrida y se convierte en parloteo cuando se usa insistentemente.

**FIGURA 9.12**

Pasos en la redacción de un informe

Fuente: Copyright Keith London.

- *No usar tecnicismos, palabras rimbombantes ni “paja”.* Por ejemplo, reemplace “DBMS” con “*database management system* (sistema de administración de bases de datos)” y “Así” con “Por consiguiente”; pruebe “útil” en lugar de “ventajoso”, y use “claramente” en lugar de “está claro que”.

Todo hombre de negocios debe tener un ejemplar de libros del tipo de *The Elements of Style*, de William Strunk Jr. y E. B. White. Este libro clásico encuadernado a la rústica podría establecer una marca sin precedente en cuanto a su relación de valor sobre costo. Apenas más grande que un libro de bolsillo es una mina de oro de información.

## > Presentación formal

El analista de sistemas frecuentemente debe elaborar presentaciones formales para comunicar la información a las numerosas personas que participan en un proyecto de desarrollo de sistemas. Las **presentaciones formales** son juntas especiales que se usan para convencer acerca de nuevas ideas y lograr la aprobación de nuevos sistemas. También pueden servir para cualquiera de los propósitos siguientes: convencer respecto de un nuevo sistema, nuevas ideas o cambios, eliminar críticas; hacer frente a preocupaciones; verificar conclusiones; aclarar hechos, e informar sobre avances. En muchos casos, la presentación formal se elabora de modo que complementa al informe escrito, más detallado.

Las presentaciones efectivas y exitosas requieren preparación considerable. El tiempo asignado a las presentaciones suele ser breve, por lo que su organización y formato son aspectos críticos. No se puede improvisar y esperar aceptación.

Las presentaciones tienen la ventaja de impactar a través de la retroalimentación inmediata y las respuestas espontáneas. Los presentes pueden responder al presentador, quien a su vez puede hacer énfasis, usar pausas oportunas y aprovechar el lenguaje corporal

### presentación formal

Junta especial que se usa para vender nuevas ideas y obtener la aprobación de nuevos sistemas.

### FIGURA 9.13 Esquema característico y asignación de tiempo para una presentación verbal

- I. Introducción (una sexta parte del tiempo total disponible)
  - A. Definición del problema
  - B. Trabajo realizado a la fecha
- II. Parte de la presentación (una sexta parte del tiempo total disponible)
  - A. Resumen de los problemas y limitaciones existentes
  - B. Descripción resumida del sistema propuesto
  - C. Análisis de factibilidad
  - D. Calendario propuesto para terminar el proyecto
- III. Preguntas y preocupaciones del auditorio (el tiempo dedicado a esta parte no se incluye en el tiempo asignado a la presentación y conclusiones; este lo determinan quienes hacen las preguntas y expresan sus preocupaciones)
- IV. Conclusiones (una sexta parte del tiempo total disponible)
  - A. Resumen de la propuesta
  - B. Solicitud de acción (dirigida a las autoridades necesarias para que se continúe el desarrollo del sistema)

para comunicar mensajes que son imposibles de explicar con la palabra escrita. La desventaja de las presentaciones es que el material presentado se olvida fácilmente, ya que el lenguaje verbal y los auxiliares visuales son transitorios. Es por eso que frecuentemente las presentaciones van seguidas de un informe escrito, ya sea resumido o detallado.

**Preparación de la presentación formal** El presentador debe conocer a su auditorio. Ello reviste importancia crucial cuando se pretende convencer con la presentación respecto de nuevas ideas y nuevos sistemas. Es frecuente que se piense en el analista de sistemas como en un temido agente del cambio en una organización. Como escribió Maquiavelo en su clásica obra *El Príncipe*:

No hay nada más difícil de realizar ni más peligroso de manejar que iniciar un nuevo orden de cosas. Porque el reformador tiene enemigos en todos los que se benefician del antiguo orden y apenas defensores tibios en quienes sacarían ventaja del nuevo orden, tibieza que resulta en parte del miedo a sus adversarios, y en parte de la incredulidad del hombre, que no cree en nada nuevo hasta que lo experimenta en la realidad.<sup>1</sup>

Las personas suelen oponerse a los cambios. Existe comodidad en lo familiar del estado actual de cosas. Sin embargo, una parte sustantiva del trabajo del analista es originar cambios: de métodos, procedimientos, tecnología, etc. Un analista exitoso debe ser un vendedor efectivo. Es del todo apropiado (y se recomienda con amplitud) que los analistas estudien formalmente técnicas de ventas. A fin de presentar con efectividad el cambio y convencer respecto de él, se necesita confianza en las ideas propias y hechos para sustentarlas. Una vez más, ¡la preparación es la clave!

Por principio de cuentas, defina las expectativas de la presentación; por ejemplo, que un objetivo es buscar la aprobación para continuar el proyecto, otro es confirmar hechos, etc. Una presentación es un resumen de ideas y propuestas orientadas hacia las expectativas del presentador.

Los ejecutivos usualmente pierden el interés ante los detalles excesivos. A fin de evitarlo, la presentación se debe organizar cuidadosamente con base en el tiempo asignado (por lo general, 30 a 60 min). Aunque cada presentación es distinta, la organización y las asignaciones de tiempo recomendadas en la figura 9.13 brindan una idea al respecto. En la figura se ilustran algunos de los temas habituales en una presentación verbal y el tiempo

<sup>1</sup>Nicolás Maquiavelo, *The Prince and Discourses*, traducido por Luigi Richi (Nueva York: Random House, 1940), 1950). Reproducido con autorización de Oxford University Press.

asignado a cada una. Note que este esquema en particular es para una presentación de análisis de sistemas. Otros tipos de presentaciones diferirían un poco del mostrado en la figura.

¿Qué más puede hacer como preparativo para la presentación? Dadas las limitaciones de tiempo, use auxiliares visuales —gráficos previamente elaborados, diapositivas de proyección, diapositivas de Microsoft PowerPoint, etc.— para sustentar su punto de vista. Al igual que un párrafo escrito, cada auxiliar visual debe comunicar una sola idea. Use las directrices que se muestran en la figura 9.14 para la preparación de imágenes y texto.

Microsoft PowerPoint incluye guías de software, llamadas *asistentes*, que sirven de apoyo a los usuarios menos experimentados en la creación de presentaciones de aspecto profesional. Los asistentes dirigen al usuario paso a paso por el proceso de desarrollo mediante una sucesión de preguntas y el ajuste de la presentación con base en las respuestas. A fin de mantener la atención de su auditorio, considere la distribución de fotocopias de los auxiliares visuales al comienzo mismo de la presentación. De esta manera, los presentes no tendrán que tomar muchas notas.



**FIGURA 9.14**  
Lineamientos de auxiliares visuales

Fuente: Copyright Keith London.

Por último, practique la presentación frente al auditorio más crítico que pueda reunir. Sea su propio crítico o, mejor aún, pida a alguien más que le haga críticas y objeciones. Practique sus respuestas a esos planteamientos.

**Realización de la presentación formal** Unos cuantos lineamientos adicionales pueden mejorar la presentación:

- *Vístase como un profesional.* La manera de vestir influye en las personas. Los libros *Dress for Success* y *The Woman's Dress for Success Book* de John T. Malloy, lecturas excelentes para la elección del guardarropa, son el resultado de estudios concernientes a los efectos de la vestimenta en los administradores.
- *Evite usar la primera persona durante la presentación.* Hable de “usted” y “nosotros” para asignar la propiedad del sistema propuesto a los administradores.
- *Mantenga el contacto visual con el grupo y muéstrese confiado.* Si no muestra a los administradores que cree en su propia propuesta, ¿por qué deberían creer en ella?
- *Tenga conciencia de sus propios hábitos.* Algunos de los más frecuentes incluyen hacer gestos excesivos con las manos, caminar incesantemente de un lado a otro y repetir ciertas expresiones o muletillas, por ejemplo “Como ustedes saben” o “¿De acuerdo?” Aunque los hábitos por sí solos no contradicen el mensaje, sí pueden distraer al auditorio.

En ocasiones, mientras se imparte la presentación algunos miembros del auditorio no escuchan. Esta falta de atención puede asumir varias formas. Algunas personas se enfrascan en conversaciones simultáneas, otras sueñan despiertas, algunas ven su reloj de pulsera, otras más que escuchar tienen una expresión de perplejidad y algunas parecen inexpresivas. Las recomendaciones siguientes pueden ser útiles para hacer que le escuchen:

- *Deje de hablar.* El silencio puede ser ensordecedor. Los mejores oradores públicos saben hacer pausas especiales con fines de énfasis.
- *Plantee una pregunta y deje que alguien del auditorio la responda.* Esto hace que el auditorio participe en la presentación y es una forma muy efectiva de detener una conversación simultánea.
- *Pruebe con un poco de humor.* No tiene que ser un comediante talentoso. Sin embargo, a todo mundo le gusta reírse. Cuente una broma acerca de usted mismo.
- *Use auxiliares.* Recorra a algún tipo de auxiliar visual para aclarar lo que dice. Escriba en un pizarrón, ilustre en el anverso de sus notas o cree un modelo físico para facilitar la comprensión del mensaje.
- *Cambie el volumen de su voz.* Al aumentarlo o disminuirlo hace que el auditorio le escuche o facilita que se le escuche. De una u otra forma habrá cambiado el volumen al que se había acostumbrado su auditorio y ésa es la mejor forma de llamar y conservar su atención.
- *Haga algo inesperado.* Deje caer un libro, revuelva sus notas o juegue con su llavero. Hacer lo inesperado casi siempre llama la atención.

Una presentación formal usualmente incluye tiempo para responder preguntas del auditorio. Este tiempo es muy importante, ya que permite aclarar conceptos confusos y hacer énfasis adicional en ideas trascendentes. Además, posibilita la interacción con su auditorio. Sin embargo, en ocasiones contestar preguntas luego de una presentación puede ser difícil y frustrante. Le hacemos las recomendaciones siguientes para responder preguntas:

- *Responda invariablemente las preguntas con seriedad, inclusive si las considera absurdas.* Recuerde que si hace sentir a alguien que es estúpido por hacer una pregunta “tonta”, le ofenderá. Además, otros miembros del auditorio dejarían de hacer preguntas por temor a recibir el mismo trato.
- *Responda al individuo que le hizo la pregunta y a todo su auditorio.* Si dirige toda su atención a quien hizo la pregunta, el resto del auditorio podría aburrirse. Si no dirige atención suficiente a esa persona, podría no quedar satisfecha. Trate de lograr el equilibrio entre ambos extremos. Si la pregunta no es de interés general para todo su auditorio, puede responderla directamente a esa persona, más adelante.

- *Resuma sus respuestas.* Sea suficientemente específico al responder las preguntas, sin enfrascarse en detalles excesivos.
- *Limite el tiempo que dedica a responder una pregunta dada.* Si necesita tiempo adicional, espere hasta terminar la presentación.
- *Sea sincero.* Si no sabe la respuesta a una pregunta, acéptelo. Por ningún concepto trate de dar la impresión de que conoce la respuesta. El auditorio se dará cuenta tarde o temprano y su credibilidad disminuirá. En vez de ello, prometa indagar al respecto y proporcionar información más adelante. O pida a alguien del auditorio investigue sobre el tema y presente posteriormente sus resultados.

**Seguimiento a la presentación formal** Como se señaló, es muy importante dar seguimiento a una presentación formal, ya que el lenguaje hablado y los asombrosos auxiliares visuales que se usan en la presentación no suelen dejar una impresión duradera. Por ello, muchas presentaciones van seguidas de informes escritos, que brindan al auditorio una copia permanente de la información transmitida.



## Resumen

1. La factibilidad es una medición del grado en que el desarrollo de un sistema de información sería beneficioso para una organización. El análisis de factibilidad es el proceso de medirla. Se trata de una evaluación continua en diversos controles del ciclo de vida. En cualquiera de esos puntos se podría cancelar, modificar o continuar el proyecto. Esto se denomina estrategia de compromiso creciente con la factibilidad.
2. Existen seis pruebas de factibilidad: operativas, culturales/políticas, técnicas, de calendario, económicas y legales.
  - a) La factibilidad operativa es una medición de la urgencia del problema o la aceptabilidad de la solución. Incluye la medición del sentir de los usuarios finales y administradores acerca de los problemas o las soluciones.
  - b) La factibilidad cultural (o política) es una medición del sentir de una persona acerca de una solución y el grado en que será aceptada.
  - c) La factibilidad técnica es una medición del grado en que son prácticas las soluciones y si se cuenta o no con la tecnología necesaria en la organización. En caso de no contar con ella, la factibilidad técnica también considera si es posible adquirirla.
  - d) La factibilidad de calendario corresponde a medir el grado en que los tiempos o fechas límite de un proyecto son razonables.
  - e) La factibilidad económica mide si una solución permitirá recuperar la inversión correspondiente o cuán rentable será. Es la más importante de las cuatro mediciones, para los administradores.
  - f) La factibilidad legal cuantifica el grado en que es posible implantar una solución dentro de las obligaciones legales y contractuales existentes.
3. El análisis de la factibilidad económica requiere pormenorizar los beneficios y los costos. Los beneficios son tangibles (fáciles de medir) o intangibles (de difícil medición). A fin de analizar correctamente la factibilidad económica, hay que tratar de estimar el valor de todos los beneficios. Los costos son de dos categorías: de desarrollo y operativos.
  - a) Los costos de desarrollo se desembolsan una sola vez en relación con el análisis, diseño e implantación del sistema.
  - b) Los costos de operación pueden ser fijos en el tiempo o variables conforme al uso del sistema.
4. Dados los costos y beneficios, la factibilidad económica se valora con las técnicas de análisis de costo-beneficio. Estas últimas determinan si un proyecto o solución será rentable, es decir, si los beneficios serán mayores que los costos durante el ciclo de vida. Son tres las formas más usadas de medir la relación costo-beneficio: análisis de la recuperación de la inversión, análisis del rendimiento sobre la inversión y análisis del valor presente neto.
  - a) El análisis de la recuperación de la inversión determina el tiempo que tardará en pagarse la inversión.
  - b) Los análisis del rendimiento sobre la inversión y del valor presente neto determinan la rentabilidad de un sistema.

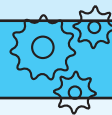
- c) Se prefiere el análisis de valor presente neto porque permite comparar alternativas con ciclos de vida distintos.
- Una matriz de soluciones alternativas del sistema es una herramienta útil para documentar las similitudes y diferencias entre diferentes soluciones del sistema que se consideran.
  - La matriz de análisis de factibilidad de soluciones sirve para evaluar y clasificar las soluciones alternativas del sistema. Junto con la matriz de análisis de factibilidad de soluciones, resulta útil presentar los resultados del análisis de factibilidad como parte de una propuesta del sistema.
  - Los informes escritos son el medio de comunicación que más usan los analistas. Consisten en elementos primarios y secundarios. Los primarios contienen información de hechos, mientras que los secundarios, por así decirlo, articulan el informe para facilitar su uso. Los informes suelen organizarse en formato factual o administrativo. El primero muestra los detalles antes de las conclusiones, mientras que en el segundo se invierte el orden. Los administradores gustan del formato administrativo porque se orienta a resultados y va directamente a la pregunta de fondo.
  - Las presentaciones formales son un tipo especial de junta en la que una persona presenta las conclusiones, ideas o propuestas a un auditorio interesado. La preparación es la clave de las presentaciones efectivas.
  - La propuesta del sistema puede ser un informe escrito formal o una presentación verbal.

## Preguntas de repaso



- ¿Qué significa la estrategia de compromiso creciente con el análisis de factibilidad?
- ¿Cuáles son los controles del análisis de factibilidad en el ciclo de desarrollo? ¿Qué debe hacerse en cada uno de ellos?
- ¿Cuáles son los objetivos de la prueba de factibilidad operativa?
- ¿Por qué es importante indagar el sentir de los usuarios finales y administradores acerca de la solución del problema que ha identificado el analista de sistemas?
- ¿Cuándo se utiliza el análisis de utilidad? ¿Cuál es el objetivo de dicho análisis?
- ¿Cuál es el objetivo de la prueba de factibilidad técnica?
- ¿Cuáles son las características de los costos operativos y de desarrollo? Señale tres ejemplos de cada tipo de costo.
- Enumere cinco ejemplos de beneficios tangibles.
- ¿Por qué el concepto del valor del dinero en el tiempo es una consideración esencial en la evaluación de la factibilidad económica?
- ¿Cuáles son las técnicas más usadas para determinar la relación costo-beneficio de un proyecto?
- ¿Cuál es el uso de las matrices de soluciones alternativas del sistema y de análisis de factibilidad de soluciones?
- En relación con informes escritos, ¿cuál es la diferencia entre los formatos factual y administrativo?
- ¿Cuáles son los pasos de redacción de un informe?
- ¿Cuáles son las ventajas y desventajas de las presentaciones?
- ¿Qué debe hacerse como seguimiento de una presentación formal?

## Problemas y ejercicios



- En este texto se describe una estrategia de compromiso creciente con la factibilidad.
  - Explique dicha estrategia y la razón por la que se aconseja en esta obra.
  - ¿Cuáles son algunos de los cambios o eventos que podrían ocurrir, que sugieren tal estrategia?
  - ¿Acaso una organización debe cancelar un proyecto si se vuelve no viable?
- En el libro también se describen tres controles para medir la factibilidad.
  - ¿Cuáles son esos controles?
  - En general, ¿con qué grado de exactitud puede determinarse la factibilidad en cada control?
  - ¿Cuál control, si lo hay, es el más crítico?
- ¿Cuáles son las cuatro categorías de pruebas de factibilidad y cuál es el criterio que se usa en cada una de ellas para medir la factibilidad?
- Suponga que es diseñador de sistemas de un proyecto que está cerca de la terminación de la fase de diseño del sistema. Se ha desarrollado un prototipo funcional y se le ha pedido que realice el análisis de utilidad. Elabore un plan de una o dos páginas en el que detalle su estrategia para ejecutar dicho análisis.
- Trabaja como analista de sistemas en el departamento de TI de una organización de tamaño mediano, con unos 300 empleados. La organización está en la fase de diseño de un proyecto para desarrollar un sistema de informes electrónicos de actividades de todos los empleados, que sustituirá al método ac-



tual, de informes impresos. Todo el trabajo se realizará en la organización, salvo por varios consultores que prestarán servicios auxiliares, como IV&V. La aplicación se usará con las computadoras existentes de los empleados, si bien será necesario adquirir varios servidores dedicados (privados). La interfaz de usuario es muy intuitiva; pero el proyecto requiere casi medio día de capacitación de todos los empleados sobre las políticas y procedimientos de uso de la nueva aplicación. El sistema no usará ninguna tecnología nueva y el personal técnico de TI cuenta con mucha experiencia en la tecnología que se aplicará. Genere una hoja de cálculo en la que detalle los costos de desarrollo estimados de una sola ocasión y los costos operativos continuos. Al respecto, los sueldos y prestaciones de los analistas de sistemas de su organización promedian 40 dólares por hora, dato que puede usar como base para estimar los sueldos y beneficios de otros empleados que participen en el proyecto.

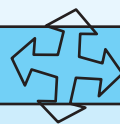
6. En el proyecto descrito en el problema anterior, el sistema de informes electrónicos de actividades reemplazará al sistema manual vigente. Describa los beneficios tangibles que podrían esperarse. Use la estrategia de “la mejor conjetura” (*best guess*) y calcule los ahorros anuales para la organización. Detalle sus supuestos en los cálculos.
7. Está diseñando un sistema basado en World Wide Web, en el que sus oficinas regionales pueden enviar en línea los informes de ventas, en vez de llenarlos a mano y enviarlos por correo postal. Se han identificado tres soluciones alternativas. Los beneficios y costos estimados durante su ciclo de vida se muestran a continuación, todos ajustados en el tiempo respecto del ciclo de vida de cinco años de cada alternativa.

	<b>Beneficios estimados durante el ciclo de vida</b>	<b>Costos estimados durante el ciclo de vida</b>
Solución candidata 1:	\$640 000	\$172 000
Solución candidata 2:	\$640 000	\$160 000
Solución candidata 3:	\$640 000	\$185 000

De conformidad con el análisis de rendimiento sobre la inversión, ¿cuál solución tendría el ROI más alto? Si la organización establece un ROI durante el

ciclo de vida de por lo menos 80%, ¿cuál de las soluciones es económicamente factible?

8. ¿Cuáles son las diferentes técnicas o métodos para la identificación de soluciones alternativas? Si tuviera que elegir uno de ellos, ¿cuál seleccionaría y por qué?
9. Trabaja como diseñador de sistemas de una compañía que fabrica herramientas eléctricas de uso industrial para contratistas. Cada mes, sus centros de servicio y ventas regionales envían juntas las órdenes de reparación impresas de trabajo realizado bajo garantía. Los envían a las oficinas generales, donde son sometidos a un proceso por lotes en un *mainframe* heredado. Luego se genera un informe, que los ingenieros analizan en búsqueda de signos de tendencias problemáticas en los nuevos modelos. El director general de la compañía ha decidido que este proceso es excesivamente lento en el ambiente de negocios actual, muy competido, y quiere reemplazar el sistema heredado tan pronto como sea posible con otro más moderno. Identifique por lo menos tres soluciones alternativas y descríbalas en una matriz de soluciones del sistema, con la figura 9.7 como ejemplo.
10. Prepare una matriz de análisis de factibilidad con las soluciones alternativas que identificó y describió en el problema anterior. Use la figura 9.9 como plantilla, pero elija los factores de ponderación que en su opinión serían más apropiados en esta situación. Para los fines de este ejercicio puede brindar una estimación de la factibilidad económica.
11. Una vez terminada la matriz de análisis de factibilidad de soluciones, es momento de escribir el informe de factibilidad. En este ejercicio, prepare un informe de factibilidad para administradores de nivel ejecutivo con el formato apropiado, que se muestra en la figura 9.10.
12. Le han pedido que presente el análisis de factibilidad y recomendaciones a los administradores ejecutivos de cada departamento de su organización en su junta semanal. Elabore un conjunto de diapositivas de PowerPoint para su uso como auxiliar visual durante su presentación.
13. Enumere por lo menos 10 ejemplos de lo que *no* se debe hacer si se pretende que una presentación sea informativa y persuasiva, además de ser bien recibida.



## Proyectos e investigación

1. Steve McConell es un autor que ha redactado numerosos libros sobre desarrollo e ingeniería de software. En su obra *Rapid Development* señala que en ingeniería el diseño es usualmente una parte del proyecto mucho menor que la construcción real. Compara los proyectos de construcción de puentes,

en los que el diseño representa casi el 10% y la construcción cerca del 90% del esfuerzo total, con los de desarrollo de software, en los que el diseño suele ser al menos el 50% del esfuerzo total del proyecto.

Explore este tema de las diferencias únicas de la ingeniería de software en comparación con otros tipos de ingeniería y resume su análisis y resultados en un ensayo de una o dos páginas. ¿Qué otros líderes de la ingeniería de software han comentado sobre el tema?

2. Trabaja como analista de sistemas en el cuartel general de la policía de caminos de su estado, que tiene oficinas distribuidas en todo el territorio estatal. Actualmente, los oficiales de la policía redactan a mano los informes de accidentes de tránsito y los revisa su sargento, que los almacena temporalmente y luego los envía cada mes por lotes al cuartel general. Un grupo de operadores de datos captura cada uno de ellos en un *mainframe heredado*, y una vez introducidos los informes de las oficinas regionales de todo el estado, un operador de computadora ejecuta el programa de edición con uso de JCL.

Se rechazan los informes con errores u omisiones importantes y se regresan a la oficina regional de origen para su corrección. Después de que se completa el proceso de edición de todas las jurisdicciones, se ejecuta un programa actualizado que añade el lote mensual de informes de accidentes de tránsito al archivo maestro de informes. Se generan informes estadísticos trimestrales y anuales. El proceso completo desde el momento de recibir los lotes de informes hasta el de la actualización del archivo maestro es de unos tres meses. Los administradores ejecutivos están interesados en sustituir el sistema con algo más moderno, que entrañe un menor uso de mano de obra, sea más preciso, resulte de más fácil acceso para los usuarios y reduzca el tiempo para la preparación de los informes estadísticos. Su tarea, como miembro del equipo del proyecto, es preparar el Informe del Estudio de Factibilidad (FSR, por sus siglas en inglés).

- a) ¿Cuáles son algunas de las opciones o alternativas que en su opinión deben considerarse? (Identifique por lo menos tres, además de “no hacer nada” o “mantener el *statu quo*”.)
  - b) Prepare una matriz de soluciones alternativas del sistema que describa las características de cada una de las opciones, con la plantilla de dicha matriz que se muestra en la figura 9.6.
  - c) Amplíe la matriz de soluciones alternativas del sistema con la plantilla que se ilustra en la figura 9.7.
  - d) Evalúe cada una de esas alternativas en cuanto a su factibilidad operativa, técnica y de calendario, con las técnicas descritas en el libro y la plantilla que se muestra en la figura 9.9.
3. Con base en la situación descrita en la pregunta anterior:

- a) Prepare una hoja de cálculo de costos estimados para cada alternativa con el formato de la figura 9.2.
  - b) Evalúe la factibilidad económica de cada alternativa con una de las tres técnicas descritas en el capítulo. ¿Cuál técnica utilizó y por qué?
  - c) Añada el análisis de factibilidad económica a la matriz de análisis de factibilidad de la pregunta anterior.
  - d) Compare y califique cada una de las alternativas. Use factores de ponderación distintos de los utilizados en el libro para cada criterio de factibilidad.
  - e) ¿Cuáles factores de ponderación eligió para los criterios distintos en su matriz de análisis de factibilidad de soluciones y por qué?
4. Los administradores están impresionados con su trabajo excelente de la matriz de análisis de factibilidad de soluciones y le han pedido que prepare el informe de la propuesta del sistema. Redacte una propuesta cuyo auditorio principal consistirá en administradores de negocios y de TI de nivel intermedio; pero también lo leerán el ejecutivo patrocinador y el director de informática. Use el formato apropiado que se muestra en la figura 9.10.
  5. Los administradores de nivel intermedio resultaron impresionados con su propuesta del sistema. Ahora quieren que prepare y exponga una presentación formal a los altos directivos del área.
    - a) Describa los pasos que daría en la preparación de la presentación formal.
    - b) Prepare una presentación de PowerPoint con las directrices propuestas en este libro o en otros libros y artículos acerca de lo que se debe hacer y lo que no se debe hacer en las presentaciones de PowerPoint.
    - c) En su opinión, ¿qué conocimientos son más críticos en la preparación de la presentación formal y por qué?
  6. Existe una gran variedad de formatos, plantillas y métodos para la preparación de propuestas de sistemas de informes de estudios de factibilidad. Busque en Internet herramientas y técnicas que puedan serle útiles.
    - a) Describa los formatos que encontró y sus fuentes.
    - b) Compare los formatos que encontró entre sí y con el descrito en esta obra. ¿Cuáles son algunas de las diferencias?
    - c) ¿Piensa que uno de los formatos tiene ventajas evidentes sobre los otros? En caso afirmativo, indique cuál y por qué lo considera mejor.
    - d) Genere lo que en su opinión es la plantilla de FSR ideal para su organización.



## Casos breves

La tienda de abarrotes Wow Munchies, de un capítulo previo, considera desarrollar un sitio en línea para la venta de abarrotes a sus clientes. El propietario del negocio piensa que ello le permitirá absorber participación de mercado de un negocio cercano, Fast Food Co., que tiene un sitio web y entregas a domicilio. El sitio permitirá que los clientes adquieran en línea cualquier artículo que tenga en existencia la tienda. Esta última *no* entregará los productos a domicilio, sino que los empacará en bolsas y los tendrá listos para que el cliente los recoja a la hora que desee. Esta tienda no cuenta con sucursales.

1. Realice un estudio de factibilidad operativa. ¿Piensa que el sitio web permitirá que la tienda aumente su participación de mercado, como tiene pensado? ¿Qué factores afectarán el éxito operativo del sitio? Entregue su ensayo, documentos sustentadores, gráficos y las entrevistas que haya realizado.
2. Lleve a cabo un estudio de factibilidad técnica. ¿Qué recomendaría a la compañía en cuanto a la creación y mantenimiento del sitio (por ejemplo, idiomas, *host* específico, encriptación)? ¿Por qué su elección afectará la factibilidad del mencionado sitio? Entregue su ensayo, documentos sustentadores, gráficos y las entrevistas que haya realizado.
3. Efectúe un estudio de factibilidad económica para la inversión en un sitio de comercio electrónico. ¿Qué tasa de descuento usó y por qué? Entregue su ensayo, documentos sustentadores, gráficos y las entrevistas que haya realizado.
4. Elabore un calendario y estudio de factibilidad de planeación para el desarrollo de este sitio web. ¿Aprecia algún factor que podría causar una demora o que exceda la fecha límite? Entregue un ensayo breve y una gráfica de Gantt.



## Ejercicios de equipo e individuales

1. Análisis de mesa redonda: los análisis de ROI suelen realizarse con la consideración de una “vida” tecnológica de 3-5 años. ¿Piensa que la tecnología tiene efecto en la empresa después de ese periodo? ¿Por qué sí o por qué no?
2. Ejercicio individual: En el capítulo previo, analizó (con el formato de mesa redonda) la importancia del conocimiento e información en el éxito económico. Todos los años dejan de otorgarse becas para estudios universitarios porque los estudiantes no las solicitan (posiblemente porque no saben que existen). Investigue sobre tales becas disponibles para usted y solicite por lo menos una.
3. Análisis de equipo/clase: ¿Qué significa tener una actitud para el éxito? ¿Piensa que la creencia de una persona en sus propias capacidades puede influir en su éxito real? En contraparte, ¿opina que la falta de confianza en sí misma perjudica su capacidad para tener éxito? ¿Qué puede hacer para desarrollar o mejorar su propia actitud para el éxito?



## Lecturas recomendadas

Bovee, Courtland L., y John V. Thill. *Business Communications Today*, 2a. ed. Nueva York: Random House, 1989.

Gildersleeve, Thomas R. *Successful Data Processing Systems Analysis*, 2a. ed. Englewood Cliffs, NJ: Prentice Hall, 1985. Este libro incluye un capítulo excelente sobre las técnicas de análisis de costo-beneficio. Las presentaciones son temas del capítulo 4. El concepto de compromiso creciente se debe a su autor.

Gore, Marvin, y John Stubbe. *Elements of Systems Analysis*, 4a. ed. Dubuque, IA: Brown, 1988. El capítulo sobre análisis de factibilidad propone una estrategia matricial interesante para identificar, catalogar y analizar la factibilidad de soluciones alternas para un sistema.

Smith, Randi Sigmund. *Written Communications for Data Processing*. Nueva York: Van Nostrand Publishing, 1976.

Stuart, Ann. *Writing and Analyzing Effective Computer System Documentation*. Nueva York: Hol, Rinehart and Winston, 1984.

Uris, Auren. *The Executive Deskbook*, 3a. ed. Nueva York: Van Nostrand Reinhold, 1988.

Walton, Donald. *Are You Communicating? You Cant't Manage without It*. Nueva York: McGraw-Hill, 1989.

Wetherbe, James. *Systems Analysis and Design: Traditional, Structured, and Advanced Concepts and Techniques*, 2a. ed. St. Paul, MN: West, 1984. Wetherbe fue pionero del marco de referencia PIECES para la clasificación de problemas. En este capítulo se amplía dicho marco al análisis de la factibilidad operativa de las soluciones.



## Métodos de diseño de sistemas

En los capítulos de la parte tres, se presentan los métodos de diseño de sistemas. El capítulo 10, Diseño de sistemas, proporciona el contexto para los capítulos posteriores al introducir las actividades de diseño de sistemas. Éste incluye la preparación de las especificaciones detalladas, basadas en computadora, con las que se cumplirán los requerimientos especificados durante el análisis de sistemas y la construcción de los prototipos de sistema. En lo referente al desarrollo de sistemas de información, el diseño de sistemas consiste en las fases de configuración, adquisición y diseño e integración.

El capítulo 11, Modelado y arquitectura de la aplicación, presenta el proceso físico del diseño de datos. De manera específica, trata sobre las decisiones de diseño concernientes a los aspectos de distribución de los datos y procesos compartidos. Esto

lleva a una arquitectura de aplicaciones consistente en unidades de diseño que se pueden asignar a diferentes miembros del equipo para su diseño, construcción y pruebas de unidad detalladas.

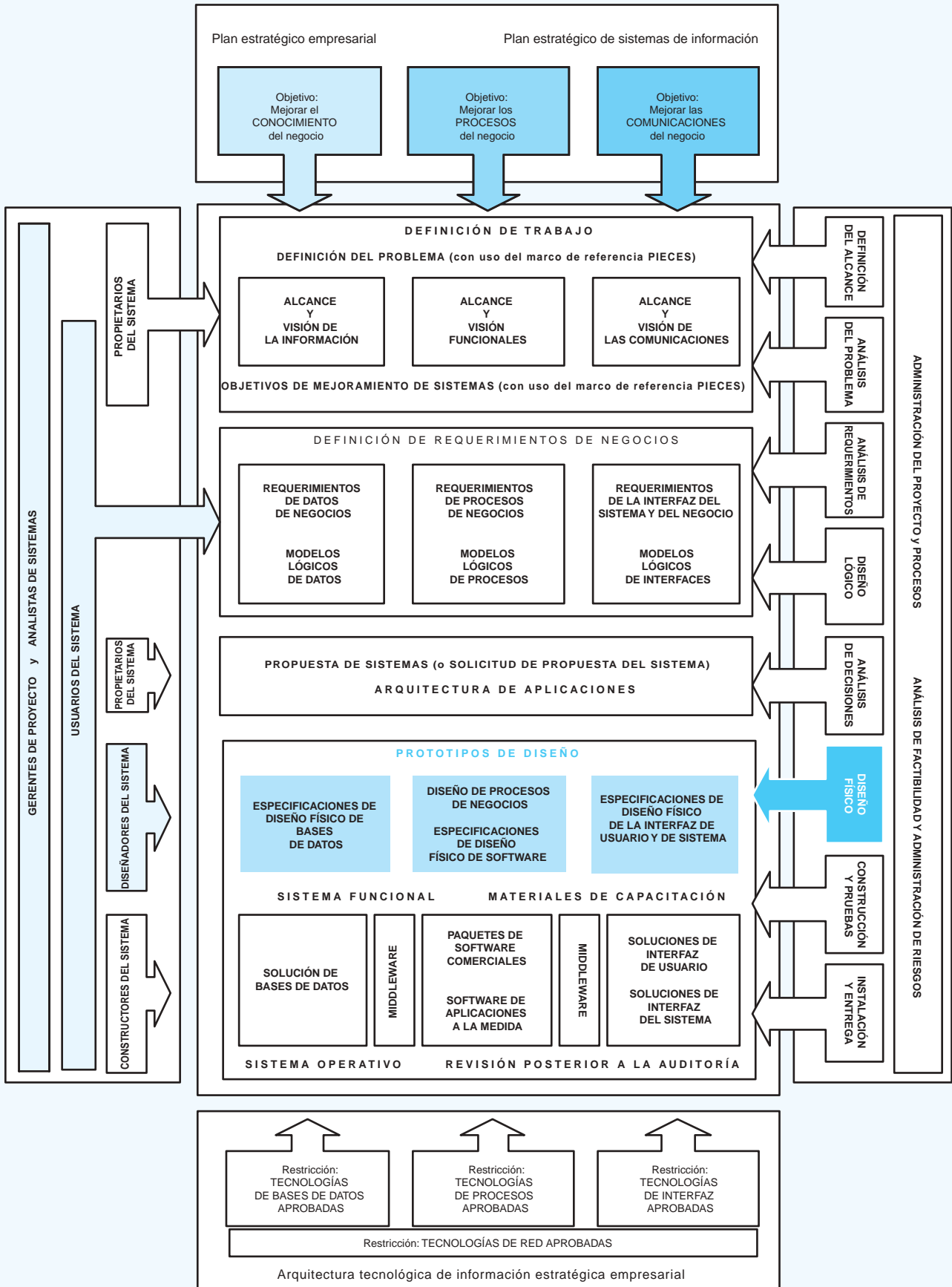
El capítulo 12, Diseño de bases de datos, presenta el diseño de los almacenes físicos de datos a partir del modelo de datos que se desarrolla en el capítulo 7.

El capítulo 13, relacionado con el Diseño de salidas y elaboración de prototipos, trata sobre dichos temas. Se presentan diferentes tipos, formatos y medios de salida. También se comenta el uso de los tipos de gráficas más comunes. El capítulo incluye la forma de diseñar y elaborar prototipos de salidas impresas y en pantalla.

En el capítulo 14, relacionado con el Diseño de entradas y elaboración de prototipos, se hace énfasis en los formatos, métodos,

medios, factores humanos y controles internos de las salidas. Se comenta el uso apropiado de los controles basados en pantalla para la entrada de datos en el diseño de pantallas de la interfaz gráfica de usuario (GUI, por sus siglas en inglés). El capítulo también pone énfasis en los prototipos como una manera de encontrar, documentar y comunicar los requerimientos del diseño de entradas.

Por último, el capítulo 15 versa sobre el Diseño de la interfaz del usuario. En él, se aprende cómo desarrollar una interfaz efectiva y amigable para una aplicación. El diseño de la interfaz de usuario es crucial, ya que la aceptación del sistema por parte del usuario suele depender de que dicha interfaz sea amigable o fácil de usar. Se muestra una interfaz basada en GUI para obtener las entradas y salidas que se diseñaron en los capítulos 13 y 14.



# 10

## Diseño de sistemas

### Panorámica y objetivos del capítulo

En este capítulo usted continuará el estudio de la fase de diseño del desarrollo de sistemas. Sabrá que ha entendido el proceso de diseño de sistemas cuando pueda:

- Describir la fase de diseño en base a los componentes de información.
- Identificar varias estrategias de diseño de sistemas y diferenciarlas.
- Describir las tareas de la fase de diseño en términos de una solución basada en computadora para un proyecto de desarrollo en la propia organización (sistema a la medida).
- Describir la fase de diseño en términos de una solución basada en computadora que involucre la adquisición de una solución de sistemas de software comercial.

Aunque en este capítulo se introducen algunas técnicas de diseño de sistemas, la intención no es describir las técnicas del diseño de sistemas. En el presente capítulo sólo se estudia el proceso de diseño de sistemas y se introducen algunas técnicas que se detallan en capítulos posteriores.

## Introducción

Se ha aprobado la propuesta para el sistema de servicios a miembros de SoundStage. Ahora, el equipo del proyecto de sistemas está convirtiendo el diseño lógico (especificar “qué” debe hacer el sistema) en el diseño físico (especificar “cómo” debe hacerlo). Son muchas las tareas por hacer, lo que incluye el diseño de la base de datos, especificaciones de cómo funciona el sistema y elaboración de prototipos de la interfaz de usuario. Algunas partes del sistema de servicios a miembros se ensamblan a partir de componentes adquiridos. Aunque esto ahorra tiempo de programación, añade pasos al proceso de diseño y pruebas de la manera en que los componentes funcionarán con los ya existentes (software creado a la medida para la organización). Bob Martínez tendrá varias tareas durante el proceso. Está ansioso de empezar. El sistema finalmente empieza a tomar forma, por lo menos en papel y en prototipos.

## ¿Qué es el diseño de sistemas?

**diseño de sistemas** La especificación de una solución detallada basada en la computadora.

En el capítulo 3 usted estudió el proceso de desarrollo de sistemas. En él, el análisis se limita intencionadamente a examinar brevemente cada fase. En este capítulo, se echa un vistazo mucho más cercano a la fase de diseño de sistemas, que sigue al análisis de sistemas. Se define el **diseño de sistemas** de información como las tareas que se enfocan en la especificación de una solución computarizada detallada. También se le llama *diseño físico*. Así pues, mientras que en el análisis de sistemas se pone énfasis el problema del negocio, el diseño de sistemas se enfoca en los aspectos técnicos o de implantación del sistema.

Como se ilustra en la página de inicio de este capítulo, el diseño de sistemas se basa en las preocupaciones técnicas de los DISEÑADORES DE SISTEMAS. Por tanto, se centra en los componentes de IS desde la perspectiva de los DISEÑADORES DE SISTEMAS. LOS ANALISTAS DE SISTEMAS sirven como facilitadores del diseño de sistemas.

Muchos de nosotros definimos muy estrictamente el proceso de diseño. Nos imaginamos trazando esquemas de los sistemas basados en computadora para su programación y desarrollo en manos de nosotros mismos o de los programadores. Así pues, se diseñan entradas, salidas, archivos, bases de datos y otros componentes de computadora. Los reclutadores de especialistas en cómputo recién egresados de las universidades se refieren a esta definición restrictiva como el “síndrome de no se inventó aquí”. En realidad, muchas compañías adquieren más paquetes de software del que crean a la medida para la propia organización. Es algo que no debe sorprenderle. ¿Por qué reinventar la rueda? Muchos sistemas son suficientemente genéricos, así mismo, los proveedores de computadoras han creado paquetes de software adecuados —aunque raras veces, si acaso, perfectos— que se pueden adquirir y posiblemente modificar para adaptarlos a los requerimientos del usuario final.

En este capítulo se examina el diseño de sistemas desde las perspectivas del desarrollo (o “construcción”) de proyectos para la organización o el de proyectos de “compra”. En primer lugar, estudiaremos algunas estrategias generales del diseño de sistemas.

## Estrategias del diseño de sistemas

Hay muchas estrategias o técnicas para llevar a cabo el diseño de sistemas. Esto abarca el *diseño estructurado moderno*, *ingeniería de la información*, *elaboración de prototipos*, *JAD*, *RAD* y *diseño orientado a objetos*. Es frecuente que se considere a estas estrategias como alternativas que compiten entre sí en el diseño de sistemas, aunque en realidad ciertas combinaciones se complementan mutuamente. Analicemos brevemente tales estrategias y el alcance u objetivos de los proyectos a los que se adecuan. La intención es desarrollar únicamente un alto nivel de entendimiento. En capítulos posteriores estudiaremos las técnicas actuales.

NOTA: Recuerde, del capítulo 3, que a veces se definen “rutas” metodológicas para dichas estrategias.



## > Estrategias basadas en modelos

El diseño estructurado, la ingeniería de la información y el diseño orientado a objetos son ejemplos de estrategias basadas en modelos. El **diseño basado en modelos** pone énfasis en el trazado de modelos de sistema pictóricos para documentar los aspectos técnicos o de implantación de un nuevo sistema.

Los modelos de diseño suelen derivarse de modelos lógicos que se desarrollan en una parte previa de la obra, en el análisis basado en modelos (capítulo 4). En última instancia, los modelos de diseño de sistemas se convierten en planos para la construcción e implantación del nuevo sistema.

Hoy, las estrategias basadas en modelos casi siempre se mejoran con el uso de herramientas automatizadas. Algunos diseñadores trazan modelos del sistema con software de gráficos de propósito general, como *Visio Professional* o *Corel Flow*. Otros diseñadores y organizaciones requieren el uso de herramientas CASE basado en repositorios o herramientas de modelado, como *System Architect*, *Microsoft Visio*, *Visible Analyst* o *Rational de IBM*. Las herramientas de CASE brindan consistencia e integridad, además de verificación de errores basada en reglas.

En seguida examinaremos las estrategias de diseño basado en modelos más usadas. Tales estrategias se incluyen en el análisis de rutas y metodología basadas en modelos, que se presenta en el capítulo 3.

**Diseño estructurado moderno** Las técnicas de diseño estructurado ayudan a los desarrolladores a manejar el tamaño y complejidad de los programas. El **diseño estructurado moderno** es una técnica orientada a procesos para dividir un programa grande en una jerarquía de módulos, lo que da por resultado un programa de computadoras más fácil de implantar y mantener (cambiar). Sus sinónimos (si bien técnicamente imprecisos) son *diseño descendente de programas* y *programación estructurada*.

El concepto es sencillo. Se diseña un programa como una jerarquía descendente de módulos. Un módulo es un conjunto de instrucciones: un párrafo, bloque, subprograma o subrutina. La estructura descendente de estos módulos se desarrolla de conformidad con diversas reglas y normas de diseño. (Por tanto, trazar simplemente una jerarquía o gráfico estructural de un programa *no* es diseño estructurado.)

El diseño estructurado se considera como técnica orientada a procesos porque hace énfasis en los componentes de PROCESO del sistema de información, de manera específica, en los procesos de software. El diseño estructurado busca dividir un programa en una jerarquía descendente de módulos que tengan las propiedades siguientes:

- Los módulos deben tener mucha *cohesión*, es decir, cada módulo debe encargarse de una y sólo una función. Esto hace que los módulos sean reutilizables en programas futuros.
- Los módulos deben *acoplarse*, lo cual significa que su dependencia mutua debe ser mínima. Ello minimiza el efecto que los cambios futuros en un módulo tendrán en otros módulos.

La cohesión y el acoplamiento también son conceptos importantes en el mundo orientado a objetos. El modelo de software que se deriva del diseño estructurado se llama *gráfico de estructura* (figura 10.1). Este gráfico se obtiene mediante el estudio del flujo de datos por el programa. El diseño estructurado se realiza durante el diseño de sistemas. No comprende todos los aspectos del diseño; por ejemplo, el diseño estructurado no ayuda a diseñar las entradas, salidas o bases de datos.

El diseño estructurado ha perdido una parte de su popularidad a raíz de que muchas aplicaciones actuales requieren técnicas más novedosas, enfocadas en técnicas de programación *orientadas a eventos* u *orientadas a objetos*. Sin embargo, todavía es una técnica muy utilizada en el diseño de software de aplicaciones para *mainframe* y se utiliza para hacer frente a las cuestiones de acoplamiento y cohesión en el nivel de “sistema”.

**Ingeniería de la información** En el capítulo 4 usted estudió que la *Ingeniería de la Información (IE)*, por sus siglas en inglés) es una técnica de planeación, análisis y diseño de sistemas de información basada en modelos y centrada en DATOS, si bien es sensible a PROCESOS. La herramienta principal de la IE es un diagrama de modelo de datos (figura 10.2). La IE implica realizar un análisis de requerimientos del área de negocios, a partir del cual

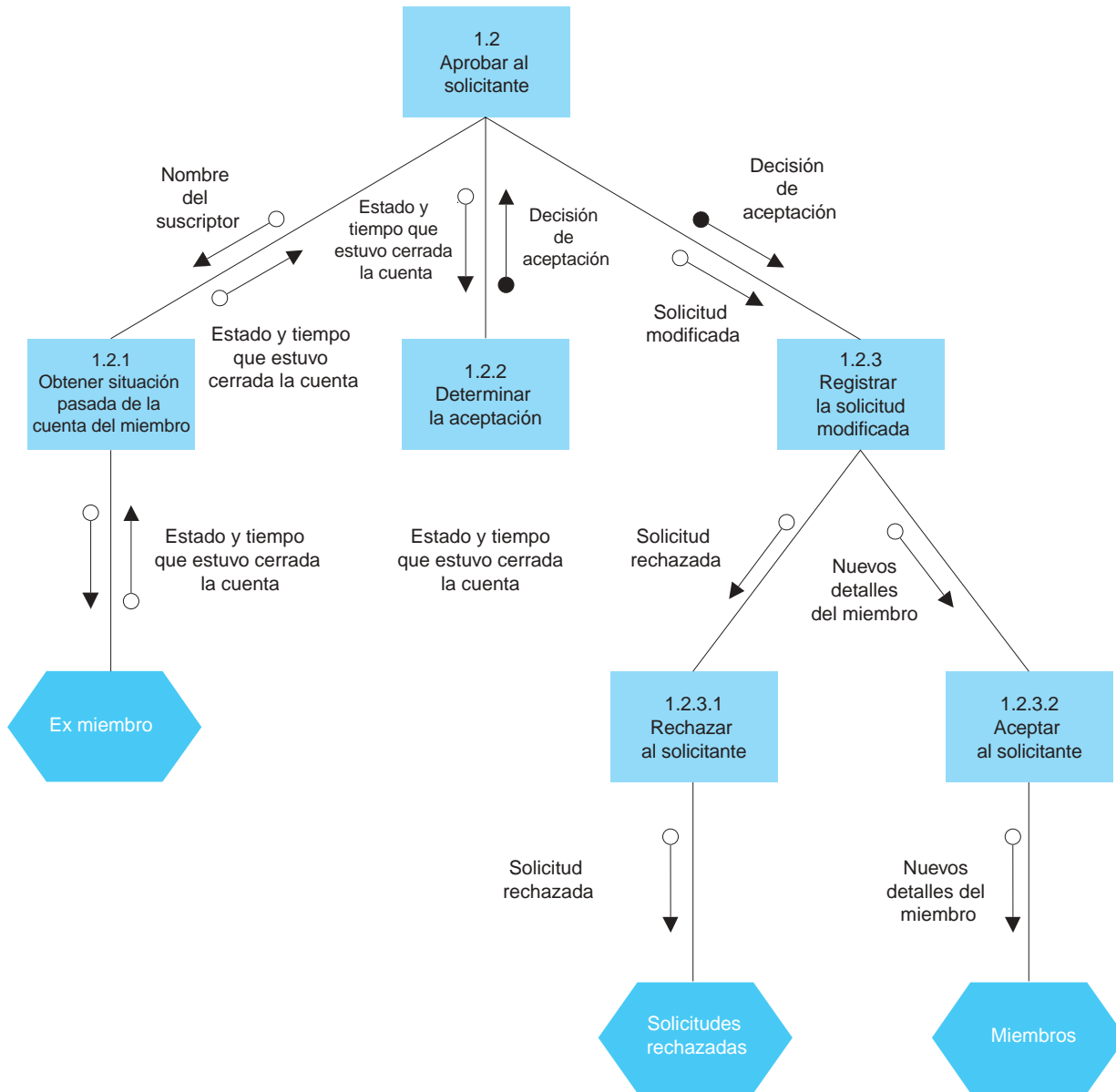
---

**diseño basado en modelos** Estrategia de diseño de sistemas que hace énfasis en dibujar modelos del sistema para documentar los aspectos técnicos y de implantación del propio sistema.

---

**diseño estructurado moderno** Técnica de diseño de sistemas que descompone los procesos del sistema en componentes manejables.

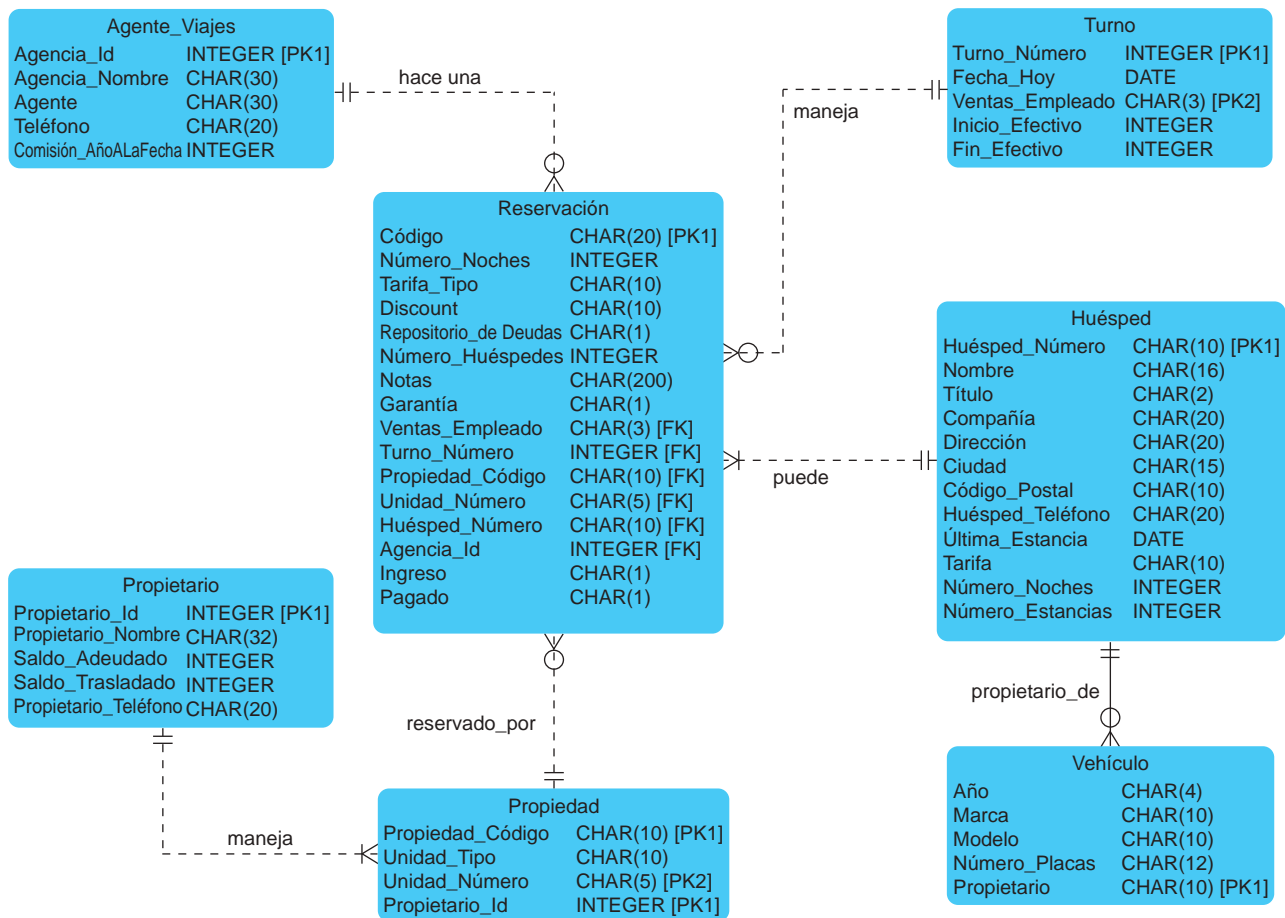




**FIGURA 10.1** Producto final del diseño estructurado

se definen y jerarquizan las aplicaciones del sistema de información. Estas aplicaciones identificadas en la IE se vuelven proyectos, a los que se pretende aplicar otros métodos de análisis y diseño de sistemas para desarrollar los sistemas de producción. Tales métodos podrían incluir alguna combinación de análisis y diseño estructurado moderno (capítulo 4), elaboración de prototipos, análisis y diseño orientados a objetos.

**Elaboración de prototipos** Ha sido tradicional que el diseño físico sea un proceso ejecutado con papel y lápiz. Los analistas trazan imágenes que muestran la distribución o estructura de salidas, entradas y bases de datos, así como el flujo de diálogos y procedimientos. Se trata de un proceso muy tardado, propenso a errores y omisiones considerables. Además, es frecuente que las especificaciones en papel resulten inadecuadas, incompletas o imprecisas.



**FIGURA 10.2** Ejemplo de Diagrama físico de entidad-relación de la ingeniería de la información

Hoy, muchos analistas y diseñadores prefieren los prototipos, estrategia de diseño basada en la ingeniería moderna. La estrategia de prototipos es un proceso iterativo que comprende una relación de trabajo estrecha entre el diseñador y los usuarios. Dicha estrategia implica varias ventajas:

- Los prototipos alientan y requieren la participación activa del usuario final. Esto mejora el estado de ánimo de dicho usuario y su apoyo al proyecto. Tal mejoría del estado de ánimo se debe a que el sistema parece real para el usuario.
- La iteración y cambio son consecuencia natural del desarrollo de sistemas, es decir, los usuarios finales tienden a cambiar de parecer. Los prototipos se ajustan mejor a esta situación natural, ya que suponen que un prototipo evoluciona, por iteración, hasta convertirse en el sistema necesario.
- Se ha afirmado frecuentemente que los usuarios finales no conocen plenamente sus requerimientos hasta que los ven implantados. Si así fuera, los prototipos apoyan tal situación.
- Los prototipos son modelos activos, no pasivos, que el usuario final puede ver, tocar, sentir y experimentar.
- Un prototipo aprobado es un equivalente funcional de una especificación de diseño en papel, con una salvedad: los errores se pueden detectar con mucha mayor anticipación.
- Los prototipos pueden aumentar la creatividad, ya que posibilitan la retroalimentación de los usuarios con mayor prontitud, lo que puede llevar a mejores soluciones.
- Los prototipos aceleran varias fases del ciclo de vida, posiblemente bypassing del programador. De hecho, los prototipos consolidan partes de fases que normalmente ocurren una después de otra.

También existen desventajas o errores con el uso de prototipos. En muchos casos, se pueden resumir con una afirmación: los prototipos alientan el uso de ill-advised shortcuts a lo largo del ciclo de vida. Por fortuna, los errores siguientes pueden evitarse con la disciplina adecuada:

- Los prototipos fomentan el regreso al ciclo de vida de “codificar, implantar y reparar” que en otros tiempos predominaba en los sistemas de información. Muchas compañías saben que, los sistemas desarrollados en lenguajes para elaboración de prototipos pueden tener los mismos problemas de mantenimiento que han plagado a los sistemas heredados desarrollados en lenguajes como *COBOL*.
- Los prototipos no niegan la necesidad de las fases del análisis de sistemas. Un prototipo puede solucionar problemas y oportunidades erróneas con igual facilidad que un sistema desarrollado en forma convencional.
- No es posible sustituir por completo las especificaciones en papel con un prototipo. Ningún ingeniero elaboraría un prototipo de motor sin ningún diseño en papel. Sin embargo, muchos profesionales de sistemas de información tratan de elaborar el prototipo sin especificaciones. Los prototipos deben usarse para complementar otros métodos, no para sustituirlos. El nivel de detalles necesarios del diseño en papel podría disminuir, sin que llegue a eliminarse.
- Son muchos los problemas de diseño que no se solucionan con los prototipos. Estos problemas pueden olvidarse inadvertidamente si no se tiene cuidado.
- Los prototipos suelen llevar al compromiso prematuro con un diseño (usualmente, el primer diseño que se desarrolla).
- Durante la elaboración del prototipo, el alcance y la complejidad del sistema pueden ampliarse rápidamente más allá de los planes originales. La situación podría salirse fácilmente de control.
- Los prototipos pueden reducir la creatividad en el diseño. La naturaleza misma de cualquier implantación —por ejemplo, un prototipo de informe— puede impedir que los analistas, diseñadores y usuarios finales busquen mejores soluciones.
- Los prototipos suelen tener funcionamiento más lento que sus equivalentes de lenguajes de tercera generación (si bien esta diferencia está dejando rápidamente de ser problemática).

Los prototipos se pueden desarrollar rápidamente con muchos de los lenguajes de cuarta generación (4GL) y los de programación orientada a objetos que se tienen en la actualidad. La figura 10.3 muestra una pantalla prototipo de un sistema. Los prototipos se pueden elaborar para salidas sencillas, diálogos de computadora, funciones clave, subsistemas enteros o inclusive todo el sistema. Cada sistema prototipo es objeto de revisión de los usuarios finales y administradores, que hacen sus recomendaciones acerca de requerimientos, métodos y formatos. Luego, se corrige, mejora o refina el prototipo para que refleje los nuevos requerimientos. El proceso de modificación y revisión continúa hasta que se acepta el prototipo. En dicho momento, los usuarios finales aprueban los requerimientos y el diseño que satisface esos requerimientos.

El diseño mediante prototipos no cumple necesariamente con todos los requerimientos de diseño. Por ejemplo, los prototipos no siempre resuelven problemas de funcionamiento y limitaciones de almacenamiento. Además, pocas veces incluyen controles internos. El analista o diseñador todavía debe especificarlos.

**Diseño orientado a objetos** El *diseño orientado a objetos* (OOD, por sus siglas en inglés) es la estrategia de diseño de advenimiento más reciente. La figura 10.4 muestra algunos de los numerosos diagramas usados en el diseño orientado a objetos.

Las tecnologías y técnicas de objetos son un intento por eliminar la separación entre DATOS y PROCESOS. Las técnicas de OOD se usan para refinar las definiciones de requerimientos de objetos identificadas con anterioridad, durante el análisis, y para definir objetos de diseño específico.

The image shows a software prototype for a 'Members' form. The interface is organized into several sections:

- Status:** A group of radio buttons with options: Dropped, Frozen (selected), Good Standing, Inactive, and Probation.
- Member Number:** A text input field containing '100003'. To its right are buttons for 'New', 'Delete', 'Edit', 'Exit', 'Memberships', and 'Orders'.
- Address Information:** A large section containing multiple text input fields: Name (Jeff Whitten), Street Address (129 Rural Route 6), P.O. Box (12999), City (Fowler), State (IN with a dropdown arrow), Zip Code (47756-3333), Area Code (319), Phone (789-3333), and Extension (empty).
- Credit Card Type:** A group of radio buttons with options: American Express, Discover, Mastercard (selected), and Visa.
- Card Number:** A text input field containing '5424-9981-2919-2912'.
- Expiration Date:** A text input field containing '2/2/96'.
- Balance:** A text input field containing '\$495.22'.
- Bonus Balance:** A text input field containing '1'.

At the top right, there is a graphic of two film reels. Below it is a navigation bar with five buttons: a double left arrow, a single left arrow, a single right arrow, a double right arrow, and a refresh/circular arrow button.

**FIGURA 10.3** Ejemplo de Pantalla prototipo

A manera de ejemplo, durante el OOD podría ser necesario que el diseñador revise las características de datos o procesos de un objeto definido durante el análisis del sistema, esto con base en una decisión de implantación de diseño. De manera similar, una decisión de este tipo puede hacer que el diseñador defina un nuevo conjunto de objetos, con una pantalla de interfaz que permita a los usuarios interactuar con el nuevo sistema.

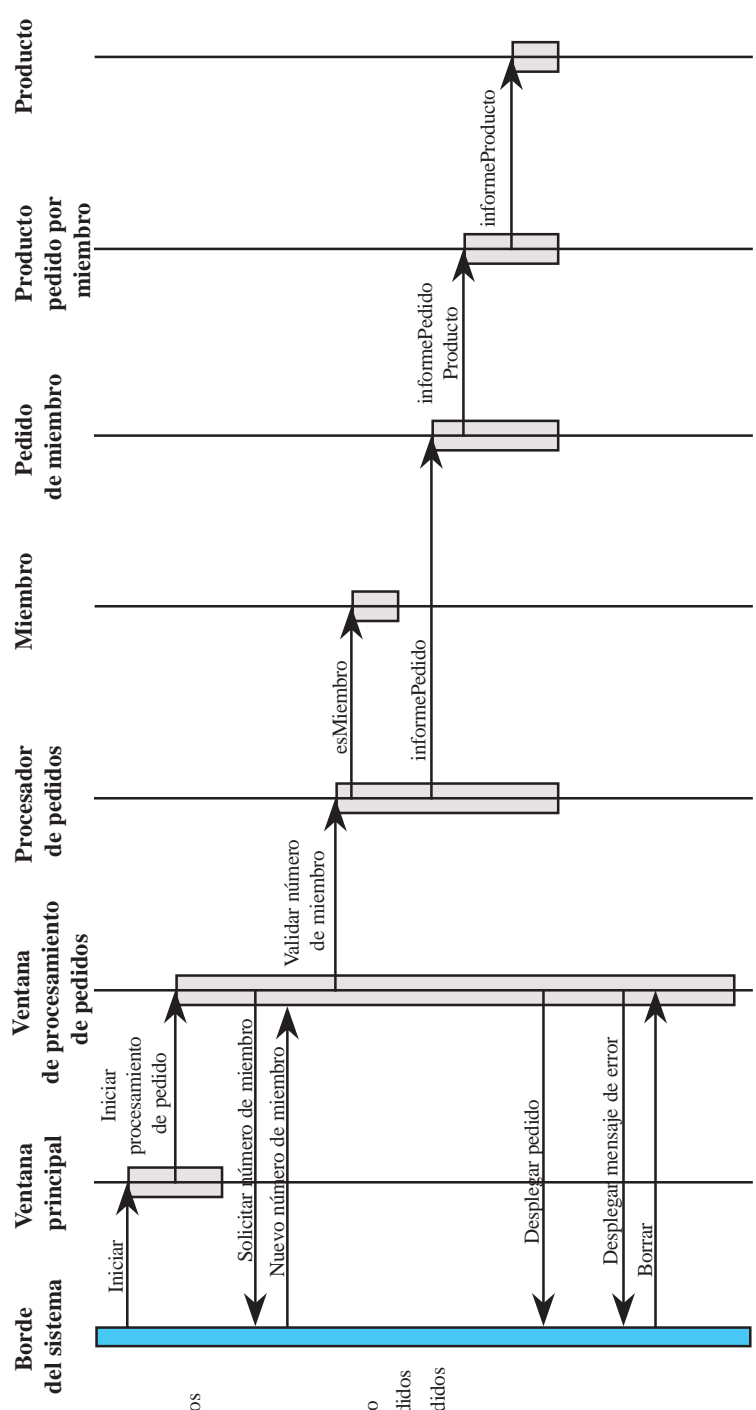
### > Desarrollo rápido de aplicaciones

Es otra estrategia de diseño muy usada en la actualidad. El **desarrollo rápido de aplicaciones (RAD)**, por sus siglas en inglés) es la fusión de varias técnicas estructuradas (en particular, la ingeniería de información basada en datos) con técnicas de prototipos y de desarrollo conjunto de aplicaciones para agilizar el desarrollo de sistemas.

El RAD precisa el uso interactivo de técnicas estructuradas y la elaboración de prototipos para definir los requerimientos de los usuarios y el diseño del sistema final. Al utilizar técnicas estructuradas, el desarrollador primero construye modelos preliminares de procesos y datos de los requerimientos de negocios. Luego, los prototipos pueden ayudar a que los analistas y usuarios verifiquen esos requerimientos, además de refinar formalmente los modelos de procesos y datos. El ciclo de modelos, prototipos, modelos, prototipos, etc., origina en última instancia una definición combinada de requerimientos de negocios y diseño técnico para su uso en la construcción del nuevo sistema.

La aceleración del diseño se logra mediante el énfasis en la participación del usuario en las sesiones de desarrollo conjunto de aplicaciones. Recuerde que el *desarrollo conjunto de aplicaciones (JAD)*, por sus siglas en inglés), introducido en el capítulo 4 y analizado con más detalles en el capítulo 5, es una técnica que se complementa con otras de análisis y diseño de sistemas al hacer énfasis en el *desarrollo participativo* de los PROPIETARIOS, USUARIOS, DISEÑADORES Y CONSTRUCTORES DEL SISTEMA. Durante las sesiones de JAD

**desarrollo rápido de aplicaciones (RAD)** Es estrategia de diseño de sistemas en la que se utilizan técnicas estructuradas, de elaboración de prototipos y de JAD para desarrollar sistemas rápidamente.



El usuario selecciona la opción "nuevo pedido de miembro"  
 Do hasta que no haya más pedidos de miembros

El usuario introduce el número de miembro  
 If número de miembro válido  
 Get encabezado actual de pedido de miembro  
 Do hasta que no haya más productos pedidos  
 Get información de productos pedidos  
 Get información de productos  
 Mostrar pedido  
 Else  
 Mostrar mensaje de error  
 Borrar mensaje  
 Endif

**FIGURA 10.4** Ejemplo de modelo de diseño orientado a objetos



para el diseño de sistemas, el diseñador de sistemas asume el papel de facilitador de varios talleres de jornada completa encaminados a solucionar diversos problemas de diseño y de productos. El JAD es un elemento esencial que contribuye mucho al énfasis del RAD en la aceleración del proceso.

### > Estrategias de diseño de sistemas *FAST*

Al igual que la mayoría de los métodos comerciales, el método *FAST* hipotético de esta obra no impone una sola estrategia de diseño de sistemas. En vez de ello, integra todas las estrategias más usadas que se mencionan en párrafos anteriores. El estudio del caso de SoundStage permite mostrar estos métodos en el contexto de una asignación inicial típica de un analista de sistemas. Las técnicas de análisis de sistemas se aplican en el marco de referencia de:

- Los componentes del sistema de información del lector (tomados del capítulo 2).
- Las fases del desarrollo de sistemas (capítulo 3).
- Las tareas de ejecución de una fase (descritas en este capítulo).

Dado este contexto, a continuación podemos estudiar el diseño de sistemas. Empezaremos por analizar su relación con el desarrollo en la propia organización, o proyectos de “construcción”. Después, examinaremos la manera en que las fases del diseño de sistemas resultan afectadas cuando se toma la decisión de adquirir un paquete comercial de software como solución.

## Diseño de sistemas para desarrollo en la organización: la solución de “construcción”

Empecemos por ubicar el diseño de sistemas de proyectos de desarrollo en la organización en el contexto relativo al ciclo de vida del sistema. Como se ilustra en la figura 10.5, una propuesta de sistema aprobada luego de la fase de análisis da lugar a la fase de diseño. Esta última tiene dos objetivos. Primero, que el analista busca diseñar un sistema que satisfaga los requerimientos y sea amigable con los usuarios finales. La ingeniería humana desempeña una función clave en el diseño. Segundo, y también muy importante, que el analista intenta presentar especificaciones claras y completas a los técnicos y programadores de computación. Como se menciona en la figura 10.5, las especificaciones de diseño físico aprobadas llevan a la fase de construcción del proyecto de desarrollo en la organización.

La figura 10.6 es un diagrama de tareas que ilustra el trabajo necesario para completar la fase de diseño. Este diagrama no impone ningún método en particular, si bien en los párrafos siguientes se describen las estrategias, herramientas y técnicas que el lector podría considerar para cada tarea de diseño. El diagrama no es más que una plantilla. El equipo y el administrador del proyecto podrían ampliarla o modificarla para reflejar las necesidades únicas de un proyecto dado.

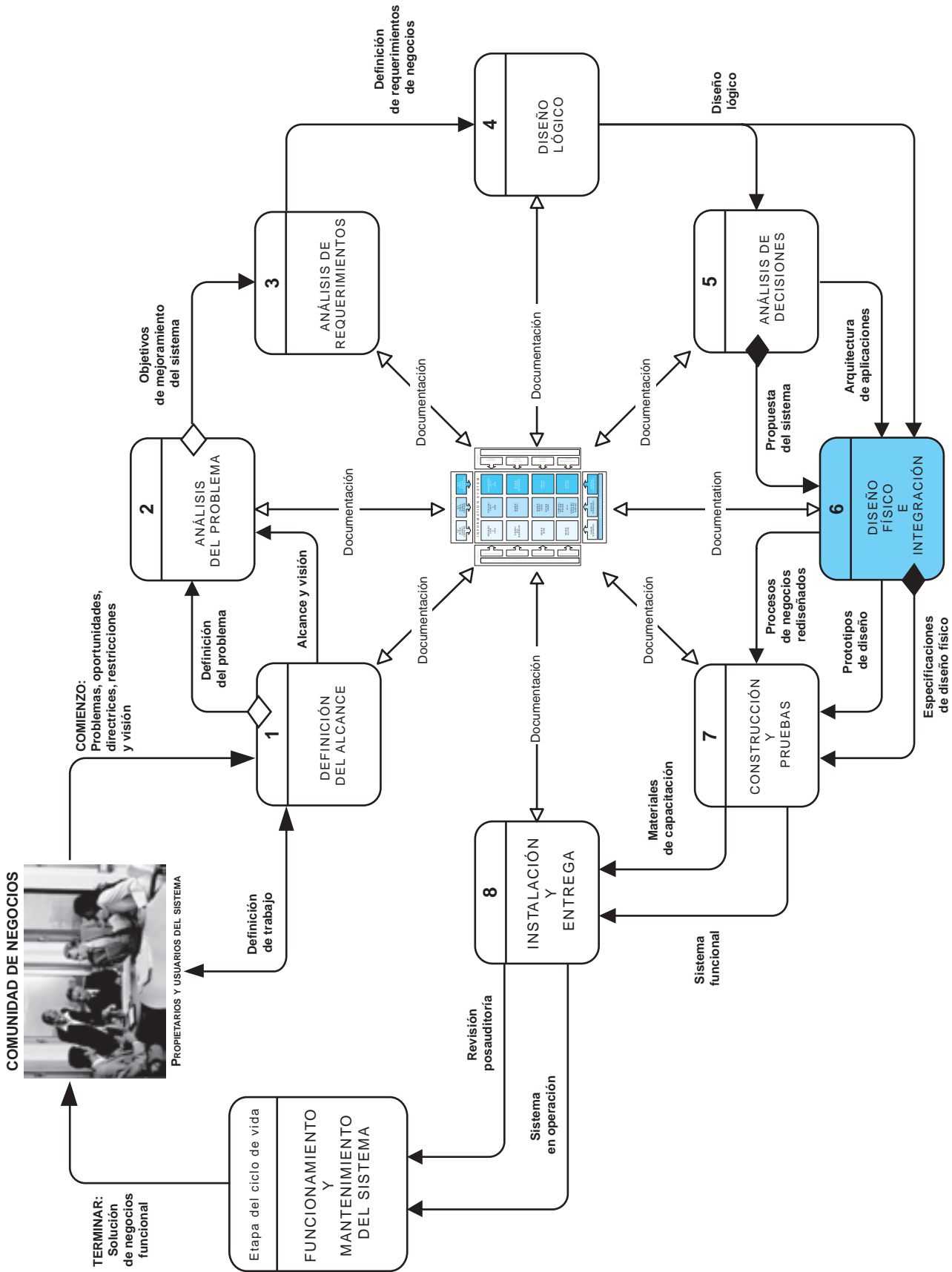
A continuación, examinemos con detalle cada tarea del diseño de sistemas.

### > Tarea 5.1: Diseño de la arquitectura de la aplicación

El propósito de la primera tarea de diseño es especificar la arquitectura de aplicaciones. La **arquitectura de la aplicación** define las tecnologías que se usarán en uno, más o todos los sistemas de información (y que se aplicarán en su construcción) con base en sus datos, procesos, interfaces y componentes de redes. Así pues, diseñar la arquitectura de la aplicación requiere considerar las tecnologías de red y la toma de decisiones sobre cómo se distribuirán los DATOS, PROCESOS e INTERFACES del sistema entre las ubicaciones del negocio.

Esa tarea se logra mediante el análisis de los modelos de datos y procesos creados inicialmente durante el análisis de requerimientos. Dados los modelos de datos y de procesos, así como la solución prevista, es necesario tomar decisiones de distribución. Hay que documentar las decisiones sobre datos, procesos e interfaces a medida que ocurran. Un

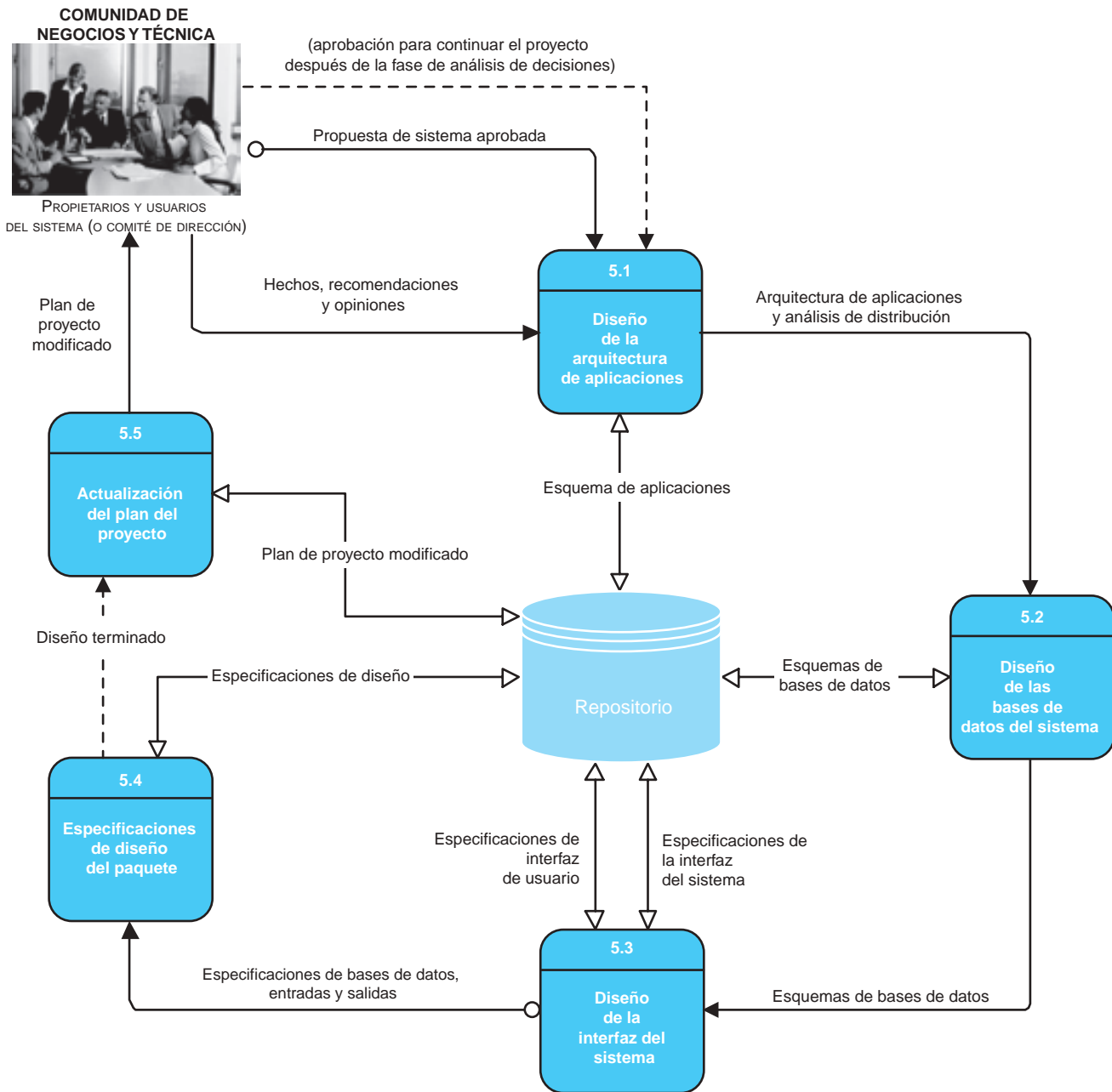
**arquitectura de la aplicación** Especificación de las tecnologías que se usarán para implantar los sistemas de información.



**FIGURA 10.5** Contexto de sistemas del desarrollo en la organización







**FIGURA 10.6** Tareas de diseño de sistemas del desarrollo en la organización

ejemplo es el **diagrama de flujo físico de datos (PDFD)**, por sus siglas en inglés) que se usa para establecer los procesos físicos y los repositorios de datos (bases de datos) en una red (figura 10.7). En el capítulo 11, se estudia el uso de los PDFD para la documentación de la arquitectura de aplicaciones.

A fin de realizar esta actividad, el analista puede necesitar la participación de un cierto número de **DISEÑADORES DE SISTEMAS** y **USUARIOS DE SISTEMAS**. Los usuarios podrían involucrarse en dicha actividad para ayudar a la solución en los problemas de datos, procesos y comunicación del negocio. Varios **DISEÑADORES DE SISTEMAS** podrían ser fundamentales en la realización de esta actividad, lo que incluye a un *administrador de bases de datos*, *administradores e ingenieros de redes*, un *administrador de aplicaciones* y otros expertos, en la medida necesaria (por ejemplo, un experto en la captura automática de datos para resolver los problemas y tecnología relacionados con los códigos de barras).

**diagrama de flujo físico de datos** Modelo de procesos usado para comunicar las características técnicas de implantación de un sistema de información.

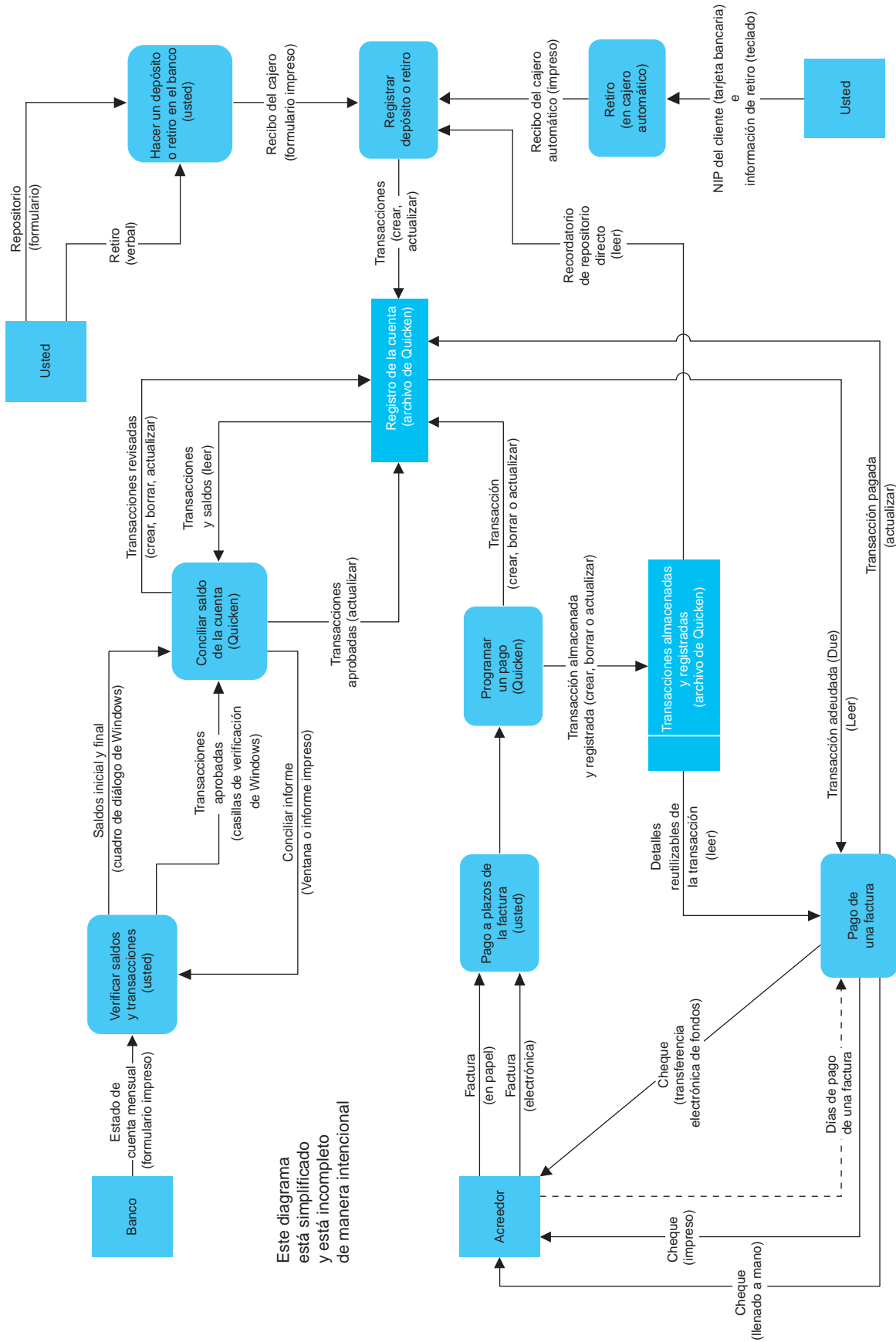


FIGURA 10.7 Ejemplo de diagrama de flujo físico de datos

Las entradas clave de esta tarea son los hechos, recomendaciones y opiniones que se solicitan a diversas fuentes y la propuesta de sistema aprobada de la fase de análisis de decisiones. El principal producto de la tarea es el análisis de distribución y arquitectura de aplicaciones, que sirve como referencia para actividades posteriores de la fase de diseño detallado.

### > Tarea 5.2: Diseño de las bases de datos del sistema

Es común que la siguiente tarea en el diseño del sistema sea desarrollar las especificaciones de diseño de las bases de datos correspondientes. El diseño de los datos va mucho más allá de la simple distribución de registros. Las bases de datos son un recurso compartido. Es habitual que muchos programadores las utilicen. Programas futuros podrían recurrir a bases de datos de maneras no previstas originalmente. Por ende, el diseñador debe prestar atención especial a diseñar bases de datos que sean adaptables a requerimientos y expansiones futuras.

El diseñador también debe analizar la forma en que los programas tendrán acceso a los datos, para mejorar su funcionamiento. El lector seguramente ya está familiarizado con diversas estructuras de datos de programación y su efecto en el rendimiento y flexibilidad. Estos problemas afectan las decisiones de organización de las bases de datos. Otras dificultades que deben atenderse durante el diseño de tales bases son el tamaño de los registros y los requerimientos de volumen de almacenamiento. Por último, las bases de datos son recursos compartidos, de modo que el diseñador también debe crear controles internos que garanticen la seguridad y las técnicas de recuperación de desastres apropiadas, por si acaso se pierden o destruyen datos.

El objetivo de esta tarea es preparar las especificaciones de diseño técnico para una base de datos que sea adaptable a requerimientos y expansiones futuras. Mientras que los ANALISTAS DE SISTEMAS participantes en el modelado de bases de datos facilitan esta tarea, los DISEÑADORES DE SISTEMAS son responsables de que se complete esta actividad. El *administrador de datos* podría participar en el diseño de la base de datos (o completarlo). Deben tenerse en cuenta las altísimas probabilidades de que el nuevo sistema aproveche una parte de la base de datos existente. Es aquí donde resulta crucial el conocimiento del administrador de bases de datos. Por último, los CONSTRUCTORES DEL SISTEMA también participan cuando se les pide elaborar un prototipo de bases de datos para el proyecto.

Como se muestra en la figura 10.6, una entrada clave en esta actividad corresponde a las decisiones de análisis de distribución y de arquitectura de aplicaciones de la tarea de diseño previa. El producto de esta tarea incluye los esquemas de bases de datos resultantes. Un ejemplo de tales esquemas se muestra en páginas previas de estos capítulos (figura 10.2). Un *esquema de base de datos* es el modelo estructural de una base de datos. Se trata de una imagen o mapa de los registros y relaciones que se pondrán en ejecución con la bases de datos misma. El desarrollo de esquemas de bases de datos es tema del capítulo 12.

### > Tarea 5.3: Diseño de la interfaz del sistema

Una vez diseñada la base de datos y, posiblemente, construido el prototipo, el diseñador de sistemas puede colaborar estrechamente con los usuarios del sistema en el desarrollo de las especificaciones de entradas, salidas y diálogos. Los usuarios finales y administradores tendrán que trabajar con las entradas y salidas, por lo que los diseñadores deben tener cuidado de solicitar sus ideas y sugerencias, particularmente en relación con el formato. Asimismo, deben buscar esas ideas y opiniones acerca de diálogos de fácil aprendizaje y uso para el nuevo sistema.

Las salidas de transacciones frecuentemente se diseñan como formularios preimpresos, en los que pueden imprimirse los detalles de las transacciones. Los informes y otras salidas se pueden imprimir directamente en papel o mostrar en la pantalla de la terminal. El formato y distribución precisos de las salidas se deben especificar. Por último, también son necesarias las especificaciones de los controles internos, para garantizar que no se pierdan, desvíen, usen incorrectamente o sean incompletas las salidas. La figura 10.8 contiene un ejemplo de diseño de salidas. Tal diseño es tema del capítulo 13.

En cuanto a las entradas, es crucial diseñar el método de captura de datos que se usará. Por ejemplo, podría diseñar un formulario en el que los datos a introducirse en el

**FIGURA 10.8**

Ejemplo de prototipo de pantalla de salida

QuickReport version 1.0  
Page 1 of 1

## Member Response to Video Title Selection of the Month

Page number 1  
Printed at 6/14/96 7:32:11 AM

Category	Potential Orders	Selection of the Month	Alternate Selection	Selection of Month + Alternatives	Number of Orders
Action Adventure	6342	2410	824	241	2867
Animated	3577	1538	644	154	1241
Comedy	954	181	38	18	716
Documentary	1486	877	45	88	477
Drama	540	389	54	39	58
Western	104	9	54	1	40
Horror	920	99	23	409	2501
Musical	209	40	78	289	103
Science Fiction	4590	2011	899	2200	5329
Sports	288	288	277	121	387
Sum of Potential Orders		19810			
Sum of Selection of the Month		7842			
Sum of Alternate Selections		2936			
Sum of Selection of the Month + Alternatives		3560			
Total Number of Orders		13779			

sistema sean inicialmente registrados. Seguramente a usted le agradecería facilitar el registro de los datos en el formulario; pero también le gustaría simplificar la entrada de los mismos del formato a la computadora o a un medio que pueda leerse a través de ella. Esto tomará una importancia especial si la introducción de datos está en manos de personas no familiarizadas con la aplicación del negocio. Así mismo, es posible que se cometan errores cada vez que se introducen datos al sistema. Por ello, se requerirá definir controles de edición que garanticen la exactitud de los datos de entrada. Un ejemplo del prototipo de pantalla de entrada de datos se ilustra con anticipación en este capítulo (figura 10.3). El diseño de entradas se estudia en el capítulo 14.

En cuanto al diseño de la interfaz o diálogos, hay que considerar factores como la familiaridad con la terminal, posibles errores y malentendidos del usuario final, necesidad de instrucciones adicionales o ayuda en ciertos puntos y contenido, así como diseño de la pantalla. Lo que se pretende es anticiparse a cada posible error o pulsión de tecla por parte del usuario, sin importar cuán improbables sean. Adicionalmente, se intenta facilitar al usuario final la comprensión de que la pantalla se desplegará en cualquier momento dado. La figura 10.9 es un ejemplo de diseño de interfaz. Este último es tema del capítulo del 15.

¡LOS USUARIOS DEL SISTEMA deben participar en esta actividad! Son ellos quienes trabajarán con las entradas, salidas y diálogos de interfaz. El grado de su participación debe incrementarse en las actividades de diseño que incluyan prototipos. Se les pide retroalimentación concerniente a cada prototipo de entradas/salidas. LOS DISEÑADORES DE SISTEMAS son responsables de que se complete esta actividad. Conviene aprovechar la experiencia de los diseñadores de sistemas especializados en el diseño de *interfaces gráficas de usuario*. Adicionalmente, los CONSTRUCTORES DE SISTEMAS pueden preparar varios diseños de pantallas para que los usuarios los revisen durante el diseño con prototipos.

Como se muestra en la figura 10.6, las entradas clave de esta actividad corresponden al esquema de bases de datos de la tarea previa y a las especificaciones de interfaz de sistema y usuario disponibles en el repositorio del proyecto. El producto de la tarea de diseño consiste en las especificaciones terminadas de bases de datos, entradas y salidas.

**FIGURA 10.9**

Ejemplo de prototipo de pantalla de interfaz de diálogo

### > Tarea 5.4: Especificaciones de diseño del paquete

Esta tarea final de diseño involucra la integración de todas las especificaciones de las tareas de diseño previas en un conjunto de especificaciones que sirva de guía para las actividades del programador durante la fase de construcción de la metodología de desarrollo de sistemas.

Sin embargo, esta tarea es mucho mayor que la de formar un paquete. El qué tan mayor sea depende de dos factores: 1) dónde se marca la línea entre las responsabilidades del diseñador y el programador del sistema, y 2) si el método y solución requiere del diseño de la estructura global del sistema o no. Muchas organizaciones han adoptado estrategias de desarrollo acelerado de sistemas que no requieren el segundo de estos factores. La estructura de programas se encargaba de aspectos de calidad que preocupaban a los desarrolladores de sistemas que usaban lenguajes de programación antiguos y tendían a ser aplicaciones basadas en las *mainframe*.

El ANALISTA DE SISTEMAS, que puede tener el apoyo de DISEÑADORES DE SISTEMAS, usualmente se encarga de esta tarea. Antes de proceder a la integración de las especificaciones de diseño y a la fase de construcción, debe realizarse el diseño del sistema con participación de todos los interesados. Si bien los USUARIOS DEL SISTEMA ya han visto y aprobado las entradas, salidas y diálogos del nuevo sistema, se debe emprender la revisión final y la aprobación del flujo global de trabajo y datos del sistema. Los PROPIETARIOS DEL SISTEMA han de contar con una oportunidad final para cuestionar la factibilidad del proyecto y determinar si es necesario ajustarlo, detenerlo o aprobar que se proceda a su construcción. El *personal de auditoría* de la empresa suele participar intensamente en esta etapa del proyecto, aportando su juicio sobre los controles internos del nuevo sistema.

Como se detalla en la figura 10.6, las entradas de esta tarea son las diversas especificaciones de bases de datos, entradas y salidas que se crearon con anterioridad. Una vez realizadas, aprobadas y organizadas como especificaciones de diseño adecuadas para construir el nuevo sistema, se pone a disposición del equipo de constructores del sistema, mediante el repositorio del proyecto. Que un administrador del proyecto ponga disponibles las especificaciones de diseño a través de un repositorio compartido es más frecuente que proporcionar a cada desarrollador una copia de un conjunto impreso de especificaciones organizadas.

## > Tarea 5.5: Actualización del plan del proyecto

Ahora que nos acercamos al final de la fase de diseño, es necesario que reevaluemos la factibilidad del proyecto y *actualicemos así mismo el plan del proyecto*. El *administrador del proyecto*, junto con los PROPIETARIOS DEL SISTEMA y el equipo del proyecto en su totalidad, facilita esta tarea. LOS ANALISTAS Y PROPIETARIOS DEL SISTEMA son clave en ella. Deben considerar la posibilidad, con base en el trabajo de diseño terminado, de que puedan requerirse ajustes al calendario global del proyecto y a las estimaciones de costos, entre otras.

Como se ilustra en la figura 10.6, esta tarea se inicia cuando el administrador del proyecto determina que se completó el diseño. El producto clave obtenido de esta tarea es el plan del proyecto actualizado. Ahora, este último debe incluir un plan detallado de la fase de construcción o desarrollo del sistema que deberá seguirse.

## Diseño de sistemas para integrar software comercial: La solución de "compra"

A continuación, examinemos el diseño de sistemas con soluciones que incluyan adquirir productos de software comercial listos para usarse (COTS, por sus siglas en inglés). El ciclo de vida de proyectos que abarca la solución de adquisición o compra se presenta en la figura 10.10. Observe que la definición de requerimientos del negocio (de software) y su integración como solución del mismo generan una sucesión de fases que están ausentes en el proceso de desarrollo de sistemas dentro de la organización, el cuál ya fue estudiado con anterioridad. Las diferencias más significativas entre los proyectos de sistemas que se compran y los que se desarrollan en la organización es la inclusión de una nueva *fase de compra* en los primeros, y otra de análisis de decisiones especiales (proceso indicado como "5A") en relación con el software y servicios en los segundos.

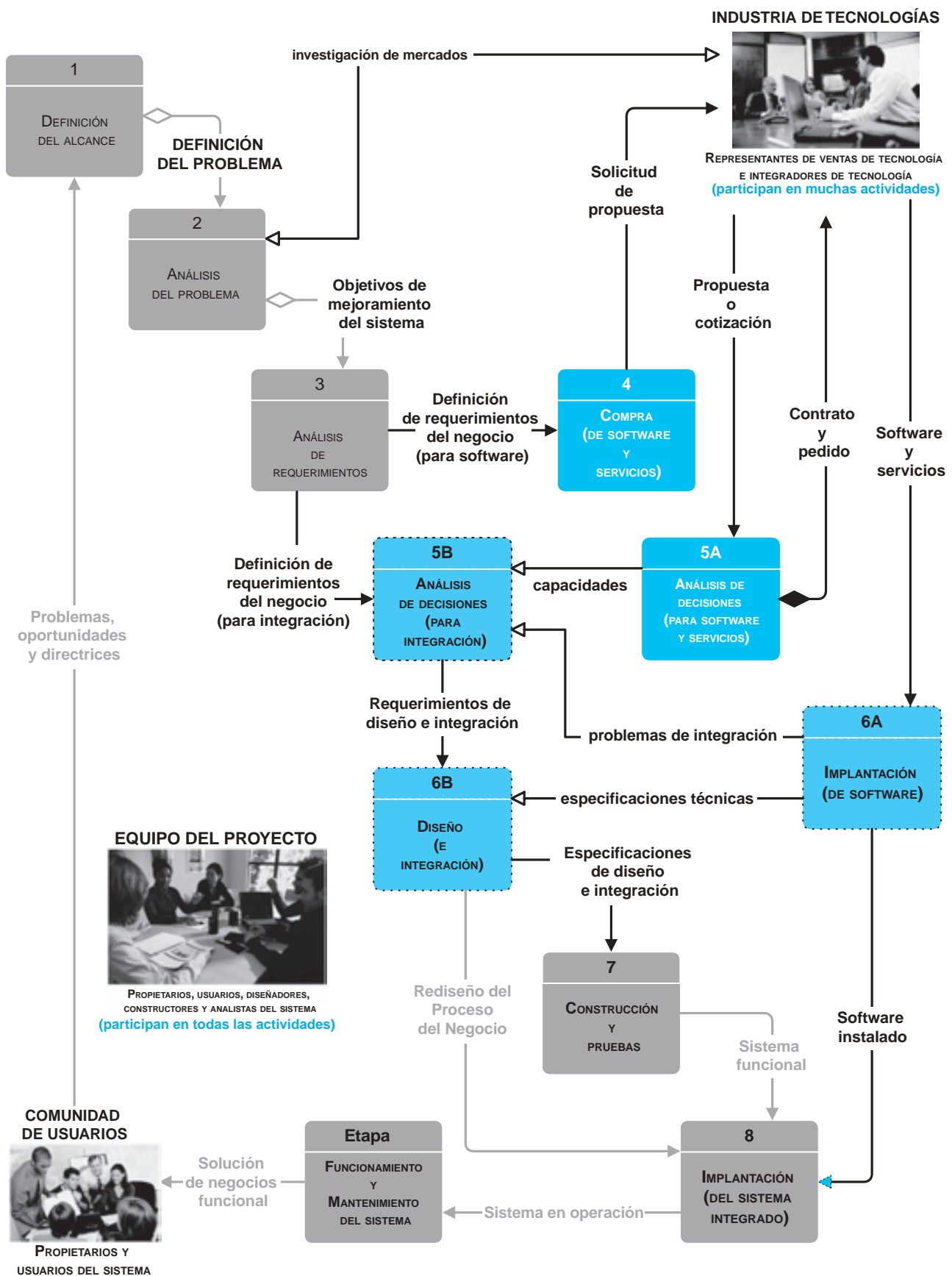
Cuando se requiere software nuevo, la elección de los productos apropiados suele dificultarse. Las decisiones se complican por factores técnicos, económicos y políticos. Una decisión errónea puede estropear el análisis y el diseño por exitosos que sean. El analista de sistemas participa cada vez más en la compra de paquetes de software (así como de periféricos y computadoras para soporte a las aplicaciones específicas que desarrolla el propio analista). Los propósitos de las fases de compra y análisis de decisiones son los siguientes:

1. Identificar e investigar productos específicos que puedan soportar la solución recomendada para el sistema de información requerido.
2. Solicitar, evaluar y jerarquizar las propuestas de proveedores.
3. Elegir y recomendar la mejor propuesta de los proveedores.
4. Firmar el contrato con el vendedor elegido para obtener el producto.

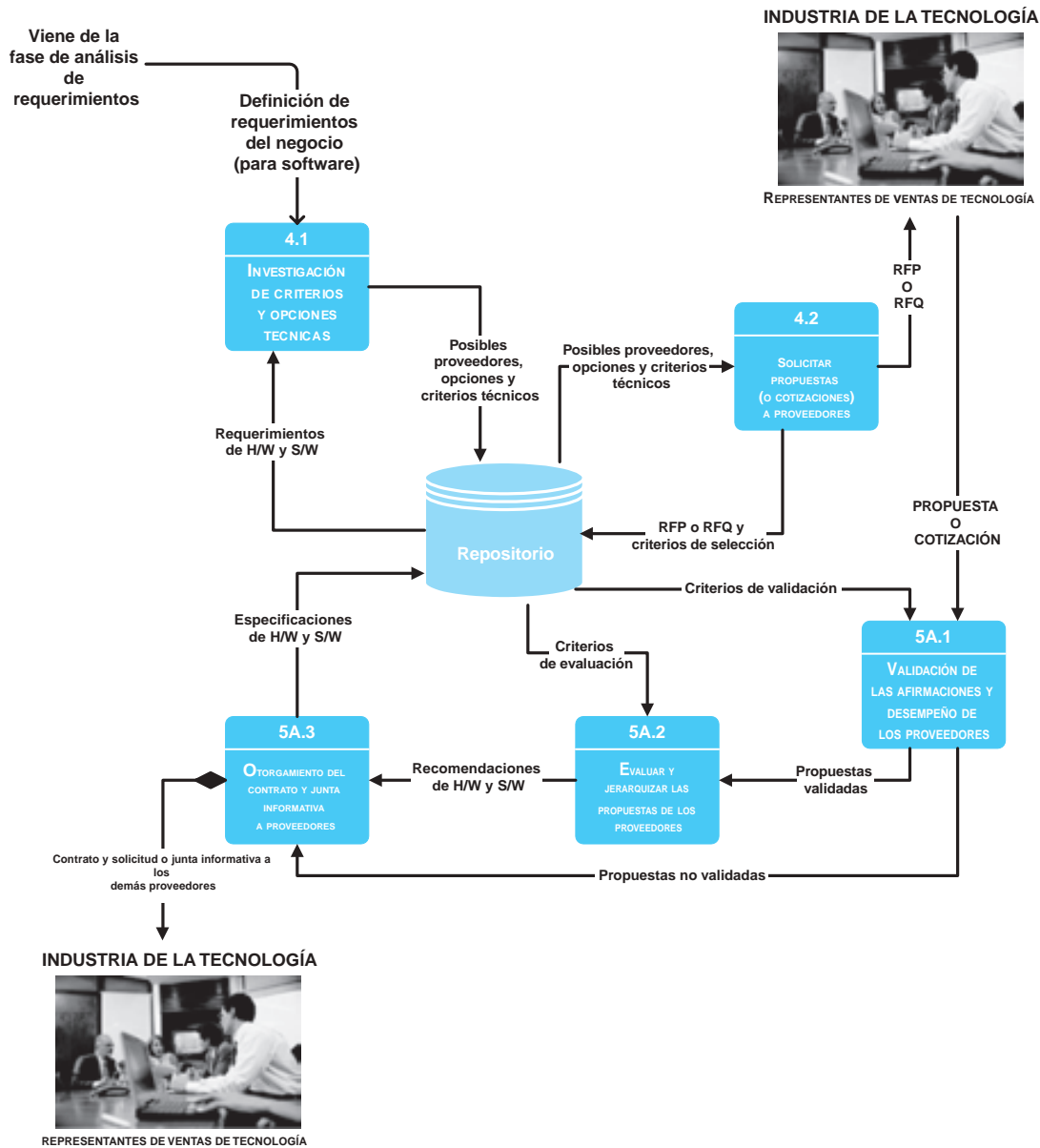
En este apartado analizaremos las tareas relacionadas con la ejecución de las fases de suministro y análisis de decisiones para una solución de compra. Como se indica en la figura 10.10, este tipo de solución afecta la manera de ejecutar otras fases del ciclo de vida (las fases afectadas se muestran en recuadros punteados). Una vez examinadas las fases de análisis de decisiones y compra, se investiga el impacto que una solución de compra tendría y cómo deberán completarse esas fases.

La figura 10.11 es un diagrama de tareas que muestra el trabajo (tareas) necesario para completar las fases de análisis de decisiones de una solución de proyecto de compra. Este diagrama no convierte en obligatorio el uso de algún método específico; sin embargo, se describen en los párrafos siguientes las estrategias, herramientas y técnicas que el lector podría considerar para cada tarea de diseño. El diagrama es sólo una plantilla. El equipo y el administrador del proyecto lo pueden ampliar o modificar para que refleje las necesidades específicas de un proyecto dado.

Las primeras dos tareas (4.1 y 4.2) son de la fase de compra, mientras que las demás (5A.1, 5A.2 y 5A.3) se relacionan con el análisis de decisiones. A continuación, se estudia con detalles cada tarea.



**FIGURA 10.10** Contexto de diseño de sistemas de una solución de software comercial listo para usarse



**FIGURA 10.11** Tareas de la fase de compra

**> Tarea 4.1: Investigación de criterios y opciones técnicas**

La primera tarea consiste en investigar las opciones técnicas. En ella, se identifican especificaciones de importancia en relación con el software o el hardware que se elegirá. Esta tarea se enfoca en los requerimientos de software o hardware que se establecen en la fase de análisis de requerimientos. Tales requerimientos detallan la funcionalidad, las características y los parámetros de desempeño críticos del nuevo software/hardware.

Muchos analistas consultan revistas y otras publicaciones periódicas apropiadas, además de Internet, como ayuda para identificar las especificaciones y problemas técnicos y de negocios que revisten importancia para la decisión de la elección. Otras fuentes de información para realizar tal investigación son:

- Es factible que haya *estándares o normas internas* para la elección de hardware y software. Algunas compañías especifican que cierta tecnología se adquiera de ciertos proveedores, si la ofrecen.



Por ejemplo, hay organizaciones que estandarizan la adquisición de marcas específicas de microcomputadoras, terminales, impresoras, sistemas de administración de bases de datos, software de comunicación de datos, hojas de cálculo y lenguajes de programación. Un poco de trabajo en este paso puede ahorrar mucha investigación innecesaria.

- Los *servicios de información* tienen como fin principal evaluar constantemente el mercado en busca de nuevos productos y aconsejar a los compradores potenciales sobre las especificaciones que han de considerar. También aportan información concerniente al número de instalaciones y la satisfacción general de los clientes con los productos.
- Los *periódicos y otras publicaciones periódicas del gremio* contienen artículos y experiencias con diversos tipos de software y hardware que valdría la pena considerar. En muchos casos, se cuenta con ellos en las bibliotecas de la facultad o compañía. También están disponibles las suscripciones (a veces, gratuitas).

Con la investigación, también deben identificarse posibles proveedores de los productos que se consideran. Después de que los analistas han completado su trabajo, se ponen en contacto inicial con esos proveedores. De esta manera, ¡los analistas están listos para enfrentar adecuadamente los discursos de ventas después de realizar esta investigación!

El propósito de esta tarea es estudiar alternativas técnicas para especificar criterios y opciones de importancia en relación con el nuevo hardware o software que se adquirirá. Esta tarea se facilita gracias al *administrador del proyecto*. Los DISEÑADORES DE SISTEMAS tienen la responsabilidad de llevar a cabo esta tarea. El diseñador podría buscar información con diversos expertos técnicos, como los administradores de datos, bases de datos, redes y aplicaciones.

Como se ilustra en la figura 10.11, una entrada clave en esta tarea es la definición de requerimientos del negocio (del software) establecida en la fase de análisis de requerimientos. El diseñador también indaga hechos adicionales sobre productos y proveedores en diversas fuentes. Los diseñadores deben tener cuidado de no obtener información únicamente del vendedor; y no porque los representantes de ventas no sean sinceros, sino porque la regla principal en ventas es poner énfasis en las ventajas del producto y desestimar sus desventajas. El principal producto obtenido de esta tarea incluye una lista de posibles proveedores, opciones de productos y criterios técnicos.

A fin de completar esta tarea, los diseñadores deben realizar una investigación amplia para averiguar hechos importantes acerca del producto de hardware/software y el proveedor. Asimismo, deben tener cuidado de corroborar sus diversas fuentes. Estas últimas se usan para identificar a posibles proveedores de los productos que interesan. La tarea que nos ocupa podría ser opcional si la compañía tiene un compromiso o contrato de adquisición de ciertos productos con un proveedor específico. Por último, El diseñador debe revisar los resultados concernientes al producto, proveedor y fabricante.

## > Tarea 4.2: Solicitar propuestas o cotizaciones a los proveedores

La tarea siguiente consiste en solicitar propuestas o cotizaciones a los proveedores. Si la organización tiene un compromiso de compra con una sola fuente (por ejemplo, IBM), la tarea es más bien informal. Basta ponerse en contacto con el proveedor y solicitar cotizaciones de precios y condiciones. Sin embargo, muchas decisiones comprenden numerosas alternativas. En esta situación, el sentido común de negocios indica sacar ventaja de la competencia existente en el mercado.

La tarea de solicitud de propuestas requiere preparar uno de dos documentos: una **solicitud de cotizaciones (RFQ)**, por sus siglas en inglés) o una **solicitud de propuestas (RFP)**, por sus siglas en inglés). La solicitud de cotizaciones se usa cuando ya se ha tomado una decisión sobre el producto específico, sin embargo éste se puede adquirir con varios proveedores. Su finalidad principal es obtener datos específicos de configuraciones, precios, convenios de mantenimiento, condiciones relativas a cambios que realicen los compradores, y el servicio. La solicitud de propuestas se utiliza cuando diversos proveedores o productos son posibles candidatos y nos interesa obtener propuestas y cotizaciones competidoras. Las RFP pueden considerarse como un súper conjunto de las RFQ. En ambos tipos de solicitudes se definen criterios de selección que se usan en una validación posterior.

---

### solicitud de cotización (RFQ)

Documento formal en el que se comunican los requerimientos técnicos y de soporte del negocio, de un paquete de software de aplicación a un solo proveedor, al que se le ha considerado capaz de proporcionar el paquete de aplicaciones y servicios solicitados.

---

### solicitud de propuesta (RFP)

Documento formal en el que se comunican los requerimientos técnicos y de soporte del negocio, para un paquete de software de aplicaciones a los proveedores que desean competir por la venta de ese paquete de aplicaciones y servicios.

El objetivo principal de las RFP es comunicar los requerimientos y características deseadas a los posibles proveedores. Esos requerimientos y características se deben clasificar como obligatorios (que debe ofrecer el proveedor), sumamente importantes (que se desea obtener del proveedor, si bien pueden obtenerse en la organización o de un tercero) o deseables (que son prescindibles). Los requerimientos también podrían categorizarse según dos criterios alternos: los que satisfacen las necesidades de los sistemas y los que llenan las necesidades del comprador (por ejemplo, servicio del proveedor).

El *administrador del proyecto* facilita esta tarea. El DISEÑADOR DEL SISTEMA también tiene la responsabilidad de completar esta actividad y puede buscar el apoyo de los *administradores de datos, bases de datos, redes y aplicaciones* cuando redacta las RFP o RFQ.

La entrada clave de esta tarea consiste en los posibles proveedores, opciones y criterios técnicos que resultan de la investigación previa. El producto obtenido más importante de esta tarea es la RFP o RFQ resultante, que reciben los candidatos a proveedores. La calidad de una RFP tiene efecto significativo en la calidad e integridad de las propuestas que se deriven. Un esquema propuesto de RFP se presenta en la figura 10.12, ya que la RFP en sí es demasiado amplia para incluirla en este libro.

## FIGURA 10.12 Solicitud de propuestas

### Solicitud de propuestas (RFP)

- I. Introducción
  - A. Antecedentes
  - B. Resumen breve de necesidades
  - C. Explicación de la RFP
  - D. Solicitud de acción al proveedor
- II. Normas e instrucciones
  - A. Calendario de eventos que llevan al contrato
  - B. Reglas básicas que regirán la decisión de la selección
    1. Quién hablará con quién y cuándo
    2. Quién paga qué
    3. Formato requerido para la propuesta
    4. Expectativas de demostración
    5. Expectativas contractuales
    6. Referencias esperadas
    7. Expectativas de documentación
- III. Requerimientos y características
  - A. Hardware
    1. Requerimientos, características y criterios obligatorios
    2. Requerimientos, características y criterios esenciales
    3. Requerimientos, características y criterios deseables
  - B. Software
    1. Requerimientos, características y criterios obligatorios
    2. Requerimientos, características y criterios esenciales
    3. Requerimientos, características y criterios deseables
  - C. Servicio
    1. Requerimientos obligatorios
    2. Requerimientos esenciales
    3. Requerimientos deseables
- IV. Cuestionarios técnicos
- V. Conclusión

Muchas de las habilidades que se adquieren con el estudio de la segunda parte de la obra, como el modelado de datos y procesos, pueden ser muy útiles para comunicar los requerimientos en la RFP. Los proveedores son muy receptivos a estas herramientas, ya que resulta más fácil encontrar correspondencia de los productos y opciones, para generar una propuesta dirigida a las necesidades del cliente. Otra habilidad de importancia es la redacción de informes (tema del capítulo 9) y cuestionarios (del capítulo 5).

### > Tarea 5A.1: Validación de las afirmaciones y desempeño de los proveedores

Las propuestas o cotizaciones se empiezan a recibir poco después de enviar las RFP o RFQ a los posibles proveedores. La validez de las propuestas no se puede ni se debe dar por sentada, sino que deben validarse las afirmaciones y el desempeño de los proveedores. Esta tarea se realiza de manera independiente para cada propuesta, sin que se comparen entre sí las propuestas.

El propósito de esta tarea es validar las solicitudes en relación con las propuestas o cotizaciones que se reciben de los proveedores. Los DISEÑADORES DE SISTEMAS SON responsables de llevar a cabo esta actividad. Una vez más, en tal validación el diseñador puede solicitar el apoyo de los *administradores de datos, bases de datos, redes y aplicaciones*.

Esta tarea se genera con la recepción de propuestas o cotizaciones de los posibles proveedores. Las salidas clave de la tarea son las propuestas validadas de proveedores y otras cuyas afirmaciones no se validaron.

A fin de completar esta tarea, el diseñador debe recopilar y revisar todos los hechos pertinentes a los requerimientos y características del producto. Debe revisar las propuestas de los proveedores y eliminar todas las que no satisfagan los requerimientos obligatorios. Si se especificaron claramente los requerimientos, ningún proveedor debe haber enviado una propuesta que tenga que eliminarse. En el caso de las que no cumplan uno o más de los requerimientos muy importantes, se verifica que sea posible satisfacer esos requerimientos o características por otros medios. En relación con cada propuesta no eliminada, el diseñador debe validar las afirmaciones y promesas del proveedor contra criterios de validación. Las afirmaciones relativas a las características y requerimientos obligatorios, muy importantes y deseables se pueden validar mediante cuestionarios llenados y listas de comprobación (incluidos en la RFP) con referencias apropiadas, que proporciona el proveedor, con otros usuarios y manuales técnicos. Las promesas se validan sólo al garantizar que estén por escrito en el contrato. Por último, el desempeño se valida óptimamente con una demostración, que es de importancia especial cuando se evalúan paquetes de software. Las demostraciones permiten obtener resultados de pruebas y datos que confirman capacidades, características y facilidad de uso.

### > Tarea 5A.2: Evaluar y jerarquizar las propuestas de los proveedores

Ahora es posible evaluar y jerarquizar las propuestas validadas. En realidad, se trata de dos pasos que constituyen otro análisis de costo-beneficio como el realizado durante el desarrollo de sistemas. Los criterios de evaluación y calificación deben establecerse antes de que ocurra la evaluación, de modo que no se incluya un sesgo de los criterios y calificaciones para favorecer inconscientemente una propuesta dada.

El patrocinador ejecutivo debe facilitar, en teoría, esta actividad. El diseñador puede hacer que participen varios expertos en la evaluación y jerarquización de las propuestas, una vez más con la posible inclusión de los *administradores de datos, bases de datos, redes y aplicaciones*.

Las entradas de esta tarea incluyen las propuestas validadas y criterios de evaluación usados para jerarquizar las propuestas. El producto clave de esta tarea consiste en las recomendaciones de hardware o software.

La capacidad para realizar una evaluación de factibilidad es una habilidad muy importante para completar esta tarea. Las técnicas y habilidades de evaluaciones de factibilidad son tema del capítulo 9. A fin de llevar a cabo esta tarea, los diseñadores en primer término deben recopilar y revisar todos los detalles concernientes a las propuestas validadas. Posteriormente, han de establecer los criterios de evaluación y el sistema de calificación. Son muchas las formas de hacerlo. Algunos métodos incluirán la ponderación de los re-

querimientos en una escala de puntos. ¡Las mejores estrategias son las de tipo monetario! Una evaluación monetaria es más fácil de defender ante los administradores que otra de puntos. Una estrategia de este tipo consiste en evaluar las propuestas con base en importes monetarios “duros” y “blandos”. Los costos “duros” son los que tienen que pagarse al proveedor elegido por el equipo o software. Se trata de costos adicionales en que se incurre con la elección de un proveedor específico (por ejemplo, si se opta por el proveedor A, se incurrirá en un gasto adicional, si se compara con el proveedor B, para superar una deficiencia del sistema que propone el primero de ellos). Esta estrategia otorga el contrato al proveedor que satisface todos los requerimientos fundamentales y al mismo tiempo ofrece el costo “duro” más bajo, además de penalizaciones “blandas” por no proporcionar las características deseadas (véase una explicación detallada de este método en los escritos de Isshiki de 1982 y Joslin de 1977 incluidos en las lecturas recomendadas del capítulo). Una vez establecidos los criterios de evaluación y el sistema de calificación, el último paso para completar la tarea es llevar a cabo la evaluación y jerarquización de las propuestas de los proveedores.

### > Tarea 5A.3: Otorgamiento del contrato y junta informativa a proveedores

Una vez jerarquizadas las propuestas de los proveedores, la actividad siguiente suele incluir la presentación de recomendaciones a los administradores para su aprobación final. De nuevo, revisten importancia las habilidades de comunicación, en particular las de venta, si el analista pretende convencer a los administradores de que usen sus recomendaciones. Una vez que se cuente con la aprobación de las recomendaciones hechas a los administradores, lo siguiente es redactar el contrato y otorgarlo al proveedor elegido. Esta actividad también suele incluir una sesión de informativa a los proveedores que no fueron seleccionados, en la que deberá tenerse cuidado de no romper las relaciones que se tienen con ellos.

El propósito de esta actividad es negociar un contrato con el proveedor que presentó la propuesta ganadora e informar a los proveedores cuyas propuestas no se eligieron. En teoría, el patrocinador ejecutivo que debe aprobar las recomendaciones y la cancelación del proyecto también ha de facilitar esta actividad. Sin embargo, es el DISEÑADOR DEL SISTEMA quien tiene que elaborar y defender la recomendación, así como otorgar el contrato. En dicha actividad, el diseñador puede solicitar el apoyo de un abogado de la organización para redactar el contrato. Las habilidades de redacción de informes y de presentación son importantes en esta tarea.

Las entradas clave incluyen la recomendación del hardware y software, así como las propuestas no validadas de las tareas de evaluación previas. En caso de que se cuente con la aprobación del patrocinador ejecutivo, se redactará posteriormente un contrato para el proveedor elegido. También se celebrará una junta de información sobre las propuestas a los proveedores no seleccionados.

A efecto de realizar esta tarea, el diseñador debe presentar primero una recomendación del hardware y software para su aprobación final. Una vez tomada la decisión de aprobación, hay que negociar el contrato con el proveedor seleccionado. Ciertas condiciones y términos especiales tendrían que ponerse por escrito en el contrato y pedido estandarizados. En teoría, ningún contrato de cómputo debe firmarse sin la asesoría de un abogado. El analista debe tener cuidado de leer y aclarar todos los acuerdos de licencia. Ninguna decisión final debe aprobarse sin el consentimiento de un contador o administrador calificados. La compra, arrendamiento y arrendamiento con opción a compra entrañan consideraciones fiscales complejas. Finalmente, por cortesía y para mantener las buenas relaciones, se celebra una junta informativa sobre las propuestas con los vendedores no seleccionados. El propósito no es brindarles una segunda oportunidad para obtener el contrato, sino más bien informarles sobre las debilidades precisas de sus propuestas o productos.

### > Impacto de la decisión de compra en las fases restantes del ciclo de vida de sistemas

No basta simplemente con comprar o construir (desarrollar) sistemas que satisfagan los requerimientos del sistema deseado. El análisis debe integrar o comunicar el nuevo sis-

tema con los numerosos sistemas existentes que son esenciales para la empresa. Muchos de estos sistemas incluyen el uso de tecnologías, técnicas y estructuras de archivos con diferencias impresionantes entre ellos.

El analista debe considerar la forma en que el sistema deseado encaja en el conjunto de sistemas del que forma parte. Los requerimientos de integración especificados son vitales para garantizar que el nuevo sistema funcione en armonía con los demás.

Como se muestra en la figura 10.10, la decisión de adquirir una solución de paquete de software comercial puede tener efecto en las fases adicionales (indicadas con recuadros punteados) del ciclo de vida de sistemas. Una vez terminada la fase de análisis de decisiones (relativas a software y servicios) y su evaluación intensiva del producto comercial, se tienen conocimientos acerca de sus características (o deficiencias). Es durante la fase de análisis de decisiones para la integración que es preciso hacer revisiones para reflejar estos nuevos conocimientos en los modelos de datos y procesos que comprende la definición de requerimientos del negocio. Cuando se recibe el software y los servicios del proveedor o proveedores, hay que ponerlos en práctica. Durante esta implantación, pueden surgir problemas de integración que también deben reflejarse en la definición de requerimientos del negocio. Estas capacidades y problemas de integración se incluyen en los requerimientos de diseño e integración.

Por último, una vez que se cuenta con los requerimientos de diseño e integración, hay que completar la fase de diseño. Esto requiere de muchas de las mismas tareas que son tema de páginas anteriores de este capítulo. La diferencia principal radica en que no se está “desarrollando” simplemente un sistema completo. En su lugar, podrían estarse diseñando especificaciones técnicas para el desarrollo de un pequeño subconjunto de programas, utilerías de software y otros componentes que son necesarios para que los procesos del negocio y el productor de software comercial se integren y funcionen apropiadamente. Considere un ejemplo. El sistema empresarial existente podría usar tecnologías de código de barras para capturar los datos. Sin embargo, el producto de software podría requerir la introducción de datos a través del teclado. En este caso, tal vez sería necesaria la adaptación del producto de software para que permitiera la introducción de datos vía el teclado o por medio de un archivo de lotes que contuviera los datos escaneados.

Este capítulo brinda una panorámica detallada del diseño de sistemas de un proyecto. Ahora está usted listo para aprender algunas de las habilidades de diseño de sistemas que se resaltan en el presente capítulo. Tal diseño depende de los requerimientos que se especifican durante el análisis de sistemas, por lo que se recomienda leer primero los capítulos 4 a 9. En el primero de ellos se presenta una panorámica del análisis de sistema, mientras que en los capítulos 5 a 9 se describen diversas técnicas y herramientas de análisis de sistemas que proporcionan las entradas básicas de las actividades de diseño de sistemas que son tema de la tercera parte de esta obra.

El orden de lectura de los capítulos de diseño de sistemas que se sigue es flexible.

## Resumen

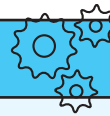


1. El diseño de sistemas de información se define formalmente como las tareas enfocadas en las especificaciones de una solución computarizada detallada. Mientras que en el análisis de sistemas se hace énfasis en el problema del negocio. En el diseño de sistemas se centra la atención en los aspectos técnicos o de implantación del sistema.
2. El diseño de sistemas se basa en las preocupaciones técnicas de los diseñadores del sistema. Por lo tanto, en lo referente a los componentes del sistema de información, tal diseño considera los componentes de dicho sistema desde la perspectiva de su diseñador.
3. El diseño de sistemas de los proyectos de desarrollo al interior de la organización o de “construcción” difiere de los de “compra”, en que en este último se adquiere un paquete de software de sistemas.
4. Son muchas las estrategias o técnicas para el diseño de sistemas. Se las puede utilizar combinadas.
  - a) Diseño estructurado moderno, técnica que se enfoca en los procesos.
  - b) Ingeniería de la información (IE, por sus siglas en inglés), que se centra en la planeación de datos y en la estrategia para generar proyectos de aplicaciones.
  - c) Elaboración de prototipos, que es un proceso iterativo que implica una relación de trabajo estrecho entre los diseñadores y usuarios para producir un modelo del nuevo sistema.
  - d) Desarrollo conjunto de aplicaciones (JAD, por sus siglas en inglés), que hace énfasis en el desarrollo con la participación de los propietarios, usuarios, diseñadores y constructores del sistema. Durante las sesiones de JAD para el diseño de sistemas, el diseñador del sistema asume la función de facilitador.
  - e) El desarrollo rápido de aplicaciones (RAD, por sus siglas en inglés) es una técnica que constituye una fusión de las diversas técnicas estructuradas con la elaboración de prototipos y el JAD para acelerar el desarrollo de los sistemas.
  - f) El diseño orientado a objetos (OOD, por sus siglas en inglés), nueva estrategia de diseño que da seguimiento al análisis orientado a objetos para refinar las definiciones de requerimientos de objetos y definir nuevos objetos específicos del diseño en cuestión.
5. En el caso de los proyectos de desarrollo en la organización (construcción), el diseño de sistemas incluye el desarrollo de especificaciones de diseño técnico que sirven de guía en la construcción y puesta en marcha del nuevo sistema. A fin de completar la fase de diseño, el diseñador del sistema debe realizar las tareas siguientes:
  - a) Diseñar la arquitectura de aplicaciones.
  - b) Diseñar la base o bases de datos del sistema.
  - c) Diseñar la interfaz del sistema.
  - d) Integrar (“conjuntar”) las especificaciones de diseño.
  - e) Actualizar el plan del proyecto.
6. El diseño de sistemas para soluciones que incluyen la adquisición de productos de software comerciales listos para usarse (COTS, por sus siglas en inglés) abarca la fase de suministro análisis de decisiones, relativa al software y a los servicios. La realización de estas fases comprende las siguientes tareas:
  - a) Investigación de criterios técnicos y opciones.
  - b) Solicitud de propuestas o cotizaciones de los proveedores.
  - c) Validación de las afirmaciones y desempeño de los proveedores.
  - d) Evaluación y jerarquización de las propuestas de los proveedores.
  - e) Otorgamiento del contrato y sesión informativa a los proveedores no elegidos.
7. No basta simplemente comprar o construir sistemas que satisfagan los requerimientos del sistema deseado. El analista debe integrar o comunicar el nuevo sistema con los numerosos sistemas existentes que son fundamentales para la empresa. Muchos de estos sistemas incluyen el uso de tecnologías, técnicas y estructuras de archivos con diferencias impresionantes entre ellas.

## Preguntas de repaso



1. ¿Cuál es la diferencia esencial entre el análisis y el diseño de sistemas?
2. ¿Cuáles son sólo algunos de los métodos basados en modelos?
3. ¿Cuáles son algunos beneficios de los prototipos?
4. ¿Cuáles son las cinco tareas de alto nivel relacionadas con el diseño de sistemas para un proyecto de desarrollo dentro de la organización?
5. ¿Por qué es necesario diseñar la arquitectura de aplicaciones?
6. ¿Qué deben tener en mente los diseñadores cuando diseñan la base o bases de datos del sistema?
7. ¿Qué es un esquema de bases de datos?
8. ¿Cuál es el objetivo del diseño de la interfaz del sistema?
9. ¿En cuáles factores específicos deben centrar su atención los diseñadores del sistema cuando diseñan la interfaz de este último?
10. ¿Qué fase dentro del diseño de sistemas deberá llevarse a cabo necesariamente cuando se compra el software en lugar de desarrollarlo dentro de la organización? ¿Cuál es el objetivo de esta fase adicional?
11. ¿Qué es una solicitud de cotizaciones (RFQ, por sus siglas en inglés)?



## Problemas y ejercicios

1. ¿Cuál es el objetivo principal del diseño de sistemas y cuáles fases se incluyen en tal diseño? Si el análisis de sistemas se realiza de manera deficiente o incompleta, ¿puede superarse tal deficiencia con un buen diseño de sistemas?
2. Haga coincidir los términos de la primera columna con las definiciones o ejemplos de la segunda columna.
 

<ol style="list-style-type: none"> <li>1. Ingeniería de la información</li> <li>2. JAD</li> <li>3. Diseño estructurado moderno</li> <li>4. Elaboración de prototipos</li> <li>5. Diseño de sistemas</li> <li>6. Relación de entidad física</li> <li>7. Acoplamiento y cohesión</li> <li>8. RAD</li> <li>9. Diseño basado en modelos</li> <li>10. Codificar, implantar y reparar</li> <li>11. Herramienta CASE basada en repositorios</li> <li>12. OOD</li> <li>13. Gráfico de estructura</li> </ol>	<ol style="list-style-type: none"> <li>A. Ingeniería de la información</li> <li>B. Propiedades del modelo de diseño estructurado</li> <li>C. Énfasis en el desarrollo participativo</li> <li>D. <i>Rational</i> de IBM</li> <li>E. Modelo derivado del diseño estructurado</li> <li>F. Datos y procesos combinados</li> <li>G. Técnica basada en modelos, centrada en datos y sensible a procesos</li> <li>H. Énfasis en modelos gráficos (imágenes) del sistema</li> <li>I. Modelo funcional incompleto construido con el RAD</li> <li>j. Técnica de descomposición de procesos</li> <li>K. Tareas de especificación de soluciones basadas en computadora</li> <li>L. Fusión de JAD, elaboración de prototipos y técnicas estructuradas</li> <li>M. Posible error en la elaboración del prototipo</li> </ol>
---	---
3. Aunque la elaboración de prototipos tiene muchas ventajas, también entraña varias debilidades y riesgos. Analice algunos de esos riesgos y debilidades. ¿Cuáles estrategias se pueden implantar para disminuir el riesgo de que ocurran?
4. Considere los problemas que plantea la pregunta 3 y escriba un memorándum de políticas y procedimientos de una o dos páginas, dirigido a todos los analistas y diseñadores de sistemas de su organización, sobre la elaboración de prototipos.
5. Usted trabaja como diseñador de sistemas en una organización. Otro diseñador que era miembro de su equipo se jubiló recientemente. Su administrador se le acerca para ofrecerle que participe en las entrevistas para encontrar un sustituto. ¿Cuáles son las cualidades que debe buscar durante las entrevistas?
6. Usted diseña actualmente una pantalla de interfaz de datos para un nuevo sistema que está en desarrollo. El propósito de la pantalla es introducir los cambios de dirección que proporcionan los conductores al organismo de control vehicular estatal. Cada operador de datos clave introducirá los datos en esta pantalla desde formularios llenados a mano, con un total cercano a 1 000 modificaciones de dirección diarias. ¿Qué principio muy importante debe tenerse en mente?
7. En su organización, es tradicional que todos los participantes en un proyecto tengan una copia impresa de las especificaciones de diseño, una vez aprobadas. Aunque implica un costo adicional, los administradores piensan que es una forma de reconocer los esfuerzos de todos los participantes en el proyecto y hacer que estos últimos mantengan su compromiso. Si a la organización no le importa el costo, ¿existe algún problema en esta práctica?
8. Complete las oraciones siguientes:  
Una parte crítica de diseñar el \_\_\_\_\_ es decidir la forma de distribuir los datos, \_\_\_\_\_ y \_\_\_\_\_ del sistema a los diferentes \_\_\_\_\_. Las bases de datos son un recurso que \_\_\_\_\_ habitualmente muchos \_\_\_\_\_ y que pueden utilizar \_\_\_\_\_ futuros todavía no conocidos para propósitos \_\_\_\_\_. En el diseño de \_\_\_\_\_, la clave es poner en \_\_\_\_\_ para que el \_\_\_\_\_ entienda qué debe hacer a continuación y anticiparse a cada tipo de \_\_\_\_\_ que pueda realizar un usuario.
9. Es un diseñador de sistemas que tiene como responsabilidad la revisión de las propuestas de proveedores. Un proveedor que ha trabajado satisfactoriamente para la compañía presentó una propuesta que no satisface varios requerimientos críticos. ¿Qué debe hacer usted?
10. Haga coincidir las definiciones o ejemplos de la primera columna con los términos de la segunda columna.
 

<ol style="list-style-type: none"> <li>A. Fase de compra</li> <li>B. Muestra procesos físicos y bases de datos en toda la red.</li> <li>C. Despliegue basado en texto sustituido</li> <li>D. Criterios de evaluación y sistema de calificación de proveedores</li> <li>E. Documento de solitud de propuestas competidoras</li> <li>F. Parte de un plan para las actividades detalladas de la fase de diseño</li> </ol>	<ol style="list-style-type: none"> <li>1. DBA</li> <li>2. Análisis de distribución</li> <li>3. Arquitectura de aplicaciones</li> <li>4. Esquema de bases de datos</li> <li>5. RFQ</li> <li>6. Auditor</li> </ol>
--	--

- |  |                              |
|--|------------------------------|
| G. Especialista responsable de la arquitectura de bases de datos | 7. COTS                      |
| H. Documento de solitud de un producto específico                | 8. PDFD                      |
| I. Especialista responsable de controles internos                | 9. Costos monetarios “duros” |
| J. Tecnologías usadas para construir un sistema de información   | 10. Solución de “compra”     |
| K. Modelo estructural para bases de datos                        | 11. GUI                      |
| L. Producto de software comercial                                | 12. RFP                      |
11. El ciclo de vida de un proyecto en el que está trabajando incluye una solución de “compra” de un producto comercial listo para uso dirigido a los especialistas en marketing de la compañía. La organización quiere solicitar propuestas competidoras. Use el formato de la figura 10.12 en la preparación de una solicitud de propuestas (RFP). (Nota: Para los propósitos de este ejercicio, no es necesario desa-

- rollar una RFP detallada completa, sin embargo, la solicitud debe contener por lo menos los detalles de alto nivel y la información que precisa cada sección).
12. Usted trabaja para una empresa de consultoría que ha sido contratada para realizar la fase de diseño de sistemas de un proyecto. La fase de análisis del sistema estará a cargo de otra empresa consultora. Durante el diseño de sistemas, usted identifica lo que sin duda alguna es un error en los requerimientos. Aunque no tiene la certeza acerca de su gravedad, de algo está convencido: si este error, que cometió la otra empresa de consultoría, se menciona al administrador del proyecto, se tendrá que detener el diseño del sistema hasta que se corrija el problema. Esto ocasionará que su compañía se atrase y que tenga que pagarles a usted y a otros consultores por no hacer nada mientras se corrige el error o bien los puede despedir. ¿Cuál es su obligación ética en esta situación?
13. La seguridad y privacidad de los datos son temas cada vez más importantes. ¿Cuáles son algunos ejemplos de problemas de seguridad y privacidad que deben tener en mente los analistas y diseñadores de sistemas, así como los administradores de bases de datos, durante el desarrollo y mantenimiento de un sistema de base de datos relacional?

## Proyectos e investigación



- Usted ha estado trabajando varios años como analista de sistemas en el departamento de IT de una compañía fabricante de gabinetes. La empresa es conocida por la calidad de sus productos y por ser líder en su ramo. El ex Director de Información (CIO, por sus siglas en inglés), de la “vieja escuela”, se jubiló recientemente y lo sustituyó un CIO más dinámico y progresista. El nuevo director está realizando sesiones de lluvia de ideas para desarrollar su primer plan de arquitectura de IT, con el objetivo de mejorar el nivel de madurez de la organización. Se ha pedido a cada uno de los analistas y diseñadores de sistemas, incluido usted, que proporcione su aportación respecto de qué estrategia o estrategias de diseño debe adoptar la organización como su norma o estándar aprobado. Use la información de este libro, su experiencia personal y cualquier investigación complementaria que lleve a cabo, para escribir un memorándum al director, en el que:
  - Proporcione antecedentes pertinentes acerca de la organización; por ejemplo, su visión, misión, objetivos estratégicos, nivel de madurez, estructura y cultura organizacionales.
  - Describa las diversas estrategias de diseño de sistemas.
  - Compare y contraste sus métodos, fortalezas y debilidades.
  - Recomiende una estrategia o combinación de estrategias específicas para su adopción como estándar dentro de la organización. La recomendación también debe incluir la justificación de las bases de su recomendación.
- Usted es miembro de un grupo numeroso de analistas y diseñadores de sistemas en un proyecto a nivel de la empresa que afecta a todas las partes de la organización, tanto en las oficinas centrales como en las regionales de todo el país. Por recomendación de sus asesores, los administradores ejecutivos han decidido emprender el diseño de sistemas en la organización. Dadas la escala del proyecto y el tamaño de la organización, se trata de un proyecto que requiere la participación de cientos de propietarios y usuarios del sistema, localizados en las oficinas centrales y regionales. Muchos de ellos participan por primera vez en un proyecto de esta naturaleza. Una de las responsabilidades que se le ha asignado es verificar que esos propietarios y usuarios entiendan sus funciones respectivas en esta fase y su importancia relativa para el éxito general del proyecto.
  - Redacte uno o más mensajes de correo electrónico dirigidos a los propietarios y usuarios del sistema en la organización acerca de la fase de



diseño y sus funciones en ella, usando como guía las tareas 5.1 a 5.5. Su objetivo es tener la certeza de que entiendan sus funciones y se comprometan con el éxito del proyecto.

- b) Después de redactar los mensajes, explique la situación a varias personas en la organización. Pídale que lean y critiquen sus mensajes en cuanto a si son claros, completos y persuasivos. ¿Cuáles fueron los resultados? Si fuera una situación real, ¿habrían entendido sus funciones y habría tenido su explicación un efecto positivo en su compromiso?
3. En el proyecto 1 de este apartado, se le pide que busque diversas estrategias de diseño de sistemas, incluida la elaboración de prototipos. En los últimos años, parece haberse incrementado de manera exponencial la cantidad de métodos y herramientas de aplicaciones para la elaboración de prototipos. Investigue en Internet algunas de las tecnologías de prototipos que están disponibles. Además, hable con varios diseñadores de sistemas que usen los prototipos y pregúnteles qué piensan acerca de las diversas herramientas de aplicaciones para la elaboración de prototipos que están disponibles en el mercado. Redacte un análisis escrito en el que describa su investigación y presente sus resultados.
  4. Trabaja en el área de IT de una organización de ventas con seis oficinas satélite en su estado. La organización planea desarrollar e implantar un sistema de información basado en World Wide Web, de modo que las oficinas satélite puedan enviar sus informes de ventas en tiempo real. Sin embargo, el área de IT es pequeña, todos sus empleados ya están dedicados plenamente a las actividades de mantenimiento y soporte o a otros proyectos y, para colmo, ninguno tiene experiencia en el desarrollo de un sistema de información basado en WWW. Así pues, los administradores han decidido subcontra-

tar el diseño y desarrollo. Su función consiste en lo siguiente:

- a) Entrevistar a proveedores de IT en su localidad acerca de sus experiencias con solicitudes de propuestas (RFP, por sus siglas en inglés), es decir, indague cuáles son las deficiencias que habitualmente aprecian en tales solicitudes y cuáles conceptos clave tendrían que incluirse para preparar una propuesta adecuada. Redacte un memorándum breve, dirigido a los administradores, en el que describa las entrevistas.
  - b) Investigar algunas de las diversas plantillas de RFP disponibles, incluida la que se muestra en la figura 10.12. Elija una y explique la razón de su elección.
  - c) Use la plantilla que seleccionó para redactar una solicitud de propuesta.
5. Ahora que redactó la solicitud de propuesta en el inciso que precede, su siguiente tarea es planear la fase de diseño de sistemas de este proyecto. Aplique lo estudiado hasta este punto sobre el diseño de sistemas y prepare un anteproyecto de alto nivel que muestre las tareas principales, recursos y horas estimadas que se necesitan, calendario y dependencias.
  6. Las organizaciones de los sectores público y privado utilizan numerosos criterios de evaluación y sistemas de calificación para evaluar y jerarquizar las propuestas de proveedores. Además, este libro hace referencia a otros sobre diversos métodos en la lista de lecturas recomendadas del capítulo. Investigue en Internet estas otras estrategias o métodos que aplican organizaciones de los sectores público y privado. Además, trate de entrevistar a personal de esas organizaciones que tienen la responsabilidad de evaluar y jerarquizar las propuestas de proveedores, así como a los proveedores que elaboran las propuestas.



## Casos breves

1. En el capítulo previo usted trabajó en el diseño de un sistema para un organismo gubernamental. Elija una tarea específica en la creación del sistema (como el desarrollo de un sitio web) y obtenga por lo menos dos propuestas de distintos proveedores respecto de esa tarea. Valide las afirmaciones y desempeño de esos proveedores. Analice los resultados y preséntelos con sus recomendaciones a su profesor.
2. Usted desarrolla actualmente un sistema complejo para una compañía grande. El código de los programas será igualmente complejo, además de que el lenguaje de programación es relativamente novedoso para el equipo de programadores. Su superior

le ha solicitado que use un prototipo evolutivo en el desarrollo del sistema. ¿Es apropiado este modelo de prototipo para su uso? Si no lo fuera, ¿cómo debe atender la solicitud de su superior?

3. En el capítulo 7 se le pide que investigue una agencia de renta de automóviles y que cree un modelo de datos para el arrendamiento de vehículos. Utilice el trabajo que realizó y las entrevistas preliminares para crear un prototipo de sistema de arrendamiento vehicular.

Nota (para el estudiante): ¿Por qué necesita realizar más entrevistas? Esto tiene como fin el hecho de que usted pueda desarrollar una interfaz así como implementar la funcionalidad que

los *usuarios desean y necesitan*. Recuerde que, además de la funcionalidad de procesos, los *usuarios determinan* el éxito o fracaso de un sistema. ¿Cómo quieren *ellos* que sea y se comporte la interfaz? ¿En qué formato desean *ellos* que se manejen los datos?

4. En el problema anterior, usted creó un prototipo para un sistema de arrendadora de automóviles con base en una asignación de un capítulo previo y sus entrevistas preliminares. Ahora que ha generado el prototipo, regrese con su cliente para presentárselo.
  - a) ¿Cuál es la reacción del cliente? ¿Le agrada la funcionalidad? ¿Qué opina sobre el diseño de la

interfaz? Ponga por escrito sus respuestas y describa su lenguaje corporal.

- b) Modifique su sistema para incorporar las sugerencias y deseos de su cliente. ¿Acaso le pidió algo irracional o imposible a estas alturas? Si así fuera, descríbalos.
- c) Entregue su prototipo de sistema inicial, el prototipo del sistema revisado (modificado) y un ensayo en el que comente las necesidades del cliente, su respuesta al prototipo, cómo manejó las sugerencias del cliente en el sistema modificado y cualesquiera otros antecedentes adicionales sobre el sistema.

## Ejercicios de equipo e individuales



1. Cree un ejercicio para mejorar la creatividad. Puede ser un rompecabezas, un proyecto de arte, una experiencia, etc. Presente su idea al profesor. Luego, éste repartirá los ejercicios de creatividad al grupo, de modo que cada estudiante tenga un ejercicio que no sea el *suyo propio*. Lleve a cabo la tarea que se le asignó.
2. Navegue en unos cuantos sitios web y ponga por escrito las características de documentos e interfaz que le hayan agradado y desagradado en particular. Lleve esa información a la clase y compártala. Luego de considerar las experiencias de cada miembro del grupo, ¿cuáles características en su

opinión son particularmente buenas y debería tener todo sitio web? ¿Qué debería hacer y qué no debería hacer un sitio web?

3. Suponga que es el líder de un equipo que constantemente se retrasa en su calendario de trabajo. Suponga también que ello se debe a la planeación deficiente y a un manejo del tiempo también deficiente, no a problemas de asignación de recursos. Si solicita a su superior que reorganice el equipo, ¿qué puede hacer en lo personal para estimular a sus compañeros de equipo de modo que cumplan con el calendario de trabajo?

## Lecturas recomendadas



*Application Development Strategies* (publicación periódica mensual). Arlington, MA: Cutter Information Corporation. Esta publicación periódica es la favorita de los autores en cuanto a estrategias de desarrollo de sistemas, métodos, CASE y otras tendencias relevantes. Cada número se centra en un solo tema.

Boar, Benard. *Applicaton Prototyping: A Requirements Definition Strategy for the 80s*. Nueva York: John Wiley & Sons, 1984. Fue uno de los primeros libros publicados sobre la elaboración de prototipos de sistemas. Incluye un buen análisis de cuándo y cómo elaborar prototipos, además de una cobertura completa de los beneficios que pueden derivarse de tal estrategia.

Coad, Peter, y Yourdon, Edward. *Object-Oriented Design*, 2a. ed. Englewood Cliffs, NJ: Yourdon Press, 1991. El capítulo 1 de esta obra es un ejemplo excelente de docencia sobre los objetos y las relaciones de los métodos de objetos con todo lo que precedió a tales métodos.

Connor, Denis. *Information System Specification and Design Road Map*. Englewood Cliffs, NJ: Prentice Hall, 1985. En este libro se comparan los prototipos con otros métodos muy usados

de análisis y diseño. Se describe de manera satisfactoria por qué no se debe elaborar un prototipo sin especificaciones.

Gane, Chris. *Rapid Systems Development*. Englewood Cliffs, NJ: Prentice Hall, 1989. Esta obra presenta una panorámica adecuada del RAD, que combina el desarrollo basado en modelos con la elaboración de prototipos en un equilibrio adecuado.

Isshiki, Koichiro R. *Small Business Computers: A Guide to Evaluation and Selection*. Englewood Cliffs, NJ: Prentice Hall, 1982. Aunque centrada en computadoras pequeñas, esta obra describe muchas de las estrategias más conocidas para la evaluación de propuestas de proveedores. También describe muchos de los pasos del proceso de selección, aunque no los expone en la perspectiva de todo el ciclo de vida de desarrollo de sistemas.

Joslin, Edward O. *Computer Selection*, edición revisada. Fairfax Station, VA: Technology Press, 1977. Aunque en cierto punto es obsoleto, en este libro clásico se describieron originalmente los conceptos y métodos de selección que todavía son aplicables. La obra proporciona comentarios profundos sobre las relaciones entre proveedores, clientes y usuarios finales.

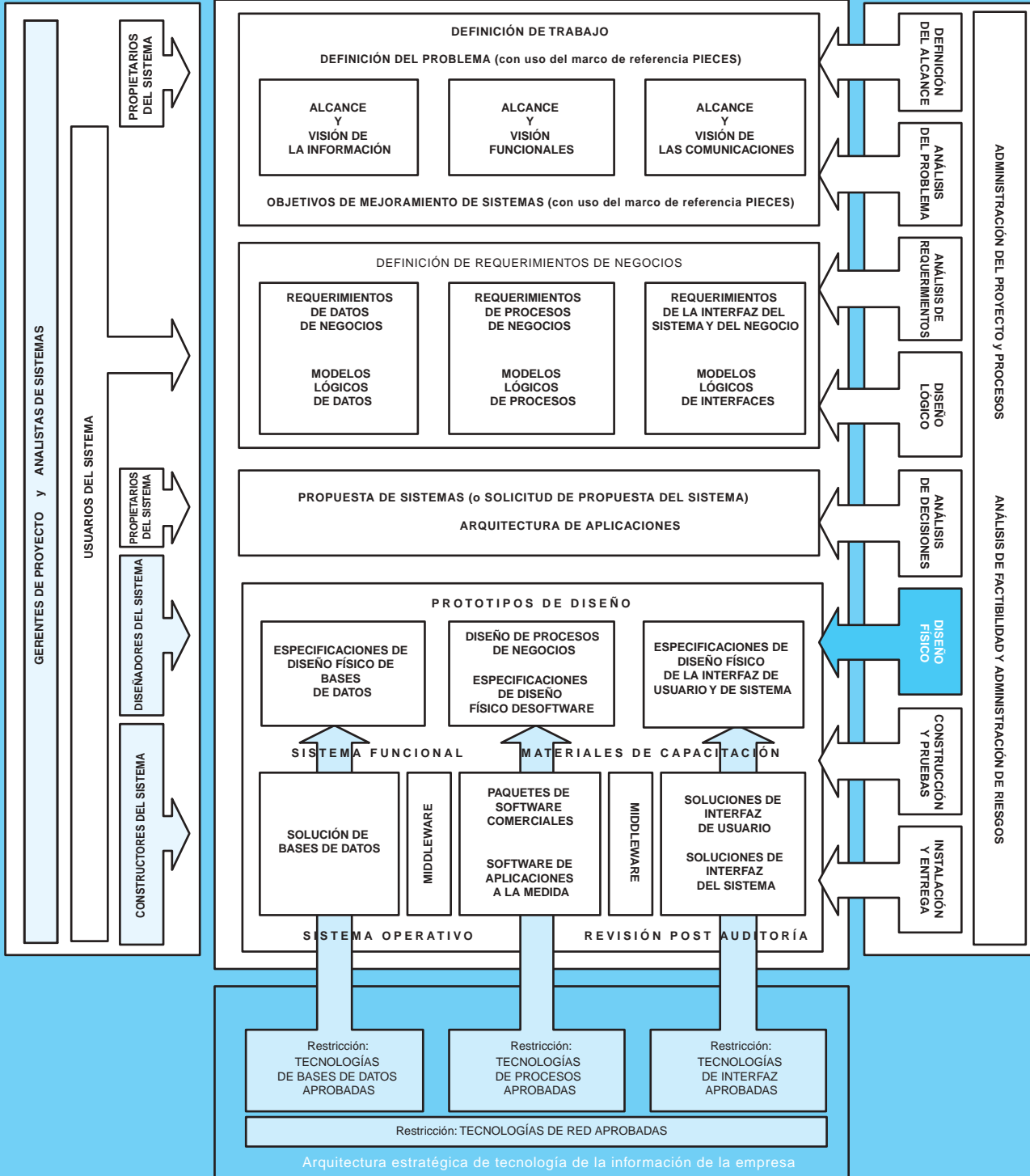
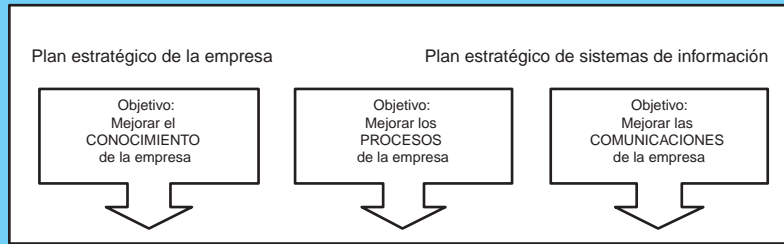
Lantz, Kenneth E. *The Prototyping Methodology*. Englewood Cliffs, NJ: Prentice Hall, 1986. Este libro brinda una cobertura excelente de los métodos de elaboración de prototipos.

Wood, Jane, y Denise Silver. *Joint Application Design: How to Design Quality Systems in 40% Less Time*. Nueva York: John Wiley & Sons, 1989. Es una obra que contiene una presentación profunda y excelente del desarrollo conjunto de aplicaciones (JAD).

Yourdon, Edward. *Modern Structured Analysis*. Englewood Cliffs, NJ: Yourdon Press, 1989. El capítulo 4 (“Moving into Design”) muestra por qué el diseño estructurado moderno es una continuación del análisis estructurado moderno.

Zachman, John A. “A Framework for Information System Architecture”, en *IBM Systems Journal* 26, núm. 3 (1987). Es un artículo en el que se presenta un marco de referencia conceptual muy usado respecto del diseño de sistemas de información.





# 11

## Modelado y arquitectura de la aplicación

### Panorámica y objetivos del capítulo

Este capítulo le enseña técnicas para diseñar la arquitectura general de la aplicación del sistema de información enfocándose en los modelos de procesos físicos. La arquitectura de aplicación de información y la modelación de los procesos físicos incluyen técnicas de distribución del conocimiento, los procesos y las comunicaciones en sus ubicaciones en la red para ambientes de computación distribuida. Los diagramas de flujo de los datos físicos se usan para documentar la arquitectura y el diseño en términos de las unidades de diseño (colección integral de los datos y de los procesos en ubicaciones específicas) que pueden diseñarse, elaborarse su prototipo, o construirse con mayor detalle e implantarse subsecuentemente como subsistemas autosuficientes. Usted sabrá que entiende la arquitectura de aplicación y el diseño de procesos cuando pueda:

- Definir la arquitectura de un sistema de información en términos del CONOCIMIENTO, LOS PROCESOS y las COMUNICACIONES; los componentes de todos los sistemas de información. De acuerdo con las tendencias actuales, estos componentes se distribuirán a través de una red.
- Diferenciar entre los diagramas de flujo de datos lógicos y físicos y explicar cómo se usan los diagramas de flujo de datos físicos para modelar la arquitectura de un sistema de información.
- Describir las alternativas de cómputo centralizada y distribuida para el diseño de los sistemas de información, incluyendo diferentes opciones de cómputo cliente/servidor y basadas en Internet.
- Describir las alternativas de base de datos y de distribución de datos para el diseño de los sistemas de información.
- Describir las alternativas de interfaces del usuario y del sistema para el diseño de los sistemas de información.
- Describir los diferentes ambientes de desarrollo de software para el diseño de los sistemas de información.
- Describir las estrategias de desarrollo o determinación de la arquitectura de un sistema de información.
- Dibujar diagramas de flujo de datos físicos para la arquitectura y los procesos de un sistema de información.

## Introducción

Se había decidido que parte del Sistema de servicios a los miembros SoundStage sería comprado, y que parte sería programado en casa. La pregunta de cuál sería la arquitectura para la parte programada del sistema seguía vigente. Bob Martínez había aprendido C++ .NET en la universidad y era un gran partidario de esa tecnología. Él señaló que usando el mismo lenguaje y el mismo marco .NET, podrían programar tanto una aplicación de escritorio de cliente/servidor para la parte interna de la aplicación y una aplicación Web para la parte de comercio electrónico de la aplicación. Las dos aplicaciones podrían incluso compartir algunos componentes. Y como la sintaxis C++ era esencialmente la misma que la de JavaScript, el código de la aplicación del lado del servidor y la de Web del lado del cliente sería muy similar.

Bob convenció a su jefa, Sandra. Sin embargo, como ella señaló, la arquitectura de aplicación aprobada de SoundStage se escribió antes de que .NET fuese liberado y no lo incluyó como una opción de desarrollo. Sandra hizo que Bob investigara y escribiera una solicitud de excepción para el comité de tecnología. Le tomó un par de borradores, pero finalmente lo escribió; en cierto modo ella pensó que el comité lo aprobaría. Estaba en lo correcto. Ahora podrían comenzar a diseñar “cómo” implantar el sistema, integrando los componentes comprados con .NET.

## La arquitectura de aplicación

El capítulo 10 presentó una visión general de alto nivel del proceso entero del diseño de sistemas. Usted aprendió que al inicio del diseño de sistemas se desarrolla un plano de la arquitectura que servirá como contorno general para el subsiguiente diseño interno y externo. Este capítulo enfoca la atención exclusivamente en ese plano y las alternativas actuales para la arquitectura de aplicación. (Los siguientes capítulos enfocan la atención en el diseño interno y externo detallado de cada componente de la arquitectura.) El plano de la arquitectura comunicará las siguientes decisiones del diseño:

- El grado en el cual el sistema de información estará centralizado o distribuido. La mayoría de los sistemas contemporáneos se extienden a través de redes, incluyendo tanto intranets (intranets) como Internet.
- La distribución de datos almacenados a través de la red. La mayoría de las bases de datos modernas son distribuidas o duplicadas a través de redes en un patrón de computación, ya sea de cliente/servidor o de la red.
- La tecnología para implantar todo el software que debe desarrollarse internamente. ¿Qué lenguaje de programación y qué herramientas se usarán?
- La integración de cualquier software comercial elegido y la necesidad de adecuar ese software a las necesidades del cliente.
- La tecnología que debe usarse para implantar la interfaz del usuario, incluyendo las entradas y las salidas.
- La tecnología que debe usarse para las interfaces con otros sistemas.

Estas consideraciones definen la arquitectura de aplicación del sistema de información. Una **arquitectura de aplicación** especifica las tecnologías que deben usarse para implantar uno o más sistemas de información (posiblemente todos). Sirve como un contorno general para el diseño, la construcción e implantación detallados.

En la mayoría de capítulos, inicialmente hemos enseñado conceptos y principios antes de introducir herramientas y técnicas. Para éste, vamos primero a introducir la herramienta primaria: los diagramas de flujo de datos físicos. Esto será por dos razones. Primera, que usted ya conoce los conceptos de sistemas y los componentes básicos de los diagramas de flujo de datos, del capítulo 8. Segunda, la herramienta es una forma elegante y relativamente simple de presentar los diferentes tipos de arquitectura de aplicación que queremos que usted aprenda.

Aunque usted aprenderá una técnica nueva en este capítulo, los diagramas de flujo de datos físicos, esto no es tan importante como los conceptos de aplicación de arquitectura que se usan para distribuir un sistema de información a través de una red de computadoras.

**arquitectura de aplicación** Especificación de las tecnologías que deben usarse para implantar los sistemas de información.

## Diagramas de flujo de datos físicos

Los diagramas de flujo de datos (DFD, por sus siglas en inglés) se introdujeron en el capítulo 8 como una herramienta de análisis de sistemas para modelar los requerimientos *lógicos* del negocio (lo que significa que no son “técnicos”) de un sistema de información. Con sólo algunas extensiones del lenguaje gráfico, los DFD también pueden utilizarse como una herramienta de diseño de sistemas para modelar el diseño y la arquitectura *físicos* (lo que significa que son “técnicos”) de un sistema de información. Los **diagramas de flujo de datos físicos** modelan las decisiones técnicas y humanas del diseño para ser implantados como parte de un sistema de información. Comunican las elecciones técnicas y otras decisiones de diseño a aquellos que realmente construirán e implantarán el sistema. En otras palabras, los DFD físicos sirven como un plano técnico para la construcción e implantación de sistemas.

Los diagramas de flujo de datos físicos fueron concebidos por Gane, Sarson y DeMarco como parte de una metodología formal de ingeniería del software llamada *análisis y diseño estructurados*. Esta metodología era especialmente bien adecuada para los sistemas y el software de información basados en transacciones de *COBOL* para las *mainframe*. La metodología requería una especificación rigurosa y detallada de las representaciones lógica y física de un sistema de información. Seguidamente, los analistas de sistemas o los ingenieros del software desarrollarían los siguientes modelos de sistemas y las especificaciones detalladas asociadas:

1. *Los DFD físicos del sistema actual*. Estos DFD físicos estaban destinados a ayudar a los analistas a identificar y analizar los problemas físicos en el sistema existente durante la fase de análisis de problemas del análisis de sistemas.
2. *Los DFD lógicos del sistema actual*. Estos DFD lógicos eran meramente una transformación de los DFD físicos anteriores que remueven todo detalle físico. Se utilizaron como un punto de partida para la fase de análisis de requerimientos del análisis de sistemas.
3. *Los DFD lógicos del sistema objetivo*. Estos DFD lógicos y sus especificaciones acompañantes (las estructuras de datos y el inglés estructurado) estaban dirigidos a representar los requerimientos no técnicos detallados para el sistema nuevo.
4. *Los DFD físicos del sistema objetivo*. Estos DFD físicos estaban dirigidos a proponer y modelar las elecciones de tecnología y decisiones de diseño para todos los procesos lógicos, flujos de datos y almacenamientos de datos. Estos diagramas (el enfoque de este capítulo) se desarrollan durante la etapa de diseño de sistemas del proyecto.
5. *Las gráficas de estructura de los elementos de software del sistema objetivo*. Los DFD físicos anteriores serían transformados en gráficas de estructura que ilustran una jerarquía de arriba abajo de los módulos de software que se conformarían con los principios aceptados de un buen diseño del software.

La metodología antes citada requería de mucho trabajo y de un rigor y una precisión significativos para que cumpliera con el resultado esperado. Hoy, el análisis estructurado y la metodología del diseño completos, tal como se acaba de describir, raramente se practican (no se ajustan tan bien a las tecnologías orientadas a objetos y al software basado en componentes), pero la diagramación de flujos de datos (tanto lógica como física) sigue siendo un legado útil y muy socorrido del análisis estructurado y de la era del diseño del desarrollo de sistemas.

En este capítulo examinaremos las convenciones gráficas para los DFD *físicos*. Los DFD físicos usan las mismas conexiones y formas básicas que los DFD lógicos (capítulo 8), a saber: a) *procesos*, b) *agentes externos*, c) *almacenamiento de datos*, y d) *flujos de datos*. En la figura 11.1 se muestra un DFD físico. Por ahora, simplemente observe que el DFD físico contiene sobre todo más detalle técnico y de implantación que su equivalente DFD lógico.

### > Los procesos físicos

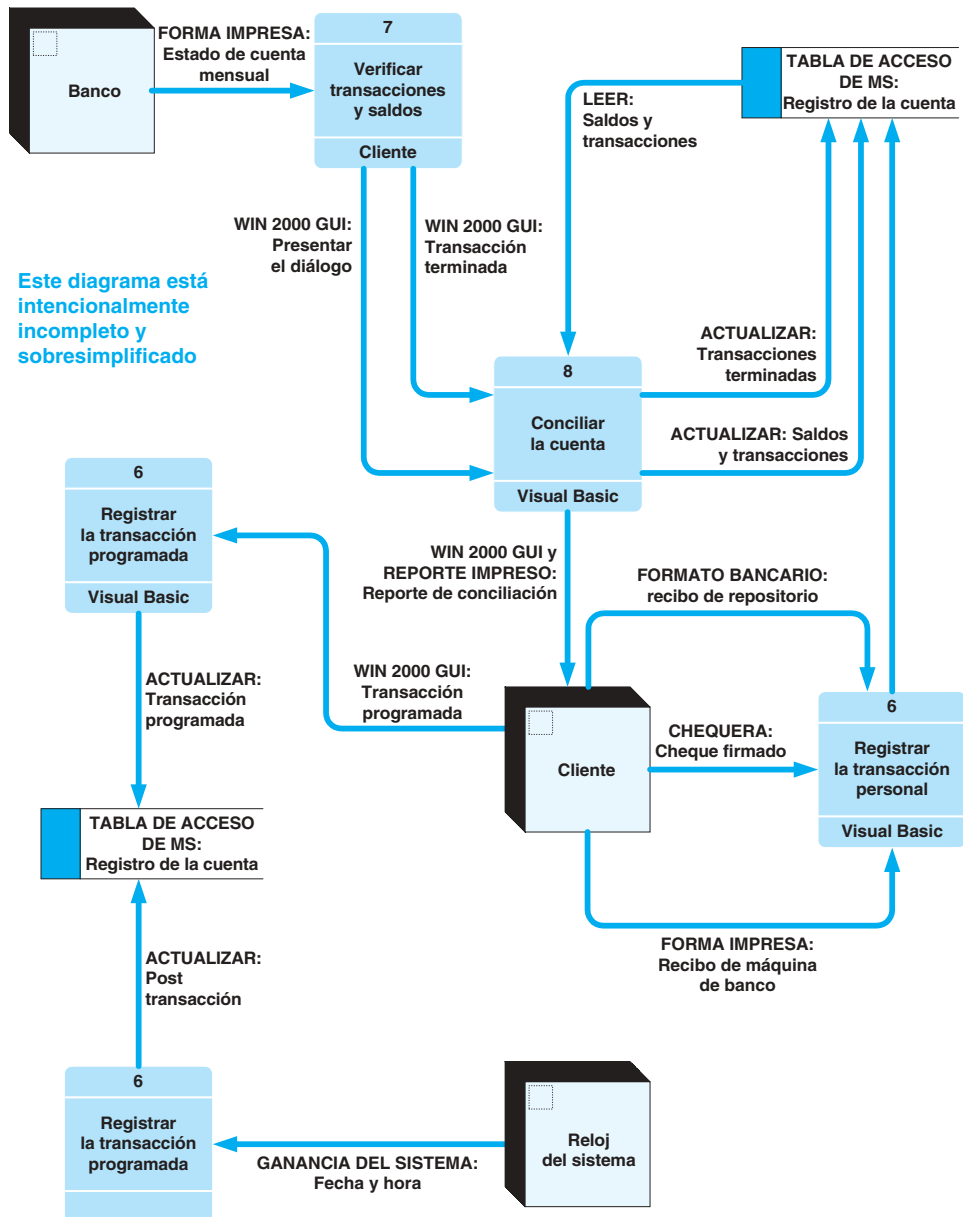
Recuerde que los procesos son las formas clave en cualquier DFD. Por eso es que se les llama *modelos de proceso*. Los DFD físicos bosquejan la implantación planificada, física, de cada proceso. Se entiende por *proceso físico* un *procesador*, como una computadora o

**diagrama de flujo de datos físicos** Modelo de procesos usado para comunicar las características técnicas de implantación de un sistema de información.



FIGURA 11.1

Un ejemplo de diagrama de flujo de datos físicos



persona, o la implantación técnica del trabajo específico a ser realizado, tal como un programa de computadora o un proceso manual.

En una etapa anterior del proyecto, durante el análisis de requerimientos, especificamos los procesos *lógicos* necesarios para cumplir con requerimientos esenciales de negocios. Estos procesos lógicos fueron modelados en los diagramas lógicos de flujo de datos, en el capítulo 8. Ahora, durante el diseño de sistemas, debemos especificar cómo se implantarán físicamente estos procesos lógicos. Como se implica en la definición anteriormente citada para los procesos físicos, hay dos características de diagramas físicos de flujo de datos:

- Los procesos lógicos frecuentemente se asignan a procesadores físicos específicos como PCs, servidores, *mainframes*, personas u otros dispositivos en una red de computadoras. Para este fin, podríamos dibujar un DFD físico para modelar la estructura de la red.
- Cada proceso lógico debe implantarse como uno o *más* procesos físicos ya que algunos procesos lógicos deben dividirse en procesos físicos múltiples por una o más de las siguientes razones:
  - Para dividir el proceso en la parte que va a ser realizado por personas y la parte que va a ser realizada por la computadora.



- Para dividir el proceso en la parte que va a ser implantada con una tecnología y la parte que va a ser implantada con una tecnología diferente.
- Para mostrar implantaciones múltiples pero diferentes del mismo proceso lógico (tal como un proceso para las órdenes en papel y un proceso diferente para las órdenes en Internet).
- Para añadir procesos que son necesarios para manejar excepciones o implantar requerimientos de seguridad y rastreos de auditoría.

En todos los casos, si usted divide un proceso lógico en procesos físicos múltiples, o añade procesos físicos adicionales, tiene que añadir todos los flujos de datos necesarios para conservar la esencia del proceso lógico original. En otras palabras, los procesos físicos de cualquier modo deben cumplir con los requerimientos lógicos de proceso.

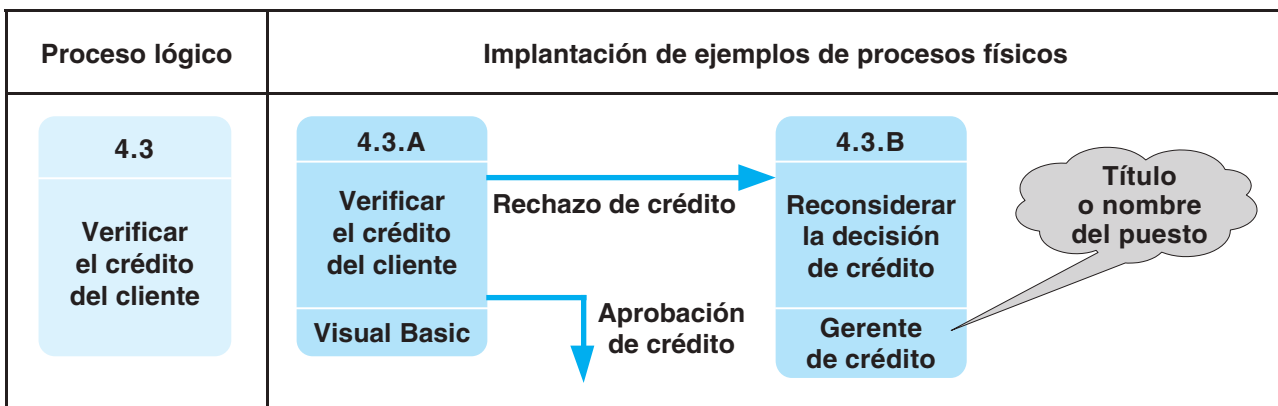
Los ID (identificadores) son optativos, pero pueden ser útiles para cotejar procesos físicos con sus contrapartes lógicas (especialmente si el proceso lógico debe implantarse con múltiples procesos físicos). Los nombres de proceso usan la misma convención de cláusula de verbo de acción + sustantivo/objeto que presentamos en el capítulo 8. Este nombre se registra en el centro de la forma (vea al margen). La implantación se registra en la parte inferior de la forma. Tal vez tenga que ajustarse esta convención dependiendo de las capacidades de su herramienta CASE o de su herramienta de diagramación automatizada. Los siguientes nombres demuestran implantaciones posibles diversas del mismo proceso lógico:

**Identificación (opcional)**  
**Verbo de acción + Frase nominal u objeto**  
**Implantación**

Procesos lógicos	Implantaciones de ejemplos de procesos físicos				
<p>4.3</p> <p>Verificar el crédito del cliente</p>	<p>4.3</p> <p>Verificar el crédito del cliente</p> <p>Ejecutivo de cuenta</p>	<p>4.3</p> <p>Verificar el crédito del cliente</p> <p>COBOL/CICS</p>	<p>4.3</p> <p>Verificar el crédito del cliente</p> <p>Visual Basic</p>	<p>4.3</p> <p>Verificar el crédito del cliente</p> <p>Libros rápidos</p>	

Si su herramienta CASE limita el tamaño de los nombres, usted podría tener que desarrollar y usar un conjunto de abreviaturas para la tecnología (y posiblemente abreviar sus verbos de acción y sus cláusulas del objeto).

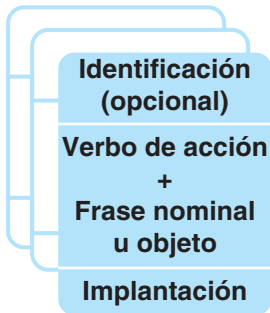
Si un proceso lógico debe ser implantado parcialmente por personas y en parte por el software, debe dividirse en procesos físicos separados, y deben agregarse los flujos de datos apropiados entre los procesos físicos. El nombre de un proceso físico a ser realizado por personas, no por un software, debería indicar quién realizaría ese proceso. Recomendamos que use títulos o puestos, no nombres propios. El siguiente es un ejemplo:



No solamente cambiamos el proceso manual, RECONSIDERAR LA DECISIÓN DE CRÉDITO, para un agente externo, GERENTE DE CRÉDITO, porque el proceso lógico entero, VERIFICAR EL CRÉDITO DEL CLIENTE está en el alcance del proyecto. Por esa razón, ambos aspectos de la implantación física también están en el alcance. El diseño no está completo hasta que especifiquemos el proceso tanto para los aspectos automatizado y manual del requerimiento de negocios.

Para los procesos computarizados, el método de implantación se escoge, en parte, de una de las siguientes posibilidades:

- Un paquete de software de aplicación comprado (por ejemplo, *Sap*, un programa de aplicación empresarial, o *Ariba*, un programa de administración de la cadena de suministro basado en Internet).
- Un sistema o un programa adquirido (por ejemplo, el *Exchange Server* de la Corporación Microsoft; un sistema de correo electrónico/mensajería, o el *WebSphere Commerce Business* de IBM, un marco de comercio electrónico).
- Un programa de aplicación existente de una biblioteca de programas, indicado simplemente como BIBLIOTECA o el NOMBRE de la biblioteca.
- Un programa a ser escrito. Típicamente, el método de implantación especifica el lenguaje o herramienta para usarse para construir el programa. Los métodos de implantación de ejemplo incluyen el VB, .NET, C++, JAVA, MS ACCESS, PERL, u ORACLE DEVELOPER.



También debe presentarse una última construcción física de proceso, el *multiproceso* (vea al margen). El multiproceso indica las implantaciones múltiples del mismo proceso o procesador físico. Por ejemplo, podemos usar este símbolo para indicar múltiples PC, la implantación de un programa nombrado en múltiples PC, o la implantación de trabajo para ser realizado por múltiples personas. Algunas herramientas de CASE no soportan esta construcción. Si no lo hacen, usted podría necesitar recurrir a nombres en plural para implicar la multiplicidad de un proceso o un procesador.

Muchos diseñadores prefieren una convención de denominación más física para los procesos de computadora. En lugar de una frase sustantivo + verbo, la sustituirían por el nombre del archivo del código fuente del programa de computadora. Considere los siguientes ejemplos:

Proceso lógico	Implantación de ejemplos de procesos físicos	
<p style="text-align: center;"><b>4.3</b></p> <p style="text-align: center;"><b>Verificar el crédito del cliente</b></p>	<p style="text-align: center;"><b>4.3.A</b></p> <p style="text-align: center;"><b>CHK_CREDIT.COB</b></p> <p style="text-align: center;"><b>COBOL + CICS</b></p>	<p style="text-align: center;"><b>4.3.B</b></p> <p style="text-align: center;"><b>appCheckCredit.vbx</b></p> <p style="text-align: center;"><b>Visual Basic</b></p>

Muchas organizaciones tienen convenciones y estándares de denominación para los nombres de programa.

Otra vez, el número de procesos físicos en un DFD físico casi siempre será mayor que el número de procesos lógicos en su DFD lógico equivalente. Los procesos pueden agregarse para reflejar la recolección de datos, el filtrado, el envío, la preparación o las revisiones de calidad; todo en respuesta a la visión de implantación que ha sido seleccionada. Además, algunos procesos lógicos pueden dividirse en múltiples procesos físicos para reflejar partes de un proceso que debe hacerse manualmente contra lo hecho por una computadora, para ser implantado con una tecnología diferente o para ser distribuido a los clientes, los servidores o las diferentes computadoras conectadas. Es importante que los DFD físicos finales reflejen todo los procesos manuales y de computadora que se requieren para la estrategia de implantación elegida.

### > Los flujos de datos físicos

Recuerde que todos los procesos en cualquier DFD deben tener al menos un flujo de datos de entrada y de salida. Un *flujo de datos físicos* representa cualquiera de lo siguiente: 1) la implantación planeada de una entrada o de una salida de un proceso físico; 2) una orden de la base de datos o acciones tales como *create* (crear), *read* (leer), *update* (actualizar), o *delete* (borrar); 3) la importación o exportación de datos para otro sistema de información a través de una red, o 4) el flujo de datos entre dos módulos o subrutinas dentro del mismo programa.

Los flujos de datos físicos se denominan tal como lo indican las plantillas en el margen. La figura 11.2 muestra la aplicación de una de estas convenciones de denominación tal como se aplican a los diferentes tipos de flujos de datos físicos.

Los DFD físicos también deben indicar cualesquiera flujos de datos para ser implantados como formas del negocio. Por ejemplo, FORMA 23: La SOLICITUD DE CURSO podría ser una forma del negocio usada por los estudiantes para registrarse a las clases. Las formas del negocio frecuentemente usan una implantación de copia múltiple (al carbón o papel carbón). En algún punto del procesamiento, las diferentes copias se dividen y viajan a los diferentes procesos manuales. Esto se muestra en un DFD físico como un flujo de datos divergentes (introducido en el capítulo 8). Cada copia debería ser denominada de manera única. Por ejemplo, en un restaurante, el cliente recibe la FORMA: COMPROBANTE DE LA TARJETA DE CRÉDITO (COPIA DEL CLIENTE) y el comerciante, a su vez, retiene la FORMA: COMPROBANTE DE LA TARJETA DE CRÉDITO (COPIA DEL COMERCIANTE).

La mayoría de flujos de datos lógicos se trasladan a los DFD físicos. Algunos pueden consolidarse en flujos de datos físicos individuales que representan formas del negocio. Otros pueden dividirse en flujos múltiples como resultado de haber dividido los procesos lógicos en procesos físicos múltiples. Otros más tendrían que duplicarse como flujos múltiples con implantaciones técnicas diferentes. Por ejemplo, el de flujo de datos lógicos ORDEN podría implantarse como todos los siguientes: FORMA: ORDEN, TELÉFONO: ORDEN (orden verbal tomada por teléfono), el HTML: ORDEN (la orden se entregó por la Red), FAX: ORDEN (orden recibida por fax), y el MENSAJE: ORDEN (una orden entregada vía correo electrónico).

### > Los agentes externos físicos

Los agentes externos se trasladan sin cambios del DFD lógico al DFD físico. ¿Por qué? Por definición, los agentes externos fueron clasificados durante el análisis de sistemas como fuera del alcance de los sistemas y por consiguiente no sujetos a cambio. Sólo un cambio en los requerimientos puede iniciar un cambio en los agentes externos.

### > Los almacenamientos de datos físicos

Del capítulo 8, usted sabe que cada almacenamiento de datos en el DFD lógico ahora representa todas las instancias de una entidad incluida en un diagrama de entidad relación. Los almacenamientos de datos físicos implantan los almacenamientos de datos lógicos. Un *almacenamiento de datos físicos* representa la implantación de uno de los siguientes: 1) una base de datos, 2) una tabla en una base de datos, 3) un archivo de computadora, 4) una cinta o un medio de respaldo de cualquier cosa importante, 5) cualquier archivo temporal o lote de procesamiento según lo necesite el programa (por ejemplo, TABLAS DE IMPUESTOS) o 6) cualquier tipo de archivo no computarizado.

Cuando la mayoría de la gente piensa en almacenamientos de datos, imagina archivos de computadora y bases de datos. Pero muchos archivos de datos no están computarizados. Inmediatamente vienen a la mente los archiveros de registros de papel; sin embargo, la mayoría de negocios están repletos con formas más sutiles de almacenamientos manuales de datos, tales como tarjetas de dirección, catálogos de papel, hojas a prueba de fraude de información variada importante y reusable, manuales de estándares, manuales de procedimientos de operación estándar, directorios y cosas por el estilo. A pesar de las predicciones acerca de la obsolescencia de los archivos de papel, seguirán siendo parte de muchos sistemas en el futuro previsible; y no es por otra razón que por las siguientes: 1) el papel es psicológicamente cómodo, y 2) el gobierno frecuentemente lo requiere.

El nombre de un almacenamiento de datos físico usa el formato indicado en el margen. Algunos ejemplos de almacenamientos de datos físicos se muestran en la figura 11.3 (vea la página 381).

Algunos diseños requieren la creación de archivos temporales para que actúen como una cola o una memoria temporal entre los procesos físicos que tienen sincronización

Método de implantación:  
nombre del flujo de datos



O









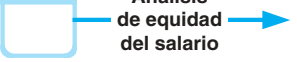


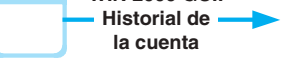



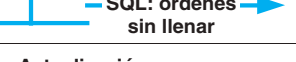
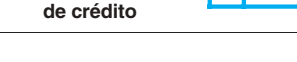






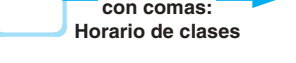




Nombre del flujo de datos  
(método de implantación)



ID (opcional)	Método de implantación: Nombre del almacenamiento de datos
------------------	--

ID (opcional)	Nombre del almacenamiento de datos (método de implantación)
------------------	---

**FIGURA 11.2**  
Flujos de datos físicos

Flujo de datos lógicos	Implantación	Ejemplo de flujo de datos físicos
	Entrada a la computadora (teclado)	
	Entrada a la computadora (Internet)	
	Entrada a la computadora (sin teclado)	
	Entrada a la computadora (archivo por lotes)	
	Salida de la computadora (impresa)	
	Salida de la computadora (en línea)	
	Crear un registro en una base de datos	
	Leer registros en una base de datos	
	Actualizar un registro en una base de datos	
	Borrar un registro en una base de datos	
	Importar un archivo de datos	
	Exportar un archivo de datos	
	Pasar datos entre módulos de un programa	
	Pasar una forma manual	

diferente. Estos archivos se documentan de la misma manera, pero sus nombres indican su estatus temporal.

Los procesos físicos, flujos de datos, agentes externos y almacenamientos de datos constituyen los diagramas de flujo de datos físicos. Y estos DFD físicos modelan la arquitectura propuesta o planeada de la aplicación del sistema de información. Subsiguientemente podemos usar ese modelo físico para diseñar los detalles internos y externos para cada almacenamiento de datos (capítulo 12) y flujo de datos (capítulos 13-15). Ahora que usted entiende los componentes básicos de los DFD físicos, usémoslos para introducir algunas de las elecciones de la arquitectura de hoy para el diseño de sistemas de información.

Almacenamiento de datos lógicos	Implantación	Almacenamiento de datos físicos
 Recursos humanos	Una base de datos (tablas múltiples)	 Oráculo: DB de recursos humanos
 Mercadeo	Una visión de la base de datos (subconjunto de una base de datos)	 Servidor SQL: DB del mercadeo del noreste
 Órdenes de compra	Una tabla en una base de datos	 Acceso a MS: órdenes de compra
 Cuentas por cobrar	Un archivo de heredados	 Archivo VSAM: cuentas por cobrar
 Tasas de impuestos	Datos estáticos	 ARREGLO: tabla de impuestos
 Órdenes	Un archivo fuera de línea	 Respaldo de CINTA: órdenes cerradas
 Empleados	Un archivo de registros en papel	 Gabinete de archivos: Registros personales
 Datos de contacto del profesorado/equipo	Un directorio	 Cuaderno: Directorio de profesores/personal
 Inscripción en los cursos por fechas	Reportes archivados (para reuso y recuperación)	 GERENTE DE REPORTE: reportes de inscripción en los cursos

FIGURA 11.3

Almacenamientos de datos físicos

## Arquitectura de tecnología de la información

La arquitectura de tecnología de la información (TI) puede ser un tema complicado digno de su propio curso y libro de texto. (Vea las Lecturas recomendadas al final de este capítulo.) En esta sección trataremos de resumir las alternativas y las tendencias contemporáneas de tecnología de la información que influyen en las decisiones de diseño en el momento en que el libro va a la prensa. Debe señalarse que las nuevas alternativas evolucionan de manera constante. Los mejores analistas de sistemas no sólo aprenderán más acerca de estas tecnologías sino que también entenderán cómo trabajan y sus limitaciones. Tal nivel de detalle está más allá del alcance de este libro. Los analistas de sistemas con frecuencia deben leer las publicaciones de tendencias para mantenerse al día con las últimas tecnologías y las técnicas para conservar a sus clientes y sus sistemas de información competitivos.

El marco de referencia del sistema de información proporciona un marco adecuado para entender la arquitectura de la tecnología de la información. Por esta razón, actualmente nuestros componentes están siendo distribuidos o duplicados a través de redes. Llamamos a este enfoque la *arquitectura distribuida de sistemas*:

- Los estándares de la arquitectura y/o las restricciones de la tecnología se representan en la fila inferior del marco de la página inicial (página 372) de este capítulo. Observe que estos estándares o decisiones se determinan ya sea como parte de un proyecto separado de arquitectura (lo que se prefiere y es cada día más común) o como parte de cada proyecto de desarrollo de sistemas.
- Las flechas que apuntan hacia arriba indican los estándares de tecnología que estimularán o restringirán los modelos de diseño.

**sistema distribuido** Sistema en el que los componentes se distribuyen a través de múltiples ubicaciones y redes de computadoras.

**sistema centralizado** Sistema en el cual todos los componentes son atendidos por una computadora central multiusuario.

## > Los sistemas distribuidos

Los sistemas de información actuales ya no son sistemas monolíticos basados en computadoras *mainframe*. En lugar de eso, se construyen sobre alguna combinación de redes para integrar sistemas distribuidos. Un **sistema distribuido** es aquel donde los componentes de un sistema de información se distribuyen entre las múltiples ubicaciones de una red de computadoras. En consecuencia, la carga de trabajo de procesamiento requerida para soportar estos componentes también se distribuye a través de las múltiples computadoras en la red.

Lo opuesto a los sistemas distribuidos son los sistemas centralizados. En los **sistemas centralizados**, una computadora central multiusuarios (generalmente una *mainframe*) aloja a todos los componentes de un sistema de información. Los usuarios interactúan con esta computadora central vía las terminales (o, actualmente, una PC que emula a una terminal), pero virtualmente todo el trabajo y procesamiento real se hacen en la computadora central.

Los sistemas distribuidos son inherentemente más complicados y más difíciles para implantar que las soluciones centralizadas. Entonces, ¿por qué la tendencia hacia los sistemas distribuidos?

- Los negocios modernos ya están distribuidos y, así, necesitan soluciones de sistemas distribuidos.
- La computación distribuida mueve información y servicios más cerca de los clientes que los necesitan.
- La computación distribuida consolida el poder increíble que resulta de la proliferación de computadoras personales a través de una empresa (y la sociedad en general). Muchas de estas computadoras personales usan solamente en una fracción de su potencial de procesamiento cuando se usan como PCs autónomas.
- En general, las soluciones distribuidas de sistema son más amigables con el usuario porque usan la PC como procesador de la interfaz del usuario.
- Las computadoras personales y los servidores de red son mucho menos caros que las *mainframe*. (Pero, sin duda, el costo total de propiedad es por lo menos igual de caro una vez que se incluyen todas las complejidades de sistema de redes.)

Hay un precio que pagar por los sistemas distribuidos. El tráfico de datos de la red puede causar tal congestión que realmente desacelera su desempeño. La seguridad de datos y la integridad también pueden comprometerse con mayor facilidad en una solución distribuida. Aun así, no hay argumentos en contra de la tendencia hacia la arquitectura de los sistemas distribuidos. Aun cuando todavía existen muchas aplicaciones centralizadas heredadas, poco a poco se están transformando en sistemas de información distribuidos.

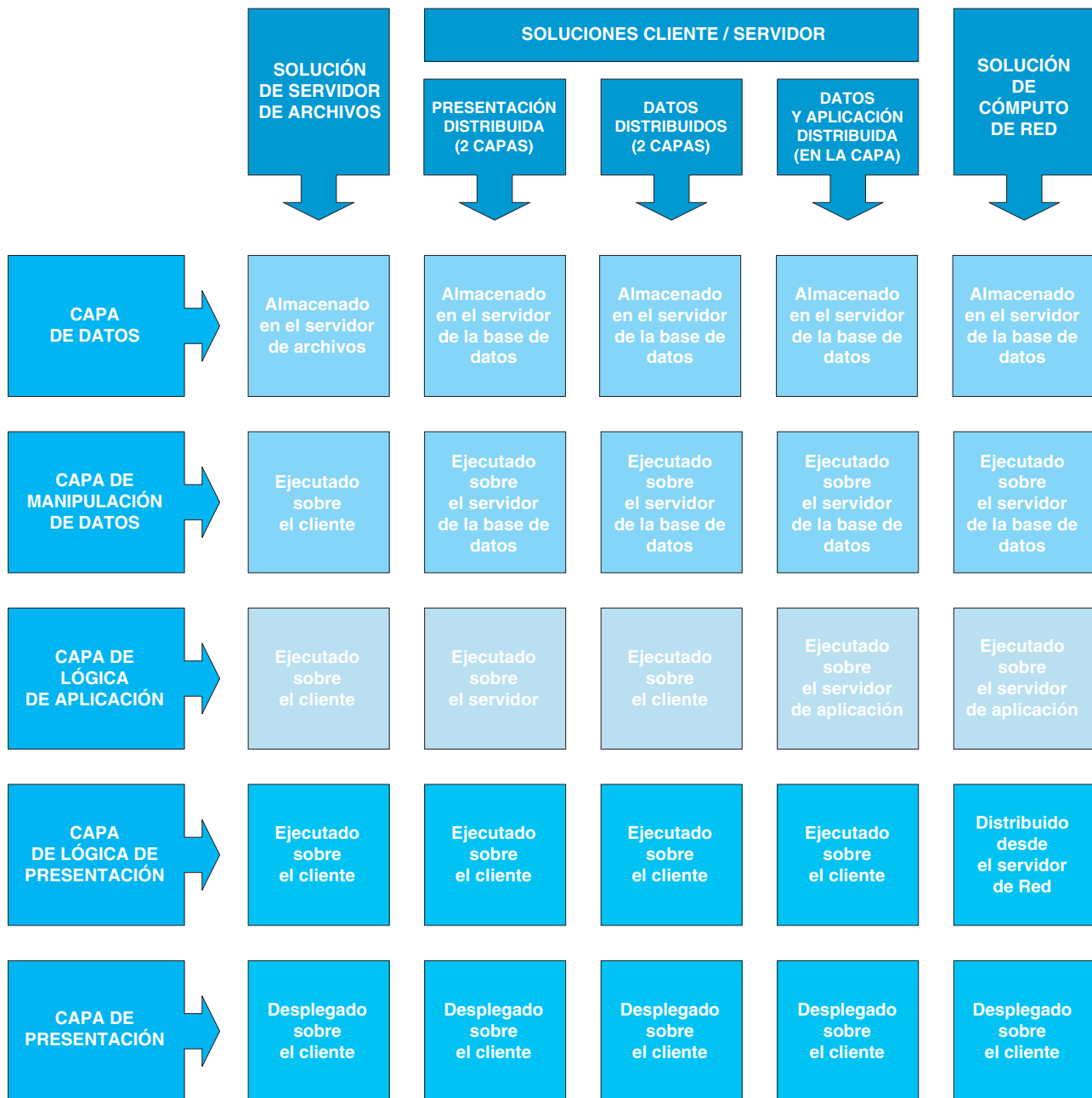
La figura 11.4 compara las diferentes arquitecturas de sistemas distribuidos. En lo conceptual, cualquier aplicación de sistema de información puede mapearse en cinco capas:

- La *capa de presentación* es la interfaz real del usuario: la presentación de entradas y salidas al usuario.
- La *capa lógica de presentación* es cualquier procesamiento que debe hacerse para generar la presentación. Los ejemplos incluyen la edición de los datos de entrada y el formateo de los datos de salida.
- La *capa lógica de aplicación* incluye toda la lógica y el procesamiento que se requiere para soportar la aplicación y las reglas reales del negocio. Los ejemplos incluyen la verificación de crédito, cálculos, el análisis de datos y cosas por el estilo.
- La *capa de manipulación de datos* incluye todas las órdenes y la lógica que se requieren para almacenar y recuperar datos hacia y desde la base de datos.
- La *capa de datos* es la información almacenada en una base de datos.

La figura 11.4 muestra a estas capas conceptuales como filas. Las columnas en la figura ilustran cómo pueden implantarse las capas en las diferentes arquitecturas distribuidas de los sistemas de información. Hay tres tipos de arquitectura distribuida de sistemas:

- *La arquitectura del servidor de archivos.*
- *La arquitectura del cliente/servidor.*
- *La arquitectura basada en Internet.*

Analicemos cada uno en el mayor detalle.



**FIGURA 11.4** Tipos de arquitectura y sistemas distribuidos

**Arquitectura del servidor de archivos** Hoy, muy pocas computadoras personales y estaciones de trabajo se usan para soportar sistemas de información autónomos. Las organizaciones necesitan compartir datos y servicios. Las redes de área local permiten que muchas PC y estaciones de trabajo estén conectadas para compartir recursos y comunicarse entre sí. Una **red de área local (LAN)**, por sus siglas en inglés) es un conjunto de computadoras cliente (generalmente PCs) conectadas a uno o más servidores (generalmente una PC más poderosa o computadoras más grandes) a través de cable o conexiones inalámbricas para distancias relativamente cortas, por ejemplo, en un solo departamento o edificio.

En los más sencillos ambientes de red de área local, una arquitectura de servidor de archivos se usa para implantar sistemas de información. Un **sistema de servidor de archivos** es una solución basada en redes de área local en la cual una computadora del servidor aloja sólo la capa de datos. Todas las demás capas de la aplicación del sistema

**red de área local (LAN)**

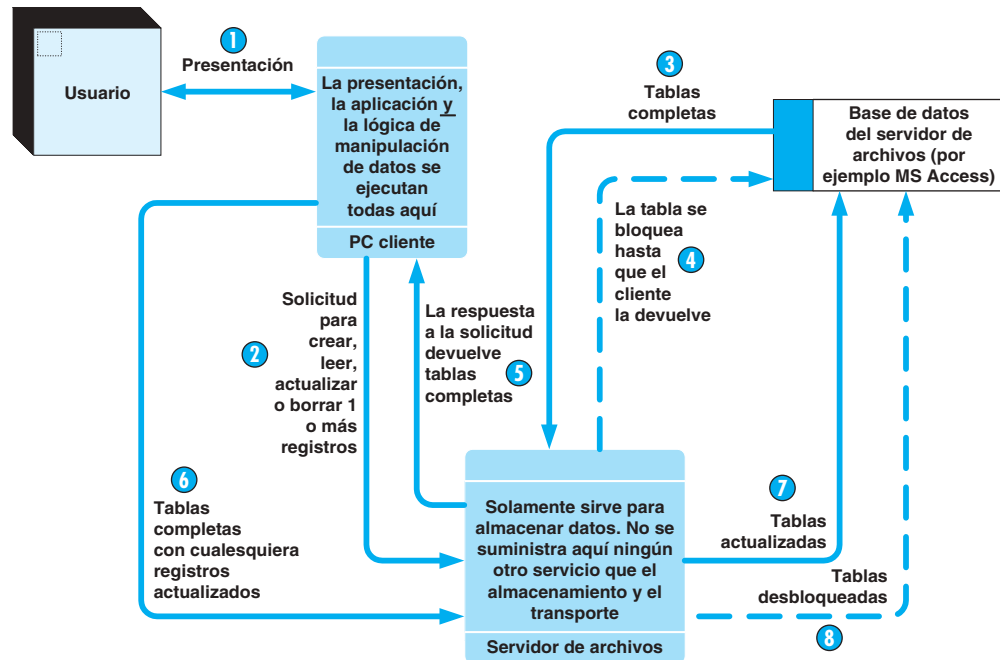
Conjunto de computadoras cliente conectadas a una distancia relativamente corta con uno o más servidores.

**sistema de servidor de archivos**

LAN en la cual un servidor aloja los datos de un sistema de información.

FIGURA 11.5

Arquitectura de un servidor de archivos



de información se implantan en la PC del cliente. (Nota: Los servidores de archivos típicamente se usan también para compartir otros archivos que no son de base de datos a través de las redes [los ejemplos incluyen los documentos de procesador de palabra, las hojas de cálculo, imágenes y gráficos, dibujos de ingeniería, presentaciones, etc.].) Una arquitectura de servidor de archivo se ilustra en la figura 11.5. Esta arquitectura es típica de aquellas que se usan para muchos motores de la base de datos de la PC como Microsoft *Access* y *FoxPro*. Aun cuando su base de datos *Access* puede guardarse en un servidor de la red, el programa *Access* debe instalarse o ejecutarse desde cada PC que use la base de datos.

Las arquitecturas del servidor de archivos son prácticas sólo para aplicaciones pequeñas de la base de datos compartidas por relativamente pocos usuarios porque la tabla o el archivo entero de registros deben descargarse primero a la PC cliente, donde la lógica de manipulación de datos se ejecutará para leer un registro individual deseado. Hay varias desventajas de este enfoque:

- Deben moverse grandes cantidades de datos innecesarios entre el cliente y el servidor. Este tráfico de datos puede reducir de manera significativa el desempeño de la red y de la aplicación.
- La PC cliente debe ser robusta (una computadora así se le llama *cliente pesado*). Estará haciendo virtualmente todo el trabajo, incluyendo la manipulación de datos, la lógica de aplicación, la lógica de presentación y la presentación. También debe tener suficiente capacidad de disco para almacenar las tablas descargadas.
- La integridad de la base de datos puede comprometerse fácilmente. Piense acerca de esto. Si se ha descargado cualquier registro para ser actualizado, el archivo entero ha sido descargado. Debe prevenirse (o ponerles candado) a otros usuarios (los otros clientes) para que no hagan cambios en cualquier otro registro en ese archivo. Mientras mayor sea el número de usuarios simultáneos, más se alentará el tiempo de respuesta por el requerimiento del candado.

Muy pocos sistemas de información de misión crítica pueden implantarse con tecnología de servidor de archivo. Entonces ¿por qué son tan populares los sistemas de administración de base de datos del servidor de archivos como el MS *Access*? Primero, las herramientas de servidor de archivo tales como el *Access* pueden usarse para desarrollar aplicaciones bastante robustas para individuos y grupos de trabajo pequeños. Segundo, y quizá más importante, las bases de datos de servidores de archivos como el *Access* pueden usarse rápidamente para construir prototipos de arquitecturas más robustas de sistemas cliente/servidor.

**Arquitecturas de cliente/servidor** El modelo de cómputo distribuido predominante actualmente se llama *cómputo de cliente/servidor* (aunque rápidamente está cediendo te-



rreno a los modelos basados en Internet). Un **sistema cliente/servidor** es una solución en la cual la presentación, la lógica de presentación, la lógica de aplicación, la manipulación de datos y las capas de datos se distribuyen entre las PC cliente y uno o más servidores.

Las computadoras cliente pueden ser cualquier combinación de computadoras personales o estaciones de trabajo, computadoras portátiles “a veces conectadas”, computadoras de mano (por ejemplo, equipos Palm o Windows Mobile), las TV de Web o cualquier dispositivo con procesadores incorporados que pudieran conectarse a la red (por ejemplo, los robots o los controladores en un piso de taller de manufactura). Los clientes pueden ser delgados o pesados. Un **cliente delgado** es una computadora personal que no tiene que ser muy poderosa (o cara) en términos de la velocidad del procesador y de la memoria porque sólo presenta la interfaz (las pantallas) al usuario; en otras palabras, actúa sólo como una terminal. Los ejemplos incluyen las terminales remotas y *X/Windows*.

En el cómputo de cliente delgado, la lógica de aplicación real se ejecuta en un servidor de aplicación remota. Un **cliente pesado** es una computadora personal, una computadora portátil o una estación de trabajo que es típicamente más poderosa (y cara) en términos de la velocidad del procesador, la memoria y la capacidad de almacenamiento. Casi todas las PC se consideran clientes pesados.

En el modelo cliente/servidor, un servidor debe ser más poderoso y capaz que un servidor en el modelo de servidor de archivos. De hecho, una computadora *mainframe* puede desempeñar el papel de servidor en una solución cliente/servidor. Más típicos, sin embargo, son los servidores de red que corren sobre sistemas operativos cliente/servidor, tales como UNIX, Windows Server 2003 o Linux. Pueden usarse varios tipos de servidores en una solución cliente/servidor. Éstos pueden residir en servidores físicos separados o pueden consolidarse en menos servidores:

- Un **servidor de base de datos** aloja una o más bases de datos compartidas (al igual que un servidor de archivos) pero también ejecuta todos los comandos de la base de datos y da servicio a los sistemas de información (a diferencia de un servidor de archivos). La mayoría de servidores de la base de datos alojan un motor de base de datos de SQL como *Oracle*, *Servidor SQL* de Microsoft, o la *DB2 Universal Database* de IBM.
- Un **servidor de transacciones** aloja servicios que finalmente aseguran que todas las actualizaciones de la base de datos para una determinada transacción del negocio tenga éxito o falle como un todo. Los ejemplos incluyen a IBM *CICS* y BEA.
- Un **servidor de aplicación** aloja los servicios y la aplicación de un sistema de información. Debe comunicarse en el extremo frontal con los clientes (para la presentación) y en el extremo trasero con los servidores de base de datos para el acceso a los datos y su actualización. Con frecuencia, un servidor de aplicación se integra con el servidor de transacciones. La mayoría de los servidores de aplicación se basan ya sea en el estándar *CORBA* de uso compartido de objetos o en el estándar *COM+* de Microsoft.
- Un **servidor de correo o servidor groupware** aloja servicios de correo electrónico, calendarización y otras funcionalidades para el grupo de trabajo. Este tipo de funcionalidad realmente puede integrarse en aplicaciones de sistemas de información. Los ejemplos incluyen a *Notes* de Lotus y *Exchange Server* de Microsoft.
- Un **servidor de Web** aloja sitios Web de Internet o intranet. Se comunica con clientes pesados y delgados proporcionando documentos (en formatos tales como el *HTML*) y datos (en formatos tales como *XML*). Algunos servidores Web se diseñan específicamente para alojar aplicaciones de comercio electrónico (por ejemplo, *WebSphere Commerce Business* de IBM).

La arquitectura cliente/servidor viene en varios tipos, cada uno de los cuales merece su propia explicación. Cada uno de estos tipos C/S también se comparan con los demás en la figura 11.4.

**Cliente/servidor: presentación distribuida** La mayoría de las aplicaciones centralizadas de computadora (o mainframe) usan una interfaz de usuario de carácter (CUI), que es demasiado vieja, enredada y torpe cuando se le compara con las interfaces gráficas de usuario de la actualidad (GUI), tales como *Windows* de Microsoft y UNIX *X/Windows* (sin mencionar los navegadores de la Red tales como como el *Firefox* de Mozilla y el *Internet Explorer* de Microsoft). Como las computadoras personales rápidamente reemplazaron a las terminales tontas, los usuarios cada vez se sintieron más cómodos con esta tecnología más nueva. Y a medida que desarrollaron familiaridad y experiencia con herramientas de productividad de la PC, tales como procesadores de texto y hojas de cálculo, quisieron

### sistema cliente/servi-

**dor** Solución de cómputo distribuido, en la cual la presentación, la lógica de presentación, la lógica de aplicación, la manipulación de datos y las capas de datos se distribuyen entre las PC clientes y uno o más servidores.

**cliente delgado** Computadora personal que no tiene que ser muy potente.

**cliente pesado** Computadora personal, computadora portátil o estación de trabajo que es típicamente poderosa.

### servidor de base de

**datos** Servidor que aloja una o más bases de datos.

### servidor de transaccio-

**nes** Servidor que aloja servicios para garantizar que para una transacción todas las actualizaciones de las bases de datos tienen éxito o fallan como un todo.

### servidor de aplicación

Servidor que aloja la aplicación lógica y los servicios de un sistema de información.

### servidor de correo o de groupware

Servidor que aloja servicios de groupware.

### servidor de Web

Servidor que aloja sitios de la red de Internet o intranet.

**presentación distribuida**

Sistema cliente/servidor en el que la presentación y la lógica de presentación se trasladan desde el servidor para residir en el cliente.

que sus aplicaciones heredadas de computación centralizada, tuvieran una apariencia y una percepción similares a las usadas en el modelo GUI.

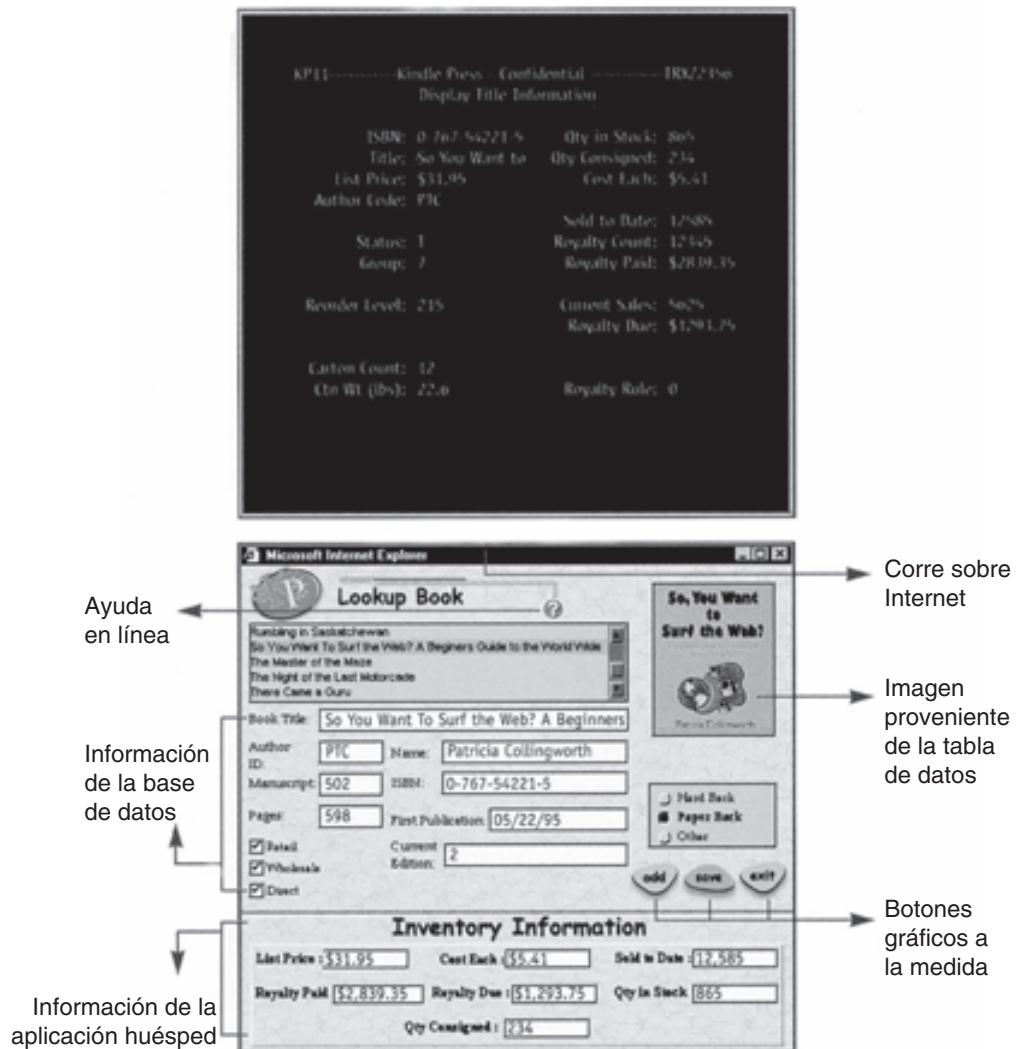
Entra la presentación distribuida. Un sistema de cliente/servidor de **presentación distribuida** es una solución en la cual las capas de presentación y de lógica de presentación se desplazan desde el servidor de un sistema heredado para residir en el cliente. La lógica de la aplicación, la manipulación de datos y las capas de datos permanecen en el servidor (por lo general un servidor mainframe). Algunas veces llamado “el cliente/servidor del pobre”, en esta alternativa se construyen y amplían las aplicaciones de cómputo centralizadas. Básicamente, las viejas CUI son retiradas de las aplicaciones heredadas y se regeneran como GUIs que correrán en la PC. En otras palabras, sólo la interfaz del usuario (o la capa de presentación) se distribuye al cliente.

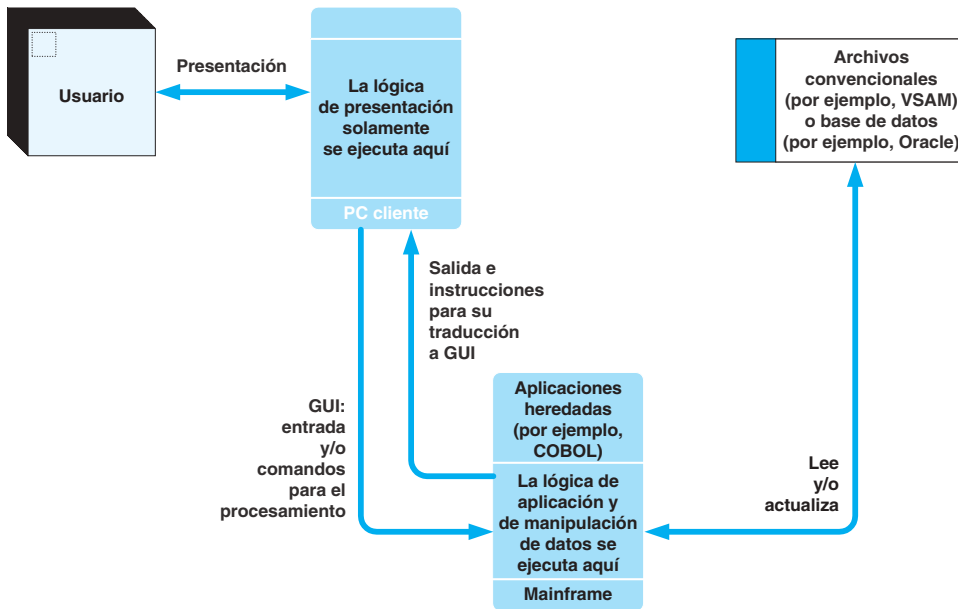
La presentación distribuida ofrece varias ventajas. Primero, puede implantarse con relativa rapidez porque la mayoría de aspectos de la aplicación heredada permanece sin cambios. Segundo, los usuarios obtienen una interfaz de usuario rápida, amigable y familiar a los sistemas heredados; una que como mínimo parece ser algo familiar a sus herramientas de productividad de la PC. Finalmente, la duración de la vida útil de las aplicaciones heredadas puede ampliarse hasta que los recursos justifiquen un nuevo desarrollo mayor de la aplicación. Las desventajas son que la funcionalidad de la aplicación no puede mejorarse de manera significativa y la solución no maximiza el potencial de la computadora de escritorio cliente, usándola sólo para la interfaz del usuario.

Una clase de herramientas CASE, algunas veces llamadas *extractor de pantalla*, automáticamente leen el CUI y generan una GUI de primera capa que puede ser modificada por un editor de la interfaz gráfica del usuario (GUI, por sus siglas en inglés). La figura

**FIGURA 11.6**

Diseño de una GUI a partir de una CUI





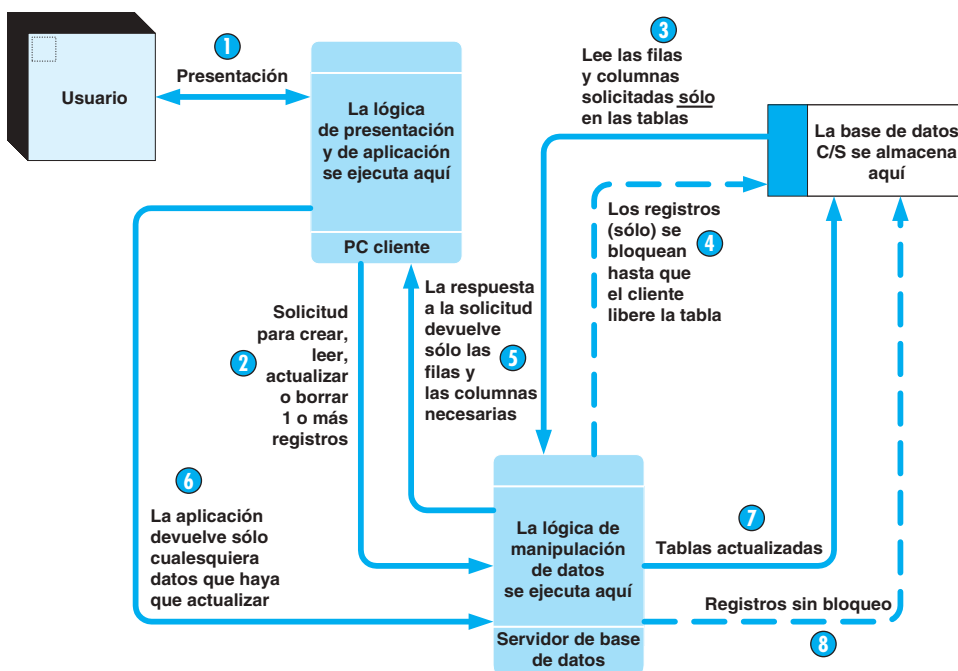
**FIGURA 11.7**  
Sistema cliente/servidor: presentación distribuida

11.6 muestra esta tecnología. La figura 11.7 muestra a un DFD físico para una solución de presentación distribuida.

**Cliente/servidor: datos distribuidos** Ésta es la forma más simple de verdadero cómputo cliente/servidor. Una red de área local generalmente conecta a los clientes con el servidor. Un sistema de **datos distribuidos** cliente/servidor es una solución en la cual los datos y las capas de manipulación de datos reside en el(los) servidor(es), y la lógica de la aplicación, la lógica de presentación y la presentación residen en los clientes. Esto se llama también *cómputo de cliente/servidor de dos capas*. Un sistema de dos capas, de cliente/servidor de datos distribuidos se ilustra como un DFD físico en la figura 11.8.

Es importante entender la diferencia entre los sistemas de servidores de archivos y los sistemas de cliente/servidor de datos distribuidos. Ambos almacenan su base de datos real en un servidor. Pero solamente los sistemas cliente/servidor ejecutan todos los comandos de manipulación de datos (por ejemplo, las instrucciones SQL para crear, leer, actualizar y borrar registros) en un servidor. Recuerde que en los sistemas servidores de archivos,

**datos distribuidos** Sistema cliente/servidor en el cual las capas de datos y de manipulación de datos se colocan en los servidores, y otras capas se colocan en los clientes. También llamado cómputo de cliente/servidor de dos capas.



**FIGURA 11.8**  
Sistema cliente/servidor: datos distribuidos (dos capas)

esas órdenes de manipulación de datos deben ser implantadas en el cliente. Las soluciones de cliente/servidor de datos distribuidos ofrecen varias ventajas sobre las soluciones de servidor de archivos:

- Hay mucho menos tráfico de red porque sólo las solicitudes a la base de datos y los registros de la base de datos que son necesarios se envían hacia y desde las estaciones de trabajo del cliente.
- La integridad de la base de datos es más fácil de mantener. Sólo los registros en uso por un cliente deben bloquearse de manera normal. Otros clientes pueden trabajar simultáneamente sobre otros registros en la misma tabla o base de datos.

La estación de trabajo del cliente todavía debe ser bastante pesada (robusta) para proveer el procesamiento para la capa lógica de la aplicación. Por lo general, esta lógica está escrita en un lenguaje de programación de cliente/servidor tal como *PowerBuilder* de la Corporación Sybase, *C++* o *Visual Basic .NET* de Microsoft. Esos programas deben ser compilados y ejecutados por el cliente. Para mejorar la eficiencia de aplicación y reducir el tráfico de la red, alguna parte de la lógica del negocio puede instalarse en el servidor de la base de datos en forma de procedimientos almacenados (que se analizará en el siguiente capítulo).

El *servidor de la base de datos* es fundamental para esta arquitectura. Los servidores de la base de datos almacenan la base de datos, pero también ejecutan las instrucciones para la base de datos de manera directa en esos servidores. Los clientes simplemente envían sus instrucciones a la base de datos del servidor. El servidor devuelve sólo el resultado del procesamiento de la instrucción a la base de datos; no bases de datos ni tablas completas. Todos los motores de base de datos de extremo alto tales como *Oracle* y el *SQL Server* de Microsoft usan este enfoque. Una arquitectura distribuida de datos puede involucrar más de un servidor de base de datos. Los datos pueden distribuirse a través de varios servidores de base de datos o duplicarse en varios servidores de base de datos.

La desventaja potencial crítica para el cliente/servidor de dos capas es que la lógica de aplicación debe duplicarse y mantenerse así en todos los clientes, quizá centenares o miles. El diseñador debe planear las actualizaciones de la versión y proporcionar controles para asegurar que cada cliente está corriendo la versión más actualizada de la lógica del negocio, así como asegurar que otro software en la PC (comprado o desarrollado internamente) no interfiere con la lógica del negocio.

**Cliente/servidor: Datos y aplicación distribuidos** Cuando el número de clientes crece, a menudo los sistemas de dos capas sufren problemas de desempeño asociados con la ineficiencia de ejecutar toda la lógica de aplicación en los clientes. Asimismo, en los sistemas de procesamiento de transacciones de múltiple usuario (también llamados *procesamiento de aplicación en línea*, u *OLAP*), las transacciones deben ser administradas por el software para asegurar que todos los datos asociados con la transacción se procesan como una unidad individual. Esto generalmente requiere una distribución que use un enfoque de cliente/servidor multicapas. Un sistema de **datos y aplicación distribuidos** de cliente/servidor es una solución en la cual: 1) los datos y las capas de manipulación de datos se colocan en su propio servidor(es), 2) la lógica de la aplicación se coloca en su propio servidor y 3) sólo la lógica de presentación y la presentación se colocan en los clientes. Esto se llama también *cómputo de cliente/servidor de tres capas* o *n capas*.

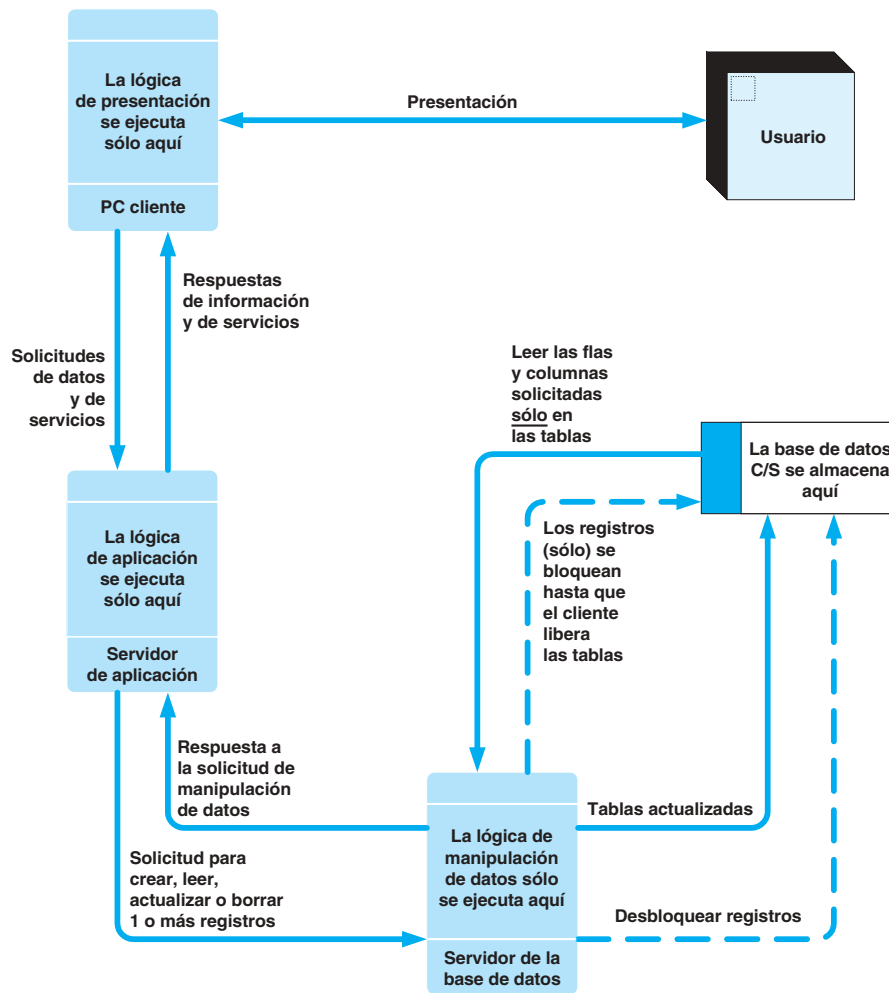
La solución de tres capas de cliente/servidor usa los mismos servidores de la base de datos que en el enfoque de dos capas. Además, el sistema de tres capas introduce un servidor de aplicación y/o de transacción. Al mover la lógica de la aplicación a su propio servidor, esa lógica ahora sólo necesita ser mantenida en el servidor. La solución de tres capas se ilustra como un diagrama físico de flujo de datos en la figura 11.9.

La lógica de tres capas de cliente/servidor puede ser escrita y repartida a través de servidores múltiples que usan lenguajes tales como Microsoft *Visual Basic .NET* y *C++* en combinación con un monitor de transacción. Las herramientas de extremo alto como *Forté* proveen una oportunidad aun mayor para distribuir los datos y la lógica de aplicación a través de una red compleja. Al igual que con la solución del servidor de base de datos, parte de la lógica de negocios podría distribuirse al servidor de base de datos en forma de procedimientos almacenados.

En un sistema de tres capas, los clientes ejecutan un mínimo de los componentes del sistema completo. Sólo la interfaz del usuario y alguna lógica de aplicación relativamente estable o personal necesitan ser ejecutadas en los clientes. Esto simplifica la configuración y la administración del cliente.

#### datos y aplicación distribuidos

Sistema cliente/servidor en el que las capas de datos y de manipulación se colocan en su(s) propio(s) servidor(es), la lógica de aplicación se coloca en su propio servidor y la lógica de presentación y la presentación se colocan en los clientes. También llamado *cómputo de cliente/servidor de tres capas* o de *n capas*.

**FIGURA 11.9**

Sistema cliente/servidor: datos y aplicaciones distribuidos (tres capas)

El inconveniente más grande del cliente/servidor de tres capas es su complejidad en diseño y desarrollo. El aspecto más difícil de un diseño de aplicación de cliente/servidor de tres capas es la partición. La **partición** es el acto de determinar cómo distribuir o duplicar los componentes de aplicación de la mejor manera a través de la red. Por fortuna, las herramientas CASE mejoran de manera constante para proveer mayor asistencia para la partición.

**Arquitecturas de cómputo basadas en Internet** Algunos consideran que las arquitecturas de sistemas basadas en Internet son la evolución más reciente de cliente/servidor. En esta sección presentamos alternativas de cómputo basadas en Internet como una forma completamente diferente de arquitectura distribuida que está rápidamente cambiando las corrientes de pensamiento de los analistas de sistemas y los tecnólogos de información sobre el diseño.

Un **sistema de cómputo basado en red** es una solución multicapas en la cual la presentación y las capas de lógica de presentación se implantan en los navegadores de Internet del lado del cliente usando el contenido descargado de un servidor de Web. Entonces la capa lógica de presentación se conecta a la capa lógica de la aplicación que corre en un servidor de aplicación, el cual subsiguientemente se conecta al servidor(es) de la base de datos en la parte trasera. ¡Piense en esto! ¡Todos los sistemas de información corriendo en navegadores (finanzas, recursos humanos, operaciones), todos ellos! El comercio electrónico es parte de esta fórmula, y al tiempo que este libro entra a la prensa, las aplicaciones de comercio electrónico acaparan la mayoría de la atención. Pero las mismas tecnologías de Internet que se usan para construir las soluciones de comercio electrónico se están usando para remodelar los sistemas de información internos de la mayoría de negocios; les llamamos *negocios electrónicos* (aunque ese término también está sujeto a múltiples interpretaciones). La computación en redes es, desde nuestro punto de vista, un alejamiento fundamental de lo que nosotros describimos como cliente/servidor.

**partición** Acto de determinar la mejor forma de distribuir o duplicar los componentes de aplicaciones a través de una red.

**sistema de cómputo basado en red** Solución de varias capas en la cual las capas de presentación y de lógica de presentación se ejecutan en navegadores Web del lado del cliente, usando el contenido descargado de un servidor Web.

Muy pocas tecnologías nuevas han presenciado un crecimiento tan explosivo en los negocios y en la sociedad como Internet o la Red (*World Wide Web*). Internet amplía el alcance de nuestra información y los sistemas de procesamiento de transacciones para incluir a los clientes potenciales, los clientes, los socios, los empleados localizados a distancia, los proveedores, el gobierno y aun los competidores. A finales de la década de 1990, Internet se usaba principalmente para establecer la presencia de una compañía en un mercado virtual y diseminar información pública acerca de los productos y los servicios y proveer un cimiento nuevo para el servicio centrado en el cliente. Hoy, sin embargo, la mayoría de los negocios se centran en el desarrollo de soluciones de comercio electrónico que permitan a los clientes interactuar de manera directa y conducir negocios en la Red (tal como las compras directas del consumidor). Hasta hemos visto la invención de los negocios virtuales, un negocio que “hace negocios” enteramente en la Red, como Amazon.com (libros y medios), ETrade (acciones y bonos), eBay (subastas), y Buy.com (aparatos y aditamentos electrónicos). Uno de los debates más intrigantes es si estas compañías virtuales de “click y mortero” pueden obtener ganancias y si realmente compiten con compañías “de cemento y ladrillo” más tradicionales; muchas de las cuales están diversificándose a toda velocidad para entrar en los cibermercados.

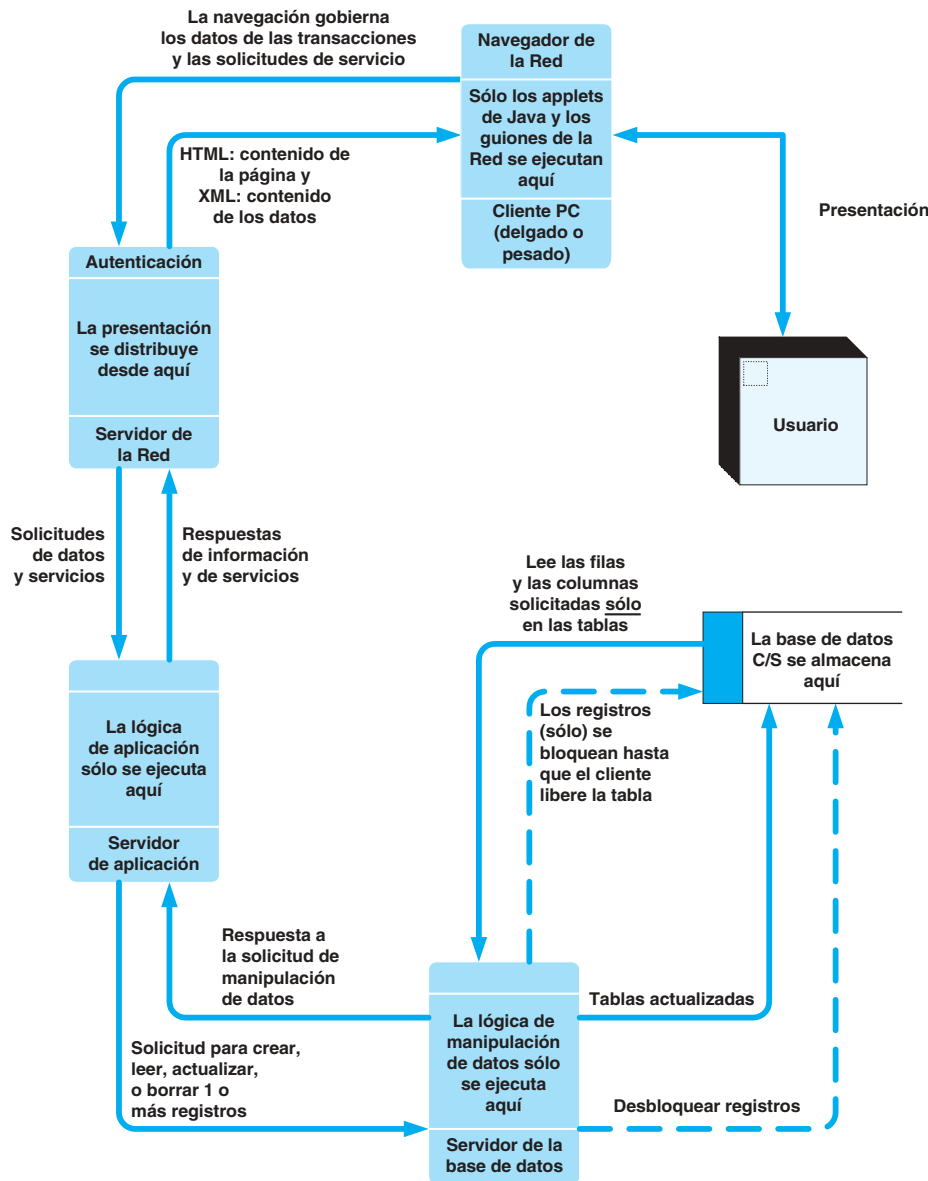
Pero el mayor potencial de esta tecnología de Internet realmente puede ser su aplicación a las aplicaciones tradicionales de los sistemas de información y el desarrollo de intranets. Una **intranet** es una red segura, generalmente corporativa, que usa la tecnología de Internet para integrar la computación de un escritorio, de los grupos de trabajo y de las empresas en un marco integral individual. Todo corre en (o al menos proviene de) un navegador; sus aplicaciones de productividad tales como el procesamiento de texto y las hojas de cálculo; todas y cada una de las aplicaciones tradicionales de sistemas de información que usted necesita para su trabajo (finanzas, adquisición, recursos humanos, etc.); todo el correo electrónico, la calendarización y los servicios del grupo de trabajo (permitiendo, por ejemplo, las reuniones virtuales y la edición en grupo de documentos) y, por supuesto, todos los enlaces externos de Internet que son relevantes para su trabajo.

El atractivo de este concepto no deberá ser difícil de entender. “La página de inicio” de cada empleado es un *portal* de todos los sistemas de información y los servicios que él o ella necesitan para cumplir con todo su trabajo. Puesto que todo corre en un navegador de Internet, ya no es necesario preocuparse por, o desarrollar, diferentes arquitecturas de múltiples computadoras (Intel contra Motorola contra RISC) o preocuparse acerca de diferentes sistemas operativos de escritorio. Un diagrama de flujo de datos físicos para el cómputo de redes se muestra en la figura 11.10. Observe que un servidor de Web se añade al modelo anterior de tres capas. El DFD también muestra ambas dimensiones del comercio electrónico (del negocio al consumidor) y de negocio electrónico (negocio a negocio) del cómputo en red.

¿Todo esto suena demasiado bien para ser verdad? ¿Algo como un ciber-Camelot? Para cuando usted lea esto, probablemente nuestra propia institución haya implantado su primer sistema de información de misión crítica de negocios electrónicos. Purdue, como todas las empresas, tiene un área de adquisición (o de compra). Compramos todo, desde lápices y mobiliario hasta computadoras e isótopos radiactivos; literalmente decenas de miles de suministros diferentes, materiales y otros productos. Mientras este libro se publica, Purdue habrá rediseñado su sistema de adquisición para ser una aplicación de combinación de intranet/Internet/extranet. Así es como funciona:

La aplicación completa corre en un navegador de la Red. Cualquier empleado de la universidad, una vez identificado, puede iniciar una requisición de compra por su navegador de la Red (la dimensión de la intranet). Los empleados hasta pueden comprar artículos en una plaza-Web de proveedores aprobados con los cuales la universidad tiene contratos en vigor (la dimensión de Internet). Cuando se haga una requisición, será lo suficientemente inteligente para saber quién debe aprobarla (en cada nivel) basándose en el costo y en el tipo de los artículos ordenados. El sistema podrá revisar de manera automática en busca de fondos disponibles para pagar la compra. Los empleados podrán auditar el flujo electrónico de la requisición a través del proceso de aprobación y en el estatus de la orden de compra. Los gerentes podrán revisar el flujo de aprobación para obtener insumos adicionales según se necesite. La mayoría de las órdenes finales serán transmitidas en forma electrónica sobre una extranet segura negocio a negocio entre la universidad y sus proveedores.

**intranet** Red de servidores que usa la tecnología de Internet para integrar el cómputo de escritorio, grupos de trabajo y cómputo empresarial.



**FIGURA 11.10**  
Sistema cliente/servidor: datos y aplicaciones distribuidos (tres capas)

Cuando los bienes ordenados sean recibidos, el destinatario hará el acuse de recibo vía el mismo sistema basado en la Web, y el pago se hará vía fondos electrónicos en la mayoría de los casos. En cualquier momento los gerentes podrán generar información útil de adquisición para empleados, departamentos, proveedores, ¡cualquier cosa! ¿Una oficina sin documentos? ¡Muy cerca!

Estas inquietantes posibilidades de desarrollo de sistemas están siendo fomentadas por algunas tecnologías emergentes fundamentales que usted debería integrar a su plan de estudios curricular si es posible:

- Es posible que la elección del lenguaje de programación para la lógica de aplicación en las arquitecturas de cómputo de la Red sea *Java*. En esencia, *Java* es un lenguaje de programación razonablemente independiente de la plataforma y está diseñado en forma específica para explotar tanto Internet como los estándares de programación orientados a objetos. *Java* está diseñado para ejecutarse en su navegador de Internet, haciéndolo menos susceptible a las diferencias en las plataformas de cómputo y en los sistemas operativos (pero todavía no es perfecto).

- La elección del lenguaje para la interfaz para las capas de presentación y de lógica de la presentación en las arquitecturas de cómputo en red actualmente es *HTML* (*HyperText Markup Language*). El *HTML* se usa para crear las páginas que corren en su navegador. Pronto otro jugador dominará en esta capa: *XML* (*Extensible Markup Language*). Este estándar ampliamente aceptado también permite a los desarrolladores definir la estructura de los datos para ser pasados a las páginas de Web, un requerimiento crítico para los sistemas de información de comercio electrónico basado en la Web y basados en intranets. El *XML* al cabo del tiempo puede reemplazar al *HTML*, o los dos estándares pueden fusionarse en un lenguaje muy poderoso.
- Al igual que con los sistemas de información tradicionales, las capas de datos y de manipulación de datos probablemente continuarán siendo implantados con los motores SQL de la base de datos.
- Los navegadores de Internet continuarán siendo importantes. De hecho, los navegadores de Internet finalmente pueden ser más importantes que su elección de un sistema operativo de escritorio. ¿Sería sorprendente que *Windows*, con cada nueva versión, se viera y se percibiera más como un navegador?

Todas estas soluciones de Internet y de intranet involucran tecnologías y estándares de vanguardia que sin duda han cambiado aun después de que se escribieran estas líneas. Nadie duda que los vendedores de tecnología desempeñarán un papel significativo en la evolución de la tecnología. A pesar del gran despliegue de las tecnologías específicas, esperamos que el Internet y las intranets se conviertan en los modelos de arquitectura más comunes para los sistemas de información del mañana.

Así es que, ¿dónde estamos en nuestro estudio de arquitectura de tecnologías de la información? Usted ha aprendido que existen varias opciones de redes y sistemas distribuidos para los sistemas de información modernos. La mayoría puede cabalmente clasificarse, ya sea como arquitecturas de cómputo cliente/servidor o de red. Para estar seguro, hay mucho más que usted debe aprender acerca de la tecnología de comunicaciones subyacente, pero que es tema de cuando menos un libro adicional. El plan de estudios de licenciatura en Sistemas de Información del modelo de la AIS/ACM/AITP recomienda que todos los egresados de sistemas de información tomen cuando menos un curso de arquitectura de tecnología de la información y un curso básico de comunicaciones de datos.

## > Arquitecturas de datos: bases de datos relacionales distribuidas

La tecnología subyacente de cómputo cliente/servidor y de redes ha posibilitado distribuir los datos sin pérdida del control. Este control se logra mediante las ventajas de la tecnología de bases de datos relacional distribuida. Una *base de datos relacional* almacena datos en forma tabular. Cada archivo se implanta como una tabla. Cada campo es una columna en la tabla. Cada registro del archivo es una fila en la tabla. Los registros relacionados entre dos tablas (por ejemplo, CLIENTES y ÓRDENES) se implantan al duplicar intencionalmente las columnas en las dos tablas (en este ejemplo, el NÚMERO DEL CLIENTE se almacena en las tablas de CLIENTES y ORDENA). Una *base de datos relacional distribuida* distribuye o duplica las tablas en los múltiples servidores de bases de datos localizados en regiones geográficamente importantes (tales como las diferentes regiones de ventas). El software requerido para implantar las bases de datos relacionales distribuidas se llama *sistema de administración de base de datos relacional distribuido*. Un **sistema de administración de base de datos relacional distribuido** (o *RDBMS distribuido*) es un programa de software que controla el acceso y el mantenimiento de los datos almacenados en el formato relacional. También suministra facilidades de respaldo, recuperación y seguridad. Se llama algunas veces *sistema de administración de base de datos cliente/servidor*.

En un RDBMS distribuido, el motor subyacente de la base de datos que procesa todos los comandos de la base de datos se ejecuta en el servidor de la base de datos. Esta estructura reduce el tráfico de datos en la red. Ésta es una ventaja significativa para todos con excepción de los sistemas más pequeños (medidos por el número de usuarios). Un sistema de administración de base de datos relacional distribuido también provee respaldo, recuperación, seguridad, integridad y procesamiento más sofisticados (aunque las diferencias parecen disminuir con cada lanzamiento nuevo de PC RDBMS).

Los ejemplos de RDBMS distribuidos incluyen a *Oracle* de la Corporación Oracle, la familia *Universal Database DB2* de IBM, *SQL Server* de Microsoft y *Sybase* de la Corporación Sybase. La mayoría de RDBMS soportan dos tipos de datos distribuidos:

### sistema de administración de base de datos relacional distribuido

Software con el que se implantan las bases de datos relacionales distribuidas.



- La *partición de datos* verdaderamente distribuye filas y columnas a los servidores dedicados de bases de datos con poca o ninguna duplicación entre servidores. Columnas diferentes pueden asignarse a diferentes servidores de bases de datos (*partición vertical*) o filas diferentes en una tabla pueden asignarse a diferentes servidores de bases de datos (*partición horizontal*).
- La *replicación de datos* duplica algunas o todas las tablas (filas y columnas) en más de un servidor de la base de datos. Tablas enteras pueden duplicarse en algunos servidores de bases de datos, mientras que subconjuntos de filas en una tabla pueden duplicarse para otros servidores de bases de datos. El RDBMS con tecnología de replicación no sólo controla el acceso y la administración de cada base de datos del servidor de bases de datos, sino que también propaga actualizaciones realizadas un servidor de bases de datos a cualquier otro servidor de bases de datos donde los datos están duplicados.

Para una determinada aplicación de sistemas de información, la arquitectura de datos debe especificar la tecnología RDBMS y el grado en el cual los datos se repartirán o replicarán. Una forma de documentar estas decisiones es registrarlas en los almacenamientos de datos físicos, como se muestra a continuación. Observe cómo usamos el área de ID para indicar los códigos de partición (P) y de replicación (M para la copia maestra y R para la copia replicada). En el caso anterior, deberemos especificar cuáles filas y/o columnas deben repartirse en la base de datos física.

Almacenamiento de datos lógicos	Almacenamiento de datos físicos con el uso de partición	Almacenamiento de datos físicos con el uso de replicación
<div style="border: 1px solid black; padding: 5px;"> <div style="background-color: #0070C0; color: white; padding: 2px 5px; display: inline-block;">1</div> <div style="margin-left: 10px;"><b>CLIENTES</b></div> </div>	<div style="border: 1px solid black; padding: 5px;"> <div style="background-color: #0070C0; color: white; padding: 2px 5px; display: inline-block;">1P.#</div> <div style="margin-left: 5px;"><b>Oracle 7: CLIENTES DE LA REGIÓN 1</b></div> </div> <hr/> <div style="border: 1px solid black; padding: 5px;"> <div style="background-color: #0070C0; color: white; padding: 2px 5px; display: inline-block;">1P.#</div> <div style="margin-left: 5px;"><b>Oracle 7: CLIENTES DE LA REGIÓN 2</b></div> </div>	<p>No es aplicable. Las sucursales no necesitan acceso a los datos de los clientes fuera de su propia región de ventas.</p>
<div style="border: 1px solid black; padding: 5px;"> <div style="background-color: #0070C0; color: white; padding: 2px 5px; display: inline-block;">2</div> <div style="margin-left: 10px;"><b>PRODUCTOS</b></div> </div>	<p>No es aplicable. Todas las sucursales necesitan acceso a los datos para todos los productos, independientemente de la región de ventas.</p>	<div style="border: 1px solid black; padding: 5px;"> <div style="background-color: #0070C0; color: white; padding: 2px 5px; display: inline-block;">2M</div> <div style="margin-left: 5px;"><b>Oracle 8i: PRODUCTOS (Maestro)</b></div> </div> <hr/> <div style="border: 1px solid black; padding: 5px;"> <div style="background-color: #0070C0; color: white; padding: 2px 5px; display: inline-block;">2R</div> <div style="margin-left: 5px;"><b>Oracle 8i: PRODUCTOS (copia replicada)</b></div> </div>

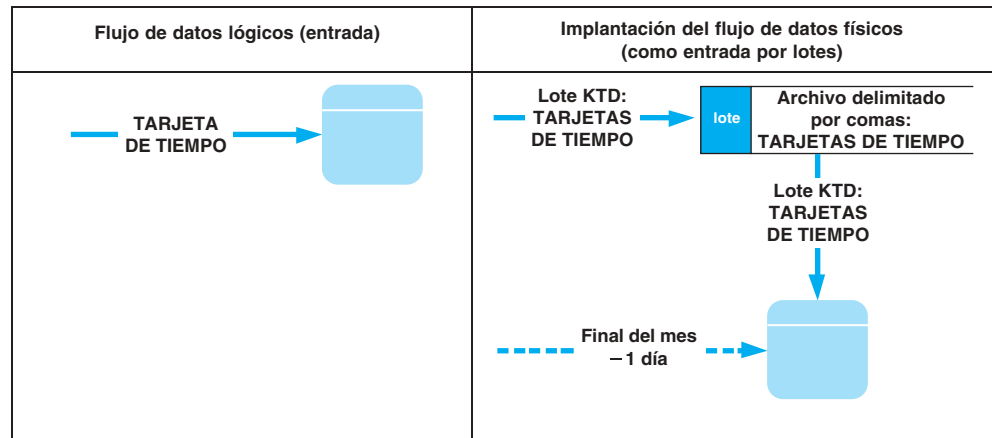
La arquitectura de DATOS de una aplicación se selecciona basándose ya sea en el modelo de cómputo de red o el de cliente/servidor y la tecnología de bases de datos necesaria para soportar ese modelo. Muchas organizaciones han estandarizado su preferencia tanto de sus PC RDBMS como de sus RDBMS distribuidas para empresas. Por ejemplo, SoundStage ha definido como sus estándares a Microsoft Access y *SQL Server*. Por lo general, un calificado administrador de bases de datos debería ser incluido en cualquier debate acerca de la tecnología de base de datos que debe usarse y las implicaciones de diseño para cualesquiera bases de datos que usarán esa tecnología.

### > **Arquitectura de interfaces: entradas, salidas y middleware**

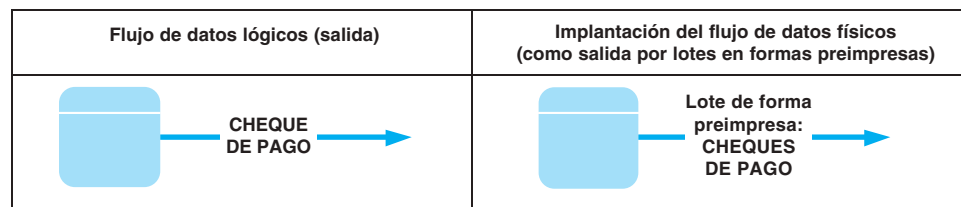
Debe tomarse otra decisión fundamental en tecnologías de la información con respecto a las entradas, salidas y conectividad entre sistemas. La decisión solía ser simple: entradas por lotes contra entradas en línea. Hoy debemos considerar las alternativas modernas tales como la identificación automática, la entrada de datos con pluma, las diversas interfaces gráficas de usuario, el intercambio electrónico de datos, las imágenes y el reconocimiento de voz, entre otros. Examinemos brevemente estas alternativas y sus diseños de DFD físicos.

**Entradas o salidas por lotes** En el procesamiento por lotes, las transacciones se acumulan en lotes para su procesamiento intermitente. Las entradas por lotes se procesan para actualizar las bases de datos y producir las salidas apropiadas. La mayoría de salidas tienden a ser generadas en papel o microficha de acuerdo con un calendario. Otras podrían generarse a petición o dentro de un cierto tiempo especificado (por ejemplo, 24 horas).

Contrario a la creencia popular, las tecnologías de entrada por lotes no son en realidad obsoletas. Hoy usted raramente ve tarjetas perforadas y lotes en cintas, pero algunos requerimientos aplicativos se prestan para su procesamiento por lotes. Quizá las entradas lleguen en lotes naturales (por ejemplo, el correo), o quizá las salidas son generadas en lotes naturales (por ejemplo, las facturas). Muchas organizaciones todavía recolectan y procesan tarjetas de asistencia por lotes. Hay, sin embargo, una tendencia definida a cambiar de la entrada por lotes hacia los enfoques en línea. Mientras tanto, el archivo de registro directo sobre disco es lo más común, y su construcción de flujo de datos físicos se vería como se muestra en seguida. Primero, observe que el nombre lógico es singular, pero el nombre del lote es plural. También observe que el lote va a un almacenamiento temporal de datos, el cual es leído por un proceso de nómina activado por la fecha.



La salida por lotes es realmente otro asunto. Muchas aplicaciones se prestan a la salida por lotes. Los ejemplos incluyen la generación de facturas, estados de cuenta, boletas de calificaciones, cheques de salario, formas de impuesto W-2 y muchos otros. Las salidas por lotes a menudo comparten una característica física común: el uso de una forma preimpresa. No debería ser difícil para usted visualizar una forma preimpresa que se alimenta a la impresora para producir cualesquiera de los ejemplos de salida antes mencionados. Una construcción de flujo de datos físicos se vería algo así como lo siguiente. Una vez más, observe el nombre plural que refleja el procesamiento por lotes.

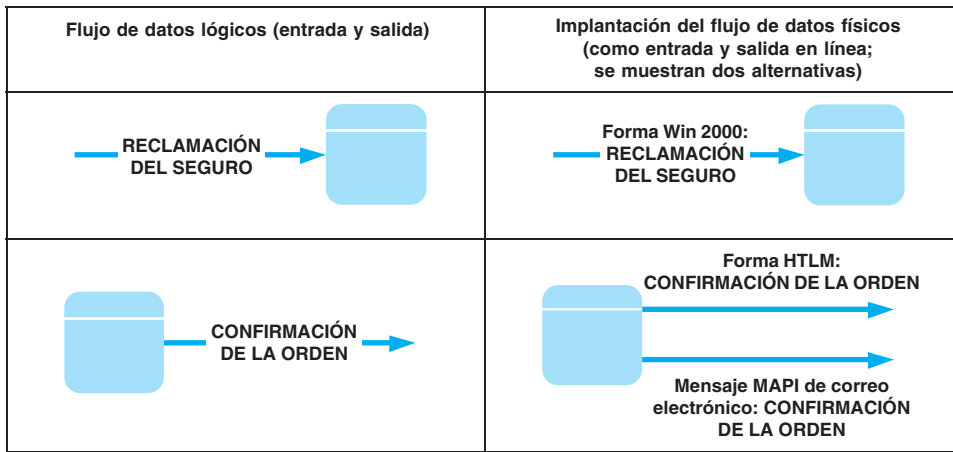


A medida que los más viejos sistemas basados en lotes se convierten en candidatos para ser reemplazados, deberán explorarse otras alternativas físicas de implantación.

**Entradas y salidas en línea** La mayoría de los sistemas ha evolucionado lentamente del procesamiento por lotes al procesamiento en línea o en tiempo real. Las entradas y salidas en línea proporcionan un diálogo más natural entre el usuario y las aplicaciones

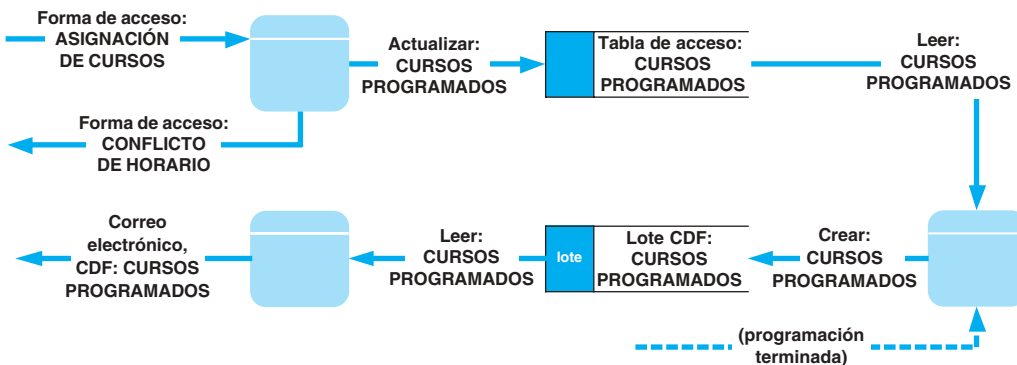
de la computadora. También proveen información retroactiva casi inmediata en respuesta a transacciones, problemas y consultas. En la acelerada economía de la actualidad, lo mejor es procesar la mayoría de transacciones comerciales y de consulta tan pronto como sea posible. Los errores se identifican y corrigen más rápido porque no hay margen de tiempo entre el ingreso de datos y la entrada (como fue el caso en el procesamiento por lotes). Además, los métodos en línea permiten mayor interacción humana en la toma de decisiones.

Hoy muchos de los sistemas están siendo diseñados para el procesamiento en línea, aun si los datos llegan naturalmente en lotes. Técnicamente, todas las aplicaciones con GUI y en Red son en línea o de tiempo real, y como ya hemos aprendido que esas arquitecturas se prefieren en la computación de cliente/servidor y de red, entonces podemos esperar que la mayoría de flujos de datos físicos se implemente con algún tipo de tecnología de GUI. Los diseños de flujo de datos físicos se parecerían a lo siguiente. Para la salida física observe que dos formatos son posibles. Pudimos haber sumado los símbolos de conjunción (capítulo 8) para hacer los flujos mutuamente contingentes (se requieren ambos) o excluyentes (uno u otro, pero no ambos).



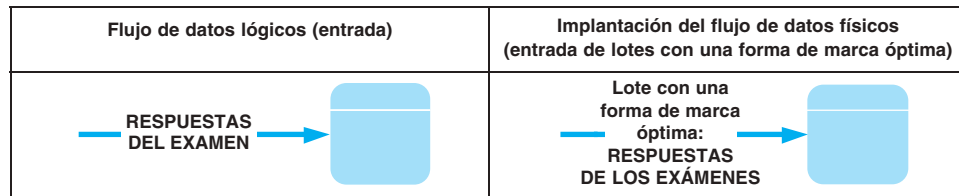
**Lote remoto** El lote remoto combina los mejores aspectos de las entradas y las salidas de lote y en línea. Las computadoras en línea distribuidas manejan la entrada y la edición de datos. Las transacciones editadas se recolectan en un archivo de lotes para la posterior transmisión a los servidores que procesan el archivo como un lote. Los resultados casi siempre se transmiten como un lote de regreso a las computadoras originales.

El lote remoto no es en realidad una alternativa nueva, pero las computadoras personales le han dado nueva vida a esta opción. Por ejemplo, cada semestre, uno de los colegas de los autores usa un programa en Microsoft Access para ingresar y probar la viabilidad de un horario de clases para su departamento académico. Cuando termina, genera un archivo delimitado por comas para transmitirlo a la unidad académica de planificación para el procesamiento por lotes. El modelo físico entero de entradas se ve como esto:



Los lotes remotos con el uso de PCs deberán obtener un nuevo impulso con los avances en la tecnología de las computadoras portátiles y de mano. Estas computadoras que pesan de cuatro onzas a cuatro libras pueden usarse para recolectar lotes de todo tipo de datos, desde realizar un inventario hasta solicitudes de hipoteca. Las entradas se recolectan como lotes remotos en el dispositivo para su posterior transmisión como lote.

**Ingreso de datos sin teclas (e identificación automática)** Teclear datos siempre ha sido una fuente importante de errores en las entradas de datos a la computadora (y las consultas). Cualquier tecnología que reduce o elimina la posibilidad de errores de teclado debería considerarse para el diseño de sistemas. En los sistemas de lotes, los errores de teclado pueden eliminarse a través de la tecnología de lectura óptica de caracteres (OCR, por sus siglas en inglés) y de lectura óptica de marcas (OMR, por sus siglas en inglés). Ambos son todavía opciones viables para diseño de entradas. El diseño de flujo de datos físicos se muestra enseguida.

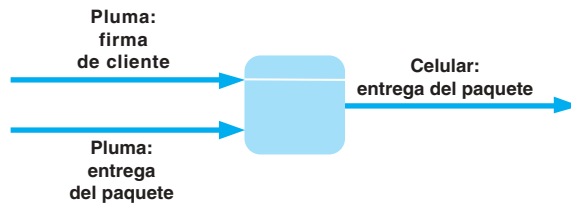


Los avances auténticos en la entrada de datos sin teclear provienen de los sistemas en línea en forma de sistemas de autoidentificación. Por ejemplo, los esquemas de código de barras (como los códigos universales de productos que son comunes en la industria de ventas al menudeo) están ampliamente disponibles para muchas aplicaciones modernas. Por ejemplo, Federal Express elabora una etiqueta basada en códigos de barra para cada uno de los paquetes cada vez que usted entrega un paquete a un centro para su entrega. Los códigos de barras pueden ser leídos y rastreados a medida que el paquete se mueve por todo el país hasta su destino final. La tecnología del código de barras está mejorando de manera constante para comprimir mayores cantidades de datos en etiquetas más pequeñas. El diseño de flujo de datos físicos se muestra enseguida. (El proceso físico receptor sería denominado como la función que realiza.)



**Entradas con pluma** A medida que los sistemas operativos basados en plumas (por ejemplo, el sistema operativo *Palm* y el *Windows Mobile* de Microsoft) están más ampliamente disponibles y usados, y las herramientas para construir aplicaciones basadas en pluma se hacen disponibles y se estandarizan; esperamos ver más diseños de sistema que aprovechen esta tecnología.

Algunos negocios ya usan esta tecnología para la recolección remota de datos. Por ejemplo, UPS usa sistemas de computadora portátil basados en plumas para ayudar a rastrear paquetes a través del sistema de entrega. El conductor introduce el número de guía del paquete en la computadora de tableta. El cliente firma en la pantalla de la computadora en el área indicada. Cuando el conductor regresa al camión y coloca la computadora de tableta de nuevo en su soporte, los datos actualizados de la entrega se transmiten por un módem celular al centro de distribución donde el sistema de rastreo del paquete actualiza la base de datos (finalmente permite al remitente saber que usted ha recibido el paquete vía una simple consulta en Internet).



**Mensajería electrónica y tecnología de trabajo en grupo** ¡El correo electrónico ha crecido! Ya no es sólo una forma para comunicarse con mayor eficacia, los sistemas de información están siendo diseñados para incorporar directamente la tecnología. Por ejemplo, Microsoft *Exchange Server* y *Lotus Notes* de IBM permiten la construcción de formas electrónicas inteligentes que pueden integrarse en cualquier aplicación. Los servicios básicos de mensajería también pueden integrarse en aplicaciones.

Por ejemplo, cualquier empleado vía una forma basada en el correo electrónico podría iniciar solicitudes de viaje. El sistema toma los datos suministrados en la forma y sigue reglas predefinidas para dirigir automáticamente la petición a las personas que toman las decisiones apropiadas. Por ejemplo, las solicitudes de viaje de menor costo podrían dirigirse directamente a un ejecutivo del negocio. Las peticiones más caras podrían dirigirse primero a un director de departamento para su aprobación y luego a un ejecutivo del negocio. Por último, las formas aprobadas pueden ingresarse de manera automática al sistema de información apropiado de procesamiento de reembolsos para su procesamiento normal. Y en cada paso, el sistema de mensajería informa del avance por medio del correo electrónico al empleado que hizo la solicitud. Un DFD físico que incluía una implantación de mensaje por correo electrónico se presentó anteriormente.

**El intercambio electrónico de datos** Los negocios que operan en muchas ubicaciones y los negocios que buscan un intercambio más eficiente de transacciones y datos con otros negocios a menudo utilizan el intercambio electrónico de datos. El **intercambio electrónico de datos (EDI)**, por sus siglas en inglés) es el flujo electrónico estandarizado de transacciones de negocios o datos entre negocios. Por lo común, muchos negocios deben comprometerse a un formato de datos estándar para hacer factible el intercambio electrónico de datos.

Con el EDI, un negocio puede eliminar su dependencia de los documentos de papel y del correo. Por ejemplo, la mayoría de las universidades ahora aceptan puntuaciones de pruebas de SAT o ACT vía el EDI de los centros nacionales de prueba. Esto se ha hecho posible porque los servicios escolares de la universidad han accedido a un formato estándar para estas puntuaciones de pruebas.

**intercambio electrónico de datos (EDI)** Flujo electrónico estandarizado de transacciones de negocios o de datos entre empresas.



**Las imágenes y el intercambio de documentos** Otra tecnología emergente de E/S se basa en las imágenes y el intercambio de documentos. Esto es similar al intercambio electrónico de datos, excepto que se transmiten y se reciben las imágenes reales de las formas y los datos. Es particularmente útil para aplicaciones en las cuales se requieren imágenes de formas o gráficos. Por ejemplo, la industria de seguros ha progresado a grandes pasos en la transmisión, almacenaje y uso de imágenes recuperadas por medios electrónicos. Otras aplicaciones de imágenes combinan datos con cuadros o gráficas. Por ejemplo, una aplicación policiaca puede almacenar, transmitir y recibir imágenes fotográficas y huellas digitales.

**Middleware** La mayor parte de las anteriores secciones se centraban en entradas y salidas: la interfaz del usuario. Pero muchos diseños de sistema requieren flujos de datos físicos de proceso a proceso.

**middleware** Utilería de software que permite la comunicación entre distintos procesadores en un sistema.

Anteriormente en este capítulo describimos diversos escenarios de cómputo cliente/servidor y de redes que en forma automática incluyen flujos de datos de proceso a proceso porque las computadoras clientes y los servidores deben hablar entre sí. Las computadoras hacen esto a través del *middleware*. El *middleware* es la utilería de software que permite la comunicación entre procesadores diferentes en un sistema. Puede incorporarse en los sistemas operativos respectivos o añadirse a través de productos de *middleware* comprados. Los productos de *middleware* permiten a los programadores ignorar los protocolos de comunicación subyacentes.

Se dice que el *middleware* es la “diagonal” de “cliente/servidor”. Hay tres clases de *middleware* que casualmente corresponden a las tres capas de nuestro marco de sistemas distribuidos: lógica de presentación, lógica de la aplicación y lógica de manipulación de datos:

- El *middleware de presentación* le permite a un programador construir componentes de la interfaz de usuario que pueden hablar con los navegadores de Internet o una GUI de escritorio. Por ejemplo, el HTTP permite que el programador se comunique con un navegador de Red a través de una Interfaz de Programador de Aplicaciones (API, por sus siglas en inglés) estándar.
- El *middleware de aplicación* permite que dos procesos escritos por programadores para procesadores diferentes se comuniquen entre sí del modo que sea más conveniente para la aplicación completa. El *middleware de aplicación* es esencial para el desarrollo de aplicaciones multicapas. Los ejemplos de *middleware de aplicación* son numerosos: las llamadas a procedimiento remoto (RPCs), las colas de mensajes y los corredores de petición de objetos.
- El *middleware de la base de datos* permite que un programador envíe los comandos de SQL al motor de la base de datos para su procesamiento a través de un API estándar.

Otros tipos comunes de *middleware* son ODBC (Object Database Connectivity, Conectividad Abierta de la Base de Datos) y JDBC (Javabeen Database Connectivity, Conectividad de la Base de Datos tipo Java), los cuales automáticamente traducen los comandos de SQL de un servidor de base de datos para usarse en otro servidor de bases de datos (por ejemplo, *Oracle* para el *SQL Server*, o viceversa).

En un diagrama de flujo de datos físicos, el *middleware* puede dibujarse especificando el nombre de la clase del *middleware* en el flujo de datos físicos (por ejemplo, ODBC).

## > Arquitecturas de proceso: el ambiente de desarrollo de software

La arquitectura de proceso de una aplicación se define en términos de los lenguajes y las herramientas de software que se usarán para desarrollar la lógica de los negocios y los programas de aplicación para ese proceso. Por lo común, esto se expresa como un menú de elecciones, porque los ambientes diferentes de desarrollo de software son adecuados para aplicaciones diferentes. Un **ambiente de desarrollo de software (SDE)**, por sus siglas en inglés) es un lenguaje y un juego de herramientas para la construcción de aplicaciones de sistemas de información. Una forma de clasificar los SDE es de acuerdo con el tipo de arquitectura de cómputo cliente/servidor o de red que soportan.

**SDE para la presentación de cómputo centralizado y distribuido** No hace mucho tiempo, el ambiente de desarrollo del software para el cómputo centralizado fue muy simple. Consistía de lo siguiente:

- Un editor y un compilador, generalmente *COBOL*, para escribir programas.
- Un monitor de transacción, generalmente *CICS*, para manejar cualesquiera pantallas terminales y transacciones en línea.
- Un sistema de administración de archivos, tal como el *VSAM*, o un sistema de administración de base de datos, como *DB2*, para manejar los datos almacenados.

¡Eso era todo! Ya que todas estas herramientas se ejecutaban en el *mainframe*, sólo que el sistema operativo de la computadora (la mayoría de las veces, *MVS*) era crítico.

La computadora personal llevó muchas nuevas herramientas de desarrollo de *COBOL* a la *mainframe*. Un *COBOL SDE* basado en PC como el Workbench de *COBOL* de Micro Focus proporcionó al programador unas herramientas de prueba y de corrección de errores y editores más poderosos a nivel de la estación de trabajo. Un programador podría hacer mucho del trabajo de desarrollo en ese nivel y luego cargaría el código en la computadora central

### ambiente de desarrollo de software (SDE)

Conjunto de lenguaje y herramientas para el desarrollo de aplicaciones.

para la prueba, el afinado del desempeño y el sistema en producción. Con frecuencia, el SDE podría estar en interfaz con una herramienta CASE y un generador de código para aprovechar los modelos de proceso desarrollados durante el análisis de los sistemas.

Finalmente, los SDE suministraron las herramientas para desarrollar sistemas cliente/servidor de presentación distribuida. Por ejemplo, el *Dialog Manager* de Micro Focus suministró a los usuarios del *COBOL Workbench* herramientas para construir interfaces de usuario basadas en *Windows* que pudieran colaborar con los monitores de transacción *CICS* y los programas de *COBOL* del *mainframe*.

**SDE para cliente/servidor de dos capas M** Actualmente el SDE típico para aplicaciones de dos capas de cliente /servidor (también llamado de datos distribuidos) consta de un lenguaje de programación basado en el cliente con conectividad incorporada de SQL para uno o más motores del servidor de bases de datos. Los ejemplos de SDE de dos capas de cliente/servidor incluyen *PowerBuilder* de Sybase y *Delphi* de Borland (versión de cliente/servidor). Por lo común, estos SDE proveen lo siguiente:

- El desarrollo rápido de aplicaciones (RAD, por sus siglas en inglés) para construir rápidamente la interfaz gráfica del usuario que se replicará y se ejecutará en todas las PC del cliente.
- La generación automática del código de la plantilla para la GUI antes citada y los eventos asociados de sistema (tales como un clic del ratón o del teclado, etc.) que usa el GUI. El programador sólo tiene que añadir el código para la lógica del negocio.
- Un lenguaje de programación que se compila para copia y ejecución en las PC clientes.
- La conectividad (en el lenguaje ya citado) para diversos motores relacionales de bases de datos y la interoperabilidad con esos motores. La interoperabilidad se logra incluyendo órdenes de bases de datos de SQL (por ejemplo, crear, leer, actualizar, suprimir y ordenar registros) que serán enviadas al motor de la base de datos para su ejecución en el servidor.
- Un ambiente artificial para probar y depurar el código para el cliente.
- Un ambiente de pruebas del sistema que ayuda al programador a desarrollar, mantener y correr un guión de prueba reusable de datos, acciones y eventos del usuario contra los programas compilados para asegurar que los cambios de código no introduzcan problemas nuevos o imprevistos.
- Un ambiente de generación de reportes para simplificar la creación de nuevos reportes de usuarios finales de una base de datos remota.
- Un sistema de ayuda para las PC clientes.

Actualmente, la mayor parte de estas herramientas vienen incorporadas en el SDE, pero han surgido vendedores independientes de herramienta de software que producen herramientas sustitutas que a menudo suministran todavía mayor funcionalidad y/o productividad que las provistas en el SDE básico. Para aprender más acerca de estas herramientas accesorias, busque en Internet el *Programmers Paradise* (Paraíso de los Programadores), un portal de Web de herramientas de desarrollo de software.

Parte de la lógica del proceso de cualquier aplicación de dos capas de cliente/servidor puede descargarse en el servidor de la base de datos en forma de procedimientos almacenados. En este caso, los procedimientos almacenados se escriben en un superconjunto (una extensión) del lenguaje SQL. Estos procedimientos son luego “llamados” desde el cliente para su ejecución en el servidor. Diferentes expertos parecen amar u odiar los procedimientos almacenados. Desde el punto de vista positivo, puede hacerse que los procedimientos almacenados construyan mejor la integridad de los datos en las tablas de bases de datos. Son reusables y verificables. Desde el punto de vista negativo, dificultan la distinción entre las capas de aplicación y de manipulación de datos de nuestro marco; son lógica de aplicación que se ejecuta en los servidores de bases de datos. Muchos diseñadores prefieren una estrategia de diseño más consistente llamada de *distribución en capas limpias* (*clean layering*). La **distribución en capas limpias** requiere que la presentación, la aplicación y las capas de datos de una aplicación estén físicamente separadas. Se dice que la estratificación limpia permite que los componentes de cada capa se revisen y amplíen sin afectar a otras capas del sistema.

---

**distribución en capas limpias** Estrategia de diseño que requiere la separación física de las capas de presentación, aplicación y datos.

**SDE para cliente/servidor de multicapas** El actual estado del arte en el desarrollo de aplicación de la empresa ocurre en los SDE para las arquitecturas de cliente/servidor de tres capas (y más allá). A diferencia de las aplicaciones de dos capas, las aplicaciones en

$n$  capas deben soportar más de 100 usuarios con tiempo de respuesta de transacción y procesamiento parecidas a los de una *mainframe*, con bases de datos de 100 gigabytes o mayores. Mientras que los SDE de dos capas antes descritos están tratando de expandirse en este mercado, una clase diferente de SDE lo domina hoy. Por lo común, los SDE de esta clase deben proveer todas las capacidades típicamente asociadas con los SDE de dos capas, además de lo siguiente:

- Soporte para plataformas heterogéneas de computación, tanto al cliente como al servidor.
- Generación de código y programación tanto para los clientes como para los servidores.
- Un gran énfasis en la reusabilidad usando marcos de aplicación de software, plantillas, componentes y objetos.
- Herramientas integradas de miniportafolios para análisis y diseño que interactúen con los generadores y editores de código.
- Herramientas que ayudan a los analistas y programadores a separar los componentes de aplicación entre los clientes y los servidores.
- Las herramientas que ayudan a los desarrolladores a implantar y manejar la aplicación terminada para los clientes y los servidores. Generalmente esto incluye las herramientas de administración de seguridad.
- La habilidad para escalar en forma automática la aplicación a plataformas mayores y diferentes, de clientes y servidores.
- La administración de la aplicación y control de nuevas versiones del software.

Los ejemplos de SDE de cliente/servidor de  $n$  capas incluyen a *Dinasty* de Dinasty y *VisualAge* de IBM (una familia de productos). Otra vez, un gran número de vendedores independientes de herramienta de software están construyendo herramientas complementarias y sustitutas para estos SDE.

**SDE para cliente/servidor en Internet e intranet** Están surgiendo herramientas de desarrollo de aplicación rápida para permitir aplicaciones de cliente/servidor en Internet y en intranet. La mayoría de estos lenguajes se construyen alrededor de un eje de cuatro tecnologías estándar:

*HTML* (Hypertext Markup Language), el lenguaje que se usa para construir la mayor parte del contenido y de los hiperenlaces de las páginas de Internet y de intranet.

*XML* (Extensible Markup Language), un lenguaje extensible para transportar datos y sus características a través de la Red.

*CGI* (Computer Gateway Interface), un estándar para publicar componentes, construcciones y enlaces gráficos de la Red Mundial.

*Java*, un lenguaje de programación de uso general para crear *servlets*, *applets* y programas independientes de la plataforma, que pueden ejecutarse desde adentro de la máquina virtual de Java (*Java Virtual Machine*) de un navegador.

Los ejemplos de SDE específicos de *Java* incluyen a *WebSphere* de IBM y *Jbuilder* de Borland. Estos SDE pueden crear aplicaciones de Internet, intranet y aplicaciones que no sean de Internet/intranet. Virtualmente todos los SDE existentes de dos capas y de  $n$ -capas también están evolucionando para soportar *HTML*, *XML*, *CGI* y *Java*.

## Estrategias de arquitecturas de aplicación para el diseño de sistemas

Al margen de cómo se llamen, todos los sistemas de información tienen una arquitectura de aplicación. Las diferentes organizaciones aplican estrategias diferentes para determinar la arquitectura de aplicación que usan. Clasifiquemos brevemente los dos enfoques más comunes.

### > La estrategia de arquitectura de aplicación empresarial

En la estrategia de arquitectura de aplicación empresarial, la organización desarrolla una arquitectura de tecnología de la información que abarque a toda la empresa para ser seguida en todos los proyectos subsiguientes de desarrollo de sistemas de información. Esta arquitectura de tecnología de la información define lo siguiente:



- La red, los datos, la interfaz y las tecnologías de procesamiento y herramientas de desarrollo aprobados (incluyendo hardware y software, y clientes y servidores).
- Una estrategia para integrar los sistemas y las tecnologías heredadas en la arquitectura de aplicación.
- Un proceso permanente para revisar continuamente la arquitectura de aplicación en cuanto a su actualidad y conveniencia.
- Un proceso continuo para investigar tecnologías emergentes y hacer recomendaciones para su inclusión en la arquitectura de aplicación.
- Un proceso para analizar peticiones de desviaciones de la arquitectura de aplicación aprobada.

Una arquitectura de aplicación inicial generalmente es desarrollada como un proyecto separado o como parte de un proyecto estratégico de planeación de sistemas de información. El mantenimiento en curso de la arquitectura de aplicación se asigna por lo regular a un grupo permanente de investigación de tecnología de la información o a un comité de arquitectura de aplicación de la empresa.

Con posterioridad a la aprobación de la arquitectura de aplicación, se espera que cada proyecto de desarrollo de sistemas de información use o escoja tecnologías basadas en esa arquitectura. En la mayoría de casos, esto simplifica mucho la fase de arquitectura de una metodología de desarrollo de sistemas. Usted sólo selecciona entre las tecnologías aprobadas según las reglas o los lineamientos de la arquitectura.

Por supuesto, aun si una tecnología es aprobada en la arquitectura de aplicación, está sujeta a un análisis de factibilidad, tal como se describe en la siguiente sección.

### > La estrategia táctica de arquitectura de aplicación

A falta de una arquitectura de aplicación que abarque a toda la empresa, cada proyecto debe definir su propia arquitectura para el sistema de información que se esté desarrollando. No obstante puede existir algún tipo de grupo de tecnología de la información de investigación y de despliegue.

Aun cuando la arquitectura de aplicación propuesta para cualquier sistema de información nuevo puede ser influida por las tecnologías existentes, los desarrolladores por lo regular tienen algo más de libertad para elegir nuevas tecnologías. Por supuesto, la decisión final debe ser defendida y aprobada como factible. La factibilidad de la tecnología de la información generalmente incluye los siguientes aspectos:

- *Factibilidad técnica.* Ya sea una medida de madurez de una tecnología, una medida de conveniencia de la tecnología para la aplicación que se esté diseñando, o una medida de la capacidad de la tecnología para trabajar con otras tecnologías.
- *Factibilidad operacional.* Es una medida de qué tan cómodos están la administración del negocio y los usuarios con la tecnología y qué tan cómodos están los gerentes de tecnología y el personal de soporte con la tecnología.
- *Factibilidad económica.* Ésta es una medida de si la tecnología es costeable o no y si es eficiente con base en costos, lo que implica que los beneficios sobrepasan a los costos.

Los criterios de factibilidad y las técnicas para medirlos fueron cubiertos en el capítulo 9.

## Modelando la arquitectura de aplicación de un sistema de información

El uso de DFD lógicos para modelar requerimientos de proceso es una práctica bastante aceptada. Sin embargo, la transición de los DFD lógicos orientados al análisis a los DFD físicos orientados al diseño ha sido históricamente algo misteriosa y oscura. Deseamos un diseño general de alto nivel que pueda servir como una arquitectura de aplicación para el sistema y un diseño general para los procesos que constituyen el sistema. Al mismo tiempo, no queremos quedar atrapados en un ejercicio contraproducente de modelado que desacelere nuestro progreso en el diseño de sistemas y el rápido desarrollo de la aplicación. Dicho de manera simple, queremos un plano que nos guíe a través del diseño detallado y la construcción. El plano identificará las unidades de diseño para la especificación detallada o el desarrollo rápido, lo que sea más productivo en nuestro proyecto.

## > Dibujo de diagramas de flujo de datos físicos

La mecánica para dibujar los DFD físicos es idéntica a la de los DFD lógicos. Las reglas de exactitud son también idénticas. Un diseño aceptable resulta en:

- Un sistema que trabaja.
- Un sistema que cumple con los requerimientos del usuario (especificado en los DFD lógicos).
- Un sistema que proporciona un desempeño adecuado (en procesamiento y tiempo de respuesta).
- Un sistema que incluye suficientes controles internos (para eliminar los errores humanos y de la computadora, asegurar la integridad y la seguridad de los datos y satisfacer las restricciones de auditoría).
- Un sistema que es ajustable a los requerimientos que siempre cambian y a las ampliaciones.

Podríamos desarrollar a un DFD físico individual para el sistema entero o un conjunto de DFD físicos para el sistema objetivo. Nuestra metodología sugiere lo siguiente:

- Deberá desarrollarse un diagrama de flujo de datos físicos para la arquitectura de red. Cada proceso en este diagrama es un procesador físico (cliente o servidor) en el sistema. Cada servidor es su propio procesador; sin embargo, es impráctico mostrar a cada cliente. En lugar de eso, cada clase de clientes (por ejemplo, una orden para ingresar un dependiente) está representado como un procesador individual.
- Para cada procesador en el modelo anterior deberá desarrollarse un diagrama de flujo de datos físicos para mostrar los eventos de cada procesos (vea el capítulo 8) que serán asignados a ese procesador. Es posible que usted elegiría duplicar algunos de eventos de procesos en procesadores múltiples. Por ejemplo, las órdenes pueden procesarse en los servidores y clientes de cada región.
- Para todos con excepción de los eventos de procesos más simples, deberán factorizarse en unidades de diseño y modelarse como un diagrama de flujo de datos físicos individual. Una **unidad de diseño** es una colección autocontenida de procesos, almacenamientos de datos y flujos de datos que comparten atributos de diseño similares. Una unidad de diseño sirve como un subconjunto del sistema total cuyas entradas, salidas, archivos y bases de datos y programas pueden ser diseñados, construidos, y la unidad probada como un subsistema individual.

**unidad de diseño** Conjunto independiente de procesos, almacenes de datos y flujos de datos que comparten atributos similares de diseño.

Un ejemplo sería un conjunto de procesos (uno o más) para ser diseñados como un programa individual. Luego la unidad de diseño podría asignarse a un solo programador (o un equipo) quien (el cual) puede trabajar independientemente de otros programadores y equipos sin afectar adversamente el trabajo de los otros programadores. Las unidades implantadas luego serían ensambladas en el sistema de aplicación final. Las unidades de diseño también pueden ser priorizadas para implantar las versiones de un sistema.

## > Los prerequisites

Formulemos la tabla para describir los prerequisites para crear DFD físicos. Se incluyen:

- Un modelo de datos lógicos (el diagrama de entidad relación creado en el capítulo 7).
- Los modelos de procesos lógicos (los diagramas de flujo de datos creados en el capítulo 8).
- Detalles del repositorio para todo lo citado antes.

Dados estos modelos y estos detalles podemos distribuir datos y procesos para crear un diseño general. Su diseño general normalmente estará restringido por uno o más de los siguientes:

- Los estándares de la arquitectura que predeterminaron la elección de los sistemas de administración de base de datos, la topología y la tecnología de la red, interfaz(es) de usuario(s), y/o los métodos de procesamiento.
- Objetivos del proyecto que se definieron al inicio y se refinaron a todo lo largo del análisis de los sistemas.

- La factibilidad de la tecnología y los métodos escogidos y deseados. (Las técnicas del análisis de factibilidad se estudiaron en el capítulo 9.)

Dentro de las limitantes de estas restricciones pueden aplicarse las técnicas resultantes.

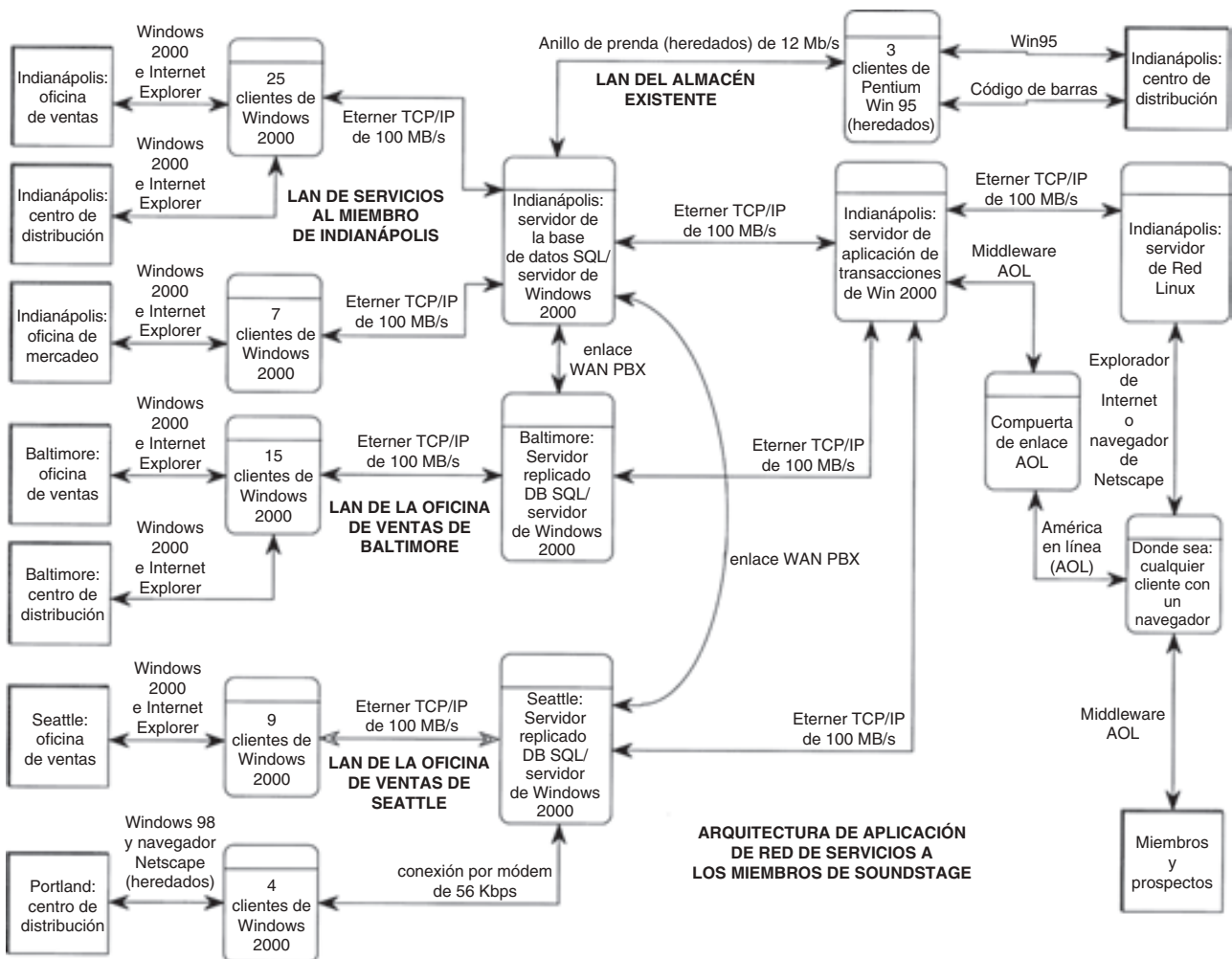
> **La arquitectura de red**

El primer DFD físico que debe dibujarse es el DFD de la arquitectura de red. Una *arquitectura de red DFD* es un diagrama de flujo de datos físicos que asigna procesadores (clientes y servidores) y dispositivos (por ejemplo, las máquinas y los autómatas) a una red y establece: 1) la conectividad entre los clientes y los servidores, y 2) dónde los usuarios interactuarán con los procesadores (generalmente sólo los clientes).

Para identificar los procesadores y sus ubicaciones, el desarrollador utiliza dos recursos:

- Si existe una arquitectura de tecnología de la información de la empresa, esa arquitectura probablemente especifica la visión del cliente/servidor que deberá lograrse.
- Deberá solicitarse el consejo de gerentes de red competentes y/o especialistas para determinar qué está en su lugar, qué es posible y qué impacto puede ejercer el sistema sobre la red de computadoras.

La arquitectura de red DFD (vea la figura 11.11) necesita estar etiquetada para mostrar la información que es diferente de los DFD normales. Estos no muestran flujos de datos



**FIGURA 11.11** DFD de arquitectura de red

por sí mismos. En lugar de eso, muestran carreteras sobre las cuales los flujos de datos pueden viajar en ambas direcciones. También, los DFD de topología de red indican lo siguiente:

- *Los servidores y sus ubicaciones físicas.* Los servidores no siempre están localizados en los sitios indicados en un diagrama de conectividad de la ubicación. El acceso del personal de la red a los servidores generalmente está en discusión. Algunas tareas de administración de red pueden lograrse remotamente y algunas tareas requieren acceso directo.
- *Los clientes y sus ubicaciones físicas.* En este caso, el diagrama de conectividad de las ubicaciones es útil para identificar “grupos” de usuarios similares (por ejemplo, DEPENDIENTES QUE RECIBEN ÓRDENES, REPRESENTANTES DE VENTAS, etc.) quienes recibirán similares servicios de clientes. Un procesador individual debería representar al grupo entero en una ubicación individual. El mismo grupo puede replicarse en ubicaciones múltiples. Por ejemplo, usted esperaría que cada REGIÓN DE VENTAS tenga tipos similares de empleados.
- *Las especificaciones del procesador.* Las descripciones del repositorio de procesadores pueden usarse para definir especificaciones del procesador como RAM, capacidad de disco duro y despliegue.
- *Los protocolos de transporte.* Las conexiones están etiquetadas con los protocolos de transporte (por ejemplo, TCP/IP) y otros parámetros físicos pertinentes.

La topología de la red DFD puede usarse ya sea para diseñar una red de computadoras o documentar el diseño de una red de computadoras existente. En uno u otro caso, la red está siendo modelada a fin de que después le podamos asignar los procesos de sistemas de información, los almacenamientos de datos y los flujos de datos a los servidores en la red.

### > Distribución de datos y asignaciones de tecnología

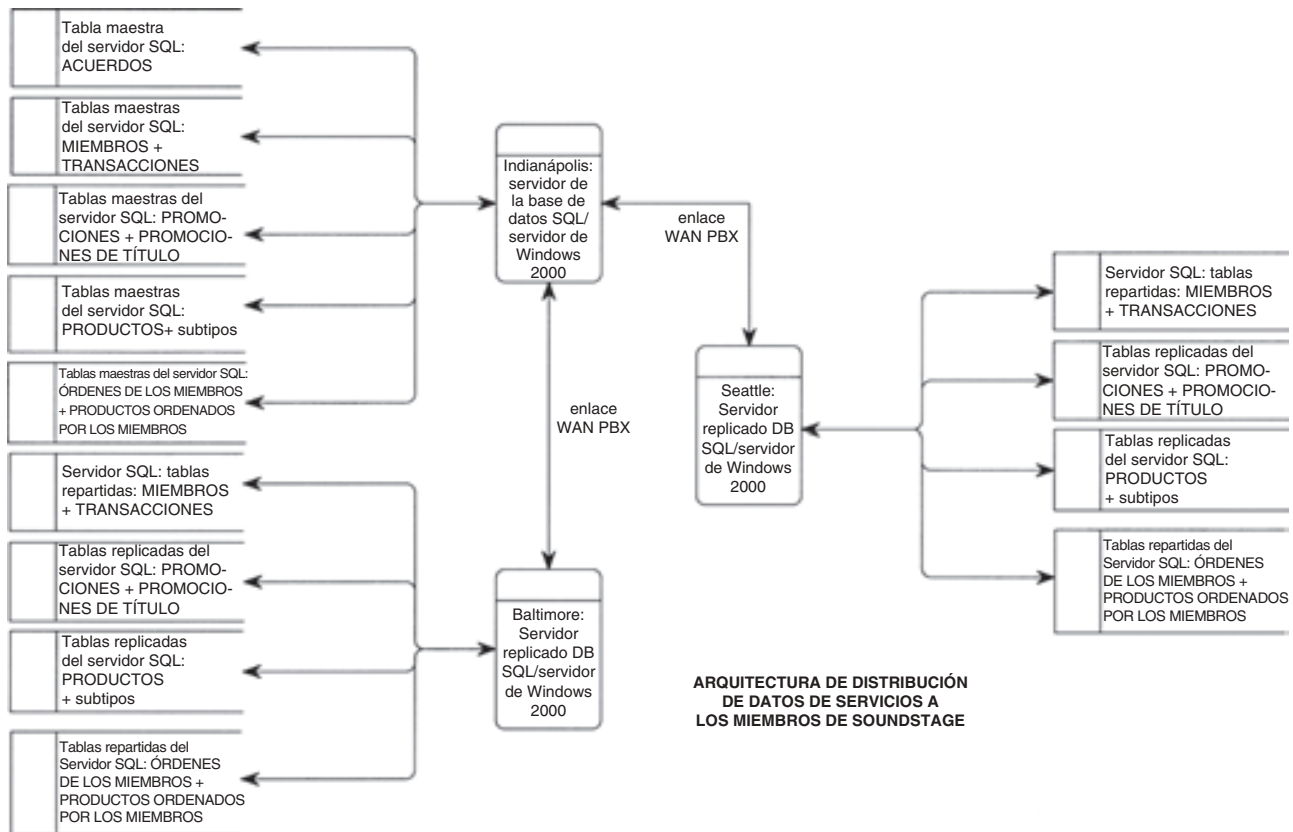
El siguiente paso es distribuir los almacenamientos de datos a los procesadores de la red. Los almacenamientos de datos lógicos requeridos ya conocen del análisis de sistemas: como almacenamientos de datos en los DFD lógicos (capítulo 8) o como entidades en los ERD lógicos (capítulo 7). Sólo necesitamos determinar dónde se guardará físicamente cada uno y cómo se implantarán.

Para distribuir los datos y asignar sus métodos de implantación, los desarrolladores utilizan tres recursos:

- Si están disponibles, las matrices de distribución de datos del análisis de sistemas (capítulos 7 y 8) modelan las necesidades de datos en las ubicaciones de negocios a partir de una perspectiva independiente de la tecnología.
- Si existe una arquitectura de tecnología de información empresarial, esa arquitectura tal vez especifica la visión de la base de datos y las tecnologías que deberán ser buscadas.
- Deberá solicitarse el consejo de administradores de datos y de bases de datos para determinar lo que está en su lugar, lo que es posible y qué impacto puede tener la base de datos en el sistema global.

Las opciones de distribución se describieron anteriormente en el capítulo y se resumen como sigue:

- *Almacene todos los datos en un solo servidor.* En este caso deberá denominarse la base de datos (consistente en tablas múltiples), y esa base de datos denominada y su método de implantación (por ejemplo, Oracle: dbmemberServices), deberá añadirse al DFD físico y deberá estar conectada al procesador apropiado.
- *Almacene tablas específicas en servidores diferentes.* En este caso, y por claridad, deberemos registrar cada tabla como un almacenamiento de datos en el DFD físico y conectar cada uno al servidor correcto.
- *Almacene subconjuntos de tablas específicas en servidores diferentes.* En este caso registramos las tablas tal como antes, excepto que indicamos cuáles tablas son subconjuntos del conjunto total de registros. Por ejemplo, la etiqueta DB2: TABLA DE ÓRDENES (REG SUBSET) señalaría que un subconjunto de todas las órdenes de una región se almacena en una tabla DB2 de bases de datos.
- *Reproduzca (duplicue) las tablas específicas o los subconjuntos en servidores diferentes.* En este caso, los almacenamientos de datos replicados se muestran en el DFD



**FIGURA 11.12** Distribución de datos y asignaciones de tecnología para *SoundStage*

físico. Una copia de cualquier tabla replicada se designa como el MASTER y todas las demás copias se designan como COPIA O REPLICADO.

¿Por qué distribuir el almacenamiento de datos? Hay muchas posibles razones. Primera, algunas instancias de datos son sólo de interés local. Segunda, el desempeño a menudo puede mejorarse situando subconjuntos de datos en ubicaciones múltiples. Finalmente, es necesario localizar algunos datos para asignar custodia de esos datos. La distribución de datos y las asignaciones de tecnología para el estudio de caso SoundStage se muestran en la figura 11.12.

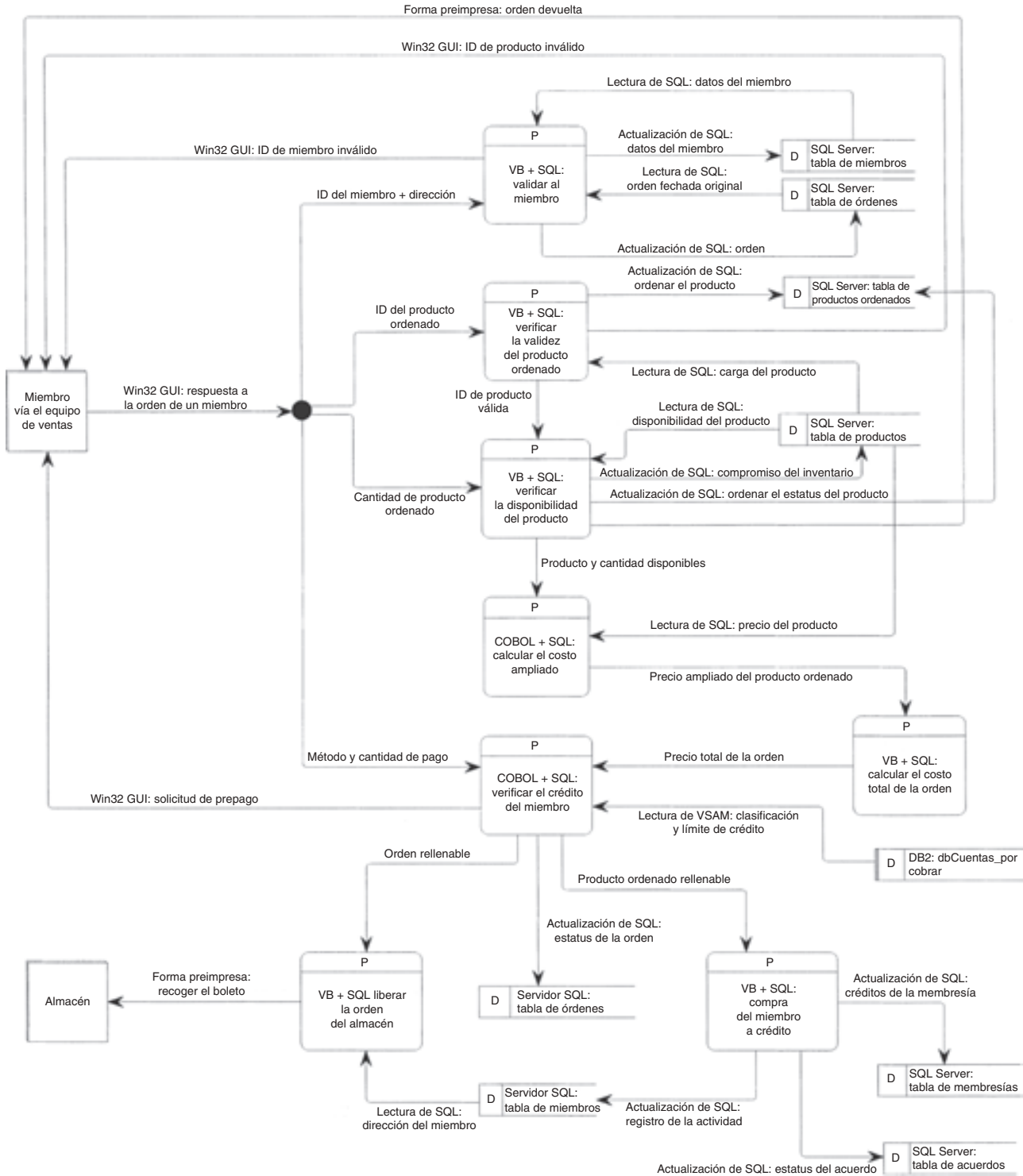
Las decisiones de distribución de datos pueden ser muy complejas; por lo común, las decisiones son guiadas por profesionales de datos y de bases de datos y se enseñan en los cursos y libros de texto de administración de datos. En este libro queremos considerar sólo cómo documentar la partición y las decisiones de duplicación.

### > Distribución de procesos y asignaciones de tecnología

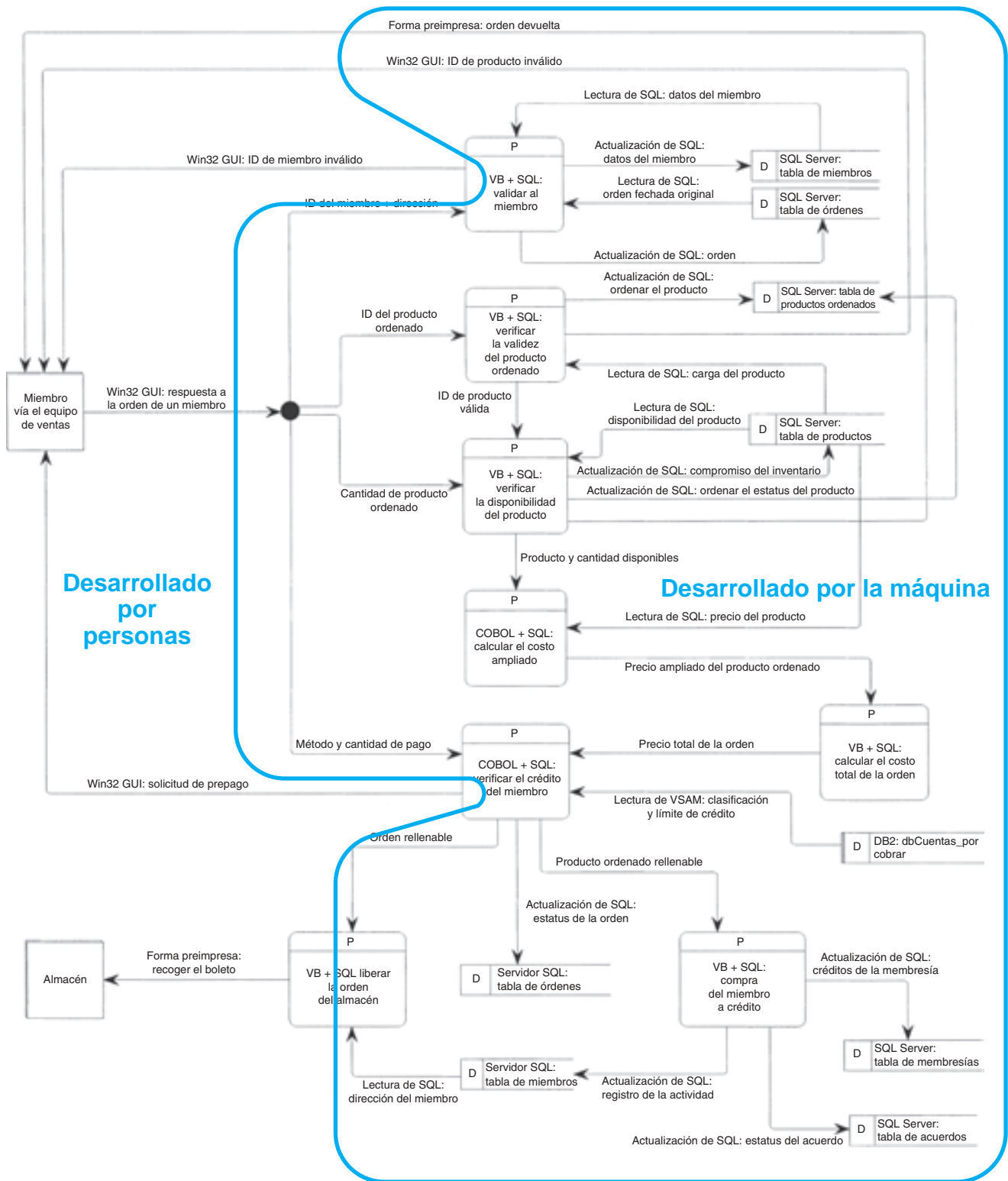
Los procesos de sistemas de información ahora pueden asignarse a los procesadores como sigue:

- Para los sistemas cliente/servidor de dos capas, todos los diagramas de eventos lógicos (capítulo 5) se asignan al cliente.
- Para los sistemas cliente/servidor de tres capas y de red, usted debe examinar de cerca el diagrama original de flujo de datos de cada evento (detallado). Usted necesita determinar cuáles de los procesos originales deberán asignarse al cliente y cuáles deberán asignarse a un servidor de aplicación. En general, la captura y edición de datos se asignan a los clientes mientras otra lógica del negocio se asigna a los servidores. Si usted reparte los diferentes aspectos de un DFD lógico entre los distintos servidores y clientes, deberá dibujar DFD físicos separados para las porciones en cada cliente y el servidor.

Después de la partición, cada DFD físico corresponde a una unidad de diseño para un evento de negocios dado. (Los eventos de negocios, o los casos de uso, se discutieron en el capítulo 6.) Para cada una de estas unidades de diseño, usted debe asignar un método de implantación, el SDE que se usará para implantar ese proceso. Usted también debe asignar los métodos de implantación a los flujos de datos.



**FIGURA 11.13** Un DFD físico para un evento



**FIGURA 11.14** La frontera persona/máquina

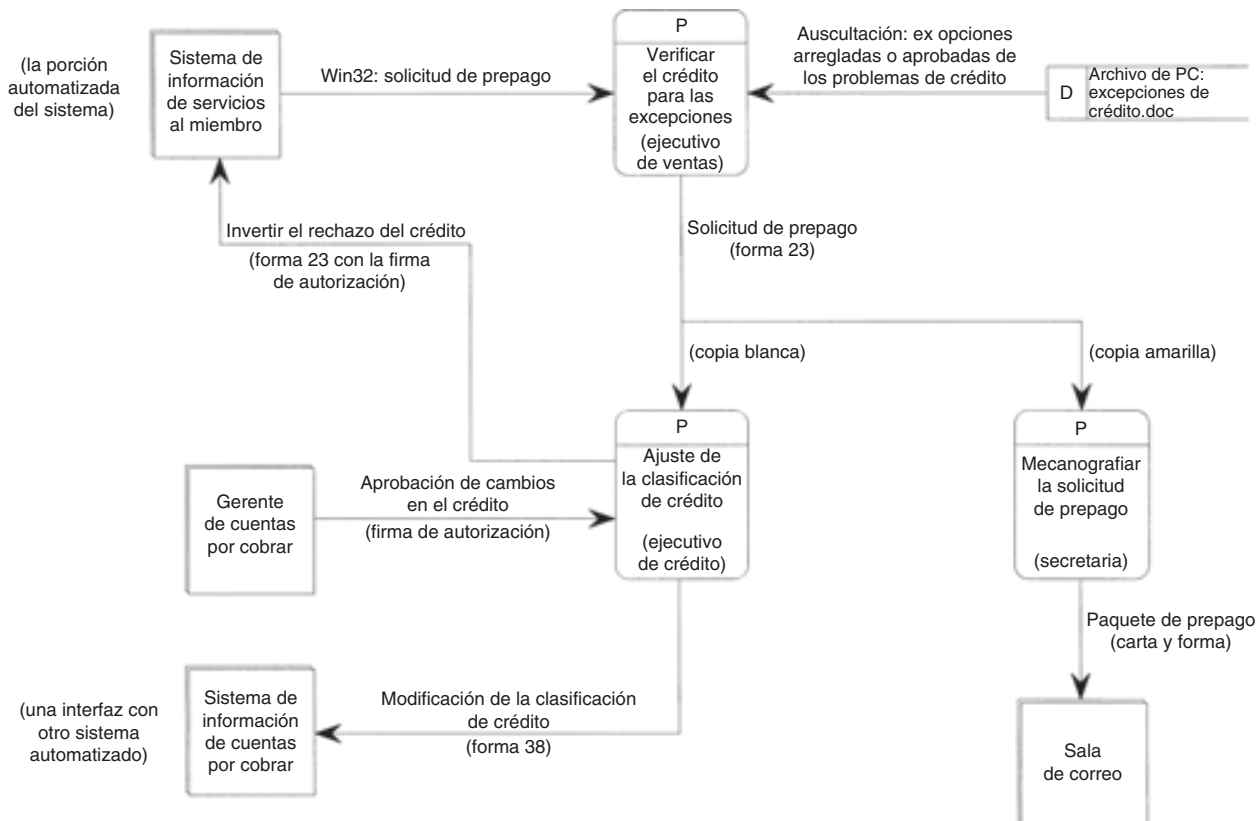
El Sistema de servicios a los miembros de SoundStage será implantado con una arquitectura de cliente/servidor multicapas y de cómputo de redes. En la figura 11.13 se muestra un DFD de muestra para un evento que debe asignarse a un cliente. Observe que se muestran los almacenamientos de datos aun cuando sabemos que han sido repartidos a un servidor de la base de datos. Esto es para el beneficio de los programadores que deben implantar el DFD.

### > Los límites entre persona/máquina

El último paso del diseño de proceso es separar cualquier porción de los DFD físicos que representan procesos manuales no computarizados. Algunas veces a esto se le llama establecer el límite (frontera) entre persona/máquina. Establecer una frontera de persona/máquina no es difícil, pero no es tan simple como usted podría pensar en un inicio. La dificultad surge cuando la frontera persona/máquina atraviesa un proceso lógico; en otras palabras, parte del proceso debe ser manual y parte del proceso debe ser computarizado. Esta situación es común en los DFD lógicos porque se dibujan sin tomar en cuenta las alternativas de implantación.

La figura 11.14 añade la frontera persona/máquina a un DFD físico. Observe que nuestra frontera atraviesa varios procesos, incluyendo el proceso de VERIFICAR EL CRÉDITO DEL MIEMBRO. La solución para este proceso requiere dos pasos:

1. Las porciones del proceso manual se apartan como una unidad de diseño separada (vea la figura 11.15). Todos estos procesos son completamente manuales. Las interfaces de las unidades de diseño manual con los procesos computarizados (en la figura 11.14) se dibujan como agentes externos. Por último, los procesos manuales en la unidad de diseño deben describir claramente a las personas que tendrán que realizarlos.
2. Si es necesario, los procesos en el diagrama original deberán ser renombrados para reflejar sólo la porción computarizada. (En la práctica, los procesos fueron ya denominados de ese modo.)



**FIGURA 11.15** Unidad de diseño manual



En este capítulo usted ha aprendido cómo dibujar el diseño de un nuevo sistema de información para satisfacer los requerimientos identificados y modelados durante el análisis de sistemas. Este diseño general del nuevo sistema guiará el diseño y la construcción detallados de ese sistema.

La mayoría de lectores proseguirá ahora a los capítulos de diseño detallado que se fundamentan en el diseño general del nuevo sistema. Para la mayoría de ustedes, recomendamos que comiencen con el capítulo 12, "Diseño de bases de datos". Muchas de las técnicas de desarrollo de aplicación rápida y de diseño mediante prototipos dependen absolutamente de la existencia de la base de datos de la información planeada del sistema. Así, el capítulo 12 es el mejor lugar para comenzar un estudio del diseño detallado. En seguida, usted puede avanzar a los capítulos que cubren otros aspectos del diseño detallado, incluyendo entradas, salidas y programas.



## Resumen

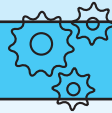
1. Los diagramas de flujo de datos físicos modelan las decisiones de diseño técnicas y humanas que deben implantarse como parte de un sistema de información. Comunican las selecciones técnicas y otras decisiones de diseño a quienes realmente construyen e implantan el sistema.
2. Una arquitectura de tecnología de la información define las tecnologías que deben usar uno, más o todos los sistemas de información. Hay cuatro categorías de arquitecturas de tecnología: la red, los datos, la interfaz y el proceso.
3. Un sistema distribuido es aquel en el que los componentes de un sistema de información se distribuyen a las múltiples ubicaciones en una red de computadoras. Las cinco capas de la arquitectura de sistemas distribuidos son *a)* presentación, *b)* lógica de presentación, *c)* lógica de la aplicación, *d)* manipulación de datos y *e)* datos.
4. Una red de área local es un conjunto de computadoras cliente conectadas a uno o más servidores ya sea a través de conexiones de cable o inalámbricas para distancias relativamente cortas.
5. Un sistema de servidor de archivos es una solución basada en LAN, en la cual una computadora servidor atiende sólo a la capa de datos. Todas las demás capas se implantan sobre el cliente.
6. El modelo de cómputo predominante es actualmente el de cliente/servidor, en el cual la presentación, la lógica de presentación, la lógica de aplicación, la manipulación de datos y las capas de datos se distribuyen entre PCs clientes y uno o más servidores. Los clientes se clasifican por su potencia como delgados o pesados. Los servidores se asignan a funciones tales como bases de datos, transacciones, aplicaciones, mensajería o grupo de trabajo, o de Internet.
7. La presentación distribuida, los datos distribuidos y los datos y la lógica distribuidos son tipos de sistemas cliente/servidor.
8. La computación de redes usa la tecnología de Internet para construir aplicaciones de Internet o de la intranet.
9. El almacenamiento de datos se implanta típicamente usando tecnología relacional de bases de datos que reparte los datos a los diferentes servidores o reproduce los datos en servidores múltiples.
10. Las opciones de la interfaz del usuario incluyen la entrada de datos por lotes, en línea, lotes remotos, sin captura (incluyendo métodos ópticos de caracteres y de código de barras), entrada con pluma, mensajes electrónicos, intercambio electrónico de datos e imágenes.
11. La interfaz del sistema se implanta normalmente usando *middleware*, el software que permite que los procesos se comuniquen entre sí.
12. Los procesos se implantan usando juegos de herramientas altamente integrados llamados ambientes de desarrollo de software.
13. Las arquitecturas de aplicación pueden desarrollarse e imponerse estratégicamente, o pueden evolucionar tácticamente sobre la base de proyecto por proyecto.

## Preguntas de repaso



1. En el análisis y diseño estructurados tradicionales, ¿qué modelos de sistemas se desarrollan y en qué orden?
2. ¿Por qué la metodología completa del análisis y diseño estructurados rara vez se emplea en la actualidad?
3. ¿Cuándo se divide un proceso lógico en procesos físicos múltiples, o si se agregan más procesos físicos, qué es importante que comprueben los diseñadores?
4. ¿Por qué el número de procesos físicos mostrados en un DFD físico generalmente es mayor que el número de procesos lógicos?
5. ¿Qué representa un flujo de datos físicos?
6. ¿Qué tipo de almacenamiento de datos es omitido con frecuencia por los diseñadores al conducir el diseño de sistemas?
7. Aunque los sistemas centralizados sean menos complejos y más fáciles de implantar, los sistemas distribuidos han aventajado bastante a los sistemas centralizados. ¿Cuáles fueron algunas de las razones para esto?
8. ¿Cuál es la diferencia entre la capa de presentación y la capa lógica de presentación?
9. ¿Qué es un sistema de servidor de archivos y qué clase de ambiente de red usa?
10. ¿Cuáles son algunas de las desventajas y limitaciones inherentes de un sistema de servidor de archivos?
11. ¿Cuál es la diferencia entre un cliente delgado y un cliente pesado?
12. ¿Cuál es la arquitectura de red que se usa en el comercio electrónico? Por favor explique cómo se relaciona cada capa.
13. ¿Cuál es la secuencia de las tareas de alto nivel para modelar la arquitectura de aplicación de un sistema de información?

## Problemas y ejercicios



1. Usted está a la mitad de la fase de diseño de sistemas de un proyecto de desarrollo de una intranet corporativa, y el equipo de proyecto celebra una reunión de planeación. Uno de los usuarios del sistema en el proyecto, quien ha dicho muy poco durante la reunión, finalmente habla más fuerte y dice: “Todos ustedes, los técnicos, siguen hablando acerca de la arquitectura de aplicación que vamos a diseñar. No tengo ni la menor idea acerca de lo que ustedes hablan”. Defina y explique al grupo lo que significa la arquitectura de aplicación para los usuarios no técnicos del sistema.
2. ¿Cuál es el propósito del diagrama de flujo de datos físicos? ¿En general, cómo difieren de los diagramas de flujo de datos lógicos? ¿Qué conexiones y formas básicas usan los DFD físicos? ¿Son los DFD físicos una herramienta de diseño heredada, o son todavía una herramienta viable en el mundo orientado a objetos de hoy?
3. Usted está trabajando en un proyecto para diseñar un sistema nuevo de órdenes para un distribuidor de partes de automóvil. Desarrolla el DFD físico y uno de los procesos lógicos es “Verificar el inventario en existencia.” Si este proceso debe ser realizado tanto por las personas como por la computadora, ¿cómo mostraría usted que éste es un proceso físico? (Nota: Use los formatos de diagrama mostrados después de la figura 11.1.) ¿Qué ocurre si el proceso está completamente hecho por computadora pero usando tecnologías diferentes?
4. Explique la replicación de datos y su propósito. ¿En qué tipo de sistema de base de datos encontraría usted la replicación de datos?
5. Complete las siguientes frases:  
Un \_\_\_\_ sistema también se llama un sistema cliente/servidor distribuido de aplicación y de datos o \_\_\_\_ . El \_\_\_\_ reside en un \_\_\_\_\_, el \_\_\_\_ reside en el servidor del cliente, y los datos y \_\_\_\_\_ en el \_\_\_\_\_.  
Un sistema de \_\_\_\_-capas también se llama \_\_\_\_\_, y el \_\_\_\_\_ reside en los clientes, y las capas de datos y las capas de manipulación de datos en el servidor.  
Un sistema cliente/servidor \_\_\_\_\_ coloca las capas de presentación y de la lógica de presentación en \_\_\_\_\_, y el \_\_\_\_\_ en el servidor.
6. ¿Qué tienen en común los sistemas de servidor de archivos y los sistemas cliente/servidor? ¿Qué tienen de diferente? ¿Cuáles son las ventajas más importantes de una solución cliente/servidor?
7. Usted está trabajando en el área de tecnología de la información de una organización creciente que está planeando implantar un nuevo sistema cliente/servidor. Al principio, habrá poco menos de 100 clientes con una cantidad sustancial de actividad de entrada de datos y de análisis de datos a través de la red. Los promotores de negocios deben tener capacidad para capturar datos y “compactarlos” rápidamente. El presupuesto para el proyecto es cuantioso y permite la compra de estaciones de trabajo y computadoras personales poderosas. Los diseñadores del equipo están indecisos entre una arquitectura de dos capas y de tres capas del

- cliente/servidor. Se dirigen a usted buscando un consejo. ¿Qué recomendaría usted? ¿Por qué?
8. La tecnología de Internet ha crecido a una tasa explosiva durante la década pasada. En opinión de muchos, las arquitecturas de cómputo de redes representan un movimiento principal en una dirección radicalmente diferente apartándose de las arquitecturas cliente/servidor. ¿Por qué es esto?
  9. Coteje los términos en la primera columna con las definiciones o los ejemplos en la segunda columna.
 

<ol style="list-style-type: none"> <li>1. Cliente delgado</li> <li>2. El almacenamiento lógico de datos</li> <li>3. El servidor de <i>groupware</i></li> <li>4. El procesador</li> <li>5. El servidor de transacción</li> <li>6. El <i>mainframe</i></li> <li>7. La capa de presentación</li> <li>8. El flujo de datos físicos</li> <li>9. El almacén de datos físicos</li> <li>10. La capa lógica de presentación</li> <li>11. La Red de Área Extendida (WAN)</li> <li>12. Estándar para compartir objetos</li> <li>13. La capa de la lógica de aplicación</li> </ol>	<ol style="list-style-type: none"> <li>A. Registros del tratamiento de pacientes</li> <li>B. Pantalla de entrada de datos</li> <li>C. SQL Insert: cuenta nueva</li> <li>D. Terminal de datos</li> <li>E. Archivo SAS: reporte de la lista de espera</li> <li>F. CORBA</li> <li>G. Aplicación de formato de reportes</li> <li>H. Sistema distribuido</li> <li>I. Sistema centralizado</li> <li>J. Exchange de Microsoft</li> <li>K. Tuxedo</li> <li>L. Cliente</li> <li>M. Aplicación de análisis estadístico</li> </ol>
--	---
  10. El proceso por lotes se ha usado desde la década de 1950 y muchas personas consideran que el proceso por lotes es un método obsoleto para procesar datos. Pero si el procesamiento por lotes es obsoleto, ¿por qué todavía se desarrollan nuevas aplicaciones de procesamiento por lotes?
  11. Coteje los términos en la primera columna con las definiciones o ejemplos de la segunda.
 

<ol style="list-style-type: none"> <li>1. Cliente/servidor de dos capas SDE</li> <li>2. Unidad de diseño</li> <li>3. EDI</li> <li>4. <i>Middleware</i> de aplicación</li> <li>5. SDE de cliente/servidor multicapas</li> <li>6. Negocio virtual</li> <li>7. SDE de la intranet cliente/servidor</li> <li>8. Partición</li> <li>9. <i>Middleware</i> de presentación</li> <li>10. Capas limpias</li> <li>11. Entrada de pluma</li> <li>12. El portal de la intranet</li> <li>13. El monitor de transacción</li> </ol>	<ol style="list-style-type: none"> <li>A. HTTP</li> <li>B. CICS</li> <li>C. Capas físicamente separadas de presentación, datos y aplicación</li> <li>D. "Página inicial" de los empleados</li> <li>E. Determinando la distribución de los componentes de aplicación</li> <li>F. <i>Windows CE</i></li> <li>G. <i>Allegris</i></li> <li>H. <i>PowerBuilder</i></li> <li>I. Recopilación autocontenida de flujos, almacenes y procesos de datos</li> <li>J. Corredores de petición de objetos</li> <li>K. Banca comercial en línea</li> <li>L. <i>XML</i></li> <li>M. Amazon.com</li> </ol>
--	---
  12. Usted está trabajando en un proyecto complejo para implantar un sistema de información a nivel de empresa en su organización. Ha terminado casi completamente de crear el diagrama de flujo de datos físicos del DFD lógico, y se percata de que hay varios procesos manuales entrelazados con los procesos computarizados. ¿Qué causó esto? ¿Cómo deberá mostrar los procesos manuales en el DFD físico, o no necesita mostrarlos en absoluto?
  13. Usted ha recibido un conjunto de DFD físicos de un nuevo sistema para revisar su nivel de aceptación. ¿Qué preguntas debería hacerse al revisarlos?

## Proyectos e investigación



1. Aunque sus amigos le hagan bromas acerca de eso, usted es un coleccionista imperturbable de canciones populares típicas de los años 1950 y 1960. Su colección ahora totaliza muchos miles de grabaciones en formatos diversos. Para ayudar a seguir mejor la pista de las grabaciones, usted ha decidido desarrollar un sistema simple de inventario en Microsoft Access. Usted quiere poder añadirle grabaciones nuevas al sistema, actualizar información de las que ya tiene, buscar en campos múltiples un artista o una grabación particular y generar reportes diversos. Diseñe el sistema usando las técnicas aprendidas hasta la fecha, luego dibuje un diagrama de flujo de datos de contexto y un diagrama de flujo de datos lógicos.
2. Muchas organizaciones han implantado intranets. Contacte o visite a varios organismos locales en el sector público y privado que tengan intranets. Encuentre la unidad o persona que sea responsable de la intranet de la organización y discuta su arquitectura de aplicación, sus características, sus políticas, sus aspectos, etcétera.
  - a) Describa cada una de las organizaciones que usted contactó.
  - b) Describa cada una de sus intranets y cómo son usadas por los empleados.
  - c) ¿Son intranets informativas básicas, o alguna de ellas se está utilizando como un portal donde todo (sus aplicaciones de escritorio y todas las

aplicaciones de los sistemas de información que usan para su trabajo) corren desde el navegador de la intranet?

- d) ¿Quién “posee” la intranet en la organización y quién es responsable de colocar el contenido o mantenerla al día?
  - e) ¿Le hacen considerar sus debates que los empleados y los dueños de la intranet de las organizaciones entienden el potencial de su intranet y lo están usando a su potencial completo? Explique su respuesta.
3. Usted es un asesor que ha sido contratado por una compañía para ayudar a actualizar su plan de arquitectura de tecnología de la información. La compañía fabrica generadores eléctricos y tiene oficinas de ventas e instalaciones de servicio a todo lo largo de América del Norte y América Central. Una de sus tareas es recomendar una estrategia de distribución de datos a la compañía. Aunque usted está familiarizado con los principios de arquitecturas de aplicación y los métodos para documentarlos, ha pasado un tiempo desde que usted estaba en la posición de recomendar un enfoque de distribución de datos. Indague en la Web o en la biblioteca de su escuela las preguntas que debería hacer, y los criterios de uso para hacer una recomendación apropiada referente al enfoque de distribución de datos que la compañía debería tomar.
  4. Las computadoras *mainframe* una vez dominaron la tecnología de la información, pero se han deslizado calladamente a la sombra de otras tecnologías, como los sistemas cliente/servidor. De cuando en cuando parece haber un torrente de información de artículos que reportan la muerte final de la *mainframe*. Pero con la misma frecuencia aparece otro despliegue impetuoso de artículos con información sobre el resurgimiento de la *mainframe*. Indague sobre el tema de las tendencias de cómputo *mainframe* en su biblioteca de la escuela, en Internet, y/o con algunos gerentes experimentados de tecnología de la información.
    - a) Describa sus fuentes de investigación y sus ubicaciones.
    - b) ¿Fue capaz de encontrar información “dura”, como el número de ventas por año de las *mainframe*?
    - c) ¿Encontró usted cualquier diferencia en el uso de las computadoras *mainframe* entre los sectores públicos y privados? Describa.
  - d) Con base en la información que encontró, ¿qué conclusiones obtendría acerca del estado actual del cómputo de *mainframe*?
  - e) ¿Qué piensa que pasará en los próximos 10 años? ¿Cuál será el papel, si hay alguno, que usted piensa que la computación con *mainframe* jugará en los sectores públicos y privados? Justifique su respuesta.
5. Nuevas herramientas de desarrollo para facilitar las aplicaciones cliente/servidor en Internet parecen surgir casi todos los días (una pequeña exageración). Investigue una de estas nuevas herramientas de desarrollo y la tecnología base, tales como *.NET* o *XML*, alrededor de la cual se construye. Luego prepare un análisis para su funcionario en jefe de información (real o hipotético) evaluando esta herramienta y/o tecnología nueva y el potencial para el uso por su organización. *Nota:* Como la audiencia objetivo de este análisis es un ejecutivo, su documento deberá tratar los puntos sobresalientes a un nivel apropiadamente alto, aunque las referencias o los enlaces para la documentación técnica detallada también deberán incluirse.
  6. Visite o estudie a una empresa de gran tamaño o una dependencia gubernamental en su área e investigue acerca de sus arquitecturas de aplicación. Vea si puede obtener una copia de su plan de arquitectura de tecnología de la información o un documento equivalente.
    - a) Describa la organización que estudió.
    - b) Describa su(s) arquitectura(s) de aplicación(es). ¿Cuál es la arquitectura de aplicación predominante?
    - c) ¿Qué tipo de arquitectura(s) basada(s) en Internet emplea? ¿Hasta qué punto juega un papel?
    - d) ¿Usa la organización todavía tecnología de *mainframe* o de minicomputadora? Si es así, describa cómo se usa actualmente.
    - e) ¿Qué cambios en la tecnología de arquitectura de aplicación se visualizan para los siguientes cinco años? ¿Cuál es la estrategia de la organización para enfrentar estos cambios tecnológicos? ¿Piensa usted que su estrategia será efectiva?
    - f) Dibuje un diagrama de contexto de alto nivel de su arquitectura global de sistemas de información.

## Casos breves



1. Considere un sitio de comercio electrónico que usted previamente investigó para Wow Munchies (la tienda de comestibles). Formule alguna hipótesis oportuna y realice cualquier investigación nueva que sea necesaria. Cree un modelo de arquitectura de cómputo en red para Wow Munchies. En un corto ensayo, enuncie sus hipótesis junto con cualquier investigación de fondo que usted haya hecho.
2. Usted encontrará, como analista de sistemas, que la persona que diseña un sistema no es a menudo quien desarrolla el programa para éste. En este capítulo se manifiesta que un diseño aceptable de un flujo de datos físicos resulta en:

- a) Un sistema que trabaja.
- b) Un sistema que cumple con los requerimientos del usuario.
- c) Un sistema que provee el desempeño adecuado.
- d) Un sistema que incluye suficientes controles internos.
- e) Un sistema que se ajusta a los siempre cambiantes requerimientos.

Encuentre un ejemplo de un diseño DFD de sistema que *parecía* aceptable, pero no dio como resultado un sistema que cumpliera con los requerimientos indicados. ¿El problema fue de diseño o de complicaciones técnicas? ¿Entendieron los programadores lo que los analistas pidieron? Comparta con la clase.

3. Usted está en un equipo al cual se le ha ordenado que diseñe un sistema para VideoStore, una com-

pañía de renta de películas. Esta compañía alquila películas sólo en tiendas (no en línea) y tiene aproximadamente 10 tiendas a todo lo largo del estado de Ohio. Intercambie opiniones, paso a paso, acerca de cómo recorrería usted el proceso de ciclo de vida, incluyendo la fase de diseño, para crear este sistema. Sea tan prolijo como usted pueda y use ejemplos específicos.

4. Regrese al material del capítulo 5 sobre PIECES y al material del capítulo 9 sobre la matriz de sistemas candidato. ¿Cuáles son las fortalezas de cada uno? Utilícelos conjuntamente para considerar tres sistemas potenciales para la tienda de alquiler de video que usted investigó en el caso previo. ¿De qué manera el uso de matrices de perspectiva múltiple le da una visión más cabal de las opciones de sistemas?



## Ejercicios de equipo e individuales

1. ¿Cuál es la diferencia entre un gerente y un líder? Encuentre un ejemplo de alguien que usted piensa es (o fue) un líder verdaderamente grande y un ejemplo de alguien que usted piensa es (fue) un gerente excelente. ¿Cuáles son las características de esa persona? ¿Piensa usted que alguien que es un gran líder también sería un gran gerente? Intercambie opiniones en la clase.
2. ¿Piensa que sea más fácil, más difícil, o aproximadamente tenga el mismo nivel de dificultad, sabotear (“hackear”) un sistema de servidor cliente

de dos capas que uno de tres capas? Averigüe las vulnerabilidades específicas de seguridad en cada arquitectura y comparta con la clase. (*Nota: No estamos tratando de crear saboteadores aquí. Pero es difícil crear un sistema razonable y escoger arquitecturas, lenguajes, estándares de código, software, etc., apropiadas, sin conocer las amenazas a la seguridad y debilidades inherentes a las elecciones.*)

3. Debate de mesa redonda: ¿Usted piensa que haya tal cosa como el sabotaje *ético*? Si lo hubiera, ¿cómo sería?



## Lecturas recomendadas

Berstein, Phillip, y Eric Newcomer. *Principles of Transaction Processing: For the Systems Professional*. San Francisco: Morgan Kaufman Publishers, 1997. Este libro estudia prácticamente todos los modelos de procesamiento de transacciones, monitores de transacciones y servidores de transacciones que se implantan actualmente.

Gane, Chris, y Trish Sarson. *Structured Systems Analysis: Tools and Techniques*. Englewood Cliffs, NJ: Prentice Hall, 1979. Este clásico sobre la modelación de procesos se convirtió en la base de los diagramas de flujo de datos físicos.

Goldman, James. *Applied Data Communications: A Business-Oriented Approach*, 2a. ed., Nueva York: John Wiley & Sons, 1998. Nuestro colega de Purdue ha escrito un excelente libro de texto para quienes buscan aprender acerca de comunicaciones y redes de datos desde la perspectiva de los negocios.

Goldman, James, Phillip Rawles, y Julie Mariga. *Client/Server Information Systems: A Business-Oriented Approach*. Nueva York: John Wiley & Sons, 1999. Nuestros colegas han escrito un libro de texto sobresaliente que introduce a los estudiantes a la arquitectura de tecnología de la información para los sistemas de información.

Kara, Dan. “Why Partition? Multitiered Application Architecture.” *In Application Development Trends*. Natick, MA: Software Productivity Group, mayo de 1997, pp. 38-46. Este artículo es-

timular nuestro interés e investigación sobre la necesidad de desarrollar y enseñar las técnicas de partición como parte de este libro.

Kara, Daniel A., *et al.* “Enterprise Application Development: Seminar Notes.” Chicago: Software Productivity Group, 12 de noviembre de 1996. Este seminario y los apuntes del SPG han fortalecido nuestra comprensión de las técnicas y tecnologías de desarrollo de aplicación de software de dos capas y de  $n$  capas.

Orfali, Robert, Dan Harkey, y Jeri Edwards. *Client/Server Survival Guide*, 3a. ed., Nueva York: John Wiley & Sons, 1999. Este manual de referencia profesional nos ha servido para tres ediciones de la evolución de la tecnología y terminología de clientes/servidores.

Renaud, Paul. *Introduction to Client/Server Systems*, 2a. ed. Nueva York: John Wiley & Sons, 1996. Éste es otro libro de referencia sobre la arquitectura básica de cómputo distribuido contemporánea.

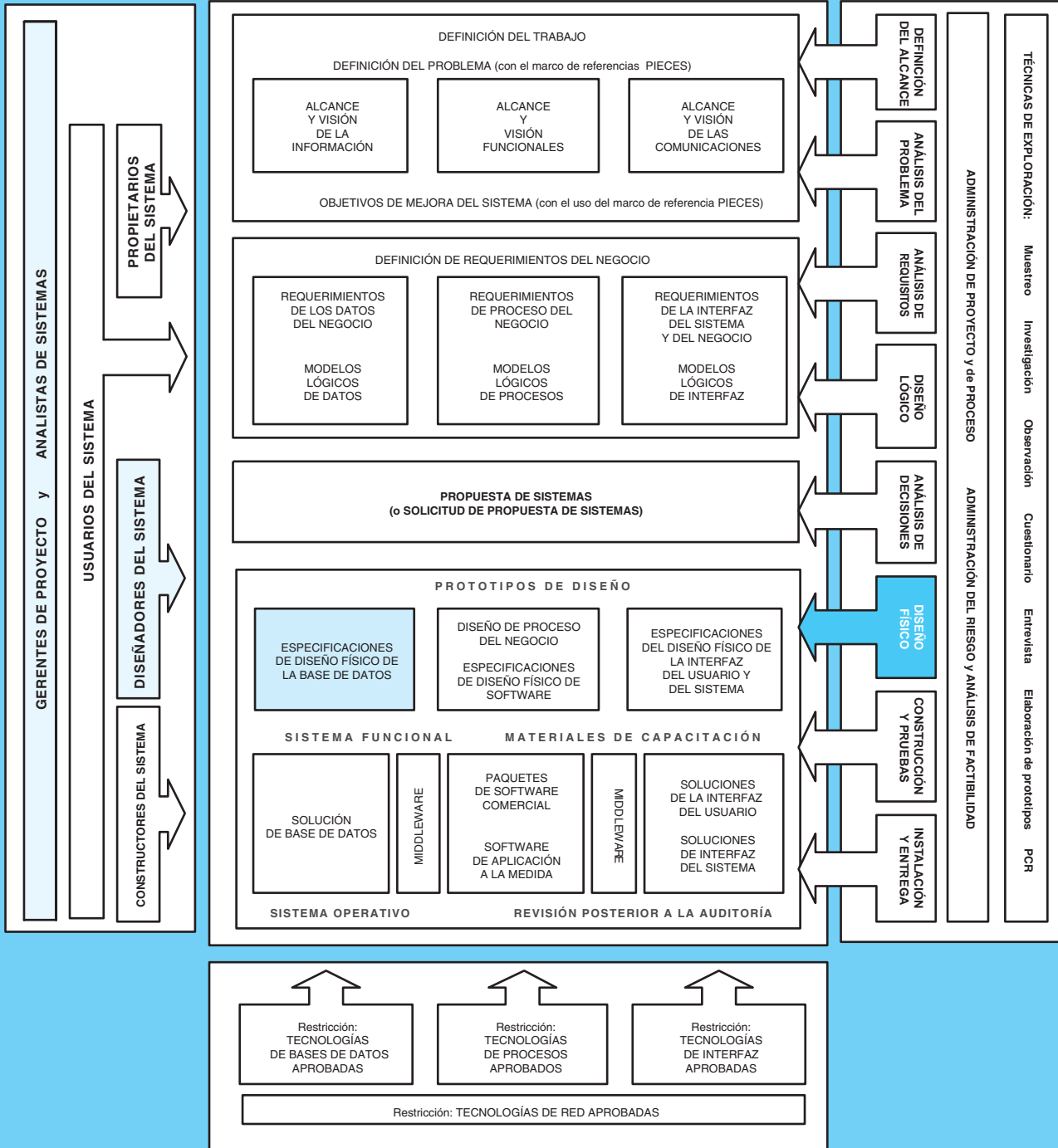
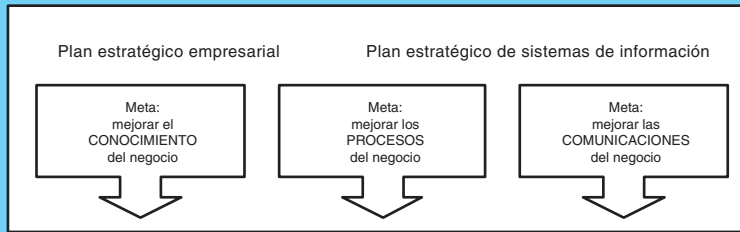
Smith, Patrick, y Steve Guengerich. *Client/Server Computing*, 2a. ed. Indianapolis, IN: SAMS Publishing, 1994. Este libro profesional se ha usado para enseñar lo básico de la tecnología y arquitectura de clientes/servidores a nuestros estudiantes en Purdue. Dada la rápida evolución de esta tecnología, ahora podría existir una tercera edición. Revise los estudios de caso de tecnología en los apéndices.

Theby, Stephen E. "Derived Design: Bridging Analysis and Design." McDonnell Douglas Professional Services: Improved System Technologies, 1987. Las técnicas descritas en este artículo son la base de una fase de STRADIS (Structured Analysis, Design, and Implantation of Information Systems; Análisis, Diseño y Programación Estructuradas de los Sistemas de Información),

una metodología de desarrollo de sistemas. La técnica se alteró y se simplificó para hacerla adecuada al nivel de este libro de texto. Como autores, quedamos muy impresionados con la técnica de diseño completa derivada tal como se expone en la metodología STRADIS.









# 12

## Diseño de bases de datos

### Panorámica y objetivos del capítulo

El almacenamiento de datos es un componente crítico de la mayoría de sistemas de información. Este capítulo enseña el diseño y la construcción de las bases de datos físicas. Sabrá que usted ha dominado las herramientas y las técnicas del diseño de bases de datos cuando pueda:

- Comparar y contrastar los archivos convencionales con las modernas bases de datos relacionales.
- Definir y dar ejemplos de campos, registros, archivos y bases de datos.
- Describir una arquitectura de datos moderna que incluya archivos, bases de datos operacionales, almacenes de datos, bases de datos personales y bases de datos para grupos de trabajo.
- Comparar los puestos del analista de sistemas, del administrador de datos y del administrador de bases de datos en relación con las bases de datos.
- Describir la arquitectura de un sistema de administración de bases de datos.
- Describir cómo una base de datos relacional implanta entidades, atributos y relaciones a partir de un modelo de datos lógicos.
- Transformar un modelo de datos lógicos en un esquema físico de base de datos relacional.
- Generar código SQL para crear estructuras de bases de datos en un esquema.

## Introducción

En la fase de análisis de decisiones del proyecto de sistemas SoundStage Member Services se decidió implantar los datos para el sistema del *Servidor SQL*. Ahora en la fase física del diseño, Bob Martínez ha estado trabajando en el diseño físico de la base de datos.

Para maximizar la producción, la base de datos completa se replicará en cada centro de distribución. Cada ejemplar de la base de datos se almacenará en un servidor Dell Power-Edge con procesadores quad Xeon y discos duros RAID de nivel 5. Afortunadamente, el *Servidor SQL* tiene capacidad incorporada para sincronizar los datos replicados.

Bob refinó el diagrama normalizado de entidad-relación que creó durante la fase de diseño lógico. Usando la herramienta CASE, el *System Architect*, revisó los nombres de tablas y campos de acuerdo con las convenciones aceptadas de denominación de SoundStage. Creó índices en todos los campos clave (comúnmente llamados campos llave) así como también en todos los campos no clave con requerimientos de un conjunto. Creó restricciones para las claves primarias y claves foráneas en las tablas. Y también creó otras restricciones para implantar reglas comerciales que requieren valores por omisión para algunos campos, requieren valores no nulos en algunos otros, o se necesita limitar las entradas del campo a un cierto dominio o rango de valores.

El *System Architect* generó automáticamente el código SQL que se usará para construir la base de datos real. Mientras tanto, Bob usó ese código SQL para crear un prototipo de escritorio de la base de datos en el Motor de Datos Microsoft (MSDE, por sus siglas en inglés), un motor de base de datos de desarrollo compatible con el *servidor SQL*. El prototipo dará a los desarrolladores algo contra lo cual puedan probar su SQL y sus programas.

## Archivos convencionales contra bases de datos

**archivo** Conjunto de registros similares.

**base de datos** Conjunto de archivos interrelacionados.

Todos los sistemas de información crean, leen, actualizan y suprimen (lo que algunas veces se abrevia como CRUD) datos. Esta información se guarda en archivos y bases de datos. Un **archivo** es una colección de registros similares. Los ejemplos incluyen un ARCHIVO DEL CLIENTE, UN ARCHIVO DE ÓRDENES y un ARCHIVO DE PRODUCTOS. Una **base de datos** es una colección de archivos interrelacionados. La palabra clave es *interrelacionados*. Una base de datos *no* es meramente una colección de archivos. Los registros en cada archivo deben permitir relaciones (piense en ellas como “punteros”) para los registros en otros archivos. Por ejemplo, una base de datos VENTAS podría contener registros de órdenes que están enlazados con sus registros correspondientes de CLIENTE y PRODUCTO.

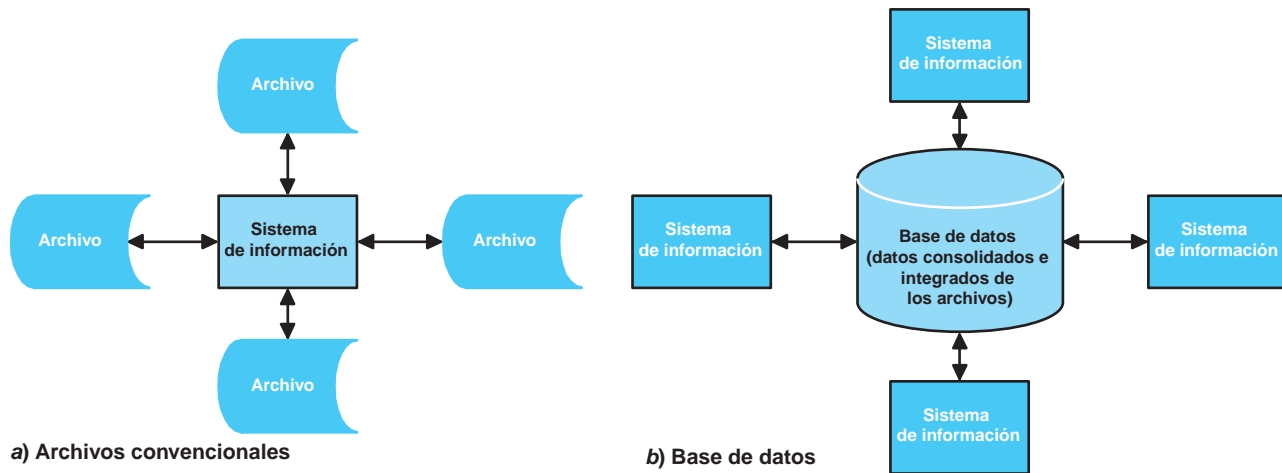
Comparemos las alternativas de los archivos y de bases de datos. La figura 12.1 ilustra la diferencia fundamental entre los ambientes de archivos y de bases de datos. En el ambiente de archivos, el almacenamiento de datos se construye alrededor de las aplicaciones que usarán los archivos. En el ambiente de bases de datos, las aplicaciones se construirán alrededor de bases de datos integradas. En consecuencia, la base de datos no depende necesariamente de las aplicaciones que la usarán. En otras palabras, dada una base de datos, pueden construirse aplicaciones nuevas para compartir esa base de datos. Cada ambiente tiene sus ventajas y sus desventajas.

Como se muestra en la página principal del capítulo, éste trata del diseño y la construcción (inicial) de la base de datos para un sistema de información. El prerrequisito es un modelo de datos (requerimientos) del capítulo 7. Los productos son un esquema de base de datos (diseño) y un programa de la base de datos (definición).

### > Pros y contras de los archivos convencionales

En la mayoría de las organizaciones, muchos sistemas de información y aplicaciones existentes se construyen alrededor de archivos convencionales. Tal vez usted ya esté familiarizado con diversas organizaciones de archivo convencionales (por ejemplo, indexada, hashed, relativa y secuencial) y sus métodos de acceso (por ejemplo, secuencial y directo) de un curso de programación. Probablemente estos archivos convencionales estarán en funcionamiento durante algún tiempo.

Los archivos convencionales son relativamente fáciles de diseñar e implantar porque normalmente se diseñan para usarse con una aplicación individual o un sistema de información, como CUENTAS POR COBRAR O NÓMINA. Si usted entiende los requerimientos de salida del usuario final para ese sistema, puede determinar fácilmente los datos que tendrán que



**FIGURA 12.1** Archivos convencionales contra bases de datos

capturarse y almacenarse para producir esas salidas, y puede definir la mejor organización de archivo para esos requerimientos.

Otra ventaja histórica de los archivos convencionales ha sido la velocidad de procesamiento. Pueden ser optimizados para el acceso de la aplicación. Al mismo tiempo, es raro que puedan ser optimizados para el uso compartido por sistemas o aplicaciones diferentes. De cualquier manera, los archivos han superado a sus contrapartes, las bases de datos; sin embargo, esta limitación de tecnología de la base de datos está desapareciendo en forma rápida gracias a computadoras más económicas y más poderosas, así como tecnología más eficiente de bases de datos.

Los archivos convencionales también tienen numerosas desventajas. La duplicación de elementos de datos en archivos múltiples normalmente se cita como la principal desventaja de los sistemas basados en archivos. Los archivos tienen tendencia a construirse alrededor de aplicaciones individuales sin considerar otras (futuras) aplicaciones. Con el paso del tiempo, debido a que muchas aplicaciones tienen necesidades comunes de datos, los elementos comunes de datos se almacenan en forma redundante en muchos archivos diferentes. Estos datos duplicados conducen a entradas duplicadas, a un mantenimiento duplicado, a un almacenamiento duplicado y posiblemente a problemas de integridad de datos (diferentes archivos que muestran valores diferentes para el mismo elemento de dato).

¿Y qué pasa si es necesario que cambie el formato de los datos? Considere el problema enfrentado por muchas firmas sobre si todos los sistemas deben soportar un código postal de nueve dígitos o años de cuatro dígitos. Ya que estos campos pueden guardarse en muchos archivos (con nombres diferentes), cada archivo tendría que ser estudiado e identificado. En consecuencia, todos los programas que usan este código postal y campos de fecha tendrían que cambiarse.

Una desventaja significativa de los archivos es su inflexibilidad y la dificultad para escaarlos. Comúnmente los archivos se diseñan para soportar los requerimientos y programas *actuales* de una aplicación individual. Las necesidades futuras (tales como nuevos reportes o consultas) con frecuencia requieren que estos archivos sean reestructurados porque la estructura original del archivo no puede soportar efectiva o eficientemente los requerimientos nuevos. Pero si elegimos reestructurar esos archivos, todos los programas que usan esos archivos también tendrían que ser reescritos (reprogramados). En otras palabras, los programas actuales se han vuelto dependientes de los archivos actuales y viceversa. Esto hace que la reorganización no sea práctica; por consiguiente, elegimos crear archivos nuevos, redundantes (los mismos datos, estructurados en forma diferente) para satisfacer los requerimientos nuevos. Pero eso agrava dicho problema de redundancia. Así, ¡los problemas de inflexibilidad y de redundancia tienden a complicarse uno al otro!

A medida que las aplicaciones y los sistemas basados en archivos heredados se convierten en candidatos de la reingeniería, la tendencia apunta en favor de reemplazar aplicaciones y sistemas basados en archivos por sistemas y aplicaciones de bases de datos. Por eso, hemos elegido centrar este capítulo en el diseño de bases de datos.

## > Pros y contras de las bases de datos

Ya hemos indicado la ventaja principal de las bases de datos: la capacidad para compartir los mismos datos a través de sistemas y aplicaciones múltiples. Una concepción errónea y común acerca del enfoque de las bases de datos es que usted puede construir una superbase de datos individual que contenga todos los elementos de datos de interés para una organización. Esta noción, no importa cuán deseable sea, no es práctica en la actualidad.<sup>1</sup> Lo cierto de tal solución es que nos tomaría toda la vida construir una base de datos tan complicada. De una manera realista, la mayoría de las organizaciones construyen varias bases de datos, cada una comparte datos con varios sistemas de información. Esto ocasiona que exista *alguna* redundancia entre las bases de datos. Sin embargo, esta redundancia se reduce lo suficiente y, finalmente, se controla.

La tecnología de las bases de datos le ofrece la ventaja de almacenar datos en formatos flexibles. Esto es posible porque las bases de datos están definidas en forma separada de los sistemas de información y programas de aplicación que los usarán. En teoría, esto nos permite usar los datos en formas no especificadas desde el principio por los usuarios finales. Debe tenerse cuidado de lograr verdaderamente esta *independencia de datos*. Si la base de datos está bien diseñada, puede tenerse fácil acceso a combinaciones diferentes de los mismos datos para satisfacer necesidades futuras de reportes y consultas. El alcance de la base de datos puede aun ampliarse sin cambiar los programas existentes que los usan. En otras palabras, pueden añadirse nuevos tipos de campos y registros a la base de datos sin afectar los programas actuales.

La tecnología de las bases de datos proporciona una capacidad superior de ser *escalable*, lo que implica que las bases de datos y los sistemas que los usan pueden ampliarse o expandirse para satisfacer las necesidades cambiantes de una organización. La tecnología de bases de datos provee una mejor tecnología para las arquitecturas cliente/servidor y la computación de redes. Por lo común, estas arquitecturas requieren que la base de datos corra en su propio servidor. El cliente/servidor y el cómputo de redes se estudiaron en el capítulo 11.

Por otra parte, la tecnología de bases de datos es más compleja que la tecnología de archivos. Se requiere un software especial llamado *sistema de administración de base de datos* (DBMS, por sus siglas en inglés). Aun cuando un sistema de administración de base de datos es todavía algo más lento que la tecnología de archivos, las limitaciones de desempeño están desapareciendo rápidamente. Considerando los beneficios a largo plazo antes descritos, la mayor parte del nuevo desarrollo de sistemas de información está usando tecnología de bases de datos.

Sin embargo, las ventajas de la independencia de datos, la enorme reducción de la redundancia de datos y el incremento de la flexibilidad tienen un costo. La tecnología de bases de datos requiere una inversión significativa. El costo de desarrollar bases de datos es más alto porque los analistas y los programadores deben aprender a usar el sistema de administración de base de datos. Finalmente, para lograr los beneficios de la tecnología de bases de datos, los analistas, los administradores y los expertos de bases de datos deben apegarse a principios rigurosos de diseño.

Otro problema potencial con el enfoque de bases de datos es la creciente vulnerabilidad inherente al uso de los datos compartidos. Usted coloca todos sus huevos en una canasta. Por consiguiente, el respaldo, la recuperación, la seguridad y la privacidad se convierten en asuntos importantes en el mundo de las bases de datos.

A pesar de los problemas analizados, el uso de las bases de datos crece a pasos agigantados. La tecnología continuará mejorando y las limitaciones de desempeño casi desaparecerán. Los métodos de diseño y las herramientas también mejorarán. Por estas razones, este capítulo enfocará la atención en el diseño de las bases de datos como una habilidad importante que deben tener los analistas de sistemas.

## Los conceptos de bases de datos para el analista de sistemas

Deberíamos comenzar con un descargo de responsabilidad. Muchos de los conceptos y los asuntos que son importantes en el diseño de bases de datos también se enseñan en los cursos de bases de datos y de administración de datos. La mayoría de los planes de

<sup>1</sup> Las aplicaciones de Planeación de Recursos Empresariales (ERP, por sus siglas en inglés) suministran una base de datos común hecha a la medida que realmente soporta casi todos los datos operacionales y administrativos que se requieren en muchas organizaciones. Por ejemplo, *SAP R/3* suministra varios miles de tablas.

estudio de sistemas de información incluyen al menos un curso así. No es nuestro intento en este capítulo reemplazar ese curso. Los estudiantes de sistemas de información deberían buscar activamente cursos que *enfocuen la atención* en las técnicas de administración de datos y de bases de datos; esos cursos cubrirán bastantes más técnicas y tecnologías pertinentes que lo que podemos cubrir en este capítulo.

Dicho lo anterior, primero introduciremos (o, para algunos de ustedes, repasaremos) los conceptos y los asuntos de bases de datos que son pertinentes a las responsabilidades del analista de sistemas en el diseño de sistemas de información. Aunque el capítulo se enfoca en el diseño de bases de datos, los lectores experimentados inmediatamente observarán que muchos de los conceptos trascienden la elección entre los archivos y las bases de datos.

## > Los campos

Los campos son comunes para archivos y bases de datos. Un **campo** es la implantación física de un atributo de datos (introducido en el capítulo 7). Los campos son la unidad más pequeña de dato *significativo* para guardarse en un archivo o base de datos. Hay cuatro tipos de campos que pueden ser almacenados: *las claves primarias, las claves secundarias, las claves foráneas o externas y los campos descriptivos*.

Una **clave primaria** es un campo cuyos valores identifican uno y sólo un registro en una entidad de datos (este concepto se introdujo previamente en el capítulo 7). Por ejemplo, el NÚMERO DE CLIENTE identifica únicamente un registro individual del CLIENTE en una base de datos, y el NÚMERO DE ORDEN identifica únicamente un registro individual de ORDEN en una base de datos. También recuerde del capítulo 7 que una clave primaria podría crearse combinando dos o más campos (llamada *clave concatenada*).

Una **clave secundaria** es un identificador *alternativo* para una base de datos. El valor de una clave secundaria puede identificar ya sea un registro individual (al igual que con una clave primaria) o un subconjunto de todos los registros (tal como todas las ÓRDENES que tienen el ESTATUS de ORDEN de devueltas). Una fila individual en una base de datos puede tener sólo una clave primaria, pero puede tener varias claves secundarias. Para facilitar la búsqueda y la clasificación, con frecuencia se crea un *índice* para las claves.

Las **claves foráneas** (también introducidas en el capítulo 7) son punteros para los registros de un archivo diferente en una base de datos. Las claves foráneas permiten que la base de datos conecte los registros de un tipo con los de otro tipo. Por ejemplo, un REGISTRO de ORDEN contiene la clave foránea NÚMERO DE CLIENTE para “identificar” o “señalar” el registro del CLIENTE que se asocia con la ORDEN. Observe que una clave foránea en un archivo requiere la existencia de la clave primaria correspondiente en otra tabla; de otra manera, ¡no señala nada! Así, el NÚMERO DE CLIENTE en un archivo de ÓRDENES requiere la existencia de un NÚMERO DE CLIENTE en el archivo de CLIENTES con objeto de enlazar esos archivos.

Un **campo descriptivo** es cualquier otro (*sin clave*) campo que almacena datos de negocios. Por ejemplo, dado un archivo de EMPLEADOS, algunos campos descriptivos incluyen NOMBRE DEL EMPLEADO, FECHA DE CONTRATACIÓN, SALARIO Y LOS SUELDOS ANUALES HASTA LA FECHA.

Los requerimientos comerciales para ambas claves y los descriptores se definieron cuando usted realizó la modelación de datos en el análisis de sistemas (capítulo 7).

## > Los registros

Los campos son organizados en registros. Los registros son comunes para los archivos y las bases de datos. Un **registro** es una colección de campos organizados en un formato predefinido. Por ejemplo, un REGISTRO DEL CLIENTE puede estar descrito por los siguientes campos (note la notación común):

CLIENTE (NÚMERO, APELLIDO, NOMBRE DE PILA, INICIAL MEDIA, APARTADO POSTAL, DIRECCIÓN DE LA CALLE, CIUDAD, ESTADO, PAÍS, CÓDIGO POSTAL, FECHA DE CREACIÓN, FECHA DE LA ÚLTIMA ORDEN, CLASIFICACIÓN DE CRÉDITO, LÍMITE DE CRÉDITO, SALDO, SALDO VENCIDO...)

Durante el diseño de sistemas, los registros estarán clasificados ya sea como registros de longitud fija o de longitud variable. La mayoría de las tecnologías de bases de datos imponen una *estructura de registro de longitud fija*, lo que implica que cada instancia de registro tiene los mismos campos, el mismo número de campos y el mismo tamaño lógico. Sin embargo, algunos sistemas de bases de datos comprimirán valores y campos sin uso para conservar espacio de almacenamiento en disco. El diseñador de la base de datos debe entender y especificar esta compresión en el diseño de bases de datos.

---

**campo** La unidad más pequeña de datos significativos que se almacena en un archivo o en una base de datos.

---

**clave primaria** Campo o grupo de campos que identifican un registro de una manera única.

---

**clave secundaria** Campo que identifica un registro individual o un subconjunto de registros relacionados entre sí.

---

**clave foránea** campo que apunta a los registros en una tabla o base de datos distinta.

---

**campo descriptivo** Campo sin clave.

---

**registro** colección de campos dispuestos en un formato predeterminado.

En sus anteriores cursos de programación (especialmente *el COBOL*), usted pudo haber encontrado *estructuras de registro de longitud variable* que permiten que registros diferentes en el mismo archivo tengan longitudes diferentes. Por ejemplo, un registro de orden de longitud variable podría contener ciertos campos comunes que ocurren una vez para cada orden (por ejemplo, NÚMERO DE ORDEN, FECHA DE ORDEN y NÚMERO DE CLIENTE) y otros campos que se repiten algunas veces basándose en el número de productos vendidos en la orden (por ejemplo, el NÚMERO DEL PRODUCTO y la CANTIDAD ORDENADA). Por lo común, las tecnologías de la base de datos prohíben (o al menos desalientan) los registros de longitud variable. Esto no es un problema, como lo mostraremos más tarde en el capítulo.

Cuando un programa de la computadora lee un registro de una base de datos, realmente recupera un grupo o *bloque* (o *página*) de registros a la vez. Este enfoque minimiza el número de accesos reales del disco. Un **factor de bloques** es el número de *registros lógicos* incluidos en una sola operación de lectura o de escritura (desde la perspectiva de la computadora). Algunas veces un bloque se llama un *registro físico*. Hoy, el factor de bloques se determina y optimiza por la tecnología escogida de bases de datos, pero un administrador capacitado de bases de datos puede tener permiso de afinar ese factor de bloques para el desempeño. Consideraciones de afinación de las bases de datos se difieren más convenientemente a un curso o libro de texto de bases de datos.

## > Archivos y tablas

Los registros similares son organizados en grupos designados *archivos*. En los sistemas de bases de datos, un archivo con frecuencia se llama *tabla*. Un **archivo** es un conjunto de todas las ocurrencias de una estructura de registro dada. Una **tabla** es la base de datos *relacional* equivalente a un archivo. La tecnología relacional de bases de datos se introducirá en poco tiempo. Algunos tipos de archivos y tablas convencionales son:

- Los **archivos maestros** o tablas maestras contienen registros que son relativamente permanentes. Así, una vez que un registro ha sido añadido a un archivo maestro, se queda en el sistema de manera indefinida. Los valores de los campos en el registro cambiarán durante su vida útil, pero los registros individuales se retienen de manera indefinida. Los ejemplos de archivos y tablas maestros incluyen CLIENTES, PRODUCTOS y PROVEEDORES.
- Los **archivos de transacciones** o las tablas de transacciones contienen registros que describen eventos de negocios. Los datos que describen a estos eventos por lo normal tienen una duración limitada de la vida útil. Por ejemplo, un registro de FACTURA es ordinariamente útil hasta que la factura se ha pagado o se ha declarado incobrable. En los sistemas de información, los registros de transacción con frecuencia se retienen *en línea* por algún lapso de tiempo. Posteriormente a la duración de su vida útil, se *archivan* fuera de línea. Los ejemplos de archivos de transacciones incluyen ÓRDENES, FACTURAS, SOLICITUDES e INSCRIPCIONES.
- Los **archivos de documentos** y las tablas de documentos contienen copias almacenadas de datos históricos para una recuperación y una revisión sencilla sin los gastos generales de regenerar el documento.
- Los **archivos de registro** y las tablas de archivos contienen los registros de archivos maestros y de transacciones que han sido suprimidos del almacenamiento en línea. Así, es raro que los registros se supriman; simplemente se mueven del almacenamiento en línea al almacenamiento fuera de línea. Los requerimientos de los archivos son determinados por los reglamentos gubernamentales y la necesidad de una auditoría o un análisis posteriores.
- Los **archivos de consulta de tablas** contienen datos relativamente estáticos que pueden ser compartidos por aplicaciones para mantener consistencia y mejorar el desempeño. Los ejemplos incluyen las TABLAS DE IMPUESTO DE VENTAS, las TABLAS DE CÓDIGO POSTAL y las TABLAS DEL IMPUESTO SOBRE LA RENTA.
- Los **archivos de auditoría** son registros especiales de actualizaciones de otros archivos, especialmente archivos maestros y de transacciones. Se usan en conjunción con archivos de registro para recobrar datos “perdidos”. Es común que los rastreos de auditoría se incorporen en las mejores tecnologías de bases de datos.

En un pasado no tan distante, los métodos de diseño de archivos requerían que el analista especificara con precisión cómo deberían secuenciarse los registros en una base de datos (lo que se llama *organización de archivos*) y cómo tendría que tenerse acceso a ellos (llamado *acceso al archivo*). En el ambiente actual de bases de datos, la misma tecnología de

---

**factor de bloques** Número de registros lógicos incluidos en una operación individual de lectura o escritura.

---

**archivo** Conjunto de todas las instancias de una estructura de registros dada.

---

**tabla** Equivalente a un archivo en una base de datos relacional.

---

**archivo maestro** Tabla de registros que son relativamente permanentes.

---

**archivo de transacciones** Tabla que contiene registros en los que se describen eventos de negocios.

---

**archivo de documento** Tabla que contiene datos históricos.

---

**archivo de registro** Tabla que contiene registros del archivo maestro y de transacciones que han sido borrados del almacenamiento en línea.

---

**archivo de consulta de tablas** Tabla que contiene datos relativamente estáticos que pueden compartirse.

---

**archivo de auditoría** Tabla que contiene registros de las actualizaciones de otros archivos.

bases de datos por lo general predetermina y/o limita la organización de archivos para todas las tablas contenidas en la base de datos. Una vez más, a un administrador capacitado en bases de datos puede dársele algún control sobre la organización, la ubicación del almacenamiento y los métodos de acceso con objeto de afinar el desempeño de la base de datos.

## > Las bases de datos

Como se describió antes, los archivos stand-alone específicos de aplicación fueron una vez el alma de la mayoría de los sistemas de información; sin embargo, están siendo reemplazados lentamente pero con seguridad por las bases de datos. Recuerde que una base de datos puede considerarse desde un punto de vista general como un conjunto de archivos interrelacionados. La interrelación significa que los registros en un archivo pueden asociarse o enlazarse con los registros en un archivo diferente.

Por ejemplo, un registro de ESTUDIANTES puede enlazarse con todos los registros de CURSOS de los estudiantes. A su vez, un registro de CURSOS puede asociarse con los registros ESTUDIANTILES que indican la terminación de ese curso. Este enlace y esta flexibilidad de dos vías nos permiten eliminar la *mayor parte* de la necesidad para almacenar redundantemente los mismos campos en los diferentes tipos de registros. Así, en un sentido muy real, los archivos múltiples se consolidan en un archivo individual: la base de datos.

La idea de relaciones entre las diferentes recopilaciones de datos se introdujo en el capítulo 7. En ese capítulo, usted aprendió a descubrir los requerimientos de datos de un sistema y a modelar esos requerimientos como *entidades* y *relaciones*. La base de datos ahora suministra la implantación técnica de esas entidades y esas relaciones.

Hoy se están construyendo tantas aplicaciones alrededor de la tecnología de bases de datos que el diseño de bases de datos se ha convertido en una habilidad importante para el analista. La historia de los sistemas de información ha conducido a una conclusión ineludible:

### ¡Los datos son un recurso que deben controlarse y administrarse!

**Arquitectura de datos** Los datos se convierten en un recurso de negocios en un ambiente de base de datos. Los sistemas de información se construyen alrededor de este recurso para proporcionar a los programadores de computadora y a los usuarios finales el acceso flexible a los datos. La **arquitectura de datos** de un negocio define cómo ese negocio desarrollará y usará tanto los archivos como las bases de datos para almacenar los datos de toda la organización, qué tecnología de archivos y de bases de datos debe usarse, y la clase de estructura administrativa que será establecida para administrar el recurso de los datos.

La figura 12.2 ilustra la arquitectura de datos hacia la cual muchas compañías han evolucionado. Como se muestra en la figura, la mayoría de las compañías todavía tienen numerosas aplicaciones convencionales de sistemas de información basadas en archivos, muchas de las cuáles se desarrollaron antes del surgimiento de la tecnología de bases de datos de alto rendimiento. En muchos casos, la eficiencia del procesamiento de estos archivos o el costo proyectado de rediseñar estos archivos han desacelerado la conversión de los sistemas a bases de datos.

Como se muestra en la figura 12.2, las **bases de datos operacionales** (o *transaccionales*) se desarrollan para soportar las operaciones cotidianas y el procesamiento de las transacciones de negocios para la gran mayoría de los sistemas de información. Estos sistemas se desarrollan (o se compran) con el paso del tiempo para reemplazar los archivos convencionales que antes soportaban las aplicaciones. El acceso a estas bases de datos se limita a programas de computadora que usan el DBMS para procesar las transacciones, mantener los datos, y generar reportes de gestión regularmente programados. También puede suministrarse algún acceso a consultas.

Muchos desarrollos de sistemas de información son renuentes a dar a los usuarios finales acceso a las bases de datos operacionales para las consultas y los reportes. El volumen de reportes y consultas no programados podrían sobrecargar a las computadoras y entorpecer las operaciones del negocio que las bases de datos debían soportar. En lugar de eso, se desarrollan los almacenes de datos (data warehouses), posiblemente en computadoras separadas.

Los **almacenes de datos** (data warehouses) almacenan datos extraídos de las bases de datos operacionales. Se usan herramientas de consulta y de soporte de decisiones para generar reportes y análisis fuera de estos almacenes de datos. Estas herramientas a menudo permiten a los usuarios extraer datos tanto de los archivos convencionales como de las bases de datos operacionales. También se le conoce como *minería de datos* (data mining).

La figura 12.2 también muestra *bases de datos personales* y *de grupos de trabajo* (o *departamentales*). La computadora personal y la tecnología local de bases de datos en red

---

#### arquitectura de datos

Definición de cómo deben desarrollarse los archivos y las bases de datos.

---

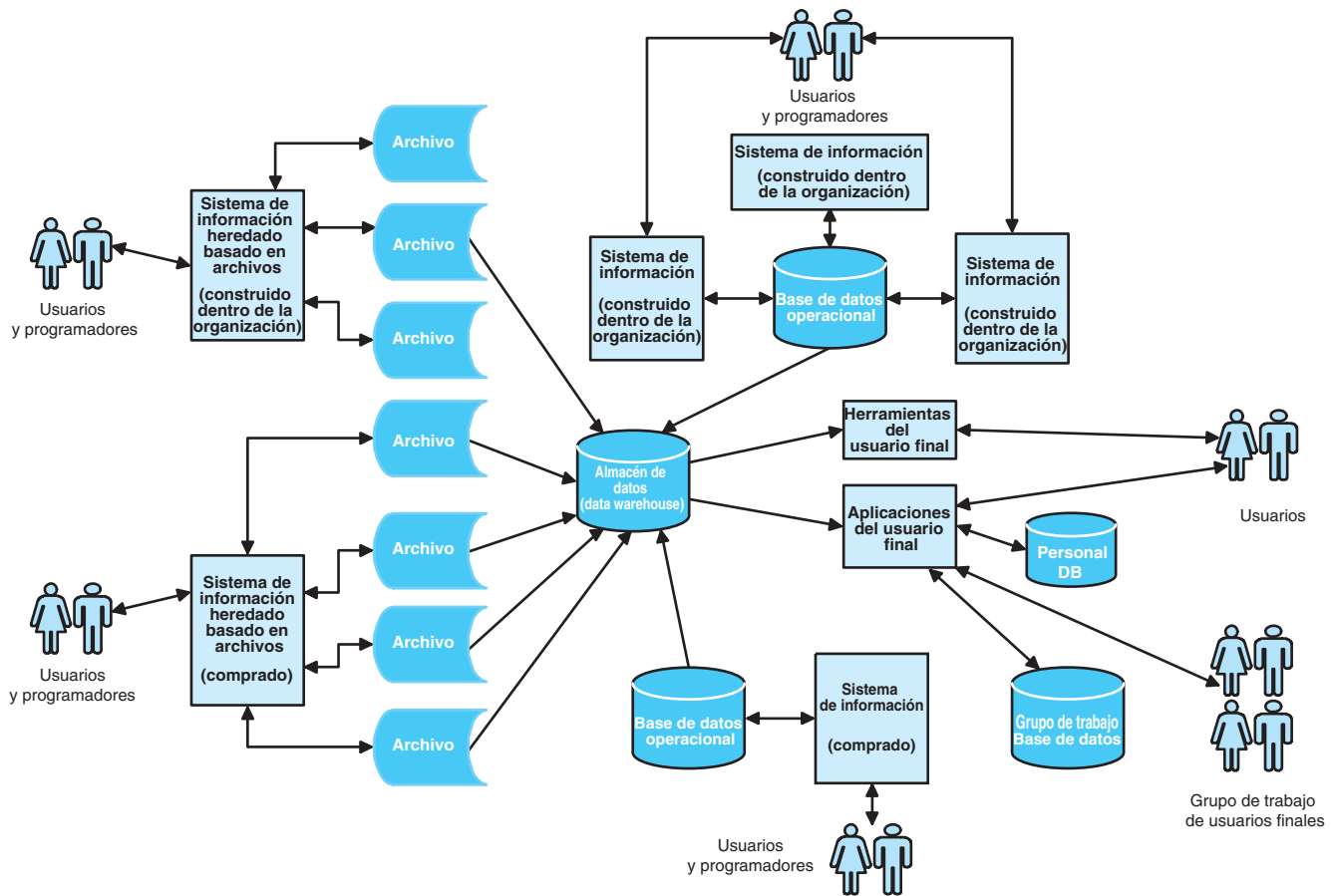
#### base de datos operacional

Base de datos que soporta las operaciones y transacciones cotidianas en relación con un sistema de información. También llamada base de *datos de transacciones*.

---

#### almacén de datos (data warehouse)

Base de datos que almacena los datos extraídos de las bases de datos operacionales.



**FIGURA 12.2** Arquitectura de datos típica y moderna

ha madurado rápidamente para permitir a los usuarios finales el desarrollo de bases de datos personales y departamentales. Estas bases de datos pueden contener datos únicos, o pueden importar datos de archivos convencionales, bases de datos operacionales y/o almacenes de datos (data warehouses). Las bases de datos personales se construyen usando tecnología de bases de datos de PC tal como *Access*, *dBASE* y *Visual FoxPro*.

La arquitectura contemporánea de datos también permite la tecnología de bases de datos a través de Internet. Por ejemplo, *Oracle 10g* provee facilidades y herramientas especiales para habilitar una base de datos a través de la Web.

Es de reconocerse que este escenario global es avanzado, sin embargo, muchas firmas están usando en la actualidad variaciones de éste. Para operar el recurso de datos a nivel de empresas, puede organizarse un equipo de especialistas en bases de datos alrededor de los siguientes administradores: un **administrador de datos** es responsable de la planeación, definición, arquitectura y administración de los datos. Uno o más **administradores de bases de datos (DBA, por sus siglas en inglés)** son responsables de la tecnología, diseño, consultas, construcción, seguridad, respaldo, recuperación y afinación del desempeño de las bases de datos. En los negocios más pequeños, estos roles pueden combinarse o asignarse a uno o más analistas de sistemas.

**Arquitectura de bases de datos** Hasta ahora, hemos hecho varias referencias a la *tecnología de bases de datos* que hace posible la antes citada arquitectura de datos. La **arquitectura de bases de datos** se refiere a la tecnología de bases de datos, incluyendo el motor de las bases de datos, las utilerías de las bases de datos, las herramientas CASE para el análisis y el diseño de las bases de datos y las herramientas de desarrollo de aplicaciones de las bases de datos. El centro de control de una arquitectura de bases de datos es su sistema de administración de bases de datos.

#### administrador de datos

Especialista en bases de datos responsable de la planeación, definición, arquitectura y administración de datos.

#### administrador de bases de datos

Especialista responsable de la tecnología, diseño, construcción, seguridad, respaldo y recuperación, así como el ajuste del desempeño de las bases de datos.

#### arquitectura de bases de datos

Tecnología de bases de datos usada como soporte para la arquitectura de los datos.

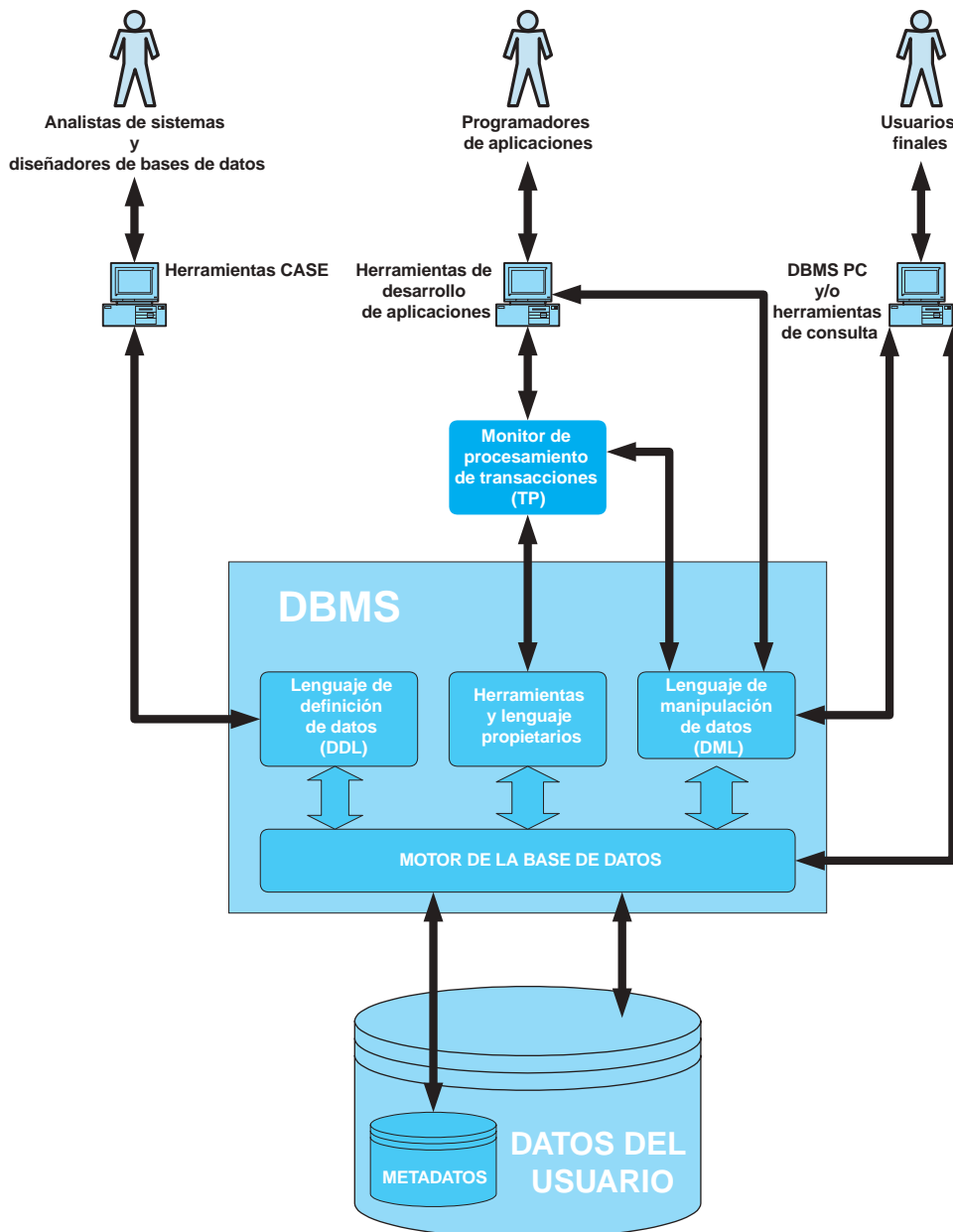


Un **sistema de administración de bases de datos (DBMS)**, por sus siglas en inglés) es software especializado de cómputo, disponible con los vendedores de computadoras, que se usa para crear, acceder, controlar y manejar la base de datos. El corazón del DBMS a menudo se llama el *motor de la base de datos*. El motor responde a órdenes específicas para crear estructuras de bases de datos y luego crear, leer, actualizar y suprimir registros en la base de datos. El sistema de administración de base de datos se le compra a un vendedor de tecnología de bases de datos tales como Oracle, IBM, Microsoft o Sybase.

La figura 12.3 describe una arquitectura típica de un sistema de administración de bases de datos. Un analista de sistemas o un analista de bases de datos diseña la estructura de los datos en términos de los tipos de registros, de los campos contenidos en esos tipos de registros y de las relaciones que existen entre los mismos. Estas estructuras se definen para el sistema de administración de base de datos usando su lenguaje de definición de datos. El **lenguaje de definición de datos (DDL)**, por sus siglas en inglés) es usado por el DBMS para establecer físicamente esos tipos de registros, campos y relaciones estructurales. Adicionalmente, el DDL define vistas de las bases de datos. Las vistas restringen la sección de una base de datos que puede usarse o a la que puede tenerse acceso por los diferentes usuarios y programas.

**sistema de administración de bases de datos (DBMS)** Software especial usado para crear, tener acceso, controlar y administrar la base de datos.

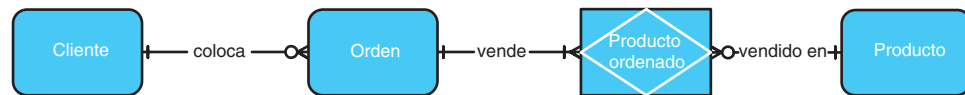
**lenguaje de definición de datos (DDL)** Lenguaje usado por un DBMS para definir una base de datos o una vista de la misma.



**FIGURA 12.3**  
Arquitectura típica de un sistema de administración de base de datos

FIGURA 12.4

Modelo lógico de datos simple



La mayoría de los sistemas de administración de base de datos almacenan tanto *datos del usuario* como *metadatos* —los datos (o especificaciones) acerca de los datos—, tales como definiciones de registro y de campo, sinónimos, relaciones de datos, reglas de validación, mensajes de ayuda y así sucesivamente. Algunos metadatos se guardan en la base de datos real, mientras otros metadatos se guardan en los repositorios de las herramientas CASE.

Para ayudar a diseñar las bases de datos, se pueden adquirir herramientas CASE, a través del vendedor de tecnología de bases de datos (por ejemplo, *Designer* de Oracle) o de un tercer vendedor de herramientas de CASE (*System Architect* de Popkin, *Visio Enterprise* de Microsoft, o *ERwin* de Computer Associates, etcétera).

El sistema de administración de bases de datos también provee un lenguaje de manipulación de datos para tener acceso y usar los datos almacenados en las aplicaciones. Se usa un **lenguaje de manipulación de datos (DML)**, por sus siglas en inglés) para crear, leer, actualizar, suprimir y navegar entre diferentes registros y tipos de registros de la base de datos; por ejemplo, de un registro de **CLIENTE** a los registros de **ÓRDENES** para ese cliente. El DBMS y el DML ocultan los detalles relativos a la forma de organizar y asignar los registros en el disco. En general, el DML es muy flexible debido a que puede ser usado por sí mismo para crear, leer, actualizar y suprimir registros o comandos, los cuáles pueden “llamarse” desde un lenguaje anfitrión de programación como el *COBOL*, *Visual Basic* o *Java*.

Muchos DBMS no requieren del uso de un DDL para construir la base de datos ni de un DML para tener acceso a la misma. En lugar de eso (o además), proporcionan sus propios comandos y herramientas para realizar esas tareas. Estas características se extienden a los DBMS basados en PCs tales como Microsoft *Access*. *Access* proporciona una interfaz de usuario gráfica simple para crear las tablas, un ambiente basado en formularios, y un lenguaje de escritura (*Visual Basic para Aplicaciones*) para acceder, navegar y mantener las tablas.

Muchos DBMS también incluyen generadores de reportes y herramientas de consulta para permitir a los usuarios el acceso y formateo de datos sin el uso directo del DML. Muchos DBMS de alto nivel se diseñan para interactuar con terminales típicas de procesamiento de transacciones para terceros.

Toda la tecnología anteriormente citada se ilustra en la figura 12.3. Actualmente casi todo el nuevo desarrollo de bases de datos está usando la tecnología relacional de bases de datos.

**Sistemas relacionales de administración de bases de datos** Hay varios tipos de sistemas de administración de bases de datos. Pueden clasificarse según la forma en que estructuran los registros. Los primeros sistemas de administración de bases de datos organizaban los registros en jerarquías o redes implantadas con índices y listas enlazadas. Actualmente, la mayoría de los sistemas exitosos de administración de bases de datos se basan en la tecnología relacional. Las **bases de datos relacionales** implantan datos en una serie de tablas bidimensionales que se “relacionan” entre sí vía claves foráneas. Cada tabla (algunas veces llamada *relación*) consta de columnas con un determinado nombre (que son los campos o los atributos) y cualquier número de filas sin denominación (que corresponden a los registros).

La figura 12.4 ilustra un modelo lógico de datos. La figura 12.5 es la implantación física relacional de las bases de datos de ese modelo de datos (llamado *esquema*). En una base de datos relacional, los archivos se ven como tablas simples bidimensionales, también conocidas como relaciones. Las filas son registros. Las columnas corresponden a los campos.

La siguiente notación abreviada para tablas se encuentra comúnmente en el diseño de sistemas y en los libros de bases de datos.

CLIENTES (NÚMERO-CLIENTE, NOMBRE-CLIENTE, SALDO-CLIENTE...)

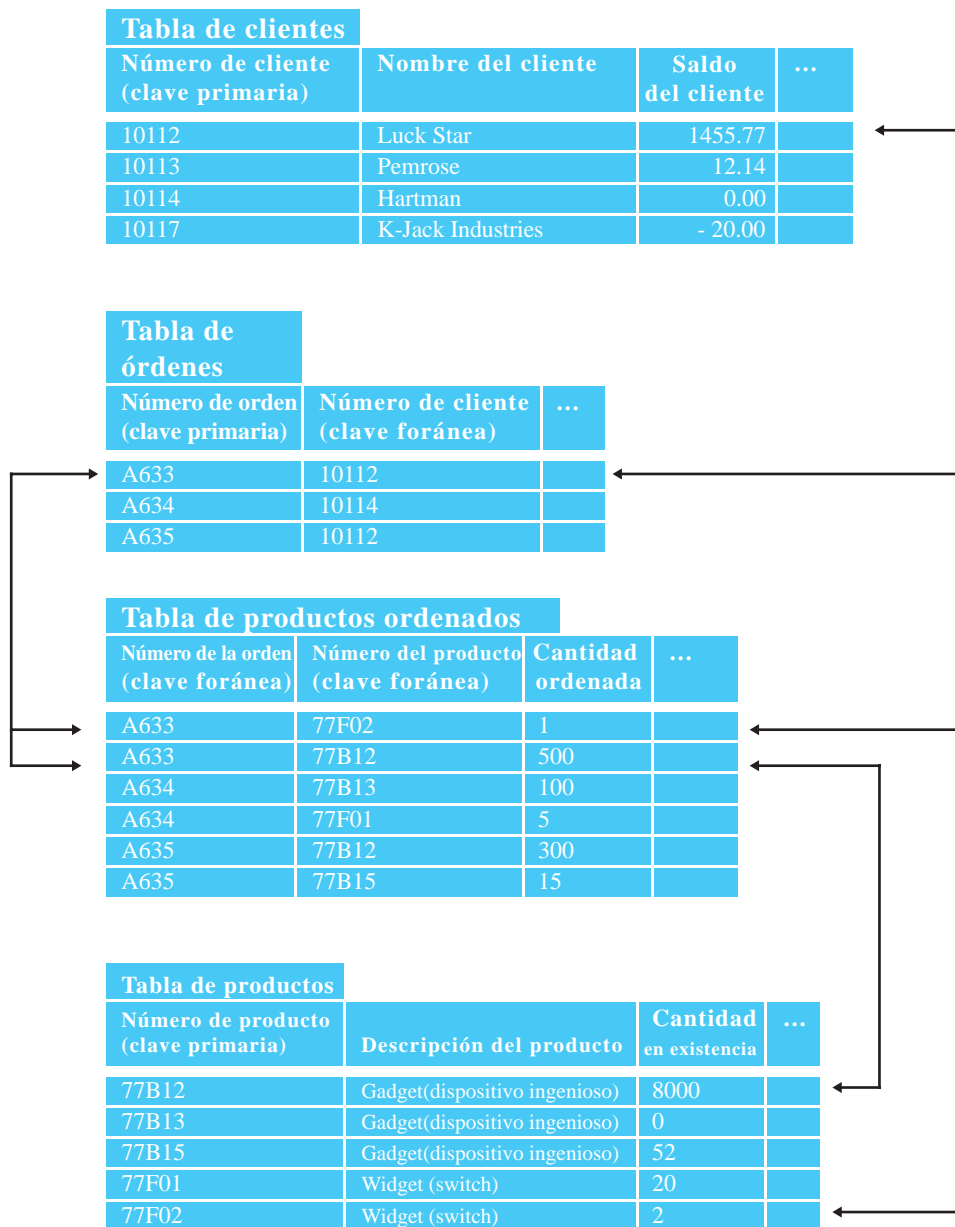
ÓRDENES (NÚMERO-ORDEN, NÚMERO-CLIENTE (FK), ...)

PRODUCTOS-ORDENADOS (NÚMERO-ORDEN (FK), NÚMERO-PRODUCTO (FK), CANTIDAD-ORDENADA...)

PRODUCTOS (NÚMERO-PRODUCTO, DESCRIPCIÓN-PRODUCTO, CANTIDAD-EN-EXISTENCIA...)

**lenguaje de manipulación de datos (DML)** Lenguaje de DBMS usado para crear, leer, actualizar y suprimir registros.

**base de datos relacional** Base de datos que implanta los datos como una serie de tablas bidimensionales que se relacionan vía las claves foráneas.

**FIGURA 12.5**

Ejemplo de un esquema de base de datos físico

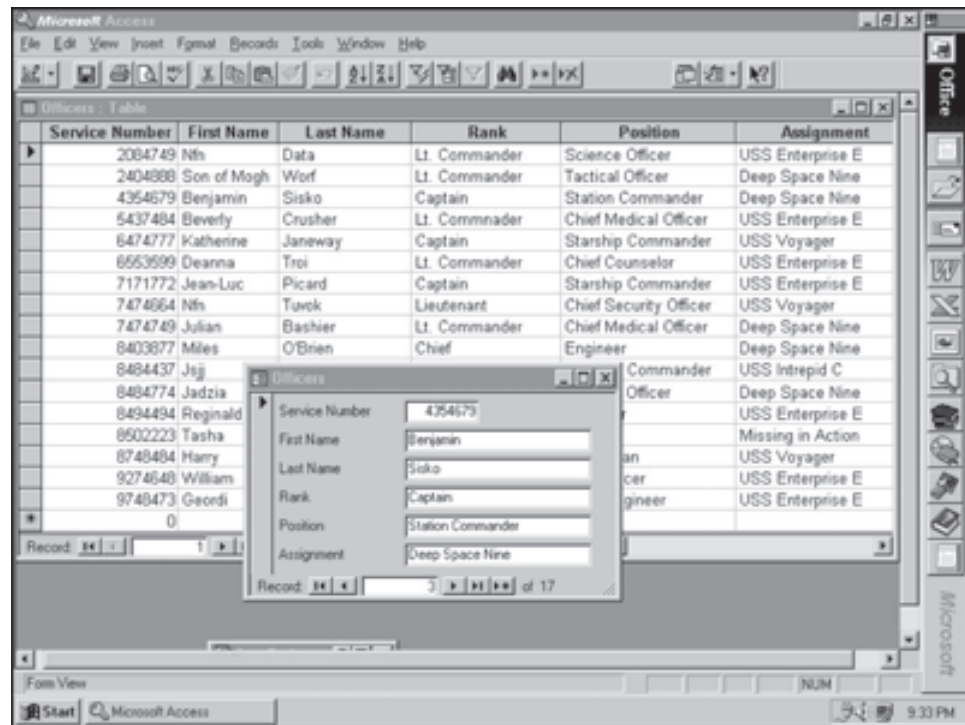
Tanto el DDL como el DML de la mayoría de las bases de datos relacionales se llaman *SQL* (pronunciado “S-Q-L” por unos y “sequel” por otros). SQL soporta la creación, el mantenimiento y el uso de la base de datos completa. Para acceder a los datos en tablas y registros, SQL provee las siguientes órdenes básicas:

- SELECCIONAR registros específicos de una tabla basada en criterios específicos (por ejemplo, SELECCIONE AL CLIENTE CUYO SALDO > 500.00).
- PROYECTAR campos específicos desde una tabla (por ejemplo, EL CLIENTE DE PROYECTO DEBE INCLUIR SÓLO NÚMERO-CLIENTE, NOMBRE-CLIENTE, SALDO).
- UNIR dos o más tablas a través de un campo común; una clave primaria y foránea (UNA EL CLIENTE Y LA ORDEN usando NÚMERO-CLIENTE).

Cuando se usan en combinación, estas instrucciones básicas pueden ocuparse de la mayoría de los requerimientos de la base de datos. Una característica fundamental del SQL es que las instrucciones regresan un conjunto de registros, no necesariamente sólo un registro individual (como en la tecnología de archivos y de bases de datos no relacionales). Las bases de datos SQL también tienen instrucciones para crear, actualizar y suprimir registros, así como para ordenarlos.

FIGURA 12.6

Interfaz de usuario diseñada con un DBMS relacional para PC (Microsoft Access)



**trigger** Programa incorporado en una tabla que se invoca automáticamente por las actualizaciones de otra tabla.

**procedimiento almacenado** Programa incorporado en una tabla que puede llamarse desde un programa de aplicación.

Las bases de datos relacionales de alto nivel también extienden el lenguaje SQL a la creación de triggers (disparadores de eventos) y procedimientos almacenados. Los **triggers** son programas incluidos dentro de una tabla que son invocados automáticamente por las actualizaciones de otra tabla. Por ejemplo, si se suprime un registro de una tabla de AERONAVE DE PASAJEROS, un trigger puede forzar la supresión automática de todos los registros correspondientes en una tabla de ASIENTOS para esa aeronave. Los **procedimientos almacenados** son programas incrustados dentro de una tabla y pueden llamarse desde un programa de aplicación. Por ejemplo, un algoritmo de validación de datos podría estar incrustado en una tabla para asegurar que los registros nuevos y actualizados contienen datos válidos *antes de que* sean guardados. Los procedimientos almacenados se escriben con una extensión propietaria de SQL, como el *Transact* de Microsoft o el *PL/SQL* de Oracle.

Tanto los triggers (disparadores) como los procedimientos almacenados son reusables porque se almacenan con las tablas mismas (como metadatos). Esto elimina la necesidad de que los programadores de la aplicación creen la lógica equivalente dentro de cada aplicación que use las tablas.

Todos los sistemas de administración de bases de datos relacionales superiores (por ejemplo, *Oracle*, *UDB/DB2* y *SQL Server*), y muchos sistemas de administración de base de datos relacionales de computadora personal (tal como *Microsoft Access*), soportan los estándares del lenguaje SQL.

Los ejemplos de DBMS relacionales de alto rendimiento incluyen a *Oracle* de Oracle Corporation, a *IBM DB2*, *SQL Server* de Microsoft (que se está usando en el proyecto SoundStage) y *Sybase* de Sybase Corporation. Muchas de estas bases de datos corren en *mainframes*, minicomputadoras y servidores de bases de datos de red. Adicionalmente, la mayoría de los DBMS de computadora personal son relacionales (o al menos en parte). Los ejemplos incluyen a *Access* y *Visual Foxpro* de Microsoft. Estos motores de bases de datos pueden funcionar tanto en computadoras personales individuales como en servidores de archivos de red de servicio local. La figura 12.6 ilustra una interfaz de usuario de un sistema de administración de base de datos relacional.

## Prerrequisito para el diseño de la base de datos: normalización

En el capítulo 7 usted aprendió a modelar requerimientos de datos para un sistema de información. Ese modelo se plasmó en un diagrama entidad-relación y en un repositorio de metadatos. El capítulo 7 también enseñó una técnica designada *análisis de datos* o

*normalización*. Esta técnica se usó para producir un modelo de datos para satisfacer los siguientes criterios de calidad:

- *Un buen modelo de datos es simple*. Por regla general, los atributos de datos que describen a una entidad deberían describir sólo a esa entidad.
- *Un buen modelo de datos es esencialmente no redundante*. Esto quiere decir que cada atributo de datos, aparte de las claves foráneas, describe a lo sumo a una entidad.
- *Un buen modelo de datos debería ser flexible y adaptable para las necesidades futuras*. Con la ausencia de estos criterios, se tiende a diseñar bases de datos que sólo satisfacen los requerimientos *actuales* del negocio.

Así que, ¿cómo logramos las metas anteriormente citadas? ¿Cómo puede diseñar usted una base de datos que pueda adaptarse a los requerimientos futuros que usted no puede predecir? La respuesta radica en el análisis de datos.

Recuerde que la normalización es una técnica de tres pasos que coloca al modelo de datos en la primera forma normal, la segunda forma normal y la tercera forma normal. El diseño de la base de datos debería proseguir sólo si el modelo lógico de datos se encuentra cuando menos en 3NF. Para una explicación más detallada, le alentamos a revisar el capítulo 7.

## Diseño convencional de archivos

Este capítulo se enfoca en el diseño de bases de datos; sin embargo, seríamos negligentes si no dijéramos unas pocas palabras acerca del diseño convencional de archivos. Primero, el diseño de archivos se simplifica por su orientación a una aplicación individual. Normalmente, los diseños de salidas y de entradas (capítulos 13 y 14) serían terminados antes que nada, ya que el diseño de archivos depende de soportar esos requerimientos aplicativos.

La mayoría de las entidades fundamentales del modelo de datos se diseñarían como registros maestros o de transacciones. Por lo común, los archivos maestros son registros de longitud fija. Las entidades asociativas del modelo de datos se conjuntan en los registros de transacción para formar registros de longitud variable (basándose en las relaciones uno a muchos). Otros tipos de archivos (no representados en el modelo de datos) se agregan como necesarios.

Dos consideraciones importantes del diseño convencional de archivos son el *acceso* y la *organización de archivos*. El analista de sistemas generalmente estudia cómo cada programa tendrá acceso a los registros en el archivo (secuencialmente o al azar) y luego selecciona una organización apropiada de archivos (por ejemplo, secuencial, indexada, haced, etcétera). En la práctica, muchos analistas de sistemas seleccionan una organización secuencial indexada (o ISAM/VSAM) para soportar la probabilidad de que diferentes programas requerirán métodos de acceso diferentes en los registros.

## El diseño moderno de la base de datos

El diseño de cualquier base de datos por lo general incluye el DBA y el personal de la base de datos. Ellos manejarán los detalles técnicos y los aspectos relacionados con una aplicación transversal (cross application issues). Aun así, es útil que el analista de sistemas entienda los principios básicos del diseño de las bases de datos relacionales.

Las reglas del diseño presentadas aquí son, de hecho, lineamientos. No podemos cubrir cada idiosincrasia. Además, puesto que SoundStage ha elegido usar el *Servidor SQL* de Microsoft como su sistema de administración de base de datos, nuestro diseño estará restringido por esa tecnología. Cada DBMS relacional presenta sus propias capacidades y restricciones. Afortunadamente, los lineamientos presentados aquí son bastante genéricos y aplicables para la mayoría de los ambientes de DBMS. Los cursos y los libros de texto de bases de datos tienen tendencia a cubrir una variedad más amplia de tecnología y productos.

La *ingeniería de sistemas asistida por computadora* (CASE, por sus siglas en inglés) ha sido un tema permanente en todo lo largo de este libro. Hay productos específicos de CASE que tratan sobre el análisis y diseño de las bases de datos (por ejemplo, *ERwin* de Computer Associates). Así mismo, la mayoría de las herramientas CASE de uso general ahora incluyen herramientas de diseño de bases de datos. En este ejemplo continuamos

usando al producto CASE *System Architect* de Popkin para el estudio de casos SoundStage. Por último, la mayoría de las herramientas CASE (incluyendo el *System Architect*) pueden generar automáticamente el código SQL para construir las estructuras de bases de datos para los sistemas de administración de bases de datos más populares. Esta capacidad de generación de código ahorra mucho tiempo.

### > Metas y prerequisites del diseño de bases de datos

Las metas del diseño de bases de datos son las siguientes:

- Una base de datos deberá proporcionar un eficiente almacenamiento, actualización y recuperación de los datos.
- Una base de datos deberá ser confiable: los datos almacenados deberán tener una integridad alta para promover la confianza del usuario sobre los mismos.
- Una base de datos deberá ser adaptable y escalable para aplicaciones y requerimientos nuevos e imprevistos.
- Una base de datos deberá soportar los requerimientos del negocio del sistema de información.

El modelo lógico de datos del sistema —en nuestro caso, un diagrama entidad-relación (ERD, por sus siglas en inglés) totalmente atribuido y normalizado— sirve como un prerequisite. Este modelo, del capítulo 7, se reproduce en la figura 12.7. Cada atributo en ese modelo debe definirse en cuanto a su tipo de datos, dominio y valores por omisión. Estas propiedades también se revisaron en el capítulo 7.

### > El esquema de la base de datos

El diseño de una base de datos se estructura como un modelo especial llamado esquema de bases de datos. Un **esquema de base de datos** es el *modelo físico* o esquema para una base de datos. Representa la implantación técnica del modelo lógico de datos. (*El System Architect* lo llama *modelo físico de datos*.)

NOTA: Debemos reconocer aquí alguna terminología potencialmente confusa. Estamos usando los términos *lógicos* y *físicos* de una manera consistente con los capítulos anteriores en este libro. Desafortunadamente, la mayoría de los libros sobre bases de datos usan los términos *conceptuales* (para referirse al modelo *lógico*) y *lógicos* (para referirse al modelo *físico*). Nos disculpamos por esta confusión inevitable de la industria.

Un esquema relacional de bases de datos define la estructura de la base de datos en términos de tablas, claves, índices y reglas de integridad. Un esquema de la base de datos especifica detalles basándose en las capacidades, terminología y restricciones del sistema de administración de base de datos escogido. Cada DBMS soporta tipos de datos y reglas de integridad, que son diferentes.

La transformación del modelo lógico de datos en un esquema físico relacional de bases de datos está dictada por algunas reglas y opciones bastante genéricas. Estas reglas y estos lineamientos se resumen como sigue:

1. Cada entidad fundamental, asociativa y débil se implanta como una tabla separada. Los nombres de las tablas tal vez tengan que formatearse según las reglas de denominación y las limitaciones de tamaño del DBMS. Por ejemplo, una entidad lógica llamada PRODUCTO ORDENADO por el MIEMBRO podría cambiar a una tabla física llamada tblMemberOrd-Prod. El prefijo y la compresión de espacios son consistentes con los estándares contemporáneos de denominación y los lineamientos en los lenguajes modernos de programación.
  - a) La clave primaria se identifica como tal y se implanta como un índice en la tabla.
  - b) Cada clave secundaria se implanta como su propio índice en la tabla.
  - c) Un índice deberá crearse para cualesquiera atributos sin clave que se identificaron como requerimientos dentro de un subconjunto de criterios (capítulo 7).
  - d) Cada clave foránea será implantada como tal. La inclusión de estas claves foráneas implanta las relaciones en el modelo de datos y permite que las tablas se unan en los programas de aplicación y de SQL.
  - e) Los atributos serán implantados como campos. Estos campos corresponden a las columnas en la tabla. Los siguientes detalles técnicos generalmente deben especificarse para cada atributo. (Estos detalles pueden inferirse de manera automática por la herramienta CASE de acuerdo a las descripciones lógicas del modelo de datos.)

#### esquema de base de datos

Modelo que representa la implantación técnica de una base de datos.

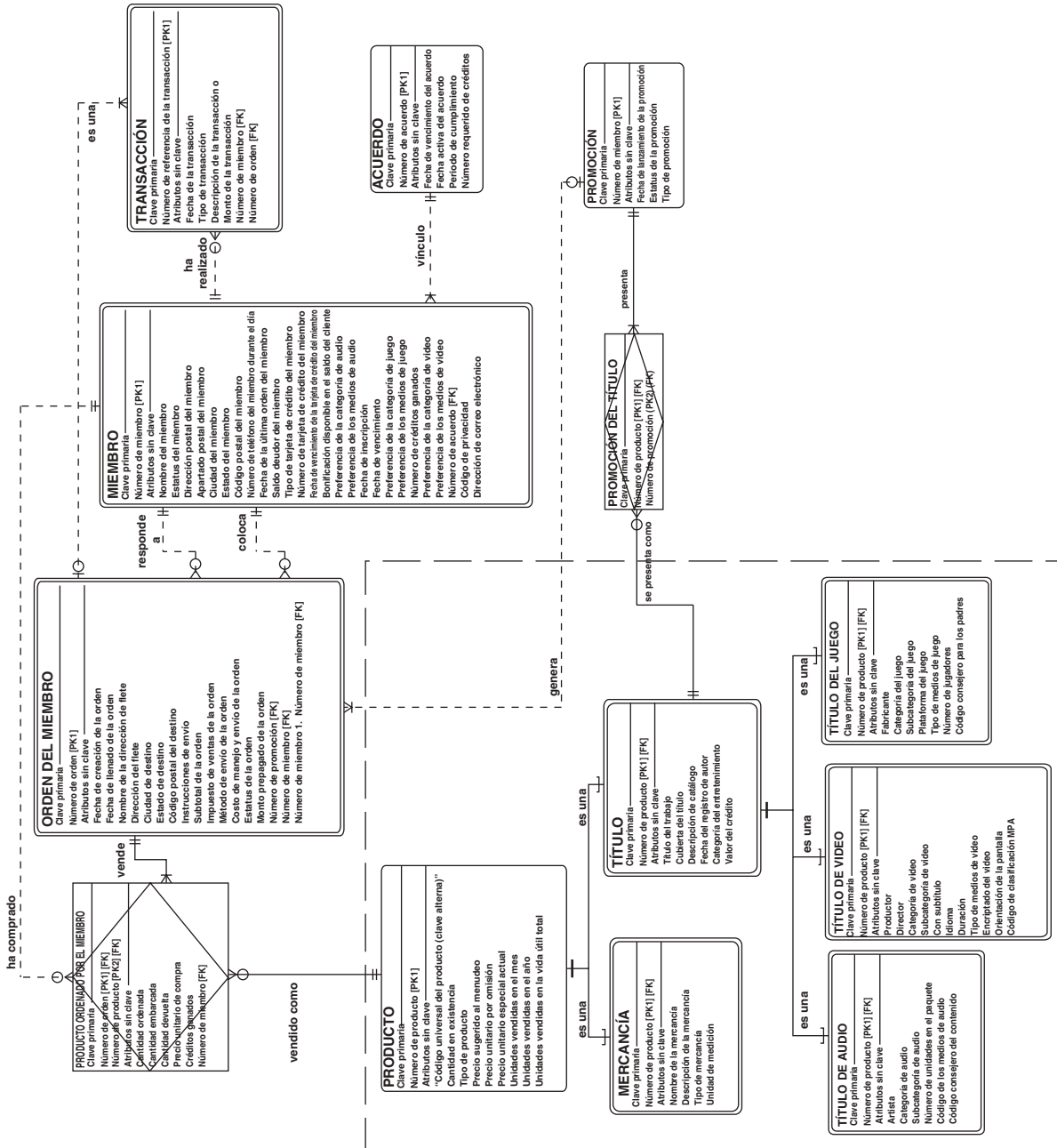


FIGURA 12.7 Modelo lógico de datos de SoundStage en la tercera forma normal



Tal vez los nombres de los campos tengan que ser acortados y reformateados según las restricciones y las reglas internas del DBMS. Por ejemplo, en el modelo lógico de datos, la mayoría de los atributos podrían tener como prefijo el nombre de la entidad (por ejemplo, el NOMBRE DEL MIEMBRO). En la base de datos física, nosotros simplemente usaríamos NOMBRE.

- i) *Tipo de datos.* Cada DBMS soporta tipos diferentes de datos y términos para esos tipos de datos. La figura 12.8 muestra diferentes tipos de datos físicos para algunos sistemas de administración de bases de datos diferentes.
  - ii) *Tamaño del campo.* Los DBMS diferentes expresan la precisión de los números reales en forma diferente. Por ejemplo, en el *Servidor SQL*, una especificación de tamaño de NÚMERO (3,2) soporta un rango de  $-9.99$  para  $9.99$ .
  - iii) *NULO O NO NULO.* ¿Debe tener el campo un valor antes de que el registro pueda pasar al almacenamiento? De nuevo, diferentes DBMS pueden requerir distintas palabras reservadas para expresar esta propiedad. Por definición, nunca deberá permitirse que las claves primarias tengan valores NULOS.
  - iv) *Dominios.* Muchos sistemas de administración de base de datos pueden editar datos de manera automática para asegurar que los campos contienen datos legales. Éste puede ser un gran beneficio para asegurar que la integridad de los datos sea independiente de los programas de aplicación. Si el programador se equivoca, el DBMS (sistema de administración de base de datos) percibe el error. Pero para los DBMS que soportan la integridad de los datos, las reglas deben especificarse con precisión en un lenguaje que sea comprendido por el DBMS.
  - v) *Valores por omisión (default).* Muchos sistemas de administración de bases de datos permiten la designación automática de un valor por omisión en caso de que un usuario o un programador creen un registro que contenga campos sin valores. En algunos casos, *NULL* sirve como el valor por omisión.
  - vi) Nuevamente, muchas de las especificaciones antes citadas se documentaron como parte de un modelo lógico de datos completo. Si ese modelo de datos fue desarrollado con una herramienta CASE, ésta puede ser capaz de traducir automáticamente el modelo de datos al lenguaje físico de la tecnología escogida de bases de datos.
2. Las entidades de supertipo/subtipo presentan opciones adicionales como sigue:
- a) Cada supertipo y cada subtipo pueden implantarse como una tabla separada (todos teniendo la misma clave primaria).
  - b) O bien, si los subtipos son de *tamaño* y contenidos de datos *similares*, un administrador de bases de datos puede elegir colapsar los subtipos en el supertipo para crear una tabla individual. Esto presenta ciertos problemas para establecer valores por omisión y para verificar los dominios. En un DBMS de alto nivel, estos problemas pueden subsanarse incluyendo la lógica por omisión y de dominio en *los procedimientos* almacenados para la tabla.
  - c) Por otra parte, los atributos del supertipo podrían duplicarse en una tabla para cada subtipo.
  - d) Podría usarse alguna combinación de las opciones antes citadas.
3. Evalúe y especifique las restricciones de integridad referencial (que se describen en la siguiente sección).

El esquema de la base de datos de SoundStage fue generado de manera automática a partir del modelo lógico de datos a través de la herramienta CASE *System Architect*. Dicho esquema se ilustra en la figura 12.9. Solicitamos su atención para que considere los siguientes aspectos numerados dentro de la figura:

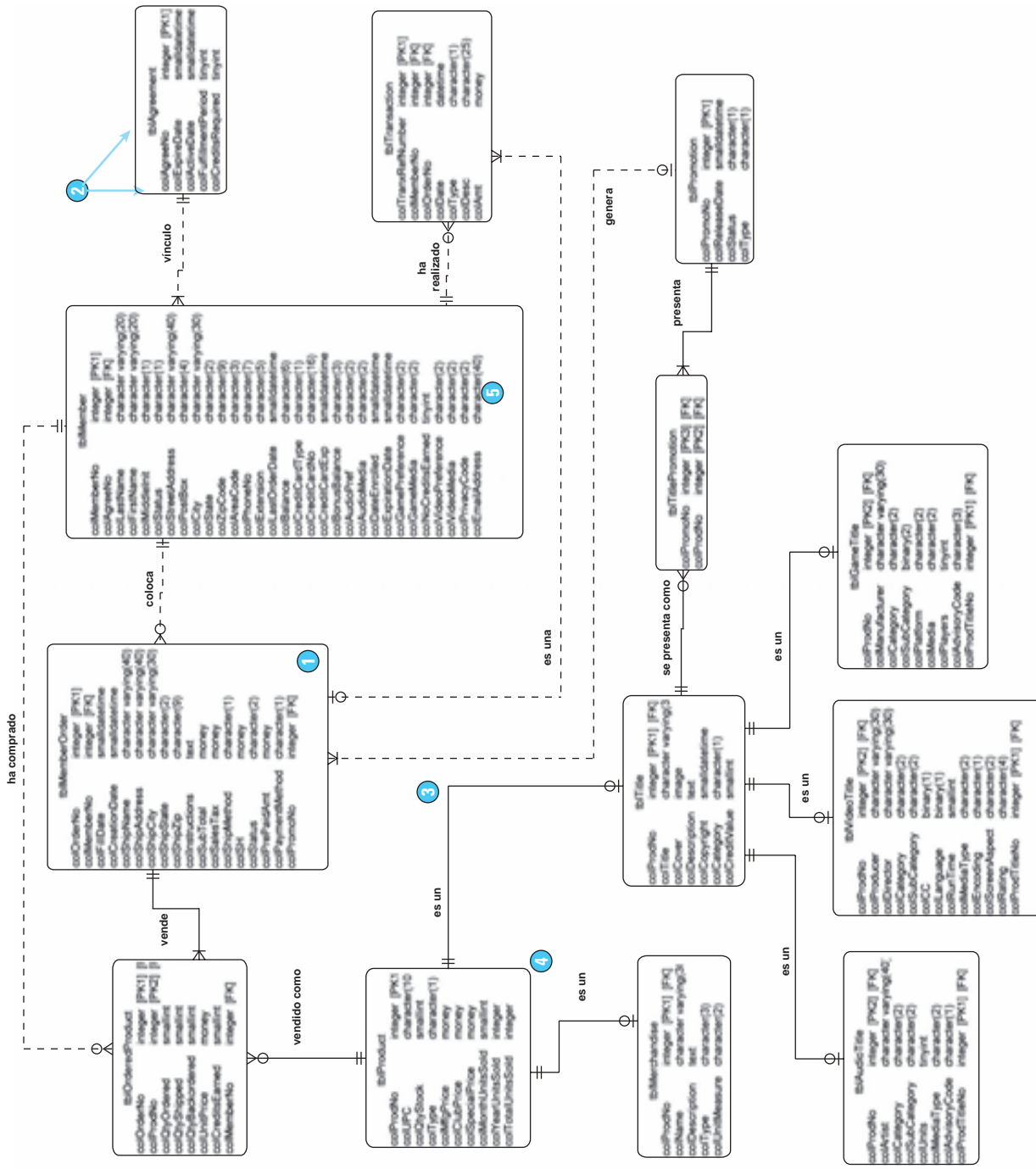
- 1 Cada rectángulo redondeado define una tabla. Las filas nombradas en el rectángulo realmente corresponden a las columnas nombradas que se crearán para la tabla.
- 2 SoundStage ha definido una convención estándar de denominación para tablas y columnas. Las convenciones se basan en los lineamientos de programación llamados la *Notación Húngara*. Cada objeto es nombrado sin espacios, guiones o subrayados. Y a cada objeto se le da un prefijo que define todos los objetos similares. Para los objetos de la base de datos se usaron los siguientes estándares:

tbl Indica una tabla de base de datos.  
col Indica una columna en la tabla.



**FIGURA 12.8** Lista parcial de los tipos de datos físicos para diferentes tecnologías de bases de datos

Tipo de dato lógico (para almacenarse en el campo)	Tipo de dato físico Microsoft Access	Tipo de dato físico Microsoft SQL Server	Tipo de dato físico Oracle
Datos con caracteres de longitud fija (úsese en campos con datos con caracteres de relativamente fija)	TEXTO	CARACTER (tamaño) o caracter (tamaño)	CARACTER (TAMAÑO)
Datos con caracteres de longitud variable (úsese en campos que requieran datos con caracteres pero en los cuales el tamaño varía demasiado; como en DIRECCIÓN)	TEXTO	CARACTERVARIABLE (tamaño máximo) o caracter que varía (tamaño máximo)	CARACTERVARIABLE2 (tamaño máximo)
Datos con caracteres muy largos (úsese para descripciones y notas largas; generalmente no más de un campo de este tipo por registro)	MEMO	TEXTO	CARACVAR LARGO o CARACVAR LARGO2
Número entero	NÚMERO	ENT(tamaño) o entero o enteropequeño o enteroinsignificante	ENTERO o NÚMERO (tamaño) o enteropequeño o byte
Número decimal	NÚMERO	DECIMAL (tamaño, cifras decimales) o NUMÉRICO (tamaño, cifras decimales)	DECIMAL (tamaño, cifras decimales) o NÚMERO (tamaño, cifras decimales) o NÚMERO
Número financiero	MONEDA	DINERO o POCODINERO	vea número decimal
Fecha (con la hora)	FECHA/HORA	FECHAHORA o FECHAHORAPEQUEÑA Dependiendo de la precisión necesaria	FECHA
Hora exacta (úsese para almacenar la fecha y la hora del reloj del sistema de la computadora)	no está soportado	HORAESTAMPA	HORAESTAMPA
Sí o No; o Verdadero o Falso	SÍ/NO	BIT	use CARÁCTER(1) y establezca un dominio de sí o no
Imagen	OBJETO OLE	IMAGEN	SINRETOQUELARGA
Hipervínculo	HIPERVÍNCULO	VARBINARIA	SINRETOQUE
¿Puede el diseñador definir nuevos tipos de datos?	NO	SÍ	SÍ



**FIGURA 12.9** Esquema inicial de la base de datos física de SoundStage

Aunque no se describan en el esquema, pueden incluirse otros prefijos comunes de bases de datos en el diccionario de datos (repositorio) de modo que esos prefijos puedan usarse para generar código correcto. Las posibilidades incluyen:

- db Indica la base de datos misma.
- idx Indica un índice construido para una tabla.
- dom Indica un dominio que puede aplicarse a uno o más campos.

- 3 Relaciones lógicas, tanto con identificación como sin identificación, se transforman en *restricciones* que se implantan usando las claves foráneas.
- 4 Elegimos hacer que cada entidad de supertipo y subtipo en la jerarquía lógica de generalización se sitúe en su propia tabla física. (Ésta fue la opción por omisión para el generador del modelo físico de datos del *System Architect*.)
- 5 Observe que el *System Architect* infirió automáticamente los tipos de datos físicos para cada campo basado en 1) la selección de *Servidor SQL* de Microsoft como el sistema objetivo de administración de bases de datos, y 2) en los tipos de datos lógicos que habíamos definido para los atributos de cada entidad durante el análisis de los sistemas. Los tipos de datos físicos pueden cambiar para reducir el espacio de almacenamiento requerido, para mejorar la integridad de los datos o para representar mejor todos los valores posibles incluidos en el dominio.

Aunque no se representa en el esquema de la base de datos, el generador del esquema también crea un índice para cada clave primaria indicada en el mismo. Usted puede agregar índices adicionales para claves secundarias únicas (como el campo del código universal de productos, o UPC, en la tblProducto) o para cualquier atributo sin clave que pueda usarse para formar subconjuntos de todos los registros en una tabla (como tblTransaction.colType). Estos índices pueden mejorar el desempeño de la base de datos final.

Algunas herramientas CASE generan esquemas de bases de datos con más detalle que el mostrado anteriormente. Por ejemplo, algunos esquemas de bases de datos indican para cada campo si éste debe adoptar o no un valor:

- NULL significa que el campo no necesita tener un valor.
- NOT NULL significa que el campo debe tener un valor. Ya que se usan claves primarias para tener un acceso único a los registros, ningún campo PK (clave primaria) puede adoptar valores NULL.

¿Querría usted arreglar las entidades en tercera forma normal al diseñar la base de datos? Por ejemplo, ¿querría usted combinar dos entidades de la tercera forma normal en una tabla individual? (por omisión, eso ya no estaría en la tercera forma normal). ¡Generalmente no! Aunque un administrador de bases de datos puede hacer tales arreglos para mejorar el desempeño de la base de datos, él o ella deberían ponderar cuidadosamente las ventajas y las desventajas de hacer esto. Aunque tales arreglos pudieran implicar más comodidad al tener menos tablas o un lograr un desempeño general mejor, tales combinaciones también pueden conducir a la posible pérdida de independencia de datos; si los campos nuevos o futuros necesitaran redividir la tabla en dos tablas, los programas tendrían que ser reescritos. Por regla general, no se recomienda combinar las entidades en tablas.

## > Los datos y la integridad referencial

La integridad de la base de datos tiene que ver con la confianza. ¿Pueden confiar el negocio y sus usuarios en los datos almacenados en la base de datos? La integridad de los datos provee los controles internos necesarios para la base de datos. Existen al menos tres tipos de integridad de datos que deben diseñarse como parte de cualquier base de datos.

**Clave de integridad** Cada tabla deberá tener una clave primaria (que puede ser concatenada). La clave primaria debe estar controlada de modo que no haya dos registros en la tabla que tengan el mismo valor de la clave primaria. (Observe que para una clave concatenada, el valor concatenado debe ser único; no los valores individuales que forman la concatenación.)

Además, nunca debe permitirse que la clave primaria de un registro tenga un valor NULL. Eso daría por terminado el propósito de la clave primaria, de identificar el registro en forma única.

Si el sistema de administración de base de datos no implanta estas reglas, deben tomarse otros pasos para asegurarlas. La mayoría de los DBMS implantan la integridad de claves.

**Integridad de dominio** Deben diseñarse controles apropiados para asegurar que ningún campo adopte un valor que esté fuera del rango de los valores legales. Por ejemplo, si el PROMEDIO GENERAL se define como un número entre 0.00 y 4.00, entonces deben implantarse controles para impedir números negativos y números mayores que 4.00.

No hace mucho, se esperaba que los programas de aplicación realizaran la edición de todos los datos. Hoy la mayoría de los sistemas de administración de base de datos son capaces de implantar reglas de dominio. Para el futuro previsible, la responsabilidad de editar los datos continuará siendo compartida entre los programas de aplicación y el DBMS.

**Integridad referencial** La arquitectura de las bases de datos relacionales implanta relaciones entre los registros en tablas vía *Claves foráneas*. El uso de claves foráneas aumenta la flexibilidad y la escalabilidad de cualquier base de datos, pero también aumenta el riesgo de errores de integridad referencial. Un error de **integridad referencial** existe cuando un valor de clave foránea en una tabla no tiene valor de clave primaria coincidente en la tabla relacionada. Por ejemplo, una tabla de FACTURAS generalmente incluye una clave foránea, NÚMERO DE CLIENTE, para “hacer referencia a” la clave primaria coincidente de NÚMERO DE CLIENTE en la tabla de CLIENTES. ¿Qué pasa si suprimimos un registro del CLIENTE? Existe *la posibilidad* de que podamos tener registros de FACTURA cuyo NÚMERO DE CLIENTE no tiene registro coincidente en la tabla de CLIENTES. Esencialmente, hemos arreglado la integridad referencial entre las dos tablas.

¿Cómo impedimos errores de integridad referencial? Una de dos cosas deberá ocurrir. Cuando consideramos eliminar los registros del CLIENTE, a su vez debemos borrar automáticamente todos los registros de FACTURA que tengan un NÚMERO DE CLIENTE coincidente (que no tenga mucho sentido en el negocio) o bien, deberemos denegar la supresión del registro CLIENTE hasta que hayamos suprimido todos los registros de FACTURA relacionados con el mismo.

La integridad referencial se especifica en forma de reglas de supresión como sigue:<sup>2</sup>

- *Sin restricciones.* Puede suprimirse cualquier registro en la tabla sin considerar ningún otro registro en cualquier otra tabla.

Al observar el modelo de datos final de SoundStage, no pudimos aplicar esta regla a ninguna tabla.

- *Suprimir:en cascada.* La supresión de un registro en la tabla debe ser automáticamente seguida por la supresión de los registros coincidentes en una tabla relacionada. Muchos DBMS relacionales pueden implantar de manera automática las reglas de suprimir:en cascada usando triggers.

En el modelo de datos SoundStage, un ejemplo de una regla válida de suprimir:en cascada sería de ORDEN DEL MIEMBRO al PRODUCTO ORDENADO POR EL MIEMBRO. En otras palabras, si suprimimos una ORDEN DEL MIEMBRO específica, deberemos suprimir automáticamente todos los PRODUCTOS ORDENADOS POR EL MIEMBRO coincidentes para esa orden.

- *Suprimir:restringir.* La supresión de un registro en la tabla debe denegarse hasta que todos los registros coincidentes se supriman de una tabla relacionada. Otra vez, muchos DBMS relacionales pueden implantar automáticamente las reglas de suprimir:restringir.

Por ejemplo, en el modelo de datos SoundStage, podríamos especificar que deberemos delegar la supresión de cualquier PRODUCTO siempre que existan PRODUCTOS ORDENADOS POR EL MIEMBRO para ese producto.

- *Suprimir:establecer el valor nulo.* La supresión de un registro en la tabla debe ser seguida automáticamente de la asignación del valor NULL a cualesquiera claves coincidentes en una tabla relacionada. De nuevo, muchos DBMS relacionales pueden implantar tal regla a través de triggers.

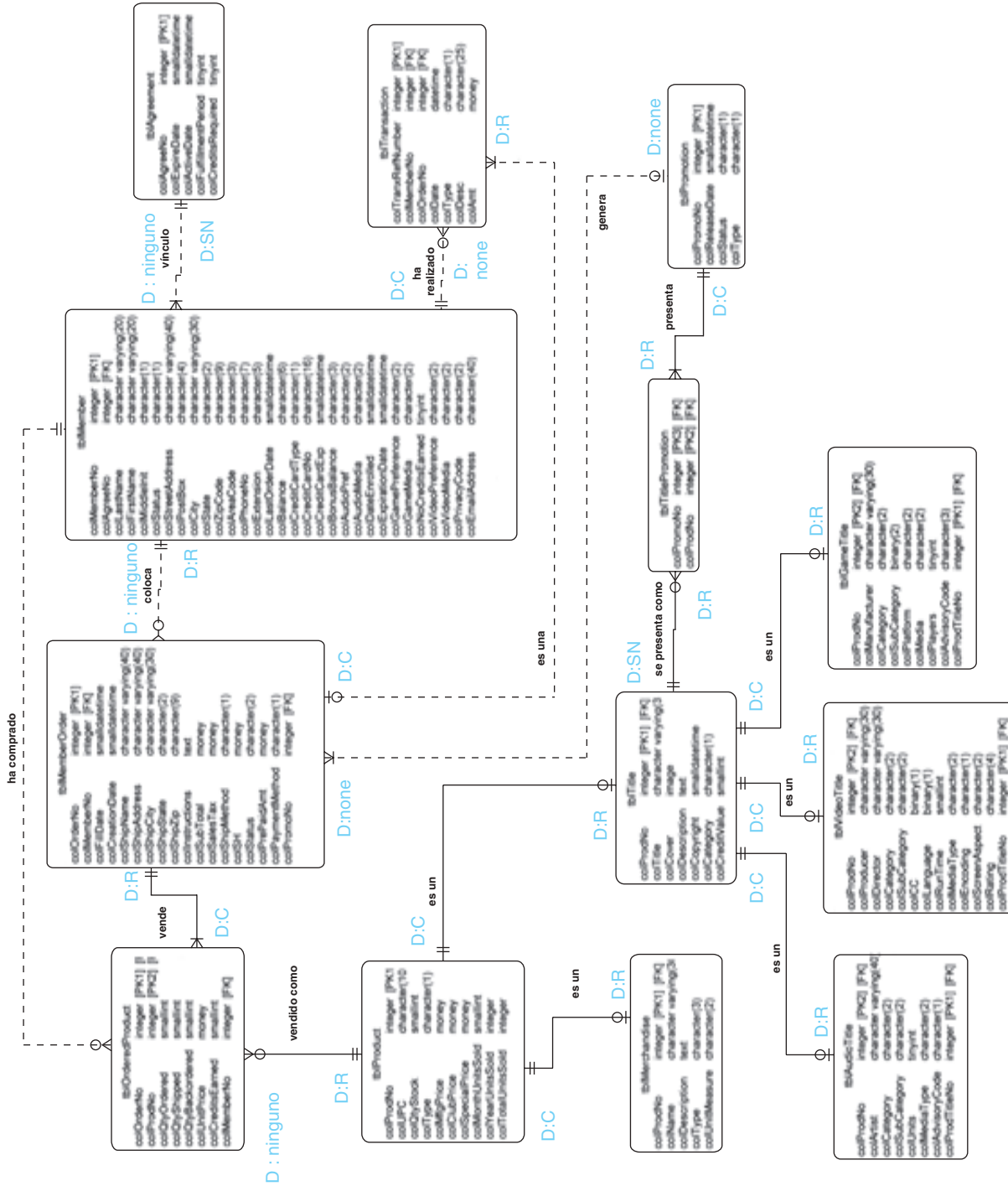
La opción suprimir:establecer el valor nulo no se usó en el modelo de datos de SoundStage. Se usa sólo cuando usted está dispuesto a suprimir un registro de tabla maestra, pero no quiere suprimir los registros correspondientes de la tabla de transacciones por razones históricas. Al asignar el valor NULL a la clave foránea, usted reconoce que el registro no apunta a un registro maestro correspondiente, pero por lo menos no lo tiene señalando un registro maestro que no existe.

El esquema final de la base de datos, se complementa con las reglas de integridad referencial, el cual se ilustra en la figura 12.10. Éste es el esquema usado para escribir el código SQL (o el equivalente) para crear las tablas y las estructuras de datos.

### integridad referencial

Seguridad de que un valor de clave foránea en una tabla tiene un valor de clave primaria correspondiente en la tabla relacionada.

<sup>2</sup> Los estudiantes de bases de datos bien informados saben que también hay reglas de inserción y de actualización para la integridad referencial. El estudio completo de estas reglas se deja a los cursos y libros de texto de bases de datos.



**FIGURA 12.10** Esquema final de la base de datos física de SoundStage

## > Los roles

Algunos estándares de bases de datos insisten en que no haya dos campos que tengan *exactamente* el mismo nombre. Esta restricción simplifica la documentación, los sistemas de ayuda y las definiciones de metadatos. Esto presenta un problema obvio con las claves foráneas. Por definición, una clave foránea debe tener una clave primaria correspondiente. Durante la modelación de datos *lógicos*, usar el mismo nombre satisface nuestro propósito de ayudar a los usuarios a entender que las claves foráneas nos permiten aparear los registros relacionados en entidades diferentes. Pero en una base de datos *física*, no siempre es necesario, o aun deseable, tener a estos nombres de campo redundantes en la base de datos.

Para arreglar este problema, a las claves foráneas se les puede dar nombres de roles. Un **nombre de un rol** es un nombre alterno para una clave foránea que claramente distingue el propósito que satisface la clave foránea en la tabla. Por ejemplo, en el esquema de base de datos de SoundStage, `NÚMERO_PRODUCTO` es una clave primaria para la tabla de `PRODUCTOS` y una clave foránea en la tabla de `PRODUCTOS ORDENADOS POR EL MIEMBRO`. El nombre no deberá cambiar en la tabla de `PRODUCTOS`. Sin embargo, puede tener sentido renombrar la clave foránea como `NÚMERO_PRODUCTO_ORDENADO` para reflejar más exactamente su papel en la tabla de `PRODUCTOS ORDENADOS POR EL MIEMBRO`.

La decisión de requerir o no nombres de los roles generalmente es establecida por el administrador de datos o de la base de datos.

## > Distribución y réplica de la base de datos

En el capítulo 7, “Modelado y análisis de datos,” introdujimos brevemente el concepto del análisis de distribución de datos lógicos. El *análisis de distribución de datos* establece cuáles posiciones comerciales necesitan acceder a cuáles entidades y atributos de datos lógicos.

Usamos una matriz simple en el capítulo 7 para mapear las entidades y los atributos a las posiciones. Muchas herramientas CASE, incluyendo el *System Architect*, incluyen facilidades para construir tal matriz. Deberíamos dar ahora alguna consideración al impacto del análisis de la distribución de datos en el diseño de bases de datos.

En el mundo actual, centrado en las redes, de multicapa, del cliente/servidor, los sistemas de información y las bases de datos raramente se centralizan. En lugar de eso, se distribuyen a través de una red que puede extenderse a lo largo de muchos edificios, ciudades, estados o países. En consecuencia, podemos necesitar dividir, distribuir o reproducir todo o parte del diseño de la base de datos a diferentes servidores físicos de bases de datos en diferentes posiciones físicas. Básicamente, necesitamos realizar un análisis de distribución de bases de datos físicos que tome en consideración lo que aprendimos durante nuestro análisis de distribución de datos lógicos.

En lo esencial, tenemos disponibles varias opciones de distribución:

- La *centralización* de la base de datos. En otras palabras, implantaríamos la base de datos en un servidor individual independientemente del número de posiciones físicas que pueden requerir acceso a ella. Esta solución es simple y es la más fácil de mantener; sin embargo, viola una regla de administración de datos que ha cobrado importancia para muchos administradores y usuarios de datos: los datos deberán estar ubicados tan cerca como sea posible de los usuarios.
- La *distribución horizontal* de datos. En esta opción, cada tabla (o las filas enteras en una tabla) se asignarían a diferentes servidores y posiciones de la base de datos. Esta opción conduce a un acceso y seguridad eficientes porque cada posición tiene sólo las tablas y filas requeridas para esa posición. Desafortunadamente, los datos no siempre pueden recombinarse con facilidad para el análisis de administración a través de los sitios.
- La *distribución vertical* de los datos. En esta opción, las columnas específicas de las tablas se asignan a servidores y bases de datos específicos. Las ventajas y las desventajas son muy parecidas a las de una distribución horizontal.
- La *replicación* de los datos. La replicación se refiere a la duplicación física de tablas enteras para las posiciones múltiples. La mayoría de los sistemas de administración de base de datos para empresas de alto nivel, incluyen tecnología de replicación que coordina las actualizaciones de las tablas y los registros duplicados para mantener la integridad de los datos. Esta solución ofrece ventajas de desempeño y de accesibilidad y reduce el tráfico de la red, pero también aumenta la complejidad de la integridad de los datos y requiere más capacidad física de almacenamiento.

---

**nombre de un rol** Nombre de la clave foránea que refleja el propósito para el que sirve ésta en una tabla.

Estas alternativas no son mutuamente excluyentes. El diseñador debe planificar con cuidado los grados de distribución y replicación de datos.

Dado nuestro esquema físico de base de datos, podemos definir *vistas* que corresponden a posiciones geográficas específicas (y las subvistas para diferentes usuarios y aplicaciones). Una vista de la base de datos puede ser muy selectiva. Puede incluir un subconjunto específico de tablas, un subconjunto específico de columnas en las tablas, o un subconjunto específico de registros en las tablas. Cada vista debe sincronizarse cuidadosamente con el esquema de la base de datos maestra de modo que los cambios en el esquema maestro puedan propagarse a las vistas si eso es lo apropiado. Las herramientas CASE pueden ser muy útiles para definir las vistas y mantener a todas ellas en sincronía.

Para el proyecto SoundStage, pensamos reproducir la base de datos entera en cada una de las tres ciudades. La integridad de los datos para tablas comunes será implantada usando la tecnología de replicación del *Servidor SQL*. El analista de sistemas comúnmente no programará las reglas de replicación. Un analista o un administrador capacitado en bases de datos harán eso. Como implantaremos el esquema completo de la base de datos físicos en el servidor de cada ciudad, no hay necesidad de definir vistas para nuestro proyecto.

### > Los prototipos de bases de datos

La construcción de prototipos no es una alternativa para los esquemas de bases de datos cuidadosamente meditados. Por otra parte, una vez que el esquema se completa, es común que pueda generarse de manera muy rápida una base de datos del prototipo. La mayoría de los DBMS modernos incluyen generadores poderosos de bases de datos, accionados por un menú que en automático crean un DDL y generan una base de datos del prototipo de ese DDL (lenguaje de descripción de datos). Entonces puede cargarse una base de datos con datos de prueba que resultarán ser útiles para la construcción de prototipos y para probar las salidas, las entradas, las pantallas y otros componentes de sistemas.

### > Planificación de la capacidad de las bases de datos

Una base de datos se guarda en disco. Al final, el administrador de la base de datos querrá una estimación de la capacidad de disco de la nueva base de datos para asegurar que se tiene suficiente espacio disponible en el disco. La planificación de la capacidad de la base de datos puede calcularse con aritmética simple como sigue. Esta fórmula simple ignora factores como el empaquetado, el codificado, y la compresión, pero al omitir esas posibilidades, usted está añadiendo capacidad de holgura.

1. Para cada tabla, sume los tamaños del *campo*. Éste es el *tamaño del registro* para la tabla. Evite las implicaciones de compresión, codificación y empaque; en otras palabras, dé por supuesto que cada carácter y cada dígito almacenados consumirán un byte de almacenamiento. Observe que los caracteres de formateado (por ejemplo las comas, los guiones, las diagonales) casi nunca se almacenan en una base de datos. Esos caracteres de formateado son agregados por los programas de aplicación que accederán a la base de datos y presentarán la salida a los usuarios.
2. Para cada tabla, multiplique el *tamaño del registro* por el número de instancias de entidad que deben incluirse en la tabla. Se recomienda que el crecimiento sea considerado para un razonable lapso de tiempo (por ejemplo, tres años). Éste es el *tamaño de la tabla*.
3. Sume los *tamaños de la tabla*. Éste es el *tamaño de la base de datos*.
4. Opcionalmente, sume un margen con capacidad de holgura (por ejemplo, 10 por ciento) para considerar factores imprevistos o estimaciones inexactas arriba. Ésta es la *capacidad anticipada de la base de datos*.

### > Generación de estructuras de bases de datos

Las herramientas CASE con frecuencia son capaces de generar código SQL para la base de datos directamente de un esquema de base de datos apoyado en CASE. Este código puede exportarse al DBMS para su compilación. Aun una base de datos pequeña como el modelo de SoundStage puede requerir 50 páginas o más de código de lenguaje de definición de datos SQL para crear las tablas, los índices, las claves, los campos y los triggers. Es claro que la capacidad de una herramienta CASE para generar en forma automática código sintácticamente correcto es una ventaja enorme de productividad. Además, casi siempre resulta ser más fácil modificar el esquema de la base de datos y regenerar el código que mantener el código de manera directa. La figura 12.11 es una muestra del código de dos páginas generado por el *System Architect* del esquema de la base de datos de SoundStage.

```

/* SQL Product = SQL Server V7 */
CREATE DATABASE dbMemberServices
CREATE TABLE tblAgreement(
    colAgreementNo          integer NOT NULL,
    colExpireDate           smalldatetime NULL,
    colActiveDate           smalldatetime NOT NULL,
    colFulfillmentPeriod   tinyint NOT NULL,
    colCreditsRequired      tinyint NOT NULL)
go
ALTER TABLE tblAgreement ADD CONSTRAINT AGREEMENT_PK
    PRIMARY KEY (colAgreementNo)
go
CREATE TABLE tblAudioTitle(
    colArtist              character varying(40) NULL,
    colCategory            character(2) NOT NULL,
    colSubCategory         character(2) NOT NULL,
    colUnits               tinyint NOT NULL          DEFAULT 1,
    colMediaType           character(2) NOT NULL,
    colAdvisoryCode        character(1) NULL,
    colProductNo          integer NOT NULL)
go
sp_bindrule domAudioCategory, 'tblAudioTitle.colCategory'
go
sp_bindrule domAudioCategory, 'tblAudioTitle.colSubCategory'
go
sp_bindefault defOne, 'tblAudioTitle.colUnits'
go
sp_bindrule domPositive, 'tblAudioTitle.colUnits'
go
sp_bindrule domAudioMedia, 'tblAudioTitle.colMediaType'
go
sp_bindrule domContentAdvisory, 'tblAudioTitle.colAdvisoryCode'
go
ALTER TABLE tblAudioTitle ADD CONSTRAINT AUDIO_TITLE_PK
    PRIMARY KEY (colProductNo)
go
CREATE TABLE tblGameTitle(
    colManufacturer        character varying(30) NOT NULL,
    colCategory            character(2) NOT NULL,
    colSubCategory         binary(2) NULL,
    colPlatform            character(2) NOT NULL,
    colMedia               character(2) NOT NULL,
    colPlayers             tinyint NOT NULL,
    colAdvisoryCode        character(3) NULL,

```

**FIGURA 12.11** Código parcial SQL para construir la base de datos de SoundStage



```

        colProductNo                integer NOT NULL)
go
sp_bindrule domGameCategories, 'tblGameTitle.colCategory'
go
sp_bindrule domGameCategories, 'tblGameTitle.colSubCategory'
go
sp_bindrule domGamePlatforms, 'tblGameTitle.colPlatform'
go
sp_bindrule domGameMediaTypes, 'tblGameTitle.colMedia'
go
sp_bindefault defOne, 'tblGameTitle.colPlayers'
go

ALTER TABLE tblGameTitle ADD CONSTRAINT GAME_TITLE_PK
    PRIMARY KEY (colProductNo)
go

CREATE TABLE tblMember(
    colMemberNo                    integer NOT NULL,
    Last_Name                      character varying((20)) NOT NULL,
    First_Name                    character varying((19)) NOT NULL,
    Middle_Initial                character((1)) NOT NULL,
    colStatus                     character(1) NOT NULL,
    colStreetAddress              character varying(40) NOT NULL,
    colPostBox                    character(4) NULL,
    colCity                      character varying(30) NOT NULL,
    colState                     character(2) NOT NULL,
    colZipCode                   character(9) NOT NULL,
    Area_Code                    character((3)) NOT NULL,
    Phone_Number                 character((7)) NOT NULL,
    Extension                    character((5)) NOT NULL,
    colLastOrderDate             smalldatetime NULL,
    colBalance                   character(6) NOT NULL          DEFAULT 0.00,
    colCreditCardType            character(1) NULL,
    colCreditCardNo              character(16) NULL             UNIQUE,
    colCreditCardExp             smalldatetime NULL,
    colBonusBalance              character(3) NULL              DEFAULT 0,
    colAudioPref                 character(2) NOT NULL,
    colAudioMedia                character(2) NOT NULL,
    colDateEnrolled              smalldatetime NOT NULL,
    colExpirationDate            smalldatetime NULL,
    colGamePreference            character(2) NOT NULL,
    colGameMedia                 character(2) NOT NULL,
    colNoCreditsEarned           tinyint NOT NULL,
    colVideoPreference           character(2) NOT NULL,
    colVideoMedia                character(2) NOT NULL,
    colAgreementNo               integer NOT NULL,
    colPrivacyCode               character(2) NULL,
    colEmailAddress              character(40) NULL)
go

```

**FIGURA 12.11** Conclusión

## La siguiente generación: diseño de las bases de datos

La tecnología de bases de datos relacionales se despliega y se usa ampliamente en los desarrollos contemporáneos de los sistemas de información. Las habilidades enseñadas en este capítulo seguirán siendo viables en el futuro previsible. Sin embargo, una nueva tecnología que está surgiendo con lentitud podría cambiar en última instancia el panorama en forma dramática: los sistemas de administración de las bases de datos basados en *objetos*.

Los sistemas de administración de las bases de datos basados en objetos almacenan

objetos reales: es decir, datos encapsulados y todos los procesos que pueden actuar sobre esos datos. Debido a que los sistemas relacionales de administración de bases de datos se usan tan ampliamente, no esperamos que este cambio ocurra en forma rápida. Además, no es probable que los vendedores de DBMS relacionales abandonen su parte del mercado sin pelear. Se espera que estos vendedores incorporen la tecnología de objetos en sus DBMS relacionales existentes o que creen nuevos DBMS objetos y suministren la transición entre los modelos relacionales y los objetos. De cualquier manera, ésta es una tecnología que debe observarse.

El ejemplo SoundStage generó en realidad más de 30 páginas de código SQL para crear la base de datos de la figura 12.9. ¿Puede imaginar usted el esfuerzo requerido para escribir a mano esa cantidad de código SQL (con exactitud razonable)? Es claro que la generación de código SQL mediante la herramienta CASE puede ser muy productiva; sin embargo, el código generado es sólo tan bueno y completo como el modelo de datos.

¡Comencemos con lo obvio! Si usted aspira a una carrera en sistemas de información, lo mejor sería que planeara tomar uno o más cursos reales de bases de datos. Los temas presentados en este capítulo representan solamente la punta del iceberg en relación con la tecnología, el desarrollo y la administración de las bases de datos. La mayoría de los planes de estudio de IS incluyen cuando menos un buen curso de administración de datos o de bases de datos para agregar valor a su educación. ¡Tómelo!

Usted apenas ha iniciado su viaje a través del diseño de sistemas. La base de datos es el *cerebro* de un sistema o una aplicación nuevos. Los capítulos siguientes se enfocarán en el diseño de otras partes cruciales del cuerpo. Los capítulos 13 a 15 enseñan el diseño de entradas, salidas e interfaces del usuario, respectivamente. Considere a las entradas, las salidas y las interfaces como el *alma* del sistema.



1. Los datos captados por un sistema de información se guardan en archivos y bases de datos. Un archivo es una colección de registros similares. Una base de datos es una colección de archivos interrelacionados.
2. Muchos sistemas heredados se construyeron con tecnología de archivos. Como los archivos se construyeron para aplicaciones específicas, su diseño fue optimizado para esas aplicaciones. Esta relación cercana entre los archivos y sus aplicaciones dificultaron reestructurar los archivos para satisfacer requerimientos futuros. Y como muchas aplicaciones usan los mismos datos, no es raro encontrar archivos redundantes con valores de datos que no siempre coinciden.
3. A medida que dichos sistemas heredados se someten a una reingeniería, generalmente se convierten a la tecnología de bases de datos. Las bases de datos bien diseñadas comparten datos no redundantes y vencen todas las limitaciones de los archivos convencionales.
4. El diseño de la base de datos es el proceso de traducir los modelos lógicos de datos (capítulo 7) a los esquemas de bases de datos físicos.
5. La unidad más pequeña de dato significativo que puede almacenarse se llama campo. Hay cuatro tipos de campos:
  - a) Una clave primaria es un campo que identifica de manera única, uno y sólo un registro en un archivo o una tabla.
  - b) Una clave secundaria es un campo que puede identificar de manera única a uno y solamente un registro en un archivo o una tabla, o puede identificar a un conjunto de registros con alguna característica común y significativa.
  - c) Una clave foránea es un campo que señala un registro relacionado en una tabla diferente.
  - d) Todos los demás campos se llaman campos descriptivos.
6. Los campos se organizan en registros, y los registros similares son organizados en archivos o tablas.
7. Una base de datos es una colección de tablas (archivos) con punteros lógicos que relacionan los registros de una tabla con los registros de otra tabla.
8. La arquitectura de datos que ha evolucionado en la mayoría de las organizaciones incluye archivos convencionales, bases de datos operacionales, almacenes de datos, bases de datos personales y de grupo de trabajo. Para coordinar esta infraestructura complicada, muchas organizaciones asignan un administrador de datos para planificar y operar el recurso total de datos y designan a administradores de bases de datos para implantar y manejar bases de datos específicas y tecnologías de bases de datos.
9. Una arquitectura de bases de datos se construye alrededor de un sistema de administración de bases de datos (DBMS) que proporciona la tecnología para definir la estructura de la base de datos y luego crear, leer, actualizar y suprimir registros en las tablas que constituyen esa estructura. Un DBMS provee un lenguaje de datos para lograr estas acciones. Ese lenguaje provee al menos dos componentes:
  - a) Un lenguaje de definición de datos para crear y mantener la estructura de la base de datos y las reglas.
  - b) Un lenguaje de manipulación de datos para crear, leer, usar, actualizar y suprimir registros en la base de datos.
10. Actualmente, los sistemas de administración de base de datos relacionales se usan para soportar el desarrollo y la reingeniería de un número abrumador de sistemas de información. Las bases de datos relacionales almacenan datos en una colección de tablas que se relacionan vía claves foráneas.
  - a) La definición de datos y los lenguajes de manipulación de la mayoría de los DBMS relacionales se consolidan en un lenguaje estándar conocido como SQL.
  - b) Los sistemas de administración de bases de datos relacionales de alto nivel soportan triggers y procedimientos almacenados, programas que se almacenan con las tablas y que pueden llamarse de otros programas basados en SQL.
11. El análisis y la normalización de los datos son técnicas para remover impurezas de un modelo de datos como una introducción para el diseño de la base de datos. Estas impurezas pueden hacer que una base de datos sea poco confiable, inflexible y no escalable.
12. Deberán tomarse decisiones de distribución y de replicación antes del diseño de la base de datos. Cada base de datos única debería ser representada por su propio submodelo lógico de datos.
13. Un esquema de bases de datos es el modelo físico para una base de datos fundamentada en la tecnología escogida de bases de datos. Las reglas para transformar un modelo lógico de datos en un esquema físico de la base de datos se generalizan como sigue:
  - a) Cada entidad se convierte en una tabla.
  - b) Cada atributo se convierte en un campo (una columna en la tabla).
  - c) Cada clave primaria y secundaria se convierte en un índice en la tabla.
  - d) Cada clave foránea implanta una relación posible entre las instancias de la tabla.
14. La integridad de la base de datos deberá verificarse y, si es necesario, mejorarse para asegurar que el negocio y sus usuarios pueden confiar en los datos almacenados.
  - a) La integridad de las claves asegura que cada registro tenga un valor único, no NULO de la clave primaria.

- b) La integridad del dominio asegura que los campos correctos almacenen sólo valores legítimos del conjunto de todos los valores posibles.
- c) La integridad referencial asegura que ningún valor de clave foránea señale un valor inexistente de clave primaria. Deberá especificarse una regla de supresión para cada relación con otra tabla. Las reglas de supresión ejecutan en cascada la supresión con los registros relaciona-

dos en otras tablas, deniegan la supresión hasta que los registros relacionados en otras tablas se supriman primero, o permiten la supresión pero colocan cualesquiera claves foráneas en las tablas relacionadas con NULL.

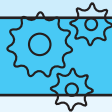
15. SQL-DDL es escrito o generado para crear la base de datos.

## Preguntas de repaso



1. ¿Qué representa el acrónimo CRUD?
2. Al observar la figura 12.1 en el libro de texto, ¿qué puede inferir usted respecto a las características de los archivos y las bases de datos convencionales?
3. ¿Por qué tienden los archivos convencionales a tener duplicación de datos?
4. ¿Por qué son los archivos convencionales fáciles de diseñar e implantar?
5. ¿Cuál es un concepto equivocado común acerca de las bases de datos?
6. ¿Por qué es más riesgoso almacenar datos en una base de datos que en un archivo?
7. ¿Qué es una clave secundaria?
8. ¿Qué es una estructura de registro de longitud fija?
9. ¿Cuáles son algunos tipos comunes de tablas y archivos convencionales?
10. Al comparar las bases de datos operacionales y los almacenes de datos, ¿cuál generalmente tiene menos actividades de CRUD? ¿Por qué?
11. ¿Qué es un motor de base de datos?
12. ¿Qué son los metadatos? Si los administradores de bases de datos necesitan definir los metadatos, ¿qué tipo de lenguaje deberán usar (DDL o DML)? ¿Por qué?
13. ¿Qué es una base de datos relacional?
14. ¿Qué diferencia hay entre un esquema de bases de datos relacionales y un esquema de bases de datos?
15. ¿Cuáles son las reglas de supresión comunes para implantar la integridad referencial?

## Problemas y ejercicios



1. Cuando un organismo se decide a reemplazar un sistema heredado, usualmente escoge un sistema contemporáneo de bases de datos en vez de un sistema tradicional basado en archivos. Pero cada tipo de sistema tiene sus propias ventajas y desventajas. Identifique si cada característica listada abajo forma parte de un sistema basado en archivos o de un sistema de base de datos.
  - El alto precio de desarrollo.
  - Generalmente diseñado para ser usado con una aplicación o sistema individual.
  - Mayores preocupaciones de privacidad de datos.
  - Redundancia controlada.
  - Desempeño subóptimo para el uso compartido por los sistemas múltiples.
  - Tiende a ser más lento.
  - Los formatos de datos son flexibles.
  - La identificación de elementos de datos es relativamente rápida y directa.
  - Diseñado para soportar requerimientos actuales.
  - Costos de capacitación más altos.
  - Los registros se enlazan con registros relacionados.
  - Estándares rigurosos de diseño.
  - Velocidad de procesamiento optimizada para una aplicación individual.
  - Tendencia hacia la redundancia de datos.
  - Mayor vulnerabilidad.
  - El almacenamiento de datos se construye alrededor del centro del sistema de información.
  - El efecto “silo”.
  - Escalable.
2. Aunque los sistemas de bases de datos se han convertido en los sistemas preferidos para los sistemas nuevos y los que son sometidos a la reingeniería, ¿existen cualesquiera situaciones donde un sistema tradicional basado en archivos podría escogerse en su lugar? Explique su respuesta.
3. El libro de texto afirma que “los datos son un recurso que debe controlarse y administrarse”. Explique esta declaración e indique si está de acuerdo o no y por qué.
4. Considere una concesión para distribución local de coches que ha estado en el negocio durante quince años más o menos. Además de realizar ventas y arrendar autos nuevos y usados, la concesión tiene un departamento de refacciones, un departamento de servicio y un departamento de hojalatería. Fue una de las primeras concesiones en automatizarse a mediados de los años setenta, y mientras la mayor parte de los sistemas de información se han actualizado periódicamente, unos cuantos de los sistemas

originales todavía están en uso, incluyendo algunos sistemas autónomos. Unos cinco años atrás, la concesión para distribución comenzó a reemplazar gradualmente algunos de los sistemas tradicionales basados en archivos con un sistema de base de datos relacional, pero la conversión está lejos de terminarse. El año pasado, el dueño leyó en una revista de aerolíneas acerca de los almacenes de datos (data warehouses), y contrató a un integrador local de sistemas para instalar uno, el cual está en curso. La concesión para distribución tiene un par de personas de tecnología de la información milusos en el equipo, sin embargo, la mayor parte del trabajo de desarrollo se subcontrata. Dado este panorama, dibuje un diagrama de alto nivel de cómo podría verse la arquitectura de datos de la concesión para distribución; use la figura 12.2 como ejemplo.

5. Uno de los sistemas autónomos heredados de la concesión para distribución en el ejercicio precedente es el sistema de plan de trabajo del vendedor. Este sistema fue desarrollado en la década de 1980 en una sola PC, usando *dBASE III*, para rastrear el plan de trabajo diario y semanal de cada vendedor. La concesión para distribución planea retirar este sistema anticuado e incorporarlo en un sistema de base de datos relacional. ¿Cuáles son dos o tres tablas que usted podría esperar encontrar en un sistema de planificación? Usando la figura 12.5 como un ejemplo, muestre estas tablas en un esquema simple de base de datos física. Incluya la clave primaria, cualesquiera claves foráneas, uno o dos atributos sin clave y algunos valores para cada objeto.
6. Coteje los siguientes términos en la primera columna con las definiciones o los ejemplos en la segunda columna.

- |                                  |  |
|----------------------------------|--|
| 1. DBMS                          | a) Implantación física de una base de datos      |
| 2. Archivo de transacciones      | b) Registro federal de códigos del país          |
| 3. Almacén de datos              | c) 5NF   |
| 4. Clave primaria                | d) Microsoft <i>Access</i>                       |
| 5. SQL                           | e) Suprimir:establecer el valor nulo             |
| 6. Herramienta CASE              | f) Sybase <i>IQ</i>                              |
| 7. Notación húngara              | g) SSN   |
| 8. Normalización                 | b) Archivo diario de admisiones del hospital     |
| 9. Archivo de consulta de tablas | i) <i>Oracle 10g</i>                             |
| 10. Integridad referencial       | j) Convención estándar de denominación de tablas |
| 11. Esquema                      | k) <i>System Architect</i>                       |
| 12. Base de datos personal       | l) ALTERAR TABLA                                 |

7. Usted es un diseñador de sistemas y tiene a un amigo que posee una librería pequeña y un negocio de ventas por correo especializado en libros raros y primeras ediciones. Las ventas totales promedian cerca de una docena de libros al día,

combinando la tienda y las órdenes por correo. Su amigo quiere comenzar a vender libros por la Red también, y ha acudido a usted para un consejo. Le han dicho que debería escoger una base de datos relacional y que *Oracle* es la mejor que puede obtener. ¿Concuerda usted?

En caso de que no, ¿qué recomendaría usted?

8. Al transformar un modelo lógico de datos en un esquema físico de base de datos relacional, qué opciones tiene el administrador de la base de datos para la implantación de las entidades de supertipo/subtipo?
9. Usted necesita calcular la capacidad anticipada de la base de datos para construir una base de datos con cuatro tablas, como se muestra enseguida:

Tabla 1	Tabla 2	Tabla 3	Tabla 4
Campo 1: 32 caracteres	Campo 1: 5 caracteres	Campo 1: 2 caracteres	Campo 1: 16 caracteres
Campo 2: 15	Campo 2: 7	Campo 2: 7	Campo 2: 30
Campo 3: 7	Campo 3: 13	Campo 3: 8	Campo 3: 12
Campo 4: 12	Campo 4: 6	Campo 4: 4	Campo 4: 7
Campo 5: 9	Campo 5: 4		Campo 5: 54
Campo 6: 12			Campo 6: 3
Campo 7: 6			Campo 7: 1
Campo 8: 2			

La tabla 1 estará cargada inicialmente con 200 000 registros, la tabla 2 con 100 000 registros y las otras dos tablas con 40 000 registros cada una. La tasa prevista de crecimiento en el número de registros es de 20 por ciento cada año para tres años.

Según los pasos de planificación de capacidad de la base de datos descrita en el capítulo, ¿cuál es la capacidad anticipada de la base de datos?

10. Usted estudia la documentación de diseño para un sistema de información muy grande usado por su organización. Como lo esperaba, todas las entidades en el modelo lógico de datos están en tercera forma normal. Pero al comparar el modelo lógico de datos con el esquema físico de la base de datos, usted observa que dos tablas mostradas separadamente en el modelo lógico de datos se han combinado en una sola tabla en el esquema físico de la base de datos. No hay notas que expliquen esto, y usted no le puede preguntar al administrador de la base de datos que firmó los documentos del diseño pero que ya no está con la compañía. ¿Cuál podría haber sido la razón para adoptar la tercera forma normal? ¿Cuáles son las consecuencias potenciales?
11. Explique el concepto de integridad referencial y dé un ejemplo. ¿Qué es un error de integridad referencial? Ponga un ejemplo y explique las consecuencias posibles de un error de este tipo.
12. Las reglas de supresión para implantar la integridad referencial incluyen suprimir: en cascada y suprimir:restringir. En general, ¿qué criterios deberá usar

un DBA para decidir si usa la regla suprimir:en cascada o suprimir:restringir para la supresión?

13. La centralización de la base de datos es una de las varias opciones de distribución, pero viola la regla de que los datos deben manejarse y almacenarse cerca de los usuarios. Discuta las razones por las

cuales muchos administradores de datos y usuarios consideran que esta regla es importante. Dado el crecimiento de aplicaciones basadas en la Red y los sistemas de información globales, ¿todavía es esta regla importante y/o viable?

## Proyectos e investigación



1. Entreviste a los funcionarios jefes de información o a los gerentes de tecnología de la información de varias organizaciones que usan tecnologías de bases de datos relacionales. Pregúnteles acerca de su experiencia en moverse de sistemas de información tradicionales de archivos simples a los sistemas de bases de datos relacionales.

- Describa los CIOs que usted entrevistó y sus organizaciones.
- Describa su ambiente de base de datos.
- ¿Cuándo se movieron sus organizaciones de los sistemas de archivo tradicionales a los sistemas de base de datos relacional?
- ¿Qué tipo de problemas encontraron y cómo fueron resueltos (o si se resolvieron)?

Si ellos tuvieran que hacerlo nuevamente, ¿qué harían en forma diferente?

2. La mayoría de las veces, los ambientes de bases de datos en una organización reflejan estructuras de datos que han sido desarrolladas durante varios años, algunas veces fortuitamente y que a menudo reflejan una variedad de estructuras y estilos arquitectónicos. Observe el ambiente existente de la base de datos en su organización o la escuela o en una compañía local.

- ¿Cuál es la antigüedad del sistema de información más viejo usado en la organización?
- ¿Cuál es la antigüedad del más nuevo?
- ¿Tiene la organización tanto sistemas basados en archivos tradicionales como sistemas de bases de datos relacionales modernos?
- ¿Cuáles son los diferentes sistemas del usuario, las herramientas de usuarios finales, las aplicaciones de usuarios finales? ¿Hay almacenes de datos (data warehouses)? ¿Hay aplicaciones basadas en la Red?
- Dibuje un diagrama de arquitectura de datos basado en el diagrama mostrado en la figura 12.2.

3. Las bases de datos modernas requieren un nivel alto de habilidad y conocimiento para estar adecuadamente soportadas. Vea si usted puede entrevistar a los administradores de bases de datos en tres o cuatro organizaciones locales que usan sistemas contemporáneos de bases de datos:

- Describa los sistemas que están bajo la dirección de los administradores de bases de datos que entrevistó.
- En término medio, ¿qué tanta experiencia y capacitación tenían en la tecnología de información antes de convertirse en administradores adecuados de bases de datos?
- En término medio, ¿cuánto tiempo dedican al año a la capacitación, formal o informal, conservando sus habilidades al día? ¿Sienten que reciben suficiente capacitación?
- Calcule el costo promedio de capacitación anual por administrador de bases de datos. Incluya tanto los costos directos por capacitación como los costos indirectos de "oportunidad perdida". Si los costos reales no se conocen, use un costo directo de capacitación de \$500/día, el sueldo promedio del administrador de base de datos y los beneficios de \$75 por hora.
- ¿Piensa usted que los costos de capacitación para los administradores de bases de datos son más altos que para los administradores de sistemas de archivos simples? ¿Por qué sí o por qué no?

4. Después de que usted hable con los administradores de la base de datos con respecto a la capacitación, también pregúnteles acerca de:

- Normalización: ¿normalizan generalmente a la tercera forma normal? ¿o Más alta?
- En general, ¿cuánto tiempo dedican a la normalización para una base de datos grande o de nivel empresarial?
- ¿Cuáles son los tres problemas crónicos más grandes afrontados por cada uno de los administradores de la base de datos?
- ¿Cada cuánto tienen que modificar y/o actualizar el esquema de la base de datos? ¿Tienen un proceso formal para identificar y hacer actualizaciones o se hacen sobre una base *ad hoc*?
- Si ellos fueran el CIO en su organización, ¿qué cambiarían?

5. Las herramientas CASE, tal como el *System Architect*, sirven para el desarrollo y soporte de las bases de datos. Busque en la Red y en publicaciones de negocios acerca de algunas de las herramientas CASE populares actualmente en uso.

- a) ¿Qué herramientas CASE encontró usted y quiénes son sus fabricantes?
  - b) ¿Cuál es el número de bases instaladas, o desarrollos de IT, que usan cada una de estas herramientas CASE?
  - c) ¿Cuál es el rango de costos para estas herramientas CASE? ¿Piensa usted que son eficientes con base en costos?
  - d) Compare y contraste estas herramientas CASE en términos de sus características y sus capacidades.
  - e) ¿Cuál usaría usted si fuese un administrador de bases de datos y el costo no fuera una preocupación? ¿Por qué?
6. Actualmente, la tecnología de bases de datos relacionales es quizá la tecnología de bases de datos más prevaleciente que se usa en los desarrollos modernos de tecnología de la información. Pero la tecnología de las bases de datos es un campo que evoluciona y tecnologías nuevas se están desarro-

llando constantemente. Busque en la Red o en su biblioteca de la escuela artículos sobre tecnologías emergentes de bases de datos, tales como los sistemas de administración de bases de datos basados en objetos. Asegúrese de incluir reportes para el cliente (white papers) de compañías como Oracle, Sybase, IBM, y Microsoft.

- a) ¿Qué artículos y ponencias encontró usted?
- b) ¿Cuáles son algunas de las nuevas tecnologías de bases de datos que entran en el mercado o que están actualmente bajo desarrollo?
- c) Compare y contraste cada una de ellas con la tecnología de bases de datos relacionales contemporánea.
- d) ¿Qué influencia tiene el crecimiento de aplicaciones de Internet en estas tecnologías de las bases de datos?
- e) ¿Cuál siente usted que podría ser un serio contendiente para reemplazar la tecnología de bases de datos relacional? ¿Por qué?



## Casos breves

Con ayuda de su profesor, coordínese con un equipo para una clase de análisis de sistemas o de programación de computadoras en otra escuela. Su asignación, a completar conjuntamente, es construir una página adecuada de Red para un negocio pequeño o una organización no lucrativa de su elección. Usted será calificado en función a la totalidad, funcionalidad, profesionalismo y trabajo de equipo. Una sugerencia de comunicación es utilizar correo electrónico lo más posible.

1. Encuentre al equipo en la otra escuela por vía telefónica, en un ambiente de reuniones virtuales, mesa redonda o correo electrónico de acuerdo con las instrucciones de su profesor. Si usted usa un ambiente de reuniones virtuales, puede necesitar instalar y aprender cómo usar el software correcto. Determine y establezca lineamientos de equipo y reglas para:
  - Fechas límite. ¿Cómo manejará el equipo una fecha tope vencida de uno de los miembros del equipo? ¿Quién se encargará de colocar la línea de tiempo?
  - La comunicación. ¿Cómo se comunicará usted? ¿Cada cuándo? ¿Se comunican mejor algunos de sus miembros utilizando un método en vez de los demás (es decir, las preferencias)?
  - Malos entendidos de diferencias personales. ¿Cómo se ocupará usted de los malos entendidos entre los miembros o las diferencias personales que surjan?
  - Las expectativas. ¿Cuáles son las expectativas de calidad del equipo? ¿El comportamiento? ¿Qué hará usted si alguien no alcanza el estándar?

Proponga a su profesor a crear un acuerdo firmado por cada uno de los miembros del equipo, concierne a cómo se enfrentarán estos asuntos.

2. Reúnanse con los dueños de negocios pequeños locales o con los representantes de organizaciones sin fines de lucro. Encuentren una compañía u organización que patrocine a su equipo para producir un sitio Web para ellos (por supuesto que sin lucro). Investiguen con el administrador de riesgos de su escuela o con el departamento jurídico qué documentación es necesaria que llenen ustedes y su "cliente". (¿Por qué es esto necesario?)
3. Determine los requerimientos del negocio o de la organización a través de las entrevistas, formas, encuestas, JAD y cosas por el estilo y cree los estudios y modelos apropiados para el sitio Web. No olvide considerar costos, aspectos legales y necesidades específicas de la compañía en sus modelos y en el ensayo. Usted será calificado en completud, exactitud, claridad y profesionalismo.
4. Cree el sitio Web usando tecnologías apropiadas, obteniendo un nombre de dominio y así sucesivamente. Establezca también un correo electrónico con el nombre de dominio. Si es apropiado para que la compañía tenga capacidades de carrito de compras y de pago en línea, asegúrese de que éstos sean completamente funcionales. Someta a una prueba de tensión nerviosa al sitio intercambiando URLs con otro equipo *antes* de que usted lo entregue a su profesor o al cliente.

## Ejercicios de equipo e individuales



1. Cree un crucigrama usando términos y conceptos que haya aprendido en su clase hasta ahora. Luego, intercambie los crucigramas con un compañero de clase. Cada uno de ustedes deberá completar el crucigrama del otro.
2. Individual: En los casos breves de este capítulo, usted completó un proyecto de sitio Web con un equipo de otra escuela. Haga comentarios sobre su experiencia en un ensayo pequeño para ser enviado a su profesor.
3. Busque en su directorio local de la sección amarilla. ¿Quiénes son los principales anunciantes en “análisis de diseño y de sistemas” en su área? Contacte al menos a tres de ellos y averigüe la clase de servicios que proporcionan, su pericia/experiencia y cuánto cobran. ¿Qué significa esta información para usted y su carrera? Traiga sus conclusiones a la clase y compártalas.

## Lecturas recomendadas



- Bruce, Thomas. *Designing Quality Databases with IDEF1X Information Models*. Nueva York: Dorset House Publishing, 1992. Éste se ha convertido rápidamente en nuestro libro favorito de diseño práctico de bases de datos. A propósito, el prólogo fue escrito por John Zachman, cuyo *Framework for Information Systems Architecture* inspiró nuestro propio marco de bloques de construcción del sistema de información.
- McFadden, Fred; Jeffrey Hoffer; y Mary Prescott. *Modern Database Management*, 5a. ed. Reading, MA: Addison Wesley, 1994. Para aquellos que buscan ampliar su educación general sobre bases de datos y administración de datos, éste es

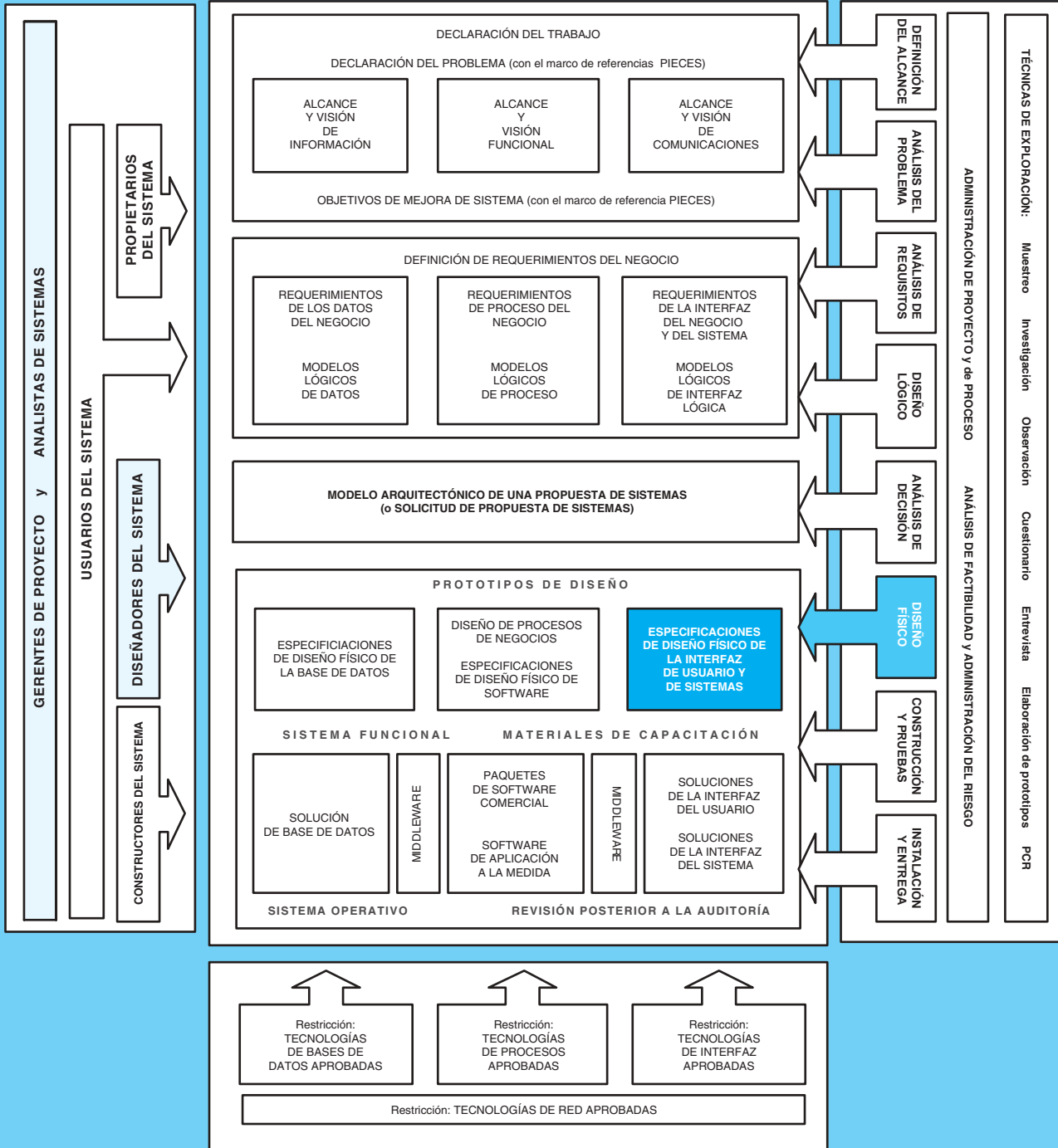
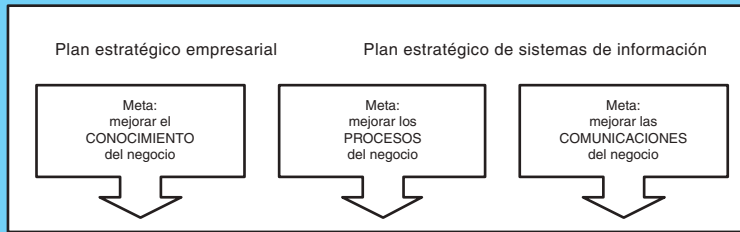
uno de los libros de texto introductorios más populares en el mercado y uno de nuestros favoritos. Estos autores hacen un trabajo especialmente minucioso al explicar el diseño de las bases de datos distribuidas (con un detalle mucho mayor de lo que es posible en nuestro libro).

- Teorey, Toby. *Database Modeling & Design: The Fundamental Principles*, 2a. ed. San Francisco: Morgan Kaufman Publishers, 1990. Éste es nuestro libro conceptual favorito de diseño de bases de datos. El Apéndice A suministra una revisión concisa del lenguaje SQL.









# 13

## Diseño de salidas y elaboración de prototipos

### Panorámica y objetivos del capítulo

En este capítulo usted aprenderá a diseñar y construir prototipos para las salidas de computadora. Sabrá cómo diseñar y construir prototipos de las salidas cuando usted pueda:

- Distinguir entre salidas internas, externas y con regreso.
- Diferenciar entre los reportes detallados, resumidos y de excepción.
- Identificar varios métodos de implantación de salidas.
- Diferenciar entre formatos tabulados, por zonas y gráficos para presentar la información.
- Distinguir entre gráficos de área, barras, de columnas, circulares, lineales, de radar, de dona, y de dispersión y sus usos.
- Describir varios principios generales que son importantes para el diseño de las salidas.
- Diseñar y construir prototipos de las salidas de computadora.

El diseño de salidas y entradas se parece un poco al problema de la secuencia del “huevo o la gallina”. ¿Qué es lo que usted debe hacer primero? En esta edición, presentamos primero el diseño de las salidas. El diseño clásico de sistemas prefiere este enfoque como parte de una prueba de validación del sistema: diseñe las salidas y luego asegúrese de que las entradas son suficientes para producir las salidas. En la práctica, esta secuencia de tareas se hace menos importante porque las técnicas del análisis de sistemas modernas predefinen sobradamente los requerimientos lógicos de las entradas y las salidas. Usted y su profesor pueden intercambiar sin problema el orden de los capítulos 13 y 14 si así lo prefieren.

## Introducción

Bob Martínez se alegró de que como parte del paso del diseño de datos él creó una base de datos del prototipo en la Microsoft Data Engine (MSDE). Realmente viene bien ahora que él diseña los reportes del sistema.

Bob creó una simple base de datos Microsoft *Access*, la conectó a la base de datos MSDE, e ingresó algunos datos de muestra. Luego, al trabajar con los prototipos de identificación creados en la fase de análisis y las narrativas de caso práctico, él usó el *Access Report Wizards* para crear un primer prototipo para cada reporte impreso. Con la retroalimentación de los usuarios, él refinó el diseño de los reportes a través de varias iteraciones. Para los reportes que requerían parámetros a la medida introducidos por el usuario, Bob creó formas de *Access* para simular la interfaz a la medida.

Por supuesto, el sistema real no usará ni Microsoft *Access* ni MSDE. Pero cuando finalmente los programadores se incorporen en la construcción del sistema, estos reportes y formas guiarán su trabajo y asegurarán que el sistema satisfaga todos los requerimientos del usuario.

## Conceptos y lineamientos del diseño de las salidas

Las salidas presentan la información a los usuarios del sistema; son el componente más visible de un sistema de información en funcionamiento. Como tal, son a menudo la base para la evaluación final de los usuarios y de la administración del valor del sistema. Durante el análisis de requerimientos, usted definió requerimientos *lógicos* de salida. Durante el análisis de decisiones, usted *pudo* haber considerado diferentes alternativas físicas de implantación. En este capítulo, usted aprenderá a diseñar las salidas físicas.

En la actualidad, la mayoría de las salidas son diseñadas a través de prototipos de construcción rápida. Éstos pueden ser simples imitaciones generadas por computadora con datos inventados, o ser generadas del prototipo de bases de datos como Microsoft *Access*, que pueden construirse rápidamente y llenarse con datos de prueba. Estos prototipos rara vez son totalmente funcionales. No contendrán características de seguridad ni acceso optimizado a los datos que serán necesarios en la versión final de un sistema. Además, en interés de la productividad, no podremos incluir cada característica del botón o de control que tendría que estar en un sistema de producción.

Durante el análisis de requerimientos, las salidas se modelaron como flujos de datos que consisten de atributos de datos. Pero aun en los análisis de requerimientos más completos, perderemos requerimientos. El diseño de las salidas puede introducir campos o atributos nuevos al sistema.

Comenzamos con un estudio de los tipos de salidas. Las salidas pueden clasificarse según dos características: 1) su distribución y audiencia, y 2) su método de implantación. La figura 13.1 ilustra esta taxonomía. Las características se discuten brevemente en las siguientes secciones.

### > La distribución y la audiencia de salidas

Una forma de clasificar las salidas es de acuerdo con su distribución dentro o fuera de la organización y las personas que las leen y usan. Las **salidas internas** están destinadas a los dueños del sistema y a los usuarios de sistema dentro de una organización; raramente salen fuera de la organización. Las salidas internas soportan ya sea operaciones de negocios cotidianas o el monitoreo de la gerencia y la toma de decisiones. La figura 13.2 ilustra tres subclases básicas de salidas internas:

- Los **reportes detallados** presentan información con poco o ningún filtro o restricciones. El ejemplo en la figura 13.2a) es un listado de todas las órdenes de compra que se generaron en una fecha particular. Otros ejemplos de reportes de detalle serían un listado detallado de todas las cuentas del cliente, las órdenes, o los productos en el inventario. Algunos reportes detallados son históricos, otros son obligatorios, es decir, requeridos por el gobierno.

**salida interna** Salida para los dueños y usuarios del sistema dentro de una organización.

**reporte detallado** Salida interna que presenta información con poco o ningún filtrado.

**FIGURA 13.1** Una taxonomía de las salidas generadas por computadora

<b>Distribución Entrega</b>	<b>Salida interna (reportes)</b>	<b>Salida con regreso (externa; luego interna)</b>	<b>Salida externa (transacciones)</b>
Impresora	<p>Información detallada, resumida, o de excepción impresa en reportes de copia dura solamente para uso de negocios internos.</p> <p>Ejemplos comunes: reportes de administración</p>	<p>Transacciones de negocios impresas en formas de negocios que finalmente se devolverán como entrada de transacciones de negocios.</p> <p>Ejemplos comunes: recibos de teléfono y comprobantes de tarjetas de crédito</p>	<p>Transacciones de negocios impresas en formas de negocios que concluyen a las transacciones de negocios.</p> <p>Ejemplos comunes: cheques de pago y estados de cuenta bancarios</p>
Pantalla	<p>Información detallada, resumida, o de excepción exhibida en monitores para uso de negocios internos.</p> <p>Los reportes pueden ser tabulares o gráficos</p> <p>Ejemplos: reportes de administración en línea y respuestas a las averiguaciones</p>	<p>Transacciones de negocios exhibidas en monitores en formas o ventanas que también se usarán para ingresar otros datos para iniciar una transacción relacionada.</p> <p>Ejemplos: Exhibición basada en la Red de precios de lista con la opción de compra de apuntar y hacer clic</p>	<p>Transacciones de negocios exhibidas en formas de negocios que concluyen las transacciones de negocios</p> <p>Ejemplos: Reporte basado en la Red que detalla las transacciones bancarias</p>
Terminales de punto de venta	<p>Información impresa o exhibida en terminales de propósito especial reservadas a funciones específicas de negocios internos. Incluye la transmisión de información por comunicación inalámbrica.</p> <p>Ejemplos: reporte de equilibrio del registro del efectivo de fin de turno</p>	<p>Información impresa o exhibida en una terminal de propósito especial con el propósito de iniciar una transacción de seguimiento de negocios.</p> <p>Ejemplos: monitor de tienda de abarrotes que permite al cliente monitorear los precios escaneados, para ser seguido por el ingreso de autorización de pagos de tarjetas de débito o de crédito</p>	<p>Información impresa o exhibida en terminales de propósito especial reservadas a los clientes.</p> <p>Ejemplos: Los saldos de las cuentas se exhiben en una máquina de cajero automático o en una impresión de billetes de lotería; también, se exhibe información de la cuenta vía televisión por cable o satélite</p>
Multimedia (audio o video)	<p>Información transformada en discurso para los usuarios internos.</p> <p>Comúnmente no se implanta para los usuarios internos.</p>	<p>Información transformada en discurso para usuarios externos que responden con datos de ingreso de discurso o tono.</p> <p>Ejemplos: horario de clase con teléfonos de tonos al tacto como parte del sistema de inscripción en los cursos</p>	<p>Información transformada en discursos para usuarios externos.</p> <p>Ejemplos: avance de película para prospectos de compradores en línea de DVD o respuesta telefónica a indagaciones de liquidaciones de hipoteca</p>
Correo electrónico	<p>Mensajes exhibidos relacionados con la información interna de los negocios.</p> <p>Ejemplos: mensajes de correo electrónico que anuncian la disponibilidad de un nuevo reporte de negocios en línea</p>	<p>Mensajes exhibidos cuyo objetivo es iniciar la transacción de negocios.</p> <p>Ejemplos: mensajes de correo electrónico cuyas respuestas se requieren para continuar el procesamiento de una transacción de negocios</p>	<p>Mensajes relacionados con las transacciones de negocios.</p> <p>Ejemplos: confirmaciones de mensajes en el correo electrónico de las transacciones de negocios realizadas vía correo electrónico comercial en la Red</p>
Hipervínculos	<p>Enlaces basados en la Red con información interna que se activa vía los formatos HTML o XML.</p> <p>Ejemplos: integración de todos los reportes del sistema de información en un sistema de archivos basado en la Red para archivado y acceso en línea</p>	<p>Enlaces basados en la Red incorporados en las páginas de ingreso basadas en la Red para proveer a los usuarios con acceso a información adicional.</p> <p>Ejemplos: en una página de subasta de la Red, hipervínculos con el historial de desempeño de un vendedor con una invitación para añadir un nuevo comentario</p>	<p>Enlaces basados en la Red incorporados en las transacciones basadas en la Red.</p> <p>Ejemplos: hipervínculos con la política de privacidad o una explicación de cómo interpretar o responder a la información en un reporte o una transacción</p>
Microficha	<p>Reportes de administración interna archivados en microfilme que requieren espacio mínimo de almacenaje físico.</p> <p>Ejemplos: salida de computadora en microfilme (output on microfilm, COM)</p>	<p>No es aplicable a menos que haya una necesidad interna de archivar documentos con respuesta.</p> <p>Ejemplos: salida de computadora en microfilme (COM)</p>	<p>No es aplicable a menos que haya una necesidad interna de copias de reportes externos.</p> <p>Ejemplos: salida de computadora en microfilme (COM)</p>

**FIGURA**  
13.2a)  
y 13.2b)

Niveles de detalle del  
reporte

a) Reportes detallados

SoundStage Entertainment Club Detailed

**PRODUCTS ORDERED ON 1/25/2000**

P.O. Number	Product Number	Product Type	Quantity In Stock	Quantity On Order
112312	102774	Merchandise	232	43
	232322	Title	23	43
	232332	Title	2	3
121212	222332	Merchandise	115	132
	546666	Title	667	1
	232554	Title	11,234	343
	200992	Title	54,321	1
232323	1212343	Title	1,324	11
	3434434	Merchandise	6,561	55
	4343434	Merchandise	112	111
	3434344	Title	3	232

Return to Summary      Close

b) Reportes resumidos

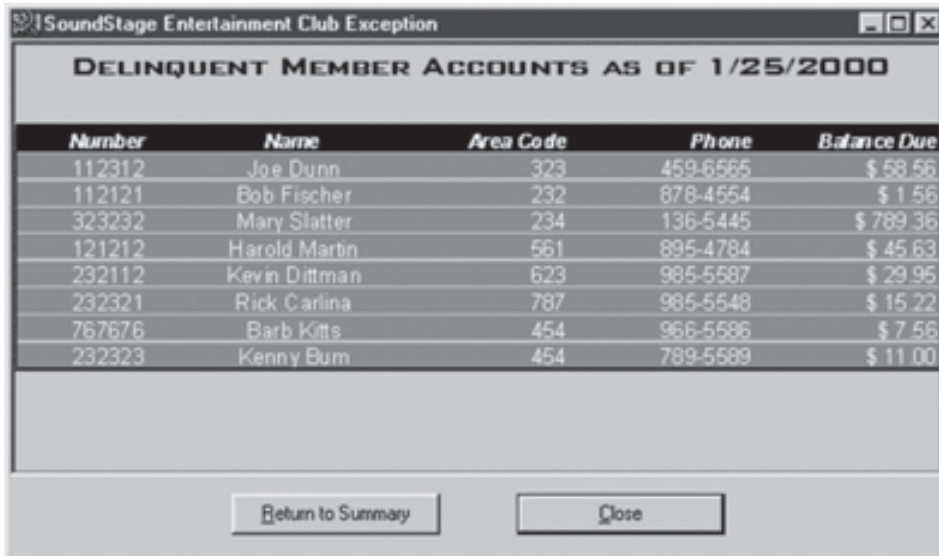
SoundStage Entertainment Club Summary

**PRODUCT SALES SUMMARY**  
**AS OF 1/25/2000**

Product Type	Product Category	Current Month's Unit Sales	Current Year Unit Sales
Merchandise	Clothing	784	4,312
	Media Accessory	541	2,079
	<b>Total:</b>	<b>1,325</b>	<b>6,391</b>
Title	Audio	667	20,439
	Game Title	11,234	12,445
	Video Title	54,321	998,872
	<b>Total:</b>	<b>66,222</b>	<b>1,031,756</b>

View Additional Reports      Close

## c) Reportes de excepción



Number	Name	Area Code	Phone	Balance Due
112312	Joe Dunn	323	459-6585	\$ 58.56
112121	Bob Fischer	232	878-4554	\$ 1.56
323232	Mary Slatter	234	136-5445	\$ 789.36
121212	Harold Martin	561	895-4784	\$ 45.63
232112	Kevin Dittman	623	985-5587	\$ 29.95
232321	Rick Carlina	787	985-5548	\$ 15.22
767676	Barb Kitts	454	966-5586	\$ 7.56
232323	Kenny Bum	454	789-5589	\$ 11.00

**FIGURA  
13.2c)**

Niveles de detalle del  
reporte

- Los **reportes resumidos** condensan la información para las personas que no quieren ahondar en los detalles. El reporte de muestra de la figura 13.2b) resume las ventas totales del mes y el año por tipo de producto y categoría. Los datos de los reportes resumidos típicamente sintetizan y resumen para indicar tendencias y problemas potenciales. El uso de gráficos (los diagramas y las gráficas) en los reportes resumidos también está ganando aceptación rápidamente porque resumen tendencias a simple vista de manera más clara.
- Los **reportes de excepción** filtran los datos antes de presentarlos al gerente como información. Estos reportes de excepción incluyen sólo excepciones a alguna condición o estándar. El ejemplo en la figura 13.2c) bosqueja la identificación de cuentas de miembros infractores. Otro ejemplo clásico de un reporte de excepción es un reporte que identifica artículos de baja existencia.

Lo contrario de las salidas internas son las salidas externas. Las **salidas externas** salen de la organización. Están destinadas a los clientes, proveedores, socios, y organismos reguladores. Generalmente concluyen o reportan sobre las transacciones de negocios. Ejemplos de salidas externas son facturas, estados de cuenta, cheques de salario, horarios de curso, boletos de avión, pases de abordar, itinerarios de viaje, recibos de teléfono, órdenes de compra, y etiquetas postales.

La figura 13.3 ilustra una salida externa de muestra para SoundStage Entertainment Club. Esta muestra, como muchas salidas externas, es inicialmente creada como una forma en blanco preimpresa, que es diseñada y duplicada por los fabricantes de formatos para usarse con impresoras de computadora.

Algunas salidas son tanto externas e internas. Comienzan como salidas externas que salen de la organización pero que al final regresan (en parte o enteramente) como una entrada interna. Las **salidas con regreso** son aquellas salidas externas que finalmente reingresan al sistema como entradas. La figura 13.4 muestra un documento con respuesta. Observe que la factura tiene porciones superior e inferior. La porción superior debe desprenderse y devolverse con el pago del cliente como una entrada.

## > Los métodos de implantación de las salidas

Suponemos que usted está familiarizado con los diferentes dispositivos de salida, tales como impresoras, graficadoras (plotters), salida de computadora en microfilm (computer output on microfilm, COM), y los monitores de la PC. Éstos son temas estándar en la mayoría de cursos introductorios de sistemas de información. En este capítulo, estamos más interesados con el producto real que con el dispositivo. Un buen analista de sistemas considerará todas las opciones disponibles para implantar una salida. Examinemos brevemente los métodos de implantación y los formatos. Usted deberá seguir refiriéndose a la figura 13.1 a medida que completamos esta introducción a la taxonomía de las salidas.

**reporte resumido** Salida interna que condensa la información para los gerentes.


**reporte de excepción** Salida interna que filtra los datos para presentar información que reporta las excepciones de alguna condición o estándar.

**salida externa** Salida que deja a la organización.

**salida con regreso** Salida externa que puede reingresar al sistema como entrada.

**FIGURA 13.3**

Documento externo típico



**SoundStage Entertainment Club**  
Fax 317-494-5222

**ORDEN  
DE COMPRA**

**El siguiente número debe aparecer en toda la correspondencia relacionada, documentos de flete, y facturas:**  
**NÚMERO DE ORDEN DE COMPRA: 712812**

<b>A:</b> CBS Fox Video Distribution 26253 Rodeo DR Hollywood, CA	<b>Fletar a:</b> SoundStage Entertainment Club Shipping/Receiving Station Building A 2630 Darwin Drive Indianapolis, IN 45213
--	--

FECHA DE LA ORDEN DE COMPRA	SOLICITANTE	FLETE VÍA	PUNTO F.O.B.	CONDICIONES
5-3-03	LDB	UPS		Neto 300

CANTIDAD	DESCRIPCIÓN	PRECIO UNITARIO	TOTAL
20000	Guerra de las Galaxias: La amenaza fantasma (VHS)	15.99	319 800.00
3000	Guerra de las Galaxias: La amenaza fantasma (DVD Dolby Digital)	19.99	59 970.00
500	Guerra de las Galaxias: La amenaza fantasma (DVD DTS)	24.99	12 495.00
8000	Guerra de las Galaxias: La amenaza fantasma (Play Station II)	16.99	135 920.00
400	Guerra de las Galaxias: Banda sonora de La amenaza fantasma (CD)	16.99	6 796.00
600	Guerra de las Galaxias: Cartel de teatro de La amenaza fantasma	4.99	2 994.00

Subtotal	537 975.00
Impuesto	37 658.25
Total	575 633.25

1. Por favor mande dos copias de su factura.  
2. Ingrese esta orden de acuerdo con los precios, condiciones, método de entrega, y especificaciones listados arriba.  
3. Por favor notifiquenos inmediatamente si no puede fletar como se especifica.

*Madge Worthy*      5-4-03

---

Autorizado por      Fecha

**La salida impresa** El medio más común para salidas de la computadora es el papel: la salida impresa. Actualmente, el papel es el medio más barato que examinaremos. Aunque la oficina sin documentos ha sido predicha por largos años, eso aún no se ha hecho una realidad. Quizás hay una adicción psicológica al papel como un medio. En todo caso, la salida del papel estará con nosotros por mucho tiempo.

La salida impresa puede producirse en impresoras de matriz, pero progresivamente se imprime en impresoras de láser, las cuales se han hecho eficientes en costo. Las salidas internas comúnmente se imprimen en papel en blanco (llamado *papel comercial*). Las salidas externas y los documentos con respuesta (salidas con regreso) son impresos en *formas preimpresas*. El diseño de formas preimpresas (tales como cheques y formatos de impuestos) está predeterminado, y las formas en blanco se fabrican en serie. Las formas preimpresas se pasan por la impresora para añadir los datos variables de negocios (como su cheque de salario y el formato de impuestos).

Quizás el formato más común de salida impresa es el tabular. La **salida tabular** presenta la información como columnas de texto y de números. La mayor parte de los programas de computadora que usted ha escrito probablemente genera informes tabulares. Todas las muestras de reportes de detalle, de resumen y de excepción ilustrados anteriormente en el capítulo (figura 13.2) eran tabulares.

Una alternativa para la salida tabular es la **salida por zonas**, la cual coloca texto y números en las áreas designadas o los cuadros de la forma o pantalla. La salida por zonas

**salida tabular** Salida que presenta la información a manera de columnas de texto y números.

**salida por zonas** Salida que presenta texto y números en las áreas designadas de una forma o pantalla.





# SoundStage Entertainment Club

2630 Darwin Drive - Bldg B  
Indianapolis, IN 45213  
317 496 0998 fax 317 494 0999

Factura No. 301231

## FACTURA

**Ciente**

Nombre KATRINA SMITH  
Dirección 3019 DURAC DR  
Ciudad LITTLE ROCK Estado AR C.P. 42653  
Teléfono 502-430-4545

Fecha de vencimiento 2/24/03  
Orden número 346910

Monto del pago

Desprenda y regrese la parte superior con el pago

Cantidad	Descripción	Precio unitario	TOTAL
1	EL INFIERNO DE LAS ÁGUILAS SE CONGELA (DVD DD)	\$19.99	\$19.99
1	LA CAJA GRAMMY (CD) ***CUENTA COMO 3 CRÉDITOS	\$21.99	\$21.99
1	LO QUE EL VIENTO SE LLEVÓ CORTE DEL DIRECTOR (DVD DS)	\$17.99	\$17.99
1	SEXTO SENTIDO (VHS)	SS CR GRATIS	\$0.00
1	LA VIDA DE UN INSECTO (VHS)	SS CR GRATIS	\$0.00
1	NASCAR 2000 (VHS)***CLOSEOUT (SIN SS CR)	\$9.99	\$9.99
SE USARON 10 CRÉDITOS DE SOUNDSTAGE PARA PAGAR PARTE DE ESTA COMPRA			
APRECIAMOS LA MANERA EN QUE USTED HA PAGADO SU CUENTA. COMO RECONOCIMIENTO HEMOS AÑADIDO 7 CRÉDITOS DE SOUNDSTAGE A SU CUENTA			
USTED PUEDE GANAR 7 CRÉDITOS AL PAGAR ESTA FACTURA EN LA FECHA DE VENCIMIENTO			

**Detalles del pago**

Efectivo  
 Cheque  
 Tarjeta de crédito

Nombre \_\_\_\_\_  
 Número de tarjeta \_\_\_\_\_  
 de crédito \_\_\_\_\_ Expira \_\_\_\_\_

Subtotal	\$69.96
Flete y manejo	\$7.00
Impuestos	\$2.95
<b>TOTAL</b>	<b>\$79.91</b>

Sólo para uso de oficina

*Por favor devuelva la parte superior de la factura con el pago.  
Haga los cheques a nombre de: SoundStage Entertainment Club.*

**DEVUELVA LA PARTE SUPERIOR CON EL PAGO**

**FIGURA 13.4** Documento típico de respuesta

se usa a menudo junto con la salida tabular. Por ejemplo, una orden de salida contiene zonas para los datos del cliente y de la orden además de tablas (o filas de columnas) para los productos ordenados.

**La salida por pantalla** El medio que más ha crecido para las salidas de la computadora es el despliegue en línea de información en un dispositivo visual de despliegue, como la pantalla de una terminal o de PC. El ritmo de la economía de hoy demanda información sobre demanda. La salida por pantalla es muy adecuada para este requerimiento.

Aun cuando la salida por pantalla provee al usuario del sistema un acceso conveniente a la información, la información es sólo temporal. Cuando la información deja la pantalla, se pierde, a menos que se vuelva a exhibir. Por esta razón, generalmente se añaden opciones de salida por la impresora a los diseños de salida por pantalla.

Gracias a la tecnología de salida por pantalla, los reportes tabulares —especialmente los reportes resumidos— pueden presentarse en formatos gráficos. La **salida gráfica** es el uso de una gráfica para transmitir información en forma que muestren tendencias y relaciones que no fácilmente se aprecian en la salida tabular.

Para el usuario del sistema, una ilustración puede ser más valiosa que las palabras. Hay numerosos tipos y estilos de gráficas para presentar información. La figura 13.5 recopila diversos tipos de gráficas que pueden ser una opción de salida con la tecnología de hoy. La tecnología de generación de reportes y el software de hojas de cálculo rápidamente pueden transformar los datos tabulares en gráficas que le permiten al lector sacar conclusiones con mayor rapidez.

La popularidad de la salida gráfica también ha sido estimulada por la disponibilidad de impresoras de gráficos baratos, fáciles de usar y software, especialmente en la industria de la PC. Más tarde en este capítulo le mostraremos un diseño gráfico alternativo para una salida de SoundStage.

**Las terminales de punto de venta** Muchas de las transacciones de hoy de venta al menudeo y del consumidor son realizadas por terminales de punto de venta (point-of-sale, POS). El ejemplo clásico es el cajero automático (automated teller machine, ATM). Las terminales de punto de venta son dispositivos de entrada y salida. En este capítulo, tenemos interés sólo en la dimensión de salida. Los ATM exhiben los saldos de las cuentas e imprimen recibos de transacción. Las cajas registradoras de POS presentan precios y totales a medida que se escanean los códigos de barras, y también producen recibos. Las terminales del punto de venta de lotería generan números aleatorios e imprimen boletos. Todas son ejemplos de salidas que deben ser diseñadas.

**Multimedia** La *multimedia* es un término acuñado para describir colectivamente cualquier información presentada en un formato diferente de números, códigos, y palabras tradicionales. Esto incluye gráficos, sonidos, imágenes, y animación digital. En general se presenta como una moderna adición a la salida por pantalla. Progresivamente, la salida multimedia está siendo impulsada por la transición de las aplicaciones de los sistemas de información hacia la Internet y las intranets (redes internas).

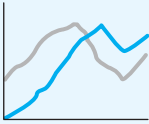
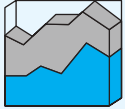
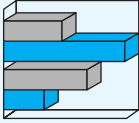
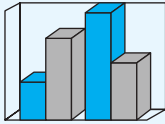
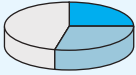
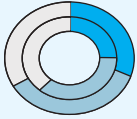

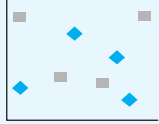
Ya hemos discutido la salida gráfica. Pero otros formatos de multimedia pueden integrarse a los diseños tradicionales de pantalla. Muchos sistemas de información ofrecen películas y animación como parte de las opciones de salidas. Las descripciones de los productos así como las instrucciones de instalación y de mantenimiento pueden integrarse en catálogos en línea usando herramientas de multimedia. Los archivos de sonido también pueden integrarse.

Pero la salida de multimedia no es exclusiva de la tecnología de despliegue en pantalla. El sonido, en forma de sistemas telefónicos basados en tonos, puede usarse para implantar una alternativa interesante de salida. Muchos bancos ofrecen a sus clientes acceso por tonos a una amplia variedad de datos de cuentas, préstamos, y transacciones.

**El correo electrónico** El correo electrónico ha transformado las comunicaciones en el mundo de los negocios moderno, si no es que a toda la sociedad. Se espera que los nuevos sistemas de información tengan capacidad de mandar mensajes. ¿Cómo impacta esto en el diseño de las salidas? Cada vez más los sistemas transaccionales tienen capacidad para funcionar en Internet. Cuando usted compra productos en Internet, casi siempre recibe como salida un correo electrónico automatizado para confirmar su orden. El correo electrónico de seguimiento le puede informar del progreso del cumplimiento de la orden e iniciar el seguimiento del cliente (una forma de salida con regreso).

Las salidas internas también pueden ser distribuidas por correo electrónico. Por ejemplo, un sistema puede enviar la notificación de la disponibilidad de informes nuevos a los usuarios interesados. Sólo esos usuarios que verdaderamente necesitan el reporte accederán a él y lo imprimirán. Esto puede generar un ahorro significativo en costos en la distribución masiva.

**salida gráfica** Salida que utiliza un gráfico para comunicar información.

	Muestra	Criterios de selección
<b>Gráfica de línea</b>		Las <b>gráficas de línea</b> muestran una o más series de datos durante un periodo. Son útiles para resumir y mostrar datos a intervalos regulares. Cada línea representa una serie o categoría de datos.
<b>Gráfica de área</b>		Las <b>gráficas de área</b> son similares a las de línea excepto que se centran en el área bajo la curva. Esa área es útil para resumir y mostrar el cambio de los datos con el tiempo. Cada línea representa una serie o categoría de datos.
<b>Gráfica de barras</b>		Las <b>gráficas de barras</b> son útiles para comparar series o categorías de datos. Cada barra representa una serie o categoría de datos.
<b>Gráfica de columna</b>		Las <b>gráficas de columna</b> son similares a las de barras excepto que las barras son verticales. También puede usarse una serie de gráficas de columna para comparar las mismas categorías para diferentes instantes o intervalos de tiempo. Cada barra representa una serie o categoría de datos.
<b>Gráfica circular (o de pastel)</b>		Las <b>gráficas circulares</b> muestran la relación de las partes con el todo. Son más útiles para resumir los porcentajes de un todo dentro de una serie individual de datos. Cada sector circular representa un artículo de la serie de datos.
<b>Gráfica de dona</b>		Las <b>gráficas de dona</b> son similares a las gráficas circulares excepto que pueden mostrar series múltiples o categorías de datos, cada uno en su propio anillo concéntrico. Dentro de cada anillo, un sector de ese anillo representa un artículo de la serie de datos.
<b>Gráfica de radar</b>		Las <b>gráficas de radar</b> son útiles para comparar diferentes aspectos de más de una serie o categoría de datos. Cada serie de datos se representa como una forma geométrica alrededor de un punto central. Las series múltiples se superponen de modo que puedan compararse.
<b>Gráfica de dispersión</b>		Las <b>gráficas de dispersión</b> son útiles para mostrar la relación entre dos o más series o categorías de datos medidos a intervalos desiguales de tiempo. Cada serie se representa con datos puntuales usando ya sea colores o viñetas diferentes.

**FIGURA 13.5** Tipos de gráficas y criterios de selección

**Hipervínculos** Actualmente muchas salidas están habilitadas en la Red de redes (también llamada simplemente la Red o la Web). Muchas bases de datos y sistemas para que ordene el consumidor están ahora habilitados en la Red. Los hipervínculos en la Red permiten a los usuarios navegar en las listas de registros o buscar registros específicos y acceder a diferentes niveles de información detallada de acuerdo con la necesidad. Obviamente, este medio puede ampliarse y de hecho lo está, a las entradas de la computadora.

La tecnología existe para transformar fácilmente los reportes internos a formatos de HTML o XML para la distribución por la intranet. Esto reduce la dependencia a los reportes impresos y los reportes de pantalla que requieren una versión o un sistema operativo específico (como *Windows*). Esencialmente, todo lo que el receptor requiere es un navegador actualizado que puede correr en cualquier plataforma de computadora (*Windows*, *Macintosh*, *Linux*, o *UNIX*).

Pero la salida habilitada en la Red va más allá de la presentación tradicional de las salidas por la Internet y la intranet. Muchos negocios han invertido dinero en sistemas de reportes internos basados en la Red que consolidan semanas, meses, y años de reportes internos tradicionales en una base de datos organizada de la cual los reportes pueden recuperarse y exhibirse o imprimirse. Estos sistemas no crean salidas nuevas. Simplemente reformatean los reportes previos para el acceso a través de un navegador. Piense en esto como un sistema de archivos de reportes disponibles en la Red. Los ejemplos de tales sistemas de reportes incluyen a DataWatch *Monarch/ES* y NSA *Report.Web*.

**El microfilme** El papel es voluminoso y requiere espacio de almacenamiento considerable. Para resolver este problema, muchos negocios usan el microfilme como un medio de salida. El primer medio de película es el microfilme. Más comúnmente, se convierten en *microficha*, hojas pequeñas de *microfilme* capaces de almacenar docenas o centenares de páginas de salida de la computadora. El uso de película presenta sus propios problemas; la microficha y el microfilme pueden producirse y leerse sólo con equipo especial.

Esto completa nuestra introducción de los conceptos de salidas. Si estudia la figura 13.1 cuidadosamente, usted puede ver que las opciones de implantación y distribución pueden combinarse para desarrollar salidas muy creativas, amigables con el usuario, y excitantes.

## Cómo diseñar y desarrollar prototipos de las salidas

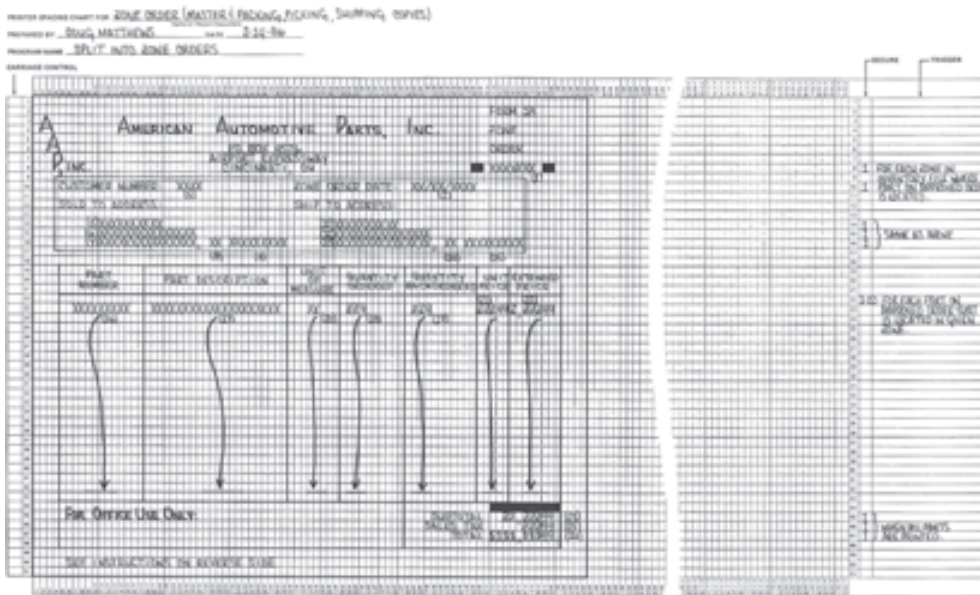
En esta sección, discutiremos y mostraremos el proceso de diseño y desarrollo de prototipos de las salidas. Presentaremos algunas herramientas para documentar y hacer prototipos del diseño de salidas, y también aplicaremos los conceptos que usted aprendió en la última sección. Mostraremos cómo pueden usarse las herramientas automatizadas para diseñar y hacer prototipos de las salidas y plantillas para los usuarios del sistema y los programadores. Como siempre, cada paso de la técnica del diseño de salida será mostrado usando ejemplos sacados de nuestro estudio del caso SoundStage Entertainment Club.

### > Las herramientas automatizadas para el diseño y el desarrollo de prototipos de las salidas

No hace mucho, las herramientas primordiales para el diseño de salidas eran las *tablas de espaciado de la impresora* (vea la figura 13.6) y *plantillas de diseño de salidas*. Hoy, este enfoque no se practica mucho. Es un proceso tedioso que no conduce a las estrategias actuales de desarrollo preferido de aplicación rápida y de desarrollo de prototipos, que usan herramientas automatizadas para acelerar el proceso de diseño.

Antes de la disponibilidad de las herramientas automatizadas, los analistas sólo podían realizar diseños preliminares de salidas para obtener una idea de cómo querían los usuarios de sistema que se vieran las salidas. Con las herramientas automatizadas, podemos desarrollar prototipos más realistas de estas salidas. Quizá la herramienta para formar prototipos menos cara y más ignorada es la hoja de cálculo común. Los ejemplos incluyen 1-2-3 de Lotus y Microsoft *Excel*. El formato tabular de una hoja de cálculo es idealmente adecuado para la creación de prototipos tabulares de salida. Y la mayoría de las hojas de cálculo incluyen medios para convertir rápidamente los datos tabulares a una variedad de formatos gráficos populares. En consecuencia, las hojas de cálculo proporcionan una forma sin precedente para rápidamente formar prototipos de la salida gráfica para los usuarios del sistema.

Posiblemente, la herramienta automatizada más comúnmente usada para el diseño de salidas es el ambiente de desarrollo de aplicación de bases de datos de la PC. Muchos



**FIGURA 13.6** Tabla de espaciamento de la impresora

de ustedes sin duda han aprendido el Microsoft *Access* ya sea en un curso básico de la PC o de desarrollo de bases de datos. Aun cuando *Access* no es lo suficientemente poderoso para desarrollar la mayoría de las aplicaciones a nivel empresarial, usted puede sorprenderse de cuántos diseñadores usan *Access* para hacer el prototipo de tales aplicaciones. Primero, proporciona herramientas de desarrollo rápido para construir con rapidez una base de datos de un solo usuario (o pocos usuarios) y agregar datos de prueba. Esos datos pueden ingresarse después a los prototipos de diseño de las salidas para aumentar su realismo. Los diseñadores pueden usar la opción de reporte de *Access* para esbozar diseños propuestos de salidas y probarlos con los usuarios.

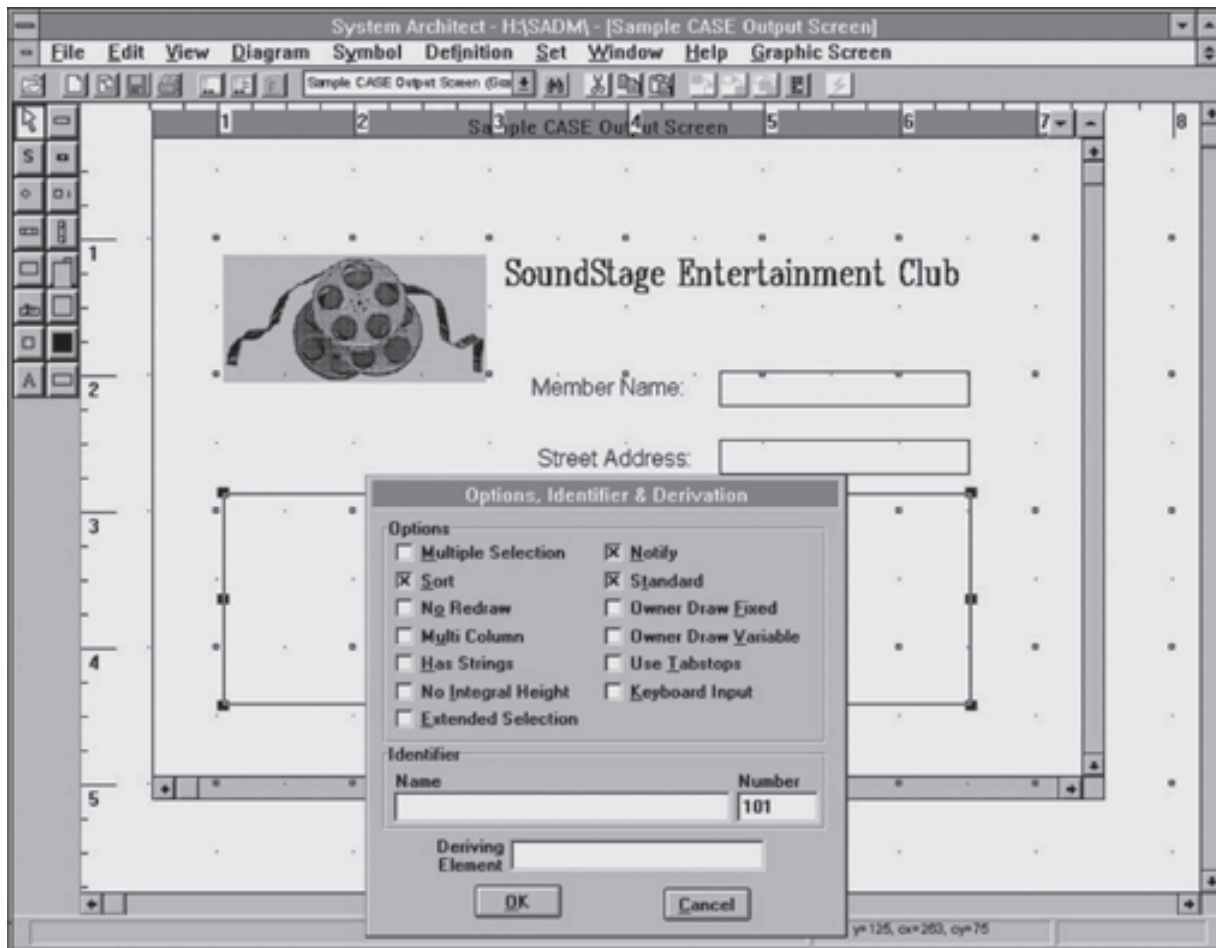
Muchas herramientas CASE incluyen opciones para el esbozo y la formación de prototipos de reportes y de pantallas usando el repositorio de proyecto creado durante el análisis de requerimientos. La opción de diseño de la pantalla del *System Architect* se muestra en la figura 13.7.

Las herramientas automatizadas que acabamos de presentar han acelerado y mejorado significativamente el proceso del diseño de salida. Pero el proceso definitivo del diseño de salidas no solamente formaría el prototipo del diseño de las salidas, sino que también serviría como la implantación final de esa salida. Esta sofisticada solución se encuentra en las herramientas de generación de reportes tales como *Crystal Reports* de Business Objects y *e.Reporting Suite* de Actuate. Estos productos crean el “código” real para ser integrados en el sistema de información operacional. La figura 13.8 ilustra dos pantallas de la herramienta de *Crystal Reports* que se usan para crear un reporte para SoundStage del prototipo de una base de datos.

### > Lineamientos del diseño de salidas

Muchos aspectos son relevantes para el diseño de salidas. La mayoría está motivada por las preocupaciones de ergonomía: el deseo de diseñar salidas que soportarán las formas en las cuales trabajan los usuarios del sistema. Los siguientes principios generales son importantes para diseño de salidas:

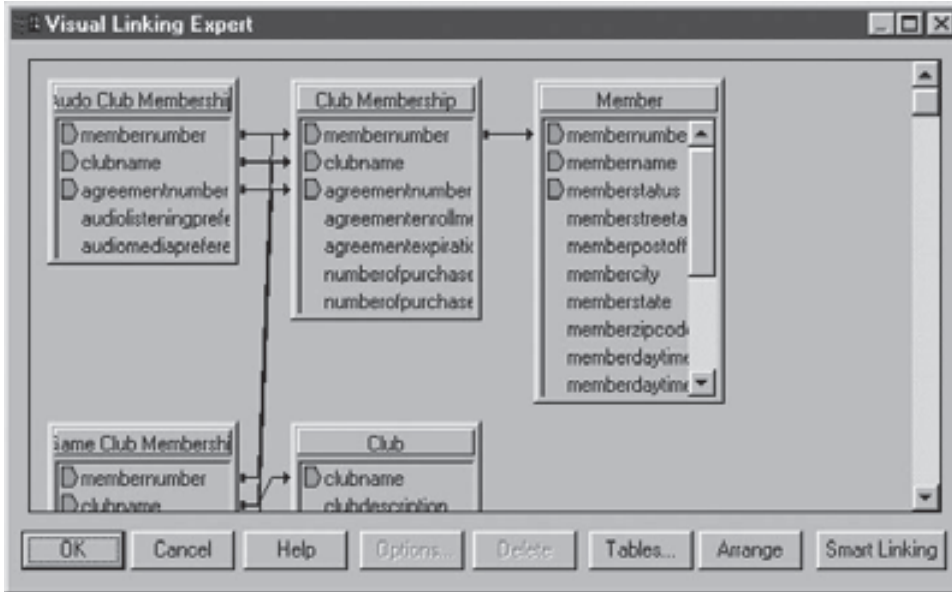
1. *Las salidas de la computadora deberían ser simples de leer e interpretar.* Estos lineamientos pueden mejorar la facilidad de lectura:
  - a) Cada salida deberá tener un título.
  - b) Cada salida deberá fecharse y tener un sello de tiempo. Esto ayuda al lector a darse cuenta de la actualidad de la información (o la falta de ella).
  - c) Los reportes y las pantallas deberán incluir secciones y encabezados para segmentar la información.



**FIGURA 13.7** Herramienta CASE para el diseño de salidas

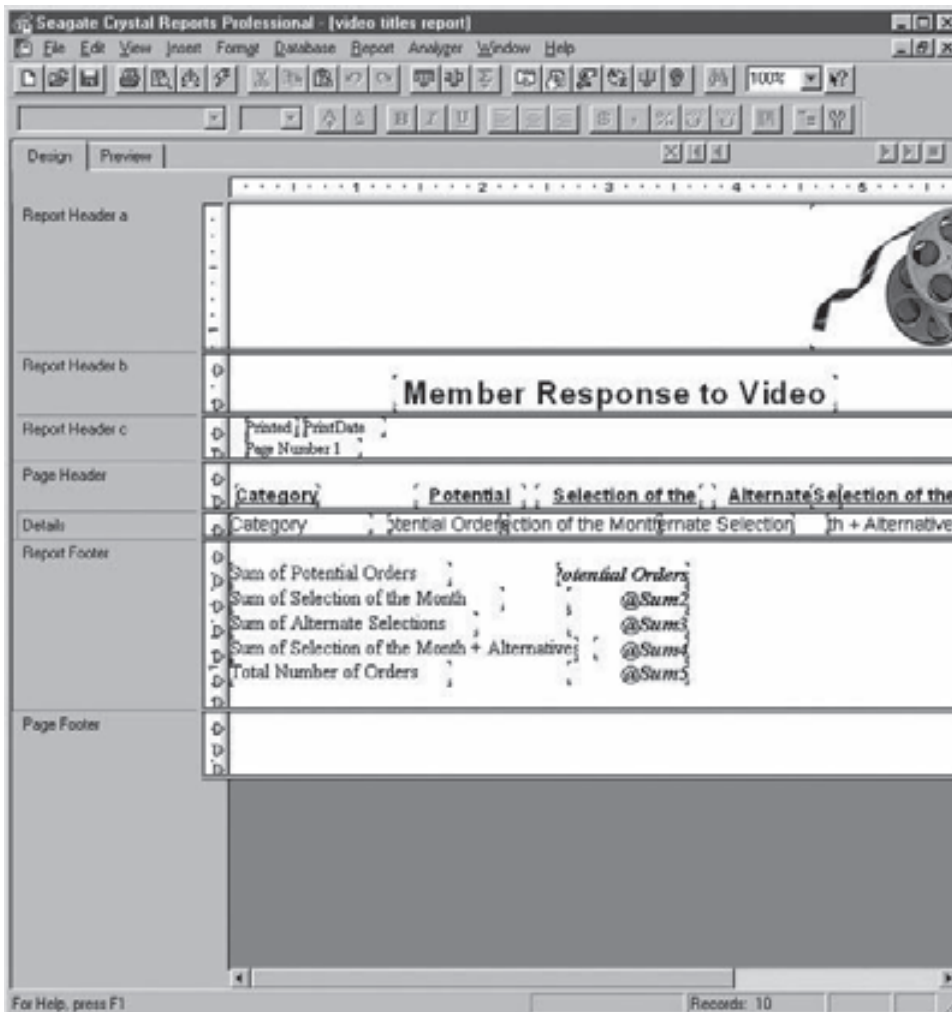
- d) En las salidas basadas en formas, todos los campos deberán estar claramente etiquetados.
  - e) En las salidas basadas en tablas, las columnas deberán estar claramente etiquetadas.
  - f) Como los encabezados de sección, los nombres de los campos, y los encabezados de columna son algunas veces abreviados para conservar espacio, los reportes deberán incluir o proveer acceso a las leyendas para interpretar esos encabezados.
  - g) Sólo la información requerida deberá imprimirse o exhibirse. En las salidas en línea, use métodos para ocultar y proveer información para expandir y contraer los niveles de detalle.
  - b) La información nunca deberá tener que ser manualmente editada para hacerse utilizable.
  - i) La información deberá estar balanceada en el reporte o en la pantalla: no deberá estar demasiado aglomerada, ni demasiado dispersa. También, provea suficientes márgenes y espaciamiento a todo lo largo de la salida para hacerla más legible.
  - j) Los usuarios deben poder encontrar fácilmente la salida, recorrerla hacia arriba y hacia abajo, y salir del reporte.
  - k) La jerga de computación y los mensajes de error deberán omitirse de todas las salidas.
2. *La oportunidad de las salidas de la computadora es importante.* La información de salida debe llegar a los destinatarios mientras ésta sea pertinente para las transacciones o las decisiones. Esto puede afectar la forma de cómo se diseñe e implemente la salida.
  3. *La distribución de (o el acceso a) las salidas de la computadora debe ser suficiente para ayudar a todos los usuarios pertinentes del sistema.* La elección del método de implantación afecta a la distribución.

a)



**FIGURA 13.8**  
Herramienta para el diseño de reportes

b)



4. *Las salidas de la computadora deben ser aceptables para los usuarios del sistema que las recibirán.* Un diseño de salida puede contener la información requerida y aun así no ser aceptable para el usuario del sistema. Para evitar este problema, el analista de sistemas debe entender cómo planea usar la salida el destinatario.

## > El proceso del diseño de salida

El diseño de salidas no es un proceso complicado. Algunos pasos son esenciales, y otros son dictados por circunstancias. Los pasos son:

1. Identificar las salidas de sistema y revisar los requerimientos lógicos.
2. Especificar los requerimientos físicos de las salidas.
3. Conforme sea necesario, diseñar formas externas preimpresas.
4. Diseñar, validar, y probar las salidas usando alguna combinación de:
  - a) Las herramientas del diseño (por ejemplo, bocetos a mano, gráficas del diseño de la impresora/monitor, o CASE).
  - b) Herramientas para formar prototipos (por ejemplo, la hoja de cálculo, DBMS para PC, 4GL).
  - c) Herramientas que generan código (por ejemplo, el generador de reportes).

En las siguientes subsecciones, examinamos estos pasos y los ilustramos con algunos ejemplos del proyecto SoundStage.

**Paso 1: Identificar las salidas del sistema y revisar los requerimientos lógicos** Los requerimientos de salida deberán haberse definido durante el análisis de requerimientos. Los diagramas de flujo de datos físicos (o las unidades del diseño, ambos descritos en el capítulo 11) son un buen punto de partida para el diseño de las salidas. Esos DFD identifican tanto las salidas de red del sistema (del proceso al agente externo) como el método de implantación.

Dependiendo de su metodología de desarrollo de sistemas y estándares, cada uno de estos flujos de datos de salida también puede describirse como un flujo de datos lógicos en un diccionario de datos o un repositorio (vea estructuras de datos, capítulo 8). La estructura de datos para un flujo de datos especifica los atributos o los campos que deben incluirse en la salida. Si esos requerimientos son especificados en la notación algebraica de relaciones (álgebra relacional), usted rápidamente puede determinar qué campos se repiten, cuáles tienen valores opcionales, etcétera. Considere la siguiente estructura de datos:

Estructura de datos que define los requisitos lógicos	Comentario
FACTURA = NÚMERO DE FACTURA	← Identificador único de la salida.
+ FECHA DE LA FACTURA	← Uno de los muchos campos que debe adoptar un valor. La falta de paréntesis indica que se requiere un valor.
+ NÚMERO DE CLIENTE	
+ NOMBRE DEL CLIENTE	← Apuntador a una definición relacionada.
+ DIRECCIÓN DE FACTURACIÓN DEL CLIENTE = DIRECCIÓN>	← Comienza el grupo de campos que se repite 1 – n veces.
+ 1 {FECHA DEL SERVICIO + SERVICIO SUMINISTRADO + CARGO POR EL SERVICIO} N	← Más campos requeridos con valores individuales.
+ SALDO ANTERIOR VENCIDO	
+ PAGOS RECIBIDOS	
+ CARGOS TOTALES POR EL NUEVO SERVICIO	
+ CARGOS POR INTERÉS	
+ NUEVO SALDO VENCIDO	
+ PAGO MÍNIMO VENCIDO	
+ FECHA DEL PAGO VENCIDO	
+ (NÚMERO POR OMISIÓN DE LA TARJETA DE CRÉDITO)	← El campo no tiene que tener un valor.
+ ((MENSAJE DE CRÉDITO   MENSAJE DE PAGO))	← El campo no tiene que tener un valor, pero si lo tiene, suministrará solamente una o dos opciones posibles.



Los prototipos de identificación pueden existir sin los requerimientos precisos. En cualquier caso, una buena declaración de requerimientos debería estar disponible en algún formato.

**Paso 2: Especificar los requerimientos físicos de las salidas** Recuerde que la fase de análisis de decisiones deberá haber establecido alguna expectativa de cómo la mayoría de flujos de datos de salida serán implantados finalmente. En relación con las salidas, las decisiones se hicieron determinando el mejor medio y formato para el diseño y la implantación basándose en:

- Tipo y objetivo de la salida.
- La viabilidad operacional, técnica, y económica.

Como la viabilidad es importante para otros aspectos además de las salidas. El primer conjunto de criterios para evaluar la viabilidad se describe en la siguiente lista:

- ¿La salida es para uso interno o externo?
- Si es una salida interna, ¿es un reporte detallado, resumido o de excepción?
- Si es un reporte externo, ¿es un documento con respuesta?

Después de asegurarse de que entiende qué tipo de reporte será la salida y cómo será usado, usted necesita ocuparse de varios asuntos del diseño:

1. ¿Cuál sería el mejor método de implantación que sirva para la salida? Se discutieron diversos métodos anteriormente en el capítulo. Usted tendrá que entender el propósito o el uso de la salida para determinar el método correcto. Puede seleccionar más de un método para una salida individual: por ejemplo, salida por pantalla con impresión optativa. Claramente, estas decisiones se resuelven de mejor manera con los usuarios del sistema.
  - a) ¿Cuál sería el mejor formato para el reporte? ¿Tabular? ¿Por zona? ¿Gráfico? ¿Alguna combinación?
  - b) Si se desea una impresión, usted debe determinar qué tipo de formato o papel será usado. El papel comercial viene en tres tamaños estándar (todo especificado en pulgadas):  $8\frac{1}{2} \times 11$ ,  $11 \times 14$ , y  $8\frac{1}{2} \times 14$ . Usted necesita determinar las capacidades y las limitaciones de la impresora pretendida.
  - c) Para la salida de la pantalla, usted necesita entender las limitaciones de los dispositivos de despliegue de los usuarios. A pesar del incremento de los monitores de alta resolución mayores que 19 y 21 pulgadas, la mayoría de los usuarios todavía tienen pantallas de 15 y 17 pulgadas y la resolución de la pantalla es tan baja como  $640 \times 480$  pixeles (en especial cuando usted llega directamente a los consumidores en aplicaciones de comercio electrónico). Todavía se recomienda que las salidas de la pantalla (incluyendo formas o páginas dentro de su aplicación) sean diseñadas para el mínimo común denominador.
  - d) Las imágenes de la forma pueden almacenarse e imprimirse con impresoras de láser modernas, eliminando así la necesidad de tratar con los fabricantes de formas en algunos negocios.
2. ¿Con qué frecuencia se genera la salida? ¿Por demanda? ¿Cada hora? ¿Diariamente? ¿Mensualmente?
 

Para las salidas programadas, ¿cuándo necesitan el reporte los usuarios de sistemas?

  - a) Los usuarios generan muchos informes por demanda. Puede ser de ayuda usar el correo electrónico automatizado para notificar a los usuarios que las nuevas versiones de los reportes están disponibles.
  - b) Si los informes deben imprimirse por el departamento de servicios de información, deben ser procesados dentro del calendario de operaciones de los sistemas de información. Por ejemplo, un reporte que el usuario del sistema necesita a las 9:00 a.m. del jueves puede tener que ser programado para las 5:30 a.m. del jueves. No se dispone de más tiempo.
3. ¿Cuántas páginas u hojas de salida se generarán para una copia individual de una salida impresa? Esta información puede ser necesaria para planificar exactamente el consumo del papel y de formas.
4. ¿Requiere la salida copias múltiples? Si es así, ¿cuántas?
  - a) Generalmente se requiere que las impresoras de impacto impriman todas las copias de una forma multicopia al mismo tiempo.
  - b) Las impresoras de láser pueden imprimir copias múltiples de una forma sólo una tras otra. Esto quiere decir que si las copias son diferentes en color o en los campos, las formas preimpresas deben cotejarse antes de la impresión final.

5. Para las salidas impresas, ¿se han completado los controles de distribución? Para las salidas en línea, los controles de acceso deberán determinarse.

Estas decisiones de diseño deberán registrarse en el diccionario de datos/repositorio del proyecto. Consideremos un ejemplo de nuestro caso el sistema de SoundStage Entertainment Club.

Una salida para SoundStage es el REPORTE RESUMIDO DE LA RESPUESTA DEL MIEMBRO. Este informe fue pedido para proveer a la gerencia interna de información con respecto a las respuestas del cliente para las ofertas promocionales mensuales. Se establecieron los siguientes requerimientos de diseño:

1. El gerente pedirá el reporte de su propia estación de trabajo. Se determinó que la información deberá presentarse como una salida por pantalla tanto en formato tabular como en formato gráfico (para determinarse vía los prototipos).
  - a) Todos los gerentes tienen monitores de despliegue de 17 pulgadas o mayores.
  - b) Los gerentes deberán tener la opción de obtener una salida de la impresora de láser vía su configuración LAN. Las salidas impresas deberán estar en papel comercial de  $8\frac{1}{2} \times 11$  pulgadas.
2. Los gerentes deben poder exhibir el informe por demanda. Ellos han pedido notificación por correo electrónico automática de la disponibilidad de cualquier versión recién generada del informe. Un hipervínculo con la última versión del reporte también deberá estar disponible en la página principal estándar del sitio de la Red de cada gerente de Servicios del Miembro, nivel 3 y superior.
3. La salida gráfica deberá ser desplegable en una pantalla individual e imprimible en una sola página. Los datos tabulares pueden imprimirse en una a dos páginas. El volumen de páginas no se considera significativo para este informe.
4. El reporte debe tener acceso restringido para los gerentes cuyas cuentas de red llevan privilegios de cuenta de nivel 3 o superiores. El reporte deberá incluir una marca de agua "Confidencial" y un mensaje que prohíba el compartir de la información o su distribución externa sin el permiso escrito de la Auditoría Interna.

**Paso 3: Diseño de formas preimpresas** Los documentos externos y con respuesta se separan aquí para una consideración especial porque contienen considerable información constante y preimpresa que debe ser diseñada antes de diseñar la salida final. En la mayoría de los casos, el diseño de una forma preimpresa se subcontrata a un fabricante de formas. El negocio, sin embargo, debe especificar los requerimientos del diseño y revisar cuidadosamente los prototipos de diseño. Los requerimientos de diseño encaran aspectos tales como los siguientes:

- ¿Qué información preimpresa debe aparecer en la forma? Esto incluye información de contacto, encabezados, etiquetas, y otra información común que debe aparecer en todas las copias de la forma.
- ¿Deberá diseñarse la forma para el envío postal? Si es así, la posición de las direcciones cobra importancia dependiendo de si se usan sobres con ventana o no.
- ¿Cuántas formas serán requeridas para imprimirse cada día? ¿Cada semana? ¿Cada mes? ¿Cada año?
- ¿Cuál será el tamaño de la forma? El tamaño de la forma, junto con el volumen (arriba), puede afectar los costos de correos.
- ¿La forma estará perforada para servir como un documento con respuesta? También, para los documentos con respuesta la posición de la dirección se hace más crítica porque la dirección del remitente para la salida externa se convierte en la dirección postal para el documento devuelto.
- ¿Qué leyendas, políticas, e instrucciones necesitan imprimirse en la forma (tanto anverso como reverso)?
- ¿Qué colores serán usados, y para cuáles copias?

Para documentos externos, hay también varias alternativas. El carbón y el carbón químico son las técnicas de la multicopia más comunes. Los carbones selectivos son una variación por medio de la cual ciertos campos en la copia maestra no se imprimirán en una o más de las copias restantes. Los campos a ser omitidos deben comunicarse al fabricante de formas. La impresión dual es una técnica por medio de la cual dos conjuntos de formas, posiblemente incluyendo el papel al carbón, se imprimen lado a lado en la impresora.

Una forma preimpresa SoundStage de salida fue previamente exhibida en la figura 13.3.

**Paso 4: Diseño, validación y salidas de prueba** Después de que las decisiones del diseño y los detalles han sido registrados en el repositorio del proyecto, debemos diseñar el formato real del informe. El formato o el esquema de una salida afectan directamente la habilidad del usuario de sistema para leerla e interpretarla. La mejor forma de esquematizar el formato es esbozar o, tanto mejor, generar una muestra del reporte o el documento. Necesitamos mostrarle ese boceto o prototipo al usuario del sistema, obtener una retroalimentación, y modificar la muestra. Es importante usar datos realistas o razonables y mostrar todos los puntos de control.

El aspecto más importante durante el paso del diseño es el formato. La figura 13.9 resume un número de aspectos del diseño y consideraciones para reportes impresos y tabulares. Muchas de estas consideraciones son igualmente aplicables a las salidas de pantalla. También, la salida de la pantalla ofrece varias consideraciones especiales que están resumidas en la figura 13.10.

La gerencia de SoundStage expresó preocupación de que la salida del RESUMEN DE RESPUESTA DEL MIEMBRO potencialmente podría volverse demasiado larga. A menudo el gerente se interesa en ver sólo información relacionada con las respuestas del miembro para una o algunas promociones diferentes del producto. Así, se decidió que el gerente necesitase la habilidad de “hacer a la medida” la salida. Las pantallas usadas para permitir al gerente especificar la hechura a la medida deberán hacerse en prototipo así como también el reporte y la gráfica que contengan la información real. La figura 13.11a) muestra el prototipo de la

**FIGURA 13.9** Principios de diseño del reporte tabular

Aspecto de diseño	Lineamiento de diseño	Ejemplos
Tamaño de página	Anteriormente, la mayoría de los reportes se imprimían en papel muy grande. Esto requería engargolado y almacenamiento especiales. Actualmente, los tamaños de página preferidos son el estándar (8½" × 11") y legal (8½" × 14"). Estos tamaños son compatibles con la predominancia de las impresoras láser en los negocios modernos.	No es aplicable.
Orientación de la página	La orientación de la página es el ancho y la longitud de una página a medida que gira. La orientación de <i>retrato</i> (por ejemplo, 8½ A × 11 L) frecuentemente se prefiere porque está orientada de la manera que orientamos la mayoría de los libros y reportes; sin embargo; frecuentemente se necesita el tipo de <i>paisaje</i> (por ejemplo, 11 A × 8½ L) para reportes tabulares porque pueden imprimirse más columnas.	
Encabezados de página	Los encabezados de página deben aparecer en cada página. Como mínimo, deben incluir un título de reporte reconocible, fecha y tiempo, y números de página. Los encabezados pueden consolidarse en una línea o usarse líneas múltiples.	<p>4 de ENERO, 2001 <span style="float: right;">PÁGINA 4 de 6</span></p> <p style="text-align: center;">SOBREINSCRIPCIONES POR CURSO</p>
Leyendas de reporte	Una leyenda es una explicación de las abreviaturas, colores, o códigos usados en un reporte. En un reporte impreso, puede imprimirse una leyenda solamente en la primera página o en cada página. En una pantalla desplegada, una leyenda puede estar disponible como una caja de diálogo emergente.	<p>LEYENDA DEL REPORTE</p> <p>ASIENTOS NÚMERO DE ASIENTOS EN EL SALÓN</p> <p>LIM LÍMITE DE INSCRIPCIÓN DEL CURSO</p> <p>REQ NÚMERO DE ASIENTOS REQUERIDOS POR DEPARTAMENTO</p> <p>RES NÚMERO DE ASIENTOS RESERVADOS POR DEPARTAMENTO</p> <p>USADO NÚMERO DE ASIENTOS USADOS POR DEPARTAMENTO</p> <p>DISP NÚMERO DE ASIENTOS DISPONIBLES POR DEPARTAMENTO</p> <p>SOBRE NÚMERO DE SOBREINSCRIPCIONES POR DEPARTAMENTO</p>

**FIGURA 13.9** Conclusión

Aspecto de diseño	Lineamiento de diseño	Ejemplos																																																
Encabezados de columna	Los encabezados de columna deben ser cortos y descriptivos. Si es posible, evite las abreviaturas. Desafortunadamente, esto no siempre es posible. Si se usan abreviaturas, incluya una leyenda (vea las “leyendas de reporte”).	Autoexplicativo.																																																
Alineamientos del encabezado	La relación de los encabezados de columna con los datos reales de la columna bajo estos encabezados puede afectar grandemente su facilidad de lectura. El alineamiento debe probarse con los usuarios en cuanto a preferencias, con énfasis especial en el riesgo de una mala interpretación de la información. Vea ejemplos de las posibilidades (que pueden combinarse).	<p>Justificación a la izquierda (buena para campos más grandes y de longitud variable):</p> <pre>NOMBRE ===== XXXXXXXX X XXXXXXXXXXX XXXXXX</pre> <p>Justificación a la derecha (buena para algunos campos numéricos, especialmente campos de dinero); asegúrese de alinear los puntos decimales:</p> <pre>CANTIDAD ===== \$\$\$ \$\$\$.</pre> <p>Centro (buena para campos de longitud fija y algunos campos de longitud moderada):</p> <pre>ESTATUS ===== XXXX XXXX</pre>																																																
Espaciamiento de columnas	El espaciamiento entre columnas impacta la su facilidad de lectura. Si las columnas están demasiado apretadas, tal vez los usuarios no diferencien apropiadamente entre las columnas. Si el espaciamiento es demasiado disperso, el usuario puede tener dificultad al seguir un renglón individual a lo largo de la página. Como regla general preliminar, ponga de 3 a 5 espacios entre cada columna.	Autoexplicativo.																																																
Encabezados de fila	Las primeras una o dos columnas deben servir como los datos de identificación que diferencien a cada fila. Las filas deben secuenciarse de manera que soporten su uso. Frecuentemente las filas se clasifican con una clave numérica o en forma alfabética.	<p>Por número:</p> <table border="0"> <tr> <td>IDENTIFICACIÓN DEL ESTUDIANTE</td> <td>NOMBRE DEL ESTUDIANTE</td> </tr> <tr> <td>=====</td> <td>=====</td> </tr> <tr> <td>999-38-8476</td> <td>MARY ELLEN KUKOW</td> </tr> <tr> <td>999-39-5857</td> <td></td> </tr> </table> <p>De manera alfabética:</p> <table border="0"> <tr> <td>SERVICIO</td> <td>CANCELAR</td> <td>SUSCRIBIR</td> <td>TOTAL</td> </tr> <tr> <td>=====</td> <td>=====</td> <td>=====</td> <td>=====</td> </tr> <tr> <td>HBO</td> <td>45</td> <td>345</td> <td>7665</td> </tr> </table>	IDENTIFICACIÓN DEL ESTUDIANTE	NOMBRE DEL ESTUDIANTE	=====	=====	999-38-8476	MARY ELLEN KUKOW	999-39-5857		SERVICIO	CANCELAR	SUSCRIBIR	TOTAL	=====	=====	=====	=====	HBO	45	345	7665																												
IDENTIFICACIÓN DEL ESTUDIANTE	NOMBRE DEL ESTUDIANTE																																																	
=====	=====																																																	
999-38-8476	MARY ELLEN KUKOW																																																	
999-39-5857																																																		
SERVICIO	CANCELAR	SUSCRIBIR	TOTAL																																															
=====	=====	=====	=====																																															
HBO	45	345	7665																																															
Formateado	Frecuentemente los datos se almacenan sin caracteres de formateado para ahorrar espacio de almacenaje. Las salidas deben reformatear a esos datos de modo que concuerden con las normas del usuario.	<table border="0"> <tr> <td>Tal como se almacena:</td> <td>Como salida:</td> </tr> <tr> <td>307877262</td> <td>307-87-7262</td> </tr> <tr> <td>8004445454</td> <td>(800) 444-5454</td> </tr> <tr> <td>02272000</td> <td>Feb 27, 2000</td> </tr> </table>	Tal como se almacena:	Como salida:	307877262	307-87-7262	8004445454	(800) 444-5454	02272000	Feb 27, 2000																																								
Tal como se almacena:	Como salida:																																																	
307877262	307-87-7262																																																	
8004445454	(800) 444-5454																																																	
02272000	Feb 27, 2000																																																	
Puntos de control	Frecuentemente, las filas representan grupos de datos significativos. Esos grupos deben agruparse lógicamente en el reporte. La transición de un grupo al siguiente se llama punto de control y frecuentemente es seguida por los subtotales del grupo.	<table border="0"> <tr> <td>JERARQUÍA</td> <td>NOMBRE</td> <td>SALARIO</td> <td></td> </tr> <tr> <td>=====</td> <td>=====</td> <td>=====</td> <td></td> </tr> <tr> <td>CPT</td> <td>JANEWAY, K</td> <td>175 000</td> <td></td> </tr> <tr> <td>CPT</td> <td>KIRK, J</td> <td>225 000</td> <td></td> </tr> <tr> <td>CPT</td> <td>PICARD, J</td> <td>200 000</td> <td></td> </tr> <tr> <td>CPT</td> <td>SISKO, B</td> <td>165 000</td> <td></td> </tr> <tr> <td></td> <td>TOTAL DE LOS EMPLEADOS EJECUTIVOS</td> <td>765 000</td> <td>☛ un punto de control</td> </tr> <tr> <td>LTC</td> <td>CHAKOTAY</td> <td>110 000</td> <td></td> </tr> <tr> <td>LTC</td> <td>DATA</td> <td>125 000</td> <td></td> </tr> <tr> <td>LTC</td> <td>RIKER, W</td> <td>140 000</td> <td></td> </tr> <tr> <td>LTC</td> <td>SPOCK, S</td> <td>155 000</td> <td></td> </tr> <tr> <td></td> <td>EXEC OFFCR TOTAL</td> <td>530 000</td> <td></td> </tr> </table>	JERARQUÍA	NOMBRE	SALARIO		=====	=====	=====		CPT	JANEWAY, K	175 000		CPT	KIRK, J	225 000		CPT	PICARD, J	200 000		CPT	SISKO, B	165 000			TOTAL DE LOS EMPLEADOS EJECUTIVOS	765 000	☛ un punto de control	LTC	CHAKOTAY	110 000		LTC	DATA	125 000		LTC	RIKER, W	140 000		LTC	SPOCK, S	155 000			EXEC OFFCR TOTAL	530 000	
JERARQUÍA	NOMBRE	SALARIO																																																
=====	=====	=====																																																
CPT	JANEWAY, K	175 000																																																
CPT	KIRK, J	225 000																																																
CPT	PICARD, J	200 000																																																
CPT	SISKO, B	165 000																																																
	TOTAL DE LOS EMPLEADOS EJECUTIVOS	765 000	☛ un punto de control																																															
LTC	CHAKOTAY	110 000																																																
LTC	DATA	125 000																																																
LTC	RIKER, W	140 000																																																
LTC	SPOCK, S	155 000																																																
	EXEC OFFCR TOTAL	530 000																																																
Fin del reporte	El final de un reporte debe indicarse claramente para asegurar que los usuarios tengan el reporte completo.	*** FINAL DEL REPORTE ***																																																

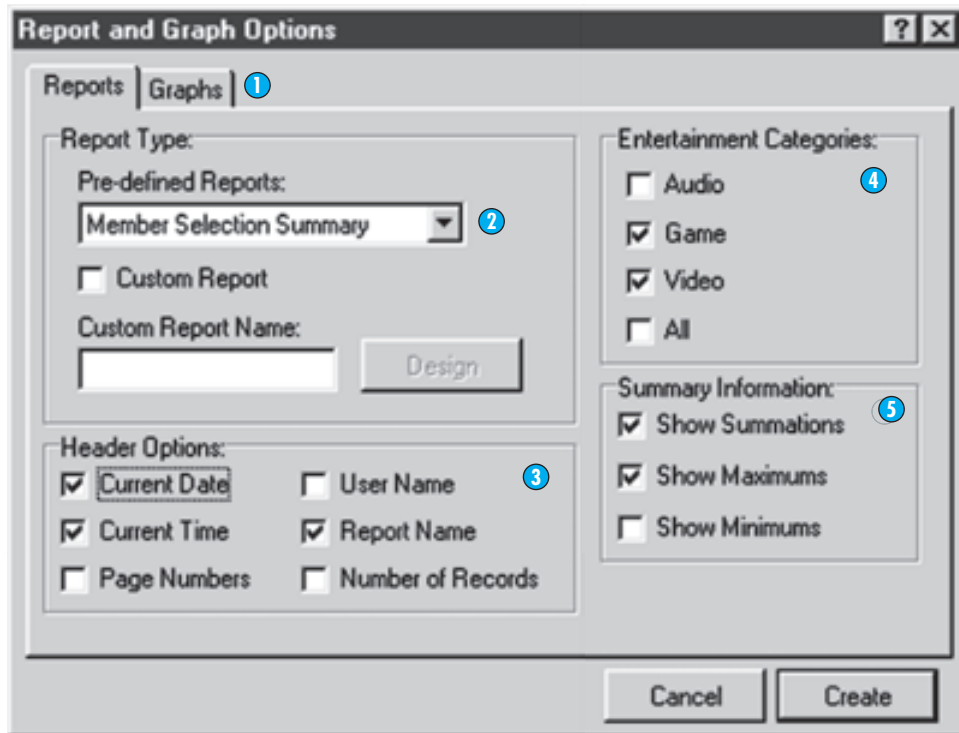
**FIGURA 13.10** Principios de diseño de salidas de pantalla

Consideración de diseño de pantalla	Lineamientos de diseño
Tamaño	Diferentes monitores soportan resoluciones diferentes. El diseñador debe considerar el “mínimo común denominador”. El tamaño de ventana por omisión debe ser menor que o igual que el monitor con la resolución más baja en la comunidad de usuarios. Por ejemplo, si algunos usuarios tienen solamente un monitor con resolución de 640 × 480 píxeles, no diseñe las ventanas para que se abran con una resolución de 800 × 600 píxeles.
Caracoleo	Las salidas en línea tienen la ventaja de que no están limitadas por la página física. Esto también puede ser una desventaja si información importante tal como los encabezados de columna se sale de la pantalla. Si es posible, congele los encabezados importantes en la parte superior de la pantalla.
Navegación	<p>Los usuarios siempre deben tener un sentido de dónde están en una red de pantallas en línea. Dado eso, los usuarios también requieren la capacidad de navegar entre pantallas.</p> <p><b>VENTANAS:</b> Las salidas aparecen en ventanas llamadas formas. Una forma puede desplegar un registro o muchos. La barra de caracoleado debe indicar dónde está usted en el reporte. Frecuentemente se proveen botones para moverse hacia delante y hacia atrás a través de los registros en el reporte y para salir del mismo.</p> <p><b>INTERNET:</b> Las salidas aparecen en ventanas llamadas páginas. Una página puede desplegar un registro o muchos. Pueden usarse botones o hipervínculos para navegar a través de los registros. También pueden usarse máquinas de búsqueda hechas a la medida para navegar a posiciones específicas dentro de un reporte.</p>
Particionamiento	<p><b>VENTANAS:</b> Se forman zonas dentro de las formas. Cada forma es independiente de la otra pero puede relacionarse. Las zonas pueden recorrerse en forma secuencial independientemente. La barra de Outlook de Microsoft es un ejemplo. Las zonas pueden usarse para leyendas o puntos de control que lleven al usuario a diferentes secciones dentro del reporte.</p> <p><b>INTERNET:</b> Los marcos son páginas dentro de las páginas. Los usuarios pueden caracolear independientemente dentro de las páginas. Los marcos pueden mejorar los reportes de muchas maneras. Pueden usarse para una leyenda, una tabla de contenidos, o información resumida.</p>
Ocultamiento de información	<p>Las aplicaciones en línea tales como las que corren en Windows o dentro de un navegador de Internet ofrecen posibilidades de ocultar información hasta que se necesite o se haga importante. Los ejemplos de este ocultamiento de información incluyen:</p> <ul style="list-style-type: none"> <li>• Controles de drill-down que muestran información mínima y suministran a los lectores maneras sencillas de expandir o el nivel de detalle de despliegue.             <ul style="list-style-type: none"> <li>– En las salidas de Windows el uso de un signo pequeño más o menos en una caja pequeña a la izquierda de un registro de datos ofrece la opción de expandir o contraer el registro para un detalle mayor o menor. Toda esta expansión y contracción ocurre dentro de la ventana de la salida.</li> <li>– En las aplicaciones de intranet, cualquier pieza dada de información resumida puede mejorarse como un hipervínculo para expandir esa información a mayor detalle. Comúnmente, la información expandida se abre en una ventana separada de modo que el lector pueda usar los botones hacia delante y hacia atrás del navegador para cambiar entre niveles de detalle.</li> </ul> </li> <li>• La información puede activar cajas emergentes de diálogo.</li> </ul>
Realzado	<p>Puede usarse el realzado en los reportes para llamar la atención de los usuarios acerca de datos equivocados, datos de excepción, o problemas específicos. El realzado también puede ser una distracción si se usa mal. La investigación en curso de los factores humanos seguirá guiando nuestro uso futuro del realzado. Los ejemplos de realzado incluyen:</p> <ul style="list-style-type: none"> <li>• Color (evitar colores que los daltónicos no pueden distinguir).</li> <li>• Fuente y tipo de letra (el cambio de tipo de letra puede llamar la atención).</li> <li>• Justificación (izquierda, derecha o al centro).</li> <li>• Uso de guiones (no se recomienda en los reportes).</li> <li>• Parpadeo (puede llamar la atención o ser molesto).</li> <li>• Video inverso.</li> </ul>
Impresión	<p>Para muchos usuarios todavía hay comodidad en los reportes impresos. Siempre provea a los usuarios con la opción de imprimir una copia permanente del reporte. Para el uso de Internet, pueden ser necesario que los reportes estén disponibles en formatos estándares industriales tales como Adobe Acrobat, que permite a los usuarios abrir y leer los reportes usando software gratis y fácilmente disponible.</p>

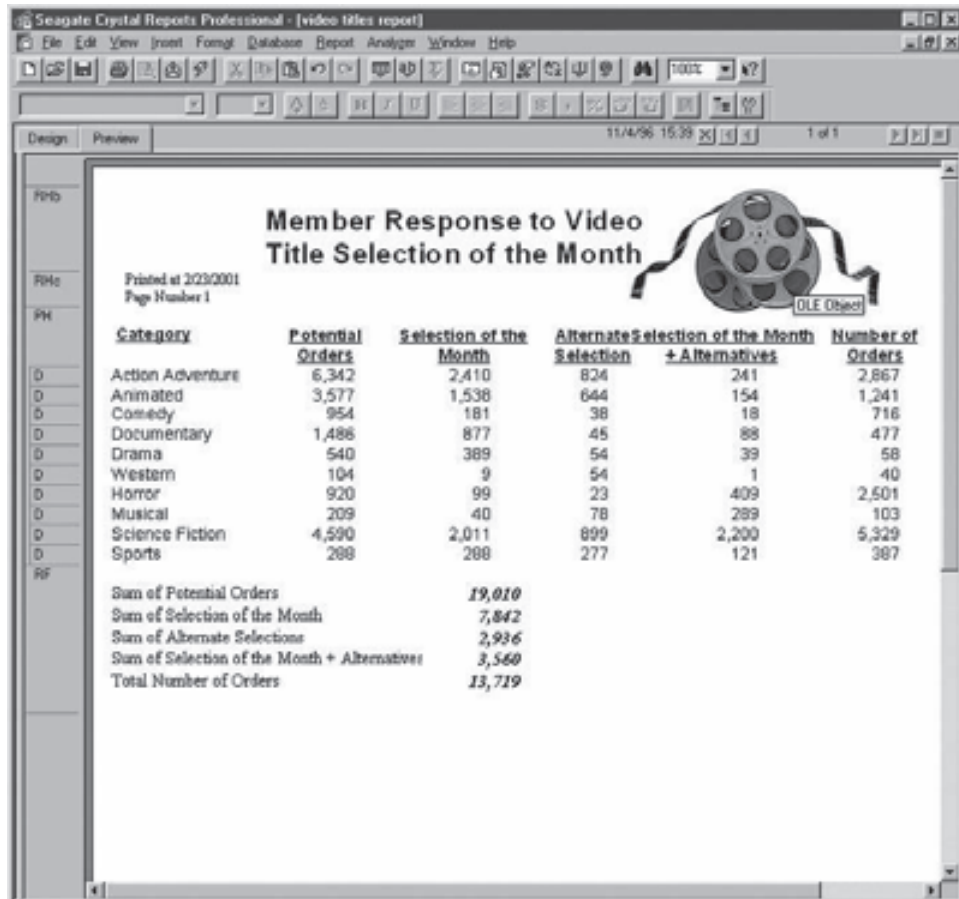
**FIGURA 13.11**

Prototipos de reportes tabulares para hacer los reportes personalizados

a) Prototipo de un reporte personalizado



b) Prototipo de un reporte tabular



pantalla que el usuario puede usar para escoger un reporte particular (o gráfica) y hacer a la medida su contenido. Deberán observarse los siguientes puntos:

- 1 Se usa una caja de diálogo con etiqueta para permitir al usuario seleccionar entre obtener un reporte y obtener una gráfica. Se usa un control de etiqueta para presentar una serie de información relacionada. Si el usuario da un clic sobre la etiqueta rotulada “Gráficas”, la información sería exhibida para hacer a la medida la salida como una gráfica.
- 2 Se usa una lista desplegable hacia abajo para seleccionar el reporte deseado. El usuario puede dar un clic sobre la flecha descendente para obtener una lista de reportes posibles entre los cuales escoger.
- 3 Se provee al usuario con una serie de casillas de verificación que corresponden a las opciones generales para hacer a la medida el reporte seleccionado. El usuario simplemente “verifica” las opciones que él desea tener en el reporte.
- 4 También se usa un grupo de casillas de verificación para permitirle al usuario seleccionar una o más categorías del producto que él desea incluir en el informe.
- 5 Otra vez, se usa un grupo de casillas de verificación para permitirle al usuario que profundice al hacer el reporte a la medida. Aquí se permite al usuario indicar el tipo de reporte resumido o de totales deseados para cada categoría del producto.

Examinemos un prototipo del reporte que resultará del diálogo previo de adecuación del reporte. La figura 13.11b) es un prototipo de una versión de salida de la pantalla del reporte real. Examine el contenido y apariencia del diseño tabular. Observe que el usuario tiene permiso de enrollar vertical y horizontalmente para observar el reporte completo. Además, se proveen botones para permitirle al usuario alternar hacia adelante y atrás para observar diferentes páginas del reporte.

Finalmente, observemos un prototipo de una versión gráfica de la salida del RESUMEN DE RESPUESTA DEL MIEMBRO (vea la figura 13.12). Observe lo siguiente:

- La gráfica está claramente etiquetada a lo largo de los ejes vertical y horizontal.
- Una leyenda ha sido provista para auxiliar en la interpretación de las barras de la gráfica.

Cuando usted está formando prototipos de las salidas, es importante involucrar al usuario para obtener una retroalimentación. Deberá permitirse al usuario realmente “ejercitar” o probar las pantallas. Parte de esa experiencia deberá incluir la demostración de cómo puede obtener el usuario instrucciones o ayuda apropiada, cómo profundizar para obtener información adicional, cómo navegar por las páginas, cómo pedir los diferentes

Ventas trimestrales del formato popular de audio de DVD

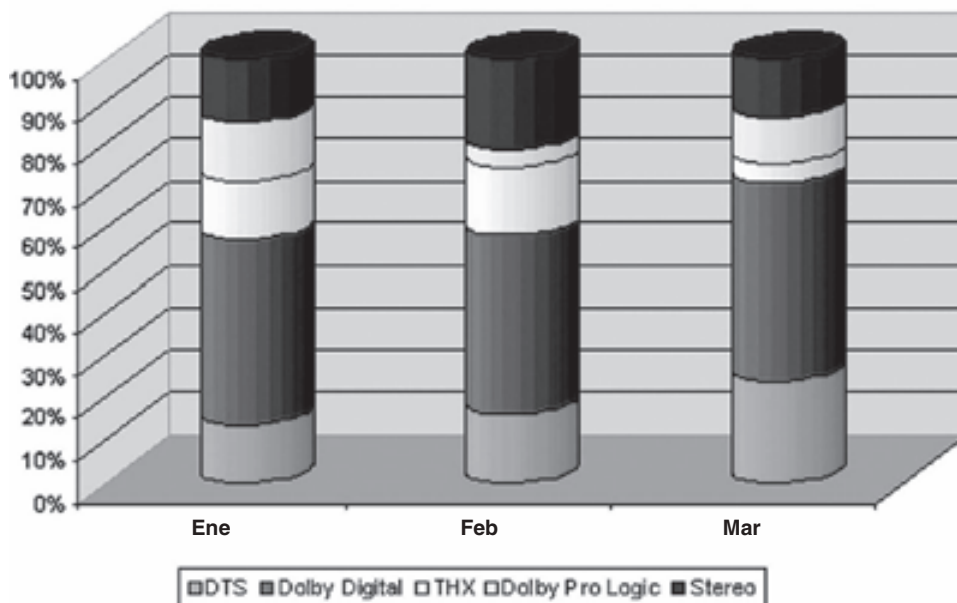


FIGURA 13.12

Prototipo del reporte gráfico

**FIGURA 13.13**

Prototipo de la salida con registro individual

**Member Order Detail**

**Order Information:** ①

Order Number:	Promotion Number:	Member Number:
1929921	2	10001
Club Name:	Order Date:	Order Fill Date:
V	1/15/00	1/30/00
Order Status:	Sales Tax:	Pre-Payment:
	\$ 4.92	\$ 0.00
Sub-Total	Shipping Fee:	Amount Due:
\$ 98.58	\$ 3.50	\$ 107.00

**Shipping Information:** ①

Ship to Name:  
Joe Smith

Shipping Address:  
4589 Johnson Drive

City: Lafayette      State: IN      Zip Code: 47867

Shipping Instruction:  
none

②

③

Refresh

Print

Close

formatos que están disponibles, cómo dimensionar las salidas, y cómo realizar pruebas de las capacidades de adecuación. Todas las características deberán mostrarse o probarse.

Hasta ahora, nosotros hemos presentado muestras sólo de un reporte tabular y de uno gráfico. Otro tipo de salida es un reporte con un registro a la vez. Los usuarios pueden navegar hacia adelante y hacia atrás a través de los registros individuales en un archivo. En la figura 13.13 se muestra un ejemplo de una pantalla para una salida con un registro a la vez. Llamamos su atención a lo siguiente:

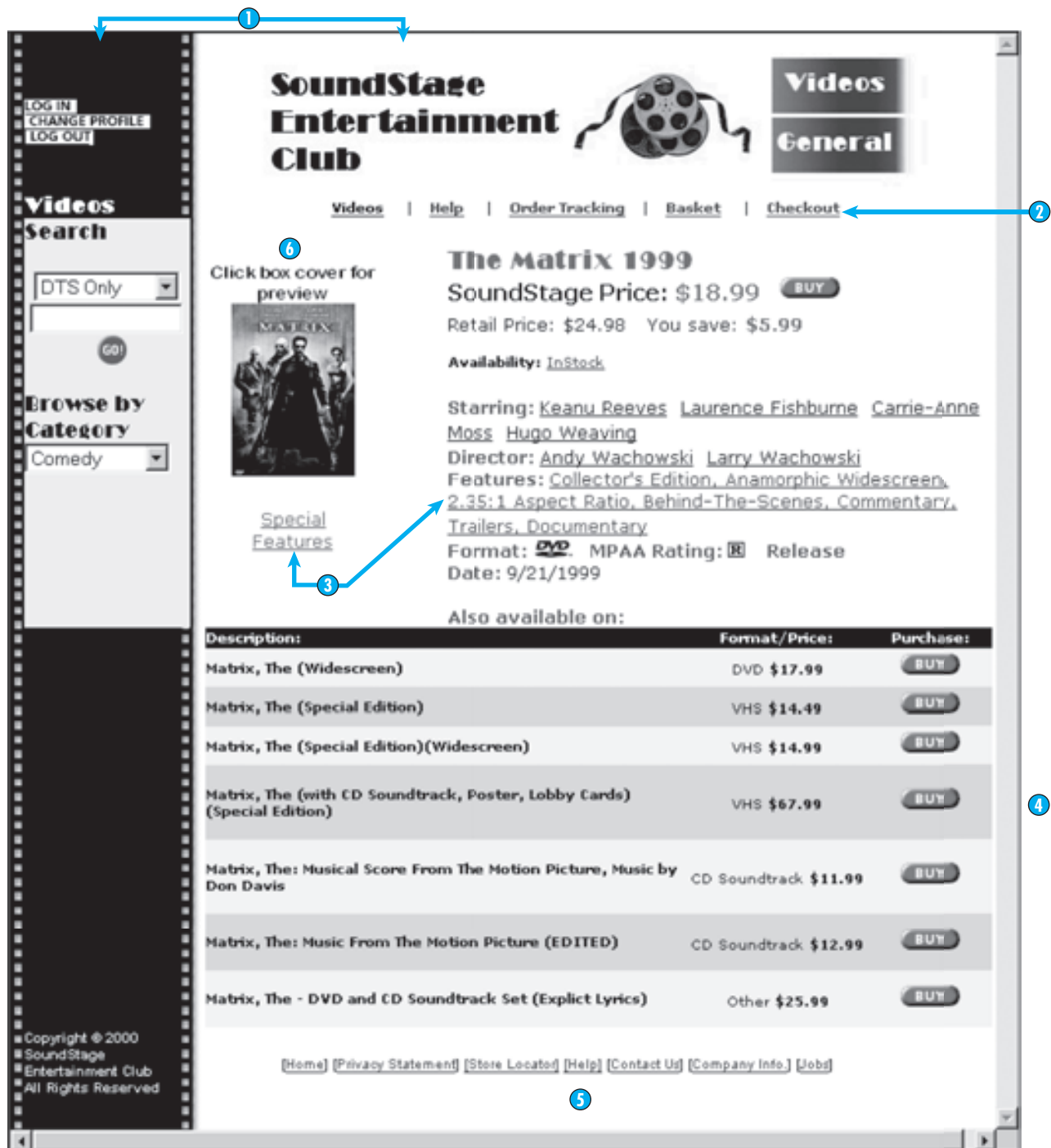
- ① Cada campo está claramente etiquetado.
- ② Se han agregado botones para la navegación entre registros. Los botones casi universalmente aceptados son para PRIMER REGISTRO, SIGUIENTE REGISTRO, REGISTRO ANTERIOR, y ÚLTIMO REGISTRO.
- ③ Añadimos botones para que el usuario obtenga una copia impresa de la salida, así como también para salir del reporte cuando termine. (Consistente con la formación de prototipos, el programador escribirá más tarde el código para salir.)

### > Las salidas basadas en la Web y el comercio electrónico

Las últimas consideraciones del diseño de salida que deseamos presentar son concernientes a las salidas basadas en la Red. El proyecto de SoundStage añadirá diversas capacidades de comercio electrónico y de negocios electrónicos al Sistema de información de servicios a los miembros. Algunas de estas capacidades afectarán al diseño de la salida.

Un requerimiento de salida lógica para el proyecto es navegar en el catálogo. Los miembros deberían poder navegar y buscar en catálogos, probablemente como un prólogo





**FIGURA 13.14** Aplicaciones de sistemas de información

para colocar las órdenes. El catálogo mismo es la salida. La figura 13.14 es una pantalla del prototipo para la salida del catálogo físico. Observe lo siguiente:

- 1 Esta salida usa marcos para permitirle al usuario enfocar separadamente la atención en la navegación y en la salida.
- 2 La pantalla usa los hipervínculos para proveer navegación a través de estructuras complejas del menú que están relacionadas con la salida.

The screenshot shows the SoundStage Entertainment Club website. The main page features the site logo, navigation links (Videos, Help, Order Tracking, Basket, Checkout), and a product listing for 'The Matrix 1999'. The product price is \$18.99, with a retail price of \$24.98, resulting in a \$5.99 savings. The availability is 'InStock'. A 'BUY' button is present. A preview room window is overlaid on the page, titled 'The Matrix Preview', showing a video player with a play button (marked with a circled '1') and a 'GET Windows Media Player' button (marked with a circled '2'). The preview room also includes a 'www.sndstage.com' URL and a 'realplayer' logo. The main page also features a search bar, a 'Videos' section, and a list of links for 'The Matrix' including 'Anamorphic Widescreen', 'The-Scenes', and 'Commentary'. A table of purchase options is visible on the right side of the page.

Format/Price:	Purchase:
DVD \$17.99	<input type="button" value="BUY"/>
VHS \$14.49	<input type="button" value="BUY"/>
VHS \$14.99	<input type="button" value="BUY"/>
VHS \$67.99	<input type="button" value="BUY"/>
Soundtrack \$11.99	<input type="button" value="BUY"/>
Soundtrack \$12.99	<input type="button" value="BUY"/>
Other \$25.99	<input type="button" value="BUY"/>

**FIGURA 13.15** Salida del Windows Media Player (un reproductor multimedia)

- ③ Los hipervínculos también permiten al usuario obtener información adicional. Esta funcionalidad se conoce como “drill-down” (profundizar).
- ④ Se usa el sombreado para separar cada línea de detalle. Esta práctica refleja el acercamiento más artístico usado para diseñar salidas basadas en la Red. También, los botones de “COMPRA” han transformado eficazmente esta salida en un disparador para entradas subsiguientes. ¡Éste es el equivalente virtual en el comercio electrónico de un documento con respuesta!

- 5 La mayoría de los diseños de pantalla de salida basados en la Red requieren pies de página estándar en la pantalla para proporcionar navegación adicional.
- 6 Una imagen puede ser un objeto seleccionable. En este caso representa otro tipo de drill-down donde el usuario puede obtener información adicional.

Otro requerimiento de salida es permitir a los miembros jugar con los videos de avances (en inglés *video trailers*) y con los archivos de audio de los productos para ver con antelación las posibles compras. La exhibición preliminar se desencadenará por un hipervínculo en la pantalla anterior, y activará a un reproductor multimedia como se muestra en la figura 13.15. Se espera que estas extensiones de salida se conviertan en la norma a medida que aumenta la popularidad de las aplicaciones basadas en Internet y en la intranet.

- 1 Las salidas basadas en la Red frecuentemente usan plug-ins. Esta pantalla de salida tiene los botones estándar asociados con un reproductor de audio o de video típico.
- 2 Comúnmente, las salidas basadas en la Red también proveen enchufes correctos o versiones con enchufe necesarias para la sesión.

Este capítulo suministró un panorama detallado del diseño y la formación de prototipos de las salidas de computadora para un proyecto de desarrollo de sistemas. Se recomienda que ahora continúe con el capítulo 14 y no que lo salte antes de pasar al capítulo 15. El capítulo 14 trata del diseño y de la formación de prototipos de las entradas de una aplicación. El capítulo 15 trata del diseño y de la formación de prototipos de la interfaz general de una aplicación. Como tal, el capítulo 15 incluye la conjunción y la presentación de las funciones de aplicaciones presentadas en los capítulos 13 y 14.

## Resumen

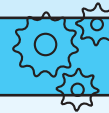


1. Las salidas pueden clasificarse según dos características:
  - a) Su distribución adentro o afuera de la organización y por las personas que las leen y usan.
  - b) Su método de implantación.
2. Las salidas internas están destinadas para los dueños del sistema y los usuarios dentro de una organización. Sólo raramente encuentran su camino fuera de la organización. Hay tres subclases de salidas internas:
  - a) Los reportes detallados: presentan información con poco o ningún filtrado o restricciones.
  - b) Los reportes resumidos: categorizan la información para gerentes que no quieren empantarse en los detalles.
  - c) Los reportes de excepción: filtran datos antes de presentarlos al gerente como información.
3. Las salidas externas dejan la organización. Están destinadas a los clientes, proveedores, socios, y organismos regulatorios. Generalmente concluyen o reportan sobre las transacciones de negocios.
4. Algunas salidas son tanto externas como internas. Comienzan como salidas externas que salen de la organización pero regresan parcial o totalmente.
5. Las salidas con regreso son aquellas salidas externas que finalmente reingresan al sistema como entradas.
6. Un buen analista de sistemas considerará todas las opciones disponibles para implantar una salida. Existen varios métodos y formatos:
  - a) El medio más común para las salidas de la computadora es el papel: la salida impresa. Comúnmente las salidas internas se imprimen en papel en blanco (llamado papel comercial). Las salidas externas y los documentos con respuesta se imprimen en formas preimpresas.
    - i) Quizás el formato más común para la salida impresa es tabular. La salida tabular presenta información como columnas de texto y números.
    - ii) Una alternativa para la salida tabular es la salida por zona, la cual coloca texto y números dentro de las áreas o los cuadros designados de una forma o una pantalla.
  - b) La salida de la pantalla es muy adecuada para el ritmo de la economía de hoy, que requiere información por demanda. La tecnología de salida por pantallas permite que los reportes se presenten en formatos gráficos. La salida gráfica es el uso de una gráfica pictórica para transmitir información en las formas que muestran tendencias y relaciones no fácilmente vistas en la salida tabular.
  - c) Muchos de las transacciones de hoy del consumidor y de ventas al menudeo se permiten o realzan por las terminales de punto de venta (POS).
  - d) Los *multimedia* es un término acuñado para describir colectivamente cualquier información presentada en un formato que no sean los números, códigos, y palabras tradicionales. Esto incluye gráficos, sonido, imágenes, y animación.
- e) El correo electrónico se está volviendo un medio de salida muy popular como una manera de llegar a audiencias grandes y generar ahorros significativos de costos.
- f) Los hipervínculos de la Red permiten a los usuarios navegar en listas de registros o buscar registros específicos y recuperar niveles diversos de información detallada por demanda.
- g) El papel requiere espacio de almacenamiento considerable. Para vencer este problema, muchos negocios usan el microfilme como un medio de salida.
7. La herramienta automatizada más comúnmente usada para el diseño de salidas es el ambiente de desarrollo de aplicaciones de bases de datos de PC. Muchas herramientas CASE también incluyen opciones para los reportes y la disposición de las pantallas, así como la formación de prototipos usando el repositorio del proyecto creado durante el análisis de requerimientos.
8. Los siguientes principios generales son importantes para el diseño de salidas:
  - a) Las salidas de la computadora deberán ser simples de leer e interpretar.
  - b) La oportunidad de las salidas de la computadora es importante: sus destinatarios deben recibir información de salida mientras la información sea pertinente para las transacciones o las decisiones.
  - c) La distribución de (o el acceso a) las salidas de la computadora deben ser suficientes para ayudar a todos los usuarios pertinentes de sistema.
  - d) Las salidas de la computadora deben ser aceptables para los usuarios del sistema que las recibirán.
9. El diseño de salidas no es un proceso complicado. Algunos pasos son esenciales, y otros son dictados por las circunstancias. Los pasos son:
  - a) Identificar las salidas del sistema y revisar los requerimientos lógicos.
  - b) Especificar los requerimientos de las salidas físicas.
  - c) Conforme sea necesario, diseñar formas externas preimpresas.
  - d) Diseñar, validar, y probar las salidas usando alguna combinación de:
    - i) Las herramientas de diseño (por ejemplo, bocetos a mano, gráficas del diseño de la impresora /despliegue, o CASE).
    - ii) Herramientas para formación de prototipos (por ejemplo, hojas de cálculo, PC DBMS, 4GL).
    - iii) Herramientas que generan código (por ejemplo, el generador de reportes).



## Preguntas de repaso

1. ¿Cuáles son algunas de las características de prototipos?
2. ¿Cómo se clasifican las salidas?
3. ¿Cuál es la diferencia entre el reporte sumario y el reporte de excepción?
4. ¿Cuáles son algunos ejemplos de reportes externos?
5. ¿Cuál es la diferencia entre la salida tabular y la salida por zonas?
6. ¿Por qué son necesarios los reportes impresos además de las salidas de pantalla?
7. ¿Cuáles son algunos ejemplos de gráficas pictóricas?
8. ¿Por qué deberán usarse las salidas gráficas?
9. ¿Cuáles son algunos de los lineamientos del diseño de salidas?
10. ¿Cuáles son los pasos básicos para el diseño de las salidas?
11. ¿Cuáles son las dos clases más importantes de criterios que los analistas deben considerar cuando especifican requerimientos de salida física? ¿Por qué son importantes?
12. ¿Cuáles son algunos de los aspectos de diseño que los analistas necesitan considerar?
13. ¿Para qué son las formas preimpresas?
14. ¿Cuál es la ventaja de usar marcos al exhibir información en la Internet?



## Problemas y ejercicios

1. Cien años atrás, si usted diseñaba un reporte, ¿qué métodos diferentes de medios y de entrega estaban disponibles? ¿Qué hay acerca de 50 años atrás? ¿Y hoy? ¿Cuál piensa usted que ha sido el cambio más grande en reportes en los pasados 100 años?
2. Usted trabaja como diseñador de sistemas para el Departamento de Programa de Asistencia Social del lugar. El director de la agencia de protección de niños del lugar está preocupado acerca de la carga de casos de la agencia y el tiempo que los casos permanecen abiertos. El objetivo de la agencia es no tener casos abiertos más antiguos que 60 días, y preferentemente ninguno mayor que 30 días. El director quiere un informe mensual mostrando el número de casos, por edad, para cada uno de los 12 trabajadores de protección del niño en la agencia. ¿Qué subclase de reporte deberá diseñar usted? ¿El formato de salida deberá ser tabular o por zonas? Describa la estructura de datos definiendo los requerimientos lógicos del informe. Use el formato descrito en el capítulo.
3. Use la información en la pregunta anterior para crear un prototipo del reporte; use una herramienta automatizada tal como Microsoft Access (o, si prefiere, puede crear un prototipo de la manera tradicional). Pueblo el informe con varios registros de muestra, por orden alfabético por apellido del trabajador.
4. El director de la agencia de protección del niño está contento con el informe, pero también le gustaría verlo en formato gráfico. ¿Qué tipo(s) de diagrama(s) sería(n) inapropiado(s)? ¿Por qué? ¿Qué tipo(s) de diagrama(s) sería(n) apropiado(s) para este tipo de reporte? ¿Por qué? ¿Cuál piensa usted que sería el mejor? ¿Por qué?
5. ¿Qué subclase de reporte diseñaría usted para el gerente de ventas de una concesión de distribución de coches cuyo trabajo es revisar las ventas de vehículos cada semana y cada año a la fecha? ¿Qué elementos de datos deberán incluirse en el reporte? ¿Qué deberá preguntar usted al gerente de ventas antes de que usted diseñe el reporte?
6. El gerente de ventas también tiene que saber, semanalmente, quién no hizo su cuota de ventas en la semana anterior y/o el año a la fecha. ¿Qué subclase de reporte es necesario en esta situación? ¿Qué elementos de datos incluiría usted, y cómo los agruparía?
7. Coteje las definiciones o los ejemplos en la primera columna con los términos en la segunda columna.
 

A. Sistema de archivos para reportes habilitados por demanda en la Red	1. Reporte detallado
B. Cajero automático	2. Gráfica de diseño de despliegues
C. Medio tradicional de salida	3. Salida externa
D. Reporte de cuentas en mora	4. Gráfica de dispersión
E. Transición entre diferentes grupos de datos	5. Reporte resumido
F. Reporte de vehículos en el inventario	6. Punto de control
G. Botón de "Compra" en el sitio Web	7. Salida por zonas
H. Herramienta de diseño de pantallas que casi ya no se usa	8. Salida con regreso

- I. Pedido de ventas  
 J. Muestra la relación entre dos o más series de datos  
 K. Reporte trimestral de ventas por región  
 L. Recibo de ventas
9. Reporte de excepción  
 10. DataWatch *Monarch/ES*  
 11. Terminal de punto de venta  
 12. Microfilme
8. El gerente de ventas le ha pedido que desarrolle una gráfica automatizada para mostrar las ventas anuales de la compañía por trimestre para los últimos cinco años. El gerente considera a los gráficos de barras como aburridos y quiere usar una gráfica circular, en lugar de eso, para mostrar el reporte de ventas de cinco años en un reporte fácil para leer. ¿Qué piensa usted de la idea del gerente? Explique.
9. El director de la Agencia de Protección del Niño está muy contento con el reporte resumido que usted diseñó en la pregunta 2. Para ayudar a los trabajadores de protección del niño a manejar y priorizar su carga de casos, al director ahora le gustaría que usted diseñe un reporte que iría a cada trabajador, mostrando sus casos abiertos, incluyendo la edad de cada caso. Aún más, el informe deberá ser un documento con respuesta, donde los trabajadores de protección del niño pueden informar el estatus de cada uno de sus casos abiertos, incluyendo la fecha estimada de terminación. ¿Qué subclase de reporte se necesita? ¿Qué elementos de datos se necesitan? ¿En qué orden deberán listarse los casos? Cree un diseño de prototipo para este informe.
10. Complete las siguientes frases.
- a) El propósito de las salidas es presentar \_\_\_\_\_ a los usuarios del sistema. Porque son la parte más \_\_\_\_\_ de un sistema de información, los usuarios del sistema y los dueños a menudo basan \_\_\_\_\_ de un sistema de información en las salidas.
- b) Al diseñar las salidas, un buen lugar para comenzar es con \_\_\_\_\_, porque identifican tanto el método \_\_\_\_\_ como el \_\_\_\_\_.
- c) Las salidas pueden ser clasificadas en categorías por dos características: 1) por su \_\_\_\_\_ y \_\_\_\_\_, y 2) por su \_\_\_\_\_.
- d) En un reporte, \_\_\_\_\_ a menudo ocurre en \_\_\_\_\_, que están acostumbrados a efectuar una transición de un \_\_\_\_\_ de datos al siguiente.
- e) En un reporte tabular, \_\_\_\_\_ es influenciado por la columna \_\_\_\_\_, lo cual generalmente debería ser 3-5.
11. Usted es un diseñador de sistemas que trabaja en la división de tecnología de la información de una corporación manufacturera grande con plantas a todo lo largo del país. El CIO le menciona a usted al pasar que el vicepresidente de mercadeo quiere un nuevo reporte a nivel ejecutivo que muestre la producción diaria por región y por oficina para revisar los niveles de producción y fijar los problemas rápidamente. Su CIO le dice a usted que tenga listo un diseño preliminar y un prototipo para pasado mañana. Con base en la información que usted ha recibido, ¿qué tipo de reporte es necesario? ¿Es para uso interno o externo? Suponiendo que el sistema de información de la corporación ya captura los datos necesarios para este informe, ¿cuáles son algunos de los aspectos de diseño restantes?
12. En el escenario anterior, qué herramienta común podía usar usted si su organización no usa las herramientas de CASE ni aquellas reservadas para la escritura de reportes para el diseño y la formación del prototipo de la pantalla? Para un reporte a nivel ejecutivo, ¿cuáles son los principios más críticos que deben aplicarse al diseñar la salida? (Recuerde, su futuro en la compañía puede depender de saber y poder aplicar estos principios.)
13. Usted se ha ofrecido a trabajar en el sitio Web de su biblioteca local. La biblioteca planifica desarrollar un catálogo en línea de libros que pueden ser apartados por los usuarios de la biblioteca desde sus computadoras domésticas por Internet. Muchos de estos usuarios son personas de la tercera edad. ¿Cuáles son algunos de los aspectos del diseño de pantallas que deben tomarse en cuenta?

## Proyectos e investigación



1. En la década de los noventa (y aun antes) hubo una gran cantidad de debate respecto a la oficina sin documentos. Algunos expertos de la industria y futuristas predijeron que en un corto tiempo, el papel se convertiría en un producto de legado en muchas organizaciones. Aun hoy la realidad parece muy diferente y, de hecho, los negocios están consumiendo y emborronando más papel que nunca.

Haga alguna investigación en la Web acerca de artículos contemporáneos y pasados en este tema.

- a) Describa los artículos que encontró.  
 b) Compare y contraste sus puntos de vista.  
 c) Contacte a un organismo grande y una dependencia gubernamental en su área. ¿Consideran ambos que la oficina sin documentos sea un

- objetivo? Si es así, ¿cuáles son sus planes para lograrlo?, y ¿se está haciendo algún progreso?
- d) ¿Qué hay acerca de su organización o escuela?
  - e) ¿Cuál es su posición sobre este tema? ¿Piensa usted que una oficina sin documentos sea un concepto viable? ¿Por qué sí o por qué no?
2. Diseñar una forma o la pantalla de la interfaz ha sido comparado con observar a un gimnasta olímpico: Se ve engañosamente simple hasta que usted trata de hacerlo realmente. Considere las siguientes preguntas:
    - a) Con base en su experiencia, así como también en sus lecturas de éste y otros libros de texto, ¿qué hace que una forma o pantalla de la interfaz sea “buena” y otra “mala”?
    - b) Escoja una forma o pantalla de interfaz que usted piense que es particularmente horrenda. Describa por qué.
    - c) Rediseñe la forma o pantalla en una que usted piense que sea “buena”.
    - d) Haga que un par de discípulos o socios comparen y critiquen las versiones de “antes de” y “después de” la forma o pantalla. ¿Cómo evaluaron su versión “después de” comparada con la versión “antes de”?
    - e) ¿Puede tener usted una forma o pantalla bien diseñada si los datos mismos que deben ser capturados no están bien diseñados? ¿Por qué sí o por qué no?
    - f) En el pueblo global de hoy, ¿una forma o interfaz de diseño que es considerada buena en una cultura sería considerada universalmente buena? ¿Qué tanta influencia tienen las diferencias culturales en el diseño?
  3. Aunque puede ser un cliché decir que vivimos un tiempo de cambio tecnológico sin precedente, es difícil verdaderamente comprender los cambios enormes que han tenido lugar en muy poco tiempo y su impacto sobre nosotros. Para ayudar a obtener una impresión de estos cambios, considere las siguientes preguntas:
    - a) Identifique las formas diferentes de métodos de salida desarrollados en los pasados 1 000 años, y dibújelas en una línea de tiempo. ¿Cuántos métodos de salida identificó usted, y cuántos de ellos estaban comercialmente disponibles en los últimos 50 años? ¿En los últimos 25? ¿En los últimos 10?
    - b) ¿Cuál es la versión más antigua de un documento con respuesta que usted puede encontrar?
    - c) Según su investigación, ¿cuándo el microfilme se hizo ampliamente disponible? ¿Cuál fue su impacto en las organizaciones del sector privado y público?
    - d) ¿Qué hay acerca de la salida por pantalla? ¿Cuándo se generalizó el uso de los monitores de PC? ¿Cuál ha sido su impacto en las organizaciones del sector privado y público?
  - e) De todos los métodos de salida en uso hoy, ¿cuál piensa usted que tuvo el impacto más significativo en los gobiernos y las culturas? ¿Por qué?
4. Las predicciones de avances tecnológicos futuros y grandes adelantos son notablemente poco confiables. Por ejemplo, en los años de 1960 algunos futuristas predijeron que nosotros todos pronto iríamos a trabajar en nuestra aeronave personal o manejando automóviles de energía atómica.
    - a) En la década de 1980, hubo una gran cantidad de especulación acerca de que los hologramas pronto podrían convertirse en una forma común y revolucionaria de salida. Indague este tema; ¿qué encontró usted?
    - b) Indague artículos recientes que describan métodos nuevos de salida que están todavía en la etapa de concepto o en el tablero de dibujo (virtual), pero que los expertos de la industria predicen que podremos ver en un futuro no muy distante. Describa lo que usted encuentre.
    - c) ¿Cuál predice *usted* que será el siguiente gran adelanto en los métodos de salida?
    - d) Si su predicción es correcta, ¿cuál es el impacto potencial en lo que usted hace, como analista de sistemas o diseñador, o en cómo lo hace?
    - e) ¿Piensa usted que la mayoría de analistas de sistemas y diseñadores deberían poner atención a las nuevas tecnologías que aun no están comercialmente disponibles si bien muchas de ellas nunca resultan bien? ¿Deberán sentarse y esperar a ver qué aparece en el mercado?
  5. Muchas organizaciones han implantado una intranet de compañía. Pero parece que relativamente pocos (al menos a partir de esta fecha) han integrado su intranet con herramientas de productividad de escritorio, como Microsoft *Office*, correo electrónico y calendarización, y las aplicaciones específicas de entrada/salida de datos usadas por los empleados.
    - a) Contacte varios organismos locales del sector privado y público. ¿Tienen actualmente una intranet implantada?
    - b) Describa cómo se usan las la intranet y que características tienen. ¿Están integradas cualesquiera de estas la intranet con las herramientas de productividad y/o las aplicaciones usadas por la organización?
    - c) Vea si usted puede echar un vistazo a varias de estas intranet. ¿Están bien diseñadas sus pantallas de interfaz? ¿Cuáles son las diferencias, si hay alguna, entre las aplicaciones de la intranet y las de la Internet que necesitan ser tomadas en consideración cuando se diseñen las interfaces de la pantalla para una intranet?

- d) ¿Qué ocurre si usted tuvo la oportunidad de diseñar una intranet totalmente integrada para su organización? ¿Qué características y funcionalidad incluiría usted?
- e) Cree un diseño del prototipo para su intranet.
6. En el ambiente global de la economía de hoy, la información sobre demanda es la norma esperada. Éste es un desarrollo muy reciente que ha tenido un impacto profundo en la organización e individuos. Considere las siguientes preguntas:
- a) En 1800, si una compañía mercante en Europa enviaba un reporte con respuesta a su agente en Nueva York, ¿cuánto tiempo la compañía esperararía que tomaría (como mínimo) para recibir el reporte de regreso?
- b) ¿Qué hay en 1900?
- c) ¿En 1950?
- d) ¿En 2005 (como máximo)?
- e) Describa lo que usted cree que sea el impacto más significativo que ha tenido este cambio en la velocidad de reporte tanto en las organizaciones como en los individuos.
- f) ¿Es este cambio extraordinario en la velocidad de reporte —y en las expectativas— bueno o malo? ¿Para la organización? ¿Para un empleado individual?

## Casos breves



1. Coleccione un ejemplo de un informe detallado, un reporte resumido, y un reporte de excepción. Entréguelos a su profesor, junto con una descripción breve de la información en ellos. ¿Cuáles son las similitudes entre los reportes? ¿Las diferencias?
2. Puede instalarse una forma en línea para “descargar” el contenido de la forma en un correo electrónico y enviarlo a una cuenta por correo electrónico específica. Encuentre los retazos de código para hacer esto, y crear una forma simple en línea que enviará el contenido a su dirección de correo electrónico. Llene y entregue su forma al menos una vez. Envíe el correo electrónico que usted ha recibido con el contenido de la forma a su profesor, junto con el URL de su forma.
3. La información sólo deberá ingresarse en un sistema de información *una vez*. Después de eso, la información deberá ser compartida digitalmente a través de los departamentos, sin necesidad del reingreso de la información. ¿Por qué es esto? ¿Qué tipo de problemas comunes de formateado de datos piensa usted que corren los diseñadores de sistemas cuando establecen formas en línea (como el que usted hizo en la pregunta 2) que mandan los datos directamente a una base de datos?
4. Encuentre un ejemplo de una salida realmente bien diseñada (podría ser una forma, un reporte, un correo electrónico, etcétera). Luego encuentre un ejemplo de uno deficientemente diseñado. Presente ambos a su clase. Dirija un debate sobre cómo mejorar el ejemplo deficientemente diseñado, usando atributos específicos de excelencia a partir de la salida bien diseñada. Usted será calificado en su habilidad para dirigirse a la clase y trabajar como un miembro del equipo para mejorar el medio de salida.

## Ejercicios de equipo e individuales



1. Individual: La universidad, la carrera, y la familia (sin mencionar todas las demás cosas que hacemos) todas toman una cantidad increíble de tiempo y de energía. Reflexione por un momento ya sea en su vida en su estado actual o cómo puede ser en el futuro. ¿Qué tan bien está usted balanceando la vida y la carrera? ¿Cómo manejará usted los conflictos futuros entre ellas? ¿Cuáles son sus prioridades en la vida? No entregue cualquier trabajo. Usted puede, si quiere, debatir en un formato de mesa redonda.
2. Individual: Parte de ser una buena persona de sistemas es leer y entender a la gente. Observe a alguien que parece un poco abatido o está teniendo un mal día. Haga algo agradable para esa persona. No importa si usted la conoce o no.
3. Para estudiantes que están a punto de graduarse y están buscando empleo: Vaya a una papelería y compre algunos artículos de papelería elegantes y correctos para escribir agradecimientos para las entrevistas. Como un equipo, desarrolle 1) una gran carta de agradecimiento (para una entrevista de trabajo), y 2) un conjunto de preguntas que usted piensa que podrían hacerle en una entrevista. Fuera de la clase, ocupe tiempo en hacer entrevistas actuadas con amigos o familia.





## Lecturas recomendadas

Andres, C. *Great Web Architecture*. Foster City, CA: IDG Books Worldwide, 1999. Los libros sobre el diseño eficaz de la interfaz con la Red están comenzando a aparecer. La ciencia de la ergonomía para las interfaces de la Red no ha progresado todavía hasta las interfaces cliente/servidor (por ejemplo, *Windows*). Aquí hay uno de los primeros títulos que explora muchas dimensiones de la arquitectura de la Red y que hace interfaz mediante el uso de ejemplos de la vida real.

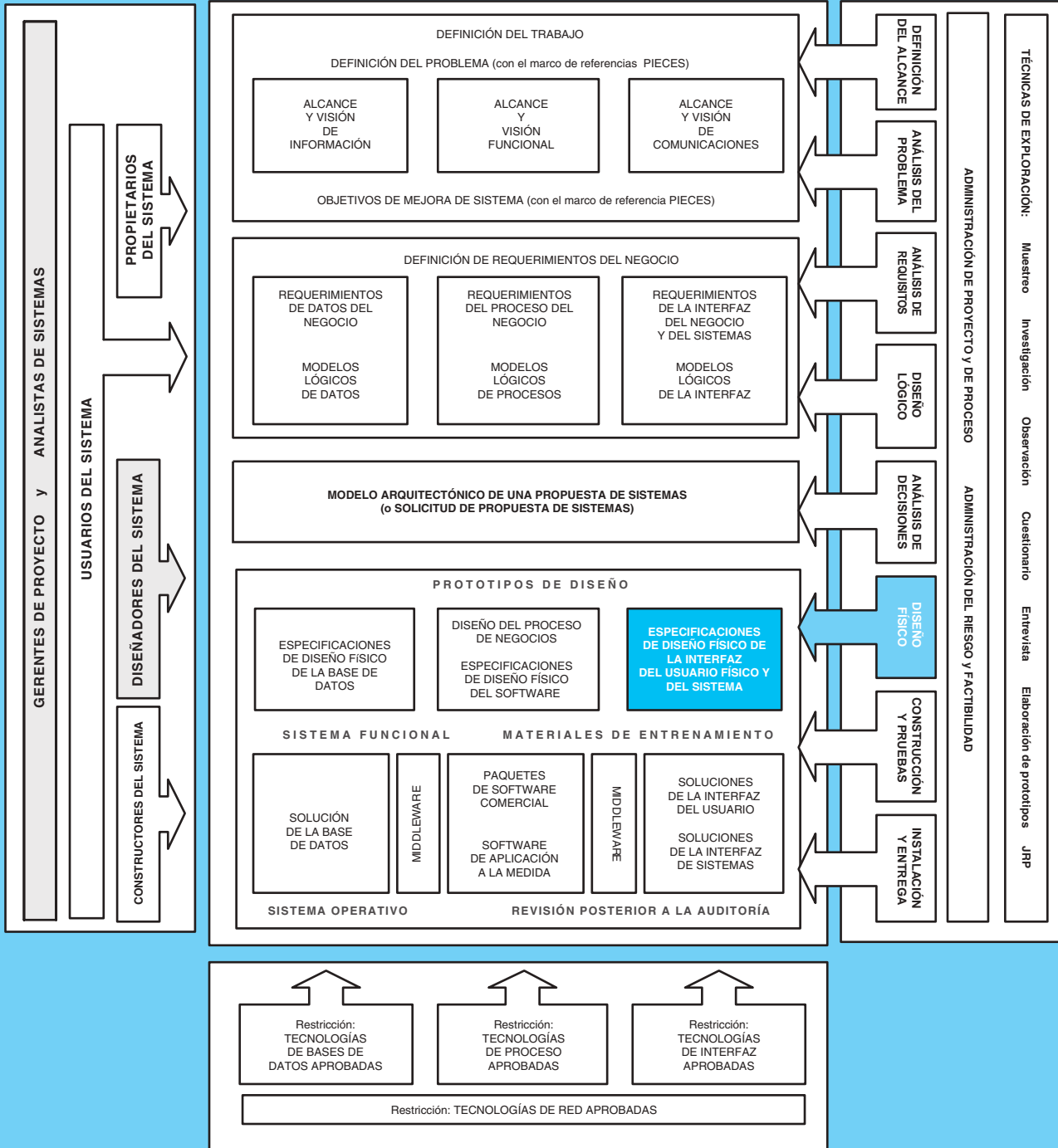
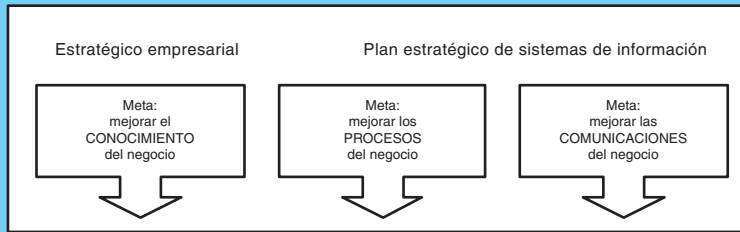
*Application Development Strategies* (periódico mensual). Arlington, VA: Cutter Information Corporation. Éste es nuestro periódico favorito orientado por temas que sigue las estrategias, metodologías, CASE, y otras tendencias importantes del

desarrollo de sistemas. Cada aspecto se centra en un tema individual. Este periódico suministrará un buen cimientto sobre cómo desarrollar los prototipos.

Galitz, W. O. *User-Interface Screen Design*. Nueva York: John Wiley & Sons, 1993. Éste es nuestro libro favorito sobre el diseño de las interfaces.

Shelly, G., T. Cashman, y H. Rosenblatt. *Systems Analysis and Design*, 3a. ed. Cambridge, MA: Course Technology, 1998. Mencionamos a nuestros competidores por su excelente cobertura del diseño de salidas impresas tabulares. Ellos ofrecen muchas más páginas de cobertura y de ejemplos que lo que nosotros pudimos en nuestra más reciente edición.





# 14

## Diseño de entradas y elaboración de prototipos

### Panorámica y objetivos del capítulo

En este capítulo usted aprenderá a diseñar las entradas de computadora. Es el segundo de tres capítulos que se enfocan en el diseño de los sistemas en línea usando una interfaz gráfica de usuario ya sea para sistemas de cliente/servidor o sistemas basados en la Red. Sabrá cómo diseñar entradas cuando usted pueda:

- Definir el formato y los medios apropiados para una entrada de computadora.
- Explicar la diferencia entre la captura de datos, el ingreso de datos, y la entrada de datos.
- Identificar y describir varias tecnologías automáticas de obtención de datos.
- Aplicar ergonomía al diseño de las entradas de computadora.
- Diseñar los controles internos para las entradas de computadora.
- Seleccionar controles apropiados basados en pantallas para los atributos de entrada que deben aparecer en una pantalla GUI de entrada.
- Diseñar una interfaz para entradas basada en la Red.

El diseño de salidas y entradas se parece un poco al problema de la secuencia del “huevo o la gallina”. ¿Qué es lo que usted va a hacer primero? En esta edición, presentamos primero el diseño de las salidas. El diseño clásico de sistemas prefiere este enfoque como parte de una prueba de validación del sistema; diseñe las salidas y luego asegúrese de que las entradas son suficientes para producir las salidas. En la práctica, esta secuencia de tareas se hace menos importante porque las técnicas del análisis de sistemas modernas predefinen sobradamente los requerimientos lógicos de las entradas y las salidas. Usted y su instructor pueden intercambiar sin problema el orden de los capítulos 13 y 14 si así lo prefieren.

## Introducción

Bob Martínez ha recibido la tarea de hacer el prototipo de la pantalla Web de entrada de órdenes de los miembros para el proyecto de sistemas SoundStage Member Services (usted puede ver su trabajo en las figuras 14.11 y 14.12). Ésta fue claramente la parte más divertida del proyecto de sistema para Bob. Una firma consultora de diseño gráfico había creado la apariencia y percepción (look and feel) generales. Pero dependía de Bob crear el prototipo real.

Él decidió usar *Visual Studio .NET* solamente porque él se sentía muy cómodo con su diseñador de la Interfaz Gráfica del Usuario (GUI). Con bastante rapidez, Bob pudo poner conjuntamente las páginas Web a fin de que tuvieran la apariencia de un carrito de compras verdadero. Por supuesto, no tenían ningún código de programación. Independientemente de lo que usted buscara, usted obtenía *The Matrix* (la película favorita de Bob), y el cuadro de datos al pie de la figura 14.11 había sido rellenado a mano, no por la base de datos. Pero la disposición incluyó todos los requerimientos que previamente habían sido identificados. Y todos los enlaces funcionaron, aunque todos ellos fueron a una página “En construcción”.

Tanto los empleados de SoundStage como los miembros seleccionados usaron las páginas Web. Habiendo obtenido una retroalimentación generalmente favorable, ahora Sandra le ha encargado a Bob refinar y expandir el prototipo. Luego regresará a los usuarios para hacer más pruebas, y obtener más refinamiento, hasta que tengan un diseño Web que realmente pueda programarse.

## Conceptos y lineamientos de diseño de entradas

“¡Basura entra! ¡Basura sale!” (del inglés Garbage in! Garbage out! *Nota del R.T.*). Esta gastada expresión es no menos cierta hoy que lo fue cuando primero aprendimos a programar una computadora. La gerencia y los usuarios toman decisiones importantes basadas en las salidas de los sistemas (capítulo 13). Estas salidas se producen de datos que son entradas o que se recuperan de las bases de datos. Y todos los datos en las bases de datos tuvieron que haber sido entradas. En este capítulo, usted va a aprender a diseñar las entradas de datos a la computadora. El diseño de las entradas sirve a un objetivo importante: capturar los datos y ponerlos en un formato adecuado para la computadora.

Hoy la mayoría de las entradas son diseñadas por prototipos que se construyen rápidamente. Estos prototipos pueden ser imitaciones simples generadas por computadora, o generarse de estructuras de la base de datos del prototipo como las desarrolladas con Microsoft *Access*. Estos prototipos rara vez son totalmente funcionales. No contendrán las características de seguridad, edición de datos, o actualización de datos que serán necesarias en la versión final de un sistema. Además, en interés de la productividad, no pueden incluir todas las características de cada botón o control que tendrían que incluirse en un sistema en producción.

Durante el análisis de requerimientos, las entradas se modelaban como flujos de datos que constan de atributos de datos. Aun en los más completos análisis de requerimientos, perderemos requerimientos. El diseño de las entradas puede introducir campos o atributos nuevos al sistema. ¡Esto es especialmente cierto si el diseño de salidas introdujo atributos nuevos a las salidas: las entradas siempre deben ser suficientes para producir las salidas!

Comenzamos con la presentación de los tipos de entradas. Las entradas pueden clasificarse según dos características: 1) cómo los datos son inicialmente capturados, ingresados, y procesados, y 2) el método y la tecnología usados para capturar e ingresar los datos. La figura 14.1 ilustra esta taxonomía. Las características se discuten brevemente en las siguientes secciones.

### > Captura de datos, entrada de datos y procesamiento de datos

Cuando usted piensa en las “entradas”, generalmente piensa acerca de dispositivos de entrada, como teclados y ratones. Pero la entrada comienza mucho antes de que los datos lleguen al dispositivo. Para realmente introducir en la computadora datos de negocios, el analista de sistemas puede tener que diseñar documentos fuente, pantallas de entradas, y métodos y procedimientos para ingresar los datos en la computadora (del cliente al formato al capturista a la computadora).

**FIGURA 14.1** Una taxonomía de las entradas

Método de proceso	Captura de datos	Entrada de datos	Procesamiento de datos
Teclado	Generalmente los datos se capturan en una forma de negocios que se convierte en el documento fuente para la entrada. Los datos pueden recolectarse en tiempo real (por teléfono).	Los datos se ingresan vía el teclado. Éste es el método de ingreso más común pero también el más susceptible de errores.	ANTERIOR: Los datos pueden recolectarse en archivos de lotes (disco) para su procesamiento como lote. NUEVO: Los datos se procesan tan pronto como han sido tecleados.
Ratón	Igual que el anterior.	Se usa conjuntamente con el teclado para simplificar la entrada de datos. El ratón sirve como un dispositivo apuntador para la pantalla. Puede usarse con interfaces de usuario geográfico para reducir errores a través de selecciones de apuntar y hacer clic.	Lo mismo que arriba, pero el uso del ratón se asocia muy comúnmente con el procesamiento en línea y en tiempo real.
Pantalla táctil	Igual que el anterior.	Los datos se ingresan en un exhibidor con pantalla táctil o en un dispositivo manejado manualmente. Los usuarios de la entrada de datos activan comandos, y seleccionan o ingresan datos usando el reconocimiento de la escritura.	En PC, las selecciones de pantalla táctil se procesan igual que arriba. En las computadoras de mano, los datos se almacenan en la misma para su procesamiento posterior como lote remoto.
Punto de venta	Los datos se capturan tan cerca como sea humanamente posible del punto de venta (o transacción). No hay documentos fuente.	Frecuentemente el cliente ingresa los datos directamente (por ejemplo, el cajero automático) o por un empleado que interactúa directamente con el cliente (por ejemplo, un cajero). El ingreso requiere terminales especializadas y reservadas que utilicen alguna combinación de las otras técnicas de esta tabla.	Los datos casi siempre se procesan inmediatamente como una transacción o petición.
Sonido	Los datos se capturan tan cerca como sea posible de la fuente, aun cuando el cliente esté ubicado remotamente (por ejemplo, en su casa o en su lugar de trabajo).	Los datos se ingresan mediante el uso de tonos (comúnmente de un teléfono). Generalmente se requiere una estructura de menú bastante rígida y opciones limitadas de entrada.	Los datos casi siempre se procesan inmediatamente como una transacción o petición.
Voz	Igual que el sonido.	Los datos (y los comandos) son hablados. Esta tecnología no es tan madura y es mucho menos confiable y común que otras técnicas.	Los datos casi siempre se procesan inmediatamente como una transacción o petición.
Marca óptica	Los datos se registran en hojas de escaneado óptico como marcas o letras, números y puntuación de formación precisa. Ésta es la forma más antigua de captura automática de datos.	Elimina la necesidad de la entrada de datos. (Se usa muy comúnmente en la educación para calificar pruebas, evaluaciones de cursos, y encuestas.)	Los datos casi siempre se procesan como lote.
Tinta magnética	Generalmente los datos se prerregistran en formatos que el cliente llena posteriormente. El cliente registra datos adicionales en el formato.	Un lector de tinta óptica lee los datos magnetizados. Los datos agregados por el cliente deben ser ingresados mediante el uso de otro método de ingreso. Esta técnica se usa en aplicaciones que requieren alta exactitud y seguridad, siendo lo más común los cheques de bancos (para el número del cheque, el número de cuenta, la identificación del banco).	Los datos casi siempre se procesan como lote.
Electromagnético	Los datos se registran directamente en el objeto que debe ser descrito por los datos.	Los datos se transmiten por frecuencia de radio.	Los datos casi siempre se procesan inmediatamente.
Tarjeta inteligente	Los datos se registran directamente en un dispositivo que debe ser portado por el cliente, el empleado, u otra persona que sea descrita por esos datos.	Los datos son leídos por lectoras de tarjetas inteligentes.	Los datos casi siempre se procesan inmediatamente.
Biométrica	Las características humanas únicas se convierten en datos.	Los datos son leídos por sensores biométricos. Las aplicaciones primarias son el monitoreo de seguridad y médico.	Los datos se procesan inmediatamente.

Esto nos lleva a nuestra primera pregunta fundamental. ¿Cuál es la diferencia entre la captura de datos y la entrada de datos? ¡Los datos ocurren! Acompañan a los eventos de negocios llamados *transacciones*. Los ejemplos incluyen ÓRDENES, TARJETAS DE TIEMPO, RESERVACIONES, y cosas por el estilo. Debemos determinar cuándo y cómo capturar los datos cuando estos “ocurren”.

La **captura de datos** es la identificación y adquisición de información nueva. ¡*Cuando* es fácil! Es siempre mejor capturar los datos tan pronto como sea posible después de que se originan. ¡*Cómo* es otra historia! Históricamente, se usaron formas especiales de papel llamadas *documentos fuente*. Los **documentos fuente** son formas usadas para registrar transacciones de negocios en términos de datos que describen esas transacciones.

Las pantallas que pueden duplicar la apariencia de casi cualquier formato de papel están reemplazando gradualmente a los formatos de papel. Esta tendencia está siendo acelerada por el comercio electrónico basado en la Web y los negocios electrónicos. Aun así, las formas comerciales son comúnmente utilizadas como documentos fuente para la entrada de datos. El diseño de los documentos fuente requiere cautela. La disposición y la legibilidad de los datos afectarán la velocidad de entrada.

El **ingreso de datos** es el proceso de traducir los datos o documentos fuente a un formato legible en la computadora. Como el ingreso de datos solía basarse cien por ciento en los teclados, los negocios empleaban ejércitos de capturistas. A medida que se hacía más común la computación en línea, la responsabilidad del ingreso de datos se desviaba directamente a los usuarios del sistema. Hoy otra transformación ocurre. Gracias a las computadoras personales y a Internet, algunos ingresos de datos se han desviado directamente al consumidor. En todos los casos, captura e ingreso de datos produce las entradas para el procesamiento de los mismos.

Los datos ingresados deben procesarse subsiguientemente: el procesamiento de datos. En este capítulo, no estamos interesados en cómo los datos son transformados en salidas. En cambio, estamos interesados en el momento del procesamiento de las entradas. ¿Cuándo es que los datos ingresados se procesan?

**El procesamiento por lotes** Ésta solía ser la forma predominante de procesamiento de datos. En el **procesamiento por lotes**, los datos ingresados son recolectados en archivos llamados *lotes*. Cada archivo es procesado como un lote de muchas transacciones. En contra de la creencia popular, algunos datos todavía son tratados en lotes. Las tarjetas de tiempo son el ejemplo clásico. La mayoría de los lotes se graban como archivos de discos (de ahí el término *tecla a disco*). Algunos sistemas más antiguos todavía graban los lotes en cinta magnética (*tecla a cinta*).

**El procesamiento en línea** Hoy la mayoría de los sistemas de información (pero no todos) han sido convertidos a procesamiento en línea. En el **procesamiento en línea**, los datos capturados se procesan inmediatamente. Anteriormente, los datos se ingresaban en terminales. Hoy, esa misma información es capturada en PC y estaciones de trabajo para aprovechar su habilidad para realizar una cierta cantidad de la validación de datos y editar antes de que se envíen a los servidores de aplicaciones y datos. Debido a las PC, raramente oímos actualmente el término *procesamiento en línea*. Generalmente oímos el término *cliente/servidor*, donde la PC es el cliente.

La mayoría de las aplicaciones actuales le presentan al usuario una *interfaz gráfica de usuario (GUI)* basada en PC. Microsoft *Windows* es la GUI dominante en los negocios de hoy. Pero el surgimiento de la Web como una plataforma para aplicaciones de Internet y de la intranet puede hacer de un navegador de Internet la interfaz de usuario más importante en el futuro. Microsoft *Internet Explorer* y Mozilla *Firefox* son los navegadores dominantes en el mercado de hoy. Este capítulo tratará las técnicas de diseño de entradas para la interfaz de cliente/servidor de *Windows* y la interfaz del navegador.

**El lote remoto** El lote y el proceso en línea representan los extremos en el espectro de procesamiento. También existe una solución combinada: el lote remoto. En el **procesamiento de lotes remotos**, los datos se ingresan usando técnicas de edición en línea; sin embargo, se recolectan en un lote en lugar de procesarse inmediatamente. Más tarde, el lote se procesa.

El lote remoto moderno puede tomar varias formas. Un ejemplo simple usa una aplicación (front-end) basada en PC para capturar y almacenar los datos. Más tarde, estos pueden ser transmitidos a través de una red para el proceso por lotes. Un ejemplo más contemporáneo del procesamiento de lotes remotos usa una computadora portátil (laptop)

---

**captura de datos** Identificación y adquisición de datos nuevos.

---

**documento fuente** Un formato usado para registrar datos acerca de una transacción.

---

**ingreso de datos** Proceso de trasladar datos en un formato entendible para la computadora.

---

**procesamiento por lotes** Método de procesamiento de datos mediante el cual se colectan datos de muchas transacciones como un archivo individual que luego se procesa.

---

**procesamiento en línea** Método de procesamiento de datos mediante el cual se procesan inmediatamente los datos de una transacción individual.

---

**procesamiento de lotes remoto** Método de procesamiento de datos mediante el cual se ingresan los datos en línea, se colectan en un lote y se procesan posteriormente.

desconectada o computadoras de bolsillo (o dispositivos PDA) para recolectar datos para su posterior procesamiento. Si usted recientemente ha recibido un paquete de UPS o de Federal Express, ha visto tales dispositivos usados por los repartidores para registrar los envíos y las entregas.

Ahora que hemos cubierto la captura de datos básica, el ingreso de datos, y las técnicas de procesamiento de datos, podemos examinar más estrechamente los métodos de entrada mostrados como filas en la figura 14.1.

## > Métodos de entrada y su implementación

Los diferentes dispositivos de entrada, como teclados y ratones, se estudian en la mayoría de cursos introductorios de sistemas de información. En esta sección, estamos más interesados en el método y su implementación que en la tecnología. En particular, nos interesa cómo afecta la elección de un método a la captura de datos, el ingreso, y el procesamiento tal como se describen en la sección anterior. Usted debería continuar estudiando la figura 14.1 a medida que presentamos estos métodos.

**El teclado** La entrada de datos por el teclado sigue siendo la forma más común de ingreso. Desafortunadamente, es la que más requiere de la edición de datos porque las personas cometen errores tecleando datos de los documentos fuente. Por fortuna, actualmente las interfaces gráficas del usuario tales como *Windows* de Microsoft y los navegadores de Internet posibilitan el diseño de pantallas en línea que reducen errores imponiendo elecciones correctas en el usuario. Exploraremos varios controles útiles de la Interfaz Gráfica del Usuario para tales interfaces en la siguiente sección.

**El ratón** Es un dispositivo de indicación usado en conjunción con interfaces de usuario gráficas. El ratón ha facilitado navegar formas en línea y dar un clic sobre comandos y opciones de entrada. Por ejemplo, los valores legítimos para un atributo pueden estar registrados en una pantalla como cajas o botones para hacer “clic”, con estos se elimina la necesidad de teclear datos. Dando como resultado menos errores en la entrada de información. Exploraremos controles basados en ratones en nuestros diseños de entrada para este capítulo.

**La pantalla táctil** Una tecnología emergente que afectará de manera importante el diseño de las entradas en un futuro cercano es el despliegue de la pantalla táctil. Tales despliegues son comunes en computadoras de bolsillo y de palma de la mano que son utilizadas en incontables aplicaciones de sistemas de información. La imagen muestra una computadora de mano Symbol Technologies basada en el *Palm Operating System (Sistema Operativo de la palma de la mano)*. Tales dispositivos simplifican muchas actividades de recolección de datos en almacenes y plantas manufactureras. Los botones de la pantalla táctil pueden estar programados para recolectar los datos. La mayoría de los dispositivos de este tipo soportan también el reconocimiento de la escritura manual. El dispositivo de Symbol Technologies también puede escanear y leer los códigos de barras (discutidos en breve).

**El punto de venta** Las terminales de punto de venta (POS) han estado con nosotros durante algún tiempo. Han terminado por reemplazar a las anticuadas cajas registradoras. Estas terminales capturan datos en el punto de venta y proveen formas ahorradoras de tiempo para ingresar los datos, realizar cálculos de las transacciones, y producir alguna salida. Al igual que las computadoras de mano descritas anteriormente, la mayoría puede escanear y leer los códigos de barras para eliminar los errores al teclear datos. Los cajeros automáticos (automatic teller machines, ATM), otra forma de terminal de punto de venta, son operados directamente por el consumidor.

**Sonido y voz** El sonido representa otra forma de entrada. Usted podría haber usado un sistema basado en teléfono de tonos para inscribirse en este curso. Tales sistemas basados en tonos requieren tecnología especial de entrada/salida que impulsa al diseño. Esos sistemas están más allá del alcance de este libro.

Una forma más sofisticada de este método de entrada usa tecnología de reconocimiento de voz para posibilitar el ingreso de los datos. En la actualidad esta tecnología es relativamente inmadura y poco confiable. Proporciona el mejor resultado para ingresar comandos, pero no datos. Sin embargo, llegará el momento en que la tecnología de reconocimiento de voz reemplace el teclado como la manera principal mediante la cual ingresemos los datos.

Los métodos restantes de entrada de datos se clasifican ampliamente como *captura de datos automática (automatic data capture, ADC)*. Con los avances que tenemos actual-



Una computadora de mano

mente con respecto a la tecnología de entradas, podemos eliminar gran parte de la intervención humana (y algunas veces toda) asociada con los métodos de ingreso discutidos en la sección anterior. Al eliminar la intervención humana podemos disminuir los errores y el retraso en tiempo asociados con la interacción humana.

**Reconocimiento óptico de marcas** La tecnología *de reconocimiento óptico de marcas* (*optical mark recognition, OMR*) ha existido durante varias décadas y está orientada principalmente al procesamiento de lotes. El ejemplo clásico son las formas con marcas ópticas usadas para responder preguntas basadas en objetivos (por ejemplo, el examen de opción múltiple) en los exámenes. La tecnología es también útil en encuestas y cuestionarios o cualquier otra aplicación donde el número de valores posibles de datos es relativamente limitado y altamente estructurado. La mayoría de aplicaciones que podrían beneficiarse de este método de entrada probablemente ya la han explotado.

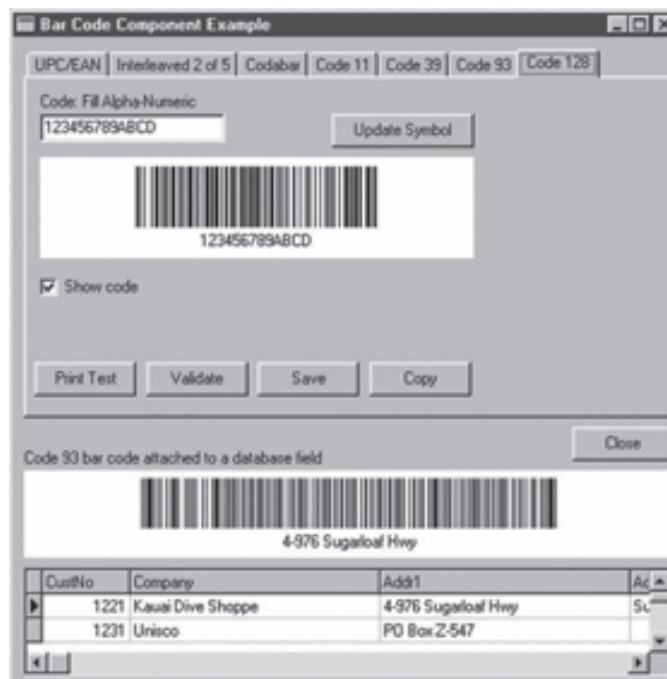
El *reconocimiento óptico de caracteres* (*optical character recognition, OCR*) prevalece menos a pesar de su madurez. Requiere que el usuario o el cliente escriban a mano cuidadosamente los datos de entrada en una forma de negocios. Si las letras y los números están contorneados de manera apropiada, un lector de OCR puede procesar las formas sin intervención humana. Esto depende de la escritura a mano del usuario o el cliente, pero funciona. Columbia House Record Club solía destinar una forma de OCR para las respuestas del cliente a las solicitudes. Al igual que la mayoría de las aplicaciones de OCR, el número de campos para ser llenados era muy pequeño (reduciendo la posibilidad de errores). Se deben implementar métodos de procesamiento para cualquier entrada de datos rechazada debido a la ilegibilidad.

Hoy la forma más prevaleciente de tecnología óptica involucra el código de barras. Los *códigos de barras* están en casi cada producto que compramos, pero la tecnología de código de barras no está limitada a las ventas al menudeo. Usted puede crear códigos de barras para casi cualquier aplicación comercial. Incluso puede integrar códigos de barras en aplicaciones basadas en *Windows*, como se muestra en la figura 14.2.

**Tinta magnética** La tecnología ADC de tinta magnética es una que usted probablemente reconocerá. En general consiste en usar tarjetas con banda magnética, pero también puede incluir el uso de reconocimiento de caracteres en tinta magnética (*magnetic ink character recognition, MICR*). ¡Actualmente están en uso en más de 1 billón de tarjetas con banda magnética! Asimismo, se les ha utilizado en varias aplicaciones comerciales, tales como transacciones con tarjetas de crédito, construcción del control de acceso de seguridad, y control de asistencia del empleado. El reconocimiento de caracteres en tinta magnética se usa ampliamente en la industria bancaria.

**FIGURA 14.2**

Códigos de barra en una aplicación de *Windows*





**Transmisión electromagnética** La tecnología electromagnética ADC se basa en el uso de radiofrecuencias para identificar objetos físicos. Esta tecnología consiste en adherir una etiqueta y una antena al objeto físico que va a rastrearse. La etiqueta contiene memoria que se usa para identificar el objeto que se rastrea. La etiqueta puede ser leída por un lector una vez que el objeto reside dentro del campo electromagnético generado por el lector. Esta tecnología de identificación se está haciendo muy popular en aplicaciones que incluyen el rastreo de objetos físicos que están fuera de la visión y en movimiento. Por ejemplo, el ADC electromagnético se está usando para el rastreo y control del transporte público, el rastreo de productos elaborados, y el rastreo de animales, entre otros.

**Las tarjetas inteligentes** La tecnología de la tarjeta inteligente tiene la capacidad de almacenar una cantidad masiva de información. Las tarjetas inteligentes son similares, aunque ligeramente más gruesas que las tarjetas de crédito. También difieren en que contienen un microprocesador, circuitos de memoria, y una batería. Considérela como una tarjeta de crédito con una computadora a bordo. Representan un medio de almacenamiento portátil del cual pueden obtenerse datos de entrada. Mientras que esta tecnología apenas comienza a incursionar en Estados Unidos, las tarjetas inteligentes son usadas diariamente por más del 60 por ciento de la población francesa. Las aplicaciones de la tarjeta inteligente son especialmente importantes en el área de registros de salud, donde el tipo de sangre de una persona, vacunaciones, y otra historia médica pasada pueden hacerse fácilmente disponibles. Otros usos pueden incluir aplicaciones tales como pasaportes, información financiera para transacciones de punto de venta, y la televisión de paga, por nombrar unos cuantos. Otra aplicación futura podría ser una tarjeta de débito que automáticamente mantenga y exhiba el saldo de su cuenta. En la imagen se muestra una tarjeta inteligente usada en una aplicación.



Una tarjeta inteligente

**Biométrica** La tecnología biométrica ADC se basa en características o rasgos humanos únicos. Por ejemplo, un individuo puede ser identificado por sus huellas digitales únicas, su patrón de voz, o el patrón de ciertas venas (la retina o la muñeca). Los sistemas biométricos ADC constan de sensores que capturan la característica de un individuo o rasgo, digitalizan el patrón de imagen, y luego comparan la imagen con patrones almacenados para identificación. El ADC biométrico es popular porque le ofrece la manera más precisa y fidedigna para identificación. Esta tecnología es especialmente popular para sistemas que requieren acceso de seguridad.

## > Temas de usuario del sistema para diseño de entradas

Ya que las entradas se originan con los usuarios del sistema, los factores humanos juegan un papel significativo en el diseño de estas. Las entradas deben ser tan simples como sea posible y diseñarse para reducir la posibilidad de que se ingresen datos incorrectos. *Deben* considerarse las necesidades de los usuarios del sistema. Con esto en mente, deberán evaluarse varios factores humanos.

El volumen de datos a ingresar deberá ser minimizado. Entre más datos se ingresen, es mayor el número potencial de errores de entrada y toma más tiempo ingresar esos datos. Por tal motivo, deberán contemplarse numerosas consideraciones con los datos que se capturan para el ingreso. Deberán seguirse los siguientes principios generales para el diseño de las entradas:

- *Capture sólo datos variables.* No ingrese datos constantes. Por ejemplo, al decidir qué elementos deben incluirse en una entrada de ORDEN DE VENTAS, necesitamos los NÚMEROS DE PARTES para todas las partes ordenadas. Sin embargo, ¿necesitamos ingresar las DESCRIPCIONES DE LAS PARTES para esas partes? La DESCRIPCIÓN DE LAS PARTES probablemente se almacena en una tabla de base de datos. Si ingresamos el NÚMERO DE PARTE, podemos buscar la DESCRIPCIÓN DE LA PARTE. Los datos permanentes (o semipermanentes) deberán guardarse en la base de datos. Por supuesto, las entradas deben diseñarse para mantener esas tablas de base de datos.
- *No capture datos que pueden calcularse o almacenarse en programas de computadora.* Por ejemplo, si usted ingresa la CANTIDAD ORDENADA y el PRECIO, no necesita ingresar el PRECIO EXTENDIDO, que es igual a la CANTIDAD ORDENADA  $\times$  PRECIO. Otro ejemplo es la incorporación de los datos de la RETENCIÓN DEL GRAVAMEN FEDERAL en tablas (arreglos) en lugar de teclear esos datos todo el tiempo.
- *Use códigos para los atributos apropiados.* Los códigos fueron introducidos anteriormente; pueden ser traducidos en programas de computadora usando tablas.

Si se usan documentos fuente para capturar datos, deberá ser fácil que los usuarios del sistema los llenen y los ingresen posteriormente en el sistema. Las siguientes sugerencias pueden ayudar:

- *Incluya instrucciones para llenar la forma.* Recuerde que a las personas no les gusta tener que leer instrucciones impresas en el reverso de una forma.
- *Minimice la cantidad de escritura a mano.* Muchas personas adolecen de mala caligrafía. El capturista o el operador del tubo de rayos catódicos puede leer mal los datos e ingresar datos incorrectos. Use cuadros de verificación siempre que le sea posible a fin de que el usuario del sistema sólo necesite comprobar los valores apropiados.
- *Los datos a ingresarse (teclarse) deberán estar secuenciados a fin de que puedan leerse como este libro, de arriba hacia abajo y de izquierda a derecha.* La figura 14.3a) demuestra un buen flujo. El usuario del sistema no deberá tener que moverse de derecha a izquierda en una línea o brincar por todos lados en la forma, como se muestra en la figura 14.3b), para ingresar los datos.

**FIGURA 14.3**

Flujo eficiente e ineficiente en un formato

**Good Entry Layout**

Applicant Information:

Social Security #: Salutation: Current Date:

First Name: Middle Name: Last Name:

Address Line 1: Telephone: Other Information:

Address Line 2:

City: State: Zip Code:

**a) FLUJO BUENO**

**Bad Entry Layout**

Applicant Information:

Social Security #: Salutation: Current Date: Other Information:

First Name: Last Name: State:

Middle Name: Telephone: Zip Code:

City: Address Line 1: Address Line 2:

**b) FLUJO MALO**

FIGURA 14.4

Diseño metafórico de una pantalla

- Cuando sea posible, use diseños basados en metáforas conocidas. El ejemplo clásico de esto es la aplicación personal de finanzas *Quicken*. La facilidad de uso del programa es realizada por su recreación sobre la pantalla de la metáfora de la chequera. El usuario extiende cheques llenando una representación gráfica del cheque. Y el registro del cheque se ve exactamente como su equivalente de papel. No todas las entradas se prestan a las metáforas, pero algunas son grandemente realizadas por la imitación (vea la figura 14.4).
- Hay otros lineamientos y aspectos específicos de la entrada de datos para los diseños de las pantallas de la interfaz gráfica del usuario. Introduciremos estos lineamientos, como sea apropiado, cuando estudiemos los controles de la interfaz gráfica del usuario para el diseño de entradas posteriormente en este capítulo, así como en los capítulos sobre el diseño de salidas y el diseño de la interfaz del usuario.

### > Controles internos: Edición de datos para entradas

Los controles internos son un requerimiento en todos los sistemas basados en la computadora. Los controles internos de las entradas aseguran que el ingreso de datos para la computadora sea preciso y que el sistema esté protegido en contra de errores accidentales e intencionales, así como el abuso, incluyendo el fraude. Se ofrecen los siguientes lineamientos de control interno:

1. El número de entradas deberá ser monitoreado. Esto es especialmente cierto con el método de los lotes, porque los documentos fuente pueden ser extraviados, perdidos, o saltados.
  - En los sistemas por lotes, los datos acerca de cada lote deberán registrarse en un talón de control de lotes. Los datos incluyen el NÚMERO DE LOTES, el NÚMERO DE DOCUMENTOS, y los TOTALES DE CONTROL (por ejemplo, el número total de elementos de línea en los documentos). Estos totales pueden ser comparados con los totales de salida en un reporte después de que el procesamiento haya sido completado. Si los totales no son iguales, debe determinarse la causa de la discrepancia.
  - En los sistemas por lotes, un control alternativo serían los cheques uno por uno. Cada documento fuente se cotejaría con la línea de detalle del reporte histórico correspondiente que confirme que el documento ha sido procesado. Esta verificación de control puede ser menester sólo cuando los totales de control de los lotes no concuerdan.
  - En los sistemas en línea, cada transacción de entrada deberá ser puesta en bitácora para un archivo separado de auditoría a fin de que pueda recuperarse y reprocesarse si hay un error de procesamiento o si los datos se pierden.

2. Deberá tenerse cuidado en asegurar que los datos son válidos. Dos tipos de errores pueden infiltrar los datos: los errores de entrada de datos y los datos inválidos registrados por usuarios del sistema. Los errores de entrada de datos incluyen los errores de copia, las transposiciones (teclear 132 como 123), y las omisiones (teclear 345.36 como 3453.6). Las siguientes técnicas se usan ampliamente para validar datos:
- *Las verificaciones de existencia* determinan si realmente han sido llenados todos los campos requeridos en la entrada. Los campos requeridos deberán estar claramente identificados como tales en la pantalla de entrada.
  - *Las verificaciones del tipo de datos* comprueban que se ingrese el tipo correcto de datos. Por ejemplo, los datos alfabéticos no deberán permitirse en un campo numérico.
  - *Las verificaciones de dominio* determinan si los datos de entrada para cada campo se sitúan dentro del conjunto válido o el rango de valores definido para ese campo. Por ejemplo, puede ponerse un rango de límite superior a la TARIFA DE PAGO para asegurar que ningún empleado perciba una tarifa mayor.
  - *Las verificaciones de combinación* determinan si es válida una relación conocida entre dos campos. Por ejemplo, si la MARCA DEL VEHÍCULO es Pontiac, entonces el MODELO DEL VEHÍCULO debe ser uno de un conjunto limitado de valores que comprende autos manufacturados por Pontiac (Firebird, Grand Prix, y Bonneville, por nombrar unos cuantos).
  - *Los dígitos autoverificadores* determinan errores de ingreso de datos en claves primarias. Un *dígito de comprobación* es un número o carácter que se anexa a un campo de clave primario. El dígito de comprobación se calcula aplicando una fórmula, como el Módulo 11, para la clave real. El dígito de comprobación verifica la correcta entrada de datos en una de las dos formas siguientes. Algunos dispositivos de entrada de datos automáticamente pueden validar datos aplicándoles la misma fórmula a los datos tal como se ingresaron por el usuario de sistema. Si el dígito de comprobación introducido no corresponde al dígito de comprobación calculado, se notifica un error. Alternativamente, los programas de computadora también pueden validar los dígitos de comprobación usando subrutinas fácilmente disponibles.
  - *Las verificaciones de formato* comparan los datos introducidos con los requerimientos conocidos de formateo para esos datos. Por ejemplo, algunos campos pueden requerir ceros iniciales, mientras que otros no. Algunos campos usan puntuación estándar (por ejemplo, los números de seguro social o los de teléfono). Un valor "A4898 DH" podría pasar una verificación de formato, mientras un valor similar "A489 ID8" no lo haría.

En el capítulo 12 usted se enteró de que la mayoría de los sistemas de administración de bases de datos realizan verificaciones de validación de datos parecidas a los descritos en la lista de arriba. Entonces, ¿por qué necesitamos controles de entrada? ¡Simple! La mayoría de las aplicaciones actuales están conectadas en red. Los datos erróneos son tanto un cuello de botella de tráfico de la red como un enemigo para el procesamiento de transacción y el tiempo de respuesta. Siempre es mejor capturar y corregir los errores de ingreso tan cercanamente como sea posible de la fuente; de ahí el énfasis en los controles de ingreso y la validación.

## Controles GUI para diseño de entradas

Como se mencionó anteriormente, la mayoría de las nuevas aplicaciones desarrolladas actualmente incluyen una interfaz gráfica del usuario (GUI). La mayoría se basa en Microsoft *Windows*, pero la creciente adopción de Internet, combinada con el comercio electrónico basado en la Web, está impulsando rápidamente algunas interfaces al navegador de Internet. Aun cuando los diseños de la interfaz gráfica de usuario proveen una interfaz más amigable, también presentan aspectos de diseño más complejos que sus predecesores. Este capítulo no intentará encarar todos los asuntos del diseño de la interfaz gráfica del usuario; libros enteros han sido escritos sobre el tema. Varios de nuestros favoritos se listan en las Lecturas recomendadas.

Más bien, este capítulo se centra en seleccionar los controles apropiados basados en pantallas para ingresar datos en una pantalla de la interfaz gráfica del usuario. Considere

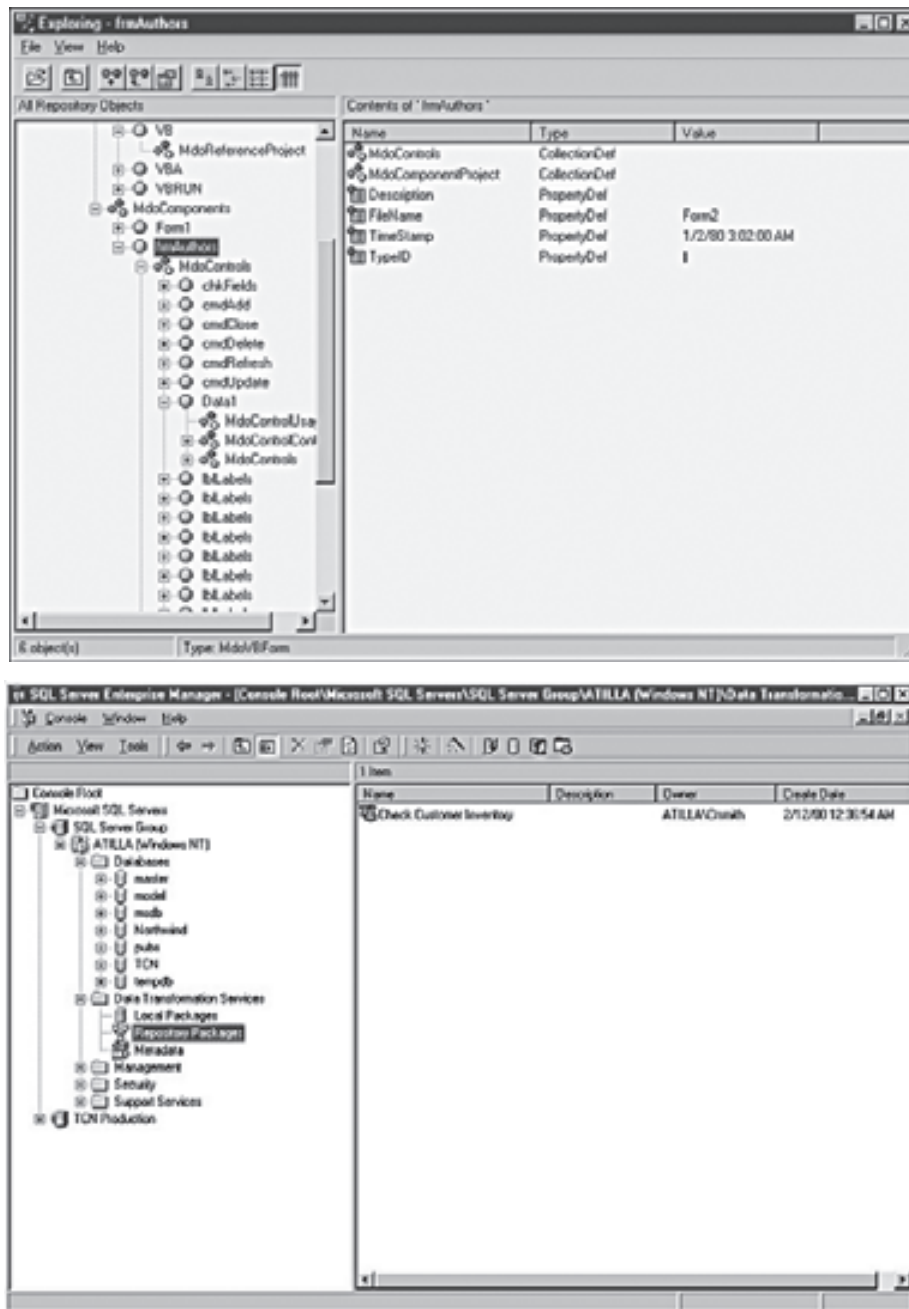


FIGURA 14.5

Elaboración de prototipos y desarrollo basados en un repositorio

los controles como “dispositivos” para construir una interfaz de usuario. Están incluidos en la mayoría de los ambientes de desarrollo de aplicación contemporáneos tales como *Access* de Microsoft y *Visual Studio .NET*, *PowerBuilder* de Sybase, *JBuilder* de InPrise, *Visual Café* de Symantec, *Visual Age* de IBM, y muchos otros. Muchas de estas herramientas comparten controles (y el código) a través del repositorio. A este enfoque se le llama *programación basada en repositorios*.

La figura 14.5 ilustra el acceso a un repositorio que contiene controles de entrada y código. El enfoque se basa en las técnicas de programación orientada a objetos y basadas en componentes que se han vuelto penetrantes en el desarrollo de aplicaciones. Esta figura bosqueja controles que podrían ser usados por programadores o analistas de sistemas diversos para hacer el prototipo de una interfaz. Los desarrolladores pueden definir, en un solo lugar, la mayoría de las propiedades y las restricciones para un campo reusable y el código de validación de datos para ese campo. Una vez definido, el objeto o control puede ser usado por otros analistas de sistemas y programadores en la organización. Este enfoque basado en el repositorio garantiza que cada instancia de campo será usada de

una manera consistente. Además las entradas del repositorio pueden variarse si las reglas del negocio así lo exigen, y no se requerirán cambios adicionales en las aplicaciones.

### > Controles GUI comunes para entradas

Esta sección examina algunos de los controles más comunes usados en formas de entrada basadas en interfaces gráficas de usuario. Presentamos el propósito, las ventajas, las desventajas, y los lineamientos para cada control. Dada esta comprensión, estamos en una buena posición para tomar decisiones concernientes a qué controles deberán considerarse para cada atributo de datos que se ingrese en nuestras pantallas. Diferiremos las transiciones entre nuestros diseños de pantalla hasta el capítulo 15, “Diseño de la interfaz del usuario”.

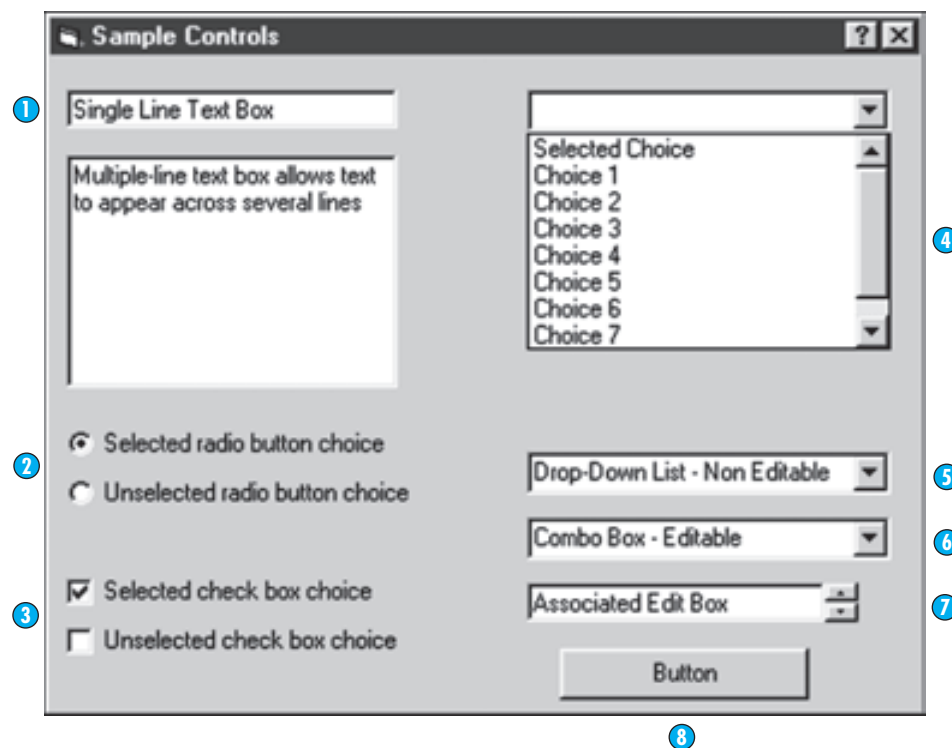
Refiérase a la figura 14.6 como una biblioteca de los controles más comunes basados en pantallas para los datos de entrada. Cada uno de los controles se discutirá. Son igualmente aplicables a las interfaces de Windows y basadas en la Web.

**Text box 1** Quizá el control más común destinado al ingreso de datos es el text box. Un *Text Box* consta de una caja de forma rectangular que está usualmente acompañada por un título. Este control requiere que el usuario teclee los datos dentro de la caja. Un Text Box puede permitir el ingreso de líneas individuales o múltiples de caracteres de datos que van a capturarse. Cuando un text box contiene líneas múltiples de datos, normalmente también se incluyen características de desplazamiento en pantalla.

Un text box se usa más apropiadamente cuando los valores de datos de entrada son ilimitados en el alcance y el analista es incapaz de proveer a los usuarios con una lista significativa de valores de los cuales pueden hacer una selección. Por ejemplo, un text box con una línea individual sería un control apropiado para capturar el apellido de un nuevo cliente porque las posibilidades para el apellido del cliente son virtualmente imposibles de predeterminedir. Un text box también sería apropiado para capturar datos acerca de las instrucciones de embarque que describen una orden particular colocada por un cliente. De nuevo, los valores posibles para las instrucciones de embarque son virtualmente ilimitados. Además, el text box con líneas múltiples sería adecuado debido a la longitud imprevisible de las instrucciones de embarque. En los casos donde el text box no es bastante grande para ver los valores enteros de datos de entrada, el text box puede usar las características de desplazamiento en pantalla y de formateo de palabras.

**FIGURA 14.6**

Controles comunes de entrada de GUI



Deberán seguirse numerosos lineamientos al usar un text box en una pantalla de entradas. Un text box deberá estar acompañado por un título descriptivo, significativo. Evite usar abreviaturas para los subtítulos. Sólo el primer carácter del texto del título deberá ser puesto en mayúscula.

La posición del título es también importante. El usuario deberá poder asociar claramente el título con el text box. Por consiguiente, el título deberá estar ubicado a la izquierda del text box o alineado a la izquierda inmediatamente por encima del text box. Finalmente, en general es también aceptado que el título sea seguido por dos puntos para ayudar al usuario a distinguir visualmente el título de la caja.

Generalmente, el tamaño del text box deberá ser bastante grande para que todos los caracteres de los datos de entrada de longitud fija sean ingresados y vistos por el usuario. Cuando la longitud de los datos que van a ingresarse es variable y pudiera volverse muy larga, deberán aplicarse las características del desplazamiento en pantalla y formateo de palabras para el text box.

**Radio Button 2** Los radio buttons proveen al usuario de una forma fácil para identificar y seleccionar rápidamente un valor particular de un conjunto de valores. Un *radio button* consta de un pequeño círculo y de una descripción textual asociada que corresponde a la elección de valor. El círculo está ubicado a la izquierda de la descripción textual de la elección de valor. Los radio buttons normalmente aparecen en grupos: un radio button por cada elección de valor. Cuando un usuario hace una selección apropiada del conjunto de valores, el círculo correspondiente a esa selección está parcialmente lleno para indicar que ha sido seleccionado. Cuando una elección es seleccionada, se deselecta cualquier elección por omisión o cualquiera previamente seleccionada. Los radio buttons también le proporcionan al usuario la flexibilidad de hacer una selección con el teclado o el ratón.

Los radio buttons son lo más apropiado cuando puede esperarse que un usuario ingrese datos que tengan un conjunto predefinido limitado de valores mutuamente excluyentes. Por ejemplo, puede pedirse a un usuario que ingrese un TIPO DE ORDEN y un SEXO. Cada uno de estos tiene un conjunto limitado, predefinido, mutuamente excluyente de valores válidos. Por ejemplo, cuando los usuarios deben ingresar un TIPO DE ORDEN, podría esperarse que indiquen uno y solamente un valor del conjunto de valores “orden normal”, “pedido urgente”, u “orden pendiente”. Para el SEXO, se esperaría que el usuario indicara uno y solamente un valor del conjunto “femenino”, “masculino”, o “desconocido”.

Hay varios lineamientos para considerar cuándo usar radio buttons como una manera de ingresar datos. Primero, los radio buttons deberán presentar las alternativas verticalmente alineadas y justificadas a la izquierda para auxiliar al usuario en la navegación. Si es necesario, las elecciones pueden presentarse con alineación horizontal, pero deberá usarse el espaciado adecuado para ayudar a distinguir visualmente las elecciones. También, el grupo de elecciones deberá agruparse visualmente para separarlo de otros controles de ingreso que aparecen en la pantalla. El agrupamiento también debería contener un título significativo apropiado. Por ejemplo, los radio buttons para masculino, femenino, y desconocido podrían estar alineados en sentido vertical y justificados a la izquierda con el título/encabezado “sexo” justificado a la izquierda arriba del conjunto.

La ordenación en secuencia de las elecciones también deberá recibir consideración. Mientras mayor sea el número de elecciones, deberá darse más consideración a la facilidad de escaneado e identificación de las mismas. Por ejemplo, en algunos casos puede ser más natural que el usuario localice elecciones que se presentan en orden alfabético. En otros casos, la frecuencia para la cual se selecciona un valor puede ser importante con relación a dónde se localiza dentro del conjunto de elecciones.

Finalmente, no se recomienda que los radio buttons se usen para seleccionar el valor de los datos de entrada cuyo valor sea simplemente un sí/no (o estado de encendido/apagado). En lugar de eso, deberá considerarse un control de casilla de verificación.

**Check Box 3** Al igual que con los text box y los radio buttons, un *check box* también consta de dos partes: una caja cuadrada seguida por una descripción textual del campo de ingreso para el cual el usuario debe proveer el valor de sí/no. Los check boxes proveen al usuario con la flexibilidad de seleccionar el valor por el teclado o el ratón. Un campo de datos de entrada cuyo valor sea sí está representado por un cuadrado que se llena con una “paloma”. La ausencia de una “paloma” significa que el valor del campo de ingreso es no. El usuario simplemente conmuta el valor del campo de ingreso de un valor/estado para el otro como se desee.

A menudo un usuario necesita ingresar un campo de datos cuyo conjunto de valores consiste en un simple valor de sí o no. Por ejemplo, a un usuario se le puede pedir un valor de sí/no para elementos tales como los siguientes datos de entrada: ¿CRÉDITO APROBADO? ¿PERSONA DE EDAD? ¿HA SIDO USTED ALGUNA VEZ CONDENADO POR FRAUDE? ¿Y PODEMOS CONTACTAR A SU PATRÓN ANTERIOR? En cada situación podría usarse un control de check box. Un control de check box le ofrece una manera visual e intuitiva para que el usuario ingrese esos datos.

El ejemplo previo representó un panorama simplificado para el uso de un check box. En una sola pantalla de ingreso de datos puede ser posible solicitarle a un usuario que ingrese valores en varios campos de entrada relacionados que tengan un valor de sí/no. Por ejemplo, un recepcionista en una clínica de salud puede estar ingresando datos de una forma que fue completada por el paciente. En una sección de esa forma, el paciente pudo haber sido interrogado acerca de varias enfermedades o acerca de su historial médico pasado y pudo haber recibido órdenes para “verificar todo lo que sea aplicable” de una lista de tipos de enfermedades diversas. Si la pantalla de entrada del recepcionista se diseña correctamente, ésta representaría cada enfermedad como un campo de entrada de datos usando un control de check box. Los controles se asociarían físicamente en un grupo en la pantalla. El grupo también recibiría un título/encabezado correcto. Reconozca que si bien los check boxes pueden ser visualmente agrupadas en la pantalla, cada una funciona como un campo de entrada independiente.

El seguimiento de estos lineamientos recomendados mejorará el uso de los controles del check box. Asegúrese de que la descripción textual es significativa para el usuario. Busque la forma de agrupar check boxes para campos de entrada relacionados con sí/no y proporcione un título descriptivo del grupo.

Con la idea de apoyar la navegación del usuario y la selección de un grupo de check boxes, arregle el grupo de controles de dichos controles a fin de que estén alineados verticalmente y justificados a la izquierda. Si es necesario, alinee horizontalmente y asegúrese de dejar un espacio adecuado para separar visualmente los controles uno del otro. Finalmente, proporcione más asistencia al usuario secuenciando de manera apropiada los campos de entrada de datos de acuerdo a su descripción textual. En la mayoría de los casos, cuando el número de controles de check box es grande, la ordenación en secuencia deberá ser alfabética. En los casos donde la descripción del texto representa rangos en dólares o alguna otra medida, la ordenación en secuencia puede ser de acuerdo con la ordenación numérica. En otros casos, como aquellos donde se agrupa un número muy limitado de controles, la base para hacer la secuencia puede ser de acuerdo con la frecuencia con que se selecciona un valor dado de sí/no del campo de datos de entrada. (Todos los campos de datos de entrada representados con el uso de un check box tienen un valor por omisión; ya sea con verificación o sin ella.)

**List box 4** Un list box es un control que requiere que el usuario seleccione un valor de un elemento de datos de una lista de selecciones posibles. El *list box* es rectangular y contiene una o más filas de valores posibles de datos. Los valores pueden aparecer ya sea como una descripción textual o una representación gráfica. Los list boxes que tengan un gran número de valores posibles pueden incluir barras de desplazamiento en pantalla para navegar a través de la lista de elecciones.

Es también común que la fila de un list box contenga más que una columna. Por ejemplo, un list box podría contener simplemente filas que tengan una columna individual de valores posibles para un artículo de datos de entrada llamado CÓDIGO DE TRABAJO. Sin embargo, tal vez sea pedir demasiado esperar que el usuario reconozca lo que cada código de trabajo realmente representa. En este caso, para colocar los valores de CÓDIGO DE TRABAJO dentro de una perspectiva significativa, el list box podría incluir una segunda columna conteniendo el TÍTULO DEL TRABAJO correspondiente para cada código de trabajo.

¿Cómo escoge uno entre un radio button y un control de list box? Ambos controles son útiles para asegurar que el usuario ingrese el valor correcto para un elemento de datos. Ambos son también apropiados cuando sea necesario hacer que las elecciones de valor sean constantemente visibles para el usuario.

La decisión está impulsada generalmente por el número de valores posibles para los elementos de datos y la cantidad de espacio de pantalla disponible para el control. Las capacidades de desplazamiento en pantalla hacen que los list boxes sean apropiados para usarse en los casos donde hay un espacio limitado de pantalla disponible y el elemento de datos de entrada tiene un gran número de valores predefinidos, mutuamente excluyentes para escoger.



Hay varios lineamientos a considerar al destinar un list box como una manera de ingreso de datos. Un list box deberá estar acompañado por un título descriptivo. Evite el uso de abreviaturas para poner subtítulos, y ponga en mayúscula sólo el primer carácter del texto del título. En general es también aceptado que el título esté seguido por dos puntos para ayudar al usuario a distinguir visualmente el título de la caja.

La posición del título es significativa. El usuario deberá poder asociar con claridad el título con el list box. Por consiguiente, éste deberá dar la apariencia de estar justificado a la izquierda, inmediatamente por encima del list box.

Hay también varios lineamientos relacionados con el list box. Primero, se recomienda que un list box contenga un valor por omisión resaltado. Segundo, considere el tamaño del list box. Generalmente, la anchura del list box deberá ser bastante grande para ingresar la mayoría de los caracteres de los datos de entrada de longitud fija para ser introducidos con la idea de que el usuario los vea. La longitud de la caja deberá permitir al menos tres selecciones y estar limitada en tamaño para contener aproximadamente siete selecciones. En ambos casos, deberán usarse características de desplazamiento en pantalla para sugerir que las selecciones adicionales están disponibles para el usuario.

Si las representaciones gráficas sirven para la selección de valores, asegúrese de que los gráficos sean significativos y verdaderamente representativos de la selección. Si se usan descripciones con texto, use letras mayúsculas y minúsculas y asegúrese de que las descripciones sean significativas. ¡Es importante que estas decisiones o estos juicios se basen en la perspectiva y las opiniones del usuario!

Usted también deberá considerar cuidadosamente la facilidad con la cual un usuario puede explorar e identificar las selecciones que aparecen en el list box. La lista de selecciones deberá estar justificada a la izquierda para ayudar en la navegación. Asegúrese de involucrar al usuario al diseñar la orden en la cual las selecciones aparecerán en la lista. En algunos casos, puede ser natural para el usuario si la lista de selecciones aparece por orden alfabético. En otros casos, la frecuencia en la cual un valor se seleccione puede ser importante con relación a donde se encuentre en la lista.

**Drop down list 5** Una drop-down list es otro control que requiere que el usuario seleccione un valor de elemento de dato de una lista de selecciones posibles. Una *drop-down list* consta de un campo rectangular de selección con un botón pequeño conectado a su lado. El botón pequeño contiene la imagen de una flecha que apunta hacia abajo y una barra. Este botón intenta sugerir al usuario la existencia de una lista oculta de valores posibles para un elemento de dato.

Cuando se requiere, la lista oculta parece “desplegarse hacia abajo” sobre el campo de selecciones revelándose para el usuario. La lista revelada tiene características parecidas al control del list box mencionado en la sección anterior. Cuando el usuario selecciona un valor de la lista de selecciones, éste se exhibe en el campo de selección y la lista de elecciones otra vez se oculta del usuario.

Un drop-down list deberá usarse en los casos donde el elemento de dato tenga un gran número de valores predefinidos y la disponibilidad del espacio de la pantalla prohíbe el uso de un list box. Una desventaja de un drop-down list es que requiere pasos adicionales por el usuario, en contraste con los controles anteriormente citados.

Muchos de los lineamientos del uso de las list boxes son directamente aplicables a las drop-down list. Una excepción es la colocación del título. Por lo general, el título de un drop-down list ya está alineado a la izquierda inmediatamente por encima de la porción del campo de selección del control o localizado a la izquierda del control.

**Combination box 6** El *combination box*, a menudo simplemente llamada una caja combinada, es un control cuyo nombre refleja el hecho de combinar las capacidades de un cuadro de texto con las de un list box. Un combination box le da al usuario la flexibilidad de ingresar un valor de elemento de dato (al igual que con un text box) o de seleccionar su valor de una lista (al igual que con un list box).

A primera vista, un combination box se parece mucho a un control drop-down list. Sin embargo, a diferencia de éste, la caja rectangular puede servir como un campo de edición para que el usuario ingrese directamente un valor del elemento de dato. Una vez que el botón pequeño es seleccionado, una lista oculta se revela apareciendo con una pequeña sangría debajo del campo rectangular de entradas.

Cuando el usuario selecciona un valor de la lista de opciones, el valor elegido se despliega en el campo de entrada y la lista de selecciones otra vez se oculta al usuario.

Un combination box se usa de la mejor manera posible cuando el espacio de la pantalla es limitado y es deseable proveer al usuario con la opción de seleccionar un valor de una lista o teclear un valor que puede o no aparecer como una opción de la misma.

Los mismos lineamientos para usar drop-down lists se aplican directamente a los combination boxes.

**El spin box** 7 es un control basado en pantallas que consta de un text box con una sola línea seguido inmediatamente por dos botones pequeños los cuales están verticalmente alineados. El botón superior tiene una flecha apuntando hacia arriba, y el botón inferior tiene una flecha señalando hacia abajo. Este control permite al usuario ingresar datos directamente en el text box asociado o seleccionar un valor dando un clic sobre los botones para que despliegue (o “recorra”) una lista de valores. Los botones tienen una unidad de medida asociada con ellos. Cuando el usuario da un clic sobre uno de los botones con flecha, aparecerá un valor en el text box. Dicho valor es manipulado dando un clic sobre dichos botones. El botón con flecha hacia arriba aumentará el valor en el cuadro de texto por una unidad de medida, mientras que el botón con flecha hacia abajo disminuirá el valor en el cuadro de texto por la misma unidad de medida.

El spin box se usa de la mejor manera posible para permitirle al usuario hacer una selección de entrada de datos usando los botones de navegación a través de un conjunto pequeño de selecciones significativas o tecleando directamente el valor de los datos en el cuadro de edición. Los valores de datos para un spin box deberán estar secuenciados en una forma previsible.

Los spin boxes deberán contener una etiqueta o subtítulo que identifique claramente el elemento de dato de entrada. Esta etiqueta deberá estar ubicada a la izquierda del cuadro de edición o alineada a la izquierda inmediatamente por encima de la caja de texto del control. Finalmente, los Spin Boxes siempre deberán contener un valor por omisión en la caja de texto del control.

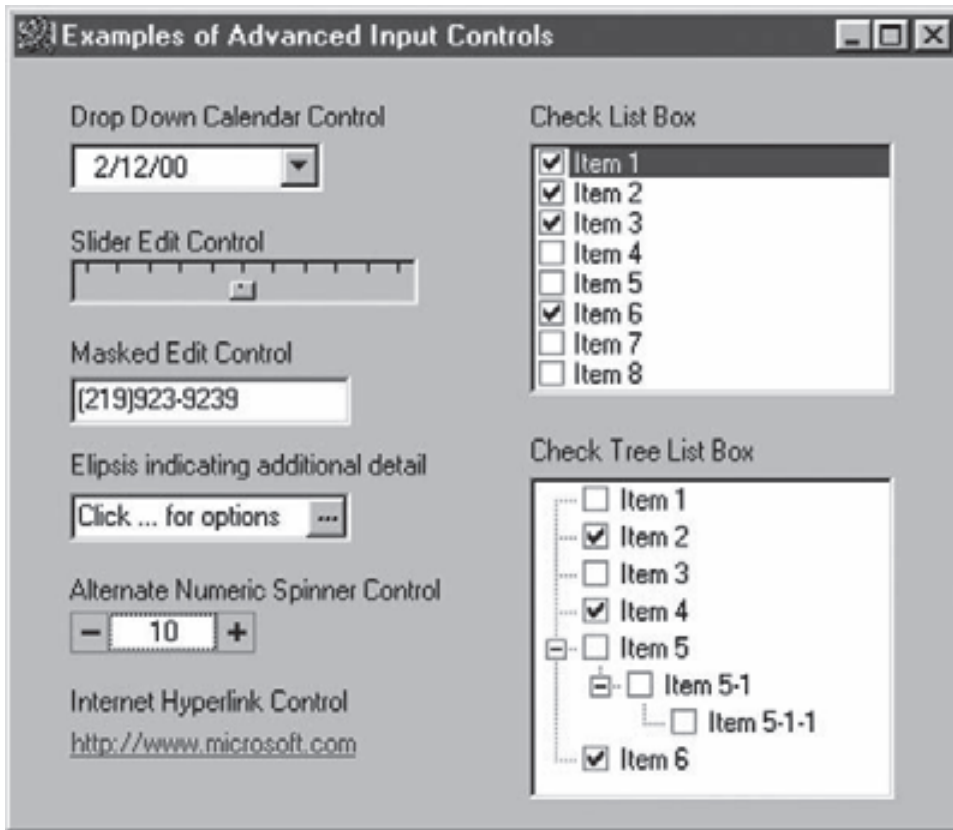
**Botones** 8 Estrictamente hablando, los botones no son controles de entrada. No contribuyen ni a la selección ni al ingreso de datos reales. No obstante, el diseño de las formas de ingreso estaría incompleto sin ellos. Los *botones* sirven para varios propósitos. Le permiten a un usuario introducir todos los datos para ser procesados, cancelar una transacción, u obtener ayuda. Pueden usarse para navegar entre las instancias de la misma forma.

Existen muchos más controles basados en pantallas que sirven para diseñar las interfaces gráficas del usuario, siendo los más comunes los mencionados arriba para captura datos. Hay otros, y usted deberá familiarizarse con ellos y con su uso correcto para el ingreso de datos. En capítulos posteriores usted estudiará otros controles destinados a otros propósitos. Manténgase al día con los desarrollos en el área de Interfaz Gráfica del Usuario ya que seguramente estarán disponibles nuevos controles.

## > Controles de entrada avanzados

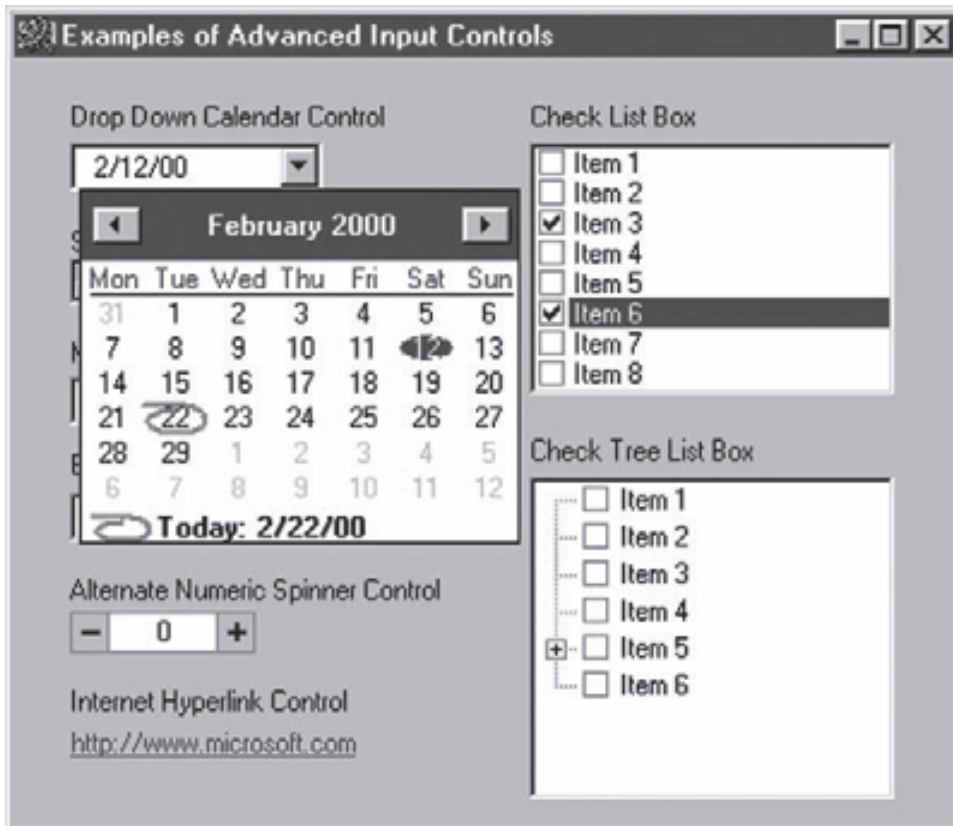
Las figuras 14.7a) y b) ilustran controles adicionales para la entrada de datos. Estos controles avanzados pueden ser usados en interfaces de *Windows* para crear una apariencia y percepción más sofisticada. Probablemente están disponibles controles equivalentes para aplicaciones basadas en la Web, pero la mayoría de las aplicaciones de comercio electrónico basadas en la Web utilizan los formatos más simples. No discutiremos estos controles en detalle, pero están resumidos como sigue:

- *Drop-down calendar*. Un campo se ilustra en la figura 14.7a). El dar un clic sobre la flecha descendente al lado de la fecha crea el calendario de aparición automática mostrado en la figura 14.7b). El calendario familiar es otro ejemplo de diseño metafórico.
- *Slider edit calendar*. Ésta es una manera no numérica de seleccionar un valor.
- *Masked edit control*. Este control crea las verificaciones del formato descritas anteriormente en el campo mismo.
- *Ellipsis control*. Al dar un clic sobre los tres puntos se crea un diálogo de aparición automática para el ingreso de datos. Podría usarse para un campo que consta de varias partes (tales como una dirección: calle, ciudad, estado y código postal).



**FIGURA 14.7**  
 Controles avanzados de entrada de GUI

a)



b)

- *Alternative numeric spinner*. Éste es un tipo diferente de caja recorrible para la entrada de datos.
- *Internet hyperlink*. Con una función similar a la de un botón, un hipervínculo puede enlazarse a las páginas Web, pero también a otros formatos de *Windows*. Ésta es una manera efectiva de ocultar los formularios de entrada de datos que no son aplicables a todos o a la mayoría de los usuarios.
- *Check list box*. Este control es útil para combinar varios check boxes en situaciones donde varios controles de este tipo pueden ser aplicables.
- *Check three list box*. Este control es útil para presentar opciones de datos que necesitan estar organizados jerárquicamente en una estructura de tipo de árbol.

## Cómo diseñar y elaborar prototipos de entradas

¿Cómo diseña usted las entradas en línea? Tradicionalmente, los diseñadores estaban preocupados con el contenido global, la apariencia, y la funcionalidad de la pantalla de entradas; en un aislamiento respecto a otras pantallas que había que diseñar. Los diseñadores sabían que simplemente diseñarían un conjunto de pantallas de menú de las cuales los usuarios seleccionarían una opción que los conduciría a la pantalla correcta de entrada. Bastante simple. Sin embargo, dados los ambientes gráficos de hoy, hay un énfasis en desarrollar un sistema global que se mezcle bien dentro del ambiente del lugar de trabajo del usuario. Este énfasis raramente da como resultado una aplicación con interfaz jerárquica, accionada por menús que caracterizó las aplicaciones más tradicionales del pasado basadas en texto o comandos.

Las siguientes secciones demostrarán cómo se completa la primera etapa del diseño de entradas. Echaremos mano de ejemplos de nuestro estudio de casos SoundStage. Examinaremos las entradas de cliente/servidor, las entradas basadas en *Windows*, las entradas basadas en la Web, y las entradas de comercio electrónico que corren en un navegador. Más tarde, en el capítulo 15, vamos a integrar las salidas (de capítulo 13) y las entradas de este capítulo en un diálogo e interfaz global de usuario.

### > Herramientas automatizadas para diseño de entradas y elaboración de prototipos

En el pasado reciente, las herramientas primarias para el diseño de entradas eran las *gráficas de trazado de registros* y *gráficas de trazado de despliegue*. Hoy este enfoque de “esquema” no es a menudo practicado. Es un proceso tedioso que no conduce a la creación de prototipos preferida actualmente, ni a las estrategias de desarrollo de aplicación rápida, que usan herramientas automatizadas para acelerar el proceso de diseño.

Antes de la aparición de las herramientas automatizadas, los analistas podían esbozar sólo dibujos aproximados de las entradas para obtener una percepción de cómo querían los usuarios del sistema que se vieran las salidas o cómo se estructurarían los registros de lotes. Con las herramientas automatizadas, podemos desarrollar prototipos más realistas de estas entradas.

Discutiblemente, la herramienta automatizada más usada para el diseño de las entradas es el ambiente de desarrollo de aplicación de base de datos de la PC. Aun cuando Microsoft *Access* no es lo suficientemente poderosa para desarrollar la mayoría de aplicaciones a nivel de empresa, usted puede sorprenderse de cuántos diseñadores usan *Access* para hacer prototipos de tales aplicaciones. Dada una estructura de la base de datos (fácilmente especificada en *Access*), usted rápidamente puede generar o crear formularios para el ingreso de datos. Usted puede incluir la mayoría de los controles de la interfaz gráfica del usuario que describimos en este capítulo. Posteriormente, los usuarios pueden ejecutar esos formularios y le pueden decir cuáles funcionan y cuáles no.

Muchas herramientas de ingeniería de software asistida por computadora incluyen instrumentos para el trazado de reportes y pantallas usando el repositorio de proyecto creado durante el análisis de requerimientos. La herramienta de diseño de pantallas del *System Architect* fue demostrado previamente en el capítulo 13, figura 13.7.

La mayoría de los lenguajes de programación basados en interfaces gráficas del usuario, como *el Visual Basic*, pueden usarse fácilmente para construir prototipos no funcionales de las entradas de datos. El término clave aquí es *no funcional*. Los formularios (pantallas) se verán reales, pero no habrá código para implementar ninguno de los botones o campos. Esa es la esencia de la creación rápida de prototipos.

## > Proceso de diseño de entradas

El diseño de entradas no es un proceso complicado. Algunos pasos son esenciales, y otros son determinados por las circunstancias. Los pasos son:

1. Identifique las entradas del sistema y revise los requerimientos lógicos.
2. Seleccione controles apropiados de la interfaz gráfica del usuario.
3. Diseñe, valide, y pruebe las entradas usando alguna combinación de:
  - a) Herramientas de trazado (por ejemplo, esquemas a mano, gráficas de trazado de impresora/despliegue, o ingeniería de software asistida por computadora).
  - b) Herramientas para crear prototipos (por ejemplo, hojas de cálculo, PC DBMS, 4GL).
4. Si es necesario, diseñe el documento fuente.

En las siguientes secciones, examinamos estos pasos e ilustramos algunos ejemplos del proyecto SoundStage.

**Paso 1: Identifique las entradas del sistema y revise los requerimientos lógicos** Los requerimientos de las entradas deberán haberse definidos durante el análisis de requerimientos. Los diagramas de flujo de datos físicos (o las unidades de diseño, ambos descritos en el capítulo 11) son un buen punto de partida para el diseño de las entradas. Esos DFD identifican tanto las salidas netas del sistema (el agente externo a procesar) como el método de implementación.

Su metodología de desarrollo de sistemas y estándares determinará si cada uno de estos flujos netos de datos de entrada también puede estar descrito como un flujo de datos lógico en un diccionario de datos o un repositorio (vea capítulo 8). La estructura de datos para un flujo de datos específica los atributos o campos que deben incluirse en la salida. Si esos requerimientos son especificados en la notación relacional algebraica, usted rápidamente puede determinar qué campos se repiten, qué campos tienen valores opcionales, etcétera. Considere la siguiente estructura de datos:

Estructura de datos que define los requerimientos lógicos	Comentario
ORDEN = NÚMERO DE ORDEN + FECHA DE LA ORDEN + NÚMERO DE CLIENTE + NOMBRE DEL CLIENTE + DIRECCIÓN DE FLETE DEL CLIENTE = DIRECCIÓN > + (DIRECCIÓN DE COBRO DEL CLIENTE = DIRECCIÓN >) + 1(NÚMERO DE PRODUCTO + CANTIDAD ORDENADA) N + (NÚMERO POR OMISIÓN DE LA TARJETA DE CRÉDITO)	← Identificador único de la salida. ← Uno de los muchos campos que debe adoptar un valor. La falta de paréntesis indica que se requiere un valor. ← Apuntador para una definición relacionada. ← Un grupo de campos que se repite 1 – n veces. Los paréntesis indican valores optativos. ← Un campo optativo, lo que implica que es uno que no tiene que tener un valor.

A falta de tales requerimientos precisos, pueden existir prototipos de descubrimiento que fueron creados durante el análisis de requerimientos. En uno u otro caso, una buena declaración de requerimientos deberá estar disponible en algún formato.

Los requerimientos de entradas especificados durante el análisis de requerimientos para el caso de estudio de SoundStage fueron revisados, y se determinó que tres entradas pertenecían al tema de VIDEOCINTA. Se determinó también que una pantalla individual de entradas pudo usarse para soportar las tres entradas: NUEVO TÍTULO DE VIDEO, TÍTULO DE VIDEO DESCONTINUADO, Y ACTUALIZACIÓN DE TÍTULO DE VIDEO. El contenido de los datos para las tres entradas deberá capturar o exhibir los siguientes datos:

NÚMERO DE PRODUCTO +  
CÓDIGO UNIVERSAL DEL PRODUCTO +  
CANTIDAD EN EXISTENCIA +  
TIPO DE PRODUCTO +  
PRECIO UNITARIO AL MENUDEO SUGERIDO POR EL FABRICANTE +  
PRECIO UNITARIO POR OMISIÓN DEL CLUB +  
PRECIO UNITARIO ESPECIAL ACTUAL +

UNIDADES VENDIDAS EN EL MES CORRIENTE +  
 UNIDADES VENDIDAS EN EL AÑO EN CURSO +  
 TOTAL DE UNIDADES VITALICIAS VENDIDAS +  
 TÍTULO DEL TRABAJO +  
 DESCRIPCIÓN DEL CATÁLOGO +  
 FECHA DE DERECHO DE AUTOR  
 VALOR DEL CRÉDITO +  
 PRODUCTOR +  
 DIRECTOR +  
 CATEGORÍA DEL VIDEO

Los atributos NÚMERO DE PRODUCTO, VENTAS MENSUALES DE UNIDADES, VENTAS DE UNIDADES AL AÑO, y TOTAL DE UNIDADES VENDIDAS no deben ser ingresados por el usuario. Más bien, deben ser generados automáticamente por el sistema. También, para la CUBIERTA DEL TÍTULO, se esperará que el usuario simplemente especifique un archivo con un mapa de bits que contendrá una imagen real del nuevo título del video.

**Paso 2: Seleccione controles apropiados de la interfaz gráfica del usuario** Ahora que tenemos una idea del contenido para nuestra entrada, podemos ocuparnos del control apropiado basado en pantallas que debe usarse para que cada atributo aparezca en nuestra pantalla. Usando el enfoque de programación basado en un repositorio, primero inspeccionaríamos para ver si tales decisiones y otras características de atributo ya han sido hechas y si se han grabado como entradas del repositorio. Si es así, simplemente reutilizaremos las entradas del repositorio que corresponden a los atributos que usaremos en nuestras pantallas de entradas. En los casos donde no hay entrada del repositorio, simplemente tendremos que crearlos.

Para escoger el control correcto para nuestros atributos, debemos empezar por examinar los valores posibles para cada atributo. Aquí hay algunas decisiones preliminares en relación con nuestros atributos de entrada identificados en el paso previo:

- NÚMERO DEL PRODUCTO, UNIDADES VENDIDAS EN EL MES CORRIENTE, UNIDADES VENDIDAS EN EL AÑO EN CURSO, TOTAL DE UNIDADES VITALICIAS VENDIDAS, CÓDIGO UNIVERSAL DEL PRODUCTO, PRECIO UNITARIO AL MENUDEO SUGERIDO POR EL FABRICANTE, PRECIO UNITARIO POR OMISIÓN DEL CLUB, PRECIO UNITARIO ESPECIAL ACTUAL, PRODUCTOR, y los atributos del DIRECTOR todos tienen valores de datos de entrada que son ilimitados en el alcance o no son editables. Como el diseñador es incapaz de proveer al usuario de una lista significativa de valores de la cual escoger, se escogió un text box de una sola línea. Como el atributo DESCRIPCIÓN DEL CATÁLOGO también se ajusta a estos criterios, fue seleccionado un multiple-line text box (conocido como “memo box” por algunos productos).
- TIPO DEL PRODUCTO, el IDIOMA, la CODIFICACIÓN DEL VIDEO, ORIENTACIÓN DE LA PANTALLA, y el TIPO DE MEDIOS DEL VIDEO contienen todos un conjunto predefinido limitado de valores. Por consiguiente, se determinó que los radio buttons serían el control basado en pantallas preferido para estos elementos de entrada.
- Se determinó que ¿TÍTULO CERRADO? es un atributo de entrada que contiene un valor de sí/no. Por consiguiente, un check box fue seleccionad como el control para este atributo.
- La CANTIDAD EN EXISTENCIA, el TIEMPO CORRIENTE, la FECHA DE LOS DERECHOS DE AUTOR, y el VALOR DE CRÉDITO contienen valores de datos que se pueden secuenciar de una manera previsible. Así, un spin box con un text box asociado sería una buena elección para estos atributos.
- Los atributos CATEGORÍA DEL VIDEO y SUBCATEGORÍA DEL VIDEO contienen un gran número de valores predefinidos. Con tantos atributos para exhibir en nuestra pantalla, se determinó que un drop-down list sería el mejor control seleccionado.
- La CUBIERTA del TÍTULO presentó un reto interesante. Su valor es de hecho una unidad, un directorio, y el nombre de un archivo que contiene una imagen del mapa de bits de la cubierta del título de video. Este atributo hará uso de un control avanzado designado *image box* para almacenar la imagen de la cubierta del título de video. Cuando este objeto sea elegido por el usuario, se usará un conjunto de controles y el diálogo especial (la interacción del usuario) para capturar la entrada para este elemento. Ilustraremos esta entrada más tarde en el paso 3.

Nuevamente, hay muchos otros controles basados en pantallas que podrían usarse para ingresar datos. Nuestros ejemplos enfocan la atención en los controles más comúnmente

**Video Titles Maintenance**

Product Type:  General Merchandise  Entertainment Merchandise

New Edit Delete Exit

**Common Information:**

UPC: 8616-21425-3 Quantity in Stock: 29 Product Number: 1477328

MSRP: \$24.99 Monthly Unit Sales: 2

Club Price: \$19.99 Year Unit Sales: 4

Special Price: \$17.99 Total Unit Sales: 6

Title: The Matrix

Starring: Keanu Reeves, Laurence Fish

Director: Andy Wachowski, Larry Wach

CC:  Rating: R Time: 136

Copyright: 1999 Credit Value: 1

Mind-warp stunts. Techno-slammin' visuals. Mega-kick action. Keanu Reeves and Laurence Fishburne lead the fight to free humankind in The Matrix, the see-and-see-again cyberthriller written and directed by the Wachowski brothers (Bound). The story sears, the special effects stake out new moviemaking territory - the move flat out rocks.

Category: SF Sub-Category: Type:  Audio  Game  Video Language:  Dubbed  English  Subtitled Video Media Type:  DVD  DVD DTS  Laserdisc  VHS Screen Aspect:  16:9  4:3  Anamorphic  Dual Format

**FIGURA 14.8** Prototipo de entrada para el mantenimiento de los títulos de video

usados. Qué tan bien complete usted esta actividad será una función de qué tan versado esté en estos controles comunes y otros más avanzados.

**Paso 3: Diseño, valide y pruebe las entradas** Este paso incluye el desarrollo de pantallas prototipo que los usuarios deben revisar y probar. Su retroalimentación puede dar como resultado la necesidad de regresar a los pasos 1 y 2 para agregar atributos nuevos y direccionar sus características.

Observemos un par de prototipos de la pantalla SoundStage. La figura 14.8 representa una posible pantalla del prototipo para manejar TÍTULO DEL VIDEO NUEVO, TÍTULO DEL VIDEO DESCONTINUADO, y ACTUALIZACIÓN DEL TÍTULO DEL VIDEO. El logotipo que aparece en la esquina derecha superior de la pantalla se incluyó para cumplir un estándar de la compañía: todas las pantallas deben exhibir el logotipo de la compañía. Los botones que también aparecen en la parte central y superior derecha de la pantalla se agregaron por la decisión de combinar las tres entradas en una pantalla individual. Fueron necesarios para proporcionar al usuario la opción de seleccionar el tipo deseado de entrada y acción de registro. Discutiremos estos botones y otros comandos y controles de navegación, así como su uso en el capítulo 15.

Observe los siguientes aspectos en la figura 14.8:

- 1 El NÚMERO DE PRODUCTO, las VENTAS MENSUALES DE LAS UNIDADES, las VENTAS DE UNIDADES AL AÑO, y las VENTAS TOTALES DE UNIDADES se proyectan en la pantalla con un color especial como una pista visual para que el usuario identifique que estos campos están cerrados y no puede ingresar datos en ellos. Estos campos son automáticamente generados por el sistema. Otros campos que aparecen en la pantalla tienen un fondo blanco como una pista visual de que pueden ser editados.
- 2 Se especificaron máscaras de edición para estos campos de entrada. El campo del CÓDIGO UNIVERSAL DEL PRODUCTO contiene guiones en posiciones especificadas. El usua-

rio realmente no ingresa estos guiones; simplemente teclea los números, y luego el contenido entero es desplegado según la máscara de edición especificada. Lo mismo es cierto para los campos del PRECIO AL POR MENOR SUGERIDO POR EL FABRICANTE, el PRECIO UNITARIO POR OMISIÓN DEL CLUB, y el PRECIO UNITARIO ESPECIAL ACTUAL. Por ejemplo, en cualquiera de estos tres campos el usuario podría teclear el número 9 y oprimir la tecla de ingreso, y el contenido sería desplegado (según la máscara de edición) con un signo de dólar y un punto decimal.

- 3 Cada campo en la pantalla ha recibido una etiqueta que es significativa para los usuarios. La retroalimentación de los usuarios indicó “CC”, era una abreviación comúnmente reconocida de “título cerrado. También, los usuarios indicaron que no era necesaria una etiqueta para la DESCRIPCIÓN DEL CATÁLOGO.
- 4 Los radio buttons relacionados han sido arreglados en un group box que contiene una etiqueta descriptiva. Los group boxes se usan con frecuencia para asociar visualmente una variedad de controles que se relacionan. Por ejemplo, los campos dentro del group box etiquetado como “información común” estaban agrupados porque el usuario asocia estos atributos con cualquier tipo de producto SoundStage. También, cada etiqueta que corresponde a una opción de radio button no es lo que realmente se ingresa y almacena en la base de datos. Más bien, lo que usted ve es el significado del valor. El valor real que es almacenado es un código. Por ejemplo, el valor de código E se guardaría realmente en lugar de “inglés” si el usuario selecciona el radio Button rotulado como “inglés” para el atributo IDIOMA.
- 5 El multiple-line text box tiene una característica de barra de desplazamiento vertical si el texto llena el text box. Ésta es una pista visual de que hay texto adicional que no aparece dentro del campo de DESCRIPCIÓN DEL CATÁLOGO.

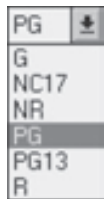
Al hacer los prototipos de las pantallas de entrada, usted necesita permitir que el usuario ejecute o pruebe las pantallas. Parte de esa experiencia debería involucrar la demostración de cómo puede obtener el usuario instrucciones o ayuda apropiada. Las nuevas versiones de productos de Microsoft usan lo que se llama “sugerencias de las herramientas” para proveer una descripción breve de los botones y las cajas que aparecen en una pantalla. La descripción de la sugerencia de las herramientas se exhibe cuando el usuario sitúa al ratón sobre la parte superior del objeto. También, la tecla F1 es universalmente aceptada para iniciar ayuda sensible al contexto. Un botón de ayuda es otra opción. Independientemente del (los) enfoque(s) que usted use, no es necesario implementar la ayuda en un prototipo.

Finalmente, los prototipos no necesitan desplegar todos los detalles a un usuario a menos que sean solicitados (o sean activados por una acción del usuario). Por ejemplo, la drop-down list para Motion Picture Association del código de VALUACIÓN DE AMÉRICA despliega sólo un valor predeterminado. Sin embargo, la flecha que apunta hacia abajo es una pista visual de que existe un List box que contiene otros valores posibles. El List box puede verse simplemente dando un clic sobre la flecha que apunta hacia abajo. El resultado de esa acción se ilustra en el margen.

El ejemplo previo fue bastante simple porque contiene sólo datos que podrían actualizarse en una tabla de bases de datos. Pero, ¿qué ocurre si una entrada incluye datos que deben actualizarse en más de una tabla? Y suponga que hay una relación de uno a muchos entre las tablas. Considere la ORDEN DEL MIEMBRO, la cual tiene una relación uno a muchos con los PRODUCTOS ORDENADOS POR EL MIEMBRO. ¿Cómo diseñamos una entrada individual para capturar los datos de ambas tablas?

La figura 14.9 representa una pantalla prototipo para ingresar al MIEMBRO y los PRODUCTOS ORDENADOS POR EL MIEMBRO en una forma individual que está segmentada en dos ventanas. Los datos del MIEMBRO están en la ventana superior, y el PRODUCTO ORDENADO POR EL MIEMBRO está en la ventana inferior. Usted puede preguntarse qué pasa si el número de PRODUCTOS ORDENADOS POR EL MIEMBRO excede el espacio destinado para esa ventana. En otras palabras, ¿dónde está la barra de desplazamiento para la ventana inferior? Muchos controles GUI de *Windows* son “inteligentes”. Si el número de filas en la ventana inferior excede el espacio, una barra de desplazamiento vertical aparecerá automáticamente.

Como un último ejemplo de *Windows*, la figura 14.10 muestra un diseño de pantalla individual que consolida tres entradas diferentes o similares de nuestros diagramas de flujo de datos: el MIEMBRO NUEVO, la CANCELACIÓN DEL MIEMBRO, y la ACTUALIZACIÓN DEL MIEMBRO. Esta forma también usa los controles estándar de entradas que hemos estudiado en



Un menú desplegable



Member Order Detail

10001 Members

Member Information:

Name: Jim Smith

Street Address: 129 Rural Route 6 P.O. Box: 12999

City: Fowler State: IN Zip Code: 47756-3333

Area Code: 319 Phone: 789-3333 Extension:

Member Orders

Alternate Selection	Selection of the Month + Alternative
824	2
844	11
38	
45	
54	
23	41
79	

Member Status: Good Standing

FIGURA 14.9

Prototipo de entrada para la orden de un miembro

este capítulo. La consolidación de flujos de datos lógicos y físicos en diseños de pantalla individual es muy común.

**Paso 4: Si es necesario, diseñe el documento fuente** Si un documento fuente se va a usar para capturar datos, también debemos diseñarlo. El documento fuente es para el usuario del sistema. En su forma más simple, el prototipo puede ser un esquema simple o la obra de un artista industrial.

Un documento fuente bien diseñado estará dividido en zonas. Algunas de ellas sirven para identificación; estas incluyen el nombre de la compañía, el nombre del formato, número oficial del formato, la fecha de la última revisión (un atributo importante que se omite a menudo), y los logotipos. Otras zonas contienen datos que identifican la ocurrencia específica del formato, como el número de secuencia del formulario (posiblemente preimpreso) y la fecha. La sección más grande del documento se usa para registrar datos de la transacción. Los datos que ocurren una vez y los que se repiten deberán estar separados en forma lógica. Los totales deberán dejarse para la sección inferior del formato porque generalmente se calculan y, por consiguiente, no se ingresan. Muchos formatos

Members Maintenance

Status:

Dropped

Frozen

Good Standing

Inactive

Probation

Member Number: 100003

New Delete

Edit Exit

Memberships Orders

Address Information:

Name: Jim Smith

Street Address: 129 Rural Route 6 P.O. Box: 12999

City: Fowler State: IN Zip Code: 47756-3333

Area Code: 319 Phone: 789-3333 Extension:

Credit Card Information:

American Express

Discover

Mastercard

Visa

Card Number: 5414-8981-2919-2912

Expiration Date: 02/2001

Balance: \$495.22

Bonus Balance: 1

Members

FIGURA 14.10

Prototipo de entrada para las compras de un miembro

The screenshot shows the SoundStage Entertainment Club website. The header includes the logo, a film reel icon, and navigation links for Videos, General, Videos, Help, Order Tracking, Basket, and Checkout. The main content area features a product page for 'The Matrix 1999' with a 'BUY' button, price information (\$18.99 vs \$24.98), and a list of special features. A table below lists various product editions and their prices.

**SoundStage Entertainment Club**

Videos | Help | Order Tracking | Basket | Checkout

**The Matrix 1999**  
 SoundStage Price: \$18.99 **BUY**  
 Retail Price: \$24.98 You save: \$5.99  
 Availability: InStock

Starring: Keanu Reeves Laurence Fishburne Carrie-Anne Moss Hugo Weaving  
 Director: Andy Wachowski Larry Wachowski  
 Features: Collector's Edition, Anamorphic Widescreen, 2.35:1 Aspect Ratio, Behind-The-Scenes, Commentary, Trailers, Documentary  
 Format: DVD MPAA Rating: R Release Date: 9/21/1999

Also available on:

Description:	Format/Price:	Purchase:
Matrix, The (Widescreen)	DVD \$17.99	<b>BUY</b>
Matrix, The (Special Edition)	VHS \$14.49	<b>BUY</b>
Matrix, The (Special Edition)(Widescreen)	VHS \$14.99	<b>BUY</b>
Matrix, The (with CD Soundtrack, Poster, Lobby Cards) (Special Edition)	VHS \$67.99	<b>BUY</b>
Matrix, The: Musical Score From The Motion Picture, Music by Don Davis	CD Soundtrack \$11.99	<b>BUY</b>
Matrix, The: Music From The Motion Picture (EDITED)	CD Soundtrack \$12.99	<b>BUY</b>
Matrix, The - DVD and CD Soundtrack Set (Explicit Lyrics)	Other \$25.99	<b>BUY</b>

Copyright © 2000 SoundStage Entertainment Club All Rights Reserved

[Home](#) [Privacy Statement](#) [Store Locator](#) [Help](#) [Contact Us](#) [Company Info](#) [Jobs](#)

**FIGURA 14.11** Prototipo de entrada para el carrito de compras de la Red

incluyen una zona de autorización para firmas. Las instrucciones deberán colocarse en una posición conveniente, preferentemente no en la parte de atrás del formato.

Las herramientas para crear prototipos se han hecho más avanzadas en estos últimos años. Los programas de hojas de cálculo como *Excel* de Microsoft pueden hacer modelos muy realistas de formatos. Estas herramientas le dan un control sobresaliente sobre los estilos y los tamaños de fuente, los gráficos para logotipos, y cosas por el estilo. Las impresoras de láser pueden producir salidas impresas excelentes de los prototipos.

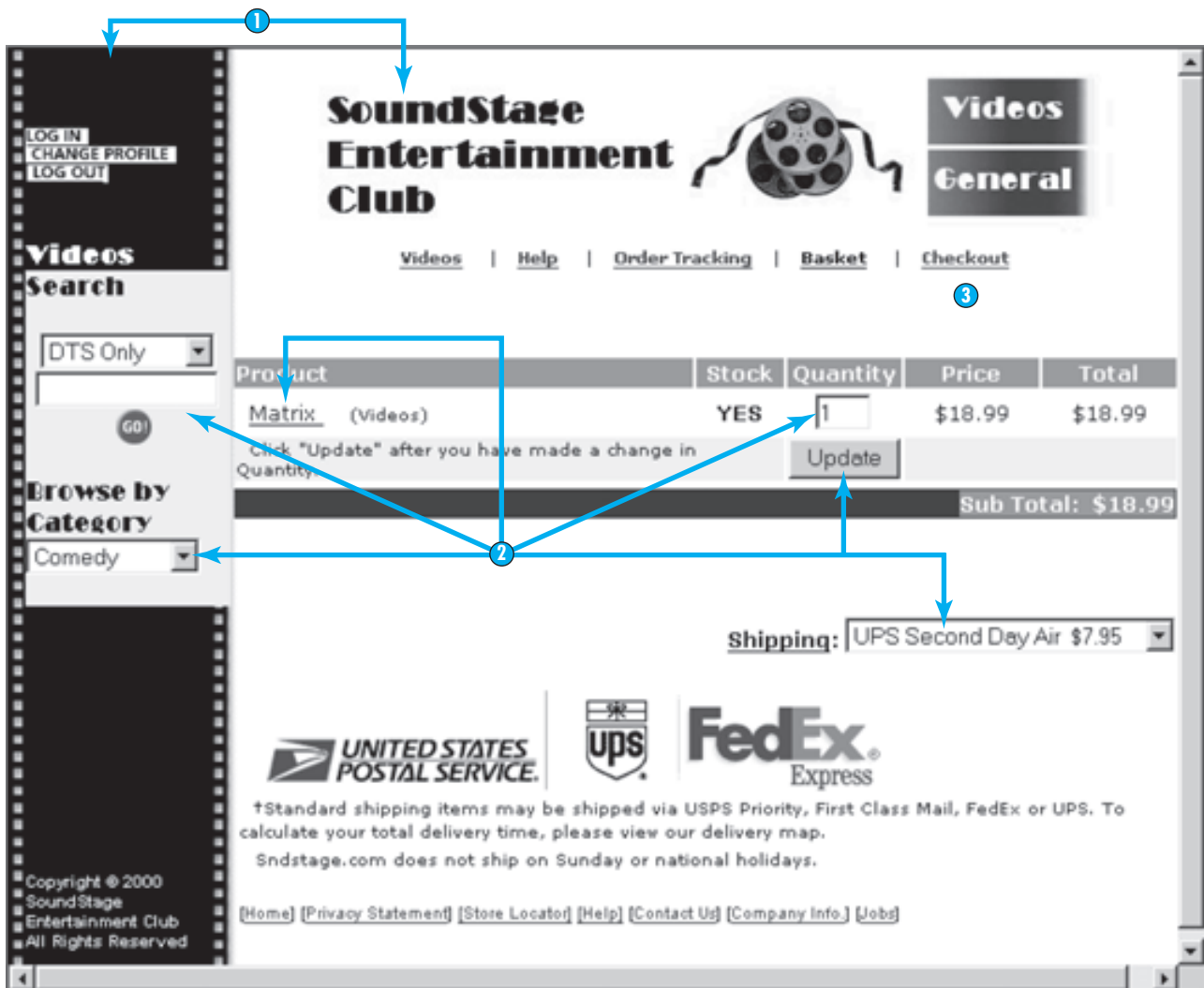
Otra forma de hacer prototipos de documentos fuente es desarrollar un modelo aproximado usando un procesador de texto. Proporcione el modelo a una de las tantas

oficinas de publicidad que pueden transformar el modelo aproximado a formatos impresionantes (tan impresionantes, de hecho, que algunas compañías actualmente desarrollan formatos así, en lugar de contratar su diseño con un fabricante de formatos).

### > Entradas basadas en la Web y el negocio electrónico (e-business)

Las últimas consideraciones sobre el diseño de entradas que queremos abordar están relacionadas con las salidas basadas en la Web. El proyecto SoundStage le añadirá diversas capacidades de comercio electrónico y de negocios electrónicos al sistema de información de Servicios para los Miembros. Algunas de estas capacidades requerirán entradas basadas en la Web que deben diseñarse.

Un requerimiento lógico de salida para el proyecto es una ORDEN DEL MIEMBRO basada en la Web. Únicamente le mostramos la versión del cliente/servidor. Ahora examinemos la versión basada en la Web. Es común presentar una fachada de Web (figura 14.11). Además de proveer al miembro de información acerca de los productos SoundStage (una salida), el miembro puede dar un clic sobre el botón de “comprar” para iniciar una compra. Eso lleva al miembro a lo que se ha convertido en una metáfora de pantalla común en las aplicaciones de comercio electrónico (e-commerce), la pantalla del *carrito de compras* (vea la figura 14.12). Las interfaces de Web tienden a ser algo más artísticas que las de



**FIGURA 14.12** Prototipo de entradas para el carrito de compras de la Red

*Windows*. Quizás eso sea parte del atractivo. La interfaz necesita ser atractiva visualmente para seducir al cliente para comprar productos a falta de un argumento de ventas verbal. En la figura 14.12:

- ❶ El “marco” del carrito de compras es independiente del marco general de navegación (a la izquierda). Este último le permite al usuario buscar y navegar por todo el sitio Web, con la esperanza de encontrar productos adicionales para añadir al carrito de compras.
- ❷ Botones, text boxes, hyperlinks, drop-down boxes, y otros controles comunes se aplican a una interfaz de Web en lugar de una interfaz de *Windows*.
- ❸ Un hyperlink (hipervínculo) de verificación envía al miembro a la siguiente “página” para completar la transacción.

La interfaz de Web le ofrece varias ventajas, como la habilidad automática para que los miembros usen los botones hacia adelante y hacia atrás para navegar en diferentes páginas de inventarios y órdenes en el sitio Web.

Este capítulo presentó un panorama detallado de las tareas de diseño de entradas de un proyecto. Si usted no ha estudiado el diseño de las salidas, debe regresar y leer el capítulo 13. De otra manera, el siguiente capítulo lógico es el capítulo 15, “Diseño de la interfaz del usuario”. El diseño de las interfaces de usuario conjunta las pantallas de entradas y de salidas en una experiencia general del usuario. Tal como hicimos en este capítulo, vamos a abordar tanto las interfaces de cliente/servidor, basadas en *Windows* como las soluciones de negocios electrónicos (e-business) basadas en la Web.

## Resumen



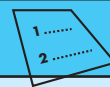
1. Varios conceptos son importantes para el diseño de las entradas. Una de las primeras cosas que usted debe aprender es la diferencia entre captura de datos, entrada de datos, y procesamiento de datos. Los medios y métodos alternativos de entradas también deben ser comprendidos antes de diseñar las entradas. Y como la entrada de datos precisa es tan crítica para el procesamiento exitoso, el mantenimiento de archivos, y la salida, usted también deberá aprender acerca de los factores humanos y los controles internos para el diseño de las entradas.
2. ¡Los datos ocurren! Acompañan a los eventos de negocios llamados transacciones. Los ejemplos incluyen órdenes, tarjetas checadoras de tiempo, reservaciones, y cosas por el estilo. Éste es un concepto importante porque los diseñadores de

sistemas deben determinar cuándo y cómo capturar los datos. El diseñador debe entender la diferencia entre lo siguiente:

- a) La captura de datos es la identificación y adquisición de datos nuevos.
  - b) Un documento fuente es una forma de papel que se usa para registrar transacciones comerciales en términos de datos que describen esas transacciones.
  - c) La entrada de datos es el proceso de traducir los datos fuente a un formato legible en la computadora. Ese formato puede ser un disco magnético, una forma óptica de marcar, una cinta magnetofónica, o un disquete flexible, por nombrar unos cuantos.
3. Los datos deben procesarse usando una de las siguientes técnicas:
- a) En el procesamiento por lotes, los datos ingresados son recolectados en archivos llamados lotes que se procesan más tarde.
  - b) En el procesamiento en línea, los datos capturados se procesan inmediatamente.
  - c) En el procesamiento remoto de lotes, los datos se ingresan usando técnicas de edición en línea; sin embargo, los datos son recolectados en lotes para su posterior procesamiento.
4. El analista de sistemas generalmente selecciona el método y el medio para todas las entradas. Los métodos o dispositivos de entrada de datos incluyen:
- El teclado
  - El ratón
  - La pantalla táctil
  - El punto de venta
  - El sonido y la voz
  - La marca óptica

- La tinta magnética
- La firma electromagnética
- Las tarjetas inteligentes
- La biométrica

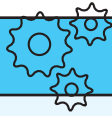
5. La mayoría de las aplicaciones nuevas que se están desarrollando actualmente consisten en pantallas que tienen una apariencia “gráfica”. Este tipo de apariencia se llama interfaz gráfica del usuario (GUI).
6. Las entradas deberán ser tan simples como sea posible y diseñarse para reducir la posibilidad de ingresar datos incorrectos. Además, también debe considerarse la necesidad de contar con capturistas de datos. Con esto en mente, los diseñadores de sistema deberán entender los factores humanos que deberán ser evaluados durante el diseño de las entradas.
7. Los controles de las entradas aseguran que la entrada de datos a la computadora sea correcta y que el sistema esté protegido en contra de errores accidentales e intencionales así como el abuso, incluyendo el fraude.
8. Cuando se diseñen las pantallas de entrada para una aplicación que contenga una apariencia de interfaz gráfica del usuario, el diseñador debe tener el cuidado de seleccionar el objeto correcto de control para cada atributo de entrada. Cada control sirve para un propósito específico, tiene ciertas ventajas y desventajas, y deberá ser usado según los lineamientos. Algunos de los controles más comúnmente usados basados en pantallas para ingresar datos incluyen el text box, el radio button, el check box, el list box, el drop-down list, el combination box, y el spin box.



## Preguntas de repaso

1. ¿Cuál es el objetivo del diseño de entradas?
2. ¿Cuál es la relación entre los documentos fuente y la entrada de datos?
3. ¿Cuál es el siguiente paso después de ingresar los datos? ¿Cuáles son los diferentes métodos que se usan para este paso, y cómo son diferentes en términos del tiempo?
4. ¿Cuáles son los diferentes métodos de ingreso descritos en este libro?
5. ¿Cuál es la diferencia está entre OMR y OCR?
6. Para la entrada biométrica, ¿qué datos se necesitan, cómo se ingresan y procesan en los sistemas de información?
7. ¿Por qué la tecnología de la tarjeta inteligente es capaz de almacenar una gran cantidad de información? ¿Cuáles son algunos ejemplos de las aplicaciones de la tecnología de la tarjeta inteligente?
8. ¿Por qué son importantes los factores humanos para el diseño de las entradas? ¿Qué principios es necesario considerar en el diseño de entradas?
9. ¿Cuáles son algunas de las técnicas que se usan para validar datos?
10. ¿Bajo qué circunstancias deberemos escoger el uso de radio buttons o de check boxes?
11. ¿Cuáles son las similitudes de un drop-down list con un combination box?
12. ¿Cuáles son algunos de los controles avanzados de entrada de datos sugeridos en el presente libro?
13. ¿Cuáles son los pasos del proceso de diseño de entradas?
14. ¿Qué es un documento fuente bien diseñado?
15. ¿Cuáles son los retos que enfrentan las interfaces de Web comparadas con las de *Windows* al diseñar las entradas sugeridas en este libro?

# Problemas y ejercicios



1. ¿Qué meta primordial nunca deberá perder de vista todo diseñador de sistemas al realizar el diseño de entradas?
2. El dueño de una cadena de restaurantes de comida rápida ha contratado a su compañía para diseñar un método para llevarle las órdenes a los clientes más rápido y, con menos esfuerzo, pero sin ninguna pérdida de calidad. Actualmente, los restaurantes de comida rápida usan el método convencional de hacer que los clientes hagan cola para hacer el pedido y pagar; luego la orden es impresa y se entrega al especialista de preparación de comida (el cocinero en jefe). ¿Puede pensar usted cómo la tecnología podría usarse para alcanzar estos objetivos?
3. Observe la siguiente sección de una pantalla de entrada de datos usada por los técnicos en una compañía para ordenar partes del almacén de la misma. ¿Hay algo equivocado con el diseño de esta pantalla de entradas?

INGRESE EL # DEL TÉCNICO: [REDACTED] INGRESE EL NOMBRE

DEL TÉCNICO: [REDACTED]

INGRESE # DE PARTE: [REDACTED] INGRESE LA DESCRIPCIÓN  
DE LA PARTE: [REDACTED]

4. Las terminales de punto de venta (POS), como las que se usan en cajeros automáticos, gasolineras, y en tiendas, se han hecho sumamente comunes por su conveniencia y su versatilidad. Pero en términos de diseño humano de la interfaz, sus métodos de entrada algunas veces dejan algo que desear. ¿Cuáles son las áreas donde usted piensa que son necesarias mejoras?
5. Conteste las siguientes preguntas falsas/verdaderas: Califique o explique sus respuestas según se necesite.
  - a) Los usuarios del sistema tienden a estar confundidos por los códigos de entrada de datos, y frecuentemente ingresan el código equivocado; por consiguiente, su uso debería ser evitado.
  - b) El proceso por lotes es todavía un proceso viable de procesamiento de datos.
  - c) Hay poca correlación en términos de la exactitud de datos entre el punto en el cual los datos se originan y el tiempo antes de la captura de los datos.
  - d) El ratón de la computadora fue inventado para optimizar el uso de interfaces gráficas del usuario en computadoras personales usando sistemas operativos *Windows* o *Apple*.
  - e) El uso del diseño basado en metáforas de las pantallas es considerado demasiado "lindo" y antiprofesional.
  - f) Los radio buttons se usan de la mejor forma sólo cuando hay muy pocos valores previamente definidos que no tienen nada en común.
6. Piense acerca de las mejores y las peores pantallas de entrada de datos que usted ha usado, haya escuchado acerca de ellas, y/o ha trabajado con ellas. Usando su propia experiencia, así como este capítulo,

liste al menos cinco requerimientos de la pantalla de entradas y/o principios (aparte del que se describe en la pregunta 3 que usted considera importantes. Explique por qué seleccionó cada uno de ellos.

7. Usted diseña una pantalla de entradas para un sistema de datos de tratamientos de pacientes que será usado por un condado en su estado. Los proveedores de tratamientos contra droga y alcohol ingresarán datos demográficos y de tratamiento de sus pacientes, luego los enviarán al Departamento de Salud Conductista del condado. El sistema correrá en una red cliente/servidor usando un sistema relacional de gestión de base de datos. Las reglas de negocios requieren una entrada para todos los elementos de datos; no hay campos discretos u optativos. Los datos incluirán una mezcla de datos alfanuméricos, datos numéricos destinados para cálculos, y fechas. Algunos de los campos de datos tendrán interdependencias entre ellos, tales como el campo para el sexo y otro campo de si la paciente está embarazada. Los identificadores únicos del paciente serán utilizados como claves y serán generados por el sistema.
 

¿Dónde deberá diseñar usted los controles y las ediciones de la entrada de datos: en el lado del cliente o en el lado del servidor de la red? ¿Por qué? ¿Qué tipos de ediciones y de verificaciones de validación deberán ser incluidos?
8. Coteje las definiciones o los ejemplos en la primera columna con los términos en la segunda columna:
 

A. Tarjeta con banda magnética	1. Slider edit calendar
B. Single-line text box con dos botones verticalmente alineados	2. La captura de datos
C. Método de procesamiento de datos de recolección y entrega de FedEx	3. La lectura óptica de marcas (OMR)
D. La descripción de Texto de la elección de valores asociados con un círculo	4. Apresurar
E. El sistema de reconocimiento de voz	5. Microsoft <i>Visual Basic</i>
F. La identificación y adquisición de datos nuevos	6. El tipo de tecnología ADC
G. El dispositivo de entradas más propenso a los errores de entrada de datos	7. La fórmula de dígito de comprobación
H. Forma de papel usada para registrar transacción(es) comercial(es)	8. El radio button
I. Ejemplo de diseño basado en metáforas de la pantalla	9. Procesamiento remoto de lotes

- J. formas con marca óptica destinadas para exámenes de preguntas por objetivos
- K. Método no numérico para la selección de valor
- L. Técnica para determinar errores de entrada de datos con teclas primarias
- M. Ejemplo de enfoque de programación basado en repositorios
9. Los text boxes pueden ser el control de uso más frecuente para la entrada de datos en la interfaz gráfica del usuario. ¿Qué convenciones y lineamientos deberá seguir el diseñador de sistemas al diseñar pantallas de entrada que incluyan text boxes?
10. Rellene los espacios vacíos para las siguientes declaraciones:
- Los documentos fuente deberán dividirse en zonas diferentes para datos \_\_\_\_\_, datos \_\_\_\_\_, y dependiendo del formato, para datos que identifiquen un \_\_\_\_\_ único del formato.
  - Después de \_\_\_\_\_ las entradas del sistema y revisar \_\_\_\_\_, el siguiente paso en el proceso \_\_\_\_\_ es \_\_\_\_\_.
  - Los controles de la interfaz gráfica del usuario \_\_\_\_\_ permiten que \_\_\_\_\_ aparezca automáticamente si el número de \_\_\_\_\_ excede el espacio adjudicado para ese \_\_\_\_\_.
  - Aunque técnicamente \_\_\_\_\_ son \_\_\_\_\_ controles de entradas, permiten a los usuarios a \_\_\_\_\_ o a \_\_\_\_\_ una transacción.
10. Sistema biométrico ADC
11. Documento fuente
12. El teclado
13. Spin box
- e) El control de entrada de datos \_\_\_\_\_ es el \_\_\_\_\_, el cual se usa de la mejor manera cuándo el alcance de los valores de los datos de entrada es \_\_\_\_\_.
11. ¿Qué preguntas básicas debería hacerse al decidirse por el mejor control de la interfaz gráfica del usuario para usarse para cada atributo de datos que debe capturarse e ingresarse? Provea ejemplos de atributos de datos, y las instrucciones para cotejar el atributo de datos con el control de la interfaz gráfica del usuario que es el mejor para cada situación.
12. Los diseños basados en metáforas genéricas, fácilmente identificables en general son bien recibidos por los usuarios del sistema, en particular los novatos, porque su familiaridad realza la percepción de ser amigable y fácil usarse. Su compañía quiere reemplazar sus formatos de papel de mensaje telefónico con una versión electrónica que pueda ser mandada como un anexo por correo electrónico. Cree un diseño metafórico de la pantalla para una de las formas comunes de mensaje telefónico de papel. (*Sugerencia:* Esto no requerirá una herramienta de diseño de la pantalla, pero puede diseñarse en *Word* o *Excel* de Microsoft. Cree el suyo, en vez de utilizar una de las plantillas que están comúnmente disponibles.)
13. Dado que Internet continúa creciendo en su influencia en los negocios, cada vez se usa más para aplicaciones que anteriormente eran aplicaciones basadas en cliente/servidor. ¿Cuáles son algunas de las similitudes en el diseño de entradas de datos entre las aplicaciones basadas en la Web y aquellas basadas en el cliente/servidor? ¿Cuáles son algunas de las diferencias? Proporcione ejemplos.

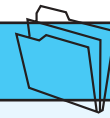


## Proyectos e investigación

1. Como lo menciona el libro de texto, muchas organizaciones utilizaron alguna vez un gran número de capturistas para realizar la entrada de datos. A medida que las computadoras personales y la computación en línea se hicieron más comunes, los usuarios del sistema empezaron a asumir la responsabilidad de la entrada de datos, y las filas de capturistas se encargieron dramáticamente en la mayoría de las organizaciones. Hoy el crecimiento explosivo de Internet está teniendo un impacto similarmente profundo en la estructura organizativa de la mayoría de las empresas grandes y organismos gubernamentales.
- Investigue la transformación en la responsabilidad de la entrada de datos, de capturistas a usuarios del sistema, que ocurrió en las décadas de los ochenta y los noventa. Discuta los aspectos que fueron enfrentaron por estas organizaciones y sus empleados.
  - Investigue la transformación que está tomando lugar actualmente a medida que Internet facilita el crecimiento de la entrada de datos basado en el cliente ¿Cuáles son las implicaciones? ¿Cuál es el impacto en la estructura organizativa de las compañías, sus empleados, y sus clientes?
  - Investigue a una empresa grande o a una dependencia gubernamental en su área. Compare su estructura organizativa hace 25 años con hace 10 años y con la actualidad. ¿Qué cambios pertinentes encontró usted?
  - Indague artículos en revistas de negocios y de tecnología de la información con respecto al impacto que la tecnología está ejerciendo sobre las organizaciones en los sectores privado y público. ¿Cuáles son algunos de los pronósticos con respecto a cuál será la siguiente transformación en la entrada de datos? ¿Piensa usted que estas predicciones se hagan realidad?
  - En general, ¿estamos mejor o peor con estos cambios?

2. A pesar de las incursiones hechas por otros métodos de entrada, el venerable teclado todavía es con mucho el método más común usado para la entrada de datos, y es probable que se quede así por algún tiempo. Pero, ¿podría mejorarse el diseño actual básico del teclado? Indague la historia del teclado, así como también los desarrollos recientes en el diseño del mismo, luego considere las siguientes preguntas:
  - a) ¿Cuál es la base de la disposición “QWERTY”?
  - b) ¿Están disponibles disposiciones más eficientes del teclado? Si es así, ¿por qué no se usan ampliamente?
  - c) Los teclados son una forma de entrada táctil. ¿Hay métodos de entrada de datos táctil aparte de pulsar una tecla que represente una letra o un número?
  - d) ¿Qué hay acerca del impacto de desórdenes repetitivos de tensión nerviosa como el síndrome carpiano del túnel, que puede ser causado o producido por los teclados? ¿Cuál es el costo anual estimado por pérdidas de productividad y cuidados médicos?
  - e) Si se le pidiera rediseñar el teclado, ¿qué cambios haría usted? ¿Por qué?
  
3. La tecnología de reconocimiento de voz se usa con bastante frecuencia para ingresar órdenes o responder a los datos de preguntas automatizados por teléfono. También, algunos expertos de tecnología predicen que el reconocimiento de voz un día reemplazará a los teclados para la entrada de datos; actualmente hay muy pocas aplicaciones que usen la tecnología de reconocimiento de voz para la entrada de datos. Indague desarrollos recientes en la tecnología de reconocimiento de voz y responda a las siguientes preguntas:
  - a) ¿Qué artículos encontró usted? ¿Cuáles son sus puntos de vista en relación con la tecnología de reconocimiento de voz?
  - b) ¿Cuál es el estado del arte actual de la tecnología de reconocimiento de voz?
  - c) ¿Piensa usted que la tecnología de reconocimiento de voz finalmente ha madurado hasta el punto donde pronto podría ser considerada una opción viable para la entrada de datos del teclado? ¿O necesita todavía más maduración? ¿O representa una calle sin salida tecnológica? Explique.
  - d) Si usted tiene acceso a las aplicaciones que incluyen software de reconocimiento de voz para datos o para la entrada de texto (o puede hacer una descarga de una copia libre de prueba sin violar ninguna restricción de otorgamiento de licencias o de uso), intente usarlas. ¿Cómo las evaluaría usted?
  
4. Varias tecnologías emergentes están clasificadas como tecnologías automáticas de captura de datos (ADC). En esencia, retiran a las personas del ciclo de entrada de datos. Una de estas tecnologías es la tecnología de la identificación de frecuencia de radio (RFID), que rápidamente se está haciendo muy común. Investigue la tecnología RFID y las aplicaciones comerciales en la Web y/o en la biblioteca de su escuela.
  - a) Explique cómo operan los dispositivos RFID.
  - b) ¿Cómo se está usando actualmente la RFID en los sectores privado y público?
  - c) ¿Cuáles son sus ventajas principales? ¿Sus desventajas?
  - d) ¿Cuáles son algunas de las implicaciones sociales, económicas, y políticas en relación con la tecnología RFID?
  - e) ¿Cuáles serán algunas de las aplicaciones en las que la tecnología RFID será usada dentro de 10 años?
  
5. Generalmente, uno no tiene que mirar demasiado lejos en la mayoría de las organizaciones para encontrar un documento fuente o pantalla de entrada de datos deficientemente diseñados, o ambos. Contacte una organización sin fines de lucro o similar y ofrézcase como voluntario para revisar sus formatos y las pantallas de entrada de datos, y rediseñe algunos de ellos.
  - a) Describa la organización para la cual usted hizo el trabajo de voluntario.
  - b) ¿Encontró usted cualesquiera problemas de diseño en los documentos fuente o en las pantallas de entrada? Describa los problemas del diseño.
  - c) Describa el documento fuente y/o la pantalla de entradas que usted rediseñó. Incluya una prueba si es posible.
  - d) ¿Qué cambios hizo usted, y qué proceso siguió para hacer los cambios? Incluya una muestra.
  - e) ¿Qué retos afrontó al rediseñar el formato y/o la pantalla?
  - f) ¿Quedó complacida la organización con su rediseño? ¿Está usando la pantalla y los nuevos formatos?
  
6. Su compañía de diseño de Web ha sido contratada por una cadena de supermercados para desarrollar una encuesta corta basada en la Web sobre lo que a los clientes les gusta y no les gusta acerca de la cadena de supermercados. El objetivo es hacer que un empleado en cada tienda camine por la tienda con una computadora portátil y escoja al azar a los clientes para encuestar sus preferencias. A los clientes se les harán de tres a cinco preguntas. Los datos de la encuesta se ingresarán directamente en la computadora portátil por una aplicación basada en la Web.
  - a) ¿Cuáles son algunas de las consideraciones de alto nivel que es necesario encarar antes de diseñar el formato de encuestas?
  - b) ¿Qué preguntas incluiría usted en la encuesta?
  - c) ¿Qué otros datos incluiría usted?
  - d) Diseñe un prototipo del formato.
  - e) ¿Fue esto más fácil o más difícil de lo que usted pensó? Describa cualesquiera retos que usted afrontó que no anticipó.





## Casos breves

1. El diseño de entradas afecta no sólo la facilidad de uso de un sistema, sino también la seguridad. Encuentre ejemplos de cómo los sistemas fallan debido a los caracteres que se ingresan en los text boxes. Investigue cómo enfrentar estos agujeros y debilidades en la seguridad. Prepare un ensayo pequeño sobre este material y preséntelo a la clase.
2. En el caso breve 2 del capítulo anterior, usted creó un formato en línea. ¿Usó apropiadamente los diferentes métodos de entrada, tales como los input boxes, los radio buttons, drop downs, y controles por el estilo? ¿Qué cambiaría usted? ¿Por qué?
3. Haga los cambios que usted sugirió en el caso breve 2. Entregue capturas de pantalla del formato antes y después de que usted hizo los cambios. Asegúrese de incluir el URL de su formato, a fin de que el profesor pueda comprobar su trabajo.
4. Investigue métodos de entrada para los usuarios inadvertidos. Escriba un pequeño ensayo que describa brevemente estos métodos de entrada y cómo los puede integrar usted en un sistema de información.



## Ejercicios de equipo e individuales

1. Debate de mesa redonda: ¿Cómo podemos simplificar el uso de los sistemas de información y computadoras (en general)? Dé ejemplos específicos.
2. Individual: A estas fechas, usted ha hecho una gran cantidad de trabajo de equipo. Recuerde la última vez que alguien perdió la calma o se sintió frustrado. ¿Se amplificó el impacto de las palabras dichas por la cólera o la frustración si fueron comunicadas por email? ¿Piensa usted que deberíamos tener más cuidado en nuestros comentarios cuando están escritos, o cuándo la otra persona no puede leer nuestro lenguaje corporal?
3. Por equipos o individual: Se ha dicho que tener fondos limitados incita a la creatividad. ¿Piensa usted que esto es cierto? Si es así, ¿cómo pueden permanecer una compañía o un individuo creativos y mentalmente hambrientos a medida que se hacen más estables financieramente?



## Lecturas recomendadas

Andres, C. *Great Web Architecture*. Foster City, CA: IDG Books Worldwide, 1999. Los libros sobre el diseño efectivo de la interfaz de Red están comenzando a aparecer. La ciencia de la ingeniería humana para las interfaces de Red todavía no ha progresado tanto como las interfaces cliente/servidor (por ejemplo, *Windows*). Aquí hay un título de los primeros que explora muchas dimensiones de la arquitectura y las interfaces de Red que usan ejemplos de la vida real.

*Application Development Strategies* (periódico mensual). Arlington, MA: Cutter Information Corporation. Éste es nuestro periódico favorito orientado al tema que sigue las estrategias, metodologías, CASE y otras tendencias relevantes del desarrollo de sistemas. Cada aspecto se centra en un solo tema. Este periódico suministrará un buen cimientito sobre cómo desarrollar prototipos de entradas.

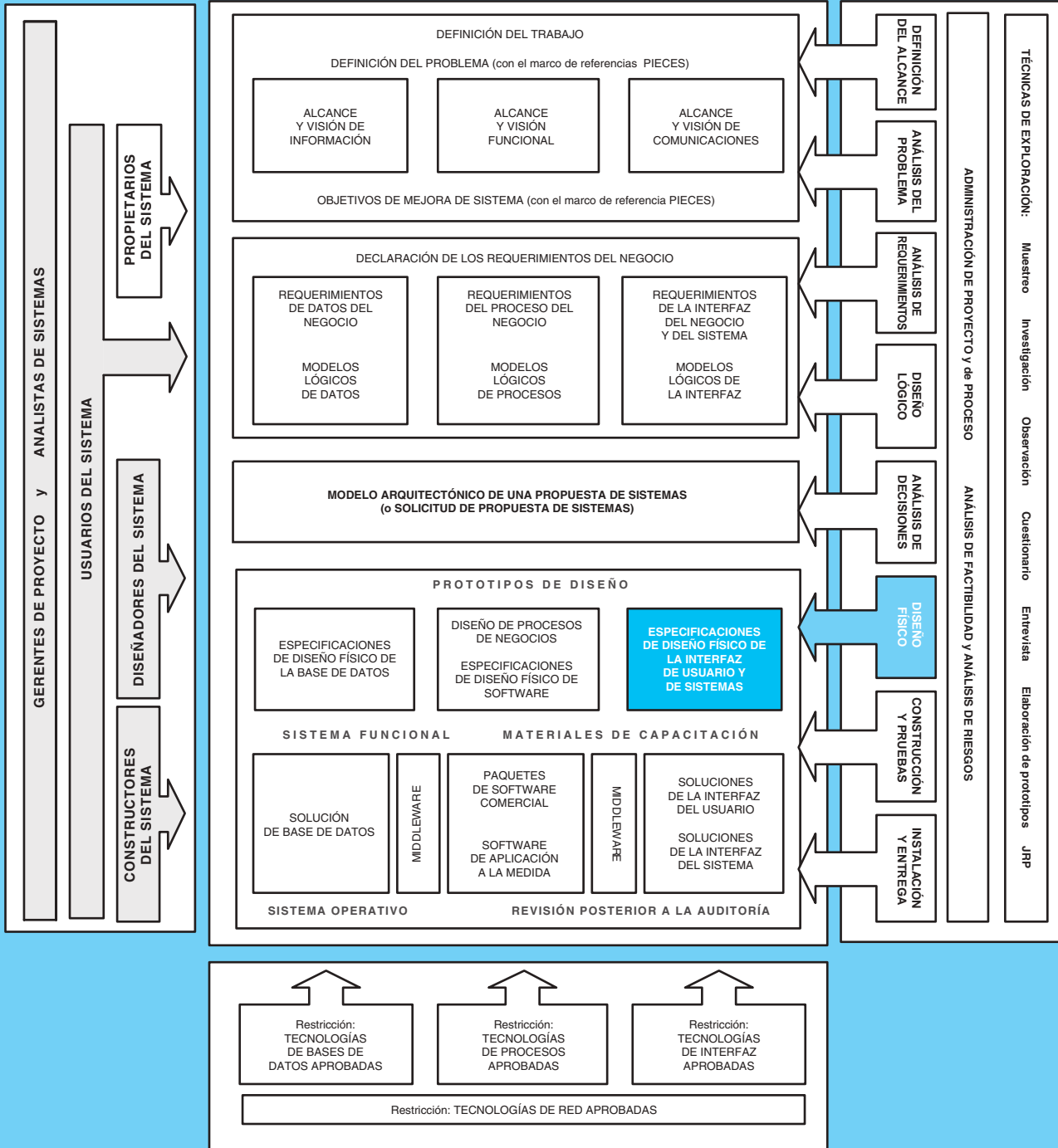
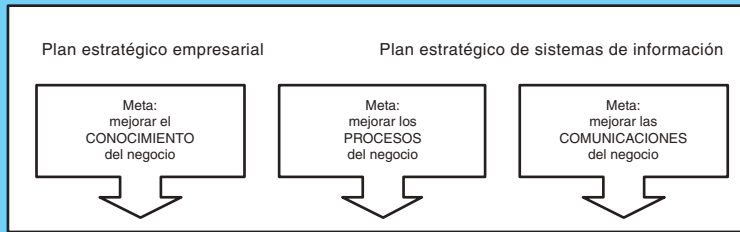
Dunlap, Duane. *Understanding and Using ADC Technologies. A White paper for the ADC Industry*. A SCAN TECH 1995 Presentation. 23 de octubre de 1995, Chicago. Estamos en deuda con nuestro amigo y colega. El profesor Dunlap es un líder en

el campo de ADC. Este artículo fue la base de gran parte de nuestra discusión sobre las tendencias en la tecnología ADC.

Fitzgerald, Jerry. *Internal Controls for Computerized Information Systems*. Redwood City, CA: Jerry Fitzgerald & Associates, 1978. Ésta es nuestra referencia estándar sobre el tema del diseño de los controles internos en los sistemas. Fitzgerald invoca una herramienta matricial única y poderosa para el diseño de los controles. Este libro va mucho más lejos que cualquier libro de texto de introducción a los sistemas; su lectura es obligada.

Galitz, W. O. *User-Interface Screen Design*. Nueva York: John Wiley & Sons, 1993. Éste es nuestro libro favorito sobre el diseño general de las interfaces del usuario. El autor ofrece varios diagramas de flujo del proceso de decisiones en la aplicación de los controles GUI para las entradas.

Kozar, Kenneth. *Humanized Information Systems Analysis and Design*. Nueva York: McGraw-Hill, 1989. Un buen tratamiento orientado al usuario del diseño de entradas.



# 15

## Diseño de la interfaz del usuario

### Panorámica y objetivos del capítulo

En este capítulo usted aprenderá a diseñar y a hacer prototipos de la interfaz del usuario para un sistema. La interfaz del usuario debe suministrar un medio amigable mediante el cual el usuario pueda interactuar con la aplicación para procesar entradas y obtener salidas. En los capítulos 13 y 14, usted aprendió cómo diseñar y hacer prototipos de las salidas y las entradas. El diseño y la elaboración de prototipos de la interfaz del usuario determinan la presentación general de la aplicación, lo cual puede requerir revisiones de los prototipos preliminares de salida y entrada. Actualmente existen dos interfaces que encontramos comúnmente: las terminales (o las microcomputadoras que actúan como terminales) que se usan conjuntamente con las mainframes y los monitores de exhibición más comunes conectados a las microcomputadoras. También hay varios estilos de estrategia para diseñar la interfaz del usuario para los sistemas. Reconocer que ha dominado el diseño de la interfaz del usuario cuando usted pueda:

- Distinguir entre diferentes tipos de usuarios de computadora y consideraciones de diseño para cada uno.
- Identificar varios factores y lineamientos importantes de la ergonomía e incorporarlos en el diseño de una interfaz del usuario.
- Integrar el diseño de salidas y entradas a una interfaz del usuario general que establezca una comunicación entre los usuarios y la computadora.
- Entender el papel de los sistemas operativos, los navegadores de la Red y otras tecnologías para el diseño de la interfaz del usuario.
- Aplicar estrategias apropiadas de la interfaz del usuario a un sistema de información. Usar un diagrama de transición de estados para planear y coordinar la interfaz del usuario para un sistema de información.
- Describir cómo puede usarse la creación de prototipos para diseñar una interfaz de usuario.

## Introducción

El diseño de la interfaz del usuario para la parte interna de cliente/servidor del sistema SoundStage Member Services ha sido un proceso más largo que lo que esperaba Bob Martínez. Parte del diseño final se muestra en las figuras 15.17, 15.18 y 15.19, pero al principio no se veía de esa manera. El primer diseño de Bob usaba iconos en lugar de texto para todos los botones del menú. Por ejemplo, una imagen de una pila de papeles indicaba las “órdenes”, y un cuadro de globos indicaba las “promociones”. Pero la retroalimentación de los usuarios que probaban la interfaz reveló que muchos de ellos no tenían ninguna pista del significado de estos iconos y desperdiciaban tiempo tratando de entenderlos. Así es que se tomó la decisión de cambiar a las etiquetas con texto. Se probaron varios tipos de controles para las pantallas mostradas en las figuras 15.18 y 15.19. Cada iteración pasó por las pruebas del usuario, llevando a revisiones y a más pruebas del mismo. Fue una gran cantidad de trabajo. Pero Bob sabía que este trabajo en el front end sería mucho menos que el de contestar miles de llamadas telefónicas de usuarios confundidos y alterados, si implementaban una interfaz del usuario que fuera deficiente.

## Conceptos y lineamientos del diseño de la interfaz del usuario

En los dos capítulos previos, nos ocupamos del diseño de las salidas y de las entradas. En este capítulo, integramos el diseño de las salidas y de las entradas en una interfaz de usuario global que establezca la comunicación entre los usuarios y la computadora. Esta comunicación determina todo, desde iniciar el sistema o introducirse a él, establecer opciones y preferencias, hasta obtener ayuda. Y la presentación de las salidas y las entradas es también parte de la interfaz. Necesitamos examinar las transiciones que pueden ocurrir de pantalla a pantalla. En las aplicaciones de cliente/servidor (por ejemplo *Windows* basado en redes) y aplicaciones de Web (por ejemplo, navegadores basados en Internet o en intranet), el usuario tiene muchos caminos alternativos a través de menús, hipervínculos, diálogos, y cosas por el estilo. Esto propicia interfaces de usuario más fáciles y amigables, pero complica enormemente el diseño y la programación.

Hoy en día, la mayoría de las interfaces de usuario son diseñadas por prototipos que se construyen rápidamente. Éstos son generados usando ambientes rápidos de desarrollo de aplicaciones tales como *Visual Studio* de Microsoft, *JBuilder* de Borland (para *Java*), o *VisualAge* de IBM (para diferentes lenguajes). Estos prototipos rara vez son **totalmente** funcionales, pero contienen bastante funcionalidad para demostrar la interfaz. Por ejemplo, un prototipo de sistema de ayuda puede ser funcional hasta el grado de solicitar algunas pantallas de muestra para exponer los niveles de ayuda. O un sistema de seguridad podría tener la suficiente funcionalidad para demostrar errores de acceso representativos aun sin la autenticación de los usuarios. Cuando nos acerquemos a la fase de la construcción del ciclo de vida, los programadores y los analistas completarán la funcionalidad.

Empezamos nuestro estudio examinando los tipos de usuarios, los factores humanos, y los lineamientos de ingeniería que afectan el diseño de la interfaz del usuario.

### > Tipos de usuarios de computadoras

En ninguna fase los factores humanos son tan importantes como en el diseño de la interfaz del usuario. Simplemente pregúntele al analista de sistemas típico que pasa la mitad del día contestando llamadas telefónicas de los usuarios que tienen dificultades para usar los sistemas de información y las aplicaciones de computadora. La consideración primordial del diseño de la interfaz del usuario es la misma que para la escritura técnica y de negocios: entender a su audiencia. En el diseño de la interfaz, la audiencia es el USUARIO DEL SISTEMA.

En la página inicial de este capítulo, resaltamos a LOS ANALISTAS DEL SISTEMA, LOS DISEÑADORES DEL SISTEMA, y LOS CONSTRUCTORES DEL SISTEMA como los actores que en realidad **realizan** el diseño de la interfaz del usuario, pero también resaltamos a LOS USUARIOS DEL SISTEMA como los actores para los cuales se destina el diseño de la interfaz del usuario. Los usuarios deben **probar** y **evaluar** ese diseño. Dicha página también ilustra que el diseño de la interfaz del usuario se realiza durante las fases de DISEÑO y CONSTRUCCIÓN y que las actividades conducen a las ESPECIFICACIONES DE LA INTERFAZ (los prototipos) y los bloques de construcción DEL PROGRAMA.

Los USUARIOS DEL SISTEMA pueden clasificarse ampliamente como expertos o novatos; y también como no discrecionales o discrecionales.

Un **usuario experto** es un usuario experimentado de la computadora que ha pasado un tiempo considerable usando programas de aplicación específicos. Generalmente se considera que su uso de la computadora es no discrecional. En la era de computación de la mainframe, a esto se le llamaba un *usuario reservado*. Los usuarios expertos generalmente están cómodos con (pero no necesariamente son expertos en eso) el ambiente operativo de la aplicación (por ejemplo, *Windows* o un navegador de la Red). Han invertido tiempo en aprender a usar la computadora. Invertirán tiempo en superar las dificultades de las interfaces de usuario menos amigables. En general, han aprendido de memoria operaciones de rutina hasta el punto de ya no buscar ni requerir retroalimentación e instrucciones excesivas de la computadora. Quieren poder cumplir con su tarea con tan pocas acciones y teclazos como sea posible.

El **usuario novato** (algunas veces llamado *usuario informal*) es un usuario de la computadora menos experimentado que generalmente usa la computadora sobre una base menos frecuente, u ocasional. El uso de una computadora puede verse como discrecional (aunque esto se está haciendo cada vez menos verdadero). Dicho de una manera sencilla, los usuarios novatos necesitan más ayuda que los usuarios expertos. La ayuda toma muchas formas, incluyendo menús, diálogos, instrucciones, y pantallas de ayuda. La mayoría de los gerentes, a pesar de su creciente buen conocimiento en el uso de computadoras, se sitúan en la categoría de los novatos. Se les paga para reconocer y solucionar problemas, sacar provecho de oportunidades, crear planes y administrar la visión; no para aprender y usar computadoras. Las computadoras son consideradas herramientas por los gerentes modernos. Cuando surge la necesidad, quieren darse cuenta de su beneficio tan rápidamente como les sea posible.

Los usuarios expertos y novatos son de hecho extremos en el conjunto de todos los usuarios. El usuario completamente novato que no ha usado una computadora se está volviendo menos común. Pocos planes de estudio universitarios no requieren de un buen conocimiento en el uso de computadoras para todas sus carreras, y los estudiantes de todas estas profesiones han descubierto el valor incrementar su experiencia interdisciplinaria en el uso de la computadora (algunas veces llamada *informática*). Los usuarios novatos generalmente se transforman en usuarios expertos a través de la práctica y la experiencia. El impacto de Internet sobre la sociedad se debe a que un número mayor de personas están adaptándose cada vez más al uso de las computadoras; creando cada año que pasa una clase de usuarios que son menos novatos y más expertos. ¿Es de causar admiración que el diseño de la interfaz del usuario está encaminándose más hacia las interfaces de Web, al parejo que las aplicaciones de *Windows*?

Es difícil imaginar que los jóvenes estudiantes y los profesionales de hoy se sientan incómodos con las computadoras. Independientemente de todo, la mayoría de los sistemas de hoy son diseñados para el usuario novato de sistemas pero adaptados al usuario experto. El enfoque recae en la facilidad de uso o en la ergonomía.

## > Los factores humanos

Antes de diseñar las interfaces del usuario, usted puede encontrar útil entender los elementos que frecuentemente causan que las personas tengan dificultades con los sistemas de cómputo. Nuestro experto favorito de diseño de interfaces del usuario, Wilbert Galitz (vea las Lecturas recomendadas) plantea los siguientes problemas de la interfaz:

- El uso excesivo de tecnicismos y acrónimos de la computadora.
- Diseño que no es obvio o menos que intuitivo.
- La incapacidad para distinguir entre acciones alternativas (“¿qué hago después?”).
- Enfoques inconsistentes de la resolución de problemas.
- Inconsistencias en el diseño.

Según Galitz, estos problemas dan como resultado confusión, pánico, frustración, aburrimiento, uso indebido, abandono, y otras consecuencias indeseables.

Para solucionar estos problemas, Galitz propone los siguientes “compromisos” indeseables del diseño de interfaces del usuario:

- *Entienda a sus usuarios y sus tareas*. Esto se hace cada vez más difícil a medida que ampliamos nuestros sistemas de información al implementar la funcionalidad de negocio a consumidor (business-to-consumer, B2C) y de negocio a negocio (business-to-business, B2B) usando Internet.

---

**usuario experto** Usuario de computadoras experimentado.

---

**usuario novato** Usuario de computadoras inexperto u ocasional.



- *Involucre al usuario en el diseño de la interfaz.* Descubra lo que a los usuarios les gusta y lo que les desagrada en sus aplicaciones actuales. Involúcrelos en el diseño de la pantalla y dialogue desde el principio. Este lineamiento se habilita fácilmente con la base de datos actual de la PC y con la tecnología rápida de desarrollo de aplicación.
- *Pruebe el sistema en usuarios reales.* La observación y la escucha son las habilidades cruciales aquí. Después de la capacitación inicial, procure evitar la capacitación excesiva y obligar a los usuarios a que aprendan el sistema. En lugar de eso, observe sus acciones y sus errores, y escuche sus comentarios y preguntas para mejor entender su interacción con la interfaz de usuario.
- *Practique el diseño iterativo.* La primera interfaz del usuario probablemente será poco satisfactoria. Espere que cualquier diseño de la interfaz de usuario experimente múltiples iteraciones en cuanto al diseño y pruebas. ¿Cuándo está terminada la interfaz? ¡Probablemente nunca! Pero Galitz sugiere que una buena meta es que 95 por ciento de los usuarios típicos (sean novatos o expertos) puedan realizar las tareas requeridas (sean de rutina o menos comunes) sin dificultad ni ayuda.

### > Las directrices de la ergonomía

Dado el tipo de usuario, deberá incorporarse en el diseño un número importante de factores de ergonomía:

- *El usuario del sistema siempre deberá darse cuenta de qué hacer después.* El sistema siempre deberá proveer instrucciones en relación a la forma de proceder, respaldar, salir, y cosas por el estilo. Varias situaciones requieren algún tipo de retroalimentación:
  - *Diga al usuario qué espera del sistema en este momento.* Esto puede plasmarse en un mensaje simple como LISTO, TECLEE UN COMANDO, SELECCIONE UNA O MÁS OPCIONES, O TECLEE DATOS.
  - *Diga al usuario que los datos han sido ingresados correctamente.* Esto puede ser tan simple como mover el cursor al siguiente campo o exhibir un mensaje tal como DATOS CORRECTOS.
  - *Diga al usuario que los datos no han sido ingresados correctamente.* Se prefieren mensajes pequeños y simples acerca del formato correcto. Las funciones de ayuda pueden sustituir estos mensajes con ejemplos e instrucciones más extensas.
  - *Explíquelo al usuario la razón de un retraso en el procesamiento.* Algunas acciones requieren varios segundos o minutos para completarse. Los ejemplos incluyen la ordenación, la indexación, la impresión, y la actualización. Mensajes simples tales como ORDENANDO; POR FAVOR ESPERE, O HACIENDO LA INDEXACIÓN; ESTO PUEDE TOMAR ALGUNOS MINUTOS O POR FAVOR ESPERE le dicen al usuario que el sistema no ha fracasado. El reloj de arena de *Windows* o el globo giratorio de *Internet Explorer* son pistas de icono de que un proceso está ocurriendo.
  - *Diga al usuario que una tarea fue completada o no lo fue.* Esto es especialmente importante en caso de que un proceso se atrase, sin embargo, lo es también en otras situaciones. Un mensaje tal como LA IMPRESIÓN ESTÁ TERMINADA O LA IMPRESORA NO ESTÁ LISTA; INTÉNTELO DE NUEVO O PÓNGASE EN CONTACTO CON SU ADMINISTRADOR DE RED será suficiente.
- *La pantalla deberá formatearse de modo que los diversos tipos de información, instrucciones y mensajes siempre aparezcan en la misma área de exhibición general.* Así, el usuario de sistemas sabe aproximadamente dónde buscar información específica. En la mayoría de los ambientes de *Windows*, los estándares a menudo dictan la posición de los mensajes de estatus o las ventanas de diálogo de aparición automática.
- *Los mensajes, las instrucciones, o la información deberán exhibirse el tiempo suficiente para permitirle al usuario del sistema leerlos.* La mayoría de los expertos recomiendan que los mensajes importantes se exhiban hasta que el usuario los reconozca.
- *Use con moderación los atributos de exhibición.* Los atributos de visualización, como parpadeo, resalte, y el video inverso pueden distraer al usuario si se usan demasiado.

El uso juicioso le permite llamar la atención hacia algo importante; por ejemplo, el siguiente campo que debe ingresarse, un mensaje o una instrucción.

- *Deberán especificarse los valores por omisión para los campos y las respuestas que el usuario debe ingresar.* En ambientes de ventanas, los valores válidos frecuentemente se presentan en una ventana separada o en una caja de diálogo (*dialogue box*) como una región con desplazamiento (*scrollable region*). El valor predeterminado, si es aplicable, usualmente deberá ser el primero y estar claramente resaltado.
- *Anticipe los errores que los usuarios podrían cometer.* Los usuarios del sistema cometerán errores, aun cuando les proporcionen las instrucciones más obvias. Si es posible que el usuario ejecute una acción peligrosa, hágase saber (por ejemplo, en un mensaje o *dialogue box* se podría leer ¿ESTÁ USTED SEGURO QUE QUIERE SUPRIMIR ESTE ARCHIVO?). ¡Un gramo de prevención lo hará llegar muy lejos!
- *Con relación a los errores, a un usuario no deberá permitírsele continuar sin corregir un error.* Las instrucciones (y los ejemplos) en relación a la forma de corregir el error deberán desplegarse. El error puede ser resaltado con sonido o color y luego puede explicarse en una *pop-up window* o un *dialogue box*. Puede definirse una opción de AYUDA para provocar el despliegue de instrucciones adicionales.
- *Si el usuario hace algo que podría ser catastrófico, el teclado deberá bloquearse para impedir cualquier otra entrada, y deberá exhibirse una instrucción para llamar al analista o al soporte técnico.*

## > Terminología y tono del diálogo

El flujo total de pantallas y mensajes se llama **diálogo**. El tono y la terminología de un diálogo son factores humanos muy importantes en el diseño de la interfaz del usuario. Con relación al tono del diálogo, se ofrecen los siguientes lineamientos:

- *Use oraciones simples, gramaticalmente correctas.* Es mejor usar lenguaje conversacional en vez de lenguaje formal escrito.
- *¡No se haga el chistoso ni el lindo!* Cuando alguien tiene que usar el sistema 50 veces al día, el humor pretendido rápidamente se desgasta.
- *No sea condescendiente.* No insulte la inteligencia del usuario del sistema. Por ejemplo, no le ofrezca alabanza o recompensas repetidas.

Con relación a la terminología usada en un diálogo de la computadora, las siguientes sugerencias pueden resultar útiles:

- *No use tecnicismos de computación.*
- *Evite la mayoría de las abreviaturas.* Las abreviaturas dan por supuesto que el usuario entiende cómo traducirlas. ¡Verifique primero!
- *Use términos simples.* Use NO ES CORRECTO en lugar de INCORRECTO. Hay menos oportunidad de una lectura errónea o de una mala interpretación.
- *Sea consistente en el uso de la terminología.* Por ejemplo, no use EDITE y MODIFIQUE para indicar la misma acción.
- *Redacte las instrucciones cuidadosamente; use verbos apropiados de acción.* Las siguientes recomendaciones deberán ser útiles:
  - Use SELECT o CHOOSE en lugar de PICK cuando se refiera a una lista de opciones. Asegúrese de indicar si el usuario puede seleccionar sólo una o más de una opción de la lista de opciones disponibles.
  - Use TYPE, NO ENTER, para pedir al usuario que ingrese instrucciones o datos específicos. El término ENTER puede confundirse con la tecla de ingreso.
  - Use PRESS, NO HIT O DEPRESS, para referirse a las acciones del teclado. Siempre que sea posible, refiérase a las teclas por los símbolos o los identificadores que, de hecho, están impresos en las teclas. Por ejemplo, el símbolo ↵ se usa en algunos teclados para designar a la tecla RETURN o ENTER.
  - Cuando se refiera al cursor del ratón en la pantalla, use el término POSITION THE CURSOR (SITÚE EL CURSOR), NO POINT THE CURSOR.

**diálogo** El flujo total de pantallas y mensajes para una aplicación.



## Tecnología de la interfaz del usuario

La mayoría de las interfaces de usuario de hoy son gráficas. La estructura básica de la *interfaz gráfica del usuario* (o *GUI*) está provista dentro del sistema operativo de la computadora o el navegador de Internet de elección. En *los sistemas de información cliente/servidor*, el cliente de la interfaz del usuario está implementado para ejecutarse dentro del sistema operativo de la PC. En *los sistemas de información de Internet y de la intranet*, la interfaz del usuario está implementada para ejecutarse dentro del navegador de Web de la PC (la cual, a su vez, se ejecuta dentro del sistema operativo de la PC).

### > Sistemas operativos y navegadores Web

El sistema operativo dominante basado en interfaces gráficas del usuario para computadoras del cliente de hoy (como en una red de cliente/servidor) es *Windows* Microsoft (versiones diversas). *Macintosh* de Apple y los sabores diversos de *UNIX* (incluyendo a *Linux*) también sustentan participación en el mercado. Para el creciente número de computadoras cliente de bolsillo y de palma de la mano, el sistema operativo dominante actual es el *Palm OS* de Palm. Microsoft *Windows Mobile* también sustenta un porcentaje de ese mercado.

Es cada vez más evidente que el sistema operativo no es el factor crucial de tecnología en el diseño de la interfaz del usuario. Las aplicaciones de Internet y de la intranet corren dentro de un navegador de Web. La mayoría de los navegadores corren en muchos sistemas operativos, posibilitando el diseño de una interfaz de usuario que es menos dependiente en la computadora misma. Las ventajas de esta *independencia de la plataforma* de la computadora deberían ser obvias. En lugar de escribir una interfaz de usuario para cada plataforma de computación y sistema operativo esperados, usted lo escribe para uno o dos navegadores. Al momento de imprimir este libro, los navegadores de Web dominantes son Microsoft *Internet Explorer* y Mozilla *Firefox*, sin embargo, pueden existir problemas con las versiones dentro de los navegadores en la comunidad del usuario.

Además de los sistemas operativos y navegadores, el diseño global de una interfaz de usuario es resaltado o restringido por las características disponibles del monitor de despliegue de los usuarios, el teclado, y los dispositivos de apuntado. Examinemos brevemente algunas de las otras consideraciones.

### > Monitor de pantalla

El tamaño del área de despliegue es crítico para el diseño de la interfaz de usuario. ¡No todas las pantallas son monitores de la PC! Todavía existe un número de *terminales* que no son PC. Las terminales son pantallas que no son de PC que meramente exhiben datos y la información transmitida por una computadora remota, generalmente una mainframe. Y aun cuando muchas terminales han sido reemplazadas por PC, con frecuencia los usuarios se ven todavía forzados a interactuar con las aplicaciones heredadas de mainframe que usan *terminales emuladoras* que abren una ventana en la pantalla donde se despliega información e instrucciones en el formato original de una terminal con una versión anterior de *Windows*. Para estas terminales y emuladores, las dos áreas de despliegue más comunes eran 25 líneas por 80 columnas y 25 líneas por 132 columnas.

Afortunadamente, el monitor de la computadora personal ha reemplazado la mayoría de las terminales, y la mayoría de aplicaciones más nuevas y rediseñadas están escritas para una interfaz gráfica. Para los monitores de PC, no medimos el desplegado en términos de líneas y columnas. Y mientras las medidas diagonales como las pulgadas son a menudo citadas, la medida más pertinente es la resolución gráfica. La resolución gráfica se mide en píxeles, el número de puntos bien definidos de luz desplegados en la pantalla. La resolución más común de hoy es 1 024 000 píxeles horizontales por 800 000 píxeles verticales en un despliegue diagonal de 17 pulgadas. Los mayores tamaños de despliegue soportan aun más píxeles; sin embargo, el diseñador generalmente deberá diseñar la interfaz de usuario con la suposición del mínimo denominador común o razonable.

Obviamente, las computadoras de bolsillo y de la palma de la mano y los despliegues especializados de terminal (como los de las ventanillas de cajero y ATM) soportan despliegues mucho más pequeños que deben ser considerados en el diseño de la interfaz del usuario.



La manera en la cual el área de despliegue se muestra al usuario es controlada por las capacidades técnicas del despliegue y las capacidades del sistema operativo. La paginación y el desplazamiento en pantalla son los dos enfoques más comunes para mostrarle al usuario el área de despliegue. La **paginación** despliega una pantalla completa de caracteres a la vez. El área completa de exhibición se conoce como una *página* (o *pantalla*). La página es reemplazada a petición por la página siguiente o la anterior, de manera muy parecida a hojear las páginas de un libro. El **desplazamiento en pantalla** mueve la información exhibida hacia arriba o hacia abajo en la pantalla, una línea a la vez. Esto es similar a la manera en que los créditos en la televisión y en las películas se desplazan hacia arriba de la pantalla al final de una película. Una vez más, los despliegues de la PC le ofrecen un rango más amplio de opciones de paginación y de desplazamiento en la pantalla.

## > Teclados y punteros

La mayoría (pero no todas) de las terminales y de los monitores están integrados con teclados. La excepción obvia es la computadora de palma de mano como el PalmPilot. Las características críticas del teclado incluyen el conjunto de caracteres y teclas de funciones.

El conjunto de caracteres de la mayoría de los teclados de la PC es bastante estándar. Estos conjuntos de caracteres pueden ampliarse con software para soportar caracteres y símbolos adicionales. Para las terminales especializadas o las estaciones de trabajo, el fabricante puede diseñar teclados hechos a la medida. La mayoría de los teclados contienen teclas especiales designadas *teclas de funciones*. Los teclados de la PC usualmente tienen 12 de estas teclas de funciones. Se ha sabido de terminales que incluyen hasta 32 teclas de funciones. Las **teclas de función** (generalmente rotuladas F1, F2, etcétera) pueden usarse para programar ciertas operaciones comunes, repetitivas en una interfaz de usuario (por ejemplo, HELP, EXIT, y UPDATE). En un sistema operativo, las teclas de función son a menudo predefinidas, pero los desarrolladores de aplicaciones las pueden hacer a la medida para los sistemas específicos. Las teclas de función deberán usarse consistentemente. Es decir, los programas de cualquier sistema de información deberán destinar siempre las mismas teclas de función a los mismos propósitos. Por ejemplo, F1 comúnmente se utiliza como la tecla de ayuda tanto en los sistemas operativos como en las aplicaciones.

La mayoría de GUI (incluyendo sistemas operativos y navegadores) usan dispositivos de apuntado que incluyen ratones, plumas, y pantallas táctiles. Obviamente, el puntero más común es el ratón. Un **ratón** es un dispositivo pequeño del tamaño de la mano que se asienta sobre una superficie plana cerca de la terminal. Tiene un rodillo pequeño en la parte inferior. A medida que usted mueve el ratón en la superficie plana, hace que el puntero se mueva a través de la pantalla. Los botones en el ratón le permiten seleccionar objetos o comandos para los cuales el cursor ha sido movido. Impulsados por la necesidad de desplazarse en las páginas Web y otros documentos, muchos ratones incluyen ahora una rueda que le permite al usuario desplazarse más fácilmente en las páginas y documentos sin usar las barras de desplazamiento.

Las *plumas* se están volviendo importantes en aplicaciones que usan dispositivos de mano (como PalmPilots). Como tales dispositivos frecuentemente no incluyen teclados, puede ser necesario diseñar la interfaz de usuario para permitir “teclear” en un teclado desplegado en la pantalla o usar un estándar de escritura a mano como *Graffiti* o *Jot*. Existen componentes prefabricados para implementar estas características comunes.

Como se observó previamente, la interfaz de usuario más común es gráfica ya sea basada en *Windows* o en navegadores de la Web. El resto de este capítulo enfocará la atención en el diseño gráfico de la interfaz de usuario.

---

**paginación** Despliegue de una pantalla completa de caracteres a la vez.

---

**desplazamiento en pantalla** Despliegue de información en los extremos superior o inferior de la pantalla, línea por línea.

---

**teclas de función** Conjunto de teclas especiales del teclado que se usa para programar operaciones especiales.

---

**ratón** Dispositivo usado para hacer que un puntero se mueva en una pantalla.

## Estilos y consideraciones de la interfaz gráfica del usuario

El diseño de la interfaz de usuario es la especificación de un diálogo o una conversación entre el usuario del sistema y la computadora. Este diálogo generalmente conduce al ingreso de datos y a la salida de información. Hay varios estilos de interfaces gráficas del usuario. Tradicionalmente estos estilos se veían como alternativas, pero cada vez están más mezclados. Esta sección presenta una visión general de varios estilos o estrategias diferentes destinados a diseñar interfaces gráficas del usuario y cómo son incorporados en las aplicaciones de hoy. Demostraremos estos estilos con aplicaciones de software populares.

## > Ventanas y marcos

La construcción básica de una interfaz gráfica de usuario (basada tanto en el sistema operativo como en el navegador) es la *ventana*. Una ventana es un área rectangular con borde. Se exhibe un título (y opcionalmente un nombre de archivo) en lo alto de cada ventana.

Una ventana puede ser más pequeña o más grande que el área real visible del monitor de despliegue. Usualmente incluye controles estandarizados en la esquina superior derecha para *maximizarse* hasta el tamaño de la pantalla de despliegue, *minimizarse* hasta un icono (en la parte inferior de la pantalla), alternarse con un tamaño previo, y *salir* (o *cerrar*).

El archivo, la forma, o el documento exhibido dentro de la ventana puede o no encajar en ella. Cuando el archivo, la forma, o el documento excede el tamaño de la ventana, se usan las *barras de desplazamiento* en el lado derecho y en la parte inferior de la ventana para navegar en ese archivo, esa forma, o ese documento e indicar la posición del cursor actual relativa al archivo, formato, o documento completo.

Una ventana puede dividirse en zonas designadas *marcos*. Cada marco puede actuar independientemente de los otros marcos en la misma ventana, usando características tales como la paginación, el desplazamiento en pantalla, los atributos de visualización, y el color. Cada marco puede definirse para servir a un propósito diferente. Los marcos son comunes tanto en *Windows* como en los navegadores de Web.

Dentro de una ventana o un marco, usted puede usar todos los *controles de la interfaz del usuario* que se usaron en los dos capítulos previos (tales como *text boxes*, *radio buttons*, *check boxes*, *drop-down lists*, *buttons*, etc.). Adicionalmente, muchos otros tipos de controles de interfaz del usuario se introducirán más tarde en el capítulo.

Finalmente, una ventana frecuentemente tiene una *barra de tareas* en la parte inferior de la ventana. Esta barra de tareas puede usarse para desplegar mensajes, avances, o herramientas especiales (para discutirse posteriormente).

## > Las interfaces operadas por menú

La estrategia de diálogo más antigua y más comúnmente utilizada es la selección por menú. Se ofrecen diferentes tipos de menús a los usuarios novatos y expertos. Las estrategias **operadas por menú** requieren que el usuario seleccione una acción a partir de un menú de alternativas.

Los diálogos operados por menú anteceden a las GUI. Un menú jerárquico típico pre-GUI se ilustra en la figura 15.1. Las opciones del menú pueden ser lógicamente agrupadas en opciones de alto nivel para simplificar la presentación. Como se muestra en la figura, si se selecciona la opción del menú principal DISPLAY WARRANTY REPORTS (EXHIBA LOS REPORTES DE GARANTÍA), aparecerá el submenú WARRANTY SYSTEM REPORT MENU (MENÚ DE REPORTES DEL SISTEMA DE GARANTÍA). Luego, si se selecciona la opción PART WARRANTY SUMMARY (RESUMEN DE LA GARANTÍA DE PARTES), se despliegan consecutivamente las pantallas de impresión de reportes y de reportes a la medida. No hay límite técnico a la profundidad de anidado de los menús jerárquicos. Sin embargo, mientras más profundo sea el anidamiento, mayor será la necesidad de crear trayectorias directas para las opciones del menú que estén profundamente anidadas para el usuario experto, quien puede encontrar frustrante el navegar a través de múltiples niveles (lo que se conoce como *screen thrashing*). Asimismo, la mayoría de los usuarios también requieren maneras de regresar a los submenús principales o de niveles más altos sin volver hacia atrás a través de cada una de las pantallas originales.

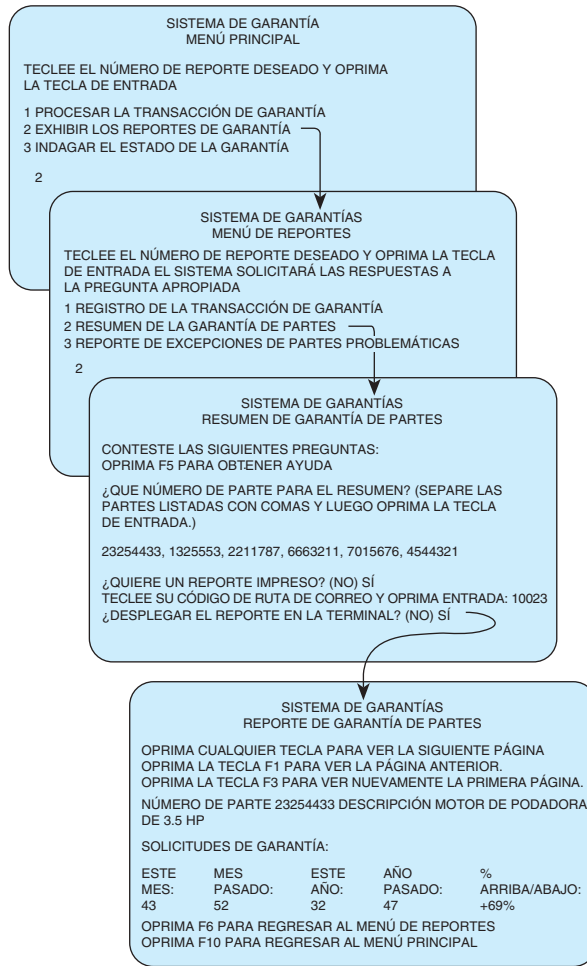
Los menús jerárquicos pre-GUI fueron relativamente fáciles de diseñar. Una gráfica de diálogo como la mostrada en la figura 15.2 (tomada de una edición anterior de nuestro libro) se usó para mapear las transiciones de pantalla a pantalla y asegurar consistencia e integridad. Pero la llegada de las interfaces gráficas del usuario complicó enormemente el diseño del menú.

En los GUI de los sistemas operativos tales como Microsoft *Windows*, los diálogos del usuario no son jerárquicos. ¡Piense acerca de eso! Entre el tiempo que usted inicia y accesa a una aplicación de *Windows*, tal como su procesador de texto, el número de trayectorias y acciones diferentes que usted sigue a través de su aplicación parece no tener fin aparentemente. ¡El diálogo no es nada jerárquico! Este diseño de diálogo no puede modelarse tan fácilmente mediante diagramas de diálogo basados en una jerarquía. Examinemos cómo operan los menús de la interfaz gráfica del usuario.

**Pull-down y cascading menus** En una interfaz gráfica del usuario, los menús son usualmente implementados con menús desplegables (pull-down) y de efecto de cascada (cascading menus) de una *barra de menús* (*menú bar*) como se muestra en la figura 15.3a). Cada

**operado por menú** Estrategia de diálogo en la que el usuario debe seleccionar una acción de un menú de opciones.





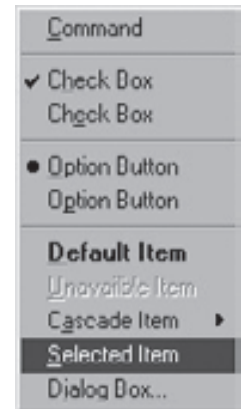
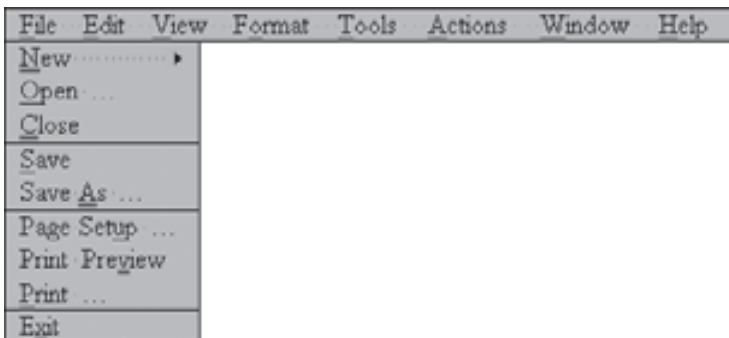
**FIGURA 15.1**  
Un diálogo clásico de un menú jerárquico

opción del menú es de hecho un grupo de acciones y órdenes relacionadas. En el margen se muestra una plantilla del menú. Muchos de estos grupos de menús son comunes para muchas o todas las aplicaciones. Por ejemplo, las aplicaciones basadas en *Windows* incluyen típicamente los siguientes grupos de menús:



Los usuarios pueden seleccionar un grupo de menús usando ya sea el ratón o un atajo del teclado (por ejemplo, oprimiendo simultáneamente la tecla Alt y la letra subrayada, lo que se llama una *tecla mnemónica*, una *tecla de atajo* o una *tecla de acceso rápido*).

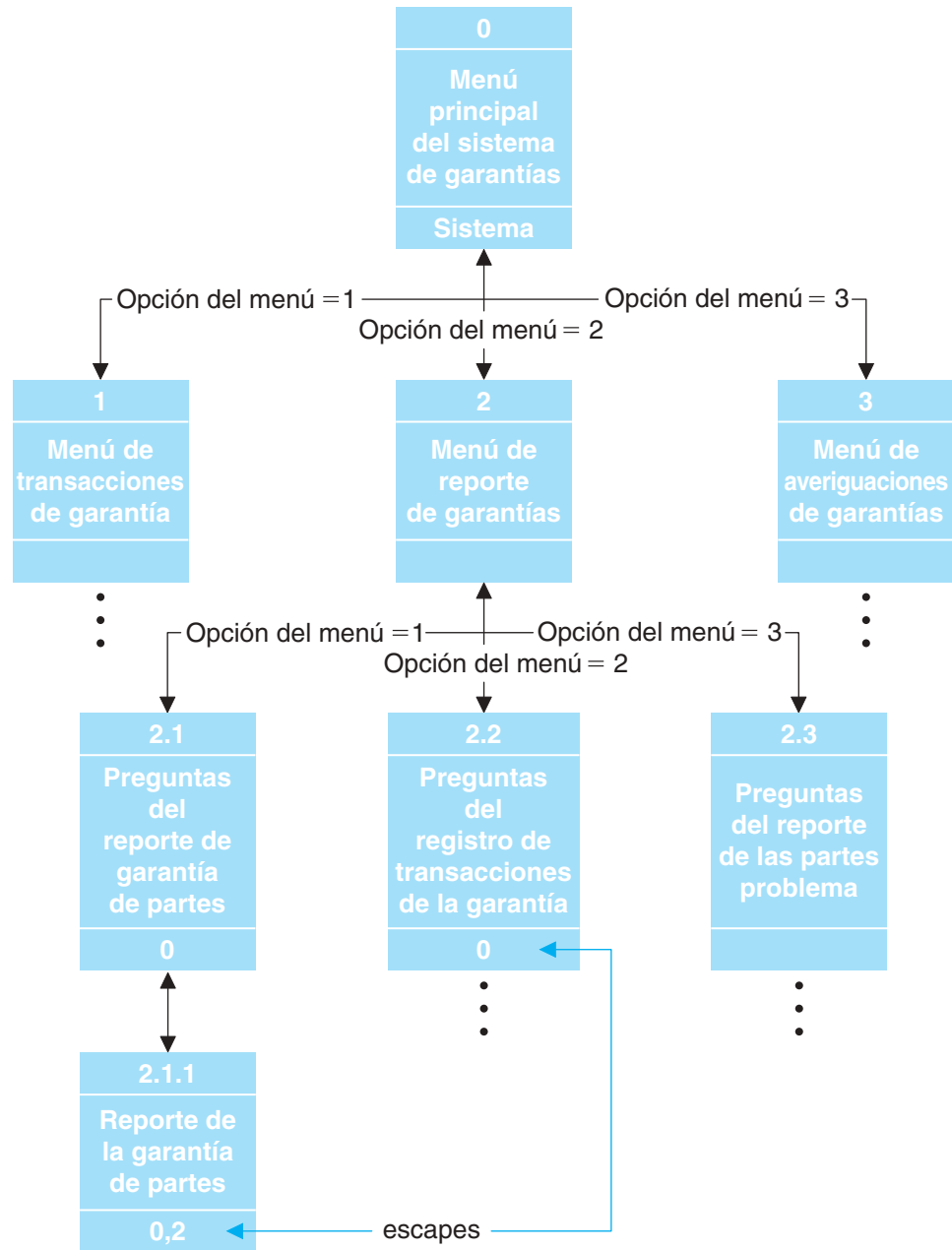
Cada grupo del menú tiene su propio *pull-down menu*. Cuando el usuario selecciona un grupo del menú bar, se despliega un submenú:



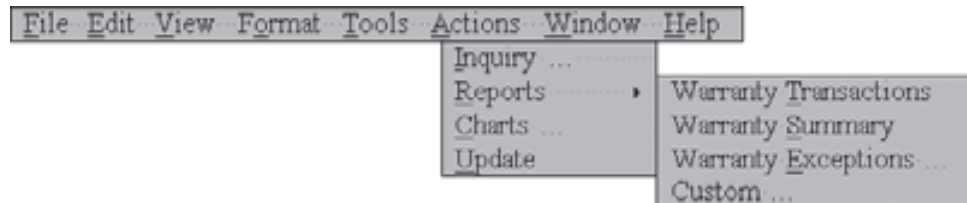
Plantilla del menú

**FIGURA 15.2**

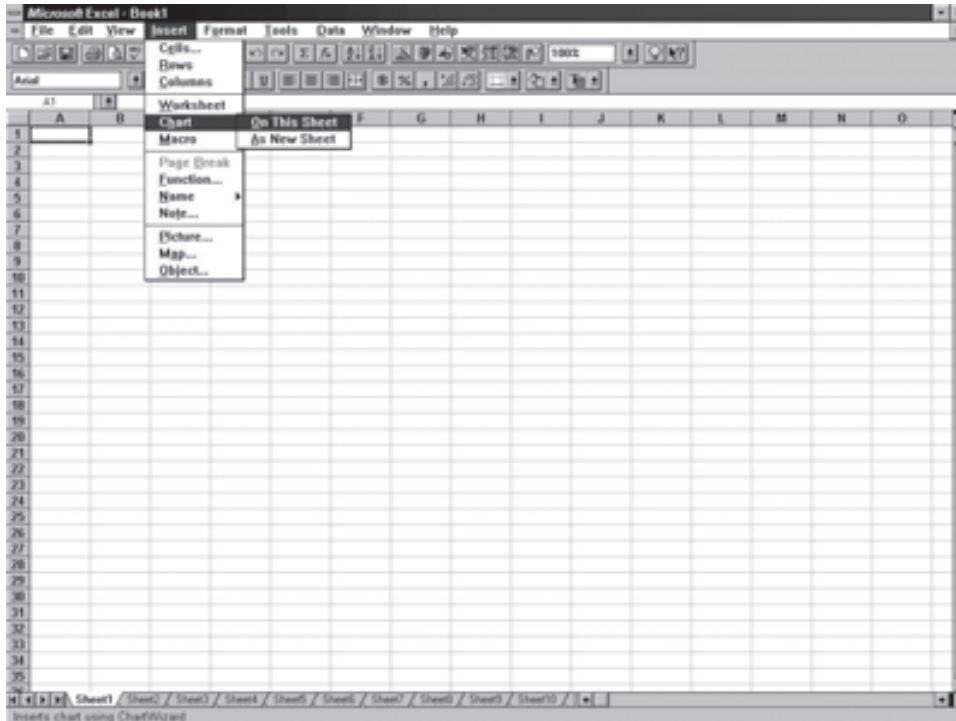
Gráfica con un diálogo de muestra



Observe que las selecciones del submenú pueden ser subagrupadas por líneas horizontales (por ejemplo, agrupando todos los comandos del submenú SAVE O PRINT). En algunos casos, una acción nombrada del submenú es seguida por elipses (tres puntos) indicando que aparecerá posteriormente una *caja de diálogo* (ventana) [vea la figura 15.3b)] para presentar opciones o recolectar instrucciones adicionales. En otros casos, una acción nombrada del submenú tendrá una flecha pequeña que indica aún otro submenú. A esto se le llama un *menú en cascada*.

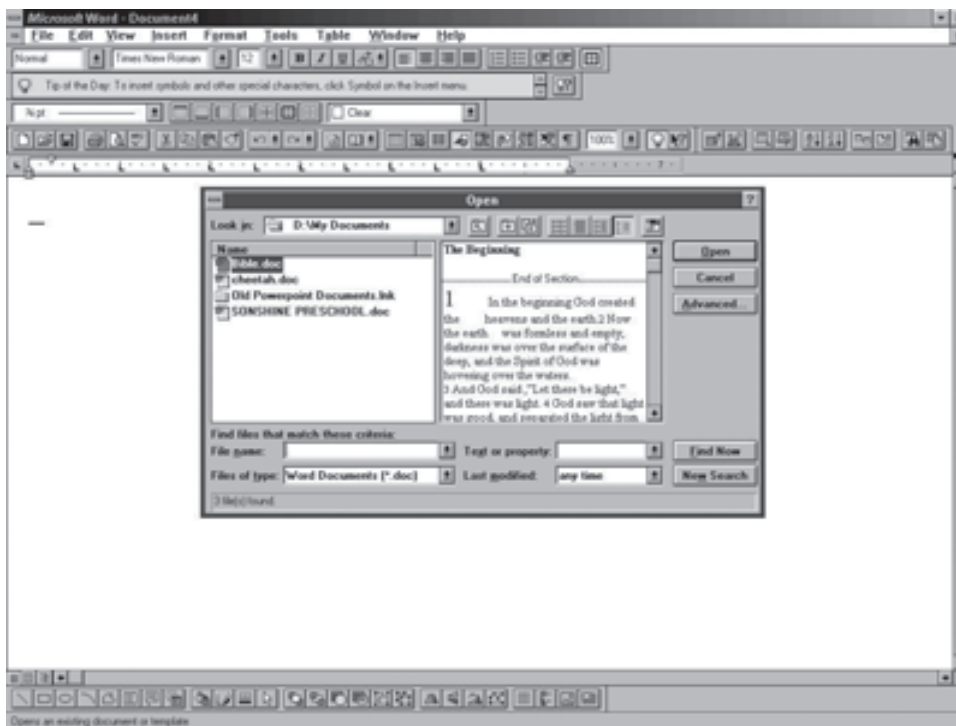


a)

**FIGURA 15.3**

a) Pull-Down y Cascading Menus (Menús desplegables y en cascada)  
 b) Dialog Box (Cuadro de diálogo)

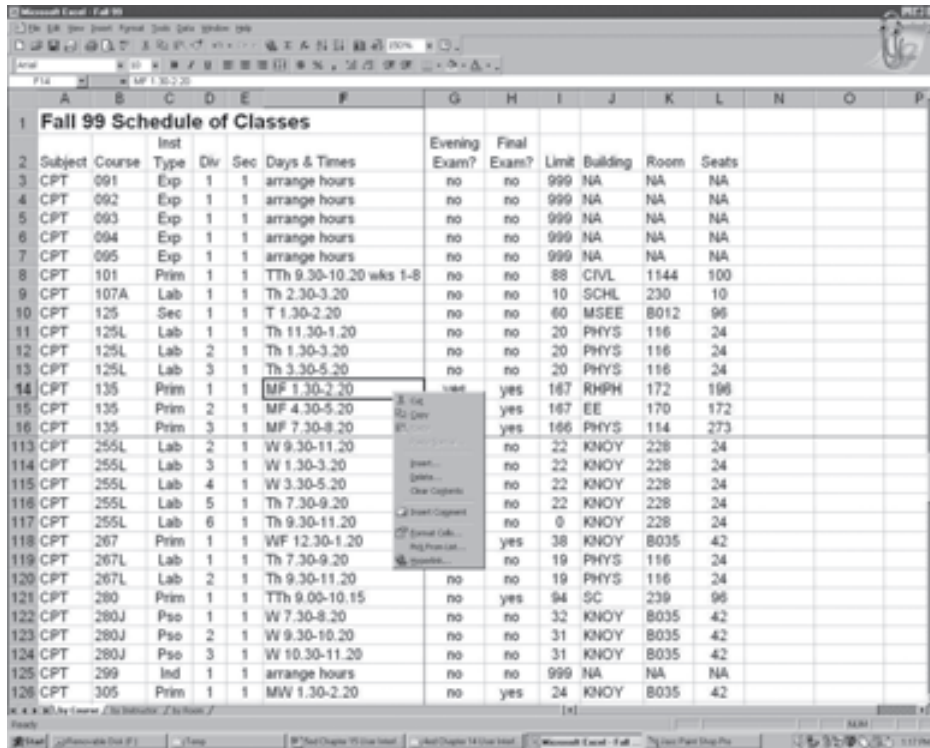
b)



**Tear-off y pop-up menus** No todos los menús se obtienen de la barra de menús. Algunas GUI permiten los *tear-off menu*. Con un menú de este tipo, el usuario puede seleccionar un drop-down menu o un cascading, “arrastrarlo” fuera de la barra de menús, y posicionarlo en otro sitio de la pantalla. Esto es especialmente útil si el menú debe usarse continuamente. Sólo una copia del menú original realmente se despliega.

FIGURA 15.4

Pop-up menus



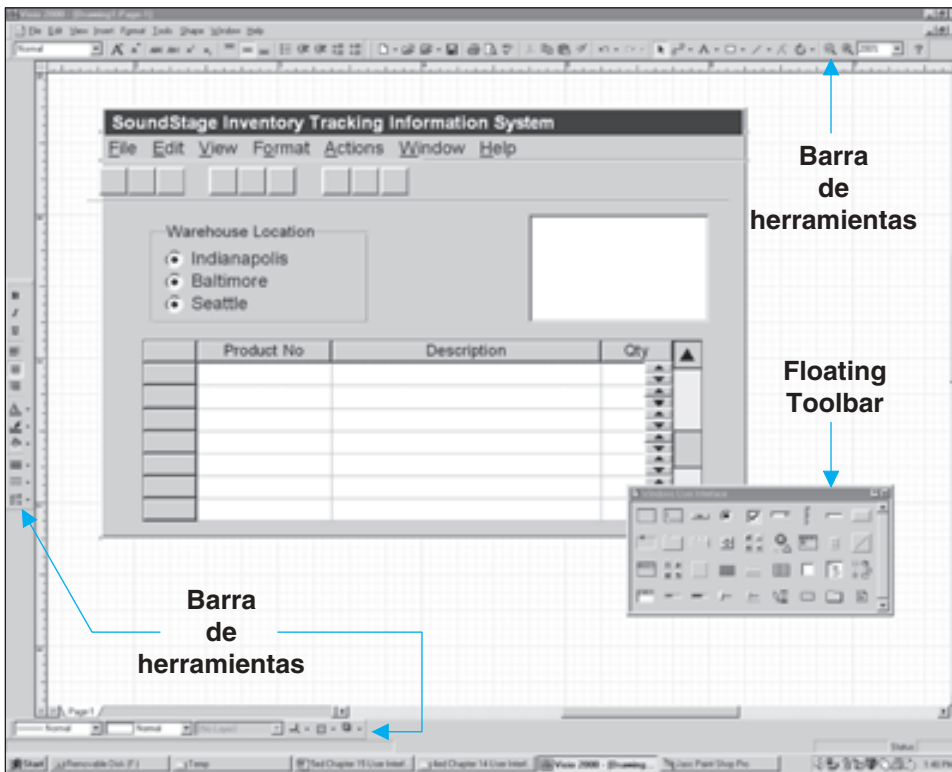
Un *pop-up menu* es sensible al contexto y dependiente de un dispositivo de indicación. Activado por el usuario al hacer clic con el botón derecho del ratón, un pop-up menú emerge de la nada (vea la figura 15.4). El pop-up menú depende de la posición en la que se encuentre el cursor en la pantalla. El cursor puede señalar un área en blanco, un campo, una celda, una palabra, o un objeto. El clic del botón derecho llamará un menú que despliegue sólo esas acciones que aplican para lo que fuere que esté en esa posición del cursor; de ahí el término *sensible al contexto*. Los pop-up menus también pueden ser en cascada. El uso de los pop-up menus es primordialmente para usuarios expertos porque no existe una pista visual de su presencia.

**Toolbar e iconic menus** Las *barras de herramientas (toolbars)* constan de *iconos* (imágenes) que representan atajos del menú para acciones y comandos que normalmente están inmersos en los drop-down y cascading menus (vea la figura 15.5). En aplicaciones de *Windows*, una toolbar de acciones comúnmente usadas se encuentra inmediatamente debajo del menú bar. El usuario puede dar un clic sobre cualquiera de estas herramientas o estos íconos para inmediatamente invocar esa acción sin pasar a través de los menús. Las toolbars pueden crearse para cualquier aplicación. Los desarrolladores de aplicaciones pueden proveer a los usuarios con alguna flexibilidad para hacer a la medida esas barras de herramientas.

Mientras la posición predeterminada para la mayoría de las toolbars está inmediatamente debajo del menú bar, muchas aplicaciones permiten que las toolbars sean reacomodadas a la izquierda, derecha, o en la parte inferior de la ventana a conveniencia del usuario. Esto se llama *atracamiento* de la toolbar. También, algunas toolbars pueden hacerse *flotar* (o moverse) dentro de cualquier posición conveniente dentro de la ventana.

**NOTA:** En aplicaciones basadas en la Web, la toolbar es provista por el navegador y no puede estar hecha a la medida para las aplicaciones específicas. Los iconos más importantes en la toolbar del navegador son PAGE FORWARD (PÁGINA ADELANTE), PAGE BACKWARD (PÁGINA ATRÁS), y los iconos de la HOME PAGE (PÁGINA DE INICIO) que son estándar para toda navegación en Internet y en la intranet basada en la Web.

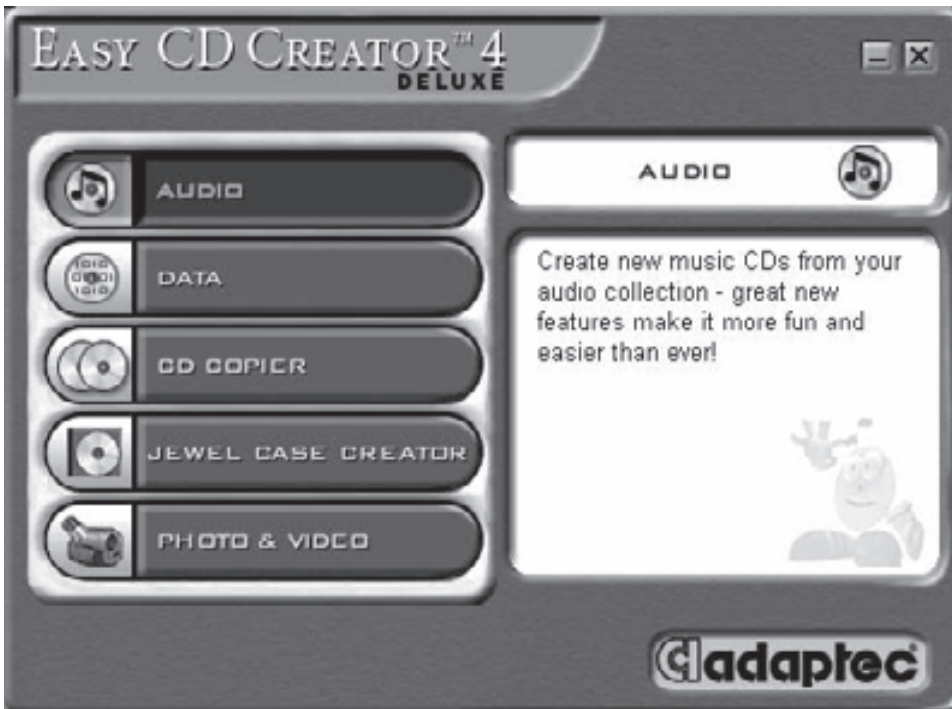
Los *menús de iconos (iconic menus)* usan imágenes para representar opciones del menú en el cuerpo principal de la ventana. En aplicaciones de *Windows*, estos menús de iconos se usan frecuentemente para proveer un centro de control (de funciones y actividades principales) para una aplicación de computadora o para documentar los pasos de ne-



**FIGURA 15.5**  
 Toolbars (Barras de herramientas)

gocios al usar una aplicación de computadora. La figura 15.6 demuestra un iconic menu. Cada botón representa una elección intuitiva del menú.

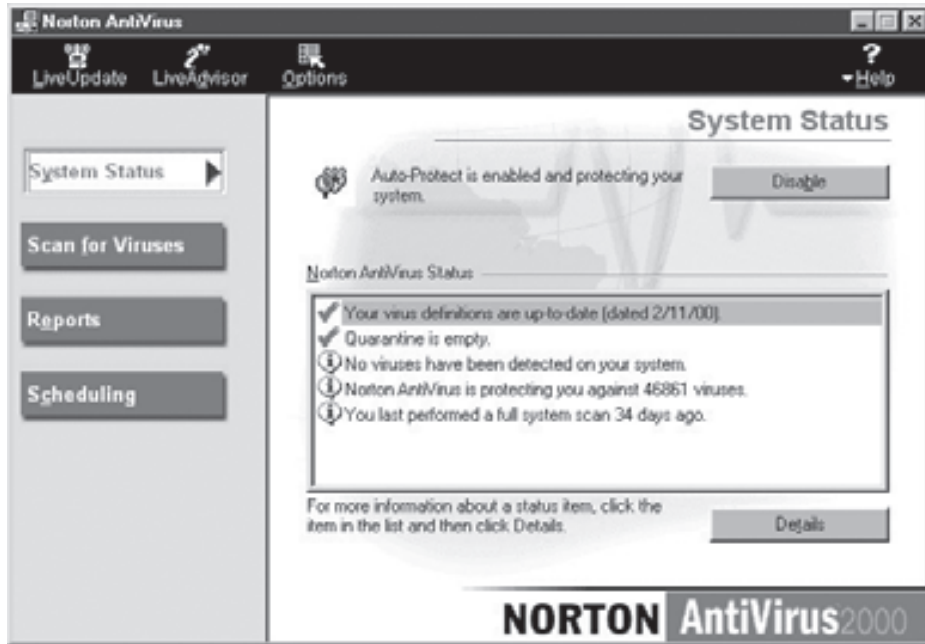
Los iconic menus son muy populares en aplicaciones basadas en la Web porque estas aplicaciones corren en el navegador; los navegadores no permiten al desarrollador alterar los comandos del menú en la barra de menú del navegador. En lugar de eso, las aplicaciones de Web frecuentemente usan imágenes, íconos y botones activados por un clic para representar las opciones del menú.



**FIGURA 15.6**  
 Iconic menu (menú de íconos)

**FIGURA 15.7**

Interface al estilo del Consumidor



La popularidad de las interfaces análogas a la Web está influyendo significativamente en el diseño de la interfaz de usuario de *Windows*. La mayoría de los sistemas de información cliente/servidor han implementado la interfaz de usuario del cliente para emular las herramientas de la PC más comúnmente usadas por el usuario tales como el procesador de palabra y la hoja de cálculo. El usuario está familiarizado con esas herramientas; por consiguiente, tiene sentido diseñar otras aplicaciones para imitar esos menús, barras de herramientas, cajas de diálogo y cosas por el estilo. Pero la popularidad de las aplicaciones basadas en la Web ha dado lugar a una interfaz nueva al estilo del consumidor.

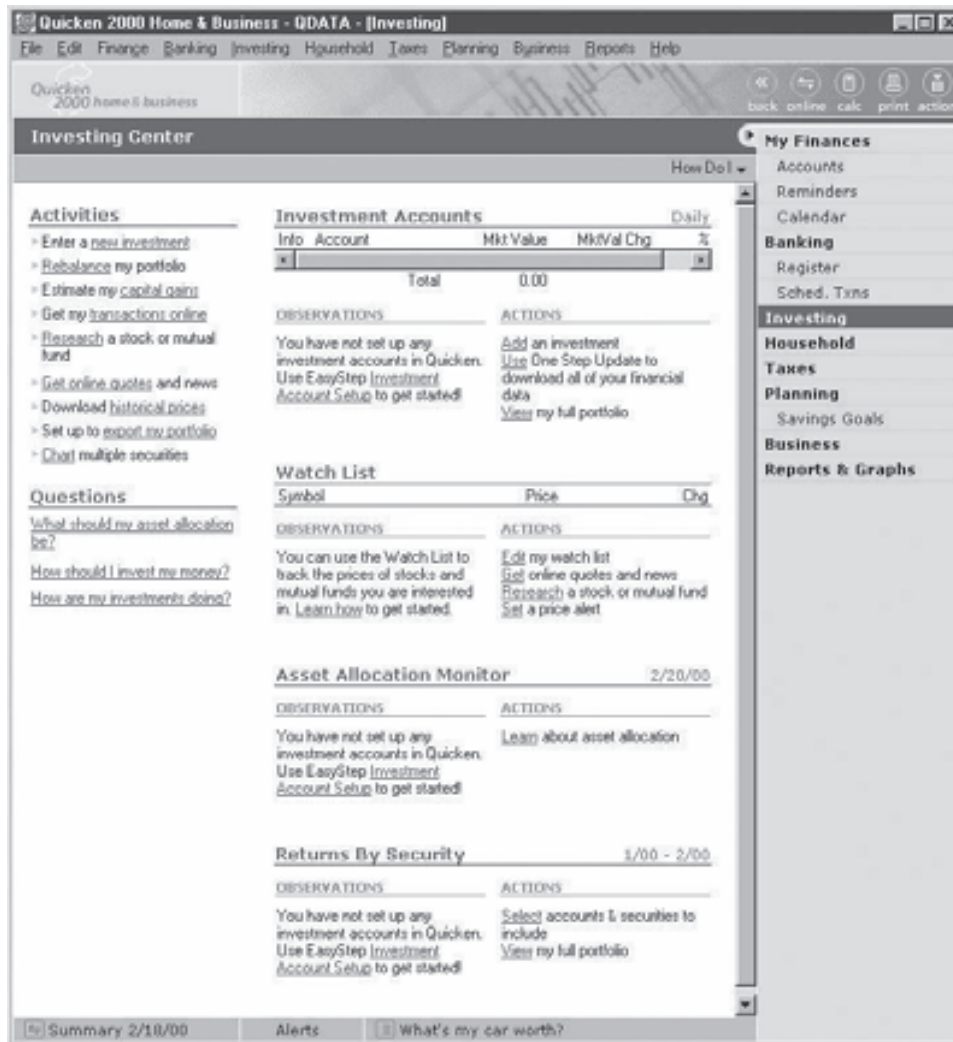
Al igual que las páginas de Web, las *interfaces al estilo del consumidor* para aplicaciones de *Windows* son algo más artísticas. Mientras las menu bars todavía pueden ser usadas, la apariencia y percepción primaria de la ventana son más parecidas al Web; así, es más amigable al consumidor. La interfaz consta de botones e íconos activados por un clic que reemplazan a los enfoques más tradicionales del menú de *Windows*. Cuando no es excesivamente complicado, esto puede ser una “cara” más amigable para aplicaciones de *Windows* que lo que se ve tradicionalmente en aplicaciones como *Office Microsoft* y *Lotus SmartSuite*. Una interfaz de *Windows* al estilo del consumidor se ilustra en la figura 15.7.

Observe la ausencia del menu bar tradicional. También observe que los botones (en el marco izquierdo) no se conforman al tamaño y el estilo tradicionales de *Windows*. La imagen de fondo es más artística, así como el uso de las fuentes y del color. (Muchas organizaciones incluyen a un diseñador gráfico como parte del equipo para desarrollar interfaces al estilo del consumidor.) Esperamos que tales estilos amigables para el consumidor sean acogidos por sistemas de información futuros basados en *Windows*.

**Hypertext e hyperlink menus** El *hipertexto* (*hypertext*) y los *hipervínculos* (*hyperlink*) son productos de interfaces de usuario contemporáneas basadas en la Web. El hipertexto y los hipervínculos fueron creados originalmente para navegar dentro y entre páginas y sitios de Web. Una palabra, un término, o una frase es señalada como un hipervínculo (usualmente formateado como texto subrayado, en general con color). El dar un clic sobre el hipervínculo hace navegar al usuario a la página asociada (o la señal de lectura en una página).

Esta tecnología puede ampliarse y adaptarse fácilmente a la implementación de menús en aplicaciones de la Internet y de la intranet basadas en la Web. Ya que estas aplicaciones corren en el navegador, y ya que la menu bar del navegador y las órdenes son fijas, no podemos implementar fácilmente menús hechos a la medida como lo hacemos en las aplicaciones de *Windows*. En lugar de eso, usamos hipertexto (*hypertext*) e hipervínculos (*hyperlinks*) para implementar esos menús en el cuerpo de la página Web. Cada opción del menú es una frase de hipertexto (o un botón o icono hipervinculado) que invoca acciones o formas en otras páginas Web. Esencialmente, este enfoque crea estructuras de



**FIGURA 15.8**

Interfaz híbrida  
Windows/Web

menú jerárquicas parecidas a las que fueron introducidas anteriormente en la figura 15.1. ¡Es algo irónico que el diseño del menú para aplicaciones basadas en la Web esté siendo conducido por un enfoque que regresa a un estilo que fue extensamente usado en aplicaciones heredadas de mainframe!

El hipertexto y los hipervínculos ya no son exclusivos de las aplicaciones de Internet y de la intranet. Muchas aplicaciones contemporáneas de *Windows* se han acercado a la popularidad de la Web presentando una interfaz de usuario *híbrida de Windows/Web*. Por ejemplo, la figura 15.8 demuestra la interfaz del usuario para el Intuit's *Quicken*, el popular programa personal de finanzas. Con sus muchos hipervínculos, se parece a una página de Web. ¡Aun cuando incluye muchas características opcionales habilitadas por la Web, es de hecho una aplicación de *Windows*! La primera pista es que trabaja en su propia ventana, no en la del navegador. También tiene su propia barra de *Windows*, completa con todos los pull-down y en cascading menus que son comunes en las aplicaciones de *Windows*. Esperamos que esta interfaz híbrida se popularice más a medida que los negocios consideren a Internet y a las intranets como el cimiento fundamental para todos los sistemas de información.

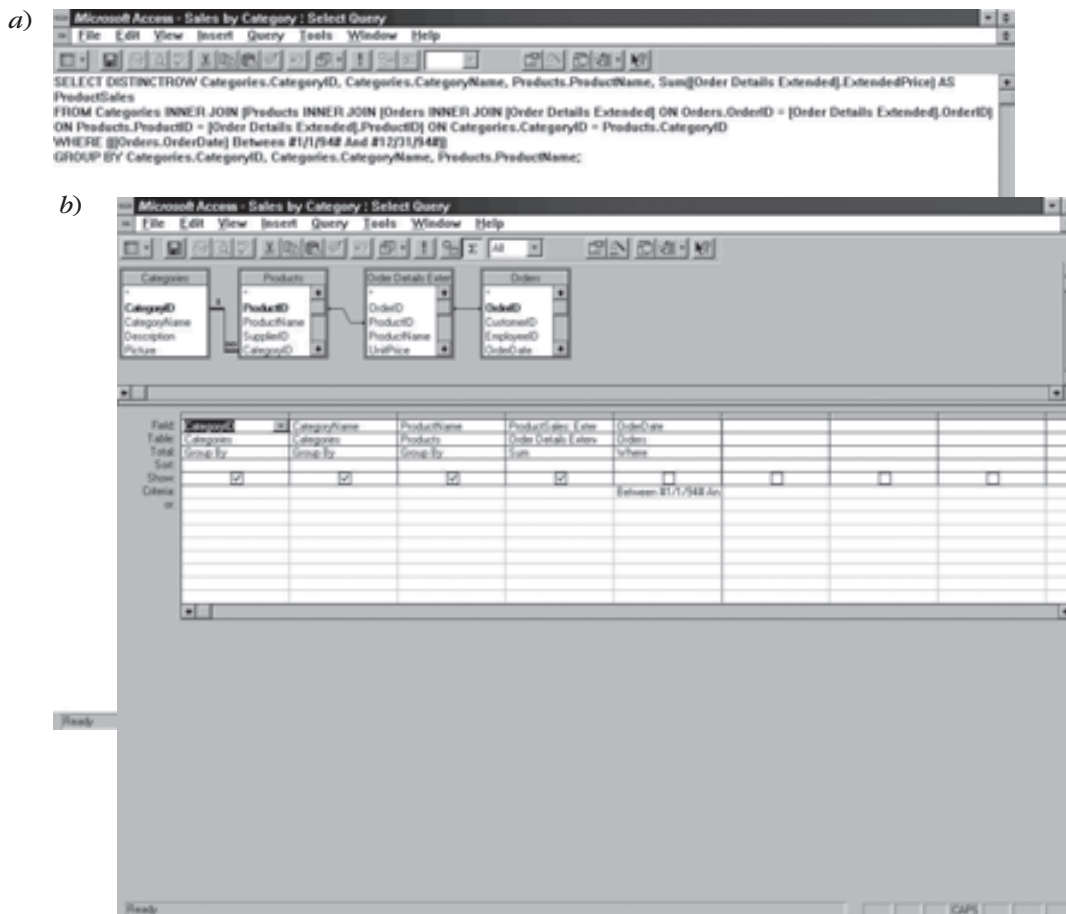
## > Interfaces de comandos

En lugar de los menús, o además de ellos, algunas aplicaciones se diseñan usando un diálogo basado en un *conjunto de instrucciones* (también llamado una *interfaz de lenguaje de comandos*). Ya que el usuario debe aprender la sintaxis del conjunto de instrucciones, esta aproximación es la más adecuada para los usuarios expertos. Tres tipos de sintaxis

pueden definirse. La determinación de cuál tipo deberá usarse depende de la tecnología disponible:

- Una *sintaxis basada en el lenguaje* se construye alrededor de un lenguaje de comandos ampliamente aceptado que puede ser usado por el usuario para invocar acciones. Los ejemplos incluyen *Query by Example (QBE)* y *Structured Query Language (SQL)*, los cuales son lenguajes de base de datos que pueden ser usados por el usuario final para acceder a los datos y crear reportes a la medida.
- Una *sintaxis mnemónica* se construye alrededor de las órdenes definidas para aplicaciones de sistemas de información a la medida. Los usuarios son provistos de una terminal en la cual pueden introducir órdenes que solicitarían acciones y respuestas del usuario de la computadora. Idealmente, las órdenes deberían ser significativas para el usuario (incluyendo todas las abreviaturas permitidas).
- *La sintaxis del lenguaje natural* permite a los usuarios introducir las preguntas y las órdenes en su lengua materna. El sistema interpreta estas órdenes contra una sintaxis conocida y pide aclaración si no entiende lo que el usuario desea.

Los estilos operados por comandos eran comunes en las aplicaciones heredadas de mainframe y en las primeras aplicaciones basadas en DOS de la PC. Pero este estilo de interacción todavía puede encontrarse en las aplicaciones gráficas de hoy en día. Por ejemplo, el producto de la base de datos de *Access Microsoft* contiene una opción de query (consulta) que le permite al desarrollador (apuntar y dar clic) desarrollar visualmente un query (consulta) (vea la figura 15.9). El desarrollador simplemente selecciona las tablas, columnas, y filas de la base de datos a incluir en una consulta, como se muestra



**FIGURA 15.9** Interfaz accionada por comandos

en la figura 15.9a). Luego, si se desea, el desarrollador puede ver y editar el código SQL a nivel de comandos que se genera con dicho query, como se muestra en figura 15.9b). Nuevamente, el enfoque del conjunto de instrucciones requiere un grado de experiencia, experiencia, y conocimiento del usuario.

### > Diálogos pregunta-respuesta

Un estilo de *diálogo pregunta-respuesta* se usa primordialmente para sustituir ya sea diálogos accionados por menú o por instrucciones. Los usuarios son cuestionados con preguntas para las cuales suministran respuestas. Las preguntas más simples involucran respuestas de sí o no; por ejemplo:

¿QUIERE USTED VER TODAS LAS PARTES? [NO].

¡Observe cómo se le ofreció al usuario una respuesta predeterminada! Las preguntas pueden estar más elaboradas. Por ejemplo, el sistema podría preguntar:

¿QUÉ NÚMERO DE PARTE LE INTERESA?

Esta estrategia requiere que usted considere todas las respuestas correctas posibles y se ocupa de las acciones a tomar si se ingresan respuestas incorrectas. ¡El diálogo pregunta-respuesta es difícil porque usted debe tratar de considerar todo lo que el usuario del sistema pudiese hacer mal!

Los diálogos pregunta-respuesta son muy populares en aplicaciones basadas en la Web. Por ejemplo, un sistema de reservación de coches puede hacer una serie de preguntas que definan la clase de coche y acuerdo de alquiler que usted requiere:

¿DÓNDE QUIERE USTED RECOGER SU VEHÍCULO DE ALQUILER?  
 ¿DÓNDE PIENSA USTED DEVOLVER SU VEHÍCULO DE ALQUILER?  
 ¿CUÁLES SON LA FECHA Y LA HORA PARA RECOGERLO?  
 ¿CUÁLES SON LA FECHA Y HORA DE DEVOLUCIÓN?  
 ¿QUÉ TIPO Y TAMAÑO DE VEHÍCULO NECESITA USTED?  
 ¿TIENE USTED ALGUNOS CUPONES PROMOCIONALES? . . .

Una drop-down list de respuestas alternativas puede acompañar a cada pregunta. Conjuntamente, estas preguntas y sus respuestas definen una transacción comercial.

### > Consideraciones especiales para el diseño de la interfaz del usuario

Además de establecer un estilo de la interfaz de usuario, los analistas deben abordar ciertas consideraciones especiales para el diseño de la interfaz de usuario. ¿Cómo serán reconocidos y autenticados los usuarios al usar el sistema? ¿Hay cualesquiera consideraciones de seguridad o de privacidad que deban acomodarse en la interfaz del usuario? Finalmente, ¿cómo obtendrán los usuarios ayuda vía la interfaz de usuario?

**Controles internos-autorización y autenticación** En la mayoría de ambientes, los usuarios del sistema deben ser autenticados y autorizados por el sistema antes de que se les permita realizar ciertas acciones. En otras palabras, los usuarios del sistema deben “introducirse” al mismo. La mayoría de los ingresos requieren una IDENTIFICACIÓN DEL USUARIO y una CONTRASEÑA. Los usuarios de sistema no deberán estar obligados a aprender y memorizar IDENTIFICACIONES DEL USUARIO y CONTRASEÑAS múltiples. Idealmente, deberían estar obligados a usar el mismo ingreso que el que se usa para su cuenta local de la red de área. (*Windows XP, NT, y 2000* permiten que esta autenticación ocurra sin que sea necesario volver a teclear en ninguno de los dos campos.)

La figura 15.10a) demuestra la interfaz del usuario para ingresar en el SoundStage. La IDENTIFICACIÓN DEL USUARIO y la CONTRASEÑA serán autenticadas contra el archivo de cuentas de la red. Observe que la contraseña se imprime como asteriscos cuando el usuario la teclea, una medida común de seguridad y de privacidad. Si la identificación del usuario o contraseña no son autenticadas, se exhibirá el diálogo de autorización de seguridad presentado en la figura 15.10b).

La autenticación es sólo la mitad de la solución. Una vez autenticado, el acceso del usuario y los privilegios de servicio para este sistema de información deben ser establecidos. Hay muchos modelos para establecer y manejar privilegios. Un lineamiento importante es asignar los privilegios a *los roles*, no a los individuos. En la mayoría de los negocios, las personas cambian de trabajo rutinariamente; son reasignadas y promovidas a responsabilidades de

**FIGURA 15.10**

- a) Pantalla de acceso con autenticación.  
b) Pantalla con error de autenticación



trabajo y roles nuevos. También, las descripciones del trabajo y los roles cambian de vez en cuando. Finalmente, las personas dejan el negocio y algunas son liquidadas. Por todas estas razones, los privilegios deberían ser asignados a los roles. Entonces es sencillo identificar los roles que cualquier IDENTIFICACIÓN DEL USUARIO puede asumir.

Para cada rol, los privilegios específicos que deben asignarse al mismo no necesitan estar definidos. Los privilegios pueden incluir permiso para leer tablas o vistas específicas; el permiso para crear, actualizar, o suprimir registros (filas) en tablas o vistas específicas; el permiso para generar y ver reportes específicos; el permiso para ejecutar transacciones específicas; Y cosas por el estilo. Aunque técnicamente no es parte de la interfaz, la definición de estos roles y permisos es necesaria tanto para el diseño de una interfaz apropiada de ingreso como para especificar funcionalmente el modelo de seguridad completa de autenticación y autorización para el sistema.

En realidad podrían aplicarse vistas diferentes del usuario para hacer a la medida la interfaz de usuario para las diferentes categorías de usuarios. Por ejemplo, es bastante fácil “empalidecer” (cambiar la fuente de negro a gris) y desactivar aquellas opciones del menú y las cajas de diálogo que deben estar restringidas para ciertas clases de usuarios de sistema.

Con el surgimiento del comercio electrónico, los consumidores y otros negocios deben tener confianza que somos quienes nosotros afirmamos ser. Los consumidores pueden proveer números de la tarjeta de crédito y otra información privada para su transmisión por Internet. Por esta razón, SoundStage compró una certificación de Web para autenticarse a sí mismo para sus miembros del club y sus miembros prospecto. En cualquier momento, usando la interfaz del navegador, los miembros SoundStage pueden ver el certificado de autenticación en la figura 15.11. Con esta certificación, el sitio de Web SoundStage exhibirá un icono de “certificación de servidor seguro” (vea al margen; el *candado*) que le diga a los consumidores que sus datos estarán encriptados (cambiados con seguridad) para asegurar que su tarjeta de crédito y los datos personales no están siendo interceptados o accedidos por otros cuando son transmitidos a través de la red.

**Ayuda en línea** ¡Ya nadie quiere leer los manuales! Al menos eso es tal como parece. Y hasta cierto punto, es justificado. Los manuales son esencialmente archivos secuenciales de información. Las personas quieren acceso directo inmediato a una ayuda sensible al

Indicador de seguridad del navegador de Internet



FIGURA 15.11

Certificado de seguridad del servidor

contexto, es decir, la ayuda que sea lo suficientemente inteligente para descubrir lo que podrían estar tratando de hacer. Hay definitivamente una tendencia hacia la construcción de sistemas de ayuda y tutoriales directamente en la aplicación. La ayuda en línea pasa a formar parte de la interfaz del usuario.

La ayuda de propósito general para una aplicación está incorporada en el menú **H**elp para aplicaciones de *Windows*. Para aplicaciones de Web, la ayuda generalmente se incorpora como páginas separadas, generalmente páginas “de aparición automática”(pop-up pages) en ventanas separadas, a fin de que el usuario también pueda permanecer enfocado en la página que inició la necesidad de ayuda.

Hoy en día, el HTML (**H**ypertext **M**arkup **L**anguage) se está volviendo gradualmente el lenguaje universal para la construcción de sistemas de ayuda para interfaces gráficas del usuario; tanto aplicaciones de Web como de *Windows*. Por ejemplo, el sistema completo de ayuda para Microsoft *Office* está escrito en HTML.

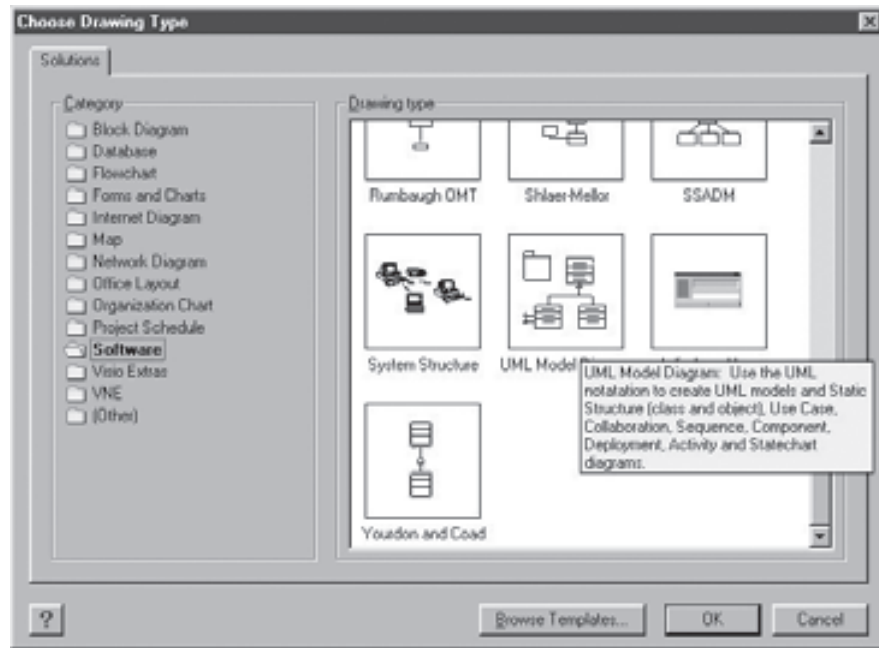
El diseño, construcción, y prueba de un sistema de ayuda se simplifican por las herramientas automatizadas de la actualidad. Un *sistema de ayuda* completo incluye una tabla de contenidos, numerosas instrucciones, ejemplos, y un índice exhaustivo. Muchos *paquetes de ayuda para la autoría*, como *el RoboHelp* de Macromedia, amplifican al procesador de texto del autor de ayuda para ayudar con la planificación, el esquema, la escritura, el indexado, y aspectos de vinculación de hipertexto de la autoría de un sistema completo de ayuda.

Un sistema de ayuda bien diseñado implementará una gran variedad de elementos de ayuda. Quizá los tipos más comúnmente encontrados de ayuda son esos que los usuarios deben iniciar. Como se mencionó anteriormente, la tecla de función F1 es casi universalmente aceptada como una orden de petición de ayuda. Asimismo, una opción estándar de la barra de menús de **H**elp se usa comúnmente para organizar y presentar diferentes niveles y tipos de ayuda en la mayoría de aplicaciones basadas en *Windows* (comercial o hecho a la medida). Finalmente, como se ilustra en la figura 15.12, las interfaces de *Windows* y basadas en la Web frecuentemente usan controles de *sugerencias de herramientas* para proveer ayuda de aparición automática (pop up help) asociada con herramientas específicas y objetos de iconos. Las sugerencias de herramientas aparecen cuando el usuario posiciona momentáneamente el cursor sobre el icono (o el objeto) en la pantalla. Los

FIGURA 15.12

Sugerencias de herramientas de ayuda

Agente de ayuda



#### ¿Qué desea hacer?

- 🔦 Cambiar el tipo de diagrama por omisión
- 🔦 Diagramas de diagnóstico
- 🔦 Cambiar el despliegue de las etiquetas, las tablas de datos, leyendas, líneas de cuadrícula, o ejes del diagrama
- 🔦 Cambiar la vista del diagrama 3-D
- 🔦 Cambiar los rótulos, los títulos y otro texto del diagrama
- 🔦 Cambiar el modo de graficar los datos

How do I customize the X-Axis on my chart?

Options

Search

Procesamiento de lenguaje natural

**agente** Objeto de software reutilizable que puede funcionar en diferentes aplicaciones y redes.

consejos de la herramienta son apropiados para todos los iconos porque el diseñador de la interfaz de usuario nunca puede tener la certeza de que la imagen o la etiqueta que aparece en un icono va a ser significativa para el usuario del sistema.

Dos características de ayuda adicionales y comunes particularmente efectivas para el usuario más novato son los help wizard (magos/experto de ayuda) y los assistants (agentes o asistentes). Como se ilustra en la figura 15.13a, un *help wizard* guía a los usuarios a través de procesos complejos presentando una secuencia de dialogue boxes (cajas de diálogo) que requieren las entradas de datos del usuario y la retroalimentación del sistema. Llamamos su atención a lo siguiente:

- 1 Como es típico con los help wizards, el diálogo usualmente incluye una serie de instrucciones o preguntas que el usuario debe responder.
- 2 El help wizard contiene explicaciones para auxiliar en la comprensión y la toma de decisiones del usuario.
- 3 El help wizard también provee un botón para pedir ayuda más detallada al completar la tarea.
- 4 El botón “siguiente” sugiere pasos adicionales o posteriores que deben ser soportados por el help wizard. (El botón “siguiente” generalmente cambia a “terminar” una vez que una secuencia de cajas de diálogo se completa.) La figura 15.13b) muestra la pantalla resultante y el paso siguiente soportado por el help wizard.

¡Microsoft y los vendedores de control del software de terceros venden a los help wizards para ayudar a los desarrolladores a construir help wizards!

Los agentes son otra tecnología con aplicaciones para los sistemas de ayuda. Los **agentes** son objetos de software reusables que pueden operar a través de aplicaciones diferentes de software y aun a través de redes. El *agente de ayuda* de Microsoft (llamado *asistente*) provee un asistente común de ayuda en las aplicaciones de *Office*. En su forma predeterminada, se presenta a sí mismo como un clip animado (vea margen). (El agente de ayuda de Microsoft puede programarse en aplicaciones hechas a la medida, tanto para *Windows* como para el navegador *Internet Explorer* de la Web.) Un solo clic del usuario en este agente de ayuda inicia la ayuda.

El agente de ayuda Microsoft es complementado por la tecnología de procesamiento del lenguaje natural (vea el margen) que le permite al usuario escribir una consulta en frases del lenguaje natural que son interpretadas por el agente para presentar las respuestas de ayuda más probables. Entonces el usuario puede seleccionar una de esas respuestas o entrar en el índice de ayuda más detallado.

a)

**Create New Account**

1 Choose the type of account to create.

**Banking & Cash**

- Checking
- Savings
- Credit Card
- Money Market
- Cash

**Investments**

- Brokerage
- IRA or Keogh
- 401(k)
- Dividend Reinvestment Plan
- Other Investment

**Home & Business**

- House (with or w/o Mortgage)
- Vehicle (with or w/o Loan)
- Asset
- Liability
- Invoices / Receivables
- Bills / Payables

2 \* If you are trying to set up a security such as a stock or bond, you should probably choose Brokerage. Click help for more information.

3 Cancel 4 Help 4 Next

**FIGURA 15.13**

Experto/mago de ayuda (Wizard)

b)

**Credit Card Account Setup**

EasyStep Summary

Account Information

Account Name: Credit Card

Description: My New Credit Card

Balance: 3.00 as of 2/21/00

Financial Institution: Ashburn Bank

Online Information

Enable Online Account Access

Optional Information

Credit Limit, if Applicable: 5,000.00

Additional Information: Info... Tax Info: Tax...

Cancel Help Back Done

El tema primordial para diseñar un buen sistema de ayuda es que el diseñador deberá anticipar errores del usuario del sistema. Cuando se diseñe la interfaz del usuario para reportar tales errores, el diseñador siempre deberá proporcionar ayuda al usuario del sistema para resolver el error. Después de salir de cualquier sesión de ayuda, se deberá regresar al usuario al lugar donde se encontraba dentro de la aplicación antes de pedir o recibir la ayuda.

## Cómo diseñar y elaborar un prototipo para una interfaz del usuario

Los ambientes gráficos de hoy en día ponen énfasis en desarrollar un sistema global que se mezcle bien con el ambiente del lugar de trabajo del usuario. Las siguientes secciones demostrarán cómo diseñar una interfaz de usuario para un ambiente gráfico. Extraeremos

tablas de los ejemplos del estudio del caso de SoundStage. Examinaremos tanto las entradas cliente/servidor, basadas en *Windows* como las entradas del comercio electrónico basadas en la Web que corren en un navegador.

### > Herramientas automatizadas para el diseño de la interfaz del usuario y la elaboración de prototipos

Las herramientas automatizadas para el soporte del diseño y la elaboración de prototipos de la interfaz del usuario son las mismas que las herramientas que identificamos en los capítulos 13 y 14 para el diseño de salidas y de entradas. La herramienta automatizada más comúnmente usada para el diseño de la interfaz del usuario es el ambiente de desarrollo de aplicación de la base de datos de la PC. La mayoría de los productos de base de datos para PC tales como el *Access* de Microsoft no son lo suficientemente poderosos para desarrollar la mayoría de las aplicaciones a nivel de empresa, pero son más que adecuados para usarse en la formación de prototipos de las pantallas de la interfaz de usuario de una aplicación. Dada una estructura de base de datos (fácilmente especificada en *Access*), usted rápidamente puede generar o puede crear formularios (formatos) para ingresar datos. Usted puede incluir la mayoría de los controles de la interfaz gráfica del usuario que describimos en este capítulo. Los usuarios posteriormente pueden ejecutar esos formularios y le pueden decir qué funciona y qué no.

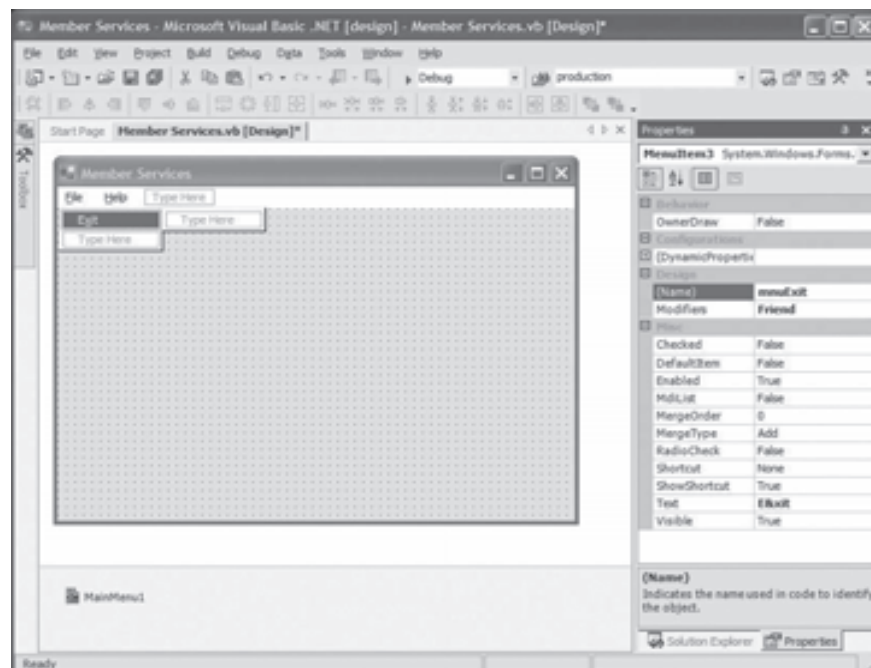
Muchas herramientas de ingeniería de software asistida por computadora también incluyen opciones para el diseño de pantallas y la creación de prototipos usando el repositorio de proyectos creado durante el análisis de requerimientos. La opción de diseño de pantallas del *System Architect* se demostró previamente en el capítulo 13, figura 13.7.

La mayoría de los ambientes de desarrollo de aplicación basados en interfaces gráficas del usuario, como *Visual Studio* de Microsoft, pueden usarse fácilmente para construir prototipos no funcionales de pantallas de la interfaz del usuario. El término clave aquí es *no funcional*. Los formatos se verán reales, pero no habrá código para implementar cualquiera de los botones o los campos. Esa es la esencia de la creación *rápida* de prototipos. Por ejemplo, la figura 15.14 muestra un diálogo *Visual Studio* para construir un menú simple.

En el capítulo 14 presentamos un número de controles de entradas que podrían incluirse en cualquier ventana. El número de controles disponibles para el diseñador de la interfaz está limitado sólo por el ambiente de desarrollo de aplicaciones que se usará para construir la interfaz. La figura 15.15 ilustra algunos controles adicionales que están disponibles en el ambiente *Visual Studio*, incluyendo barras de inspección, columnas ordenadas con títulos, controles de plantilla, cuadros de lista de directorio (directory list boxes), y listas desplegables (drop-down lists) sin entradas.

**FIGURA 15.14**

Construcción de un menú con *Visual Basic*





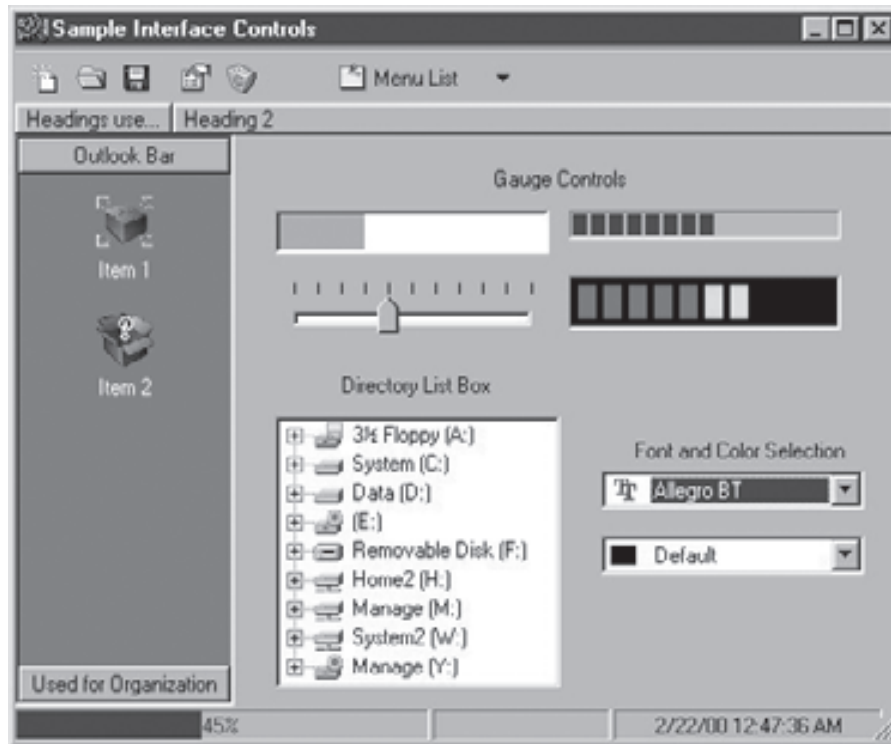


FIGURA 15.15

Controles adicionales de la interfaz del usuario

## > Proceso de diseño de la interfaz del usuario

El diseño de la interfaz del usuario no es un proceso complicado. Los pasos básicos que intervienen son:

1. Realice un mapa del diálogo (comunicación) de la interfaz de usuario.
2. Realice un prototipo del diálogo y de la interfaz del usuario.
3. Obtenga la retroalimentación del usuario.
4. Si es necesario, regrese a los pasos 1 o 2.

En realidad, los pasos no son estrictamente secuenciales en la práctica. En lugar de eso, los pasos son iterativos: por ejemplo, a medida que los prototipos se desarrollan, son revisados por los usuarios del sistema, quienes proveen una retroalimentación que puede requerir revisiones o un prototipo nuevo. En las siguientes subsecciones, examinamos estos pasos en una sola iteración e ilustramos algunos ejemplos del proyecto SoundStage.

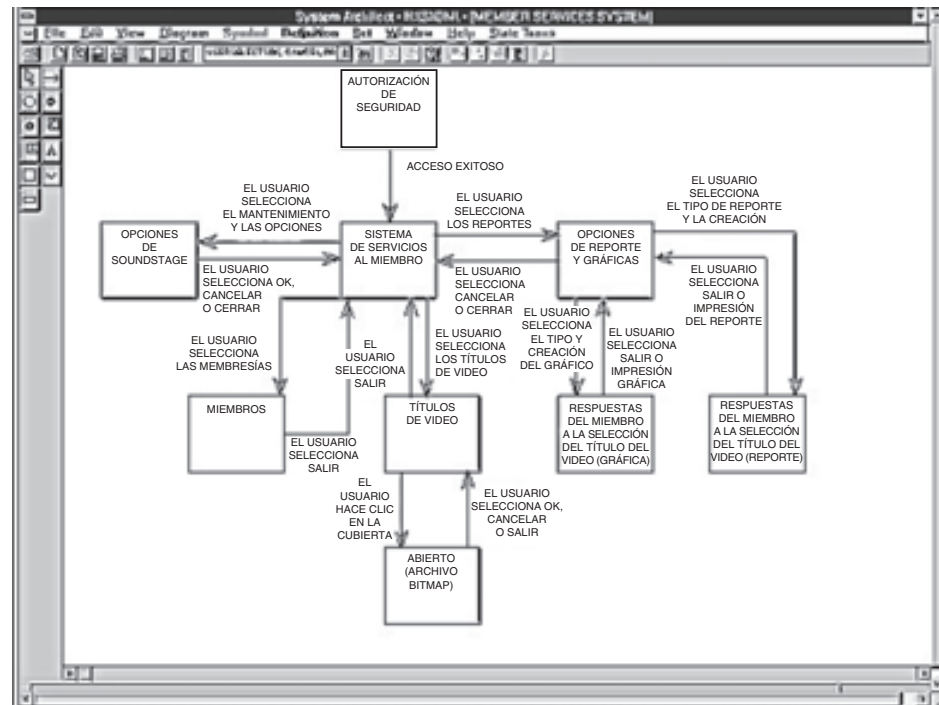
**Paso 1: Realice un mapa del diálogo** ¿Una interfaz de usuario típica puede incluir muchas pantallas posibles (que pueden constar de varias ventanas), quizá centenares! Cada pantalla puede ser diseñada y puede hacerse su prototipo. Pero, ¿qué hay acerca de la coordinación de estas pantallas?

Las pantallas se generan típicamente en un orden específico. Usted también debe ser capaz de alternarlas. Adicionalmente, algunas pantallas pueden aparecer sólo bajo ciertas condiciones. Para hacer las cosas aun más difíciles, algunas pantallas pueden desplegarse de manera repetitiva hasta que se cumpla alguna condición. Esto suena casi como un problema de programación, ¿no es cierto? Necesitamos una herramienta para coordinar las pantallas que pueden generarse en una interfaz de usuario. Un **diagrama de transición de estado (state transition diagram, STD)** se usa para esquematizar la secuencia y las variaciones de pantallas que pueden generarse cuando el usuario del sistema se sienta ante la terminal. (Los autores están usando el término *pantalla* en un sentido general. Cuando se diseñan interfaces gráficas, el término puede referirse a una pantalla completa de despliegue, una ventana, o una caja de diálogo [dialogue box].) Usted puede considerarlo como un mapa de rutas. Cada pantalla es análoga a una ciudad. No todas las carreteras pasan a través de todas las ciudades. Los rectángulos se usan para representar pantallas de despliegue. Las flechas representan el flujo de control y el evento de disparo que hace que la pantalla se active o reciba el enfoque.

**diagrama de transición de estado (STD)** Herramienta que se usa para ilustrar la secuencia y variación de pantallas que pueden ocurrir durante una sesión de usuario.

**FIGURA 15.16**

Diagrama de transición de estado parcial de SoundStage



Los rectángulos describen sólo lo que puede aparecer durante el diálogo. La dirección de las flechas indica el orden en el cual estas pantallas aparecen. Una flecha separada, cada una con su propia etiqueta, se dibuja para cada dirección porque las diferentes acciones disparan un flujo de control desde y hacia una pantalla dada.

Examinemos un diálogo que está bajo construcción para el proyecto SoundStage (vea la figura 15.16). El diagrama de transición de estado de SoundStage parcialmente terminado está siendo desarrollado usando un producto de ingeniería de software asistida por computadora, el *System Architect* de Popkin. Observe lo siguiente:

- 1 El diagrama de transición de estado parcial incluye referencias a algunas de las pantallas de entrada de SoundStage desarrolladas en el capítulo 14.
- 2 El diagrama también incluye referencias a algunas de las pantallas de salida diseñadas en el capítulo 13.
- 3 La pantalla de EL MIEMBRO QUE REPARA AL SISTEMA será una pantalla nueva que necesitará ser diseñada y deberá hacerse su prototipo. Esta pantalla servirá como la ventana principal de la aplicación. Jugará un papel principal en cuanto a proporcionar al usuario la habilidad de obtener acceso a las pantallas de entrada y salida del sistema, que se diseñaron anteriormente. También proveerá al mismo de la habilidad de completar varias funciones adicionales (más allá del procesamiento de entradas y salidas) que comúnmente se establecen durante el diseño de la interfaz del usuario. La GUI será accesible únicamente cuando los usuarios hayan proporcionado la AUTORIZACIÓN DE SEGURIDAD requerida en la pantalla y hayan accedido con éxito al sistema.
- 4 La pantalla de SOUNDSTAGE OPTIONS (OPCIONES DE SOUNDSTAGE) es otra pantalla nueva que debe crearse. Esta pantalla permitirá a los usuarios establecer diversas opciones y acciones por omisión para usarse durante su sesión; por ejemplo, seleccionar una impresora, hacer zooming (acercamientos o alejamientos), y muchas otras opciones más.

Los diagramas de transición de estado como el que se presenta en la figura 15.16, pueden ser muy grandes, especialmente cuando se añaden al diagrama todas las pantallas de entradas, salidas, ayudas, y otras más. Por consiguiente, es común dividir el diagrama en un conjunto de diagramas separados más simples y más fáciles de leer.

**Paso 2: Realice un prototipo del diálogo y de la interfaz del usuario** Recuerde que tenemos algunas pantallas nuevas para diseñar y para hacer su prototipo. Algunas de estas pantallas nuevas fueron identificadas para conjuntar la aplicación así como sus pantallas



FIGURA 15.17

Menú principal de SoundStage

de entrada y salida las cuales fueron diseñadas con anterioridad. Algunas pantallas fueron identificadas para proporcionar a los usuarios flexibilidad en cuanto a que la interfaz de la aplicación quede hecha a la medida y pueda ajustarse a sus necesidades. Asimismo, otras pudieron haber sido identificadas para que tengan que ver con los controles del sistema, tales como el respaldo y la recuperación.

Examinemos algunas pantallas nuevas que tenían que crearse para el Sistema de servicios a los miembros de SoundStage. A los usuarios del sistema primero se les presentaría la pantalla de autenticación de las entradas que fue estudiada anteriormente en el capítulo (vea la figura 15.10). De acuerdo con el diagrama de transición de estado, el acceso exitoso de un usuario genera como resultado la pantalla del menú principal del Sistema de servicios a los miembros de SoundStage esquematizada en la figura 15.17. Observe lo siguiente:

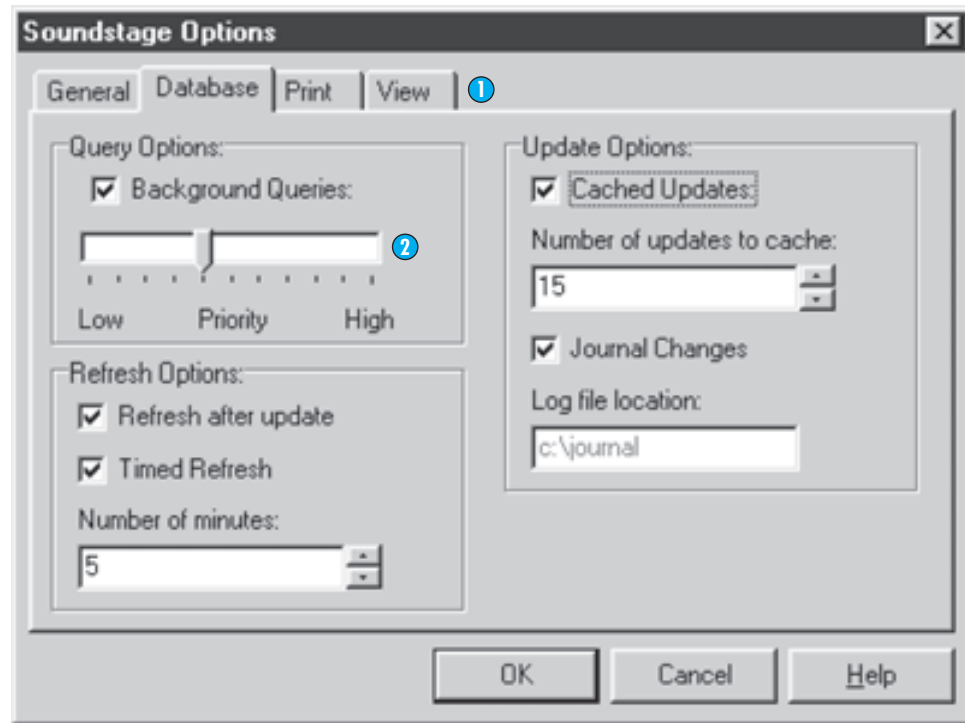
- 1 Los usuarios y sus privilegios de acceso son confirmados. Basándose en el privilegio de acceso de los usuarios, ciertas funciones serán habilitadas y deshabilitadas.
- 2 A través de una selección en el menu bar o de un vertical menu of buttons, el usuario puede completar operaciones de negocios comunes de los Servicios al Miembro. Estos botones permiten al usuario procesar transacciones apropiadas vía las pantallas de entradas diseñadas, cuyo prototipo se creó anteriormente. Se usaron etiquetas de texto para los botones porque el analista fue incapaz de establecer iconos (imágenes) con los que todos los usuarios podrían identificarse rápidamente como una representación de las operaciones. La barra de menú (menu bar) y los botones contienen teclas de acceso rápido para proporcionar al usuario con la flexibilidad de poder hacer una selección por medio del teclado o el ratón. Se usó un group box (cuadro de grupo) para asociar visualmente los botones que representan a las operaciones relacionadas.
- 3 El usuario tiene la capacidad de completar diversas operaciones de mantenimiento de rutina.

Por medio del menú bar de la pantalla del SISTEMA DE SERVICIOS AL MIEMBRO, los usuarios pueden elegir o establecer opciones para su sesión de trabajo. Esta pantalla nueva es esquematizada en la figura 15.18.

- 1 Esta pantalla utiliza *tabuladores (tabs)* como una manera de permitirle al usuario alterar cuatro conjuntos relacionados de opciones.
- 2 Se usa un *slider* control para permitir al usuario ajustar la prioridad de las consultas (queries). Este control es frecuentemente usado por entidades o elementos cuyos valores son presentados como una representación espacial y cuando es suficiente obtener un valor aproximado en vez de un valor preciso.

**FIGURA 15.18**

Pantalla de opciones y preferencias de SoundStage

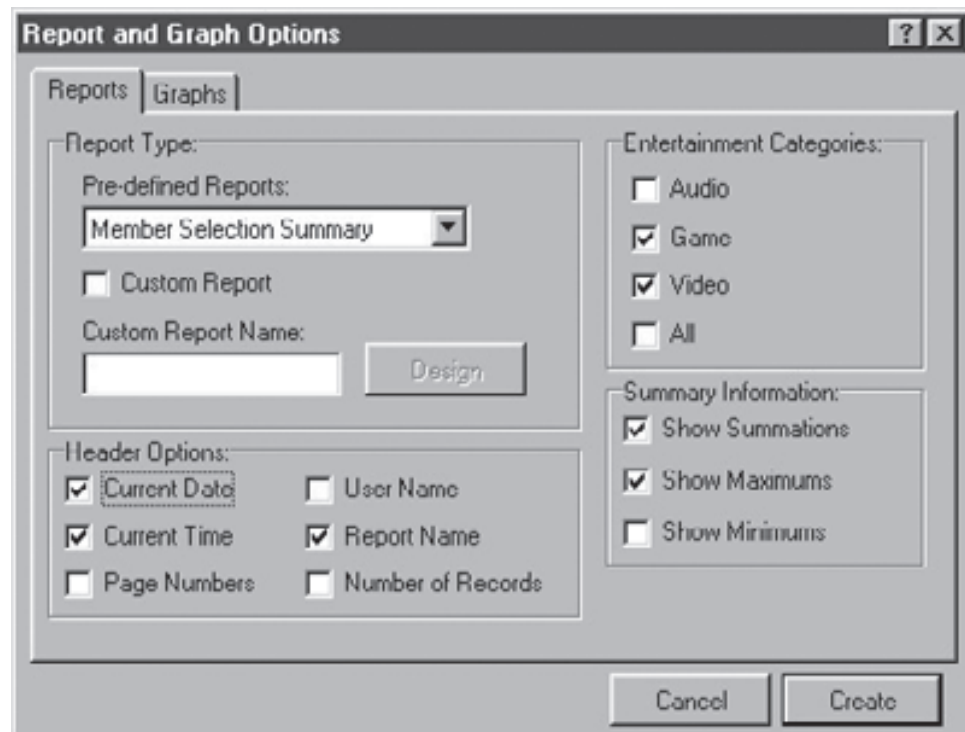


En realidad, el analista necesitaría hacer el prototipo del contenido y la apariencia de los tabuladores “General”, “Print,” y “View” así como también del tabulador “Database”. Según el diagrama de transición de estado, esta pantalla le devolverá el control a la ventana padre, SISTEMA DE SERVICIOS DEL MIEMBRO.

Según el diagrama de transición de estado, a los usuarios del sistema se les debe dar la oportunidad para especificar preferencias de adecuación de reportes a la medida. La figura 15.19 esquematiza una pantalla del prototipo que permite a los usuarios de SoundStage escoger un reporte particular (o una gráfica) y hacer a la medida su contenido.

**FIGURA 15.19**

Pantalla de reporte de diálogo hecho a la medida de SoundStage



Estudie el diagrama de transición de estado y las pantallas que hemos examinado para observar cómo funcionaría esta parte del diálogo global del sistema. Estudiando el conjunto completo de pantallas, usted puede descubrir que es necesario revisar algunas de ellas. Aspectos tales como color, la consistencia en los nombres de los botones más comunes, las opciones del menú, y otros detalles de forma y percepción pueden requerir revisarse. Finalmente, el apego a cualesquiera estándares que gobiernen los GUI deberá ser confirmado.

**Paso 3: Obtenga la retroalimentación del usuario** Poder ejecutar (o probar) la interfaz del usuario es una ventaja crucial de todos los ambientes para hacer prototipos a los que nos hemos referido a todo lo largo de este capítulo. La ejecución (o *la prueba*) de *la interfaz del usuario* implica que los usuarios del sistema experimenten y prueban el diseño de la interfaz antes de realizar la programación extensiva y la implementación real del sistema en funcionamiento. Los analistas pueden observar esta experimentación para mejorar el diseño.

A falta de herramientas para hacer el prototipo, el analista al menos deberá simular el diálogo con los usuarios del sistema a través de esquemas de pantalla. La retroalimentación del usuario es esencial en el diseño de la interfaz del usuario. El analista deberá motivar al usuario para participar en las pruebas de la interfaz de la aplicación. Finalmente, el analista deberá regresar a los pasos 1 y 2 cuando se conozcan los cambios necesarios.

Este capítulo proporcionó un panorama detallado del diseño y la formación de prototipos de la interfaz del usuario. Le recomendamos que repase el capítulo 3 para analizar dónde se sitúa el diseño de software en el proceso general del desarrollo de sistemas.

Mapa de aprendizaje



## Resumen

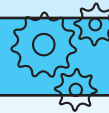
1. El diseño de la interfaz del usuario tiene que ver con el diálogo entre el usuario y la computadora. Se ocupa de todo desde el inicio o acceso al sistema hasta la presentación final de las salidas y entradas deseadas.
2. La mayoría de las interfaces de usuario se diseñan a través de la construcción rápida de prototipos usando ambientes rápidos de desarrollo de aplicación. Rara vez estos prototipos son totalmente funcionales.
3. En relación con el diseño de la interfaz del usuario, los usuarios del sistema pueden clasificarse ampliamente como expertos o novatos:
  - a) Un usuario experto es un usuario experimentado con la computadora que pasará un tiempo considerable usando programas de aplicación específicos. El uso de una computadora generalmente se considera no discrecional.
  - b) El usuario novato es un usuario de la computadora menos experimentado que generalmente usará una computadora con menos frecuencia, o aun ocasionalmente. El uso de una computadora puede verse como discrecional.
4. Varios factores humanos frecuentemente hacen que las personas tengan dificultad para aceptar los sis-

- temas de cómputo, incluyendo estos problemas de la interfaz:
- El uso excesivo de tecnicismos de la computadora y los acrónimos.
  - Diseño que no es obvio o menos que intuitivo.
  - La incapacidad para distinguir entre acciones alternativas (“¿Qué hago después?”).
  - Enfoques inconsistentes para la resolución de problemas.
  - Inconsistencias en el diseño.
5. Galitz propone los siguientes “mandamientos” primordiales de diseño de la interfaz del usuario:
- Entienda a sus usuarios y sus tareas.
  - Involucre a los usuarios en el diseño de la interfaz.
  - Pruebe el sistema en usuarios reales.
  - Practique el diseño iterativo.
6. Dado el tipo de usuario para un sistema, hay varios factores importantes de ergonomía que deberán ser incorporados en el diseño:
- El usuario del sistema siempre deberá darse cuenta de qué hacer después.
  - La pantalla deberá diseñarse de modo que los tipos diversos de información, instrucciones, y mensajes siempre aparezcan en la misma área general de despliegue.
  - Los mensajes, las instrucciones, o la información generada por el sistema deberá ser desplegada ampliamente como para permitirle al usuario del sistema leerla.
  - Usar con moderación los atributos de despliegue.
  - Deberán especificarse valores predeterminados para los campos y las respuestas que el usuario debe ingresar.
  - Anticiparse a los errores que los usuarios pueden cometer.
  - No deberá permitirse que un usuario continúe sin corregir un error.
  - El usuario del sistema nunca deberá obtener un mensaje operativo de sistema o un error fatal.
  - Si el usuario hace algo que pudiera ser catastrófico, el teclado deberá bloquearse para impedir cualquier entrada posterior, y deberá desplegarse una instrucción para llamar al analista o al soporte técnico.
7. El flujo global de pantallas y mensajes se llama un diálogo. En relación al tono del diálogo, se ofrecen los siguientes lineamientos:
- Use oraciones simples, gramaticalmente correctas.
  - No sea chistoso ni lindo.
  - No sea condescendiente.
- Con relación a la terminología usada en un diálogo de la computadora, las siguientes sugerencias pueden ser útiles:
- No use tecnicismos de la computadora.
  - Evite la mayoría de las abreviaturas.
  - Use términos simples.
  - Sea consistente en el uso de terminología.
  - Redacte las instrucciones cuidadosamente; use verbos de acción apropiados.
8. La mayoría de las interfaces del usuario de hoy son gráficas. La estructura básica de la interfaz gráfica del usuario (o GUI) se proporciona ya sea dentro del sistema operativo de la computadora o en el navegador del Internet.
9. El diseño global de una interfaz de usuario es realzado o restringido por las características disponibles del monitor de despliegue, el teclado, y los dispositivos apuntadores.
10. Hay varios estilos de interfaces gráficas del usuario, incluyendo diálogos accionados por menú, por instrucciones, y de pregunta-respuesta.
11. Las estrategias accionadas por menú requieren que el usuario seleccione una acción de un menú de alternativas. La implementación del menú de la interfaz gráfica del usuario puede incluir:
- Los pull-down y cascading menus (menús desplegables y de efecto de cascada).
  - Los tear-off y pop-up menus.
  - Los toolbar iconic menus (menús de barra de herramientas y de íconos).
  - Los hypertext e hyperlink menus (menús de hipertexto y de hipervínculo).
12. Las interfaces accionadas por comandos son diseñadas usando un diálogo basado en un conjunto de instrucciones. Pueden usarse tres tipos de sintaxis para el conjunto de instrucciones:
- La sintaxis basada en el lenguaje, que se construye alrededor de un lenguaje de comandos ampliamente aceptado que puede ser usada por el usuario para invocar acciones.
  - La sintaxis mnemónica, la cual se construye alrededor de las órdenes definidas para las aplicaciones de sistemas de información hechas a la medida.
  - La sintaxis del lenguaje natural, que permite a los usuarios introducir preguntas y comandos en su lengua materna.
13. Un estilo de question-answer dialogue (diálogo de pregunta-respuesta) se usa primordialmente para sustituir ya sea diálogos (o comunicación) accionados por menú o por instrucciones. Los usuarios son cuestionados con preguntas para las cuales suministran respuestas.
14. Los controles internos y la ayuda en línea son algunas consideraciones especiales que deberán considerarse en el diseño de la interfaz del usuario.
15. El diseño de la interfaz del usuario consta de tres pasos iterativos:
- Haga un mapa del diálogo de la interfaz de usuario.
  - Haga un prototipo del diálogo y de la interfaz del usuario.
  - Obtenga retroalimentación del usuario.



## Preguntas de repaso

1. ¿Por qué los usuarios del sistema deberán intervenir en el proceso de diseñar las interfaces del usuario?
2. ¿Quiénes son los usuarios expertos? ¿Por qué son llamados expertos?
3. ¿Por qué algunas interfaces de usuario pueden causar que los usuarios se sientan confundidos, aterrizados, o frustrados?
4. ¿Qué significa probar el sistema en usuarios reales?
5. ¿Qué deberíamos hacer para asegurar que los usuarios del sistema se den cuenta de qué hacer en el sistema?
6. ¿Cómo deberán manejar los errores las interfaces?
7. ¿Cuáles son algunos factores que deberán considerarse en cuanto a la terminología usada en los diálogos de la computadora?
8. ¿Por qué los navegadores de Web se están haciendo más importantes al diseñar aplicaciones?
9. Explique la paginación y el desplazamiento en pantalla.
10. ¿Qué deberemos considerar cuando diseñamos teclas de función para nuestras aplicaciones?
11. ¿Por qué se usan las plumas en las aplicaciones?
12. ¿Cuál es la relación entre las ventanas y los marcos?
13. ¿Cuáles son las características de un pop-up menu?
14. ¿Cuáles son los pasos del proceso de diseño de la interfaz del usuario?
15. ¿Cuál es la herramienta que se usa para facilitar el mapeo de un diálogo?



## Problemas y ejercicios

1. Un colega diseñador le ha pedido que revise el diálogo que va a usarse en varias pantallas para una aplicación nueva. ¿Piensa usted que los siguientes mensajes cumplen con los lineamientos para el tono y la terminología? Explique su respuesta según se necesite.
  - a) Un mensaje de error que dice la FECHA DE DESCARGA DEBE SER EN LA FECHA DE ADMISIÓN O DESPUÉS.
  - b) Una instrucción que dice INTRODUZCA AHORA EL NOMBRE DEL CLIENTE.
  - c) Un mensaje de error que dice LOS DATOS ESTÁN AHORA TAN FUERA DEL ALCANCE QUE HAN SALIDO DEL SISTEMA SOLAR.
  - d) Una pregunta que cuestiona a los clientes ¿QUIEREN RDF EL ACR ANTES DE "UCI'ING" CMIS?
  - e) Un mensaje de error que dice NO SE PREOCUPE -NO TODOS LO HACEN BIEN LA PRIMERA VEZ.
2. Las interfaces accionadas por menú, aunque más antiguas que las de la interfaz gráfica del usuario, son todavía muy comunes. ¿Qué tipo de estrategia de diálogo del usuario emplea una interfaz accionada por menú? ¿Cuál es la principal diferencia entre las interfaces accionadas por menú y las interfaces GUI? ¿Cuál es la ventaja principal de una interfaz accionada por menú, y cuál es su desventaja principal?
3. Seleccione una aplicación que se use en su organización o escuela y que tenga más de una pantalla de la interfaz del usuario. También, trate de seleccionar una aplicación que usted no haya usado antes, o al menos una que no haya usado por algún tiempo. Pruebe el diseño de la interfaz del usuario contra cada uno de los lineamientos de ergonomía descritos en el capítulo. Asegúrese de ingresar datos incorrectos y correctos.
 

Describa la aplicación y las pantallas de la interfaz del usuario que usted probó. ¿Cuántos lineamientos pudieron satisfacer la aplicación, y cuáles fueron? ¿Cuántos lineamientos no fueron satisfechos, y cuáles fueron?
4. Considerando la aplicación descrita en el ejercicio anterior, ¿qué cambios haría usted a las pantallas de la interfaz del usuario para hacerlas cumplir con los lineamientos de ergonomía descritos en este libro? Sea específico en sus respuestas. Haga que un par de condiscípulos y/o compañeros de trabajo revisen sus cambios, uno como usuario y el otro como diseñador. ¿Encontraron problemas nuevos?
5. Conteste las siguientes preguntas como falsas/verdaderas. Explique sus respuestas como sea necesario.
  - a) Deberán usarse diferentes verbos de acción en el diálogo de la pantalla para describir las acciones requeridas del teclado para añadir variedad e interés.
  - b) La mayoría de los gerentes son usuarios expertos, porque necesitan un nivel alto de experiencia con la PC con objeto de administrar eficazmente.
  - c) Las organizaciones deberán esperar que los diseñadores expertos, quienes vienen muy bien recomendados y que están en la cima de la escala salarial, necesitarán refinar y modificar sus diseños de la interfaz del usuario varias veces antes de que el resultado sea satisfactorio para la organización.
  - d) El diseño de la interfaz del usuario de *Windows* a menudo toma prestado de los estilos y técnicas de la interfaz de Web.
  - e) Las aplicaciones necesitan sólo un tipo de menú de ayuda o de diálogo.
  - f) Los usuarios aprecian los mensajes listos o humorísticos de la pantalla.
  - g) El proceso de designación de las interfaces del usuario es directo y fácil de entender.

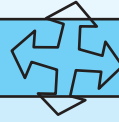
6. En el diseño de las interfaces del usuario, debe considerarse la seguridad de la información y la privacidad. Describa algunos de los lineamientos y consideraciones que deben ser tomadas en consideración para incorporar controles internos al diseño de la interfaz del usuario.
7. Rellene los espacios vacíos para las siguientes declaraciones:
- Algunas \_\_\_\_\_ interfaces usan una \_\_\_\_\_ sintaxis que permite a los usuarios hacer preguntas en su \_\_\_\_\_.
  - El \_\_\_\_\_ paso en el diseño de la interfaz del usuario es \_\_\_\_\_ para los \_\_\_\_\_ pasos como parte del \_\_\_\_\_ proceso hasta que los usuarios estén \_\_\_\_\_ con el diseño.
  - SQL usa \_\_\_\_\_ sintaxis que permite \_\_\_\_\_ a los usuarios \_\_\_\_\_ la base de datos.
  - Los buenos \_\_\_\_\_ lineamientos incluyen al usuario \_\_\_\_\_ posible \_\_\_\_\_, y \_\_\_\_\_ el usuario de una acción \_\_\_\_\_ antes de que la acción sea \_\_\_\_\_.
  - Un lineamiento para establecer \_\_\_\_\_ los privilegios de los usuarios es basar sus privilegios en \_\_\_\_\_, no en \_\_\_\_\_.
8. Coteje las definiciones o ejemplos en la primera columna con los términos en la segunda:
- |  |   |
|--|---|
| A. Frecuentemente requirió navegación en un menú de nivel múltiple                       | 1. Plataforma independiente             |
| B. La secuencia global de pantallas y los mensajes de una aplicación                     | 2. La interfaz al estilo del consumidor |
| C. Enfoque de pantalla completa para exhibir cada vez el área vista por el usuario       | 3. Terminales emuladoras                |
| D. La información se mueve una línea arriba o abajo a la vez                             | 4. El diagrama de transición de estado  |
| E. Experiencia con la computadora en campos de estudio múltiples relacionados            | 5. El menú de íconos                    |
| F. Pantalla de Windows que emplea una "cara" artística parecida a la Web                 | 6. Screen thrashing                     |
| G. Zonas independientes dentro de una ventana  | 7. La sintaxis mnemónica                |
| H. Las interfaces del usuario que no son dependientes de un sistema operativo específico | 8. B2C                                  |
| I. Herramienta gráfica usada para mostrar la variación y secuencia de la pantalla        | 9. Diálogo                              |
| J. Software para exhibir el formato de la pantalla de la mainframe en una ventana        | 10. Desplazamiento en pantalla          |
| K. Interfaz de lenguaje de comandos significativa para el usuario                        | 11. Informática                         |
| L. Funcionalidad basada en las transacciones de negocios y el consumidor                 | 12. Paginación                          |
| M. Representación pictórica del menú de opciones en el cuerpo principal de la ventana    | 13. Marcos                              |
9. Se le ha pedido diseñar una serie de pantallas de la interfaz de usuario que serán usadas por empleados y clientes. Usted con orgullo desarrolla sus prototipos, esperando aplausos y reconocimiento. En lugar de eso, los usuarios que prueban su prototipo —tanto clientes como empleados— están confundidos, frustrados, y aun enojados. ¿Cuáles son algunos de los problemas de la interfaz que podrían haber encontrado que los hicieron reaccionar así? Cuando usted se dirige a su jefe para un consejo, su jefe le formula una serie de preguntas con respecto a los pasos que usted siguió. ¿Cuáles son las preguntas que probablemente le formulará?
10. En el pasado, la mayoría de las aplicaciones de software venían con un grueso manual del usuario. Muchos de estos manuales del usuario han desaparecido cediendo el paso a los sistemas de ayuda en línea sofisticados y a los tutoriales. Si usted es un diseñador de sistemas que diseña el sistema de ayuda para una aplicación nueva, ¿Cuáles son algunas de las consideraciones importantes que debe tener en mente?
11. No es raro para una aplicación usar centenares de pantallas, ventanas, y dialogue boxes (cajas de diálogo) en su interfaz de usuario. Coordinar el orden y las condiciones bajo las cuales éstos aparecen puede ser un proceso difícil que está propenso a errores. Para ayudar a coordinar y documentar este proceso, se usan diagramas de transición de estado (STD) para ilustrar las condiciones bajo las cuales aparecen las pantallas, las ventanas, y las cajas de diálogo, así como su secuencia.
- Leve una aplicación a su organización o a su escuela con la cual usted está familiarizado. Cree un diagrama de transición de estado para una parte del sistema, usando la figura 15.16 como ejemplo.
12. Suponga que usted es parte de un equipo de proyecto que ha sido contratado por una compañía que se mueve de la tecnología de la mainframe a la tecnología del cliente/servidor. Usted está trabajando en el diseño de la interfaz del usuario. La compañía quiere que esta aplicación establezca el tono para el desarrollo de aplicaciones posteriores. Usted tiene rienda suelta para desarrollar las convenciones y estándares para las pantallas de la interfaz de usuario



que se usarán para la apariencia y percepción de las aplicaciones posteriores. Cree una lista de una cuartilla de lo que usted crea que sean las convenciones y los estándares más importantes.

13. Las aplicaciones GUI y de Web proporcionan a los usuarios de una colección variada de rutas o cami-

nos a través de las diferentes partes de la aplicación. El precio que se paga por proporcionar esta amigabilidad y facilidad de uso recae en un diseño y programación complicados. ¿Es posible tener interfaces de fácil manejo y amigables que no requieran una programación y un diseño complicados?



## Proyectos e investigación

1. Este capítulo, así como los dos capítulos anteriores, se centraron en la captura y la presentación de los datos e información. En 1983, Edward Tufte escribió un libro llamado *The Visual Display of Quantitative Information*. Muchas revistas destacadas consideran que este libro es *el* trabajo definitivo en este tema. Desde entonces, Tufte ha escrito muchos otros libros y artículos sobre el despliegue de información y datos, cómo modela las formas en que pensamos, y cómo puede tener consecuencias profundas. Vaya al sitio Web de Tufte en <http://www.edwardtufte.com/tufte> y lea algunos de los artículos y debates del foro. El sitio Web de Tufte también contiene numerosos enlaces con otros sitios; usted puede encontrarlos igualmente interesantes y/o valiosos.
  - a) Describa algunos de los artículos y los puntos de vista que usted encontró en el sitio Web.
  - b) ¿Qué es “chatarra de diagramas”?
  - c) ¿Qué tiene que decir Tufte acerca de los diagramas de administración de proyectos? ¿Son sus sugerencias viables en su opinión?
  - d) ¿Cuál es el punto de vista de Tufte sobre el *Power Point* de Microsoft? ¿Está usted de acuerdo? ¿Por qué sí o por qué no?
  - e) Suponga que usted inicia una carrera como diseñador de sistemas. Después de leer el material en el sitio Web de Tufte, describa esos conceptos, lineamientos, o puntos de vista, si los hay, qué piensa usted con respecto a que son absolutamente vitales para todo diseñador de sistemas entenderlos y aplicarlos.
2. El libro de texto hace referencia a otro autor, Wilbert Galitz. Es uno de varios escritores contemporáneos (varios de los cuales son citados en la sección de lecturas recomendadas), quienes son reconocidos como los líderes en el área del diseño de la interfaz humana. Use Internet para investigar artículos recientes y debates de foro en el tema de los lineamientos del diseño de la interfaz humana y de la ergonomía.
  - a) Describa los artículos que usted encontró, incluyendo sus autores y puntos de vista
  - b) Discuta y compare cualesquiera puntos de vista contrastantes que usted encontró en este tema.
  - c) ¿Cuáles son los pensamientos de los autores sobre la tendencia hacia la convergencia de *Windows* y de las interfaces con el navegador?
    - ¿Piensan ellos que finalmente habrá poca diferencia entre las dos si es que la hay?
  - d) ¿Cuáles son sus predicciones, si las hay, en relación con las innovaciones tecnológicas que fundamentalmente pueden cambiar el diseño de la interfaz humana en el futuro?
  - e) Con base en su investigación, ¿usted considera que la investigación en el área de ergonomía y diseños de la interfaz humana es casi tan sofisticada y adelantada como se dice que es? ¿Por qué sí o por qué no?
3. Hasta ahora, hemos investigado los puntos de vista de los líderes en el campo de presentación de información y diseño de la interfaz humana para sistemas de cómputo. ¿Qué hay acerca de los expertos en otras áreas?
  - a) Encuentre y entreviste a varios artistas gráficos locales y diseñadores en su área. ¿Cuáles son sus pensamientos y sus puntos de vista en relación con cómo los gráficos pueden utilizarse como un método de comunicación?
  - b) ¿Fueron sus puntos de vista conceptualmente similares o diferentes en comparación con esos que usted investigó en las preguntas previas?
  - c) ¿Encontró usted cualquier cosa que podría ser de beneficio para usted como diseñador de sistemas?
  - d) ¿Qué hay acerca del campo de la psicología industrial? Haga alguna investigación en esta área, y/o entreviste a algunos educadores o profesores en su escuela que dominen esta disciplina. ¿Fueron diferentes sus puntos de vista o similares a los otros que usted investigó o entrevistó?
  - e) ¿Puede pensar usted acerca de algún campo o disciplina que podría ser de utilidad para el diseño de las interfaces humanas? Si es así, invéstiguelos y reporte lo que usted considera como pertinente y valioso.
4. Las herramientas automatizadas del diseño de pantallas se están haciendo cada vez más poderosas y sofisticadas. Use la Web para encontrar varias de las herramientas de diseño de vanguardia. Vaya a los sitios Web de los fabricantes y revise sus características. Si las versiones de prueba están disponibles, descárguelas.
  - a) ¿Qué herramientas automatizadas de diseño de pantallas encontró? ¿Quién las fabrica?
  - b) Compare y contraste sus características y su funcionalidad. Describa sus características diferentes en una matriz.

- c) Si usted fuera un diseñador independiente, ¿cuál escogería, si hubiera alguna? ¿Por qué?
- d) ¿Esperaría usted ver una diferencia significativa en su productividad al utilizar una de estas herramientas? ¿Qué tanta diferencia.
- e) ¿Siente usted que el uso de estas herramientas realzaría u opacaría su creatividad? Explique su respuesta.
5. Frecuentemente se escucha el término de “rediseñar” al gobierno y hacerlo que opere más como la iniciativa privada. Esto suscita la pregunta de si hay una diferencia fundamental entre los sectores público y privado, y si esto puede tener un impacto en cómo los sistemas deberán diseñarse en forma diferente, dependiendo de si están destinados a un gobierno o una organización del sector privado.
- a) Haga una encuesta entre los diseñadores de sistema en los sectores privado y público. Pregúnteles cuáles son sus prioridades y problemas importantes en términos del diseño de interfaces humanas.
- b) ¿Qué diferencias encontró usted?
- c) ¿Qué similitudes?
- d) ¿Piensa usted que se tiene mucho en común de modo que el mismo conjunto de lineamientos pueda aplicarse a las agencias del sector público y privado? ¿Por qué o por qué no?
- e) ¿Dada una selección y dado que el sueldo y los beneficios fueran los mismos, ¿preferiría usted ser un diseñador de sistemas en un organismo público? ¿Una agencia privada? ¿Por qué sí o por qué no?
6. Algunos consideran que el diseño de las pantallas de la interfaz para B2C y B2B tienen un objetivo fundamentalmente diferente en comparación con otros tipos de pantallas de la interfaz. Específicamente, el propósito de estos sitios Web es seducir a los consumidores y los negocios para comprar sus productos o servicios.
- a) Indague artículos sobre este tema. ¿Qué encontró usted, y cuáles fueron los puntos de vista?
- b) Resuma la diferencia en la apariencia, si la hay, entre diseñar pantallas convencionales de la interfaz humana de entrada/salida y diseñar pantallas para los sitios de Web B2C y B2B?
- c) ¿Está de acuerdo usted con estos puntos de vista? ¿Por qué sí o por qué no?
- d) ¿Qué tipo de capacitación piensa usted que sería más valiosa para los sitios Web B2C y B2B; una enfocada en diseño de sistemas o una en comercialización y publicidad? Explique su respuesta.

## Casos breves



Hay un debate, que comienza en la página 517, sobre los aspectos de factores humanos y ergonomía en el diseño de la interfaz del usuario. El espíritu del debate es que es imperativo que entendamos a las *personas* que usarán el sistema y hagamos una interfaz de sistema que *ellas* comprendan y puedan usar. Pero éste no es un asunto académico; es un asunto de habilidades de las personas y de comprensión de las mismas.

- Entreviste a alguien a quien usted no conozca bien y que esté completamente fuera del área técnica. Su meta es entender a esa persona, sus necesidades y deseos en relación a la automatización por computadora. Cosas que usted necesita considerar en su entrevista:
  - Entienda a esa persona como una *persona*: ¿Quién es? ¿Cuáles son sus preferencias y sus miedos? ¿Tiene un cónyuge? ¿Niños? ¿Qué hay acerca de los deportes? ¿Los pasatiempos? ¿Trabaja dentro o fuera de la casa? Si usted la entrevista en su propio “espacio” (casa, oficina, etc.), tome nota de los efectos personales que están a la vista. ¿Qué le dicen estas cosas sobre esa persona?
  - Entiéndala como un usuario de la computadora: ¿Cuáles son sus experiencias con la misma? ¿Para qué tipos de cosas ha usado la computadora? ¿Para qué *no usaría* una computadora?
- Usando el conocimiento que usted obtuvo de su entrevista en el caso breve 1, diseñe una interfaz para el individuo que usted entrevistó. ¿Qué modificaciones del diseño de la interfaz está usted haciendo a fin de que el programa se ajuste al individuo? Explíquelo en detalle, y envíele los resultados a su profesor.
- Encuéntrese con la persona que usted entrevistó en el caso breve 1 y preséntele el prototipo del diseño que usted creó. Obtenga su retroalimentación sobre el diseño. ¿Le gusta? ¿Pudo navegar las páginas? ¿Qué hay acerca del diseño de las entradas? ¿Hay alguna cosa que cambiaría? ¿Qué le gusta y que le desagrada específicamente acerca de la interfaz que usted creó? Otra vez, observe su lenguaje corporal. ¿Está esa persona diciéndole todo? ¿Qué está diciéndole su lenguaje corporal? Observe su influencia en la situación. Documente la entrevista y envíele los resultados a su profesor.

- Basado en su segunda entrevista (caso breve 3), revise su diseño de la interfaz. Luego entregue a su profesor su trabajo de los tres casos breves previos y esta revisión como un documento profesional y

completo. Asegúrese de incluir un debate breve de lo que usted aprendió de la persona a quien usted entrevistó, y de esta experiencia.



## Ejercicios de equipo e individuales

- Por equipos o individual: Escriba todos los tecnicismos comunes que usa cuando describe un sistema. Por equipos o individual, escriba una versión no técnica, “que cualquiera entendería” de cada uno de los términos técnicos en su lista. Intercambie opiniones en un formato de mesa redonda en la clase.
- Por equipos: Considere la TV. ¿Qué la hace tan fácil de usar? (Casi todo el mundo parece ser capaz de

verla.) Si usted rediseñara cualquier cosa, ¿qué sería, y cómo lo rediseñaría?

- Individual: Haga una reflexión propia de sus fortalezas y sus debilidades. *a)* Tome un momento y disfrute de algo positivo acerca de usted mismo. *b)* Identifique una acción específica que usted pueda tomar para enfrentar una debilidad que posea, y luego ejecute la acción necesaria.



## Lecturas recomendadas

Andres, Clay. *Great Web Architecture*. Foster City, CA: IDG Books Worldwide. Éste es un título interesante. Usa un enfoque de “diseño por ejemplo” basado en las entradas de “arquitectos destacados de la Web” para ilustrar y discutir los sistemas basados en Web, incluyendo muchos con aspectos de comercio y negocio electrónico. Ésta no es una obra académica, y sin embargo es interesante.

Galitz, Wilbert. *User-Interface Screen Design*. Nueva York: Wiley QED, 1993. Ignore la fecha. Este libro sigue siendo nuestro libro favorito sobre el diseño de las interfaces porque es muy sólido conceptualmente. Galitz enseña aquí sobre el diseño de las interfaces de las estaciones de trabajo, las PC, y las mainframe, basándose en principios y lineamientos bien pensados. Estamos esperando la actualización. ¡Sería bueno que pudiéramos tener la capacidad de desarrollar un curso opcional completo formado alrededor de este notable libro!

Galitz, Wilbert. *It's Time to Clean Your Windows: Designing GUI That Work*. Nueva York: John Wiley & Sons, 1994. Éste es otro excelente libro que suministra una referencia no sesgada sobre el diseño de las interfaces gráficas.

Hix, Deborah, y H. Rex Hartson. *Developing User Interfaces: Ensuring Usability through Product & Process*. Nueva York: John Wiley & Sons, 1993. John Wiley & Sons deben monopolizar los libros de diseño de interfaces del usuario. Estos autores tienen raíces académicas. El libro es un poco difícil de leer, sin embargo muy bien organizado y escrito. Nos agrada especialmente la integración con el análisis y el diseño de sistemas.

Horton, William K. *Designing & Writing Online Documentation: Help Files to Hypertext*. Nueva York: John Wiley & Sons, 1990. Pudimos proporcionar únicamente un vistazo superficial de este importante tema.

Mandel, Theo. *Elements of User Interface Design*. Nueva York: John Wiley & Sons, 1997. Aquí hay un libro algo más nuevo y muy comprensivo que incluye parte de los primeros fundamentos de diseño para la Web.

Martin, Alexander, y David Eastman. *The User Interface Design Book for the Applications Programmer*. New York: John Wiley & Sons, 1996.

Microsoft Corporation. *Microsoft Windows User Experience: Official Guidelines for User Interface Developers and Designers*. Redmond, WA: Microsoft Press, 1999. Éste es el estándar oficial para el diseño de las interfaces del usuario para *Windows*. Hay muchos avisos de las intenciones de Microsoft para maximizar la experiencia del usuario.

Schmeiser, Lisa. *Web Design Templates Sourcebook*. Indianapolis, IN: New Riders Publishing, 1997. Ésta no es una obra académica y ni siquiera una obra tradicional del mercado profesional. Atrajo nuestra atención porque usa un enfoque de plantilla basada en HTML (más de 300) para presentar diseños que finalmente puedan evolucionar hasta productos terminados. Desde nuestra perspectiva, esto representa un intrigante giro sobre el modelo para hacer prototipos.

Weinschenk, Susan, y Sarah C. Yeo. *Guidelines for Enterprise-Wide GUI Design*. Nueva York: John Wiley & Sons, 1995. Es claro que John Wiley & Sons es la editorial líder del mercado en este tema.

Página 60	Getty/PhotoDisc (DIL)
Página 70	Getty/PhotoDisc (DIL)
Página 75	Getty/PhotoDisc (DIL)
Página 79	Getty/PhotoDisc (DIL)
Página 82	Getty/PhotoDisc (DIL)
Página 82	Getty/PhotoDisc (DIL)
Página 85	Getty/PhotoDisc (DIL)
Página 85	Getty/PhotoDisc (DIL)
Página 85	Getty/PhotoDisc (DIL)
Página 85	Getty/PhotoDisc (DIL)
Página 86	Getty/PhotoDisc (DIL)
Página 103	Getty/PhotoDisc (DIL)
Página 111	Getty/PhotoDisc (DIL)
Página 119	Getty/PhotoDisc (DIL)
Página 129	Getty/PhotoDisc (DIL)
Página 133	Getty/PhotoDisc (DIL)
Página 136	Getty/PhotoDisc (DIL)
Página 136	Getty/EyeWire (DIL)
Página 352	Getty/PhotoDisc (DIL)
Página 353	Getty/PhotoDisc (DIL)
Página 359	Getty/PhotoDisc (DIL)
Página 359	Getty/PhotoDisc (DIL)
Página 359	Getty/PhotoDisc (DIL)
Página 360	Getty/PhotoDisc (DIL)
Página 360	Getty/PhotoDisc (DIL)
Página 487	Blackberry Wireless Handheld <sup>TM</sup> de Research In Motion (RIM)



Los números de página en **negritas** indican la ubicación de definiciones de glosario.

## A

Activado por la voz, 487

**actividad transversal del ciclo de vida** *Toda actividad que se traslapa en múltiples fases del proceso de desarrollo de sistemas. Entre sus ejemplos, se cuentan la identificación de hechos, documentación, presentación, estimación, análisis de factibilidad, administración de proyectos y procesos, administración del cambio y administración de la calidad.* **68-69**

Actividades de negocios

arquitecturas de aplicaciones, 389-390

diseño de entradas, 507-508

diseño de salidas, 472-475

**actor** *Cualquier cosa que necesite interactuar con el sistema para intercambiar información,* **189**

agentes externos, 284

glosario, 193

identificación, 193

ADE; véase **ambiente de desarrollo de aplicaciones**

**administrador de base de datos (database administrator, DBA)** *Especialista responsable de la tecnología, el diseño, la construcción, la seguridad, el respaldo, la recuperación y el ajuste del desempeño de las bases de datos,* **10, 424**

**administración de calidad total** *Enfoque integral para facilitar mejoras y administración de calidad dentro de una empresa.*

**administrador de datos** *especialista en bases de datos responsable de la planeación, definición, arquitectura y administración de datos,* **424**

**administración de proceso** *Actividad continua que documenta, enseña, supervisa y mejora el uso de los métodos (el "proceso") y estándares que elige una organización para el desarrollo de sistemas. La administración de procesos se relaciona con fases, actividades, productos y normas de calidad que se deben aplicar de manera constante a todos los proyectos.* **54, 69**

**administración de proyectos** *Proceso de definir el alcance, planear, dotar de personal, organizar, dirigir y controlar un proyecto para desarrollar un sistema de información con costo mínimo, dentro del tiempo especificado y con calidad aceptable,* **16, 54, 69, 72**

**administrador de proyecto** *Profesional experimentado que acepta la responsabilidad de planear, monitorear y controlar proyectos con respecto a un calendario, presupuesto, entregables, satisfacción del cliente, normas técnicas y calidad del sistema,* **16**

**administración de requerimientos (o requerimientos)** *Proceso de administrar los cambios de los requerimientos,* **156-157**

**administración de riesgos** *Proceso de identificar, evaluar y controlar los problemas que podrían surgir con un proyecto antes de que se conviertan en una amenaza para la terminación exitosa del proyecto mismo con la puesta en práctica del sistema de información. La administración de riesgos se lleva a cabo mediante el análisis o evaluación de riesgos,* **56**

**agente** *Objeto de software reutilizable que puede funcionar en diferentes aplicaciones y redes,* **534**

**agente externo** *Una persona, unidad organizacional, sistema u organización ajenos a otro sistema que interactúan con este último. También llamado entidad externa,* **261**

en diagramas de flujo de datos, 259, 261-262

en diagramas de contexto de flujo de datos, 281, 284

eventos iniciados por, 283

físico, 379

nombres, 262

Agujeros grises, 267

Agujeros negros, 267

**alcance** *Límites de un proyecto: las áreas de un negocio que el proyecto podría atender (o no),* **111**

**almacén de datos** *Datos guardados para uso posterior. Sinónimo de archivo y base de datos,* **262**

en diagramas de flujo de datos, 259, 262-263

entidades y, 262, 291

externos, 281

físicos, 379

nombres, 262-263, 379

**almacén de datos de operación (operational data store, ODS),** 25, 67

Almacenamiento físico de datos, 379

Amazon.com, 390

**ambiente de desarrollo de aplicaciones (application development environment, ADE)** *Herramienta integrada para desarrollo de software que proporciona todas las funciones necesarias para el desarrollo de nuevo software de aplicaciones con rapidez y calidad máximas. Un sinónimo muy usado es ambiente de desarrollo integrado (integrated development environment, IDE),* **89-91**

interfaz de usuario, 38

uso de, 34-35, 516

**ambiente de desarrollo de software (software development environment, SDE)** *Lenguaje y conjunto de herramientas para el desarrollo de aplicaciones,* **398-400**

ambiente de desarrollo integrado (*integrated development environment, ADE*); véase **ambiente de desarrollo de aplicaciones**

Ambler, Scott W., 21, 96, 209

**ampliación del alcance** *Fenómeno común en el que se incrementan los requerimientos y expectativas de un proyecto, frecuentemente sin considerar el efecto en su presupuesto y cronograma,* **62**

**análisis basado en modelos** *Una estrategia de solución de problemas que hace énfasis en trazar modelos de sistemas de imágenes para documentar y validar los sistemas existentes o propuestos. En última instancia, el modelo de sistema se convierte en el plano para diseñar y construir un sistema mejorado,* **103-105**

**análisis de causa y efecto** *Técnica en la que se estudian problemas para determinar sus causas y efectos,* **122**

**análisis de datos** *Técnica usada para mejorar un modelo de datos, frente a su implantación como base de datos,* **241, 428-429**

**análisis de diferencias** *Comparación de los requerimientos de negocios y técnicos de un paquete de aplicación comercial contra las características y capacidades de un paquete de aplicación comercial específico para definir los requerimientos que no se pueden satisfacer;* **83**

**análisis de factibilidad** *Proceso con el que se mide la factibilidad,* **69, 312**

en la fase de análisis de decisiones, 137-139

evaluación del software comercial, 363-364

puntos de verificación durante el análisis de sistemas, 314-315

**análisis de sistemas de información** *Las fases de desarrollo de un proyecto de desarrollo de sistemas de información que se centran principalmente en los problemas y requerimientos de negocios, con independencia de la tecnología que pueda usarse o se use para implantar una solución al problema,* **102**; véase también **análisis de sistemas**

**análisis del sistema** *Estudio del dominio de un problema de negocios para recomendar mejoras y especificar los requerimientos del negocio y las prioridades para la solución,* **141**

**análisis del rendimiento sobre la inversión (ROI)** *Técnica en la que se compara la rentabilidad de soluciones alternas durante el ciclo de vida del sistema,* **323**

**análisis de la recuperación de la inversión** *Técnica para determinar si una inversión se recuperará y cuándo lo hará, 321*

**análisis estructurado** *Técnica centrada en procesos y operada por modelos que se usa para analizar un sistema existente, para definir los requerimientos de negocios de un nuevo sistema o para ambos objetivos. Los modelos son imágenes que ilustran los componentes del sistema: procesos, entradas, salidas y archivos, 104-105; véanse también modelado de datos; modelado de procesos*

**análisis orientado a objetos (object-oriented analysis, OOA)** *Estrategia usada para: 1) estudiar objetos existentes con el fin de indagar si se pueden reutilizar o adoptar para nuevos usos, y 2) definir objetos nuevos o modificados que se combinarán con objetos existentes en una aplicación de cómputo de negocios útil, 141; véase también modelado de objetos*

**análisis rápido de arquitectura (o prototipos)** *Estrategia que intenta derivar modelos de sistemas (como se describe con antelación en la misma sección del capítulo) a partir de sistemas existentes o descubriendo prototipos, 107*

Análisis y diseño estructurados, 64, 76-77, 277, 375

**análisis y diseño orientados a objetos (object-oriented analysis and desing, OOAD)** *Conjunto de herramientas y técnicas para el desarrollo de sistemas que utiliza tecnología basada en objetos para construir un sistema y su software, 77*

**analista de sistemas** *Especialista que estudia los problemas y necesidades de una organización para determinar la forma en que las personas, los datos, los procesos y la tecnología de la información pueden lograr óptimamente mejoras para la empresa, 11*

capacidades necesarias, 14-15

ética, 15

funciones, 11-12

lugar en la organización, 12-13

perspectivas de carrera, 13, 141

relaciones con otros interesados, 7, 11

Analistas; véase **analista de sistemas**

Anderson, John F., 182

Andres, C., 481, 513, 547

Andrews, D. C., 182

**aplicación de administración de proceso** *Herramienta automatizada que ayuda a documentar y gestionar un método y sus rutas, productos y normas de administración de la calidad. Un sinónimo emergente es methodware, 91*

**aplicación de administración (gerencia) de proyectos** *Herramienta automatizada que ayuda a planear las actividades de desarrollo de sistemas (de preferencia, con el uso de métodos aprobados), estimar y asignar recursos (lo que incluye personal y costos), planear actividades y recursos, vigilar el avance contra el plan y el presupuesto, controlar y modificar el plan y recursos, e informar sobre el avance del proyecto, 91; véase también Microsoft Project*

Aplicaciones habilitadas para web, 424; véase también **negocios electrónicos**

Apple

Macintosh, 520

Mac/OS, 77

**archivo** *Conjunto de todas las instancias de una estructura de registros dada, 418, 422; véase también almacén de datos*

comparación con bases de datos, 418-419

de almacenamiento, 422

de auditoría, 422

de búsqueda de tabla, 422

de transacciones, 422

diseño, 429

documento, 422

maestro, 422

muestreo, 158-159

pros y contras, 418-419

**archivo de archivos** *Tabla que contiene registros del archivo maestro y de transacciones que han sido borrados del almacenamiento en línea, 422*

**archivo de auditoría** *Tabla que contiene registros de las actualizaciones de otros archivos, 422*

**archivo de búsqueda en una tabla** *Tabla que contiene datos relativamente estáticos que pueden compartirse, 422*

**archivo de documento** *Tabla que contiene datos históricos, 422*

**archivo de inventario (audit file)** *Tabla que contiene registros de actualizaciones de otros archivos, en especial para recuperar datos "perdidos", 422*

**archivo de transacciones** *Tabla que contiene registros en los que se describen eventos de negocios, 422*

**archivo de registro** *Tabla que contiene datos históricos, 422*

**archivo histórico** *Tabla que contiene registros de archivos maestros y de transacciones que se han eliminado del almacenamiento en línea, 422*

**archivo maestro** *Tabla de registros que son relativamente permanentes, 422*

Archivos convencionales; véase **archivo**

Armour, Frank, 193n, 209

**arquitectura de aplicaciones** *Especificación de las tecnologías que deben usarse para implantar los sistemas de información, 374*

Arquitectura de aplicaciones, 401

Arquitectura de aplicaciones empresariales, 400-401

**arquitectura de base de datos** *Tecnología de base de datos usada como soporte para la arquitectura de los datos, 424-426*

**arquitectura de datos** *Definición de la manera en que se deben desarrollar los archivos y las bases de datos, 423-424*

**arquitectura de sistemas de información** *Marco de referencia en el que los diversos participantes, con perspectivas distintas, pueden organizar y ver los componentes fundamentales de los sistemas de información, 26*

normas, 53-54

perspectivas de los participantes, 26-27

Asistente digital personal (PDA); véase **Computadoras de bolsillo**

Asistentes de, 534

**asociación** *Una relación entre un actor y un caso de uso en la que interactúan entre sí, 190*

Association for Information Systems (AIS), 160

Association for Information Technology Professionals (AITP), 160

ATM; véase **Cajeros automáticos**

**atributo** *Una propiedad o característica descriptiva de una entidad. Sus sinónimos incluyen elemento, propiedad y campo, 214; véase también campo*

compuesto, 214

derivados, 244-246

dominios, 214, 239, 275

en flujos de datos, 272

tipos de datos, 214, 275

valores por omisión, 215

**atributo compuesto** *Un atributo que consiste en otros atributos. Sus sinónimos en los diversos lenguajes de modelado de datos son numerosos, como atributo concatenado, atributo compuesto y estructura de datos, 214*

**atributo derivado** *Un atributo cuyo valor se puede calcular a partir de otros atributos o derivar de los valores de esos otros atributos, 244-246*

**atributos de datos** *La unidad fundamental de datos que tiene significado para los usuarios y la empresa, 272; véase también atributos*

Ayuda en línea, 532-535

Agente de ayuda, 534

- B**
- backlog (Proyectos en cartera)** Repositorio de propuestas de proyectos que no se pueden financiar ni dotar de personal porque su prioridad es más baja que la de los proyectos aprobados para desarrollo de sistemas. Nótese que las prioridades cambian al paso del tiempo, por lo que un proyecto en cartera podría ser aprobado en una fecha futura, 57
- Barras de herramientas, 526
- balance** Concepto que requiere que los diagramas de flujo de datos con niveles diferentes de detalle reflejen consistencia y completud, 290
- base de datos** Conjunto de archivos interrelacionados, 418, 423; véanse también **almacén de datos**; **base de datos relacional**
- comparación con archivos convencionales, 418-419
  - datos distribuidos, 438-439
  - diseño; véase **diseño de bases de datos**
  - elaboración de prototipos, 439
  - empresarial, 107
  - grupo de trabajo, 423
  - operacional, 423
  - personal, 423
  - planeación de capacidad, 439
  - pros y contras, 420
- base de datos operacional** Una que brinda soporte a las operaciones y transacciones cotidianas en relación con un sistema de información. También llamada base de datos de transacciones, 423
- base de datos relacional** Una base de datos que implanta los datos como una sucesión de tablas bidimensionales que se relacionan mediante claves foráneas, 426; véase también **tabla**
- comandos SQL, 427
  - distribuida, 392-393
  - esquema, 426, 430-432
- Bases de datos grupos de trabajo, 423
- Bases de datos personales, 423
- BEA Systems, 385
- WebLogic, 34
- beneficio intangible** Un beneficio que se considera difícil o imposible de cuantificar, 319-320
- beneficio tangible** Un beneficio que puede ser fácilmente cuantificado, 318-319
- Beneficios
- intangibles, 319-320
  - tangibles, 318-319
- Berdie, Douglas R., 182
- Berstein, Phillip, 413
- Boar, Bernard, 370
- Boehm, Barry, 150-152
- Booch, Grady, 97
- Borland
- Delphi, 399
  - JBuilder, 89, 400, 516
- Botones, 498
- Botones de radio, 495
- Bovee, Courtland L., 339
- BPR; véase **rediseño de proceso de negocios**
- Brainstorming**; véase **lluvia de ideas**
- Brooks, Fred, 186, 209
- Bruce, Thomas A., 234n, 255, 448
- bugs**; véase **errores de programación**
- Business Objects Crystal Reports, 461
- Buy.Com, 390
- C**
- C#, 34, 77, 388
- C++, 34, 77
- Cajeros automáticos (Automated teller machines, ATM), 456, 487
- campo** La unidad más pequeña de datos significativos que se almacena en un archivo o base de datos, 421
- dominios, 432
  - en esquemas de bases de datos, 430-432
  - nombres, 430-432, 438
  - tamaños, 432
  - tipos de datos, 432
  - valores por omisión, 432
- campo descriptivo** Un campo sin clave, 421
- captura automática de datos** 487-489
- captura de datos** Identificación y adquisición de nuevos datos, 584
- cardinalidad** Los números mínimo y máximo de instancias de una entidad que pueden relacionarse con una sola instancia de otra entidad, 217
- Carnegie Mellon University, Software Engineering Institute, 49
- Carta de definición de proyecto, 109, 116
- Cartas de transmisión, 330
- CASE; véase **ingeniería de software asistida por computadora**
- Cash Jr., William B., 183
- Cashman, T., 481
- Casillas de combinación, 492
- Casillas de verificación, 495-496
- caso de uso** Escenario de negocios o evento respecto del cual el sistema debe proporcionar una respuesta definida. Los casos de uso evolucionaron a partir del análisis orientado a objetos; pero su utilización se ha vuelto común en muchos otros métodos de análisis y diseño de sistemas, 130, 188, 284
- actores, 189, 193
  - dependencias, 203-204
  - análisis de, 194-195, 284
  - entidades y, 284
  - extensión, 190
  - giro de eventos, 200
  - glosario, 196
  - jerarquía de, 202-203
  - nombres, 195-196, 199
  - requerimientos de negocios, 194-196
  - resumen, 191
  - tipos de, 199
  - uso durante el ciclo de vida, 188-189
- caso de uso de extensión** Uno que consiste en los pasos extraídos de otro más complejo para simplificar el caso original y, así, ampliar su funcionalidad. El caso de uso de extensión de hecho extiende la funcionalidad del caso de uso original, 190
- caso de uso de requerimientos del negocio** Un caso de uso creado durante el análisis de requerimientos para capturar la interacción entre un usuario y el sistema con independencia de los detalles de tecnología e implantación. También llamado caso de uso esencial, 194; véase también **caso de uso**
- identificación, 194-196
  - narrativa, 198-200
- caso de uso resumen** Un caso de uso que reduce la redundancia entre dos o más casos de uso al combinar los pasos comunes existentes en estos casos. Otro caso de uso utiliza o incluye el caso de uso resumen, 191
- CGI (Computer Gateway Interface), 400
- Chrissis, Mary Beth, 97
- Christerson, Magnus, 209
- ciclo de vida de sistema** El ciclo de vida de un sistema de información se divide en dos etapas: 1) desarrollo de sistemas y 2) operación y mantenimiento de sistemas; primero se construye y luego se usa y se mantiene. Tarde o temprano el ciclo culmina con el desarrollo de un nuevo sistema, 50-51
- Clase de objeto; véase **objeto**

**clave (llave)** *Atributo (o grupo de atributos) que asumen un valor único para cada instancia de entidad. A veces se lo denomina identificador, 215; véase también clave foránea*

alterna, 216  
 candidata, 216  
 concatenada, 215-216, 421  
 de tabla, 421  
 inteligente, 234-235  
 primaria, 216  
 secundaria, 216, 421, 430  
 selección, 234-235

**clave alterna** *Clave candidata que no se ha seleccionado para convertirla en clave primaria. Sinónimo de clave secundaria, 216; véase también clave secundaria*

**clave candidata** *Una de varias claves (llaves) que sirven como clave primaria de una entidad. También llamada identificador candidato, 216*

**clave (llave) concatenada** *Grupo de atributos que identifica de manera específica una instancia de una entidad. Entre sus sinónimos, se incluye clave compuesta, 215-216*

**clave (llave) foránea** *El campo que apunta a los registros en una tabla o base de datos distinta, 219, 421, 430*  
 integridad referencial, 436  
 nombres de funciones, 438

**clave inteligente** *Un código empresarial cuya estructura comunica datos acerca de una instancia de la entidad, 234-235*

**clave primaria** *Campo o grupos de campos que identifican a un registro de manera única, 216, 421, 430, 435*

**clave secundaria** *Campo que identifica un registro individual o un subconjunto de registros relacionados entre sí, 421, 430; véase también clave alterna*

**cliente ligero** *Una computadora personal que no tiene que ser muy potente, 385*

**cliente pesado (o gordo) (fat client)** *Una computadora personal, computadora tipo notebook o estación de trabajo que es típicamente poderosa, 384, 385, 388*

CMM; véase (CMM) **Modelo de Madurez de la Capacidad (Capability Maturity Model)**

Coad, P., 370  
 Códigos alfabéticos, 235  
 Códigos de barras, 396, 488  
 Códigos de bloque, 235  
 Códigos de posiciones significativas, 235  
 Códigos jerárquicos, 235  
 Códigos seriales, 234  
 Cold Fusion, 38, 89  
 Columbia House Record Club, 488

**comercio electrónico (e-commerce)** *Compra y venta de bienes y servicios por medio de Internet*

arquitecturas de aplicaciones, 389-390  
 carritos de compras, 507-508  
 diseño de entradas, 507-508  
 diseño de salidas, 472-475  
 empresa a cliente (B2C), 43  
 empresa a empresa (B2B), 43  
 problemas de seguridad, 532

**comité de dirección** *Un comité de gerentes ejecutivos de negocios y sistemas que estudia y jerarquiza propuestas de proyectos que compiten entre sí, con el fin de determinar cuáles generarán más valor para la organización, de las cuales, algunas se aprobarán para que continúe el desarrollo de sistemas. También llamado comité de dirección, 57, 115*

Componentes de comunicación, 35-38  
 Componentes de proceso, 31-32, 34-35  
 Componentes de sistemas de información, 27  
 comunicaciones, 35-38  
 conocimientos, 27, 30-31

proceso, 31-32, 34-35  
 redes y, 38-39

**compromiso ajustado** *Estrategia en la que se reevalúan continuamente la factibilidad y los riesgos a lo largo de un proyecto. El presupuesto y la fecha límite del proyecto se ajustan en concordancia, 55, 312-314*

Computadoras de bolsillo, 396, 487, 520

Computer Associates  
 Erwin, 88, 426, 429

Computer Ethics Institute, 15, 16

Conectividad abierta de bases de datos (open database connectivity, ODBC),  
 herramientas, 38, 398

Conexión, 531

Conjuntos de instrucciones, 529-231

Connor, Denis, 370

Componentes de conocimientos, 27, 30-31

**conservación de datos** *Práctica de garantizar que un flujo de datos contenga sólo los datos que necesita el proceso receptor, 271*

**constructor de sistema** *Especialista técnico que construye sistemas de información y sus componentes con base en las especificaciones de diseño que generan los diseñadores de sistemas, 10-11*

componentes de comunicaciones y, 38  
 componentes de conocimientos y, 31  
 componentes de proceso y, 34-35  
 perspectiva del sistema de información, 27

Consulta por ejemplo (QBE), 530

Consultores, 16

Control deslizante, 539

Copi, I. R., 308

CORBA, estándar para compartir objetos, 385

Corel Flow, 345

Correo electrónico (E-mail), 397, 458

**costo fijo** *Costo que ocurre a intervalos regulares y con una tasa relativamente fija o estable, 318*

**costo variable** *Costo que ocurre en proporción a algún factor de uso, 318*

Costo-beneficio, análisis

beneficios, 318-320

costos, 317-318

Costos, 317-318

fijos, 318

operativos, 318

de oportunidad, 321

de desarrollo de sistemas, 318

variables, 318

Costos de operación, 318

Costos de oportunidad, 321

CPI; véase **mejora continua de procesos**

CRC; véase **tarjetas de responsabilidad de colaboración de clases**

CRM; véase **administración de la relación con los clientes**

Crystal Reports, 461

Cuadro (caja) de combinación (combo box), 497-498

Cuadros (caja) de lista, 496-497

**comité ejecutivo (steering body)** *Un comité de gerentes ejecutivos de negocios y sistemas que estudia y jerarquiza propuestas de proyectos que compiten entre sí, con el fin de determinar cuáles proyectos generarán más valor para la organización y, de tal suerte, se deben aprobar para que continúe su desarrollo de sistemas. También llamado comité de dirección, 115-116*

**criterios de definición de subconjuntos** *Uno o más atributos cuyos valores finitos dividen las instancias de entidades en subconjuntos. A veces llamados entradas de inversión, 216*

**cuestionario** *Documento que permite al analista recopilar información y opiniones de los encuestados, 162-164*



**cuestionario de formato fijo** *Uno con preguntas que requieren elegir una respuesta entre las disponibles, definidas previamente, 163-164*

**cuestionario de formato libre** *Uno diseñado para ofrecer a quien lo responde una mayor libertad en sus respuestas. Se hace una pregunta y la persona registra su respuesta en el espacio previsto después de la pregunta, 163*  
Curtis, Bill, 97

## D

**Data mining** (minería de datos), 423

DataWatch Monarch/ES, 460

**data warehouse** *Base de datos en la que se guardan datos extraídos de bases de datos operacionales, 423*

matriz de datos a proceso de CRUD, 301-302

**datos distribuidos** *Sistema de cliente/servidor en el cual las capas de datos y manipulación de datos se colocan en los servidores, y otras capas se colocan en los clientes. También llamado cómputo de cliente/servidor de dos capas, 387-388, 404-405*

opciones, 438-439

replicación, 438-439

**datos y aplicaciones distribuidas** *Sistema de cliente/servidor en el que las capas de datos y manipulación se colocan en sus propios servidores, la lógica de la aplicación se coloca en su propio servidor, y la lógica de presentación y la presentación se colocan en los clientes. También llamado cómputo de cliente/servidor de tres capas o n capas, 399-400*

Davis, William S., 182

DBA; véase **administrador de bases de datos**

DBMS; véase **sistema de administración de bases de datos**

DDL; véase **lenguaje de definición de datos**

De Marco, Tom, 96, 308, 375

**declaración de trabajo** *Contrato de los administradores y la comunidad de usuarios para desarrollar o mejorar un sistema de información; define visión, alcance, restricciones, requerimientos de usuarios de alto nivel, plan y presupuesto. Sinónimo de carta de definición de proyecto, plan de proyecto y convenio de nivel de servicio, 62*

**definición de problemas** *Declaración y categorización de problemas, oportunidades y directrices; también suele incluir las restricciones y una decisión inicial de la solución. Sus sinónimos abarcan estudio preliminar y evaluación de factibilidad, 62*

Dejoie, Roy, 182

depende de, véase **dependencia**

**dependencia** *Relación entre casos de uso que indica que un caso de uso no puede realizarse hasta que se haya realizado el otro caso de uso, 191-192*

**dependencia transitiva** *Cuando el valor de un atributo sin clave depende del valor de otro similar, que no sea por derivación, 246-248*

depósito, véase **repositorio**

**desarrollo basado en modelos** *Estrategia de desarrollo de sistemas que hace énfasis en trazar modelos de sistemas como ayuda para visualizar y analizar los problemas, definir los requerimientos de negocios y diseñar los sistemas de información, 66, 72, 44-76*

Desarrollo conjunto de aplicaciones (joint application development, JAD), 349-351

**desarrollo (o diseño) centrado en el usuario** *Proceso de desarrollo de sistemas basado en la comprensión de las necesidades de los interesados y las razones por las que deben desarrollarse los sistemas, 187*

**desarrollo rápido de aplicaciones (RAD)** *Estrategia de diseño de sistemas que utiliza técnicas estructuradas, de elaboración de prototipos y JAD para desarrollar sistemas rápidamente, 349*

futuro de, 141

método FAST, 78-80

modelado lógico, 64

técnicas de análisis, 106

timeboxing (tiempo fijo), 80, 130

ventajas y desventajas, 80

**desarrollo rápido de aplicaciones (rapid application development, RAD)** *Estrategia de desarrollo de sistemas que hace énfasis en la rapidez de desarrollo mediante la participación amplia del usuario en la construcción rápida, iterativa e incremental de una sucesión de prototipos funcionales de un sistema, que tarde o temprano se convierten en el sistema definitivo (o en una versión de él), 78*

**descomposición** *Acto de separar un sistema en sus componentes, 264-266*

**descomposición funcional** *Acto de dividir un sistema en subcomponentes, 188*

**desplazamiento en pantalla** *Despliegue de información en los extremos superior o inferior de la pantalla, línea por línea, 521*

DFD; véase **diagrama de flujo de datos**

Diagrama de contexto, 74, 120-122, 194

**diagrama de caso de uso** *Diagrama que muestra las interacciones entre el sistema, otros sistemas y usuarios. En otras palabras, describe gráficamente quién usará el sistema y de qué manera el usuario espera interactuar con el sistema, 188, 196-198*

Diagrama de colaboración; véase **diagrama de comunicación**

**diagrama de dependencia de casos de uso** *Representación gráfica de las dependencias entre casos de uso, 203-204*

**diagrama de descomposición** *Herramienta que se usa para ilustrar la descomposición de un sistema. También llamado gráfica de jerarquía, 265-266*

evento, 284, 287

funcional, 277, 281-282

propósito, 277

**diagrama de descomposición funcional** *Diagrama que divide el sistema en funciones o subsistemas lógicos. 277, 281-282*

Diagrama de descomposición de eventos, 284, 287

**diagrama de evento** *Un diagrama de flujo de datos para manejar un solo evento, así como los agentes y almacenes de datos que proporcionan entradas o reciben salidas, 277, 287-299*

**diagrama elemental** *Diagrama de flujo de datos que muestra los procesos elementales, almacenes de datos y flujos de datos de un evento individual, 277, 291*

**diagrama de flujo de datos de contexto** *Un diagrama que muestra el sistema como una "caja negra" y sus interfaces principales con su ambiente, 277 280-281*

**diagrama de flujo de datos (data flow diagram, DFD)** *Modelo de procesos usado para mostrar el flujo de datos por un sistema y la tarea o procesamiento que realiza el sistema. Sus sinónimos incluyen los gráficos de burbuja, gráficos de transformación y modelo de procesos, 259; véase también **modelado de procesos***

equilibrio, 290-291

agentes externos, 259, 261-262

de contexto, 277, 280-281

diagramas de eventos, 277

diferencias con los flujogramas, 259-261

errores mecánicos, 267

físicos; véase **diagrama de flujo físico de datos**

flujos de datos, 259, 267-270

lógicos, 375

primitivos, 277, 291

procesos, 259, 266-267

símbolos, 259, 264

tiendas de datos, 259, 262-263

uso de, 77, 104, 105

uso en el rediseño de procesos de negocios, 261, 276-277

**diagrama de flujo de datos físico (o de flujo físico de datos)** *Modelo de procesos usado para comunicar las caracterís-*

*ticas técnicas de implantación de un sistema de información, 353, 375; véase también diagrama de flujo de datos*

- agentes externos, 379
- almacenes de datos, 379
- arquitecturas de redes, 403-404
- dibujo, 402
- diseño de entradas con, 501-502
- diseño de salidas con, 464
- flujos de datos, 379
- límites persona/máquina, 408
- modelado de arquitecturas de aplicaciones, 401-402
- prerrequisitos, 402
- procesos, 375-378
- restricciones, 402-403
- unidades de diseño, 402, 405-406, 464
- uso de, 375, 380

**diagrama de Ishikawa** *Herramienta gráfica usada para identificar, explorar e ilustrar problemas, así como las causas y efectos de esos problemas. Es frecuente que se le llame diagrama de causas y efectos o diagrama de espina de pescado (porque se asemeja al esqueleto de un pez), 153-154*

**diagrama de sistema** *Diagrama de flujo de datos que fusiona los diagramas de eventos de todo un sistema o parte de él, 277, 289-290*

**diagrama de transición de estados (state transition diagram, STD)** *Herramienta que se usa para ilustrar la secuencia y variación de pantallas que pueden ocurrir durante una sesión de usuario, 537-538*

**diagrama de entidad-relación (entity relations diagram, ERD)** *Modelo de datos en el que se utilizan diversas notaciones para mostrar datos con base en las entidades y las relaciones que se describen con esos datos, 105, 212-213; véase también modelado de datos*

Diagrama físico de datos, 379

**diálogo** *El flujo total de las pantallas y los mensajes para una aplicación, 519*

**diálogo del usuario** *Especificación de la forma en cómo el usuario se mueve de una ventana o página a otra e interactúa con los programas de aplicaciones para realizar tareas útiles, 37; véase también Menús*

Diálogos de pregunta-respuesta, 531

Diccionario de datos; véase **repositorio**

Dígitos de comprobación, 492

Dígitos de autoverificación, 492

**directriz** *Un nuevo requerimiento que imponen los administradores, organismos gubernamentales o alguna otra influencia externa, 57*

Diseño de base de datos, 355; véase también **esquema de bases de datos**

- directrices, 429
- generación de código SQL, 429, 439, 442
- herramientas automatizadas, 426, 429, 432, 435
- integridad de los datos, 435-436
- integridad referencial, 436
- objetivos, 430
- prerrequisitos, 430

**diseñador de sistemas** *Especialista técnico que traduce los requerimientos de negocios de los usuarios del sistema y las restricciones en soluciones técnicas. Diseña las bases de datos, entradas, salidas, pantallas, redes y software que podrán satisfacer los requerimientos de los usuarios del sistema, 10*

- componentes de comunicaciones y, 37-38
- componentes de conocimientos y, 31
- componentes de procesos y, 34
- especialidades, 10
- perspectiva del sistema de información, 27

Diseñadores; véase **diseñador de sistemas**

**diseño basado en modelos** *Estrategia de diseño de sistemas que hace énfasis en dibujar modelos de sistemas para documentar los aspectos técnicos y de implantación del sistema, 345*

Diseño de entradas, 355-356; véanse también **Interfaces gráficas de usuario (GUI)**;

- correo electrónico, 397
- de lotes remoto, 395-396, 486
- imágenes, 397
- intercambio de documentos, 397
- intercambio electrónico de datos, 397
- introducción con lápiz óptico, 396
- introducción de datos sin el teclado, 396
- middleware, 397-398
- procesamiento en línea, 394-395, 486
- procesamiento por lotes, 394, 486
- tecnología de grupos de trabajo, 397

controles, 491-492

elaboración de prototipos, 484, 500, 503-505

entradas basadas en web, 507-508

herramientas automatizadas, 500, 506-507

métodos de implantación

- biométrica, 489

- captura de datos automática, 487-489

- pantalla sensible al tacto, 487

- ratón, 487

- reconocimiento de marcas ópticas, 487-488

- sonido y voz, 487

- tarjetas inteligentes, 489

- teclado, 487

- terminales de punto de venta, 487

- tinta magnética, 488

- transmisión electromagnética, 489

problemas de los usuarios, 489-491

proceso, 501

- diseño de documentos fuente, 505-507

- elección de controles de GUI, 502

- realimentación del usuario, 503-505

- requerimientos lógicos, 501-502

taxonomía de entradas, 484

tipos de entradas, 484-486

Diseño de salidas, 459; véase también **diseño de interfaces del sistema**

arquitecturas

- correo electrónico, 397

- imágenes, 397

- intercambio de documentos, 397

- intercambio electrónico de datos, 397

- middleware, 397-398

- procesamiento en línea, 394-395

- procesamiento por lotes, 394

- tecnología de grupo de trabajo, 397

elaboración de prototipos, 452, 460-461, 467, 471

formas preimpresas, 394, 456, 466

guías, 461-464

herramientas automatizadas, 460-464

proceso

- diseño, 467, 469

- diseño de formas preimpresas, 466

- elaboración de prototipos, 467, 471

- realimentación de usuarios, 471-472

- requerimientos de salidas físicas, 465-466

- requerimientos lógicos, 464-465

Diseño de salidas

- diseños, 467, 471

- prototipos, 467, 471

**diseño de sistemas** *La especificación de una solución detallada, basada en la computadora, 344*

elaboración de prototipos, 346-348  
 estrategia de desarrollo rápido de aplicaciones, 349-351  
 estrategias, 344  
 estrategias basadas en modelos, 345  
   diseño estructurado moderno, 345  
   diseño orientado a objetos, 348-349  
   herramientas automatizadas, 345  
   ingeniería de información, 105, 345-346  
 métodos FAST, 66-67, 351  
 modelado de procesos, 279  
 objetivo, 351  
 tareas de desarrollo internas  
   actualización del plan del proyecto, 358  
   arquitectura de aplicaciones, 351, 353-355  
   diseño de base de datos, 355  
   diseño de la interfaz de usuario, 355-356  
   paquete de especificaciones de diseño, 357  
 tareas para la integración de software comercial, 358-360  
   afirmaciones del proveedor y validación de funcionamiento, 363  
   criterios y opciones técnicas, 360-361  
   evaluación y determinación de jerarquía de propuestas, 363-364  
   fase de abasto, 358, 360-363  
   fase de análisis de decisiones, 358, 360  
   juntas de información con el proveedor, 363  
   negociaciones del contrato, 364  
   solicitud de propuesta, 361

**diseño de sistemas** *Técnica complementaria (de la de análisis de sistemas) de solución de problemas que reensambla los componentes de un sistema en el sistema completo, con la esperanza de mejorarlo. Ello puede abarcar la adición, eliminación y cambio de componentes en relación con el sistema original, 102; véase también sistema, diseño*

**diseño del sistema** *Especificación o construcción de una solución técnica, basada en computadora, a los requerimientos de negocios identificados en un análisis de sistemas. (Nota: el diseño asume cada vez más la forma de un prototipo.), 16*

**diseño estructurado moderno** *Técnica de diseño de sistemas que descompone los procesos del sistema en componentes manejables, 345*

**diseño físico** *La traducción de los requerimientos de un usuario de negocios en un modelo de sistemas que muestra una ejecución técnica de tales requerimientos. Sus sinónimos comunes abarcan diseño técnico o, para describir las salidas, modelo de ejecución. El antónimo del diseño físico es el diseño lógico (descrito con antelación en el mismo capítulo), 66-67; véase también sistemas, diseño*

**diseño lógico** *La traducción de los requerimientos de usuarios de negocios en un modelo de sistemas que muestre sólo los requerimientos del negocio y no cualquier posible diseño o ejecución técnica de esos requerimientos. Sus sinónimos frecuentes abarcan diseño conceptual y diseño esencial, de los cuales el segundo se refiere a modelar la "esencia" de un sistema o sus "requerimientos esenciales" con independencia de cualquier tecnología. El antónimo de diseño lógico es diseño físico, que se define más adelante en el mismo capítulo, 64*

Diseño lógico, 64, 131  
   casos de pruebas de aceptación, 134  
   elaboración de prototipos, 134  
   estructuración de requerimientos funcionales, 134  
   validación de requerimientos funcionales, 134

Diseño técnico; véase **diseño físico**

distribución de datos; véase **datos distribuidos**

**distribución en capas limpias (clean layering)** *Estrategia de diseño que requiere la separación física de las capas de presentación, aplicación y datos, 39, 399*

DML; véase **datos, lenguaje de manipulación**

**documentación** *Actividad continua de registrar hechos y especificaciones de un sistema para referencia actual y futura, 69*  
   durante el desarrollo, 53  
   en el desarrollo ágil, 133  
   existente, 157-159

**documento fuente** *Un formato usado para registrar datos acerca de una transacción, 486*  
   diseño, 489-491, 505-507  
   elaboración de prototipos, 506-507

**documentos de definición requerimientos** *Documento formal que comunica los requerimientos de un sistema propuesto a los interesados clave y sirve como un contrato del proyecto de sistemas. Sinónimo de declaración de requerimientos, especificación de requerimientos y especificación funcional, 155-156*  
 Documentos, intercambio, 397

**dominio** *Propiedad de un atributo que define los valores que puede recibir correctamente el atributo, 214*

**dominio** *Valores legítimos para un atributo, 275*  
   comprobaciones de validación, 492  
   de campos de bases de datos, 432  
   definición, 239

Dos capas, cómputo de cliente-servidor de; véase **datos distribuidos**

DSS; véase **sistemas de soporte a las decisiones**

Dunlap, Duane, 513  
 Dunne, Peter, 183  
 Dynasty, 400

## E

Eastman, David, 547  
 eBay, 390  
 E-business; véase **negocios electrónicos**

E-commerce; véase **comercio electrónico**

ED; véase **duración esperada**

EDI; véase **intercambio electrónico de datos**

EDS; véase **Electronic Data Systems**

Edwards, Jeri, 413  
 e.Reporting suite de Actuate, 461

EAI; véase **integración de aplicaciones empresariales**

**efectividad de costos** *Resultado obtenido al lograr el equilibrio entre los costos de desarrollo, mantenimiento y operación de un sistema de información y los beneficios derivados de éste durante su ciclo de vida. La actividad de costos se mide con el análisis de costo-beneficio, 55, 320*

EIS; véase **sistema de información ejecutiva**

**elaboración de prototipos** *Técnica de construcción rápida de un modelo funcional a la vez que incompleto de un sistema de información con las herramientas de desarrollo rápido de aplicaciones, 35; véase también elaboración de prototipos de identificación*  
   diseño de entradas, 484, 500, 503-505  
   diseño de la interfaz de usuario, 516, 534, 538-541  
   diseño de salidas, 452, 460-461, 467, 471  
   herramientas automatizadas, 460-461  
   uso en el análisis de sistemas, 105-107, 134, 170-171  
   uso en el diseño de sistemas, 346-348  
   ventajas y desventajas, 347-348

**elaboración de prototipos de identificación** *Técnica usada para identificar los requerimientos de negocios de los usuarios al hacerlos reaccionar a una implantación rápida y no acabada de esos requerimientos, 106-107, 134*

**elaboración de prototipos de identificación (discovery prototyping)** *El acto de construir un modelo funcional o representativo a pequeña escala de los requerimientos de los usuarios para identificar o verificar esos requerimientos, 170-171*

Elementos primarios (de informes), 329  
 Elementos secundarios (de informes), 329, 330  
 Empresa virtual, 390

**entidad** *Una clase de personas, lugares, objetos, eventos o conceptos respecto de los cuales es necesario capturar y guardar datos, 213-214*

almacenes de datos y, 262, 291  
 asociativa, 218-219, 222  
 atributos; véase **atributo**  
 casos de uso y, 284  
 clave, 215-116  
 débiles, 221  
 definiciones, 231  
 identificación de, 231-232  
 empresa, 30  
 fuerte (independiente), 220  
 generalización, 225, 237  
 hija, 219  
 historia del ciclo de vida de, 284  
 madre, 219  
 nombres, 231  
 supertipos y subtipos, 225, 237, 432

**entidad asociativa** *Una entidad que hereda su clave (llave) primaria de más de una entidad, 218-219, 222*

**entidad hija** *Una entidad de datos que deriva uno o más atributos de otra, llamada madre. En una relación de una a muchas, la hija es la entidad que está en el lado de "muchas", 219*

**entidad madre** *Una entidad de datos que contribuye con uno o más atributos a otra entidad, llamada hijo. En una relación de uno a muchos, la entidad madre está en el lado de "una", 219*

**entrada de datos** *Proceso de trasladar datos en un formato entendible para la computadora, 486*

entradas basadas en web, 507-508

**entrevista** *Técnica de identificación de datos en la que el analista de sistemas recopila información de individuos mediante la interacción cara a cara, 164*

conducción, 168  
 elección de los entrevistadores, 166  
 escucha en, 168-169  
 estructurada, 165  
 guía, 166  
 lenguaje corporal y proxémica, 169-170  
 no estructurada, 165  
 preguntas, 165-166  
 preparación, 166  
 seguimiento, 168  
 ventajas y desventajas, 165

**entrevista estructurada** *Una en la que el entrevistador tiene un conjunto de preguntas específicas que hace al entrevistado, 165*

**entrevista no estructurada** *Una que se realiza únicamente con un objetivo o tema en mente y con pocas preguntas específicas, si acaso. La entrevista se basa en que el entrevistado proporcione un marco de referencia y dirija la conversación, 165*

**equilibrio** *Concepto que requiere que los diagramas de flujo de datos en distintos niveles de detalle reflejen consistencia e integridad, 290-291*

Equipos; véase **Proyectos, equipos**

Equipos de proyecto, 14

ERD; véase **diagrama de relaciones de entidades**

Ernest, Kallman, 21

ERP; véase **software para la planeación de recursos de la empresa**

Errores de programación; 68; véase **también causa de mantenimiento de programas**

Escritos; véase **Informes escritos**

Escucha, 168-169

London, Keith, 182

ESP; véase **proveedor de servicios externo**

Especialización; véase **generalización/especialización**

Especificación funcional; véase **documento de definición de requerimientos**

**especificaciones de interfaz** *Diseños técnicos en los que se documenta la manera en que los usuarios del sistema interactuarán con él y la forma en que el sistema mismo lo hará con otros sistemas, 37*

**especificaciones de software** *Diseño técnico de los procesos de negocios para automatizarlos o brindarles soporte mediante programas de computadora que escriben los constructores de sistemas, 34*

**esquema de aplicación** véase **programa de aplicación, 279**

**esquema de bases de datos** *Modelo que representa la implantación técnica de una base de datos, 430*

creación de, 355, 430-432  
 generación de código SQL, 429, 439, 442  
 generados con System Architect, 432, 435  
 relacionales, 426

Esquemas; véase **esquemas de bases de datos**

Eventos de condición, 283

**estimación** *Predicción calculada de los costos y esfuerzos necesarios para el desarrollo de un sistema. Un sinónimo hasta cierto punto menos pretencioso es conjetura informada, que usualmente significa que la estimación se basa en la experiencia como datos empíricos, si bien carece de rigor; en otras palabras, una simple conjetura, 69*

extractor de pantalla (screen scapers), 386

**estrategia de desarrollo en cascada** *Una forma de análisis y diseño de sistemas que completa cada fase una después de otra y solamente una vez, 69, 72*

**estrategia de desarrollo iterativo** *Una de análisis y diseño de sistemas que completa el sistema de información en iteraciones sucesivas. Cada iteración realiza una cierta parte del análisis, del diseño y de la construcción. Sus sinónimos incluyen estrategia de desarrollo incremental y espiral, 69-72*

**estrategia orientada a objetos** *Técnica basada en modelos que integra los datos y procesos en conceptos llamados objetos. Los modelos de objetos son imágenes que ilustran los objetos del sistema desde diversas perspectivas, como la estructura, comportamiento e interacciones entre ellos, 105*

**estructura de datos** *Disposición específica de los atributos de datos que definen una instancia individual de un flujo de datos, 272*

descripción, 391  
 flujos de datos, 268, 272-273  
 flujos de datos de entrada, 501-502  
 flujos de datos de salida, 464  
 importancia, 273  
 notación, 272-273  
 tipos, 272

Estructura, gráficos, 77, 345, 375

Estudio preliminar; véase **definición de problema**

Ética, 15

E-trade.com, 390

**evento** *Unidad lógica de trabajo que debe completarse como un todo. A veces llamado transacción, 266*

de estados, 283  
 descripciones de procesos, 288, 289  
 diagramas de flujo de datos físicos, 405-406  
 empresarial, 32, 266  
 externo, 283  
 temporal, 190, 283

**evento temporal** *Uno del sistema activado por el tiempo, 190, 283*

Eventos de negocio; véase **evento**

Eventos externos, 283

## F

**factibilidad** *Medición del grado en que un sistema de información será beneficioso o práctico para una organización, 312*

cultural (política), 315-316  
 económica, 317, 320, 401  
 legal, 317  
 operativa, 315, 401

- programa, 316  
pruebas, 315-317  
técnica, 316, 401
- factibilidad** *Medición de cuán benéfico sería el desarrollo de un sistema de información para una organización, 69*
- factibilidad cultural (o política)** *Medición del grado en que la solución será aceptada en un ambiente organizacional dado, 315-316*
- factibilidad de calendario** *Medición del grado en que es razonable el cronograma de un proyecto, 316*
- factibilidad económica** *Medición de la relación costo-beneficio de un proyecto o solución, 317, 401*  
análisis de recuperación de la inversión, 321-323  
rendimiento sobre la inversión, 323  
técnicas, 320  
valor presente neto, 323-324  
valor del dinero en el tiempo, 320-321
- factibilidad legal** *Medición del grado en que una solución se puede implantar dentro del marco de las obligaciones legales y contractuales existentes, 317*
- factibilidad operativa** *Medición del grado en que una solución satisface los requerimientos del sistema identificados para resolver problemas y aprovechar las oportunidades que se visualizan para el sistema, 315, 401*
- factibilidad técnica** *Medición del grado en que una solución técnica es práctica y de la disponibilidad de recursos técnicos y experiencia, 316, 401*
- Factores ergonómicos en el diseño de la interfaz de usuario, 516, 517-518
- Fase de abastecimiento, 358
- Fase de análisis de decisiones (o alternativas), 134-136  
actualización del plan del proyecto, 139  
adquisición de software comercial, 358, 360  
análisis de factibilidad, 137-139, 314-315, 324  
análisis de soluciones candidatas, 137-139  
comparación de soluciones candidatas, 139, 324-327  
identificación de soluciones candidatas, 136-137, 325  
método FAST, 65-66  
propuesta de sistema, 139-141
- Fase de análisis de problemas, 117-118
- Fase de análisis de requerimientos, 127  
actualización del plan de proyecto, 130  
administración continua de requerimientos, 131  
comunicación de la declaración de requerimientos, 131  
identificación de requerimientos, 127-130  
jerarquía de requerimientos, 130  
método FAST, 63-64  
modelado de datos, 227-228
- Fase de construcción; véase **construcción de sistemas**
- Fase de definición del alcance  
análisis de factibilidad, 314  
comunicación del plan del proyecto, 115-116  
cronograma y presupuesto basales, 115  
método FAST, 59-62, 109  
negociación de alcance, 114  
problema base e identificación de oportunidades, 111-114  
valía del proyecto base, 115
- factor de bloque** *Número de registros lógicos incluidos en una operación individual de lectura o escritura, 422*
- FAST** *Método teórico usado en este libro para demostrar un proceso representativo de desarrollo de sistemas. Las letras de la sigla corresponden a Framework for the Application of Systems Thinking ("marco de referencia para la aplicación del pensamiento de sistemas"; además, "fast" en inglés significa "rápido"), 51-52*  
como método ágil, 72  
desarrollo impulsado por modelos, 72, 74-76  
estrategia de desarrollo rápido de aplicaciones, 78-80
- estrategia de implantación del paquete comercial, 81-83  
estrategias de diseño de sistemas, 351  
estrategias híbridas, 84  
fases, 52-53, 57-59  
análisis de decisiones, 65-66  
análisis de problemas, 62-63  
análisis de requerimientos, 63-64  
análisis de sistemas, 102, 108, -109  
construcción y pruebas, 67  
definición de alcance, 59-62, 109  
diseño lógico, 64  
instalación y entrega, 67-68  
integración y diseño físico, 66-67  
mantenimiento del sistema, 84-87
- Federal Express, 396, 486
- Finkelstein, Clive, 255
- Firefox, 486, 520
- Fitzgerald, Ardra F., 182
- Fitzgerald, Jerry, 182, 513
- flat-file** *Archivos planos, 31*
- flujo de control** *Una condición o un evento sin datos que desencadena un proceso, 269-270*
- flujo de datos** *Datos que entran o salen de un proceso, 267*  
atributos, 272  
compuesto, 268-269  
conservación, 271  
convergentes, 275-276  
divergentes, 275-276  
en diagramas de flujo de datos, 259, 267-270  
estructuras de datos, 268, 272-273  
descripción, 291  
notación, 272-273  
tipos, 272  
físicos, 379  
flujos de control, 269-270  
introducción, 501-502  
lógico, 379  
nombres, 270, 379  
salida, 464
- flujo de datos compuesto** *Un flujo de datos que consiste de otros flujos de datos, 268-269*
- flujo de datos convergente** *La fusión de varios flujos de datos en uno solo, 275-276*
- flujo de datos divergente** *Uno que se divide en flujos de datos múltiples, 275-276*
- flujo de trabajo** *El flujo de transacciones y procesos de negocios para garantizar que se realicen las comprobaciones y aprobaciones debidas, 34*
- Flujogramas, 77, 259-261
- Formato administrativo, 330
- Formato de hechos, 329-330  
verificaciones de formato, 492
- Fowler, George, 182
- "Framework for Information Systems Architecture" (Zachman), 38-39
- "Framework for the Application of Systems Thinking"; véase **FAST**
- Freund, John F., 182
- Fuente; véase **documento fuente**
- función** *Conjunto de actividades relacionadas y continuas de una empresa, 266; véase también función de empresa*
- función de negocio** *Grupo de procesos relacionados que soporta al negocio. Las funciones se pueden descomponer en otras subfunciones y así sucesivamente hasta llegar a procesos que realizan tareas específicas, 31-32*
- G**
- Galitz, Wilbert O., 44, 481, 513, 517, 547
- Gane, Chris, 96, 182, 370, 375, 413

Gartner Group, 48

Gause, Donald C., 21, 147, 150n, 182

**generalización** Concepto según el cual los atributos comunes a varios tipos de una entidad se agrupan en su propia entidad, **225, 237**

**gerente de proyecto** Profesional experimentado que acepta la responsabilidad de planear, vigilar y controlar proyectos en cuanto al plan, presupuesto, productos, satisfacción del cliente, normas técnicas y calidad del sistema, **16**

Gildersleeve, Thomas R., 55n, 96, 168n, 182, 308, 339

Goldman, James, 44, 414

Gore, Marvin, 339

**grado** Número de entidades que participan en una relación, **217**

Gráficos, 458

Gregory, William, 255

Grillo, John, 21

Groupware, tecnología, 385

Guengerich, Steve, 414

GUI; véase **Interfaces gráficas de usuario**

## H

Harkey, Dan, 414

Harmon, Paul, 308

Hartson, H. Rex, 547

Hay, David C., 255

**herencia** En los casos de uso, relación entre actores creada para simplificar el dibujo cuando un actor resumen hereda las funciones de múltiples actores, **192**

ambiente de desarrollo de aplicaciones; véase **ambiente de desarrollo de aplicaciones**

CASE; véase **ingeniería de software asistida por computadora**

de administración del proyecto, 91

de aplicaciones de gerente de proceso, 91

de autoría de la ayuda, 533

de elaboración de prototipos, 460-461

de repositorios; véase **repositorio**

de soporte metodológico, 87-88

diseño de interfaz de usuario, 516, 536

para diseño de entradas, 500, 506-507

para diseño de salidas, 460-461

para la planeación conjunta de requerimientos, 173

Hipertexto, 528-529

Hipervínculos, 460, 474, 528-529

Hix, Deborah, 547

Hoffer, Jeffrey, 448

Hojas de cálculo, 460, 506

Horton, William K., 447

HTML (Hypertext Markup Language, lenguaje de marcación de hipertexto), 385, 392, 400, 533

Hunter, Richard, 48

## I

IBM, 361, 425

CICS, 385

DB2 Universal Database, 54, 385, 393, 424, 428

Lotus Notes, 385, 397

MQ Messaging, 54

Rational, 345

VisualAge, 400, 492, 516

Websphere, 34, 54, 89, 106, 385, 400

Iconos, 526-527

**identificación de hechos** Proceso de recopilar información acerca de problemas, oportunidades, requerimientos de solución y prioridades del sistema. También denominado recopilación de información, **107**

**identificación de hechos** Proceso formal de usar la investigación, juntas, entrevistas, cuestionarios, muestreo y otras técnicas para recopilar información acerca de los problemas, requerimientos y preferencias concernientes al sistema. También se llama recopilación de información o recolección de datos, **68, 154**

estrategia, 176-177

para el modelado de procesos, 279

técnicas, 107-108, 157; véase también **planeación de conjunto de requerimientos (joint requirements planning, JRP)**

cuestionarios, 162-164

elaboración de prototipos de identificación, 170-171

entrevistas, 164-170

investigación y visitas al sitio, 159-160

muestreo de la documentación existente, 157-159

observación del ambiente de trabajo, 160-162

**identificación de hechos** Proceso formal de usar la investigación, entrevistas, juntas, cuestionarios, muestreo y otras técnicas para recopilar información acerca de problemas, requerimientos y preferencias concernientes al sistema. También se llama recopilación de información o de datos, **68-69**

**identificación de requerimientos** Proceso y técnicas que usan los analistas de sistemas para identificar o extraer problemas de los sistemas y requerimientos de solución de la comunidad de usuarios, **107, 150-152**; véanse también **identificación de hechos**; **planeación conjunta de requerimientos (JRP)**; **modelado de casos de uso**

métodos, 107-108

proceso

administración de requerimientos, 156-157

análisis de requerimientos, 154-155

identificación y análisis de problemas, 152-154

documentación, 154, 155-156

identificación de hechos, 154

Identificación y análisis de problemas, 152-154

IE; véase **ingeniería de información**

Implantar sistemas de información, 351, 353-355, 374

distribución de datos, 404-405

distribución de proceso, 405

estrategia de empresa, 400-401

modelado de, 401-402

táctica, 401

Independencia de plataforma, 520

**ingeniería de información** Técnica operada por modelos y centrada en DATOS, pero sensible a PROCESOS, para la planeación, el análisis y el diseño de sistemas de información. Los modelos de IE son imágenes que sincronizan los datos y procesos del sistema, **105**

Imágenes, 397

Immon W. H., 44

alcance, 114

ampliación del alcance, 62

ímpetu para, 57

lanzamiento, 116

patrocinadores, 116

planeación, 115

**instancia de entidad** Una sola instancia de una entidad, **214**

**implantación del sistema** Construcción, instalación, pruebas y entrega de un sistema a producción (es decir, a las operaciones cotidianas), **17**

Implantación, fase de; véase **implantación del sistema**

Implantación, modelos; véanse **diseño físico**; **modelo físico**

Impresora, gráficos de espaciado, 460

Informe de conclusiones, 330

**informe de excepciones** Una salida interna que filtra datos para presentar información en la que se notifica de excepciones a alguna condición o estándar, **455**

**informe detallado** Salida interna en la que se presenta información con filtrado mínimo, si acaso; véase **reporte detallado**, **452**

Informe ejecutivo, 330

**Informes escritos**

- cartas de transmisión, 330
- formato administrativo, 330
- formato de hechos, 329-330
- longitud, 329
- necesidad para, 329
- organización de, 329-330
- redacción, 330-331

**Informes; véase también Salidas y reportes**

- de excepción, 455
- de resumen, 455
- detallados, 452
- formatos, 467, 471
- herramientas de diseño, 461
- prototipos, 467, 471

**ingeniería de información (information engineering, IE)** *Una técnica conducida por modelos y centrada en datos, pero sensible a procesos, para la planeación, el análisis y el diseño de sistemas de información. Los modelos de IE son imágenes que ilustran y sincronizan los datos y procesos del sistema, 105*

- métodos de diseño, 345-346
- modelos, 64, 105, 213

**ingeniería de software asistida por computadora (computer-assisted software engineering, CASE)** *Uso de herramientas de software automatizadas que brindan soporte al dibujo y análisis de modelos de sistemas y a las especificaciones correspondientes. Algunas herramientas de CASE también posibilitan la elaboración de prototipos y generación de código, 88; véase también System Architect 2001 (Popkin)*

- depósitos, 88, 91
- diseño de base de datos, 426, 429, 432, 435
- diseño de entradas, 500
- diseño de interfaz de usuario, 386
- diseño de salidas, 461
- funciones, 88
- futuro de, 141
- generación de estructura de base de datos, 439, 442
- ingeniería hacia adelante, 88-89
- ingeniería inversa, 88-89, 107
- modelado de datos, 228-230, 248
- modelado de procesos, 279
- modelos de diseño de sistemas, 345
- uso en análisis de sistemas, 104

lineamientos de ingeniería humana (ergonomía), 518-519

**ingeniería inversa** *Uso de tecnología que lee el código de programas a partir de bases de datos, programas de aplicación o interfaces de usuarios existentes y genera automáticamente el modelo de sistemas equivalente, 88, 107*

**ingeniería hacia adelante (forward engineering)** *Una capacidad de herramienta CASE que permite generar código inicial del software o base de datos directamente a partir de modelos de sistemas, 88*

**inglés estructurado** *Sintaxis de lenguaje para especificar la lógica de un proceso, 295-300*

**inicio del sistema** *Planeación inicial de un proyecto para definir el alcance de negocios, los objetivos, el plan y el presupuesto iniciales, 16*

Inprise Jbuilder, 492

Instalación, 67-68

Integración; véase **integración de sistemas**

Integridad, 435

Integridad de dominio, 436

**integridad referencial** *La seguridad de que un valor de clave foránea en una tabla tiene un valor de clave primaria correspondiente en la tabla relacionada, 436*

**intercambio electrónico de datos (electronic data interchange, EDI)** *Flujo electrónico normalizado (estandarizado) de transacciones de negocios o datos entre empresas, 397*

**interesado** *Toda persona que tiene interés en un sistema de información existente o propuesto. Los interesados y grupos de interés pueden ser trabajadores técnicos y no técnicos. También puede tratarse de trabajadores de la misma compañía o de otras, 7, 26-27*

Interfases de usuario híbridas Windows/Web, 529

Interfaz gráfica de usuario (graphical user interfaces, GUI); véase **también Interfaces de usuario, diseño**

- controles, 492-494, 522
  - avanzados, 498-500
  - botones, 498
  - botones de selección, 495
  - casillas de combinación, 497-498
  - casillas de verificación, 495-496
  - cuadros de lista, 496-497
  - cuadros de texto, 494-495
  - spin box, 498
  - en Visual Basic, 536
  - listas descendentes, 497
  - selección, 502
- estilos, 521
- marcos, 522
- menús; véase **Menús**
- problemas de diseño, 37
- uso de, 486
- ventanas, 522

Interfases gráficas de usuario (graphical user interfaces, GUI); véase **también Interfaz de usuario**

- controles, 492-494, 522
  - avanzados, 498-500
  - botones, 498
  - botones de selección, 595
  - casillas de combinación, 497-498
  - casillas de verificación, 495-496
  - cuadros de lista, 496-497
  - cuadros de texto, 494-495
  - spin box, 498
  - en Visual Basic, 536
  - listas descendentes, 497
  - selección, 502
- de sistemas operativos, 520
  - estilos, 521
  - marcos, 522
  - menús; véase **Menús**
  - problemas de diseño, 37
  - uso de, 486
  - ventanas, 522

Interfases operadas por instrucciones, 529-531

Interfaz de estilo consumidor, 528

Interfaz de usuario, 37, 355-356; véanse **también Interfaz gráfica de usuario (GUI); Entradas, diseño; Salidas, diseño**

- conceptos, 516
- controles, 522, 536
- controles internos, 531-532
- de inicios de sesión, 531
- de tipo consumidor, 528
- diálogo de usuario, 37
- diálogos de pregunta-respuesta, 531
- elaboración de prototipos, 516, 536, 538-541
- factores humanos, 516, 517-518
- guías, 517-519
- herramientas automatizadas, 386, 516, 536
- híbridas de Windows/Web, 529
- menús; véase **Menús**
- operadas por instrucciones, 529-531
- orientada a objetos, 77
- problemas, 517
- proceso, 535-536, 537

elaboración de prototipos, 538-541  
 gráficos de diálogo, 537-538  
 pruebas, 541  
 realimentación del usuario, 541  
 sistemas de ayuda, 532-535  
 tecnología  
   dispositivos de puntero, 521  
   monitores para visualización, 520-521  
   sistemas operativos, 520  
   teclados, 521  
 tono del diálogo y términos, 519  
 Internet; *véanse también* **negocios electrónicos; comercio electrónico; Navegadores web**  
   ambientes de desarrollo de software, 400  
   arquitecturas de aplicaciones basadas en, 389-392  
   correo electrónico, 397, 458  
   importancia, 390  
   portales, 390  
**intranet** (*intranet*) *Red de servidores que usa la tecnología de Internet para integrar computadoras personales, grupos de trabajo y cómputo empresarial, 390, 400*  
**entrada de datos** *Proceso de traducir datos a un formato legible para computadoras, 484-486; véase también* **Entrada de datos, diseño**  
 Introducción de datos sin teclado, 396  
 Introducción del informe, 330  
 Intuit; *véase* **Quicken**  
 IS; *véase* **sistema de información**  
 Ishikawa, Kaoru, 153  
 Isshiki, Koichiro R., 364, 370  
 TI; *véase* **tecnología de la información**

## J

Jacobsen, Ivar, 97, 187, 209, 284  
 JAD; *véase* **desarrollo conjunto de aplicaciones**  
 Java, 34, 38, 54, 77, 391, 400  
 JDBC (JavaBean database connectivity), 398  
 Gráfica de jerarquía; *véase* **diagrama de descomposición**  
 Jonnson, Patrick, 209  
 Joslin, Edward O., 364, 370  
 JPP; *véase* **planeación conjunta del proyecto**  
 JRP; *véase* **planeación conjunta de requerimientos**  
 Conjunciones, 269, 276

## K

Kara, Daniel A., 413  
 Kawasaki, 153  
 King, William R., 44  
 Kozar, Kenneth, 513

## L

LAN; *véase* **red de área local**  
 Lantz, Kenneth E., 371  
 Lápices, 396, 521  
 Larman, Craig, 203, 209  
**lenguaje corporal** *Información no verbal que comunicamos los seres humanos, 169-170*  
**lenguaje de definición de datos (data definition language, DDL)** *Lenguaje usado por un DBMS para definir una base de datos o una vista de la misma, 425*  
**lenguaje de manipulación de datos (data manipulation language, DML)** *Lenguaje de DBMS utilizado para crear, leer, actualizar y eliminar registros, 426*  
 Leventhal, N. S., 182  
 Levine, Martin, 22  
 Linderman, James, 21

Linkletter, Art, 169  
 Linux, 520  
 lista de casos de uso, 277, 283-284  
**lista de eventos-respuestas** *Una lista de eventos de negocios a los que el sistema debe proporcionar respuestas. Similar a una lista de casos de uso, 277, 283-284*  
 Listas descendentes, 497  
**lluvia de ideas** *Técnica de generación de ideas que alienta a los participantes para que propongan tantas ideas como sea posible en un periodo breve sin ningún análisis hasta que se hayan agotado todas las ideas, 175-176*  
 Lord Jr., Kenniston W., 182  
 Lotus  
   Notes, 385, 397  
   SmartSuite, 528  
   1, 2, 3, 460

## M

Macromedia  
   Cold Fusion, 38, 89  
   Dreamweaver, 38, 54  
   RoboHelp, 533  
 Malloy, John T., 334  
 Mandel, Theo, 547  
**manejador de evento (event handler)** *Un proceso que manipula un evento dado en la lista de eventos-respuestas, 277*  
 Maquiavelo, Nicolás, 332  
 Marcos, 522  
 Mariga, Julie, 44, 413  
 Martin, Alexander, 547  
 Martin, James, 213, 255, 308  
**matriz de análisis de factibilidad de soluciones** *Herramienta usada para clasificar las soluciones alternativas del sistema, 139, 327*  
**matriz de asociación proceso a ubicación** *Tabla usada para documentar procesos y la ubicación en la que se deben ejecutar, 302*  
**matriz de datos a ubicación de CRUD** *Una matriz que se usa para el mapeo de los requerimientos de datos a sus ubicaciones, 250*  
   CRUD Creación, lectura, actualización, borrado (Create, Read, Update, Delete) Datos de usuarios, 426  
**matriz de jerarquía y prioridad de casos de uso** *Herramienta usada para evaluar los casos de uso y determinar su importancia, 202-203*  
**matriz de soluciones alternativas del sistema** *Herramienta usada para documentar similitudes y diferencias entre soluciones alternativas del sistema, 137-139*  
 Matties, Leslie H., 296, 308  
 McClure, Carma, 308  
 McConnell, Steve, 97  
 McFadden, Fred, 448  
 Mcmenamin, Stephen M., 266, 308  
 Mejora; *véase* **mejora del sistema**  
 Mejora del sistema, 17  
 Mellor, Stephen J., 255  
 Menú, 522  
   sensible al (dependiente del) contexto, 526  
   de barras de herramientas, 526  
   de iconos, 526-527  
   descendentes, 522-524  
   en cascada, 522-524  
   hipertexto e hipervínculos, 528-529  
   pop-up, 525  
   tear-off, 525  
 Barra de menú, 522-523  
 Menús de iconos, 526-527



**metadatos** Datos correspondientes a otros datos, 228, 426

Methodware, véase **aplicación para administración del proceso**

**método ágil** Integración de diversos enfoques del análisis y diseño de sistemas para su aplicación según se considere apropiado al problema que se intenta resolver y el sistema que se está desarrollando. 108

AIS; véase **Association for Information Systems**

AITP; véase **Association for Information Technology Professionals**

**método de desarrollo de sistemas** Estrategia formal del proceso de desarrollo de sistemas; un proceso estandarizado que incluye actividades, métodos, mejores prácticas, productos y herramientas automatizadas que se usan en el desarrollo de sistemas de información, 50; véanse también **FAST**; **desarrollo rápido de aplicaciones (RAD)**

ágil, 64, 72

basado en modelos, 64, 72, 74-76

clasificación de, 72

comparación con el ciclo de vida del sistema, 50-51

ejemplos, 52

fases y actividades, 52-53

herramientas automatizadas, 87-88

métodos estructurados, 64

principios subyacentes, 52-56

uso de, 50

Metodología; véase **método de desarrollo de sistemas**

Métodos estructurados, 64

Métodos y procedimientos, sección, 330

MICR, véase **Tinta magnética, reconocimiento de caracteres**

Micro Focus

COBOL Workbench, 398, 399

Dialog Manager, 399

Microficha, 460

Microfilme, 460

Microsoft, 425; véase también **Windows**

Access, 106, 384, 424, 426, 428, 460-461, 484, 492, 500, 530-531, 536

agentes de ayuda, 534

COM +, 385

Excel, 460, 506

Exchange Server, 385, 397

FoxPro, 384, 428

Internet Explorer, 77, 486, 520

Office, 528, 533, 534

PowerPoint, 333

SQL Server, 54, 385, 393, 428

Transact SQL, 428

Visio, 104, 345

Visual Basic .NET, 34, 77, 106, 388

Visual Basic, 500

Visual Studio .NET, 34, 38, 54, 89, 399, 492, 516, 536

Windows Mobile, 396, 520

Microsoft Project, 91

**middleware** Software de utilería que permite interactuar a software de aplicación y de sistemas que usan distintas tecnologías, 38, 91, 397-398

Middleware de bases de datos, 398

Miller, Granville, 193n, 209

Miller, Irwin, 182

MIL-STD-498, 155

MIS; véase **sistema de información administrativa**

Mitchell, Ian, 183

**modelado de casos de uso** Proceso de modelación de las funciones de un sistema en términos de eventos de negocios, quiénes iniciaron los eventos y cómo responde el sistema a estos eventos, 187

administración de proyectos y, 202-204

beneficios, 187

desarrollo, 187

objetivo, 193

proceso, 193

casos de uso de requerimientos de negocios, 194-196

construcción de diagramas, 196-198

identificación de actores, 193

narrativa de casos de uso, 196-198

relaciones, 190-192

**modelación de procesos** técnica que se usa para organizar y documentar los procesos de un sistema, 259

**modelado de datos** Técnica para organizar y documentar los datos de un sistema. A veces llamado modelado de bases de datos, 77, 212; véanse también **diagrama de entidad relación**

**modelo** representación pictórica de la realidad, 258

**modelo de madurez de la capacidad (capability maturity model (CMM))** Marco estándar de referencia para evaluar el nivel de madurez del desarrollo de sistemas de información de una organización y sus procesos y productos administrativos. Consiste en cinco niveles de madurez, 49-50

**modelo físico** Representación pictórica técnica que ilustra lo que es un sistema o lo que hace y cómo se implanta el sistema. Los sinónimos son modelo de implantación y modelo técnico, 258

**modelo lógico** Representación pictórica no técnica que ilustra lo que es un sistema o lo que hace. Los sinónimos son modelo esencial, modelo conceptual y modelo de negocios, 258

Motor de la base de datos, 425

**muestreo aleatorio** Técnica de muestreo que se caracteriza por no contar con ningún patrón o plan predeterminado para seleccionar los datos de la muestra, 159

**muestreo estratificado** Técnica de muestreo sistemático que intenta reducir la varianza al dispersar el muestreo (por ejemplo, con una selección de los documentos o registros por fórmula) y evitar estimaciones muy altas o muy bajas, 159

Modelado de datos estratégicos, 225-227

Modelado de datos lógicos; véase **modelado de datos**

Modelo integral de datos, 228

**modelado de objetos** Técnica que intenta fusionar los datos y procesos en un solo concepto, llamado objeto. Los modelos de objetos son diagramas que documentan un sistema en cuanto a sus objetos y sus interacciones. El modelado de objetos es la base del análisis orientado objetos y de los métodos de diseño, 77

**modelado de proceso** Técnica centrada en procesos que se popularizó con el análisis estructurado y los métodos de diseño en los que se aplican modelos de requerimientos de procesos de negocios con el fin de derivar diseños de software efectivos para un sistema. Con el análisis estructurado, se introdujo una herramienta de modelado que se llama diagrama de flujo de datos para ilustrar el flujo por una sucesión de procesos de negocios. El diseño estructurado convirtió los diagramas de flujo de datos en modelo de procesos, los gráficos estructurales, para ilustrar una estructura descendente de software que satisface los requerimientos de negocios, 76-77, 259; véase también **diagrama de flujo de datos (data flow diagram, DFD)**

en el diseño de sistemas, 279

en las fases del análisis de sistemas, 277-279

estratégico, 276

habilidades, 141

herramientas automatizadas, 279

ingeniería inversa, 277

operado por eventos, 277-278

para rediseño de procesos de negocios, 261, 276-277

proceso, 277, 280

diagrama de flujo de datos de contexto, 280-281

diagramas de descomposición de eventos, 284, 287

diagramas de descomposición funcional, 277, 281-282

diagramas de eventos, 287-289

diagramas de sistema, 287-288

diagramas primitivos, 277-291  
 equilibrio, 290-291  
 identificación de hechos, 279  
 listas de casos de uso, 283-284  
 listas de eventos-respuestas, 283-284  
 sincronización con modelos de datos, 301-302

**modelo** *Una representación de la realidad con imágenes, 104, 258; véanse también modelado de datos; modelado del proceso; modelo del sistema*

**modelo de datos basado en claves** *Uno que incluye entidades y relaciones con cardinalidades precisas que resuelven relaciones inespecíficas en entidades asociativas, además de incluir claves primaria y alterna, 228, 234-235*

Modelo conceptual; véase **modelo lógico**

Modelo de ambiente; véase **diagrama de contexto de flujo de datos**

**modelo de datos de la aplicación** *Modelos de datos para un solo sistema de información completo, 227*

**modelo de datos de contexto** *Uno que incluye entidades y relaciones, sin atributos, 227, 232-234*

modelo de datos empresariales, 226-227

**modelo de datos lógico** *Técnica de solución de problemas que descompone el sistema en sus componentes para estudiar el grado en que éstos funcionan e interactúan para lograr su propósito, 102*

- análisis de requerimientos; véase **identificación de requerimientos**
- estrategias, 103
  - acelerada, 105-107
  - basada en modelos, 103-105
  - orientado a objetos, 105
  - tradicional, 104-105
- fase de análisis de decisiones, 134-136
  - actualización del plan del proyecto, 139
  - análisis de factibilidad, 137-139, 314-315, 324
  - análisis de soluciones candidatas, 137-139
  - comparación de soluciones candidatas, 139, 324-327
  - identificación de soluciones candidatas, 136, 137, 325
  - propuesta de sistema, 139-141
- fase de análisis de requerimientos, 127
  - actualización del plan del proyecto, 130
  - administración continua de requerimientos, 131
  - comunicación de la declaración de requerimientos, 131
  - identificación de requerimientos, 127-130
  - priorizar requerimientos del sistema, 130
  - modelado de datos, 227-228
- fase de análisis del problema, 117-118
  - actualización del plan del proyecto, 125
  - análisis de factibilidad, 314
  - análisis de problemas y oportunidades, 122
  - análisis de procesos de negocios, 122-124
  - comunicación de resultados, 125-126
  - dominio del problema, 117-122
  - modelado de datos, 227
  - modelado de procesos, 277-279
  - objetivo, 117
  - objetivos de mejora del sistema, 124-125
- fase de diseño lógico, 131
  - casos de prueba de aceptación, 134
  - elaboración de prototipos, 134
  - estructuración de requerimientos funcionales, 133
  - validación de requerimientos funcionales, 134
- futuro de, 141

Modelo de datos normalizados, 228

**modelo de sistema** *Imagen de un sistema que corresponde a la realidad o a una realidad deseada. Los modelos de sistemas facilitan la mejora de la comunicación entre los usuarios, analistas, diseñadores y constructores del sistema, 64*

Modelo de sistema "tal y como es", 74

**modelo físico** *Representación visual técnica de qué es un sistema o qué hace y la forma de ponerlo en práctica. Sus sinónimos son modelo de puesta en práctica y modelo técnico, 74, 258*

**modelo integral de datos (fully attributed data model)** *Un modelo de datos que incluye todas las entidades, atributos, relaciones, criterios de subconjuntos y cardinalidades precisas, 228, 237-239*

Modelo de procesos empresariales, 276

**modelo lógico** *Representación visual de qué es o hace un sistema. Sus sinónimos son modelo esencial, modelo conceptual y modelo de negocios, 74, 258*

Modelos de negocios; véase **modelo lógico**

Modelos esenciales; véase **modelo lógico**

Modelos técnicos; véase **modelo físico**

Monitores de visualización, 520-521

Moses, John, 183

Mozilla Firefox, 486, 520

Muchos a muchos, relación; véase **relación inespecífica**

Multimedia, 458

**muestreo** *Proceso de recopilar una muestra representativa de documentos, formatos y registros, 158-159*

## N

**narración (o narrativa) de caso de uso** *Texto que describe el evento del negocio y la forma en que el usuario interactuará con el sistema para lograr la tarea, 188*

- desarrollo, 198-200

- giro de eventos, 200

Netscape

- Commerce Server, 385

- Navigator, 77

Newcomer, Eric, 413

Niku

- open Workbench, 91

- Project Manager, 91

**nombre de función** *Nombre de clave foránea que refleja su propósito en una tabla, 438*

- criterios de subconjuntos, 216

- nombres de, 237

**nombre de un rol** *Nombre de clave foránea que refleja el propósito para el que sirve ésta en una tabla, 438*

normalización

- criterios de evaluación, 240

- en las fases del análisis de sistemas, 227-228

- estratégico, 225-227

- habilidades, 141

- herramientas automatizadas, 228-230, 248

- ingeniería inversa, 227, 229

- proceso, 225, 227, 228, 230

- análisis de modelo, 240-241, 428-429

- identificación de entidad, 231-232

- modelo de datos basado en llaves, 228, 234-235

- modelo de datos de contexto, 227, 232-234

- modelo integral de datos, 228, 234-235

- simplificación por inspección, 248

- sincronización con los modelos de procesos, 301-302

- transformación en esquemas de bases de datos, 430-432

- vistas de ubicación, 248-250

**normalización** *Técnica de análisis de datos que los organiza en grupos para formar entidades no redundantes, estables, flexibles y adaptables, 228, 231*

- como requerimiento para el diseño de bases de datos, 428-429

- herramientas automatizadas, 248

- primera forma normal (first normal form, 1NF), 241-244

- segunda forma normal (second normal form, 2NF), 241, 244

- tercera forma normal (third normal form, 3NF), 241, 244-248

NSA Report Web, 460

cómputo de cliente/servidor de N-capas; véase **datos y aplicaciones distribuidos**

## O

**objetivo** *Una medición del éxito. Es algo que se espera lograr, si se tienen recursos suficientes, 124-125*

**objeto** *Encapsulación de datos (llamados propiedades) que describen a una persona, objeto, sitio o evento, con todos los procesos (llamados métodos) permitidos para usar o actualizar los datos y propiedades. La única forma de tener acceso a los datos del objeto o actualizarlos es usar los procesos predefinidos del objeto, 105*

Objetos de negocio; véase **objeto**

Objetos, sistemas de administración de bases de datos de, 442

**observación** *Técnica de identificación de hechos en la que el analista de sistemas participa u observa a otra persona que realiza actividades para aprender sobre el sistema, 160-162*

OCR; véase **Reconocimiento óptico de caracteres**

OD; véase **duración optimista**

ODBC; véase **Conectividad abierta de base de datos (open data-base connectivity, ODBC), herramientas**

OMG; véase **Object Management Group**

OMR; véase **Reconocimiento de marcas ópticas**

OOA; véase **análisis orientado a objetos**

OOD; véase **análisis y diseño orientados a objetos**

OOD; véase **diseño orientado a objetos**

**operado por menús** *Estrategia de diálogo en la que el usuario debe seleccionar una acción de un menú de opciones, 522*

**oportunidad** *Ocasión de mejorar la organización, incluso en ausencia de un problema identificado, 57*

OPT soluciones posibles, alternativas, 114

Oracle Corporation

bases de datos Oracle, 54, 385, 393, 424, 425, 428

Designer, 98, 426

Developer, 89

Oracle Forms, 54

PL/SQL, 428

Orfali, Robert, 413

Orr, Ken, 97

Overgaard, Gunnar, 209

## P

**paginación** *Despliegue de una pantalla completa de caracteres a la vez, 521*

Palm, 396, 487, 520

Palmer, John F., 266, 308

Pantalla de salida, 556; véase también **Interfaces gráficas de usuario (GUI)**

Pantallas sensibles al tacto, 487

**paquete de aplicación comercial** *Una aplicación de software que se puede comprar y personalizar (dentro de ciertos límites) para satisfacer las necesidades de negocios de numerosas organizaciones o de una industria específica. Sinónimo de sistema comercial listo para usarse (comercial off-the-shelf, COITS), 80*

efecto en el ciclo de vida del desarrollo de sistemas, 358, 364-365

estrategia de implantación FAST, 81-83

fase de diseño del sistema, 358-360

    criterios y opciones técnicas, 360-361

    fase de abasto, 358, 360-363

    fase de análisis de decisiones, 358, 360

    fase de evaluación y clasificación, 363-364

    negociaciones de contratos, 364

    reuniones informativas con el proveedor, 364

    solicitud de propuesta, 361-363

    validación de credenciales y desempeño del proveedor, 363

implantación, 365

solicitudes de propuestas, 81, 361-363

ventajas y desventajas, 83-84

Paquetes; véase **paquete de aplicaciones comerciales**

Paquetes de autoría de, 533

Paquetes de software; véanse **Herramientas automatizadas; paquete de aplicaciones comerciales**

Paradice, David, 182

Parrington, Norman, 183

**parálisis de análisis** *Término satírico acuñado para describir una situación común en proyectos, en la que el modelado excesivo del sistema desacelera el avance hacia la puesta en práctica de la solución de sistema que se pretende implantar, 64, 119, 277*

**partición** *Acto de determinar la mejor forma de distribuir o duplicar los componentes de aplicaciones a través de una red, 389, 393*

**partición de eventos** *Una estrategia de análisis estructurada en la que un sistema se divide en subsistemas en base a los eventos de negocios y la respuestas a esos eventos, 277*

Paulk, Mark C., 97

partición de datos, 393

PD; véase **duración pesimista**

PDFD; véase **diagrama de flujo físico de datos**

Persona/máquina, límites, 408

**periodo de recuperación de la inversión** *Periodo que transcurre antes de que los beneficios acumulados superen a los costos acumulados, 321*

**PIECES** *Marco de referencia desarrollado por James Wetherbe, para clasificar problemas en categorías: rendimiento, información, economía, control, eficiencia y servicio (Performance, information, economics, control, efficiency, service), 57, 150, 315*

**plan estratégico de la empresa** *Plan formal (de tres a cinco años) de todo negocio que define su misión, visión, objetivos, estrategias, normas de comparación y mediciones de avance y logros. Es usual que el plan estratégico se complemente con planes estratégicos de unidades de negocios, en los que se define la manera en que cada unidad de negocios contribuirá al plan de la empresa. El plan de sistemas de información es uno de esos planes en el nivel de unidad, 55*

**plan estratégico de sistemas de información** *Plan formal (tres a cinco años) para construir y mejorar la infraestructura de tecnología de la información y las aplicaciones de sistemas de información que usan dicha infraestructura, 55, 57*

**planeación conjunta de requerimientos (joint project planning, JRP)** *Uso de talleres facilitados para reunir a todos los propietarios, usuarios y analistas de un sistema y a ciertos diseñadores y constructores de sistemas con el fin de realizar conjuntamente el análisis de sistemas. La JRP por lo general se considera como parte de un método más amplio, llamado desarrollo conjunto de aplicaciones (JAD), que es una aplicación más completa de las técnicas de JRP al proceso de desarrollo de sistemas en su totalidad, 108*

**planeación conjunta de requerimientos (joint requirements planning, JRP)** *Proceso por el que se realizan juntas de grupo muy estructuradas para analizar problemas y definir requerimientos, 171*

    agenda, 175

    beneficios, 176

    conducción de las sesiones, 175-176

    diseño de sala, 174

    facilitador, 172, 175

    modelado de datos y, 228

    para el modelado de procesos, 279

    participantes, 172-173, 175

    patrocinador, 172

    planeación, 173-175

    producto final, 176

    ubicaciones, 173-175

**planeación de recursos de una empresa (enterprise resource planning, ERP)** *Aplicación de software que integra por*

completo los sistemas de información usados en muchas de las funciones empresariales centrales o básicas, o todas ellas (lo que incluye el procesamiento de transacciones e información administrativa correspondiente a esas funciones empresariales), 80-81

Planeación estratégica, 57, 276

PMBOK; véase **Project Management Body of Knowledge**

**política** Conjunto de reglas que rige un proceso de negocios, 32-34, 299

Popkin; véase **System Architect 2001**

Portales, 390

POS; véase **Punto de venta, terminales**

**pregunta abierta** Una pregunta que permite al entrevistado responder de la manera que le parezca apropiada, 165

**pregunta de respuesta cerrada** Una que restringe la respuesta a opciones específicas o a respuestas directas y breves, 165

Prescott, Mary, 448

**presentación** La actividad continua de comunicar hallazgos, recomendaciones y documentación para su revisión por los usuarios interesados y administradores. Las presentaciones pueden ser escritas o verbales, 69; véanse también **Presentación formal; Informes escritos**

**presentación distribuida** Sistema cliente/servidor en el que la presentación y la lógica de la presentación se trasladan desde el servidor para residir en el cliente, 385-386, 398-399

**presentación formal** Una junta especial que se usa para vender nuevas ideas y obtener la aprobación de nuevos sistemas, 331

auxiliares visuales, 333

conducción, 334-335

preparación para, 332-334

seguimiento, 335

ventajas y desventajas, 331-332

Presentación, middleware, 398

**primera forma normal (1NF)** Una entidad cuyos atributos no tienen más que un solo valor para una sola instancia de esa entidad, 241-244

**problema** Situación indeseable que impide a la organización lograr plenamente su misión, visión, metas y objetivos, 11, 57

actualización del plan del proyecto, 125

análisis de factibilidad, 314

análisis de problemas y oportunidades, 122

análisis de procesos de negocios, con 122-124

comunicación de resultados, 125-126

dominio del problema, 117-122

método FAST, 62-63

modelado de datos, 227

modelado de procesos, 277

objetivo, 117

objetivos de la mejora del sistema, 124-125

**procedimiento** Conjunto de instrucciones paso a paso y lógica para lograr un proceso de negocios, 32

**procedimientos almacenados** Programa incorporado en una tabla que puede llamarse desde un programa de aplicación, 428

procesamiento de datos

en línea, 388, 394-395, 486

por lotes, 394, 486, 491

por lotes remoto, 395-396, 486

**procesamiento en línea** Método de procesamiento de datos mediante el cual se procesan inmediatamente los datos de una transacción individual, 388, 394-395, 486

**procesamiento por lotes** Método de procesamiento de datos mediante el cual se colectan datos de muchas transacciones como un archivo individual, que luego se procesa, 394, 486

controles, 491

remoto, 395-396, 486

**procesamiento remoto de lotes** Método de procesamiento de datos mediante el cual se ingresan los datos en línea, se colectan en un lote y se procesan posteriormente, 395-396, 486

**proceso** Trabajo que realiza un sistema en respuesta a condiciones o flujos de datos entrantes. Sinónimo de transformación, 264

descomposición, 264-266

distribución, 302, 305

elementales, 267, 291, 300

en diagramas de flujo de datos, 259, 263

físico, 375-380

instrucciones (lógicas), 291, 295

lenguaje de procedimiento, 295-300

límites persona/máquina, 408

lógico, 266-267, 376-377

métodos de ejecución, 377-378

nombres, 377, 378

sistemas como, 263-264

tablas de decisión, 299-300

**proceso de desarrollo de sistemas** Conjunto de actividades, métodos, mejores prácticas, productos y herramientas automatizadas que los interesados (véase el capítulo 1) que se usan para desarrollar y mejorar continuamente los sistemas de información y software (véanse capítulos 1 y 2). 48; véase también **desarrollo de sistemas, 48**

actividades transversales del ciclo de vida, 68-69

Capability Maturity Model, 49-50

estandarizado, 48, 50

estrategia de cascada, 69

estrategia iterativa, 69-72

estrategia secuencial, 69

rutas alternas, 72-74

**proceso elemental** Actividad o tarea discreta y detallada que se requiere para completar la respuesta a un evento. También llamado **proceso esencial**, 267, 291, 350

Proceso Unificado de Rational (Rational Unified Process, RUP), 64

**programa de aplicación** Representación basada en un lenguaje y legible para la computadora sobre lo que se supone que debe hacer un proceso de software o la forma en que éste presuntamente ha de realizar su tarea, 35; véase también **esquema de aplicación**, 279

Programación; véase **construcción de sistemas**

programación basada en repositorios, 492-294, 502

Programación estructurada, 297, 345

Programación extrema, 64

Programadores, 10

Programas, 34-35

módulos, 345

Programmers Paradise, 399

**propietario del sistema** Patrocinador y proponente ejecutivo de un sistema de información, que generalmente se encarga del financiamiento del proyecto y del desarrollo, operación y mantenimiento del sistema de información, 7, 8

componentes de comunicaciones y, 35

componentes de conocimientos y, 27, 30

componentes de procesos y, 31-32

perspectiva del sistema de información, 26-27

**propuesta de sistema** Informe o presentación de una solución recomendada, 139-141; véase también **Informes escritos**

**prototipo** Modelo a pequeña escala, representativo o de trabajo de los requerimientos de los usuarios o de un diseño propuesto de un sistema de información. Todo prototipo puede omitir ciertas funciones o características hasta el momento en que haya evolucionado en grado suficiente para convertirse en una puesta en práctica aceptable de los requerimientos, 78, 105

de bases de datos, 439

ingeniería inversa, 107

Procesos físicos, 375-378

**proveedor de servicios externo (external service provider, ESP)** Un analista, diseñador o constructor de sistemas que vende

su conocimiento y experiencia a otras empresas para ayudarlas en la compra, el desarrollo o la integración de sus soluciones de sistemas de información; puede estar afiliado a una organización de consultoría o servicios, **16**

Proveedores; véase **paquete de aplicaciones comerciales**

**proxémica** *Relación entre las personas y el espacio que las rodea, 170*

**proyecto** *Secuencia de actividades que son completadas en tiempo, sin presupuesto, y de acuerdo a una especificación*

cancelación, 55

fallas, 48, 69, 186

presupuestos, 115

Proyectos cancelados, 55

Prueba de aceptación, 134

**prueba de aceptación de sistemas** *Una prueba que se realiza con la versión final del sistema, en la que el usuario realiza tareas de verificación, validación y auditoría, 134*

Pruebas

de aceptación, 134

de interfaz de usuario, 541

Punto de venta, terminales, 458, 487

Purdue University, 494-495

## Q

QBE; véase **Consulta por ejemplo (Query by Example, QBE)**, 530

Quicken, 491, 529

## R

RAD; véase **desarrollo rápido de aplicaciones**

Railroad Paradox, 160

Rational ROSE, 88, 104

**ratón** *Dispositivo usado para hacer que un puntero se mueva en una pantalla, 488, 521*

Rawles, Phillip, 44, 413

funcionales; véase **requerimientos funcionales**

no funcionales, 127, 150

obligatorios, 130

PIECES, clasificación, 150

recomendables, 130

Reconocimiento de caracteres de tinta magnética (magnetic ink character recognition, MICR), 488

Reconocimiento de voz, 487

Reconocimiento óptico de caracteres (optical character recognition, OCR), 396, 488

Reconocimiento óptico de marcas (optical mark recognition, OMR), 396, 487-488

Recopilación de información; véase **identificación de hechos**

**red de área local (local area network, LAN)** *Conjunto de computadoras cliente conectadas a una distancia relativamente corta con uno o más servidores, 383-384*

Redes; véanse también **Internet**; **red de área local**

arquitectos, 10

arquitecturas, 403-404

estrategia de disposición en capas limpia, 39

función en los sistemas de información, 39

intranets, 390, 400

**rediseño de procesos de negocios (business process redesign, BPR)** *La aplicación de métodos de análisis de sistemas con el objetivo de cambiar y mejorar significativamente los procesos de negocios fundamentales de una organización, con independencia de la tecnología de información, 108*

diagramas de flujo de datos usados en, 261, 276-277

métodos de, 108

proyectos iniciados a partir de, 57

**registro** *Conjunto de campos dispuestos en un formato predeterminado, 421-422, 439*

Registros de longitud fija, 421

Registros de longitud variable, 421-422

Reingruber, Michael, 255

**relación** *Una asociación de negocios natural entre dos o más entidades, 216-217*

cardinalidad, 217

grado, 217

identificatoria, 221

inespecífica (muchos a muchos), 221-224

múltiple, 234

N-aria, 218

no identificatoria, 220-221

nombres, 217, 232

recursiva, 218

Relación de objetos y clases; véase **objetos, relaciones entre clases**

Relación en el modelado de casos de uso, 190-192

asociaciones, 190

de extensión, 190

dependencia en, 191-192

herencia, 192

usos (incluyente), 191

**relación de identificación (identidad)** *Relación en la que la clave de la entidad madre es también parte de la clave primaria de la entidad hija, 221*

**relación inespecífica** *Una relación en la que muchas instancias de una entidad se relacionan con otras muchas de otra entidad. También llamada relación de muchos a muchos, 221-224*

**relación no identificada** *Relación en la que cada entidad participante tiene su propia clave primaria independiente, 220-221*

**relación recursiva** *Una que existe entre instancias de una misma entidad, 218*

Relaciones todo-partes; véase **agregación**

Renaud, Paul, 413

Replicación, 438-439

Replicación de datos, 393

**reporte detallado** *Salida interna que presenta información con poco o ningún filtrado, 452*

**reporte de excepción** *Salida interna que filtra los datos para presentar información que reporta las excepciones de alguna condición o estándar, 455*

**reporte resumido** *Salida interna que condensa la información para los gerentes, 455*

**repositorio** *Base de datos o directorio de archivos donde los desarrolladores de sistemas guardan toda la documentación, conocimientos y herramientas de uno o más proyectos o sistemas de información. Los depósitos usualmente se automatizan para facilitar el almacenamiento, recuperación, así como para compartir la información, 69*

**repositorio** *Ubicación o conjunto de ubicaciones donde los analistas de sistemas, diseñadores de sistemas y constructores de sistemas guardan toda la documentación relacionada con uno o más sistemas o proyectos. 102; véase también **repositorio CASE***

modelos de datos guardados en, 228

opciones de ejecución, 102

**repositorio CASE** *Base de datos de desarrolladores de sistemas donde pueden almacenar modelos de sistemas, descripciones y especificaciones detalladas, y otros productos del desarrollo de sistemas. Sus sinónimos abarcan enciclopedia y diccionario de datos, 88, 91; véase también **repositorio***

**requerimiento funcional** *Descripción de las actividades y los servicios que debe brindar un sistema, 127; véase también **identificación de requerimientos***

casos de uso, 129-130

elaboración de prototipos, 134

estructuración, 133

identificación, 150

validación, 134

**requerimiento funcional** *Algo que debe hacer el sistema de información, 150*

**requerimiento no funcional** *Descripción de otras características y restricciones que definen un sistema satisfactorio, 127*

**requerimiento no funcional** *Propiedad o cualidad que debe tener el sistema. Ejemplos incluyen seguridad, facilidad de uso, rendimiento, etcétera, 150*

**requerimientos de datos** *Representación de los datos de los usuarios con base en entidades, atributos, relaciones y reglas, 30, 248-250*

Requerimientos de negocio; véase **requerimientos de sistemas**

**requerimientos de proceso** *Expectativas del usuario acerca de los requerimientos de un proceso de negocios y sus sistemas de información, 32-34*

**requerimiento del sistema** *Algo que el sistema de información debe hacer o una propiedad que debe poseer. También llamado requerimiento del negocio, 150; véanse también Fase de análisis de requerimientos; identificación de requerimientos,*

costo de los errores, 150-151

critérios, 152

**requerimientos (requisitos)** véase **identificación de requerimientos**

**restricción** *Algo que limita la flexibilidad en la definición de una solución según los objetivos que se tengan. En lo esencial, es imposible modificar las restricciones, 124*

**restricción** *Algún factor, limitación o restricción que puede limitar una solución o un proceso de solución de problemas, 62*

**resumen del informe** *Una salida interna que clasifica la información para los gerentes, 455*

Resumen ejecutivo, 330

RFP; véase **solicitud de propuesta**

RFQ; véase **solicitud de cotización**

Robertson, James, 183, 266, 309

Robertson, Suzanne, 183, 266, 309

RoboHelp, 533

Rosenblatt, H., 481

Rumbaugh, James, 97

RUP; véase **Proceso Unificado de Rational (Rational Unified Process)**

## S

**salida con regreso** *Salida externa que puede reingresar al sistema como entrada, 455, 466*

**salida interna** *Salida para los dueños y usuarios del sistema dentro de una organización, 452*

**salida por zona** *Una que presenta el texto y números en las áreas designadas de una forma o pantalla, 456-458*

**salida externa** *Una que se promueve fuera de la organización, 455, 466*

**salida gráfica** *Una salida que utiliza un gráfico para comunicar información, 468*

**salida interna** *Una para los propietarios y usuarios del sistema al interior de una organización, 452, 455*

**salida tabular** *Salida que presenta la información a manera de columnas de texto y números, 466, 467*

Salidas; véase también **Informes**

distribución y auditorio, 452, 455

externas, 455, 466

internas, 452, 455

métodos de puesta en práctica, 455

basadas en Web, 460, 472-475

correo electrónico, 458

en pantalla, 458

hipervínculos, 460, 474

impresas, 456-458

microfilmes, 460

multimedia, 458

terminales de puntos de venta, 458

rotación, 455, 466

taxonomía, 452

Salidas basadas en Web, 460, 472-475

Salidas impresas, 456-458

Salidas multimedia, 458

Salvendy, G., 483

Sarson, Trish, 375, 413

Schlaer, Sally, 255

Schmeiser, Lisa, 547

SCM; véase **administración de la cadena de suministro**

SDE; véase **ambiente de desarrollo de software,**

sección de hechos, 330

**segunda forma normal (second normal form, 2NF)** *Una entidad cuyos atributos que no son la clave primaria dependen de la clave primaria completa. 241, 244*

Seguridad

claves de conexión, 531

del comercio electrónico, 532

privilegios de, 531-532

Spin box (Selector numérico), 498

**Servidor computadora generalmente con gran capacidad de procesamiento de datos, en contraste con una computadora personal (PC) que actúa como cliente; véase también sistema cliente/servidor**

de aplicación, 385

de base de datos, 385, 388

de mensajería o groupware, 385

transacción, 385

Web, 385, 389

**servidor de aplicaciones** *Un servidor que aloja la aplicación lógica y los servicios de un sistema de información, 385*

**servidor de bases de datos** *Un servidor que aloja una o más bases de datos, 385, 388*

**servidor de mensajería o groupware** *Uno que aloja servicios de groupware, 385*

**servidor de transacciones** *Un servidor que aloja servicios para garantizar que para una transacción todas las actualizaciones de una base de datos tienen éxito o fallan como un todo, 385*

**servidor Web** *Uno que aloja sitios de Internet o intranets, 385, 389*

Sethi, Vikram, 44

Shelly, G., 481

Silver, Denise, 147, 183, 371

Sintaxis basada en lenguaje, 530

Sintaxis del lenguaje natural, 530

Sintaxis nemotécnica, 530

**sistema** *Grupo de componentes interrelacionados que funcionan juntos para lograr un resultado buscado, 6*

como proceso, 263-264

**sistema centralizado** *Uno en el cual todos los componentes son atendidos en una computadora central multiusuario, 382*

**sistema cliente-servidor** *Solución de cómputo distribuido, en la cual la presentación, la lógica de presentación, la lógica de aplicación, la manipulación de datos y las capas de datos se distribuyen entre las PC clientes y uno o más servidores, 384-385*

ambientes del desarrollo de software, 398-400

datos distribuidos, 387-388, 404-405

datos distribuidos y aplicación, 388-389

de dos capas, 387-388, 399

de tres capas, 388-389, 399-400

presentación distribuida, 385-386, 398-399

**sistema de administración de bases de datos (database management system, DBMS)** *Software especial que se usa para crear una base de datos, tener acceso a ella, controlarla y administrarla, 425-426; véase también bases de datos relacionales*

- lenguaje de definición de datos (data definition language, DDL), 425
- lenguaje de manipulación de datos (data manipulation language, DML), 426
- objeto, 442
- pros y contras, 420
- sistema de administración de bases de datos relacionales distribuidas** *Software con el que se implantan las bases de datos relacionales distribuidas, 392-393*
- sistema de automatización de oficinas** *Sistema de información que brinda soporte a la amplia gama de actividades de oficina de los negocios para mejorar el flujo de trabajo entre los empleados, 7*
- Sistemas operativos, 520
- Sitios web 385, 389
- Smalltalk, 77
- Smith, Patrick, 413
- Smith, Randi Sigmund, 339
- Software Engineering Institute, Carnegie Mellon University, 69
- sistema de cómputo en red** *Solución de varias capas en la cual las capas de presentación y de lógica de presentación se ejecutan en navegadores Web del lado del cliente, usando el contenido descargado de un servidor Web, 389-392*
- sistema de comunicación y colaboración** *Sistema de información que posibilita la comunicación más efectiva entre los empleados, socios, clientes y proveedores, para mejorar su capacidad de colaboración, 7*
- sistema de información, SI (information system, IS)** *Conjunto de personas, datos, procesos y tecnología de la información que interactúan para recopilar, procesar, guardar y proporcionar como salida la información necesaria para brindar soporte a una organización, 6*
- back-office, 25
  - clases, 6-7, 25-26
  - como inversiones de capital, 55
  - componentes básicos, 27
    - comunicaciones, 35-38
    - conocimientos, 27, 30-31
    - procesos, 31-32, 34-35
    - redes y, 38-39
  - front-office, 24-25
  - federación de, 24-25
  - impulsores de empresas, 16-17
  - transversal, 52
- sistema de información de front-office** *Un sistema de información que brinda soporte a funciones de negocios que se extienden hacia fuera, a los clientes de la organización, 24-25*
- sistema de información administrativa (management information system, MIS)** *Sistema de información que provee informes orientados a la administración basado en el procesamiento de las transacciones y operaciones de la organización, 6*
- sistema de información de back-office** *Un sistema que brinda soporte a las operaciones internas del negocio de una organización, además de permitir el acceso a los proveedores, 25*
- sistema de información ejecutiva (executive information system, EIS)** *Sistema de información que brinda soporte a las necesidades de planeación y evaluación de los administradores de nivel ejecutivo, 7*
- sistema de información transversal** *Sistema que brinda soporte a procesos de negocios pertinentes de varias funciones de negocios sin importar los límites organizacionales tradicionales, como las divisiones, departamentos, centros y oficinas, 33*
- sistema de procesamiento de transacciones (transaction processing systems, TPS)** *Sistema de información en el que se capturan y procesan los datos relativos a transacciones de negocios, 6, 388*
- sistema de servidor de archivos** *Una LAN en la cual un servidor aloja los datos de un sistema de información. 383-384, 387-388*
- sistema de soporte a las decisiones (decisión support system, DSS)** *Sistema de información que ayuda a identificar oportunidades de toma de decisiones o proporciona información que ayuda a tomarlas, 7*
- sistema distribuido** *Uno en el que los componentes se distribuyen a través de múltiples ubicaciones y redes de computadoras, 382; véase también cliente/servidor, sistema*
- arquitectura, 382
  - arquitectura del servidor de archivos, 383-384, 387-388
  - partición, 389, 393
  - ventajas y desventajas, 382
- sistema experto** *Sistema de información en que se captura la experiencia de los expertos humanos y luego el sistema usa la experiencia capturada para beneficio de quienes no son expertos, 7*
- recuperación de sistema, 68
- Sistemas de, 532-535
- solicitud de cotización (request for quotation, RFQ)** *Documento formal en el que se comunican los requerimientos técnicos y de soporte del negocio de un paquete de software de aplicación a un solo proveedor, al que se le ha considerado capaz de proporcionar el paquete de aplicaciones y servicios solicitados. 81, 361-363*
- solicitud de propuesta (request for proposal, RFP)** *Documento formal que comunica los requerimientos técnicos y de soporte del negocio para un paquete de software de aplicaciones a los proveedores que desean competir por la venta de ese paquete de aplicaciones y servicios, 81, 361-363*
- Sonido, entradas, 487
- de prototipos, 107
  - modelos de datos, 227, 229
  - modelos de proceso, 277
  - soporte de herramientas CASE, 88-89, 107
- soporte del sistema** *Soporte de naturaleza técnica y continua para los usuarios de un sistema, así como el mantenimiento necesario para hacer frente a errores, omisiones o requerimientos nuevos que puedan surgir, 68*
- actividades, 68
  - entropía del sistema, 56
  - mejoras al sistema; véase mejora
  - método FAST, 68, 84-87
  - recuperación del sistema, 68
  - caso de uso; véase análisis de sistemas
- SQL (Structured Query Language)
- generación de bases de datos, 429, 439, 442
  - instrucciones, 427
  - uso por los usuarios finales, 530
- Stallings Jr., Warren D., 182
- Standish Group, 186
- STD; véase diagrama de transición de estados
- Steiner, James B., 182
- Structured Query Language; véase SQL
- Strunk Jr., Williams, 331
- Stuart, Ann, 339
- Stubbe, John, 339
- subtipo** *Una entidad cuyas instancias pueden heredar atributos comunes en la entidad supertipo, 225, 237, 432*
- supertipo** *Entidad cuyas instancias guardan atributos que son comunes a uno o más subtipos de entidades, 225, 227*
- Sybase Corporation, 425
- Powerbuilder, 38, 54, 89, 106, 388, 399, 492
  - base de datos Sybase, 393, 424, 428
- Symantec Visual Café, 492
- Symbol Technologies, 487
- System Architect 2001 (Popkin), 88, 104
- descripciones de procesos elementales, 300
  - diagramas de transición de estados, 538
  - diseño de base de datos, 426, 429, 432, 435, 439, 442
  - diseño de pantalla, 461, 500, 536

modelado de datos, 230, 232, 438  
 modelado de procesos, 279, 291  
 modelos de diseño de sistemas, 345

## T

**tabla** *Equivalente a un archivo en una base de datos relacional, 422*

clave, 421, 430, 435  
 índices, 430, 435  
 nombres, 430  
 procedimientos guardados, 428  
 registros, 421-422, 439  
 relacional, 426  
 tamaños, 439

**tabla de archivos de consulta** *Una que contiene datos relativamente estáticos, que pueden compartirse, 422*

**tabla de decisiones** *Forma tabular de presentación que especifica un conjunto de condiciones y las acciones correspondientes, 154, 297, 299-300*

Tabulación, 539

Tareas, diagramas, 109

Tarifas de descuento, 321-322

Tarjetas inteligentes, 489

Taylor, David, 44

Teclados, 487, 521

**teclas de función** *Conjunto de teclas especiales del teclado que se usa para programar operaciones especiales, 521*

Tecnología biométrica, 489

Tecnología de groupware, 385 software para trabajo en equipo o colaborativo

**tecnología de la información (information technology, IT)** *Término contemporáneo que describe la combinación de la tecnología de computadoras (hardware y software) con la de telecomunicaciones (redes de datos, imágenes y voz), 6*

Tecnología de trabajo en grupos (o workgroup), 397

Teorey, Toby, 255, 448

**tercera forma normal (third normal form, 3NF)** *Una entidad cuyos atributos de clave no primaria son independientes de otros atributos de claves no primarias, 241, 244-248*

Terhune, Alyse D., 44

Texto, cuadros, 494-495

Theby, Stephen E., 414

Thill, John V., 339

**timeboxing (tiempo fijo)** *La imposición de un periodo no prorrogable, usualmente de 60 a 90 días, al término del cual debe entregarse para su funcionamiento la primera (o siguiente) versión de un sistema, 80*

**timeboxing** *Técnica que entrega funcionalidad y requerimientos de sistemas de información mediante el control de versiones. El equipo de desarrollo selecciona el subconjunto más pequeño del sistema que al ser puesto en práctica por completo genera valor inmediato para los propietarios y usuarios del sistema. Se desarrolla ese subconjunto, de preferencia en seis a nueve meses o menos. En forma subsiguiente, se desarrollan versiones del sistema con valor añadido, en marcos cronológicos similares, 130*

**tipo de datos** *Clase de datos que se puede almacenar en un atributo, 275*

comprobación de validación, 492  
 físicos, 432

**tipo de datos** *Propiedad de un atributo que identifica el tipo de datos que se puede guardar en el atributo, 214*

Herramienta de instrucciones (Tool tips), 533-534

TPS; véase **sistema de procesamiento de transacciones**

TQM; véase **administración de calidad total**

**trabajador de la información** *Toda persona cuyo trabajo entrañe la creación, recopilación, procesamiento, distribución y uso de información, 7*

**trabajador del conocimiento** *Todo trabajador cuyas responsabilidades se centran en un área especializada de conocimientos, 9*

**muestreo de trabajo** *Técnica de identificación de hechos que requiere numerosas observaciones tomadas a intervalos aleatorios, 162*

Transacciones; véase **evento**

Transformación; véase **proceso**

Transmisión electromagnética, 489

**trigger** *Programa incorporado en una tabla que se invoca automáticamente por las actualizaciones de otra tabla, 428*

## U

U.S. Department of Labor (Departamento de Trabajo de Estados Unidos), 7, 13

MIL-STD-498, 155

UNIX, 520

UPS, 396, 486

Uris, Auren, 339

**unidad de diseño** *Un conjunto independiente de procesos, almacenes de datos y flujos de datos que comparten atributos similares de diseño, 402, 405-406, 464*

**usuario del sistema** *"Cliente" que usa con regularidad un sistema de información o se ve afectado por él; capturando, validando, introduciendo, respondiendo, almacenando e intercambiando datos e información, 7-8, 9*

componentes de comunicaciones y, 35-37

componentes de conocimientos y, 30-31

componentes de procesos y, 32-34

externo, 9-10

interno, 9

perspectiva del sistema de información, 26-27

realimentación sobre el diseño de la interfaz de usuario, 471-472, 503-505, 541

tipos, 516-517

**usuario experto** *Un usuario de computadoras experimentado, 517*

**usuario móvil** *Usuario cuya ubicación cambia constantemente pero que requiere acceso a los sistemas de información desde cualquier lugar, 10*

**usuario novato** *Usuario de computadoras inexperto u ocasional, 517*

**usuario remoto** *Usuario que no se ubica físicamente en las instalaciones donde se encuentran los sistemas de información pero que necesita y tiene acceso a ellos, 10*

Usuarios; véase **usuario del sistema**

Usuarios externos, 9-10

Usuarios internos, 9

## V

Valor del dinero en el tiempo, 320-321

**valor presente** *El valor presente de una unidad monetaria en un momento futuro, 322*

Verificaciones de existencias, 492

**valor presente neto** *Técnica de análisis que compara los costos y beneficios anuales descontados de diferentes alternativas, 323-324*

Verificación del tipo de dato, 492

videos de avance (en inglés video trailers), 475

Visible Systems, Visible Analyst, 88, 104, 345

Visio Enterprise, 426

Visio Professional, 104, 345

Visual Basic, 500, 536

Visual Basic .NET, 34, 77, 106, 388

Visual Studio .NET, 34, 38, 54, 89, 399, 492, 516, 536

Vitalari, Nicholas P., 57n, 309

**valor por omisión** *El que se registra si el usuario no especifica otro valor, 215, 432*



**W**

Walton, Donald, 169n, 183, 339  
Watson, Mark, 308  
navegadores web; *véase también* **Internet**  
  barras de herramientas, 526  
  como interfaz de usuario, 389-392, 486, 492, 527-528  
  independencia de plataformas, 520  
Web (la red de redes o la web), 460  
Weber, Charles V., 97  
Weinberg Gerald M., 22, 147, 150n, 160, 182, 183  
Weinschenk, Susan, 547  
Wetherbe, James, 57, 97, 147, 309, 339  
White, E. B., 331  
Windows Mobile, 396, 520  
Windows  
  controles de entradas avanzados, 498-500

  diálogos de usuario, 522  
  interfaz de usuario, 486, 492  
  interfaz de usuario orientada a objetos, 77  
  predominio en el mercado, 520  
Wood, Jane, 147, 183, 371

**X**

XML (Extensible Markup Language), 38, 385, 392, 400

**Y**

Yeo, Sarah C., 547  
Yourdon, Edward, 147, 266, 277, 309, 370, 371

**Z**

Zachman, John A., 38-39, 44-45, 147, 371



