

Capítulo 1

Conceptos de bases de datos

Introducción y objetivos

El inicio de un curso sobre bases de datos debe ser, sin duda, la definición de *base de datos* y la presentación de los sistemas de gestión de bases de datos (el *software* que facilita la creación y manipulación de las mismas por parte del personal informático). Algunos de estos sistemas, ampliamente utilizados, son PostgreSQL, MySQL y Oracle.

Ya que este texto está dirigido a estudiantado de las ingenierías informáticas, es interesante conocer qué papeles puede desempeñar el personal informático en el entorno de una base de datos. Éstas han tenido sus predecesores en los sistemas de ficheros y tienen por delante un amplio horizonte, por lo que antes de comenzar su estudio resulta conveniente ubicarse en el tiempo haciendo un recorrido por su evolución histórica. El capítulo termina con una exposición sobre las ventajas y desventajas que las bases de datos conllevan.

Al finalizar este capítulo, el estudiantado debe ser capaz de:

- Definir qué es una base de datos y qué es un sistema de gestión de bases de datos.
- Reconocer los subsistemas que forman parte de un sistema de gestión de bases de datos.
- Enumerar las personas que aparecen en el entorno de una base de datos y sus tareas.
- Asociar los distintos tipos de sistemas de gestión de bases de datos a las generaciones a las que pertenecen.
- Enumerar las ventajas y desventajas de los sistemas de bases de datos y asociarlas al motivo por el que se producen: la integración de datos o el sistema de gestión de la base de datos.

1.1. Base de datos

Una *base de datos* es un conjunto de datos almacenados en memoria externa que están organizados mediante una estructura de datos. Cada base de datos ha sido diseñada para satisfacer los requisitos de información de una empresa u otro tipo de organización, como por ejemplo, una universidad o un hospital.

Antes de existir las bases de datos se trabajaba con *sistemas de ficheros*. Los sistemas de ficheros surgieron al informatizar el manejo de los archivadores manuales para proporcionar un acceso más eficiente a los datos almacenados en los mismos. Un sistema de ficheros sigue un modelo descentralizado, en el que cada departamento de la empresa almacena y gestiona sus propios datos mediante una serie de programas de aplicación escritos especialmente para él. Estos programas son totalmente independientes entre un departamento y otro, y se utilizan para introducir datos, mantener los ficheros y generar los informes que cada departamento necesita. Es importante destacar que en los sistemas de ficheros, tanto la estructura física de los ficheros de datos como la de sus registros, están definidas dentro de los programas de aplicación.

Cuando en una empresa se trabaja con un sistema de ficheros, los departamentos no comparten información ni aplicaciones, por lo que los datos comunes deben estar duplicados en cada uno de ellos. Esto puede originar inconsistencias en los datos. Se produce una inconsistencia cuando copias de los mismos datos no coinciden: dos copias del domicilio de un cliente pueden no coincidir si sólo uno de los departamentos que lo almacenan ha sido informado de que el domicilio ha cambiado.

Otro inconveniente que plantean los sistemas de ficheros es que cuando los datos se separan en distintos ficheros, es más complicado acceder a ellos, ya que el programador de aplicaciones debe sincronizar el procesamiento de los distintos ficheros implicados para garantizar que se extraen los datos correctos. Además, ya que la estructura física de los datos se encuentra especificada en los programas de aplicación, cualquier cambio en dicha estructura es difícil de realizar. El programador debe identificar todos los programas afectados por el cambio, modificarlos y volverlos a probar, lo que cuesta mucho tiempo y está sujeto a que se produzcan errores. A este problema, tan característico de los sistemas de ficheros, se le denomina también falta de *independencia de datos lógica-física*.

Una base de datos se puede percibir como un gran almacén de datos que se define y se crea una sola vez, y que se utiliza al mismo tiempo por distintos usuarios. En una base de datos todos los datos se integran con una mínima cantidad de duplicidad. De este modo, la base de datos no pertenece a un solo departamento sino que se comparte por toda la organización. Además, la base de datos no sólo contiene los datos de la organización, también almacena una descripción de dichos datos. Esta descripción es lo que se denomina *metadatos*, se almacena en el *diccionario de datos* o *catálogo* y es lo que permite que exista independencia de datos lógica-física.

1.2. Sistema de gestión de bases de datos

El *sistema de gestión de la base de datos* (en adelante SGBD) es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, además de proporcionar un acceso controlado a la misma. Se denomina *sistema de bases de datos* al conjunto formado por la base de datos, el SGBD y los programas de aplicación que dan servicio a la empresa u organización.

El modelo seguido con los sistemas de bases de datos es muy similar al modelo que se sigue en la actualidad para el desarrollo de programas con lenguajes orientados a objetos, en donde se da una implementación interna de un objeto y una especificación externa separada. Los usuarios del objeto sólo ven la especificación externa y no se deben preocupar de cómo se implementa internamente el objeto. Una ventaja de este modelo, conocido como abstracción de datos, es que se puede cambiar la implementación interna de un objeto sin afectar a sus usuarios ya que la especificación externa no se ve alterada. Del mismo modo, los sistemas de bases de datos separan la definición de la estructura física de los datos de su estructura lógica, y almacenan esta definición en la base de datos. Todo esto es gracias a la existencia del SGBD, que se sitúa entre la base de datos y los programas de aplicación.

Generalmente, un SGBD proporciona los servicios que se citan a continuación:

- El SGBD permite la definición de la base de datos mediante un *lenguaje de definición de datos*. Este lenguaje permite especificar la estructura y el tipo de los datos, así como las restricciones sobre los datos.
- El SGBD permite la inserción, actualización, eliminación y consulta de datos mediante un *lenguaje de manejo de datos*. El hecho de disponer de un lenguaje para realizar consultas reduce el problema de los sistemas de ficheros, en los que el usuario tiene que trabajar con un conjunto fijo de consultas, o bien, dispone de un gran número de programas de aplicación costosos de gestionar. Hay dos tipos de lenguajes de manejo de datos: los *procedurales* y los *no procedurales*. Estos dos tipos se distinguen por el modo en que acceden a los datos. Los lenguajes procedurales manipulan la base de datos registro a registro, mientras que los no procedurales operan sobre conjuntos de registros. En los lenguajes procedurales se especifica qué operaciones se debe realizar para obtener los datos resultado, mientras que en los lenguajes no procedurales se especifica qué datos deben obtenerse sin decir cómo hacerlo. El lenguaje no procedural más utilizado es el SQL (*Structured Query Language*) que, de hecho, es un estándar y es el lenguaje de los SGBD relacionales.
- El SGBD proporciona un acceso controlado a la base de datos mediante:
 - Un sistema de seguridad, de modo que los usuarios no autorizados no puedan acceder a la base de datos.

- Un sistema de integridad que mantiene la integridad y la consistencia de los datos.
- Un sistema de control de concurrencia que permite el acceso compartido a la base de datos.
- Un sistema de control de recuperación que restablece la base de datos después de que se produzca un fallo del *hardware* o del *software*.
- Un diccionario de datos o catálogo, accesible por el usuario, que contiene la descripción de los datos de la base de datos.

A diferencia de los sistemas de ficheros, en los que los programas de aplicación trabajan directamente sobre los ficheros de datos, el SGBD se ocupa de la estructura física de los datos y de su almacenamiento. Con esta funcionalidad, el SGBD se convierte en una herramienta de gran utilidad. Sin embargo, desde el punto de vista del usuario, se podría discutir que los SGBD han hecho las cosas más complicadas, ya que ahora los usuarios ven más datos de los que realmente quieren o necesitan, puesto que ven la base de datos completa. Conscientes de este problema, los SGBD proporcionan un mecanismo de *vistas* que permite que cada usuario tenga su propia vista o visión de la base de datos. El lenguaje de definición de datos permite definir vistas como subconjuntos de la base de datos.

Todos los SGBD no presentan la misma funcionalidad, depende de cada producto. En general, los grandes SGBD multiusuario ofrecen todas las funciones que se acaban de citar e incluso más. Los sistemas modernos son conjuntos de programas extremadamente complejos y sofisticados, con millones de líneas de código y con una documentación consistente en varios volúmenes. Lo que se pretende es proporcionar un sistema que permita gestionar cualquier tipo de requisitos y que tenga un 100 % de fiabilidad ante cualquier tipo de fallo. Los SGBD están en continua evolución, tratando de satisfacer los requisitos de todo tipo de usuarios. Por ejemplo, muchas aplicaciones de hoy en día necesitan almacenar imágenes, vídeo, sonido, etc. Para satisfacer a este mercado, los SGBD deben evolucionar. Conforme vaya pasando el tiempo, irán surgiendo nuevos requisitos, por lo que los SGBD nunca permanecerán estáticos.

1.3. Personas en el entorno de las bases de datos

Hay cuatro grupos de personas que intervienen en el entorno de una base de datos: el administrador de la base de datos, los diseñadores de la base de datos, los programadores de aplicaciones y los usuarios.

El *administrador de la base de datos* se encarga de la implementación física de la base de datos: escoge los tipos de los ficheros de datos y de los índices que deben crearse, determina dónde deben ubicarse ficheros e índices y, en general,

toma las decisiones relativas al almacenamiento físico en función de las posibilidades que le ofrezca el SGBD con el que trabaje. Además, el administrador de la base de datos se encarga de establecer la política de seguridad y del acceso concurrente. También se debe preocupar de que el sistema se encuentre siempre operativo y procurar que los usuarios y las aplicaciones obtengan buenas prestaciones. El administrador debe conocer muy bien el SGBD con el que trabaja, así como el equipo informático sobre el que esté funcionando.

Los *diseñadores de la base de datos* realizan el diseño de la base de datos, debiendo identificar los datos, las relaciones entre ellos y las restricciones sobre los datos y sobre sus relaciones. El diseñador de la base de datos debe tener un profundo conocimiento de los datos de la empresa y también debe conocer sus *reglas de negocio*. Las reglas de negocio describen las características principales sobre el comportamiento de los datos tal y como las ve la empresa. Para obtener un buen resultado, el diseñador de la base de datos debe implicar en el proceso a todos los usuarios de la base de datos, tan pronto como sea posible.

Una vez se ha diseñado e implementado la base de datos, los *programadores de aplicaciones* se encargan de implementar los programas de aplicación que servirán a los usuarios finales. Estos programas de aplicación son los que permiten consultar datos, insertarlos, actualizarlos y eliminarlos. Estos programas se escriben mediante lenguajes de tercera generación o de cuarta generación.

Los *usuarios finales* son los clientes de la base de datos: la base de datos ha sido diseñada e implementada, y está siendo mantenida, para satisfacer sus requisitos en la gestión de su información.

1.4. Historia de los sistemas de bases de datos

Los predecesores de los sistemas de bases de datos fueron los sistemas de ficheros. Un sistema de ficheros está formado por un conjunto de ficheros de datos y los programas de aplicación que permiten a los usuarios finales trabajar sobre los mismos. No hay un momento concreto en el que los sistemas de ficheros hayan cesado y hayan dado comienzo los sistemas de bases de datos. De hecho, todavía existen sistemas de ficheros en uso.

Se dice que los sistemas de bases de datos tienen sus raíces en el proyecto estadounidense de mandar al hombre a la luna en los años sesenta, el proyecto Apolo. En aquella época, no había ningún sistema que permitiera gestionar la inmensa cantidad de información que requería el proyecto. La primera empresa encargada del proyecto, NAA (*North American Aviation*), desarrolló una aplicación denominada GUAM (*General Update Access Method*) que estaba basada en el concepto de que varias piezas pequeñas se unen para formar una pieza más grande, y así sucesivamente hasta que el producto final está ensamblado. Esta estructura, que tiene la forma de un árbol, es lo que se denomina una *estructura jerárquica*. A mediados de los sesenta, IBM se unió a NAA para desarrollar GUAM en lo que después fue IMS (*Information Management*

System). El motivo por el cual IBM restringió IMS al manejo de jerarquías de registros fue el de permitir el uso de dispositivos de almacenamiento serie, más exactamente las cintas magnéticas, ya que era un requisito del mercado por aquella época.

A mitad de los sesenta, General Electric desarrolló IDS (*Integrated Data Store*). Este trabajo fue dirigido por uno de los pioneros en los sistemas de bases de datos, Charles Bachmann. IDS era un nuevo tipo de sistema de bases de datos conocido como *sistema de red*, que produjo un gran efecto sobre los sistemas de información de aquella generación. El sistema de red se desarrolló, en parte, para satisfacer la necesidad de representar relaciones entre datos más complejas que las que se podían modelar con los sistemas jerárquicos y, en parte, para imponer un estándar de bases de datos. Para ayudar a establecer dicho estándar, el grupo CODASYL (*Conference on Data Systems Languages*), formado por representantes del gobierno de EEUU y representantes del mundo empresarial, fundaron un grupo denominado DBTG (*Data Base Task Group*), cuyo objetivo era definir unas especificaciones estándar que permitieran la creación de bases de datos y el manejo de los datos. El DBTG presentó su informe final en 1971 y aunque éste no fue formalmente aceptado por ANSI (*American National Standards Institute*), muchos sistemas se desarrollaron siguiendo la propuesta del DBTG. Estos sistemas son los que se conocen como sistemas de red, sistemas CODASYL o DBTG.

Los sistemas jerárquico y de red constituyen la *primera generación* de los SGBD. Estos sistemas presentan algunos inconvenientes:

- Es necesario escribir complejos programas de aplicación para responder a cualquier tipo de consulta de datos, por simple que ésta sea.
- La independencia de datos es mínima.
- No tienen un fundamento teórico.

En 1970, Edgar Frank Codd de los laboratorios de investigación de IBM, escribió un artículo presentando el *modelo relacional*. En este artículo presentaba también los inconvenientes de los sistemas previos, el jerárquico y el de red. Pasó casi una década hasta que se desarrollaron los primeros sistemas relacionales. Uno de los primeros es System R, de IBM, que se desarrolló para probar la funcionalidad del modelo relacional, proporcionando una implementación de sus estructuras de datos y sus operaciones. Esto condujo a dos grandes desarrollos:

- El desarrollo de un lenguaje de consultas estructurado denominado SQL, que se ha convertido en el lenguaje estándar de los sistemas relacionales.
- La producción de varios SGBD relacionales durante los años ochenta, como DB2 y SLQ/DS, de IBM, y Oracle, de Oracle Corporation.

Hoy en día, existen cientos de SGBD relacionales, tanto para microordenadores como para sistemas multiusuario, aunque muchos no son completamente fieles al modelo relacional.

Los SGBD relacionales constituyen la *segunda generación* de los SGBD. Sin embargo, el modelo relacional también tiene sus debilidades, siendo una de ellas su limitada capacidad al modelar los datos. Se ha desarrollado mucha investigación desde entonces tratando de resolver este problema. En 1976, Peter Chen presentó el modelo entidad-relación, que es la técnica más utilizada en el diseño de bases de datos. En 1979, Codd intentó subsanar algunas de las deficiencias de su modelo relacional con una versión extendida denominada RM/T (1979) y más recientemente RM/V2 (1990). Los intentos de proporcionar un modelo de datos que represente al mundo real de un modo más fiel han dado lugar a los modelos de datos semánticos.

La evolución reciente de la tecnología de bases de datos viene marcada por una mayor solidez en las bases de datos orientadas a objetos, la extensión de las bases de datos relacionales y el procesamiento distribuido. Esta evolución representa la *tercera generación* de los SGBD.

Por su parte, los sistemas de gestión de bases de datos relacionales han ido evolucionando estos últimos años para soportar objetos y reglas, y para ampliar el lenguaje SQL y hacerlo más extensible y computacionalmente completo, dando lugar a lo que se conoce como sistemas objeto-relacionales.

Durante la última década, el impacto de los avances en la tecnología de las comunicaciones ha sido muy importante. Esto ha contribuido a que en las empresas se haya producido una mayor distribución de la gestión automática de la información, en contraste con la filosofía centralizadora predominante en la tecnología inicial de bases de datos. Las bases de datos distribuidas posibilitan el procesamiento de datos pertenecientes a distintas bases de datos conectadas entre sí. El emplazamiento lógico de cada una de las bases de datos se denomina nodo, conteniendo cada uno su sistema de gestión de bases de datos, junto con las utilidades y facilidades propias del soporte distribuido. Los nodos, por lo general, están ubicados en emplazamientos físicos distantes geográficamente, y se encuentran conectados por una red de comunicación de datos.

Por otra parte, los sistemas de bases de datos activas han sido propuestos como otro paradigma de gestión de datos que satisface las necesidades de aquellas aplicaciones que requieren una respuesta puntual ante situaciones críticas. Como ejemplos se puede citar el control del tráfico aéreo o las aplicaciones de control de plantas industriales. Este paradigma también puede ser utilizado para soportar varias de las funciones del propio sistema de gestión de bases de datos, como son: el control de accesos, el control de la integridad, el mantenimiento de vistas o el mantenimiento de atributos derivados. El factor común en todas estas aplicaciones es la necesidad de responder a sucesos, tanto externos como internos al propio sistema. A diferencia de los sistemas pasivos, un sistema de gestión de bases de datos activas responde automáticamente

ante determinadas circunstancias descritas por el diseñador. La mayoría de los sistemas de gestión de bases de datos comerciales incorporan la posibilidad de definir reglas, por lo que son, en cierto modo, sistemas activos.

Las investigaciones sobre la relación entre la teoría de las bases de datos y la lógica se remontan a finales de la década de los setenta. Estas investigaciones han dado lugar a las bases de datos deductivas, que permiten derivar nuevas informaciones a partir de las introducidas explícitamente por el usuario. Esta función deductiva se realiza mediante la adecuada explotación de ciertas reglas de conocimiento relativas al dominio de la aplicación, utilizando para ello técnicas de programación lógica y de inteligencia artificial.

Los sistemas de múltiples bases de datos permiten realizar operaciones que implican a varios sistemas de bases de datos, cada uno de los cuales puede ser centralizado o distribuido. Cada sistema de bases de datos que participa es denominado *componente*. Si todos los sistemas de gestión de bases de datos de los diferentes componentes son iguales, el sistema de múltiples bases de datos es homogéneo; en caso contrario, es heterogéneo. Un sistema de múltiples bases de datos es un sistema federado de bases de datos si permite una doble gestión: una de carácter global, realizada por el sistema de gestión de bases de datos federadas y otra en modo autónomo e independiente del sistema federado, realizada por parte de los sistemas componentes.

La influencia de la Web lo abarca todo. En su desarrollo se han ignorado las técnicas de bases de datos, por lo que se han repetido los errores cometidos en las primeras generaciones de los sistemas de gestión de bases de datos. La Web se puede ver como una nueva interfaz de acceso a bases de datos, y muchos sistemas de gestión de bases de datos ya proporcionan almacenamiento y acceso a datos a través de XML. Pero la Web puede también ser considerada como una inmensa base de datos, siendo éste un tema de investigación en pleno auge.

Por otra parte, los grandes almacenes de datos (*data warehouses*) ya han demostrado que si son implementados convenientemente, pueden ser de gran ayuda en la toma de decisiones y en el procesamiento analítico en tiempo real OLAP (*On-Line Analytical Processing*). Los datos son extraídos periódicamente de otras fuentes y son integrados en el almacén. Estos datos, relevantes para la empresa, son no-volátiles y se agrupan según diversas granularidades en el tiempo y en otras dimensiones. En la actualidad, existe una gran competencia entre las extensiones de los sistemas de gestión de bases de datos comerciales para incorporar las características de este tipo de sistemas, y la creación de productos específicos.

La explotación de datos (*data mining* o *knowledge discovery in databases*) trata de descubrir conocimientos útiles y previamente no conocidos a partir de grandes volúmenes de datos, por lo que no sólo integra técnicas de bases de datos, sino también de estadística y de inteligencia artificial. Las investigaciones se han plasmado rápidamente en productos comerciales, con un desarrollo reciente bastante importante.

Existen también muchos trabajos de investigación en temas tales como las bases de datos temporales y las bases de datos multimedia. Las bases de datos temporales intentan, en primer lugar, definir un modelo de datos que capture la semántica del tiempo en el mundo real, y, en segundo lugar, realizar una implementación eficiente de tal modelo. Los recientes avances en el almacenamiento de distintos tipos de información, como voz, imágenes o sonido, han tenido su influencia en las bases de datos, dando lugar a las bases de datos multimedia.

La rápida evolución que la tecnología de bases de datos ha experimentado en la última década, así como la variedad de nuevos caminos abiertos, han conducido a investigadores y asociaciones interesadas, a reflexionar sobre el futuro de esta tecnología. Estas reflexiones quedan recogidas en numerosos debates y manifiestos que intentan poner orden en un campo en continua expansión.

1.5. Ventajas e inconvenientes de los sistemas de bases de datos

Los sistemas de bases de datos presentan numerosas ventajas gracias, fundamentalmente, a la integración de datos y a la interfaz común que proporciona el SGBD. Estas ventajas se describen a continuación.

- *Control sobre la redundancia de datos.* Los sistemas de ficheros almacenan varias copias de los mismos datos en ficheros distintos. Esto hace que se desperdicie espacio de almacenamiento, además de provocar faltas de consistencia de datos (copias que no coinciden). En los sistemas de bases de datos todos estos ficheros están integrados, por lo que no se almacenan varias copias de los mismos datos. Sin embargo, en una base de datos no se puede eliminar la redundancia completamente, ya que en ocasiones es necesaria para modelar las relaciones entre los datos, o bien es necesaria para mejorar las prestaciones.
- *Control sobre la consistencia de datos.* Eliminando o controlando las redundancias de datos se reduce en gran medida el riesgo de que haya inconsistencias. Si un dato está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez, y está disponible para todos los usuarios inmediatamente. Si un dato está duplicado y el sistema conoce esta redundancia, el propio sistema puede encargarse de garantizar que todas las copias se mantengan consistentes. Desgraciadamente, no todos los SGBD de hoy en día se encargan de mantener automáticamente la consistencia.
- *Compartición de datos.* En los sistemas de ficheros, los ficheros pertenecen a los departamentos que los utilizan, pero en los sistemas de bases de

datos, la base de datos pertenece a la empresa y puede ser compartida por todos los usuarios que estén autorizados. Además, las nuevas aplicaciones que se vayan creando pueden utilizar los datos de la base de datos existente.

- *Mantenimiento de estándares.* Gracias a la integración es más fácil respetar los estándares necesarios, tanto los establecidos a nivel de la empresa como los nacionales e internacionales. Estos estándares pueden establecerse sobre el formato de los datos para facilitar su intercambio; pueden ser estándares de documentación, procedimientos de actualización y también reglas de acceso.
- *Mejora en la integridad de datos.* La integridad de la base de datos se refiere a la validez de los datos almacenados. Normalmente, la integridad se expresa mediante restricciones o reglas que no se pueden violar. Estas restricciones se pueden aplicar tanto a los datos, como a sus relaciones, y es el SGBD quien se encargará de mantenerlas.
- *Mejora en la seguridad.* La seguridad de la base de datos consiste la protección de la base de datos frente a usuarios no autorizados. Sin unas buenas medidas de seguridad, la integración de datos en los sistemas de bases de datos hace que éstos sean más vulnerables que en los sistemas de ficheros. Sin embargo, los SGBD permiten mantener la seguridad mediante el establecimiento de claves para identificar al personal autorizado a utilizar la base de datos. Las autorizaciones se pueden realizar a nivel de operaciones, de modo que un usuario puede estar autorizado a consultar ciertos datos pero no a actualizarlos, por ejemplo.
- *Mejora en la accesibilidad a los datos.* Muchos SGBD proporcionan lenguajes de consulta o generadores de informes que permiten al usuario hacer cualquier tipo de consulta sobre los datos, sin que sea necesario que un programador escriba una aplicación que realice tal tarea.
- *Mejora en la productividad.* El SGBD proporciona muchas de las funciones estándar que el programador necesita escribir en un sistema de ficheros. A nivel básico, el SGBD proporciona todas las rutinas de manejo de ficheros típicas de los programas de aplicación. El hecho de disponer de estas funciones permite al programador centrarse mejor en la función específica requerida por los usuarios, sin tener que preocuparse de los detalles de implementación de bajo nivel. Muchos SGBD también proporcionan un entorno de cuarta generación consistente en un conjunto de herramientas que simplifican, en gran medida, el desarrollo de las aplicaciones que acceden a la base de datos. Gracias a estas herramientas, el programador puede ofrecer una mayor productividad en un tiempo menor.

- *Mejora en el mantenimiento* gracias a la independencia de datos. En los sistemas de ficheros, las descripciones de los datos se encuentran inmersas en los programas de aplicación que los manejan. Esto hace que los programas sean dependientes de los datos, de modo que un cambio en su estructura, o un cambio en el modo en que se almacena en disco, requiere cambios importantes en los programas cuyos datos se ven afectados. Sin embargo, los SGBD separan las descripciones de los datos de las aplicaciones. Esto es lo que se conoce como independencia de datos, gracias a la cual se simplifica el mantenimiento de las aplicaciones que acceden a la base de datos.
- *Aumento de la concurrencia*. En algunos sistemas de ficheros, si hay varios usuarios que pueden acceder simultáneamente a un mismo fichero, es posible que el acceso interfiera entre ellos de modo que se pierda información o, incluso, que se pierda la integridad. La mayoría de los SGBD gestionan el acceso concurrente a la base de datos y pueden garantizar que no ocurran problemas de este tipo.
- *Mejora en los servicios de copias de seguridad y de recuperación ante fallos*. Muchos sistemas de ficheros dejan que sea el usuario quien proporcione las medidas necesarias para proteger los datos ante fallos en el sistema o en las aplicaciones. Los usuarios tienen que hacer copias de seguridad cada día, y si se produce algún fallo, utilizar estas copias para restaurarlos. En este caso, todo el trabajo realizado sobre los datos desde que se hizo la última copia de seguridad se pierde y se tiene que volver a realizar. Sin embargo, los SGBD actuales funcionan de modo que se minimiza la cantidad de trabajo perdido cuando se produce un fallo.

La integración de los datos y la existencia del SGBD también plantean ciertos inconvenientes, como los que se citan a continuación.

- *Alta complejidad*. Los SGBD son conjuntos de programas muy complejos con una gran funcionalidad. Es preciso comprender muy bien esta funcionalidad para poder sacar un buen partido de ellos.
- *Gran tamaño*. Los SGBD son programas complejos y muy extensos que requieren una gran cantidad de espacio en disco y de memoria para trabajar de forma eficiente.
- *Coste económico del SGBD*. El coste de un SGBD varía dependiendo del entorno y de la funcionalidad que ofrece. Por ejemplo, un SGBD para un ordenador personal puede costar 500 €, mientras que un SGBD para un sistema multiusuario que dé servicio a cientos de usuarios puede costar entre 10000 y 100000 €. Además, hay que pagar una cuota anual de mantenimiento que suele ser un porcentaje del precio del SGBD. En los últimos años han surgido SGBD libres (*open source*) que ofrecen una gran funcionalidad y muy buenas prestaciones.

- *Coste del equipamiento adicional.* Tanto el SGBD, como la propia base de datos, pueden hacer que sea necesario adquirir más espacio de almacenamiento. Además, para alcanzar las prestaciones deseadas, es posible que sea necesario adquirir una máquina más grande o una máquina que se dedique solamente al SGBD. Todo esto hará que la implantación de un sistema de bases de datos sea más cara.
- *Coste de la conversión.* En algunas ocasiones, el coste del SGBD y el coste del equipo informático que sea necesario adquirir para su buen funcionamiento es insignificante comparado al coste de convertir la aplicación actual en un sistema de bases de datos. Este coste incluye el coste de enseñar a la plantilla a utilizar estos sistemas y, probablemente, el coste del personal especializado para ayudar a realizar la conversión y poner en marcha el sistema. Este coste es una de las razones principales por las que algunas empresas y organizaciones se resisten a cambiar su sistema actual de ficheros por un sistema de bases de datos.
- *Prestaciones.* Un sistema de ficheros está escrito para una aplicación específica, por lo que sus prestaciones suelen ser muy buenas. Sin embargo, los SGBD están escritos para ser más generales y ser útiles en muchas aplicaciones, lo que puede hacer que algunas de ellas no sean tan rápidas como antes.
- *Vulnerable a los fallos.* El hecho de que todo esté centralizado en el SGBD hace que el sistema sea más vulnerable ante los fallos que puedan producirse.