

# Tema1. Introducción a las Bases de Datos

---

## Contenido

1	Introducción.....	2
2	Sistemas de archivos frente a bases de datos .....	2
2.1	Sistemas de archivos.....	3
2.1.1	Organización de archivos .....	3
2.1.2	Desventajas del uso de sistemas de archivos.....	4
2.2	Sistemas de bases de datos.....	5
2.2.1	Ventajas de los sistemas de bases de datos.....	6
2.2.2	Desventajas de los sistemas de bases de datos .....	7
3	Lenguajes de definición y manipulación de datos .....	8
4	Abstracción de la información. Arquitectura de una BD .....	9
5	Independencia de los datos.....	11
6	Sistemas de gestión de bases de datos (SGBD) .....	12
6.1	Funciones de un SGBD .....	12
6.2	Interacción del sistema operativo con el SGBD .....	13
6.3	Componentes de los SGBD.....	13
6.4	Lenguajes de la base de datos y diccionario de datos .....	14
6.5	El gestor de la base de datos.....	14
6.6	El administrador y los usuarios de la base de datos.....	15
6.7	Clasificación de los SGBD: según el modelo lógico.....	16
6.8	Clasificación de los SGBD: según el número de usuarios y según el número de sitios18	
6.9	Clasificación de los SGBD: según el coste y según el propósito .....	19
7	Tipos de usuarios de un SGBD .....	20

## 1 Introducción

En este módulo vamos a estudiar las **Bases de Datos** y su uso en el desarrollo de Aplicaciones Informáticas. Y vamos a comenzar desde el principio, con los conceptos y definiciones básicas. Empezaremos hablando de lo que las Bases de Datos contienen: Información.

Existen muchas definiciones para el significado de la palabra informática, pero todas coinciden en que **la informática es la ciencia que estudia el tratamiento automático y racional de la información.**

Es decir, el fin último de la informática es el tratamiento de la información.

La información tratada puede ser volátil, esto es, información que una vez tratada no es necesario almacenar y por lo tanto es desechada, se borra. O por el contrario, como ocurre **a menudo, es necesario el almacenamiento de la información para su futura recuperación, consulta y tratamiento.**

Es así como surge la necesidad de buscar cómo almacenar la información de manera que permita un tratamiento eficiente de la misma.

En un principio, el procesamiento de datos se basaba en sistemas de archivos o ficheros.

**Un archivo o fichero es una colección de información (datos relacionados entre sí) localizada o almacenada como una unidad en algún medio de almacenamiento.**

Un ejemplo de uso de archivos sería el fichero con todos los datos de los clientes de un banco, es decir, el conjunto de registros de clientes de un banco almacenados en un dispositivo de memoria secundaria.

Existen diferentes clasificaciones atendiendo a distintos criterios:

- por su función,
- por los elementos del archivo,
- por los tipos de acceso,
- etc.

Más adelante veremos la clasificación según el tipo de acceso, por tener algunas implicaciones con las bases de datos.



## 2 Sistemas de archivos frente a bases de datos

Las bases de datos se usan para almacenar los datos y los ficheros se usan también para ese fin.

Realmente cualquier base de datos está formada a fin de cuentas por ficheros, integrados y relacionados de cierta forma para facilitar el trabajo a los programadores, de forma que se puedan olvidar de muchos de los detalles del almacenamiento y la gestión de esos ficheros.

Se siguen usando los ficheros individuales, ajenos a una base de datos, por parte de las aplicaciones. Aunque realmente en las aplicaciones de gestión su uso es cada vez menor, y podríamos decir que es algo residual.

**Usar sistemas de archivos o bases de datos para nuestra aplicación, es algo que dependerá en gran medida del tipo de aplicación, ya que cada opción tiene sus características propias, que aportan ventajas e inconvenientes según qué tipo de problemas queramos resolver.**

Vamos a contrastar las características de los sistemas de archivos frente a las bases de datos.

## 2.1 Sistemas de archivos

**Un sistema de ficheros o archivos es un conjunto de programas que prestan servicio a los usuarios finales. Cada programa define y maneja sus propios datos.**

Los sistemas de ficheros surgieron al tratar de informatizar el manejo de los archivadores manuales con objeto de proporcionar un acceso más eficiente a los datos.

El **procesamiento tradicional** de datos se caracteriza por:

- una **proliferación de ficheros** en sistemas de archivos,
- específico cada uno de ellos **para una determinada aplicación**,
- la cual era implementada **para satisfacer una demanda particularizada**, es decir,
- **se creaba la estructura específica para dar solución a una demanda concreta, en lugar de establecer un sistema centralizado** en donde almacenar todos los datos de la organización o empresa, se escogió un modelo descentralizado en el que cada sección o departamento almacena y gestiona sus propios datos.

Esto propiciaba:

- un ambiente **desordenado y heterogéneo**,
- en el que se encontraban **diferentes programas, diferentes programadores, diferentes formatos y diferentes lenguajes**,
- todos ellos **orientados más a los procedimientos que a los datos**,
- y además esos procedimientos solían ser **muy específicos** de las distintas demandas.
- Todo esto ocasionaba el que los programas de aplicación se desarrollaban de forma independiente para cada demanda lo que daba lugar a que en ocasiones los datos se duplicaran, con toda una serie de inconvenientes que de ello se derivan (aumento de las necesidades de almacenamiento, valores contradictorios para un mismo dato almacenado en ficheros distintos, etc.)

### 2.1.1 Organización de archivos

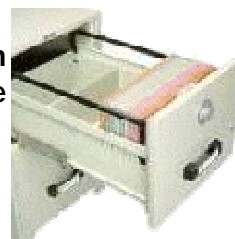
NO todos los ficheros tienen la misma estructura NI la misma forma de almacenar y tratar los datos que contienen.

**Las organizaciones de archivos representan la forma en la que se disponen los datos para su almacenamiento en un fichero en un dispositivo de almacenamiento secundario.**

Tenemos cuatro tipos:

**Secuencial:**

1. **Los registros se almacenan y graban en el mismo orden en el que fueron introducidos (uno detrás de otro).**
2. Para acceder a un registro hay que leer previamente todos los anteriores.
3. Para insertar hay que hacerlo al final y no es posible borrarlo.
4. Es posible utilizar archivos auxiliares para poder hacer estas operaciones aunque son muy complejas.
5. Se pueden utilizar tanto cintas de almacenamiento secuencial como discos de acceso directo.



6. Aunque es la organización más simple, su acceso y utilización son muy lentas y complejas.
7. Hay que evitarla siempre que se pueda.

#### **Directo o Aleatorio:**

1. **Un registro se almacena en una posición determinada del dispositivo de almacenamiento (disco) calculada a partir de la aplicación de un algoritmo o utilizando otros métodos al campo clave del registro**
2. Para acceder al registro sólo hay que conocer su clave, averiguar su posición y acceder al registro directamente.
3. Sólo puede ser usado en un dispositivo de acceso directo (sobre cintas, por ejemplo, no es posible)
4. Se puede acceder a la información de manera directa o secuencial.
5. El problema fundamental es la definición del algoritmo de cálculo de posiciones que debe permitir un reparto homogéneo por el disco.
6. El tratamiento de los registros es muy rápido tanto en acceso como en inserción y borrado.

#### **Indexado:**

1. Utilizamos un **fichero de índices** (sólo con el **campo clave**, de acceso secuencial) que nos indica la posición de los datos buscados en otro fichero de datos (acceso directo).

#### **2.1.2 Desventajas del uso de sistemas de archivos**

Los sistemas de archivos tienen unas características que los hacen poco apropiados según para qué aplicaciones, y en concreto para las aplicaciones de gestión.

El procesamiento de datos mediante sistemas de archivos genera una serie de inconvenientes, entre los que destacaremos los siguientes:

1. **Redundancia de información.** Un mismo dato puede estar repetido varias veces, en varios ficheros. Esto aumenta el coste de almacenamiento. Además complica las actualizaciones de los datos, ya que cada modificación debe afectar a todas las copias del dato.
2. **Inconsistencia de los datos.** La redundancia genera la posibilidad de que se produzca este problema. Ocurre cuando tenemos almacenadas dos copias del mismo dato y, cada una tiene valores distintos, debido a que se ha actualizado en un caso pero no en otro.
3. **Dificultad en el acceso a los datos.** El procesamiento de datos mediante sistemas de archivos no permite un acceso eficiente, y en general requiere más tiempo que el acceso a las bases de datos.
4. **Dificultad en la reestructuración de la información.** Resulta complicado añadir nuevos campos en ficheros existentes. Ante una necesidad nueva (previamente imprevista) de acceso a los datos existen dos alternativas:
  1. Implementar un nuevo programa que obtenga los datos a partir de los ficheros, o bien
  2. Extraer los datos manualmente a partir de la salida de algún otro programa existente.

5. **Tratamiento ineficaz de los datos.** No se pueden modificar ni actualizar todos los datos al mismo tiempo.
6. **Dificultad del tratamiento de los ficheros en su conjunto** debido a la incompatibilidad de los lenguajes de programación.
7. **Dependencia excesiva del formato (aislamiento de datos).** La información en los ficheros está en varios formatos no estándares. Estos formatos deberán ser conocidos **Sólo se pueden realizar las consultas que se han tenido en cuenta a la hora de escribir los programas de aplicación**, las consultas son limitadas por el número de programas de aplicación creados (número de consultas = número de programas de aplicación).
8. **Descentralización de los datos** (por no estar todos integrados en la misma colección).
9. **Las actualizaciones y modificaciones de los ficheros no son compartidas por toda la organización** lo cual lleva a grandes confusiones.

Debido a estas desventajas, para acceder a los sistemas de ficheros es necesario desarrollar unos programas escritos especialmente para ellos. Como los lenguajes de programación pueden ser distintos e incompatibles, hacen que los ficheros sean difíciles de procesar de manera conjunta. Esto, en numerosas ocasiones, dio lugar a que en las organizaciones hubiera una proliferación de programas de aplicación para resolver todo tipo de consultas, hasta el punto de desbordar al departamento de proceso de datos, que no daba abasto para validar, mantener y documentar dichos programas.

Resumiendo, podemos afirmar que **en el procesamiento de datos mediante sistemas de archivos existe una fuerte dependencia entre las aplicaciones y la organización física de los datos.**

## 2.2 Sistemas de bases de datos

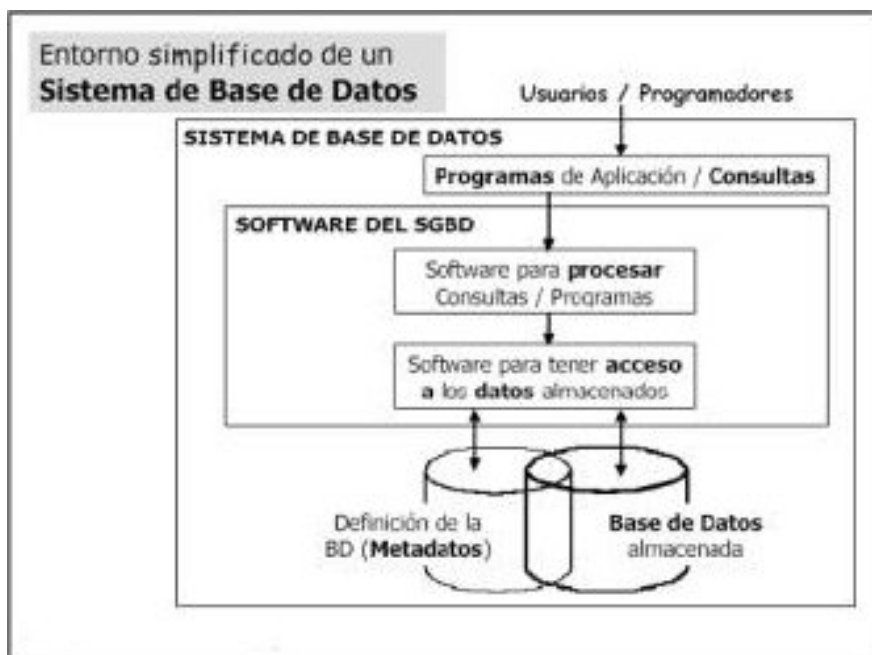
Para solventar los problemas de los sistemas de archivos a la hora del tratamiento de la información surge a mediados de los años 60 un concepto nuevo, **las bases de datos.**

En 1970 Codd propuso el **modelo relacional**. Este modelo es el que ha marcado la línea de investigación durante muchos años y, aunque actualmente han surgido **modelos orientados a objetos**, se sigue utilizando mucho el modelo relacional.

**Base de datos es un conjunto exhaustivo no redundante de datos estructurados organizados independientemente de su utilización y su implementación en máquina, accesibles en tiempo real y compatibles con usuarios concurrentes con necesidad de información diferente.**

Como podemos apreciar las **diferencias** más importantes entre las bases de datos y los sistemas de archivos son:

- Una Base de Datos es un conjunto exhaustivo **no redundante** de datos, mientras que una de las desventajas del uso de sistemas de archivos es la redundancia de los datos.
- En la base de datos los datos están organizados **independientemente de su utilización y su implementación en máquina**. Si recordamos, en los sistemas de archivos los datos se organizan y se estructuran en función de la utilización de los mismos.
- En las bases de datos **los datos son compatibles con usuarios concurrentes con necesidad de información diferente**. En el sistema de archivos los datos se estructuran en función de la necesidad de información, es decir, puede surgir incompatibilidad de los datos según las necesidades de los usuarios.



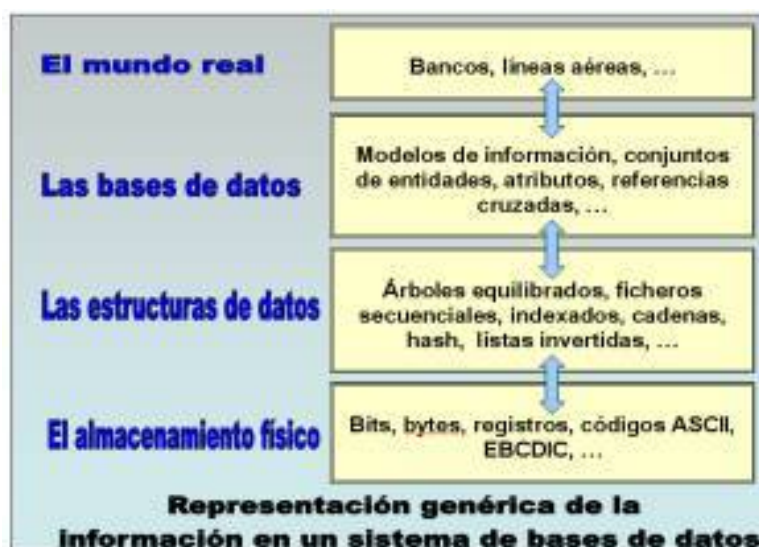
**Las bases de datos requieren básica y fundamentalmente un software de gestión que facilite las operaciones y las interfaces con los usuarios.**

Esto es, como veremos más adelante, **el Sistema de Gestión de Bases de Datos** (S.G.B.D. o en inglés, DBMS-Data Base Management System).

### 2.2.1 Ventajas de los sistemas de bases de datos

Como hemos visto, un sistema de bases de datos es:

- un **sistema de información** orientado hacia los datos,
- que pretende recuperar y almacenar la información de manera eficiente y cómoda.
- Surge en un intento de resolver las dificultades del procesamiento tradicional de datos,
- teniendo en cuenta que los datos suelen ser independientes de las aplicaciones.



Veamos qué ventajas aportan los sistemas de bases de datos respecto a los sistemas de archivos.

- **Independencia de los datos respecto de los procedimientos.** El usuario tiene una visión abstracta de los datos, sin necesidad de ningún conocimiento sobre la implementación de los ficheros de datos, índices, etc. Esto supone un gran ahorro en los costes de programación, de forma que la modificación de la estructura de los datos no suponga un cambio en los programas y viceversa. Sin ella, el mantenimiento de la base de datos ocuparía el 50% de los recursos humanos dedicados al desarrollo de cualquier aplicación.
- **Disminución de las redundancias** y en consecuencia,
- **Disminución** de la posibilidad de que se produzca **inconsistencia de datos**.
- **Mayor integridad de los datos.**
- **Mayor disponibilidad de los datos.**
- **Mayor seguridad de los datos.**
- **Mayor privacidad de los datos.**
- **Mayor eficiencia** en la recogida, codificación y entrada en el sistema.
- **Interfaz con el pasado y futuro:** una base de datos debe estar abierta a reconocer información organizada físicamente por otro software.
- **Compartición de los datos.** Los datos deben poder ser accedidos por varios usuarios simultáneamente, teniendo previstos procedimientos para salvaguardar la integridad de los mismos.

### 2.2.2 Desventajas de los sistemas de bases de datos

Con todas las ventajas que tiene el usar sistemas de bases de datos frente a sistemas de archivos, ¿por qué utilizar estos últimos?

Pues la contestación es que hasta ahora no existe nada perfecto en informática, y el uso de los sistemas de bases de datos tiene también sus **desventajas**.

Podríamos destacar las siguientes:

- **Desventajas relativas a la implantación:**
  - Instalación costosa en equipos y software.
  - Ausencia de estándares que facilite su uso.
  - Instalación larga y difícil.
  - Falta de rentabilidad a corto plazo.
- **Desventajas relativas a los usuarios:**
  - Necesidad de formación de un personal especializado.

Entonces, ¿cuándo usar un sistema de archivos y cuándo un sistema de bases de datos a la hora de plantearnos una aplicación?

Generalizando, **usaremos el sistema de archivos cuando la cantidad de datos a guardar sea tan reducida que no justifique las desventajas del uso de los sistemas de bases de datos.**

Un **ejemplo** típico sería realizar una aplicación de agenda en la cual sólo hay que guardar campos de texto ordenados por hora, día, mes y año. Aquí no sería recomendable el uso de un SGBD.



### 3 Lenguajes de definición y manipulación de datos

Hemos comentado que almacenar los datos en sistemas de bases de datos proporciona mayor independencia de los datos respecto de las aplicaciones, mayor independencia respecto a la máquina física en que se guardan, y mayores posibilidades a la hora de compartir esos datos.

Las herramientas que se usan para conseguir estas características tan deseables para el usuario son fundamentalmente los **lenguajes de definición y manipulación de datos**.

Los **SGBD** proporcionan a los usuarios diferentes tipos de lenguajes de bases de datos para interactuar con ellos. Existen distintas clasificaciones de este tipo de lenguajes. Nosotros vamos a usar la que los divide en **lenguajes de definición** de datos y **lenguajes de manipulación** de datos. A continuación pasamos a presentar cada uno de ellos:

#### Lenguajes de definición de datos.

Más conocidos por su acrónimo anglosajón **DDL (Data Definition Language)**. Los lenguajes de definición de datos (DDL) permiten:

- Especificar el **esquema de la base de datos**,
- Modificar la estructura del esquema,
- Especificar las condiciones de integridad
- Mejorar el acceso a la información.

Con estos lenguajes se obtiene el **esquema conceptual** de la base de datos. Al ejecutar instrucciones del DDL los datos se almacenan o se consultan en el **diccionario de datos**. Este diccionario es una estructura que almacena **metadatos** (información acerca de los datos). Este diccionario es fundamental, ya que para cualquier operación es lo primero que debe consultarse.

En muchos SGBD el DDL es también utilizado para definir esquemas internos y esquemas externos (**vistas**).

En algunos SGBD se separa el lenguaje de definición de almacenamiento (ADL) y el lenguaje de definición de vistas (VDL) que son usados para definir esquemas internos y externos respectivamente.

#### Lenguajes de manipulación de datos.

Más conocidos por su acrónimo anglosajón **DML (Data Manipulation Language)**, o por otros nombres como lenguajes de acceso de datos, de gestión de datos o de consulta.

Los lenguajes de manipulación de datos (DML) permiten realizar sobre los datos las operaciones de:

- **Consulta**,
- **Actualización**,
- **Inserción** y
- **Borrado**.

#### Tipos de DML

Los DML se pueden clasificar según distintos criterios.

**Según la forma de indicar la información** que queremos obtener:



- **Lenguajes procedimentales.** Permiten especificar en el momento de recuperar información no sólo qué información se desea recuperar sino también cómo se desea hacer la operación. Ej. IMS y resto de SGBD jerárquicos o en red.
- **Lenguajes declarativos o no procedimentales.** Sólo permiten especificar qué información se desea recuperar, pero no el modo de hacerlo. Ej. SQL en los SGBD relacionales.

**Según la forma de utilizar las sentencias del DML:**

1. **Modo Conversacional.** Se dice que un lenguaje se utiliza en modo conversacional si el usuario puede ejecutar las sentencias de modo interactivo a través de un intérprete
2. **Modo Diferido.** Se dice que un lenguaje se utiliza en modo diferido cuando las sentencias quedan pospuestas en el tiempo, incluyendo todo un grupo de ellas en algún fichero para ejecutarlas por lotes.

**Según la forma que las sentencias tienen de recuperar la información:**

- **De bajo nivel o "de procedimiento":**
  - Las instrucciones recuperan registro a registro y se necesita procesar los datos usando bucles.
  - Deben estar embebidos en un lenguaje de programación.
- **De alto nivel o "de no procedimiento":**
  - Las instrucciones recuperan un conjunto de registros.
  - Permite introducir instrucciones interactivamente.
  - Las instrucciones también pueden estar embebidas en un lenguaje de programación. Un lenguaje de este tipo es SQL, en el cual profundizaremos a lo largo de este módulo.

Se dice que el lenguaje de manipulación de datos SQL es **declarativo** porque se especifica **qué** datos obtener y **no cómo** obtenerlos.

Otros términos referidos a DML habitualmente usados son:

- **Lenguajes de consulta:** Así se llama a los DML que se usan interactivamente.
- **Lenguaje anfitrión:** El que permite tener embebidas instrucciones del DML
- **Sublenguaje de datos,** sería el DML embebido en otro lenguaje.

## 4 Abstracción de la información. Arquitectura de una BD

Una **base de datos** es en esencia una colección de archivos relacionados entre sí, de la cual los usuarios pueden extraer información sin considerar las fronteras de los archivos.

Un **objetivo** importante es proporcionar a los usuarios una visión abstracta de los datos, es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos.

Para que el sistema sea manejable, los datos se deben extraer eficientemente. Esto implica la utilización de estructuras de datos complejas que se ocultan al usuario, de modo que se proporciona **abstracción de la información**.

Existen diferentes **niveles de abstracción** para simplificar la interacción de los usuarios con el sistema:

- Nivel **Interno**, el del almacenamiento físico.

- Nivel **Conceptual**, el del administrador, y
- Nivel **Externo**, el del usuario.

A continuación explicamos en qué consiste cada uno de estos niveles.

#### Nivel físico o nivel interno.

Este nivel tiene un esquema interno. Es la representación del nivel más bajo de abstracción. En éste **se describe en detalle la forma de almacenar los datos en los dispositivos de almacenamiento**, es decir, los datos y sus relaciones a nivel físico del almacenamiento secundario (por ejemplo, mediante señaladores o índices para el acceso aleatorio a los datos). El nivel interno emplea un modelo de datos físicos y describe todos los detalles para su almacenamiento, así como los caminos de acceso para la base de datos.

#### Nivel conceptual.

Este nivel tiene un **esquema conceptual**. Es el siguiente nivel más alto de abstracción. **Describe qué datos son almacenados realmente en la base de datos y las relaciones que existen entre los mismos**, describe la base de datos completa en términos de su estructura de diseño. El nivel conceptual de abstracción lo usan los administradores de bases de datos, quienes deben decidir qué información se va a guardar en la base de datos.

#### Nivel de visión o nivel externo.

Este nivel incluye varios **esquemas externos** o **vistas de datos**. Es el nivel más alto de abstracción. **Es lo que el usuario final puede visualizar del sistema terminado**, describe sólo una parte de la base de datos al usuario acreditado para verla. El sistema puede proporcionar muchas visiones para la misma base de datos.

Ha de resaltarse el que los tres esquemas o niveles de datos no son más que descriptores de los datos.

#### Los únicos datos que existen realmente están en el nivel físico.

En un SGBD basado en la arquitectura de tres niveles cada grupo de usuarios hace referencia exclusivamente a su nivel externo. Por tanto, el SGBD debe

- Transformar una solicitud expresada en términos de un nivel externo en una solicitud expresada en términos del nivel conceptual y luego
- Transformar una solicitud en el nivel interno que se procesará sobre la base de datos almacenada.

El proceso de transformar solicitudes y resultados de un nivel a otro se denomina **correspondencia o transformación (mapping)**



## 5 Independencia de los datos

Antes hemos visto algunas de las ventajas de las Bases de Datos con respecto a los sistemas de ficheros:

- **La base de datos tiene un formato estándar que es conocido por los programadores,**
- Además, este **formato es totalmente independiente de los procesos** para los que vayan a ser utilizados dichos datos.
- Dicho de una forma más clara, **con las bases de datos conseguimos almacenar los datos en un formato estándar independiente de los procesos y de la manera en que se desee presentar la información.**
- Así, **tampoco es necesario tener distintos archivos para distintos formatos**, sino que nos puede bastar con uno sólo.
- **Los programadores no tienen por qué preocuparse del formato en el que están almacenados los datos**, sólo programar los accesos (edición, modificación, inserción y borrado) a los mismos.

Por lo que se deduce que **esto es una forma más eficiente** de tratamiento de los datos.

Definimos **independencia de datos** como la **capacidad para modificar el esquema de un nivel del sistema de la base de datos sin tener que modificar el esquema del nivel inmediato superior.**

Podemos definir **dos tipos de independencia** de los datos. La independencia de los datos puede ser lógica y física.

- **Por independencia lógica** de los datos se entiende que los cambios en el esquema lógico no deben afectar a los esquemas externos que no utilicen los datos modificados. Es decir, **es la capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación.** Esto es, podemos modificar el esquema conceptual por ejemplo, para ampliar la base de datos (añadiendo un nuevo tipo de registro o un elemento de datos) o para reducir la base de datos (eliminando un tipo de registro o un elemento de datos). Esta modificación en el esquema conceptual no deberá afectar a los esquemas externos que sólo se refieran a los datos restantes. En resumen, después de una reorganización lógica del esquema conceptual los programas de aplicación que hagan referencia a los elementos del esquema externo deberán funcionar igual que antes. Además, las restricciones podrán modificarse en el esquema conceptual sin afectar a los esquemas externos ni a los programas de aplicación.
- **Por independencia física** de los datos se entiende que el esquema lógico no se vea afectado por cambios realizados en el esquema interno, correspondientes a modos de acceso, etc. Es decir, **es la capacidad de modificar el esquema interno sin tener que alterar el esquema conceptual (o los externos).**

## 6 Sistemas de gestión de bases de datos (SGBD)

Las **bases de datos** requieren básica y fundamentalmente un Software de Gestión que facilite las operaciones y las interfaces con los usuarios. Esto es el Sistema de Gestión de Bases de Datos (**S.G.B.D.**).

En resumen, el SGBD ha de estar diseñado de forma que las ventajas que se han señalado como propias de las BD sean una realidad.

Las **operaciones típicas** que debe realizar un SGBD son las siguientes:

- Aquéllas que afectan a la totalidad de los datos.
  - Creación,
  - Reestructuración,
  - Consultas a la totalidad
- Las que tienen lugar sobre registros concretos, que suelen llamarse operaciones de Actualización:
  - Altas o inserciones,
  - Bajas o borrados,
  - Modificaciones
  - Consultas selectivas.

### 6.1 Funciones de un SGBD

Las funciones de un SGBD son:

#### A) Función de descripción o definición.

Esta función debe permitir al administrador de la BD **especificar los elementos de datos** que la integran, su **estructura** y las **relaciones** que existen entre ellos, las reglas de integridad semántica, los controles a efectuar antes de autorizar el acceso a la BD, etc., así como las características de tipo físico y las vistas lógicas de los usuarios.

Esta función, realizada por el **lenguaje de descripción o definición de datos (DDL)** propio de cada SGBD, debe suministrar los medios para definir las tres estructuras de datos o vistas (externa, conceptual e interna), especificando las características de los datos a cada uno de estos niveles.

- A **nivel interno**, se ha de indicar el espacio de disco reservado para la base de datos, la longitud de los campos, su modo de representación (**lenguaje para la definición de la estructura interna**).

- A **nivel conceptual** se proporcionan herramientas para la definición de las entidades y su identificación, atributos de las mismas, interrelaciones entre ellas, restricciones de integridad, etc.; es decir el **esquema de la base de datos (lenguaje para la definición de estructura lógico global)**.
- A **nivel externo**, se deben definir las vistas de los distintos usuarios a través del **lenguaje para la definición de estructuras externas**. Además, el SGBD se ocupará de la transformación de las estructuras externas orientadas a los usuarios a las estructuras conceptuales y de la relación de éstas y la estructura física.

### B) Función de manipulación.

Permite a los usuarios **buscar, añadir, suprimir o modificar los datos** de la base de datos, siempre de acuerdo con las especificaciones y las normas de seguridad dictadas por el administrador. Se llevará a cabo por medio de un **lenguaje de manipulación de datos (DML)** que facilita los instrumentos necesarios para la realización de estas tareas.

También se encargan de definir **la vista externa** de todos los usuarios de la base de datos o vistas parciales que cada usuario tiene de los datos definidos con el DDL.

### C) Función de utilización.

Esta función reúne todas las **interfaces que necesitan los diferentes usuarios** para comunicarse con la BD y proporciona un conjunto de procedimientos para el administrador. Incluye funciones de servicio como cambiar la capacidad de los ficheros, obtener estadísticas de utilización, cargar archivos, etc. y los relacionados con la seguridad física (copias de seguridad, rearranque en caso de caída del sistema, etc.) y de protección frente a accesos no autorizados. Esto lo realiza el **Lenguaje de Control de Datos (DCL)**.

## 6.2 Interacción del sistema operativo con el SGBD

El funcionamiento del SGBD está interrelacionado con otros componentes de software, especialmente con el SO. Aunque para su estudio tendríamos que elegir un SGBD específico, hay algunos aspectos de funcionamiento comunes a todos.

Las aplicaciones se apoyan en las facilidades del SGBD, y estos, a su vez, se apoyan en los métodos del acceso del SO.

## 6.3 Componentes de los SGBD

Para realizar todas las funciones descritas anteriormente, es necesario que el SGBD cuente con una serie de componentes:

1. **Lenguajes de la base de datos.**
2. **El diccionario de datos.**
3. **El gestor de la base de datos.**

4. **El administrador de la base de datos**
5. **Usuarios de la base de datos.**

## 6.4 Lenguajes de la base de datos y diccionario de datos

Vamos a empezar con los **lenguajes de la base de datos**:

Los distintos lenguajes que se utilizan en todas las tareas relacionadas con la creación, mantenimiento y uso de la base de datos:

- a. El lenguaje de definición de datos (DDL).
- b. El lenguaje de manipulación de datos (DML).
- c. El lenguaje de control de datos (DCL).

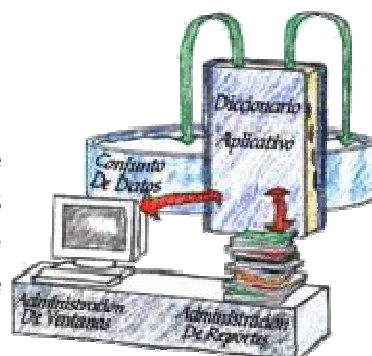
Ya en secciones anteriores se ha hablado de ellos, por lo que no es necesario decir nada más.

Continuamos con el **diccionario de datos**.

El diccionario de datos es un conjunto de archivos que contienen información acerca de los datos que se almacenan en la base de datos. Se trata de una "**metabase de datos**", es decir, una base de datos que contienen información sobre la base de datos (datos acerca de los datos)

En el diccionario de datos se encuentra almacenada:

- **La representación de los datos a los tres niveles de abstracción** (esquema lógico, físico y subesquemas externos de la base de datos). Existen un conjunto de reglas para transformar un esquema en otro (**mapa de reglas**).
- **Las restricciones de privacidad y acceso a los datos** definidas por el DDL y DCL.
- **Las reglas, normas o restricciones referentes a la seguridad** de los datos.
- **Otras informaciones** referentes a garantizar la integridad de los datos.



## 6.5 El gestor de la base de datos

El **gestor** de base de datos o **monitor**, es un componente software encargado de garantizar el correcto, seguro, íntegro y eficiente acceso y almacenamiento de los datos. Este componente es el encargado de proporcionar una interfaz entre los datos almacenados y los programas de aplicación que los manejan.

Puede verse el gestor de la base de datos como un intérprete entre el usuario y los datos. Toda operación que se quiere realizar "contra" la base de datos debe ser previamente autorizada

por el gestor de la misma, el cual, una vez interpretada y validada, o bien realiza la operación devolviendo el resultado de la misma al programa que lo solicitó o bien lo rechaza.

Los **principales componentes** del gestor de la base de datos son los siguientes:

- **Control de autorización.** Este módulo comprueba que el usuario tiene los permisos necesarios para llevar a cabo la operación que solicita.
- **Procesador de comandos.** Una vez que el sistema ha comprobado los permisos del usuario, se pasa el control al procesador de comandos.
- **Control de la integridad.** Cuando una operación cambia los datos de la base de datos, este módulo debe comprobar que la operación a realizar satisface todas las restricciones de integridad necesarias.
- **Optimizador de consultas.** Este módulo determina la estrategia óptima para la ejecución de las consultas.
- **Gestor de transacciones.** Este módulo realiza el procesamiento de las transacciones.
- **Planificador (scheduler).** Este módulo es el responsable de asegurar que las operaciones que se realizan concurrentemente sobre la base de datos tienen lugar sin conflictos.
- **Gestor de recuperación.** Este módulo garantiza que la base de datos permanece en un estado consistente en caso de que se produzca algún fallo.
- **Gestor de buffers.** Este módulo es el responsable de transferir los datos entre memoria principal y los dispositivos de almacenamiento secundario. A este módulo también se le denomina **gestor de datos**.

El gestor de la base de datos es responsable de garantizar:

- La **privacidad** de los datos,
- Su **seguridad**,
- Su **integridad**,
- El **acceso concurrente** sin pérdida de integridad y
- La **interacción con el sistema operativo** de forma que entienda los procedimientos definidos por el DML a través del denominado **procesador de consultas**.

## 6.6 El administrador y los usuarios de la base de datos

El **Administrador de la base de datos**:

Es una persona o grupo de personas encargadas de la función de administración de la base de datos.

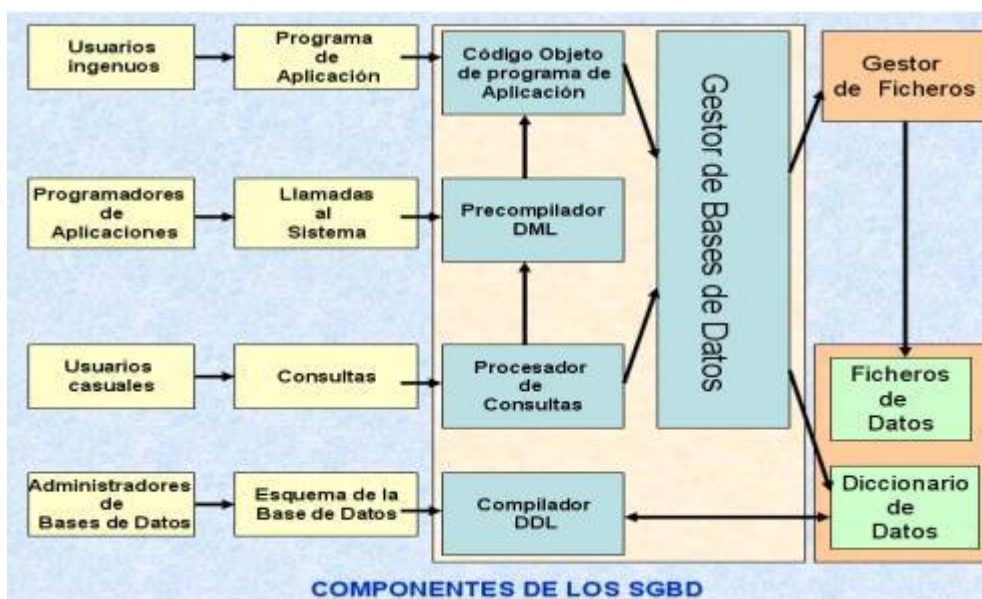


En el siguiente apartado "Tipos de usuarios de un sistema de gestión de base de datos" lo estudiaremos en profundidad.

Los **usuarios de la base de datos**:

Como un componente más de la base de datos, se pueden considerar a los usuarios, a los que estudiaremos en profundidad en el siguiente apartado, si bien ahora podemos hacer la siguiente división genérica de los mismos:

- **Usuarios normales o terminales:** interactúan con la base de datos a través de programas de aplicaciones. No son especialistas y sólo ven su vista externa correspondiente.
- **Usuarios técnicos:** son profesionales informáticos que desarrollan los programas de aplicación que van a ser utilizados por los usuarios normales de la base de datos.
- **Usuarios directivos:** que fijarán al administrador de la base de datos los objetivos de la base de datos para que respondan a los objetivos generales de la empresa. Estos además fijan los recursos de los que dispondrá la base de datos.



## 6.7 Clasificación de los SGBD: según el modelo lógico

El criterio principal que se utiliza para clasificar los SGBD es el modelo lógico en que se basan. Los modelos lógicos empleados con mayor frecuencia en los SGBD comerciales actuales son el relacional, el de red y el jerárquico. Algunos SGBD más modernos se basan en modelos orientados a objetos. Estudiaremos también otros criterios de clasificación.

¿Qué categorías establece la clasificación de los SGBD según el modelo lógico?

Básicamente tres:

El **modelo relacional** se basa en el concepto matemático denominado "relación", que gráficamente se puede representar como una tabla. En el modelo relacional, los datos y las

relaciones existentes entre los datos se representan mediante estas relaciones matemáticas, cada una con un nombre que es único y con un conjunto de columnas.

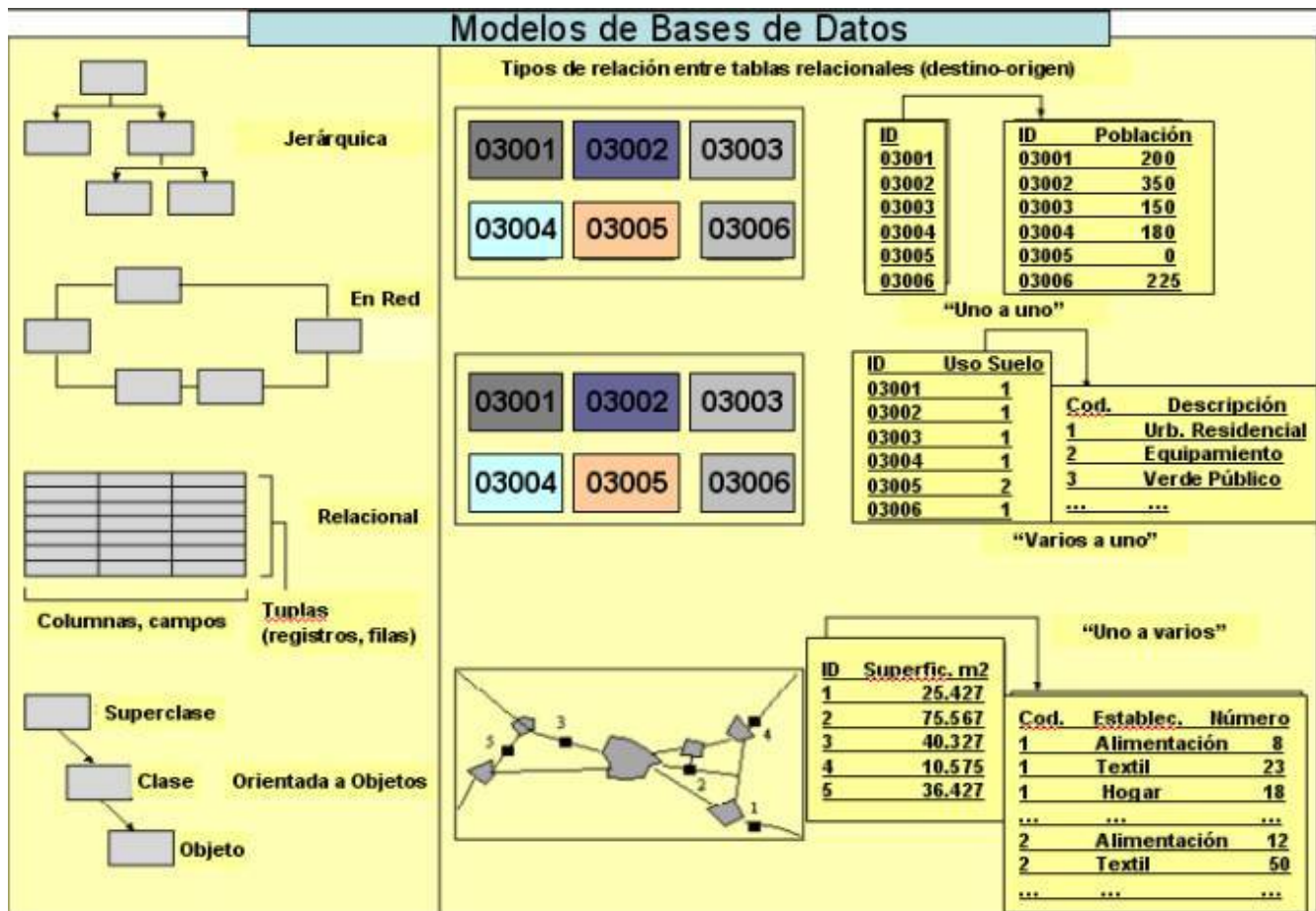
En el modelo relacional la base de datos es percibida por el usuario como un conjunto de tablas. Esta percepción es sólo a nivel lógico (en los niveles externo y conceptual de la arquitectura de tres niveles), ya que a nivel físico puede estar implementada mediante distintas estructuras de almacenamiento.

El **modelo de red** representa los datos como colecciones de registros y las relaciones entre los datos se representan mediante conjuntos, que son punteros en la implementación física. Los registros se organizan como un grafo: los registros son los nodos y los arcos son los conjuntos. El SGBD de red más popular es el sistema IDMS.

El **modelo jerárquico** es un tipo de modelo de red con algunas restricciones. De nuevo los datos se representan como colecciones de registros y las relaciones entre los datos se representan mediante conjuntos. Sin embargo, en el modelo jerárquico cada nodo puede tener un solo padre. Una base de datos jerárquica puede representarse mediante un árbol: los registros son los nodos, también denominados segmentos, y los arcos son los conjuntos. El SGBD jerárquico más importante es el sistema IMS.

La mayoría de los SGBD comerciales actuales están basados en el modelo relacional, mientras que los sistemas más antiguos estaban basados en el modelo de red o el modelo jerárquico. Estos dos últimos modelos requieren que el usuario tenga conocimiento de la estructura física de la base de datos a la que se accede, mientras que el modelo relacional proporciona una mayor independencia de datos. Se dice que el **modelo relacional es declarativo** (se especifica qué datos se han de obtener) y los **modelos de red y jerárquico son navegacionales** (se especifica cómo se deben obtener los datos).

El **modelo orientado a objetos** define una base de datos en términos de objetos, sus propiedades y sus operaciones. Los objetos con la misma estructura y comportamiento pertenecen a una clase, y las clases se organizan en jerarquías o grafos acíclicos. Las operaciones de cada clase se especifican en términos de procedimientos predefinidos denominados métodos. Algunos SGBD relacionales existentes en el mercado han estado extendiendo sus modelos para incorporar conceptos orientados a objetos. A estos SGBD se les conoce como sistemas **objeto-relacionales**



## 6.8 Clasificación de los SGBD: según el número de usuarios y según el número de sitios

Puedes comprobar que la clasificación según el número de usuarios a los que se da servicio, es extremadamente simple, y que no tenemos mucho que comentar sobre ella.

- Clasificación **según el número de usuarios a los que se da servicio**:
  - Los sistemas **monousuario** sólo atienden a un usuario a la vez, y su principal uso se da en los ordenadores personales.
  - Los sistemas **multiusuario**, entre los que se encuentran la mayor parte de los SGBD, atienden a varios usuarios al mismo tiempo.
- Clasificación **según el número de sitios en los que está distribuida la BD**.
  - Casi todos los SGBD son **centralizados**: sus datos se almacenan en un solo ordenador. Los SGBD centralizados pueden atender a varios usuarios, pero el SGBD y la base de datos en sí residen por completo en una sola máquina.
  - En los SGBD **distribuidos** la base de datos real y el propio software del SGBD pueden estar distribuidos en varios sitios conectados por una red.

- Los SGBD **distribuidos homogéneos** utilizan el mismo SGBD en múltiples sitios. Una tendencia reciente consiste en crear software para tener acceso a varias bases de datos autónomas preexistentes almacenadas en SGBD **distribuidos heterogéneos**. Esto da lugar a los SGBD **federados o sistemas multibase** de datos en los que los SGBD participantes tienen cierto grado de autonomía local.

Muchos SGBD distribuidos emplean una arquitectura cliente-servidor.



## 6.9 Clasificación de los SGBD: según el coste y según el propósito

Vamos con la primera de esas dos clasificaciones.

- Clasificación de los SGBD **según el coste**:
  - La mayor parte de los paquetes de SGBD cuestan entre 10.000 y 100.000 euros.
  - Los sistemas monousuario más económicos para microcomputadores cuestan entre 100 y 3.000 euros.
  - En el otro extremo, los paquetes más completos cuestan más de 100.000 euros.
- Clasificación de los SGBD según el propósito:
  - También los SGBD pueden ser de propósito general o de propósito específico.
  - Cuando el rendimiento es fundamental, se puede diseñar y construir un SGBD de propósito especial para una aplicación específica, y este sistema no sirve para otras aplicaciones. Muchos sistemas de reservas de líneas aéreas son SGBD de propósito especial y pertenecen a la categoría de **sistemas de procesamiento de transacciones en línea (OLTP)**, que deben atender un gran número de transacciones concurrentes sin imponer excesivos retrasos.
  - Los SGBD **de propósito general**, al contrario que los anteriores, están diseñados para atender a cualquier tipo de aplicaciones. A cambio, nunca

pueden ofrecer la máxima eficiencia para cada caso, sino que deben adoptar soluciones de compromiso que satisfagan razonablemente a cualquier aplicación o usuario, para cualquier tipo de datos almacenados.

## 7 Tipos de usuarios de un SGBD

La variedad de usuarios de un sistema de base de datos es tal que no existe una clasificación exacta de los mismos, pero vamos a intentar ponerte sobre la pista, haciendo una clasificación en la que se distinguen las siguientes clases:

### **Administrador de la base de datos (ABD).**

También conocidos por su acrónimo anglosajón DBA (Data Base Administrator).

**Tiene el control centralizado de la base de datos y es el responsable de su buen funcionamiento. Es el encargado de autorizar el acceso a la base de datos, de coordinar y vigilar su utilización y de adquirir los recursos software y hardware que sean necesarios.**

¿Cuáles son sus responsabilidades?:

- La **definición del esquema canónico o lógico** de la base de datos. Es decir, la codificación mediante sentencias del DDL del conjunto de definiciones que representan las características del problema.
- La **definición del esquema físico** de la base de datos. Es decir, el conjunto de sentencias en DDL que definen las estructuras de almacenamiento y los métodos de acceso a la información.
- La **definición de subesquemas o visiones externas** o de usuario de la base de datos.
- El **control de la privacidad de los datos**, mediante la **concesión de privilegios a usuarios**. Además, hay que tener en cuenta que la Ley Orgánica 15/1999 de 13 de diciembre de Protección de Datos de Carácter Personal, la LOPD, tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales. Por tanto, el administrador de la base de datos deberá velar por la privacidad de esos datos almacenados en los ficheros de la organización.
- **Mantenimiento de los esquemas**, es decir, introducción de cualquier modificación en los distintos esquemas o nuevas definiciones.
- La **especificación de los procedimientos necesarios para el mantenimiento de la seguridad de los datos**, es decir, cómo recuperarse de un fallo de la base de datos.

Las herramientas que utiliza el ABD para realizar estas funciones son:

- Lenguajes de Definición de Datos (DDL).
- Utilidades del SGBD.
- Utilidades de monitorización.

- Herramientas de ayuda al diseño (CASE).
- Diccionario de datos.

### Diseñadores de bases de datos.

**Se encargan de identificar los datos que se almacenarán en la base de datos y de elegir las estructuras apropiadas para almacenar dichos datos.**

Generalmente, estas tareas se realizan antes de que se implemente la base de datos y se carguen los datos. Los diseñadores tienen la responsabilidad de comunicarse con los futuros usuarios de la base de datos con el fin de presentar un diseño que satisfaga los requerimientos que den respuesta a las necesidades de los usuarios. Es muy común que los diseñadores formen parte del personal del ABD y suelen asumir otras responsabilidades una vez acabado el diseño de la base de datos.

### Operadores y personal de mantenimiento.

Forman parte del personal del ABD y **son los responsables del funcionamiento y mantenimiento reales del entorno software y hardware del sistema de base de datos.**

### Usuarios finales.

Los **usuarios finales** son las personas cuyo trabajo requiere acceder a la base de datos para consultarla, actualizarla y generar informes. La base de datos existe para que ellos la utilicen. Existen las siguientes categorías de usuarios finales:

- **Usuarios finales ocasionales.** Estos acceden de vez en cuando a la base de datos pero es muy probable que necesiten información diferente en cada ocasión. Utilizan un lenguaje de consulta de bases de datos avanzado para especificar sus solicitudes y suelen ser gerentes de nivel medio o alto u otras personas que examinan la base de datos ocasionalmente. Estos usuarios aprenden unos pocos recursos del SGBD que pueden utilizar de forma repetida.
- **Usuarios finales simples o paramétricos.** Suelen ser la porción más considerable de la totalidad de los usuarios finales. La **función principal de su trabajo gira en torno a consultas y actualizaciones constantes de la base de datos, utilizando tipos estándar de consultas y actualizaciones.** Como ejemplo de este tipo de usuarios tenemos a los cajeros de las entidades bancarias, los cuales revisan saldos y realizan reintegros y depósitos de dinero. Los encargados de reservas de los hoteles, líneas aéreas...etc. Este tipo de usuario no necesita saber de los recursos del SGBD más que entender los tipos de transacciones estándar diseñadas e implementadas para que ellos las usen.
- **Usuarios finales avanzados.** Suelen ser personal altamente cualificado que está suficientemente familiarizado con los recursos del SGBD como para implementar sus aplicaciones de forma que cumplan sus complejos requerimientos. Este tipo de usuario intenta conocer la mayoría de los recursos del SGBD para satisfacer sus complejos requerimientos.

- **Usuarios finales autónomos.** Mantienen bases de datos personales mediante la utilización de paquetes de programas comerciales que cuentan con interfaces de fácil uso basados en menús o en gráficos. Un ejemplo puede ser el usuario de un paquete fiscal que almacena varios datos financieros personales para usos fiscales. Este tipo de usuarios adquieren gran habilidad para utilizar un paquete de software específico.

**Analistas de sistemas y Programadores de aplicaciones** Aunque **no son considerados como usuarios** en sí de una base de datos, sí es conveniente conocer la figura de los mismos para una mejor comprensión de la estructura de un SGBD.

- **Los analistas de sistemas** determinan los requerimientos de los usuarios finales, sobretodo de los simples o paramétricos, y desarrollan especificaciones para transacciones programadas que satisfagan dichos requerimientos.
- **Los programadores de aplicaciones** implementan esas especificaciones en forma de programas y luego prueban, depuran, documentan y mantienen estas transacciones programadas.

Para realizar estas tareas, los analistas y programadores deben conocer a la perfección toda la gama de capacidades del SGBD.