

▼ La evolución de la tecnología de bases de datos

La sofisticación de la tecnología moderna de las bases de datos es el resultado de la evolución que a lo largo de varias décadas ha tenido lugar en el procesamiento de los datos y la gestión de la información. La tecnología de acceso a los datos se ha desarrollado desde los métodos primitivos de los años cincuenta hasta los potentes e integrados sistemas de hoy en día, arrastrados de un lado por las necesidades y las demandas de la administración y de otro, restringida por las limitaciones de la tecnología.

Las expectativas de la administración han crecido paralelamente a la evolución de la tecnología. Los primeros sistemas de procesamiento de datos ejecutaron las tareas administrativas para reducir el papeleo. Más recientemente, los sistemas se han expandido hacia la producción y la gestión de la información, la que se ha convertido en un recurso vital para las compañías. Actualmente, la función más importante de los sistemas de bases de datos consiste en proporcionar el fundamento a los sistemas de información para la gestión corporativa.

La implantación de los cambios tecnológicos ha estado guiada por las necesidades genuinas de los negocios. La administración únicamente autorizará un nuevo sistema informático cuando vea un beneficio claro al modificar el costo del sistema. Y a pesar de las dificultades y los riesgos, en muchos casos se han experimentado ventajas. Además, el final no está aún a la vista y tardará tiempo en llegar. Las nuevas tecnologías, tales como las bases de datos orientadas a objetos y la plataforma cliente/servidor, plantean nuevos problemas y darán lugar a sistemas más potentes para el futuro.

La estrecha interrelación que existe entre la tecnología de las bases de datos y las necesidades empresariales puede ser más fácil de comprender si nos acercamos a la experiencia de la compañía International Product Distribution.

▼ Un caso: La compañía International Product Distribution

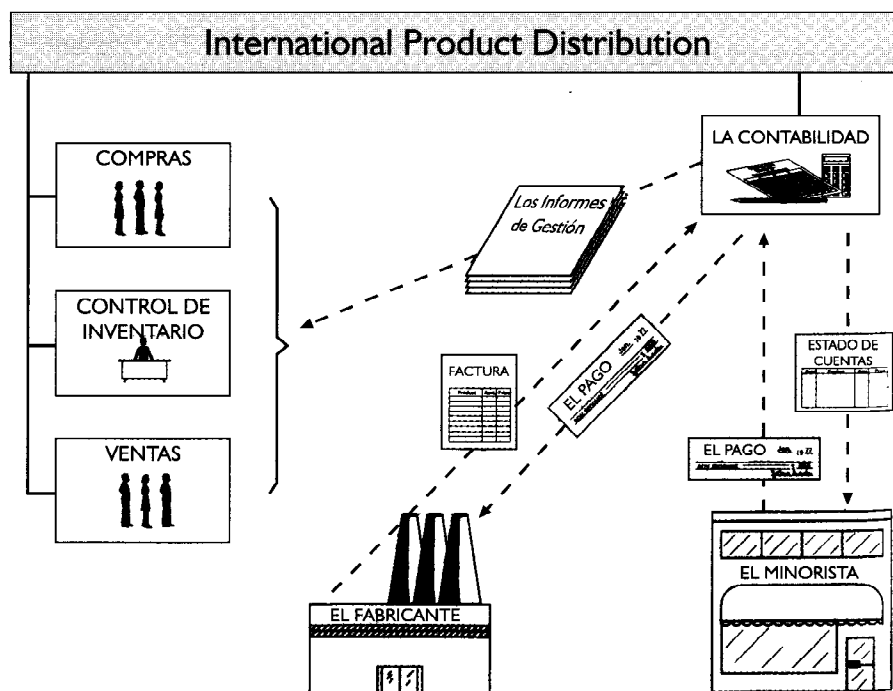
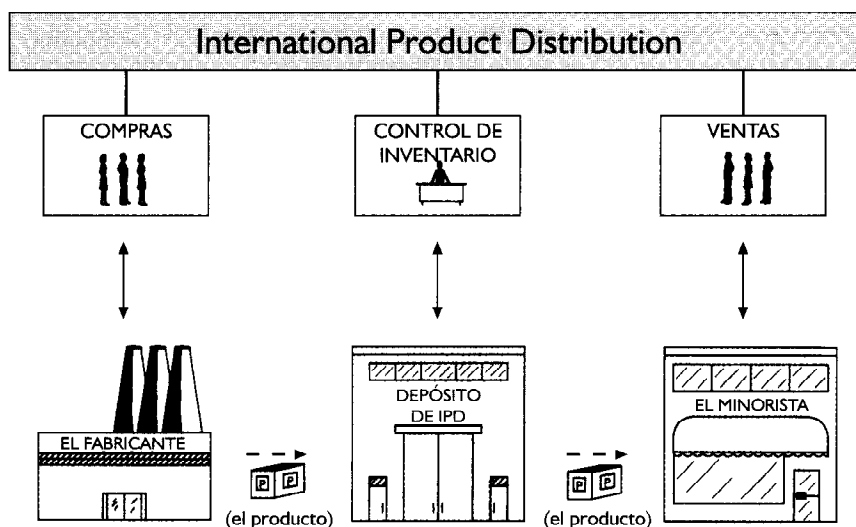


Susan Broadbent es la fundadora, propietaria y presidenta de International Product Distribution (IPD), que vende por encima de 3.500 productos de más de 300 fabricantes en países de todo el mundo. La IPD tiene su sede en Chicago, con oficinas internacionales en Bruselas, Buenos Aires, Lagos, Nueva Delhi, Tokio y Sydney. Alrededor de 2.700 representantes de ventas trabajan localmente en más de 100 países, cada uno de los cuales informa a una oficina regional. La compañía tiene unas ventas anuales de alrededor de 500 millones de dólares y unos beneficios en torno a 50 millones de dólares.

Después de vender ropa de niños durante un cierto número de años producida por un mismo fabricante a los grandes almacenes de Chicago, Susan decidió que podría aumentar significativamente sus ingresos si representaba a varios fabricantes. Así, fundó International Product Distribution. Su concepto era sencillo. (1) Identificar los fabricantes en diversos países cuyos productos mostraran constantemente una calidad reconocida. (2) Identificar a los minoristas que apoyaran la venta de tales productos. (3) Establecer fuertes relaciones comerciales, tanto con los fabricantes como con los minoristas, proporcionando a estos últimos los productos de los fabricantes apropiados.

Inicialmente, contaba con muy poco personal y se relacionaba únicamente con minoristas de Chicago y fabricantes del medio oeste. Sin embargo, en poco tiempo estuvo abasteciendo productos a los comerciantes desde St. Louis hasta Cleveland. Sus primeras ventas internacionales las hizo a unos almacenes en Toronto. Después de tres años, la compañía IPD tuvo representantes en Europa y dos años más tarde en Tokio. Seguidamente se abrieron oficinas en Buenos Aires, Sydney, Lagos y Nueva Delhi. Cada oficina empleaba compradores y representantes de ventas. Los productos comprados se vendían en el país de fabricación o se exportaban para su venta en otro país. La Figura 1.1 ilustra la interrelación existente entre la compañía IPD y sus clientes, tanto los minoristas como los suministradores. Como se puede ver, los productos circulan desde el fabricante a los almacenes de IPD y desde ahí al minorista.

Al principio, los registros de las ventas, de las compras de los productos y el inventario de la compañía se efectuaban a mano. No obstante, al finalizar el segundo año, el volumen del negocio se había extendido tanto que fue necesario comprar un minicomputador para almacenar esta información y confeccionar informes, facturas y pagos, tal y como se muestra en la Figura 1.2. Sanford Mallon fue contratado para desarrollar este sistema orientado a archivos y para dirigir un grupo formado por programadores, operadores para la entrada de datos y personal de operaciones.



▼ Sistemas orientados a los archivos

Los sistemas computacionales se utilizaron inicialmente en los negocios para funciones de contabilidad: las cuentas a cobrar, las cuentas a pagar, la nómina y otras. Estas funciones eran imprescindibles para que el negocio funcionara. Por consiguiente, el costo de los computadores que podían ejecutar estas funciones era fácil de justificar. Por ejemplo, el

sistema de procesamiento de datos. Un sistema automatizado para procesar los datos de los registros de una organización.

esfuerzo manual requerido para la nómina o las cuentas a cobrar era tan grande que, si un sistema automatizado pudiera reemplazar al sistema manual, pagaría por sí solo la inversión en corto tiempo.

Estos sistemas se llamaron **sistemas de procesamiento de datos** debido a que ejecutaban las funciones habituales de tratamiento de los registros. Sin que esto pueda sorprender a los programadores y a los analistas que diseñaron estos sistemas, se dejaron influir durante su programación por la inclinación natural de imitar los procedimientos manuales existentes. Así, los archivos en el computador se correspondían con los archivos de papel y los registros en los archivos del computador contenían la información que podía almacenar una carpeta individual de un archivo en un sistema manual.

La Figura 1.3 muestra algunos archivos y datos de prueba del sistema original orientado a archivos de la compañía International Product Distribution. Cada tabla representa un archivo del sistema. Es decir, se tiene un archivo **CLIENTE**, un archivo **REPRESENTANTE DE VENTAS**, un archivo **PRODUCTO** y así sucesivamente. Cada fila representa un registro en el archivo. De esta manera, el archivo **PRODUCTO** contiene tres registros. Cada uno de estos registros contiene los datos sobre un producto diferente. Los conceptos o campos de los datos individuales en el archivo **PRODUCTO** son **ID_PRODUCTO**, **DESC_PRODUCTO**, **ID_FABRICANTE**, **COSTO** y **PRECIO**¹.

| CLIENTE | | | | PAGOS | |
|-------------------------|---------------------|------------------|---------------|-----------------|--------------|
| ID_CLIENTE | NOMB_CLIENTE | DIRECCIÓN | PAÍS | BALANCE INICIAL | HASTA EL MES |
| 100 | Hnos. Watabe | Apdo. 241, Tokio | Japón | 45.551 | 40.113 |
| 101 | Matlzl | Salzburg | Austria | 75.314 | 65.200 |
| 105 | Jefferson | B 918, Chicago | EUA | 49.333 | 49.811 |
| 110 | Gómez | Santiago | Chile | 27.400 | 28.414 |
| REPRESENTANTE DE VENTAS | | | | | |
| ID_REP | NOMB_REP | ID_DIRECTOR | OFICINA | COMM_% | |
| 10 | Rodney Jones | 27 | Chicago | 10 | |
| 14 | Masaji Matsu | 44 | Tokio | 11 | |
| 23 | François Moire | 35 | Bruselas | 19 | |
| 37 | Elena Hermana | 12 | B.A. | 13 | |
| 39 | Goro Azuma | 44 | Tokio | 10 | |
| PRODUCTO | | | | | |
| ID_PRODUCTO | DESC_PRODUCTO | ID_FABRICANTE | COSTO | PRECIO | |
| 1035 | Suéter | 210 | 11.25 | 22.00 | |
| 2241 | Lámpara de Mesa | 317 | 22.25 | 33.25 | |
| 2518 | Escultura de Bronce | 253 | 13.60 | 21.20 | |
| VENTA | | | | | |
| FECHA | ID_CLIENTE | ID_REP | ID_PRODUCTO | CANTIDAD | PRECIO TOTAL |
| 02/08 | 100 | 14 | 2241 | 200 | 6650,00 |
| 02/12 | 101 | 23 | 2518 | 300 | 6360,00 |
| 02/12 | 101 | 23 | 1035 | 150 | 3300,00 |
| 02/19 | 100 | 39 | 2518 | 200 | 4240,00 |
| 02/22 | 101 | 23 | 1035 | 200 | 4400,00 |
| 02/25 | 105 | 10 | 2241 | 100 | 3325,00 |
| 02/25 | 110 | 37 | 2518 | 150 | 3180,00 |
| FABRICANTE | | | | | |
| ID_FABRICANTE | NOMB_FABRICANTE | DIRECCIÓN | PAÍS | | |
| 210 | Confecciones Kiwi | Aukland | Nueva Zelanda | | |
| 253 | Obras de Bronce | Lagos | Nigeria | | |
| 317 | Lámparas Llana | Lima | Perú | | |

¹ Para respetar el sentido del humor del autor, los nombres de personas, organizaciones, lugares y direcciones, utilizados en los ejemplos a lo largo de todo el libro, se han mantenido igual que en el original (N. del T.).

Por el momento, supongamos que se tiene acceso a todos estos archivos secuencialmente. Es decir, que cada registro puede leerse y procesarse únicamente después que todos los registros que lo preceden en el archivo hubieran sido leídos. Eso era lo que ocurría en la compañía IPD en los años sesenta, cuando el almacenamiento en disco era todavía relativamente caro. La mayoría de los archivos se almacenaban en cinta magnética y se tenía acceso y se procesaban los registros en secuencia. Por lo general, estos archivos se procesaban en lotes, lo que significaba que todos los registros de un archivo se procesaban al mismo tiempo, normalmente cada noche después del cierre del negocio.

Los archivos se utilizaban para varias aplicaciones diferentes. Por ejemplo, el programa de las cuentas a cobrar generaba los estados de cuentas para los clientes. Utilizaba los archivos CLIENTE y VENTA. Ambos archivos se ordenaban según el campo ID_CLIENTE y se fusionaban para crear un estado impreso, como se muestra en la Figura 1.4. El campo SALDO_INICIAL en el archivo CLIENTE se actualizaba para reflejar los nuevos cargos. Los pagos, que se recibían y procesaban previamente por otro programa, teniendo en cuenta el archivo CLIENTE, se registraban en el campo PAGOS_HASTA_MES_ACTUAL del archivo CLIENTE y se mostraban en el estado de cuentas.

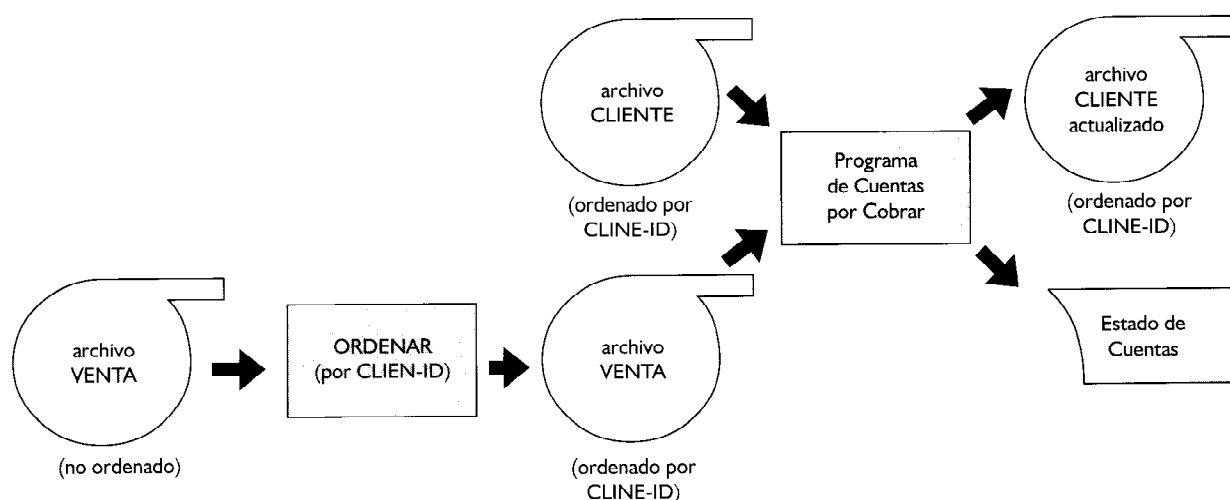
programa de aplicación

Un programa como éste que realiza una tarea específica de valor práctico en una empresa es un **programa de aplicación** o un elemento de un *software de aplicación*. El conjunto de programas que trabajan colectivamente para realizar un grupo de tareas relacionadas entre sí es un *sistema de aplicación*.

Necesidad del proceso de acceso directo a datos

acceso directo a los datos

Las limitaciones de los sistemas orientados a archivos puramente secuenciales no los privaron de ser herramientas eficaces para producir pagos, facturas y otros informes una o dos veces al mes. Sin embargo, para ejecutar muchas tareas rutinarias en los negocios se necesita el **acceso directo a los datos** —la capacidad de tener acceso y procesar directamente un registro dado sin ordenar primero el archivo o leer los registros en secuencia. Para comprender algunos de los problemas que ocurren cuando *no* existe el acceso directo a los datos, se verán dos ejemplos del pasado en IPD.





“Sandy, no comprendo por qué nuestros operadores de entrada de datos tienen que introducir tanto la cantidad como el precio total siempre que introducen una transacción de venta. Nosotros tenemos el precio del producto en el archivo PRODUCTO. ¿Por qué nuestra gente no puede introducir solamente la cantidad y el ID del producto, que en cualquier caso introducen siempre, y que el sistema calcule el precio total?”

Estamos en la época inicial del procesamiento de datos en la compañía IPD y Susan Broadbent está preocupada por la cantidad de trabajo innecesario que hacen los operadores. Cuantos más datos tienen que introducir, más caro resulta. Su pregunta es natural, teniendo en cuenta que la compañía IPD ha invertido ya una considerable suma de dinero para comprar una computadora, desarrollar un paquete de programas y pagar al personal que lo realiza.

“Susan, nosotros tenemos un sistema de archivos secuenciales. Esto significa que el acceso a todos los registros en un archivo se tiene que realizar en orden. Cuando ejecutamos el programa de cuentas a cobrar, trabajamos con los archivos CLIENTE y VENTA, por tanto, no tenemos acceso al registro del producto.”

“Me temo que no comprendo lo que estás diciendo.”

“Bien, aquí hay algunos datos de prueba de los archivos. (Véase Figura 1.3.) Observa cómo el archivo CLIENTE está ordenado según el ID del Cliente. Antes de ejecutar el programa de las cuentas a cobrar, también ordenamos el archivo VENTA según el ID del Cliente. Entonces, cuando ejecutamos las cuentas a cobrar, leemos ambos archivos. Cuando los IDs del Cliente coinciden en ambos archivos, actualizamos el saldo inicial en el archivo CLIENTE e imprimimos una factura. En la factura listamos los pagos hasta el mes actual y detallamos todos los encargos de ese cliente, que se muestran en el archivo VENTA. Hacemos eso manteniendo el mismo registro de CLIENTE mientras leemos, sucesivamente, cada uno de los registros de VENTA que le corresponden. Observa cómo, en este ejemplo, el archivo VENTA muestra dos encargos del cliente 100, tres encargos del cliente 101 y un encargo de cada uno de los clientes 105 y 110. Como resultado, hemos minimizado la cantidad de tiempo empleada en la lectura de los datos de la cinta magnética, que es la parte más lenta de la ejecución del programa.”

“Ahora mira qué sucedería si el programa tuviera que calcular el precio total. La primera venta en el archivo tiene el ID de Producto 2241. Fíjate que éste se encuentra en la mitad del archivo PRODUCTO. Por tanto, para obtener el precio de ese producto, tendríamos que leer hasta la mitad del archivo PRODUCTO. El registro de venta siguiente corresponde al producto 2518, que es el próximo registro en el archivo PRODUCTO. Pero después, el registro de VENTA que sigue corresponde al producto que está al comienzo del archivo PRODUCTO. Para obtener el precio de ese producto, estaríamos obligados a regresar al comienzo del archivo PRODUCTO. ¿Ves cómo tendríamos que estar saltando continuamente a través del archivo PRODUCTO, leyendo hacia adelante un gran número de registros, rebobinando la cinta después y así sucesivamente?”

“Sí. Un gran consumo de tiempo,” contestó Susan.

“Correcto. A largo plazo es más barato tener a nuestros operadores introduciendo los datos del precio total.”

“Yo no estoy tan segura de que sea más barato, Sandy. Los representantes de ventas frecuentemente calculan mal el precio, entonces tenemos importes incorrectos en las facturas, clientes descontentos y una disminución de los beneficios. No podemos vivir con este problema. Necesitamos una solución.”

“La mejor solución sería cambiar los archivos CLIENTE, PRODUCTO y otros más, a archivos de acceso directo (secuencial indexado). Por ejemplo, utilizaríamos el ID del Producto como la clave del archivo de los productos. Entonces, podríamos cambiar nuestro programa de cuentas a cobrar, de modo que tenga acceso a cualquier registro de ese archivo en el momento que queramos. Lo único que necesitamos saber es el ID del Producto correspondiente al registro que deseamos.”

“¿Esto nos daría alguna otra ventaja?”

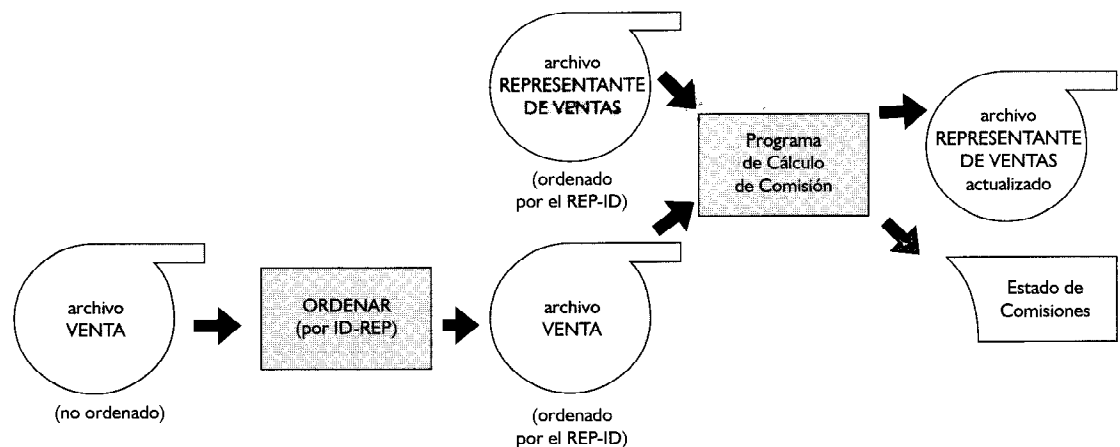
“Por supuesto. Por ejemplo, puedes obtener el saldo actualizado de los clientes siempre que lo necesites. También podemos utilizar los archivos de acceso directo para validar los ID de los Productos, los ID de los Fabricantes, los ID de los Representantes de Ventas y otros. Eso nos ayudará a eliminar los errores en las facturas de los clientes, que están provocando sus quejas.”

“Eso suena fabuloso. ¿Cuándo lo podremos lograr?”

Volviendo hacia atrás por un momento se pueden ver más de cerca los problemas con los sistemas de archivos secuenciales. Para que se procesen las ventas contra el archivo CLIENTE en el programa de cuentas a cobrar, las ventas deben estar ordenadas por el ID del Cliente. Como es muy probable que las ventas se introduzcan aleatoriamente, el archivo VENTA debe ordenarse antes de que pueda utilizarse como entrada a este programa. Esto se muestra en la Figura 1.4. Aun cuando el archivo VENTA esté ordenado por el ID del Cliente, probablemente estará desordenado por completo con respecto al ID del Producto. Por esta razón no se puede calcular el precio total de una venta a partir del precio en el archivo PRODUCTO. Esto conduce a que los operadores introduzcan datos redundantes, lo que requiere un esfuerzo manual adicional e introduce una probabilidad mayor de error.

El requisito de que todos los archivos se procesen secuencialmente conduce también, de otra forma, a realizar trabajo adicional. Por ejemplo, otra de las aplicaciones de la compañía IPD calcula las comisiones por ventas y genera un informe de comisiones. Para calcular la comisión de un representante de ventas, hay que ordenar el archivo VENTA nuevamente, esta vez para ponerlo en orden según ID_REP. Ahora se puede procesar el archivo VENTA contra el archivo REPRESENTANTE_DE_VENTAS para crear los informes de comisiones correspondientes a los representantes de ventas (Figura 1.5).

Este ejemplo ilustra algunas de las limitaciones más serias de los sistemas de archivos puramente secuenciales. Estos problemas se resolvieron parcialmente con la introducción de los archivos de acceso directo y, particularmente, de los archivos secuenciales indexados (ISAM)², que se utilizaron ampliamente en los años sesenta. Los archivos de acceso directo permiten la recuperación de los registros aleatoriamente, a diferencia de los de acceso secuencial. El registro deseado en un archivo de acceso directo puede recuperarse inmediatamente. Los archivos ISAM son los archivos de acceso directo más utilizados en



² ISAM - Indexed Sequential Access Method (N. del T.).

clave. Campos de datos que identifican unívocamente un registro en un archivo.

procesos de tipo comercial. Estos archivos permiten que uno o más campos de datos —llamados conjuntamente **clave**— se utilicen para identificar precisamente qué registro se recuperará. Los archivos ISAM proporcionaron un medio potente y práctico para dotar de gran flexibilidad a las aplicaciones comerciales. (En el Capítulo 10 se discuten más detalladamente los aspectos físicos de los sistemas de bases de datos.)

Sin embargo, los archivos de acceso directo solamente proporcionaron una solución parcial. Para lograr una solución más completa a estos problemas fue necesario introducir los sistemas de gestión de bases de datos.

La información como un recurso

sistema de información. Sistema automatizado que organiza los datos para producir información.

En el diálogo anterior, Sanford sugirió la posibilidad de buscar el saldo de un cliente siempre que se necesitase. La pregunta “¿Cuál es el balance actual de los hermanos Watabe?” es una pregunta al **sistema de información**. Tales preguntas ilustran la transición tan significativa que tuvo lugar a finales de los años sesenta y principios de los setenta en el sentido de que los sistemas computacionales comerciales pasaron del *procesamiento de los datos al procesamiento de la información*. Este cambio reflejó una conciencia creciente con respecto a que la información era mucho más que simples registros relacionados con el negocio. Gradualmente, en las empresas comenzaron a darse cuenta del valor de la información y del enorme potencial que los sistemas computacionales representaban para organizar y administrar este recurso, recién reconocido como tal. A finales de los años sesenta, esto condujo a una demanda muy fuerte de **sistemas de información para la gestión**. Estos sistemas utilizarían los datos ya disponibles en el computador para brindar respuesta a un amplio espectro de preguntas de gestión o administración.

sistema de información para la gestión. Sistema que provee información para la gestión o la administración.

En este contexto se hace una distinción entre *datos e información*. Comúnmente, los **datos** se consideran como hechos aislados. Por ejemplo:

datos. Hechos aislados.

“Hnos. Watabe es una compañía ubicada en Japón”

información. Datos organizados o resumidos.

es un hecho. Este hecho está contenido en un único registro en el archivo CLIENTE. Los archivos de un sistema contienen millares de tales hechos. Los archivos, por lo tanto, contienen datos. Por otra parte, la **información** corresponde a los *datos procesados*. En este sentido, se quiere decir que la información se puede ver como los datos organizados o resumidos. Por ejemplo, se podría querer conocer el saldo actual total de la compañía Hnos. Watabe o, quizás, se podría pedir el saldo actual promedio de todos los clientes en Europa. Las respuestas a tales preguntas serían informaciones.

Por supuesto, cada hecho o elemento de los datos puede recibir el nombre de información. Pero se hace referencia, en primer lugar, a la información que sería valiosa para los distintos niveles de dirección de una empresa, incluso a nivel ejecutivo —especialmente para la toma de decisiones—. Esta información es normalmente de alto nivel, resumida a partir de un conjunto mucho más grande de hechos. Por consiguiente, la información es diferente de los datos.

En los años más recientes, el impacto significativo que ha tenido la información sobre la planificación y la toma de decisiones en las organizaciones ha conducido a un reconocimiento siempre creciente de que la información es un recurso que tiene valor y, por lo tanto, necesita estar organizada y administrada. Aun cuando en las empresas se acostumbra a trabajar con activos tangibles, tales como el dinero, las instalaciones y el personal, cuyo valor puede evaluarse con cierta precisión, ha sido muy difícil de medir el valor de la información. Sin embargo, está claro que si los directivos tienen buena información, es más probable que puedan tomar decisiones pertinentes y certeras con un mayor impacto positivo en su negocio. Y viceversa, si su información es pobre, ellos deben trabajar con más incertidumbre y es menos probable que tomen decisiones convenientes. El desarrollo de los sistemas de bases de datos se convirtió en crucial para proporcionar información correcta y oportuna a los directivos.

base de datos. Una colección de datos interrelacionados que se puede utilizar por uno o más programas de aplicación.

sistema de base de datos. Una base de datos, un sistema de gestión de bases de datos, con el *hardware* y el personal apropiados.

sistema de gestión de bases de datos (SGBD). Un sistema computacional que facilita la gestión de las bases de datos.

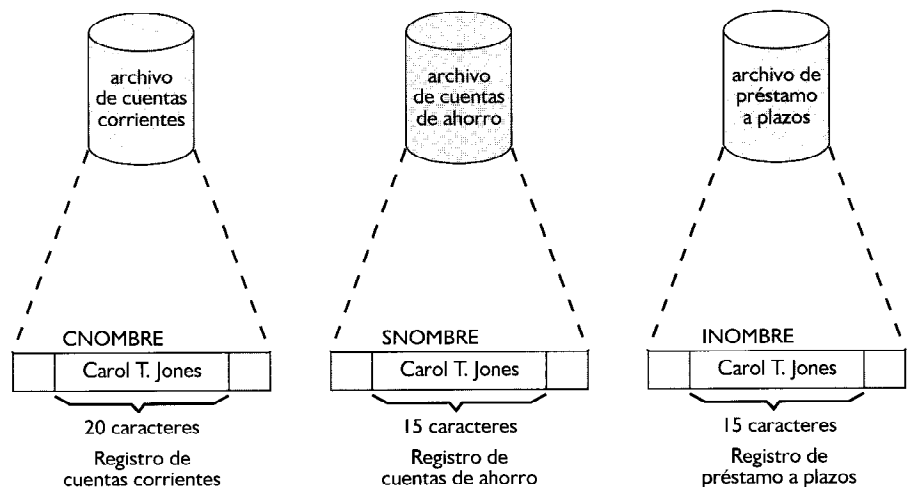
Una **base de datos** es una colección de elementos de datos interrelacionados que pueden procesarse por uno o más sistemas de aplicación. Un **sistema de base de datos** está formado por una base de datos, por un sistema computacional de propósito general —llamado **sistema de gestión de bases de datos (SGBD)**— que manipula la base de datos, así como por el *hardware* y el personal apropiados. Un SGBD normalmente se compra a un vendedor de sistemas computacionales y es el medio con el cual un programa de aplicación o un usuario final examina y manipula los datos almacenados en la base de datos. Al finalizar este capítulo se estudiarán los componentes de un sistema de base de datos con más detalle.

Un sistema de base de datos, adecuadamente diseñado, integra los datos comunes a varias unidades funcionales de la compañía y facilita su manipulación. Además de simplificar la inserción, la eliminación y la modificación cotidianas de los registros, los sistemas de base de datos facilitan la identificación y la cuantificación de las relaciones derivadas entre los elementos de los datos, la recopilación de la información en resúmenes estadísticos, la inferencia sobre las posibles tendencias del negocio y otras operaciones. Mediante tales facilidades, el sistema de base de datos transforma los datos puros en información.

Otras limitaciones de los sistemas de archivo tradicionales

A pesar de la introducción de los archivos de acceso directo, pronto se hizo obvio que a los sistemas de archivo de cualquier tipo era inherente un conjunto de deficiencias: (1) redundancia de los datos; (2) pobre control de los datos; (3) capacidades inadecuadas de manipulación de los datos, y (4) esfuerzo excesivo de programación.

Redundancia de los Datos. Una dificultad importante era que muchas aplicaciones utilizaban sus propios archivos especiales de datos. Así, algunos elementos de los datos eran comunes a varias aplicaciones. En un banco, por ejemplo, el mismo nombre de cliente podría aparecer en un archivo de cuentas corrientes, en un archivo de cuentas de ahorros y en un archivo de préstamos a plazos (Figura 1.6). Además, si bien el nombre del cliente era siempre el mismo, el campo relacionado con él frecuentemente tenía un nombre diferente en los diversos archivos de cuentas. Así, CNOMBRE en el archivo de cuentas corrientes era SNOMBRE en el archivo de cuentas de ahorros e INOMBRE en el archivo de préstamos a plazos. El mismo campo podría también tener una longitud diferente en los diversos archivos. Por ejemplo, CNOMBRE podría llegar a tener hasta 20 caracteres, pero SNOMBRE e INOMBRE podrían limitarse a 15 caracteres. Esta redundancia aumen-



que pue-
está for-
llamado
si como
vende-
ón o un
zar este
lle.

numes a
simpli-
istemas
privadas
estadísti-
mediante
nación.

ue a los
redun-
e mani-

es uti-
os eran
cliente
horros
cliente
e dife-
tas co-
ivo de
en los
eteres,
umen-

tó los gastos de administración para el mantenimiento y el almacenamiento, así como el riesgo de inconsistencia entre las diversas versiones de los datos comunes.

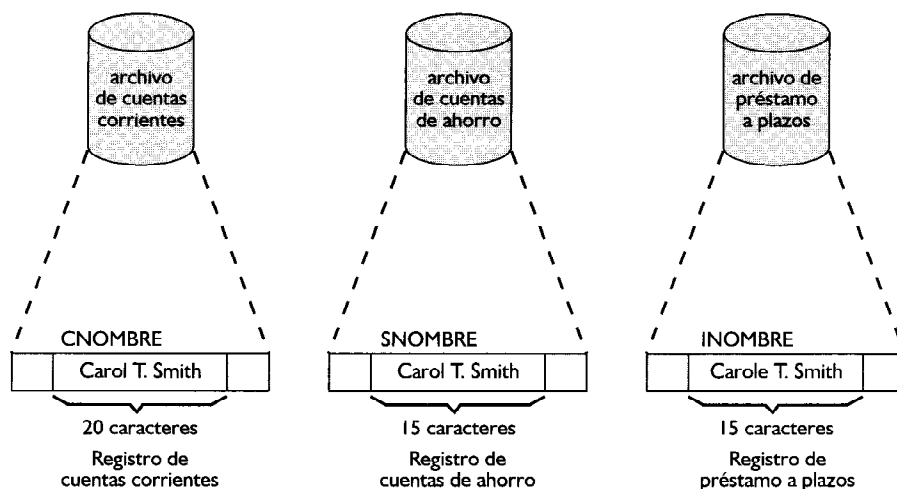
Supóngase que el nombre de un cliente se modificó de Carol T. Jones a Carol T. Smith. El campo del nombre del cliente podría actualizarse inmediatamente en el archivo de cuentas corrientes, actualizarse la semana siguiente en el archivo de cuentas de ahorros y actualizarse incorrectamente en el archivo de préstamos a plazos (Carole T. Smith) (Figura 1.7). Con el tiempo, tales diferencias pueden ocasionar una degradación seria en la calidad de la información contenida en los archivos de los datos. Estos tipos de inconsistencia de los datos pueden afectar también la exactitud de los informes. Suponga que se desea producir un informe para la gerencia mostrando todos los clientes que tienen una cuenta para cheques o una cuenta de ahorros y un préstamo a plazos. Carol T. Smith se omitiría erróneamente en el informe porque su nombre aparece como Carole T. Smith en el archivo de préstamos a plazos. Como se mostrará, los sistemas de base de datos pueden eliminar la redundancia de los datos, ya que todas las aplicaciones comparten un fondo común de datos. La información esencial como el nombre o la dirección del cliente aparecerá solamente una vez en la base de datos. De este modo, se puede introducir el cambio de un nombre o de una dirección una sola vez y saber que las aplicaciones tendrán acceso a datos consistentes.

Pobre control de los datos. En los sistemas de archivos, como se señaló anteriormente, no había control centralizado al nivel de los elementos de los datos. El mismo elemento de los datos solía tener varios nombres, dependiendo del archivo en que estuviera contenido.

En un nivel más básico, siempre existe la posibilidad de que los diversos departamentos de una compañía sean inconsistentes en su terminología. Un banco, por ejemplo, puede utilizar el término cuenta para significar una cosa cuando se aplica a los ahorros y algo bastante diferente cuando se aplica a los préstamos. El término que tiene significados diferentes en contextos diferentes se llama un **homónimo**. Por el contrario, las palabras diferentes pueden significar la misma cosa. Una compañía de seguros puede referirse a una *póliza* y a un *caso* y puede querer significar lo mismo con ambas palabras. Dos términos que significan la misma cosa se llaman **sinónimos**. Un sistema de base de datos soporta el control centralizado de los datos y contribuye a eliminar la confusión ocasionada por los homónimos y por los sinónimos.

homónimo. Un término que tiene diferentes significados en diferentes contextos.

sinónimo. Término que significa lo mismo.



Capacidades inadecuadas de manipulación de los datos. Los archivos secuenciales indexados permitieron que las aplicaciones tuvieran acceso a un registro particular mediante una clave, como el ID del Producto. Por ejemplo, si se conocía el ID del Producto para las lámparas de mesa, se podía tener acceso directamente al registro que le corresponde dentro del archivo PRODUCTO. Esto fue suficiente mientras solamente se quiso un registro único.

Sin embargo, supongamos que se quiso un conjunto de registros interrelacionados. Por ejemplo, se podía estar interesado en identificar todas las ventas de la compañía IPD al cliente Maltzl. Quizás se necesitaba conocer el número total de ventas, o el precio promedio, o los productos que estaban siendo comprados y de cuáles fabricantes. Dicha información sería difícil, si no imposible, de obtener con un sistema de archivos porque estos sistemas son incapaces de proveer conexiones fuertes entre los datos contenidos en archivos diferentes. Los sistemas de base de datos se desarrollaron específicamente para facilitar la interrelación de los datos en archivos diferentes.

Esfuerzo de programación excesivo. Un nuevo programa de aplicación con frecuencia requería un conjunto completamente nuevo de definiciones de los archivos. Aunque un archivo existente podía contener algunos de los datos necesarios, la aplicación frecuentemente requería otros elementos de los datos. Como resultado, el programador tenía que recodificar todas las definiciones de los elementos de los datos necesarios ya existentes en el archivo, así como todos los elementos de los datos nuevos. De esta manera, en los sistemas orientados a archivos había una interdependencia muy fuerte entre los programas y los datos.

Aún más importante, la manipulación de los datos en los lenguajes orientados a archivos, como el COBOL, era difícil para las aplicaciones complejas. Esto significó que ambos esfuerzos en la programación de aplicaciones de gestión, tanto inicial como de mantenimiento, fueran significativos.

Las bases de datos brindan una separación entre el programa y los datos, de modo que los programas pueden ser, en cierta medida, independientes de los detalles de definición de los datos. Al garantizar el acceso a un fondo común de datos compartidos y al soportar lenguajes poderosos para la manipulación de los datos, los sistemas de base de datos eliminan una gran cantidad de programación inicial y de mantenimiento.

▼ Los sistemas de bases de datos

Los sistemas de bases de datos superan estas limitaciones de los sistemas orientados a los archivos. Al tolerar una estructura de datos centralizada, integrada, los sistemas de base de datos eliminan los problemas de redundancia y de control de los datos. Si una base de datos centralizada está disponible para toda la compañía y, por ejemplo, tuvo que cambiarse el nombre de un cliente, dicho cambio está disponible para todos los usuarios. Los datos se controlan por medio de un diccionario de datos/directorio (DD/D) de los datos, que en sí mismo está controlado por un grupo de empleados de la compañía, conocidos como los administradores de la base de datos (ABDs)³. Los nuevos métodos de acceso a los datos simplifican mucho el proceso de relacionar los elementos de los datos, a la vez que mejoran la manipulación de los datos. Todas estas características de los sistemas de base de datos simplifican el esfuerzo de programación y mantenimiento de los programas.

En la actualidad, estamos inmersos en varias décadas de largo esfuerzo por desarrollar sistemas de gestión de bases de datos cada vez más poderosos. Este proceso ha sido testigo del desarrollo evolutivo de los sistemas basados en tres **modelos de datos** fundamentales, y que no son más que métodos conceptuales para estructurar los datos. Estos tres modelos de datos son el jerárquico, en red y el relacional. En las dos secciones siguientes se esbozará el desarrollo histórico de los sistemas de base de datos, apoyado en estos tres modelos.

modelo de datos. Un método conceptual para estructurar los datos.

³ En inglés DBAs - *DataBase Administrators* (N. del T.).

Sistemas de los modelos jerárquico y en red

Los archivos secuenciales indexados resolvieron el problema del acceso directo a un registro único dentro de un archivo. Por ejemplo, al revisar una vez más la Figura 1.3. Si tuviéramos el registro de la primera venta mostrado en el archivo VENTA, pero se quisiera conocer el nombre y la dirección del cliente involucrado en la venta, se podría utilizar simplemente el ID del Cliente (100) para buscar el registro del cliente en el archivo CLIENTE. Esto informa que el cliente que emitió la orden fue Watabe Bros.

Ahora supongamos que se quiere invertir el proceso. En lugar de querer conocer el cliente involucrado en una venta, se quieren conocer todas las ventas realizadas a un cliente determinado. Se comienza a partir del registro de cliente correspondiente a Watabe Bros, y ahora se quieren todos sus registros de venta. Esto no se puede hacer directamente en un sistema de archivo. Precisamente, para tales aplicaciones se desarrollaron originalmente los sistemas de base de datos

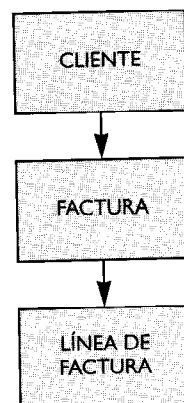
Los primeros sistemas de base de datos, introducidos a mediados de los sesenta, estaban basados en el **modelo jerárquico**, que presume que todas las interrelaciones entre los datos pueden estructurarse como jerarquías. Para ilustrar esto, se modifica ligeramente la base de datos de la Figura 1.3. Ahora, en lugar de ventas que sólo contienen un producto único, se tienen las facturas que a su vez contienen las líneas de factura. Cada cliente tiene varias facturas y cada factura tiene varias líneas. Cada línea registra la venta de un producto único. La Figura 1.8 muestra algunos ejemplos. Los archivos FACTURA y LÍNEA DE FACTURA reemplazan el archivo VENTA en la Figura 1.3.

La Figura 1.9 ilustra cómo se puede construir una jerarquía que muestra las interrelaciones entre los clientes, las facturas y las líneas de factura. Se considera que un cliente es el "propietario" de las facturas, las que a su vez son "propietarias" de las líneas de factura. En un sistema jerárquico de base de datos, estos tres archivos se conectan entre sí mediante punteros físicos o campos de datos añadidos a los registros individuales. Un **puntero (apuntador)** es una dirección física que identifica dónde puede encontrarse un registro sobre el disco. Cada registro de cliente contendría un apuntador al primer registro de factura correspondiente a ese registro de cliente. Los registros de factura contendrían a la vez punteros a otros registros de factura y a los registros de línea de factura. De esta manera, el sistema sería capaz de recuperar fácilmente todas las facturas y las líneas de factura que se aplican al cliente determinado.

modelo jerárquico. Un modelo de datos que resume que todas las interrelaciones entre los datos pueden estructurarse como jerarquías.

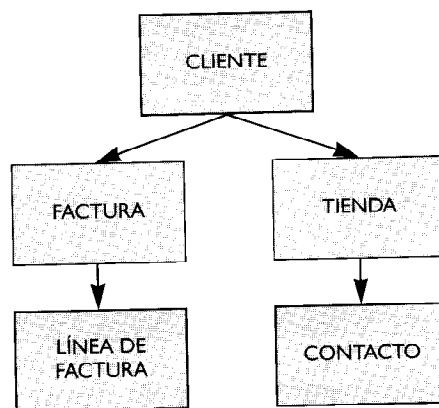
puntero. Una dirección física que identifica dónde puede encontrarse un registro sobre el disco.

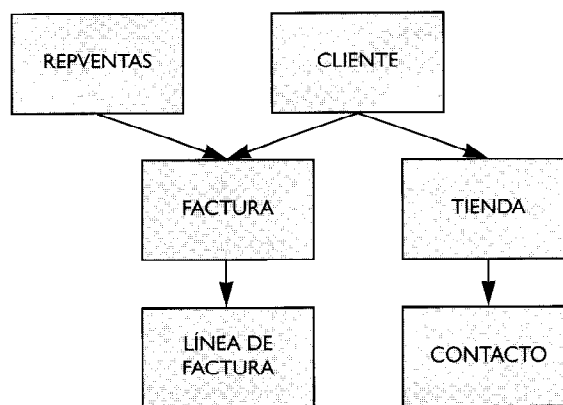
| CLIENTE | | | | PAGOS | |
|------------------|--------------|----------------|----------|-----------------|---------------------|
| ID_CLIENTE | NOMB_CLIENTE | DIRECCIÓN | PAÍS | BALANCE INICIAL | HASTA EL MES ACTUAL |
| 100 | Hnos. Watabe | Box 241, Tokio | Japón | 45.551 | 40.113 |
| 101 | Matlzl | Salzburg | Austria | 75.314 | 65.200 |
| 105 | Jefferson | B 918, Chicago | USA | 49.333 | 49.811 |
| 110 | Gómez | Santiago | Chile | 27.400 | 28.414 |
| FACTURA | | | | | |
| FACTURA-# | FECHA | ID_CLIENTE | ID_REP | | |
| 1012 | 02/10 | 100 | 39 | | |
| 1015 | 02/14 | 110 | 37 | | |
| 1020 | 02/20 | 100 | 14 | | |
| LÍNEA DE FACTURA | | | | | |
| FACTURA-# | LÍNEA-# | ID_PRODUCTO | CANTIDAD | PRECIO TOTAL | |
| 1012 | 01 | 1035 | 100 | 2200,00 | |
| 1012 | 02 | 2241 | 200 | 6650,00 | |
| 1012 | 03 | 2518 | 300 | 6360,00 | |
| 1015 | 01 | 1035 | 150 | 3300,00 | |
| 1015 | 02 | 2518 | 200 | 4240,00 | |
| 1020 | 01 | 2241 | 100 | 3325,00 | |
| 1020 | 02 | 2518 | 150 | 3180,00 | |



Suponga que estamos interesados en agregar información sobre los clientes a la base de datos jerárquica. Por ejemplo, como los clientes son compañías de tiendas por departamentos, podríamos querer tener un listado de las tiendas para cada cliente. En ese caso, expandiríamos el diagrama de la Figura 1.9 para que se vea como el de la Figura 1.10. CLIENTE todavía está relacionado con FACTURA, la que está relacionada con LÍNEA DE FACTURA. Sin embargo, CLIENTE también se relaciona con TIENDA y TIENDA está relacionada con CONTACTO. Se llama CONTACTO a un comprador a quien vendemos la mercancía para una tienda en particular. Como se ve en este diagrama, CLIENTE está en la raíz de una jerarquía de la cual puede derivarse una gran cantidad de información.

Estas figuras muestran el tipo de interrelaciones entre archivos que pueden realizarse fácilmente utilizando el modelo jerárquico. Sin embargo, se comprobó muy rápidamente que este modelo tenía algunas limitaciones importantes, ya que no todas las interrelaciones podrían expresarse fácilmente en una estructura jerárquica. Por ejemplo, para tomar el caso actual un paso más allá, es obvio que nos interesamos no solamente por la relación entre los clientes y las facturas, sino que también nos interesa la relación entre las facturas y los representantes de ventas. Es decir, queremos conocer todas las facturas que ha producido un representante de ventas determinado, de modo que se puedan emitir las instrucciones para su comisión. Esta nueva interrelación se muestra en la Figura 1.11.





hijo. Un registro "subordinado" en una interrelación jerárquica.

padre. Un registro "propietario" en una interrelación jerárquica.

red. Una interrelación de datos en la cual un registro puede estar subordinado a registros de más de un archivo.

Sin embargo, este diagrama no es una jerarquía. En una jerarquía, un **hijo** puede tener solamente un **padre**. En la Figura 1.10, FACTURA es un hijo y CLIENTE es su padre. No obstante, en la Figura 1.11 FACTURA tiene dos padres —REPVENTAS y CLIENTE—. Estas redes se denominan diagramas. A causa de la necesidad obvia de manipular tales interrelaciones, a finales de los años sesenta se desarrollaron los sistemas de base de datos *en red*. Al igual que los sistemas de base de datos jerárquicos, los sistemas de base de datos en red emplearon punteros físicos para enlazar entre sí los registros de diferentes archivos.

El SGBD jerárquico dominante es el IMS de IBM, desarrollado a mediados de los sesenta. Entre mediados de los sesenta y principios de los setenta se desarrollaron y se comercializaron exitosamente varios SGBD en redes y este modelo de datos se normalizó, eventualmente, como el modelo CODASYL. En capítulos posteriores se estudiarán extensamente ambos modelos de datos y las facilidades que brindan para la definición y la manipulación de los datos.

Sistemas de bases de datos relacionales

El uso de punteros físicos era simultáneamente una fortaleza y una debilidad de los sistemas de bases de datos jerárquicos y en red. Los punteros eran una fortaleza porque permitieron la recuperación rápida de los datos que tuvieran interrelaciones predeterminadas. La debilidad estaba en el hecho de que estas interrelaciones tenían que definirse *antes* de que el sistema fuera puesto en explotación. Era difícil, si no imposible, recuperar datos basados en otras interrelaciones. En la medida en que los usuarios se familiarizaron con los sistemas de base de datos y con su potencia para manipular los datos, rápidamente encontraron estas limitaciones inaceptables, como lo muestra esta polémica en la compañía IPD.



"Cordelia, estamos cada vez más frustrados con la cantidad de preguntas que nuestro sistema de base de datos no puede responder fácilmente. Cuando Sandy y tú nos convencisteis de que debíamos cambiar hacia un sistema de base de datos en red, argumentaste que tendríamos la posibilidad de obtener respuestas a la mayoría de las preguntas que fuéramos capaces de formular."

Susan Broadbent, presidenta ejecutiva de la compañía IPD, y Dick Greenberg, gerente de ventas de IPD, están hablando con la administradora de la base de datos de

IPD (ABD), Cordelia Molini, sobre los defectos de su actual sistema de gestión de bases de datos en red. Esto no es una queja nueva.

Con cada nuevo sistema, los directivos de IPD se percataban de que comenzaban a hacer preguntas que el sistema no respondía como les hubiera gustado. Aunque el sistema en red les sirvió durante varios años, ahora estaba llegando al momento en el que cada vez es más y más difícil obtener la información que se necesita. Cordelia conocía bien estas frustraciones.

"Bueno, tal vez nosotros debimos decir muchas preguntas que les gustaría hacer. La experiencia nos ha demostrado que existe una variedad muy extensa de preguntas que los usuarios quisieran formular." Cordelia continúa: "El problema es que el sistema en red depende de punteros físicos que conectan entre sí los datos en diferentes archivos. Si haces una pregunta que no corresponde naturalmente con esos punteros, no podemos responder a ella sin una cantidad considerable de programación. Ahora dime, ¿puedes darme una idea mejor de los tipos de preguntas para las cuales necesitas respuestas?"

Dick respondió inmediatamente: "Cordelia, queremos responder a todo tipo de preguntas. Realmente no podemos caracterizarlas por tipo, pues esto sería muy restringido. Idealmente, nos gustaría tener la posibilidad de formular cualquier pregunta que quisiéramos y, si la respuesta está en los datos, que el sistema sea capaz de darla."

"Lo que tú dices indica que necesitamos considerar seriamente el cambio hacia un sistema relacional de bases de datos. Los punteros físicos no se usan en un sistema relacional. Los datos se pueden relacionar siempre que exista una conexión lógica, luego no tenemos que preocuparnos por definir cuáles interrelaciones son las que el sistema utiliza con mayor probabilidad."

Susan pregunta: "¿Significa esto, por ejemplo, que Dick puede preguntar si los productos hechos en Ghana se venden bien en Corea? O ¿con qué éxito un representante de ventas en Río está vendiendo los equipos electrónicos de Amsterdam?"

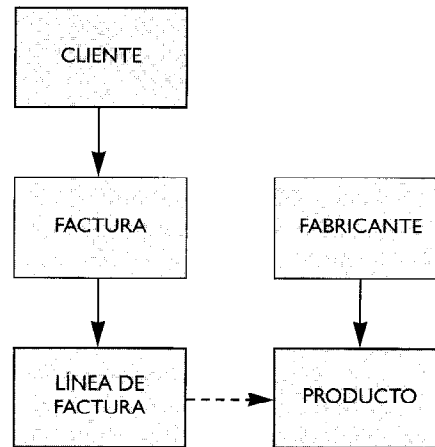
"Sí. Ambas preguntas son fáciles de responder en un sistema relacional. Además, te percatarás de que no tendrás que trabajar cerca de un programador tan a menudo. Los directivos que están deseosos de aprender un lenguaje de manipulación de datos relativamente simple pueden responder a muchas de sus propias preguntas haciéndolas directamente al sistema. Susan, ¿tú qué piensas?"

"Esto suena como algo que merece una investigación adicional. Si la tecnología es tentadora, deberíamos pensar seriamente en movernos hacia ella. ¿Por qué no profundizas en esto y simultáneamente haces una propuesta?"

En 1970, E. F. Codd publicó un artículo revolucionario (Codd, 1970) que desafió fuertemente el juicio convencional de la "condición" de las bases de datos. Codd argumentó que los datos deberían relacionarse mediante interrelaciones naturales, lógicas, inherentes a los datos, más que mediante punteros físicos. Es decir, las personas debían ser capaces de combinar los datos de fuentes diferentes, si la información lógica necesaria para hacer la combinación estaba presente en los datos. Esto abrió una nueva perspectiva para los sistemas de gestión de información, ya que las interrogaciones a las bases de datos no necesitarían, en adelante, limitarse a las interrelaciones indicadas por los punteros físicos.

Para ilustrar las deficiencias de los sistemas de bases de datos que se fundamentan en punteros físicos, considérese la Figura 1.12. En ella se muestra que CLIENTE, FACTURA y LÍNEA DE FACTURA están conectados por punteros físicos. FABRICANTE y PRODUCTO también están conectados. La línea punteada entre PRODUCTO y LÍNEA DE FACTURA indica que están relacionados lógicamente, ya que cada línea de factura se refiere a un producto específico.

Sin embargo, suponga que PRODUCTO no ha sido conectado con LÍNEA DE FACTURA mediante un puntero físico. ¿Cómo se puede obtener el informe siguiente?



Para cada cliente, identificar los fabricantes cuyos productos ha ordenado el cliente.

Esto requiere navegar desde CLIENTE a través de FACTURA y LÍNEA DE FACTURA, hasta PRODUCTO y, de ahí, subir hasta FABRICANTE. Puesto que la conexión física entre LÍNEA DE FACTURA y PRODUCTO no existe en la base de datos, esta navegación no se puede hacer mediante las facilidades normales de las bases de datos. En su lugar, para obtener la información pedida, deben utilizarse las técnicas antiguas y engorrosas del procesamiento de archivos. Ello necesitaría una cantidad considerable de compleja programación. En contraste con esto, los sistemas de base de datos que soportan la recuperación de los datos, tomando en consideración las interrelaciones lógicas, podrían resolver fácilmente tales problemas.

En su artículo, Codd propuso un modelo simple de datos en el que todos ellos se representarían en tablas constituidas por filas y columnas. A estas tablas se les dio el nombre matemático de *relaciones*, y por eso el modelo se denominó modelo relacional. Codd también propuso dos lenguajes para manipular los datos en las tablas: el álgebra relacional y el cálculo relacional (se estudiarán en el Capítulo 6). Ambos lenguajes soportan la manipulación de los datos sobre la base de operadores lógicos en lugar de los punteros físicos utilizados en los modelos jerárquico y en red.

Al manipular los datos sobre una base conceptual en vez de una base física, Codd introdujo otra innovación revolucionaria. En los sistemas de base de datos relacionales, los archivos completos de datos se pueden procesar con instrucciones sencillas. Sin embargo, los sistemas tradicionales requieren que los datos se procesen de registro en registro. El enfoque de Codd mejoró enormemente la eficiencia conceptual de la programación de la base de datos.

La manipulación lógica de los datos también hace factible la creación de lenguajes de interrogación más accesibles al usuario no especialista en computación. Aunque es bastante difícil crear un lenguaje que pueda ser utilizado por *todas* las personas sin considerar su experiencia previa en computación, los lenguajes relacionales de consulta hacen posible el acceso a las bases de datos para un grupo de usuarios cada vez mayor.

La publicación de los artículos de Codd, a principios de los años setenta, provocó una conmoción en la actividad de las comunidades de desarrollo de sistemas de investigación y de sistemas comerciales, en la medida en que trabajaban para producir un sistema de gestión de bases de datos relacional. El resultado fue la aparición de sistemas relacionales durante la última mitad de los setenta que soportaban lenguajes como el Structured Query Language (SQL), el Query Language (Quil) y el Query-by-Example (QBE). A

medida que las computadoras personales se hicieron populares durante los años ochenta, los sistemas relacionales para ellas también estuvieron disponibles. En 1986, el SQL se adoptó como la norma ANSI para los lenguajes relacionales de bases de datos. Esta norma se actualizó en 1989 y en 1992. En el Capítulo 7 se discuten diferentes aspectos del SQL-92.

Todos estos desarrollos hicieron avanzar enormemente el estado del arte en los sistemas de gestión de bases de datos y aumentaron la disponibilidad de información en las bases de datos colectivas. El enfoque relacional ha resultado bastante ventajoso. Además, han continuado las promesas en cuanto a los resultados de las investigaciones con vistas a garantizar capacidades cada vez más poderosas a medida que se alcanza una comprensión más completa de las necesidades del usuario con respecto a los sistemas de bases de datos.

Actualmente, los sistemas relacionales son un estándar en el mercado, especialmente en operaciones comerciales. Naturalmente, tanto los sistemas orientados a archivos, como también los sistemas de base de datos jerárquicos y en redes, son todavía abundantes y, para ciertas aplicaciones, constituyen la solución más eficiente en función de los costos. Sin embargo, durante algún tiempo, la tendencia clara de las compañías ha sido migrar a los sistemas relacionales siempre que fuera posible.

Aun así, sería un error asumir que los sistemas de bases de datos relacionales, ahora disponibles, representan la última palabra en el desarrollo de los SGBD. Los sistemas relacionales de hoy aún están evolucionando y, en algunos aspectos significativos, cambiando su naturaleza subyacente para permitir a los usuarios plantear problemas más complejos. Desde nuestro punto de vista, uno de los cambios más importantes está ocurriendo en el área de las bases de datos orientadas a objetos. En el Capítulo 4 se estudiará un modelo conceptual de datos que contiene aspectos característicos utilizados en las bases de datos orientadas a objetos. Un desarrollo adicional de gran importancia es la aparición de la plataforma cliente/servidor como la base para los cálculos y el acceso a las bases de datos en una organización. En la sección siguiente se presenta una panorámica de este concepto.

La Figura 1.13 muestra la cronología del desarrollo histórico de los métodos de acceso a los datos, que ya se ha discutido. La Tabla 1.1 muestra una comparación entre algunas características de los diversos métodos de acceso a los datos.

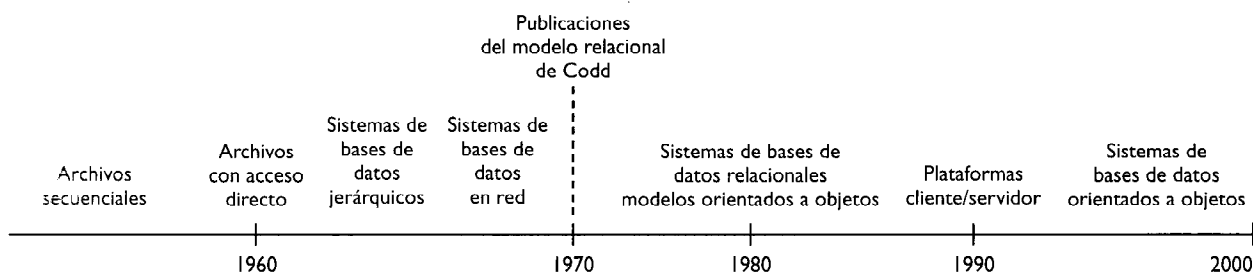
| <i>Método de acceso a los datos</i> | <i>Características</i> |
|-------------------------------------|---|
| Archivos secuenciales | Todos los registros en un archivo deben procesarse en secuencia |
| Archivos con acceso directo | Soportan el acceso directo a un registro específico. Es difícil el acceso a varios registros relacionados con un registro simple |
| Bases de datos jerárquicas | Soportan el acceso a varios registros relacionados con un registro simple Se restringen a las interrelaciones jerárquicas entre los datos Dependientes de punteros físicos predefinidos |
| Bases de datos en red | Soportan las interrelaciones jerárquicas y no-jerárquicas en redes entre los datos Dependientes de punteros físicos predefinidos |
| Bases de datos relacionales | Soportan todas las interrelaciones lógicas entre los datos El acceso a los datos es lógico, independiente de las técnicas de instrumentación física |

ochenta,
el SQL se
Esta nor-
ertos del

n los sis-
ón en las
Además,
n vistas a
ompre-
bases de

cialmen-
archivos,
lundan-
e los cos-
ha sido

les, aho-
sistemas
os, cam-
mas más
está ocu-
studiará
las bases
arición
s de
est con-



▼ Direcciones actuales-plataformas cliente/servidor



“De acuerdo, Sandy, explícame por qué piensas que nos ayudaría hacer la conversión a una plataforma cliente/servidor. De cualquier modo, ¿qué es cliente/servidor?”

Susan y Sanford continúan la discusión tecnológica en que los encontramos comenzando este capítulo. La compañía IPD ha progresado mediante sistemas de archivos y de bases de datos y ahora está considerando el siguiente paso. Naturalmente, antes de hacer un movimiento ellos quieren saber hacia dónde van y si esto será eficiente en función de los costos.

“Una plataforma cliente/servidor normalmente es una red local de área —LAN⁴— a la que se enlaza un conjunto de computadores personales y que contiene un computador especial que llamamos un servidor. Los computadores personales son los clientes, que solicitan al servidor diferentes servicios.”

“¿Por ejemplo?”

“El servidor les puede enviar programas tales como procesadores de textos u hojas de cálculo o puede ejecutar las consultas a la base de datos y enviarles los resultados. La idea básica es que cada máquina ejecute lo que haga mejor. El servidor recupera y actualiza los datos, el cliente realiza los cálculos especiales y presenta los datos al usuario.

“¿Esto será barato?”

“No necesariamente.”

“¿Entonces, qué ventajas ofrece?”

“Nuestros sistemas serán mucho más potentes y fáciles de utilizar. Cada usuario final tendrá para trabajar una Interfaz gráfica de usuario⁵. Asimismo, como hay un computador sobre cada escritorio, los sistemas tendrán muchas funciones adicionales. Seremos capaces de aumentar también las capacidades con mayor facilidad y a menor costo, simplemente, incrementando el número de máquinas. Y estaremos en una posición más flexible para sacar ventaja del hardware y del software nuevos.”

“Hasta aquí suena bien. Conformar un grupo para estudiar las ventajas y los riesgos. Si tu informe es favorable, podemos seguir adelante.”

La introducción del PC IBM en 1981 estableció la estación de trabajo personal como una norma en la oficina. El tratamiento de textos, las hojas de cálculo y otros *software* justificaron por sí solos el uso de estas máquinas. Además, para ellos era natural estar enla-

⁴ LAN, *Local Area Network* (N. del T.).

⁵ Interfaz Gráfica de Usuario, *Graphical User Interface (GUI)* (N. del T.).

plataforma cliente/servidor. Una red local que consiste en computadores clientes, que reciben servicios de un computador servidor.

servidor de bases de datos. Un programa que corre en un computador servidor para dar servicios de bases de datos a las máquinas clientes.

interfaz Gráfica de Usuario (GUI). Pantallas y funciones que proporcionan al usuario final un medio gráfico para tener acceso al sistema de computación.

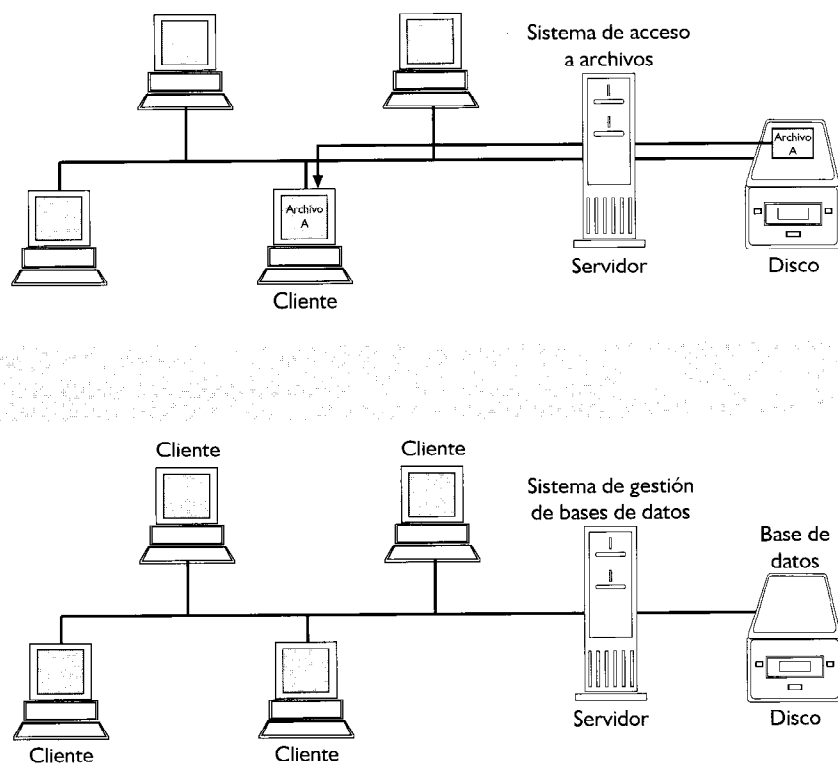
sistemas abiertos. El concepto de conectar una variedad de computadores con diferentes *hardware* y *software* para trabajar coordinadamente con el fin de lograr los objetivos del usuario.

interoperabilidad. El estado que caracteriza a múltiples sistemas heterogéneos que se comunican y contribuyen a la terminación de una tarea común

zados entre sí para permitirle a los usuarios comunicarse por medio del correo electrónico y tener acceso a los recursos comunes, tales como impresoras y discos. Inicialmente los servidores se instalaron para controlar la impresión y el acceso a los archivos. Ellos constituyeron los *servidores de impresión* y los *servidores de archivos*. Por ejemplo, el servidor de archivos, al responder a la solicitud de un cliente sobre un archivo específico, enviaría el archivo completo a través de la red al computador cliente (Figura 1.14). Hoy, sin embargo, la mayoría de los servidores son **servidores de bases de datos** —programas que se ejecutan en el *hardware* servidor y proporcionan los servicios de las bases de datos a los computadores clientes (Figura 1.15). Así, un cliente que está ejecutando un proceso de una aplicación y necesita una interrogación (consulta) a una base de datos, envía la petición al servidor de la base de datos y éste le devolverá los datos solicitados. El programa de aplicación puede enviar también datos al servidor con la solicitud de actualizar la base de datos. El servidor efectuaría esta actualización.

La potencia de la plataforma cliente/servidor descansa en el concepto de división de funciones. El cliente es el computador frontal que tiene interfaz directamente con el usuario. Manipula la **Interfaz Gráfica de Usuario (GUI)** y realiza los cálculos y otros programas de interés para el usuario final. El servidor es el que gestiona fundamentalmente el acceso a los datos atendiendo las diferentes peticiones de los clientes, también realiza funciones de control y seguridad de acceso a los datos.

Conceptualmente, las plataformas cliente/servidor son parte del concepto de **sistemas abiertos**, en el cual todo tipo de computadores, sistemas operativos, protocolos de redes y otros *software* y *hardware* pueden interconectarse y trabajar coordinadamente para lograr los objetivos del usuario. Sin embargo, en la práctica, los problemas de alcanzar tal variedad de sistemas operativos, protocolos de redes, sistemas de base de datos y otros, que trabajen conjuntamente pueden ser en extremo peligrosos. El objetivo de los sistemas abiertos consiste en lograr la **interoperabilidad**, que es el estado de dos o más siste-



mas heterogéneos comunicándose y contribuyendo cada uno a alguna parte del trabajo que corresponde a una tarea común.

En cierto sentido, el enfoque cliente/servidor es la culminación de una percepción temprana de la potencia del cálculo distribuida conjuntamente con el control de y el acceso a los datos inherentes a un computador centralizado. Los visionarios anunciaron la inminencia de la computación distribuida hace más de dos décadas, pero sólo recientemente esto se convirtió en una realidad. Aunque muchos problemas difíciles de la interoperabilidad están por resolverse, las perspectivas de aumentar continuamente la potencia de cálculo y el acceso a las bases de datos al nivel del usuario final nunca han sido más estimulantes.

▼ Los sistemas de base de datos: el hardware, el software, los datos, las personas

Hasta aquí, se han discutido, en términos generales, los sistemas de base de datos y sus capacidades. Ahora es el momento de acercarse a los componentes que constituyen tal tipo de sistema. Un sistema de base de datos es algo más que simples datos o que los datos en combinación con un software de gestión de bases de datos. En una organización, un sistema de base de datos *completo* está compuesto por cuatro componentes: el hardware, el software, los datos y las personas.

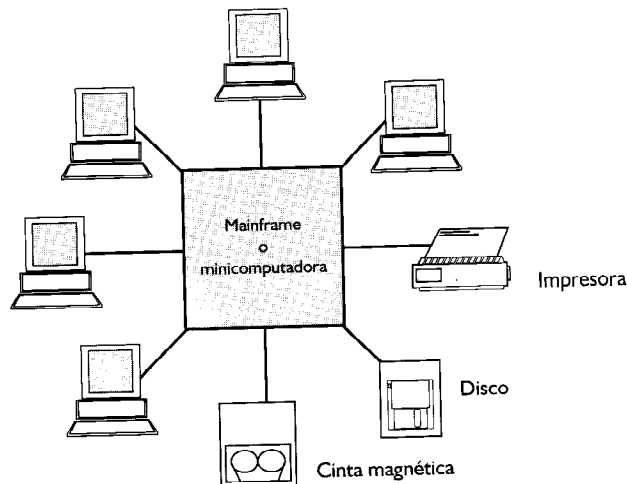
El hardware

El hardware es el conjunto de dispositivos físicos sobre los que reside una base de datos. Consiste en una o más computadoras, unidades de disco, vídeo-terminales, impresoras, unidades de cinta magnética, cables de conexión y otros equipos auxiliares y de conexión del equipamiento.

Las computadoras, utilizados para procesar los datos en la base de datos, pueden ser un *mainframe*, minicomputadoras o computadoras personales. En el ejemplo dado con anterioridad, la compañía IPD inicialmente comenzó el procesamiento con una minicomputadora y luego escaló a un *mainframe*. El *mainframe* y las minicomputadoras se han utilizado tradicionalmente de forma autónoma para soportar el acceso de varios usuarios a una base de datos común. Las computadoras personales se utilizan frecuentemente con bases de datos autónomas controladas y manipuladas por un usuario único. No obstante, también pueden conectarse a una red cliente/servidor, garantizando el acceso de varios usuarios a una base de datos común almacenada sobre unidades de disco y controlada por un computador servidor. El servidor en sí mismo puede ser una computadora personal más potente, una minicomputadora, un *mainframe* o una computadora más potente con multiprocesador. La Figura 1.16 ilustra una variedad de configuraciones de hardware.

Las unidades de disco constituyen el mecanismo de almacenamiento principal para las bases de datos y son esenciales, pues ellas permiten el acceso directo, sin el cual el procesamiento de las bases de datos sería imposible. Las computadoras personales, los vídeo-terminales y las impresoras se utilizan para introducir y recuperar información de las bases de datos. Las unidades de cinta magnética garantizan un respaldo barato y rápido de los datos que están almacenados en las unidades de disco.

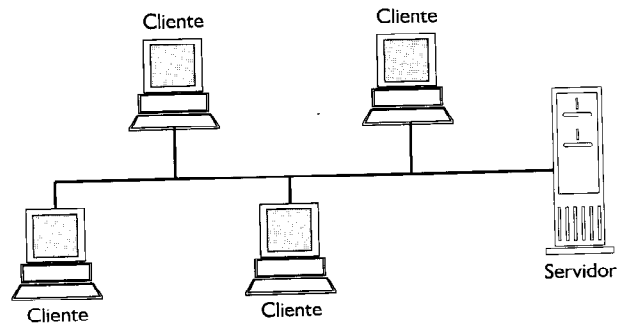
El éxito de los sistemas de base de datos ha dependido fuertemente de los adelantos en la tecnología del hardware. Para mantener y controlar la cantidad enorme de datos almacenados en una base de datos se requiere una memoria principal y un espacio de almacenamiento en disco muy grandes. Adicionalmente, se necesitan computadoras rápidas, redes y periféricos para ejecutar el alto número de accesos requerido para recuperar la información en un tiempo aceptable en un ambiente que tenga una cantidad grande de usuarios. Afortunadamente, el hardware ha sido cada vez más potente y más barato durante los años de desarrollo de la tecnología de las bases de datos. Esto ha hecho posible el uso difundido de los sistemas de bases de datos.



(a) Mainframe o minicomputadora que permite el acceso desde varias terminales



(b) Computadora personal que usa una base de datos almacenada



(c) Computadoras personales conectadas en una red cliente/servidor, todas las cuales tienen acceso a la base de datos almacenada en un servidor

El software

Un sistema de base de datos incluye dos tipos de software:

- El software de propósito general para la gestión de bases de datos, comúnmente llamado sistema de gestión de bases de datos (SGBD) (en inglés, DBMS).
- El software de aplicación, que usa las facilidades del SGBD para manipular la bases de datos con el fin de llevar a cabo una función específica de la compañía, tal como la emisión de los estados o el análisis de las tendencias de las ventas.

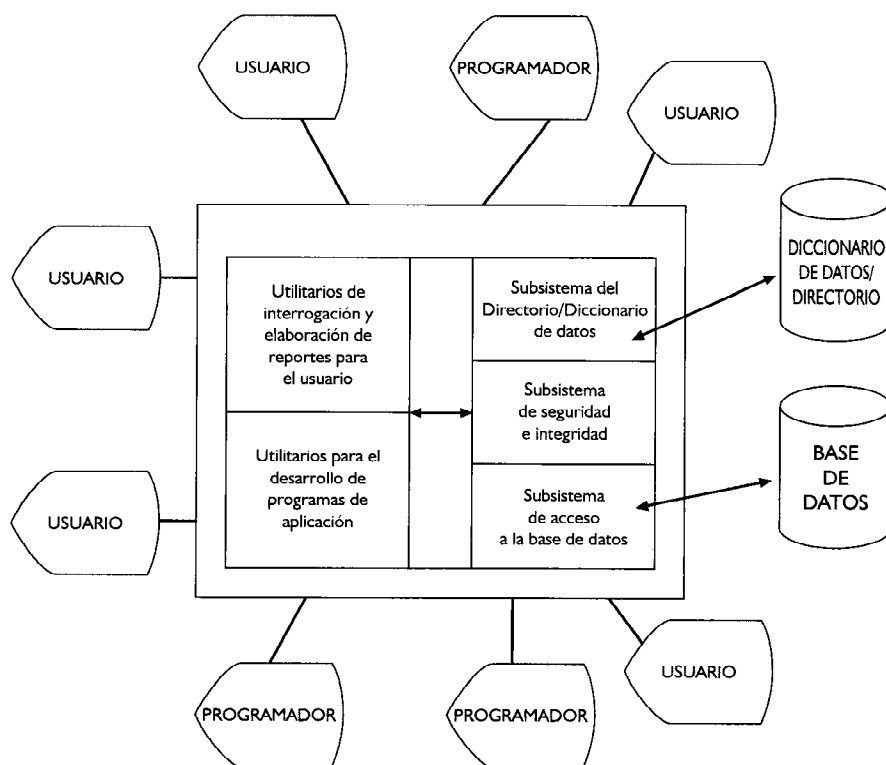
El software de aplicación generalmente se escribe por los empleados de una compañía para resolver un problema específico. Puede estar escrito en un lenguaje de programación estándar, tal como COBOL o C, o puede escribirse en un lenguaje (usualmente llamado lenguaje de cuarta generación) suministrado con el sistema de gestión de bases de

datos. El software de aplicación usa las facilidades del SGBD para el acceso a y la manipulación de los datos en la base de datos, proporcionando los informes o los documentos necesarios para los requisitos de información y de procesamiento de la compañía.

El *sistema de gestión de bases de datos* (SGBD) es un *software*, parecido a un sistema operativo o a un compilador, que brinda un conjunto de servicios a los usuarios finales, los programadores y otros (Figura 1.17). Como su nombre indica, el SGBD existe para facilitar la gestión de una base de datos. Con este fin, un SGBD típicamente brinda la mayoría de los servicios siguientes:

- Herramienta para la definición y el control centralizados de los datos, conocida como diccionario de datos/directorio (DD/D) o catálogo
- Mecanismos de seguridad e integridad de los datos
- Acceso concurrente a los datos para varios usuarios
- Utilidades para la consulta, la manipulación y la elaboración de informes orientados al usuario
- Utilidades para el desarrollo de sistemas de aplicación orientados al programador

Diccionario de datos/Directorio (DD/D). El subsistema del diccionario/directorio de datos almacena las definiciones de todos los elementos de los datos en la base de datos. Esto incluye el nivel primario de los elementos de los datos (campos), las estructuras de los datos a nivel de grupo y a nivel de registro y los archivos o tablas relacionales. El DD/D no sólo mantiene esta información, sino que, además, almacena las interrelaciones que existen entre las diversas estructuras de datos. Adicionalmente, mantiene los índices que se utilizan para garantizar el acceso a los datos rápidamente. Asimismo, almacena las definiciones de los formatos de las pantallas y de los informes, que pueden utilizarse por varios programas de aplicación.



metadatos. Datos en el diccionario de datos que describen la base de datos.

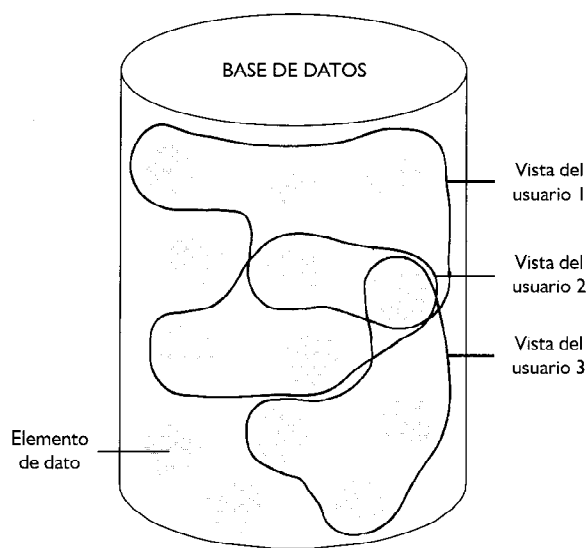
vista de los datos. Una definición de una porción restringida de la base de datos; también llamada una vista.

El diccionario de datos puede verse como una parte de la base de datos en sí misma. De esta manera, la base de datos es autodescriptiva, pues contiene información que describe su propia estructura. La información en el diccionario de datos se llama **metadatos** o "datos sobre los datos." Los metadatos están disponibles para la interrogación y la manipulación, como lo están los otros datos en la base de datos.

Mecanismos de seguridad e integridad de los datos. La base de datos es un recurso valioso que necesita protección. El SGBD garantiza la seguridad de la base de datos limitando el acceso a la misma al personal autorizado. Los usuarios autorizados, generalmente, estarán restringidos en cuanto al acceso a ciertos datos en particular y a aquéllos que puedan actualizar. Tal acceso se controla frecuentemente mediante contraseñas y mediante las **vistas de los datos**, que constituyen definiciones de porciones restringidas de la base de datos, como se ilustra en la Figura 1.18. La integridad y la consistencia de la base de datos se protegen por medio de restricciones sobre los valores que pueden tomar los elementos de los datos y por las capacidades de recuperación y de respaldo suministradas por el SGBD. Las definiciones de las restricciones de los datos se almacenan en el diccionario de datos. La recuperación y el respaldo se soportan por el software, el que automáticamente almacena los cambios de la base de datos en el catálogo y brinda los medios para restablecer el estado actual de la base de datos en el caso de un fallo del sistema.

Acceso concurrente a los datos por varios usuarios. Una de las funciones principales del SGBD es garantizar el acceso, la recuperación y la actualización de los datos en la base de datos. El SGBD proporciona los mecanismos físicos que permiten a varios usuarios tener acceso de forma rápida y eficiente a diferentes datos relacionados. Esta posibilidad se extiende a usuarios localizados remotamente, quienes tendrán acceso a la base de datos mediante un sistema de telecomunicaciones. Las utilidades del SGBD brindan una interfaz con los sistemas de telecomunicaciones, de modo que las solicitudes de datos y las respuestas resultantes se enrutan adecuadamente.

La centralización de los datos en una base de datos aumenta la probabilidad de que dos o más usuarios quieran tener acceso a los mismos datos concurrentemente. Si el SGBD permite esto, los dos usuarios seguramente afectarían el trabajo del otro y podrían dañarlo. Así, es importante que el SGBD proteja los datos, a los que tiene acceso un usuario, de la actualización simultánea por otro usuario. Para hacer esto, el SGBD utiliza mecanismos



n sí misma.
ón que des-
metadatos
a y la mani-

un recurso
datos limi-
generalmen-
quéllos que
y median-
de la base
la base de
ar los ele-
tradas por
diccionario
automática-
s para res-

principales
en la base
s usuarios
posibilidad
e de datos
na interfaz
s respues-

dad de que
el SGBD
an dañar-
uario, de
evanismos

sofisticados de bloqueo para proteger los datos que se están actualizando por un usuario, mientras al mismo tiempo garantiza el acceso concurrente a la base de datos y un tiempo de respuesta del sistema a otros usuarios aceptable.

Consultas e informes dirigidos al usuario. Uno de los aspectos más valiosos de un SGBD es que brinda herramientas de manipulación de los datos dirigidas al usuario. Estos lenguajes de interrogación fáciles de usar permiten a los usuarios formular sus *consultas* y pedir informes únicos directamente de la base de datos. Esto releva al personal de programación de la compañía de la carga de formular estas *consultas* o de escribir paquetes de programas de aplicación de propósito especial.

Los generadores de informes están asociados a los lenguajes de *consulta*. Frecuentemente el lenguaje de *consulta* contendrá facilidades para estructurar los resultados de las *consultas* como informes. La *consulta* formulada puede almacenarse para su uso ulterior y sus resultados pueden producirse en forma de un informe ordinario. Cuando este es el caso, el lenguaje de *consulta* puede considerarse como un generador de informes. Además, los generadores de informes pueden también brindar facilidades mucho más poderosas que las disponibles en el lenguaje de *consulta*.

Facilidades para el desarrollo de aplicaciones. Aparte de facilitar al usuario el acceso a la base de datos para obtener información, el SGBD también garantiza una asistencia importante al programador de aplicaciones. Tales herramientas, como los generadores de pantallas, de menús y de reportes; los generadores de aplicaciones; los compiladores, y las facilidades para la definición de los datos y las vistas de los datos son estándares. Más importantes aún, los sistemas de base de datos modernos brindan componentes del lenguaje mucho más poderosos que los de los lenguajes tradicionales, lo que contribuye a que el proceso de programación sea apreciablemente más eficiente.

Los datos

Obviamente, ningún sistema de base de datos puede existir sin los datos, los hechos básicos sobre los que se fundamentan las necesidades de información y de procesamiento de una compañía. Sin embargo, el factor esencial a considerar es que los datos que conforman una base de datos tienen que ser cuidadosa y lógicamente estructurados. Las funciones del negocio deben analizarse, los elementos de los datos y las interrelaciones deben identificarse y definirse cabalmente y estas definiciones deben almacenarse de manera precisa en el diccionario de datos. Entonces, los datos pueden obtenerse e introducirse en la base de datos según la estructura definida. Una base de datos construida en armonía con estos procedimientos puede ser un recurso poderoso para garantizar información oportuna a la organización.

Las personas

El caso de la compañía IPD permite identificar dos tipos diferentes de personas a las que concierne la base de datos. Susan Broadbent y Dick Greenberg son los **usuarios**, las personas que necesitan la información de la base de datos para desarrollar su responsabilidad primaria en el negocio, que en sí misma se encuentra en cierta área funcional diferente. Por el contrario, Sanford Mallon y Cordelia Molini son **profesionales de la computación**, personas cuya responsabilidad primaria en el negocio está en el diseño y el mantenimiento del sistema de base de datos y su paquete de programas de aplicación asociado para el beneficio de los usuarios. Ejemplos de personas en cada una de estas categorías pueden ser las siguientes:

Usuarios: Los ejecutivos, los gerentes, los administradores, el personal de oficina

Profesionales de la computación: Los administradores de la base de datos, los analistas, los programadores, los diseñadores del sistema y de la base de datos, los administradores de los sistemas de información

Las personas encargadas de los **procedimientos** usados para lograr las metas del sistema constituyen una parte importante de este componente. Virtualmente, ningún sistema

usuarios. Personas que necesitan información de la base de datos para desarrollar su responsabilidad primaria en el negocio.

profesionales de la computación. Personas responsables del sistema de base de datos y del paquete de programas de aplicación asociado.

procedimiento. Instrucciones escritas que describen los pasos necesarios para realizar una tarea determinada en un sistema.

automatiza por completo una tarea del usuario. Es necesario desarrollar procedimientos manuales para brindar una interfaz suave entre los usuarios del sistema y el sistema en sí mismo. Un ejemplo de procedimiento sería el control de auditoría por parte de los usuarios con respecto a que la cantidad total de dinero depositada en el banco correspondiente a un día determinado concuerde con la cantidad total de efectivo recibido mostrada por el sistema ese mismo día. Normalmente, existen muchos procedimientos en un sistema y el éxito del sistema, con frecuencia, depende mucho de la habilidad con que tales procedimientos se desarrollen para articular con las funciones del sistema, así como de la estructura del sistema en sí mismo.

Interrelación (relación) entre los cuatro componentes del sistema

La figura 1.19 resume la interrelación entre los cuatro componentes de un sistema de base de datos. Los *profesionales de la computación* (los diseñadores de base de datos y analistas) en consulta con los *usuarios* identifican las necesidades de *datos* y diseñan las estructuras de la base de datos para responder a estas necesidades. Entonces, las estructuras de la base de datos se especifican al SGBD mediante el diccionario de datos. Los *usuarios introducen los datos* en el sistema siguiendo *procedimientos* específicos. Los datos introducidos se almacenan en los medios de hardware, tales como discos y cintas. Los *programas de aplicación* que tienen acceso a la base de datos se escriben por los profesionales de la computación y por los usuarios para ejecutarlos sobre los *computadores*. Estos programas utilizan el lenguaje de órdenes del SGBD y hacen uso de la información contenida en el diccionario de datos. Estos programas generan información, que puede utilizarse por los ejecutivos y los gerentes para tomar decisiones en el negocio. Los programas de aplicación también pueden generar las notificaciones de facturas y otros documentos utilizados por los clientes del negocio. De este modo, puede verse que en un sistema adecuadamente diseñado y en funcionamiento, sus cuatro componentes —el hardware, el software, los datos y las personas— conforman un sistema único con el fin de alcanzar las metas de la organización.

