

# **UT1. SISTEMAS DE ALMACENAMIENTO DE LA INFORMACIÓN INDICE**

1. Ficheros. Tipos de ficheros
  - 1.1 Ficheros de texto
  - 1.2 Ficheros binarios
2. Sistemas de ficheros
3. Bases de Datos
4. Sistemas Gestores de bases de datos
5. Componentes de los SGBD
  - 5.1 Lenguajes de los SGBD
  - 5.2 El diccionario de datos
  - 5.3 El administrador de la BD

- 6. Arquitectura de los sistemas de bases de datos
- 7. Opciones de funcionamiento de un SGBD
  - 7.1 SGBD monocapa
  - 7.2 SGBD de dos capas
  - 7.3 SGBD de tres o más capas
- 8. Sistemas Gestores de Bases de Datos Comerciales
  - 8.1 Licencias de software
  - 8.2 SGBD relacionales de código cerrado
  - 8.3 SGBD relacionales de código abierto

# 1.Ficheros. Tipos de ficheros

Los ficheros son estructuras de información para poder almacenar datos. Suelen tener un nombre y una extensión que determina el formato de la información que contiene.

Tradicionalmente, los ficheros se han clasificado de muchas formas:

- Según su contenido: texto, binario.
- Según su organización: secuencial, directa, indexada. La organización de un fichero dicta la forma en que se han de acceder a los datos.
- Según su utilidad: maestro, movimiento, histórico. La utilidad de un fichero indica qué uso se va a hacer de él.

Hoy en día estas dos últimas clasificaciones han quedado en desuso. Actualmente un sistema operativo trata un fichero desde dos puntos de vista:

- Según su contenido: texto o binario.
- Según su tipo: imágenes, ejecutables, videos, etc.

# 1.1 Ficheros de texto

Los ficheros de texto se denominan planos ya que no tiene ningún tipo de formato. Únicamente contienen texto, saltos de línea y la marca de fin de fichero. Suelen estar escritos en formato ASCII, cada carácter se representa mediante un valor numérico. Estos ficheros son generados mediante un editor de texto, **NO** un "procesador" de textos como Word.

Los ficheros de texto pueden tener una extensión que identifica su contenido, por ejemplo:

- Ficheros de configuración, suelen tener extensión, .ini, .inf, .conf,...
- Ficheros de código fuente, suelen tener extensión, .sql, .c, .java,...
- Ficheros de páginas web, suelen tener extensión, .html, .php, .xml ,.css,...

## 1.2 Ficheros binarios

Los ficheros binarios son los que contienen información de cualquier tipo codificada en binario cuyo fin es el almacenamiento y procesamiento en ordenadores.

Por ejemplo los archivos informáticos que almacenan texto formateado o fotografías, así como los archivos ejecutables que contienen programas, vídeos. Así tenemos:

- Procesadores de texto: .doc, .odt, ...
- Imagen: .bmp, .gif, .jpg, .png, ...
- Ejecutables: .com, .exe, ...
- Vídeo: .avi, .mov, .mpg, ...
- Comprimidos: .tar, .zip, ...



## 2. Sistemas de ficheros

Antes de aparecer los SGBD (década de los setenta), la información se trataba y se gestionaba usando los típicos sistemas de gestión de ficheros. Mediante varios programas que manejaban ficheros el usuario obtenía información. Este sistema de información permitía informatizar el uso de los ficheros manuales que se utilizaban en las empresas.

En los sistemas de ficheros, la definición de los datos se encuentra codificada dentro de los programas de aplicación en lugar de almacenarse de forma independiente, y además el control del acceso y la manipulación de los datos viene impuesto por los programas de aplicación.

## **Inconvenientes de los sistemas de ficheros**

- Redundancia de datos. Se produce duplicidad de datos desperdiciando espacio de almacenamiento y se crea inconsistencia cuando ficheros con los mismos datos no contienen la misma información.
- Dependencia de datos. Si se modifica la estructura de algún fichero se necesitará alterar todos los programas que lo utilicen.
- Formatos de ficheros incompatibles. Los ficheros sólo pueden ser utilizados en el lenguaje programación con el que se hayan creado.

Todos estos inconvenientes hacen posible el fomento y desarrollo de SGBD.

Se dice que los sistemas de ficheros son sistemas orientados al proceso, ya que, en ellos, los datos están subordinados a los procesos. Cada aplicación dispone de sus propios ficheros, donde están los datos almacenados. La descripción de estos datos y sus interrelaciones aparecen en los programas que hacen uso de ellos.

Por lo que surge la idea de separar los datos contenidos en los archivos de los programas que los manipulan, es decir, que se pueda modificar la estructura de los datos de los archivos sin que por ello se tengan que modificar los programas con los que trabajan. Se trata de estructurar y organizar los datos de forma que se pueda acceder a ellos con independencia de los programas que los gestionan.

# 3.Bases de datos

Una base de datos es un conjunto de datos almacenados entre los que existen relaciones lógicas y ha sido diseñada para satisfacer los requerimientos de información de una empresa u organización.

La base de datos es un gran almacén de datos que se define una sola vez y que se utiliza al mismo tiempo por muchos departamentos y usuarios. 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.

Los sistemas de bases de datos separan la definición de la estructura de los datos, de los programas de aplicación y almacenan esta definición en la base de datos. Si se añaden nuevas estructuras de datos o se modifican las ya existentes, los programas de aplicación no se ven afectados ya que no dependen directamente de aquello que se ha modificado.

Las bases de datos son sistemas orientados a los datos, estos se almacenan independientemente de las aplicaciones.

# 4. Sistemas Gestores de bases de datos

El objetivo primordial de un gestor es proporcionar eficiencia y seguridad a la hora de extraer o almacenar información en las B.D. Los sistemas gestores de BBDD están diseñados para gestionar grandes bloques de información, que implica tanto la definición de estructuras para almacenamiento como de mecanismos para la gestión de la información.

El SGBD es una aplicación que permite a los usuarios definir, crear y mantener la BD y proporciona un acceso controlado a la misma.

## 5. Componentes de los SGBD

Los SGBD son paquetes de software muy complejos que deben proporcionar una serie de servicios que van a permitir almacenar y explotar los datos de forma eficiente. Los componentes principales son los siguientes:

- Lenguajes de los SGBD.
- El diccionario de datos
- El administrador de la BD



## 5.1 Lenguajes de los SGBD

Todos los SGBD ofrecen lenguajes e interfaces apropiadas para cada tipo de usuario: administradores, diseñadores, programadores de aplicaciones y usuarios finales.

Este lenguaje denominado SQL, está estandarizado por la ISO, es decir, todas las bases de datos que soporten SQL, deben tener la misma sintaxis a la hora de aplicar el lenguaje. Se divide en cuatro sublenguajes, que permiten al SGBD cumplir con las funcionalidades requeridas.

- Lenguaje de manipulación de datos, DML: se utiliza para leer y actualizar los datos de la BD. Es el utilizado por los usuarios para realizar consultas, inserciones, eliminaciones y modificaciones.
- Lenguaje de definición de datos, DDL: se utilizan para especificar el esquema de la BD, las vistas de los usuarios y las estructuras de almacenamiento. Es el que define el esquema conceptual y el esquema interno. Lo utilizan los diseñadores y los administradores de la BD

- Lenguaje de control de datos, DCL: permiten al administrador gestionar el acceso a los datos contenidos en la base de datos.
- Lenguaje de control de transacciones, TCL: el propósito de este lenguaje es permitir el control de las transacciones.

## 5.2 El diccionario de datos

El diccionario de datos es el lugar donde se deposita información acerca de todos los datos que forman la BD. Es una guía en la que se describe la BD y los objetos que la forman. En una BD relacional, proporciona información acerca de:

- La estructura lógica y física de la BD.
- Las definiciones de todos los objetos de la BD: tablas, vistas, índices, disparadores, procedimientos, funciones, etc.
- El espacio asignado y utilizado por los objetos.
- Los valores por defecto de las columnas de las tablas.
- Información acerca de las restricciones de integridad.
- Los privilegios y roles otorgados a los usuarios.
- Auditoría de información, como los acceso a los objetos.

## 5.3 El administrador de la BD

En los sistemas de gestión de BD actuales existen diferentes categorías de usuarios. Estas categorías se caracterizan porque cada una de ellas tiene una serie de privilegios o permisos sobre los objetos que forman la BD. En los sistemas Oracle las categorías más importantes son:

- Los usuarios de la categoría DBA(Data base Administrador), cuya función es precisamente administrar la base y que tienen, el nivel más alto de privilegios.
- Los usuarios de la categoría RESOURCE, que pueden crear sus propios objetos y tienen acceso a los objetos para los que se les ha concedido permiso.
- Los usuarios del tipo CONNECT, que solamente pueden utilizar aquellos objetos para los que se les ha concedido permiso de acceso.

El DBA tiene gran responsabilidad ya que posee el máximo de privilegios. Será el encargado de crear los usuarios que se conectarán a la BD. En la administración de una BD siempre hay que procurar que haya el menor número de administradores, a ser posible una sola persona.

El objetivo principal de un DBA es garantizar que la BD cumple con los fines previstos por la organización, lo que incluye una serie de tareas como:

- Instalar SGBD en el sistema informático.
- Crear las BBDD que se vayan a gestionar.
- Crear y mantener el esquema de la BD.
- Crear y mantener las cuentas de usuario de la BD.
- Arrancar y parar SGBD, y cargar las BBDD con las que se ha de trabajar.

- Colaborar con el administrador del S.O. en las tareas de ubicación, dimensionado y control de los archivos y espacios de disco ocupados por el SGBD.
- Colaborar en las tareas de formación de usuarios.
- Establecer estándares de uso, políticas de acceso y protocolos de trabajo diario para los usuarios de la BD.
- Suministrar la información necesaria sobre la BD a los equipos de análisis y programación de aplicaciones.
- Efectuar tareas de explotación como:
  - ☐ Vigilar el trabajo diario colaborando en la información y resolución de dudas de los usuarios de la BD.
  - ☐ Controlar los acceso, tasas de uso, cargas en el servidor, anomalías, etc.
  - ☐ Efectuar copias de seguridad periódicas de la BD.

- ❑ Restaurar la BD después de un incidente material a partir de las copias de seguridad.
- ❑ Estudiar las auditorías del sistema para detectar anomalías, intentos de violación de la seguridad, etc.
- ❑ Ajustar y optimizar la BD mediante el ajuste de sus parámetros, y con la ayuda de las herramientas de monitorización y de las estadísticas del sistema.

En su gestión diaria, el DBA suele utilizar una serie de herramientas de administración de la BD. Estas herramientas han adquirido sofisticadas prestaciones y facilitan en gran medida la realización de trabajos que, hasta no hace demasiado, requerían de arduos esfuerzos por parte de los administradores.

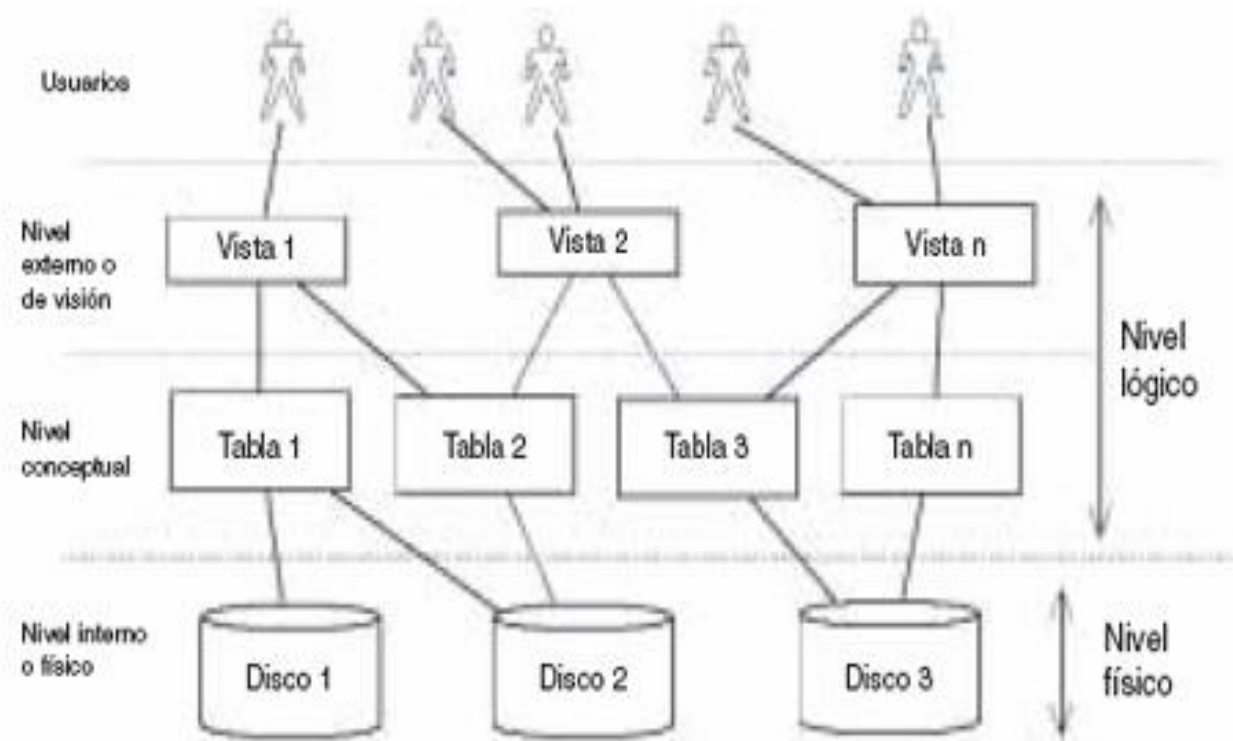


## 6. Arquitectura de los sistemas de bases de datos

En 1975, el comité ANSI-SPARC (American National Standard Institute-Standards Planning and requirements Comité) propuso una arquitectura de tres niveles para los SBGD cuyo objetivo principal era el de separar los programas de aplicación de la BD física. En esta arquitectura el esquema de una BD se define en tres niveles de abstracción distintos:

- Nivel interno o físico: el más cercano al almacenamiento físico, es decir, tal y como están almacenados en el ordenador. Describe la estructura física de la BD mediante un esquema interno. Este esquema se especifica con un modelo físico y describe los detalles de cómo se almacenan físicamente los datos: los archivos que contienen la información, su organización, los métodos de acceso a los registros, los tipos de registros, la longitud, los campos que los componen, etcétera.

- Nivel externo o de visión: es el más cercano a los usuarios, es decir, es donde se describen varios esquemas externos o vistas de usuarios. Cada esquema describa la parte de la BD que interesa a un grupo de usuarios en este nivel se representa la visión individual de un usuario o de un grupo de usuarios.
- Nivel conceptual: describe la estructura de toda la BD para un grupo de usuarios mediante un esquema conceptual. Este esquema describe las entidades, atributos, relaciones, operaciones de los usuarios y restricciones, ocultando los detalles de las estructuras físicas de almacenamiento.



Con la arquitectura a tres niveles se introduce el concepto de independencia de datos, se definen dos tipos de independencia:

- Independencia lógica: la capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación. Se podrá modificar el esquema conceptual para ampliar la BD o para reducirla. Por ejemplo, si se elimina una entidad, los esquemas externos que no se refieran a ella no se verán afectados.
- Independencia física: la capacidad de modificar el esquema interno sin tener que alterar ni el esquema conceptual, ni los externos. Por ejemplo, se pueden reorganizar los archivos físicos con el fin de mejorar el rendimiento de las operaciones de consulta o de actualización, o se pueden añadir nuevos archivos de datos porque los que había se han llenado. La independencia física es más fácil de conseguir que la lógica, pues se refiere a la separación entre las aplicaciones y las estructuras físicas de almacenamiento.

Hoy en día se definen más niveles de modo que se habla de cinco niveles en realidad. Empezando desde el más cercano al usuario:

- Nivel externo. Tal cual se explicó antes. En realidad los esquemas de este nivel son los últimos que se crean y lo hacen programadoras/es y analistas bajo la dirección de las/os analistas.
- Nivel conceptual. Con la misma idea indicada anteriormente. En realidad el esquema (los planos) conceptual de la base de datos es lo primero que se diseña por los o las analistas o diseñadores de la misma utilizando un modelo para realizar los esquemas. El modelo Entidad/Relación sigue siendo el modelo más popular.

- Nivel lógico. Acerca más el esquema anterior a la física de la base de datos. En este nivel se hace referencia a estructuras lógicas del tipo de SGBD a manejar (tablas, filas, columnas por ejemplo en el modelo relacional de bases de datos, el modelo lógico más utilizado). Este nivel sigue siendo manejado por los analistas. En muchos casos (aunque ciertamente es peligroso) los diseñadores/as de la base de datos empiezan por este nivel saltándose el anterior. En la actualidad el modelo relacional sigue siendo el modelo más habitual para crear esquemas a nivel lógico.
- Nivel interno. Es el primero en el proceso de modelado de la base de datos que se realiza sobre el software gestor de la base de datos (teniendo en cuenta que lo externo, las aplicaciones, se crean más tarde). Usa el lenguaje de la base de datos para crear las estructuras de datos definidas en el nivel lógico. Este nivel lo maneja el administrador de la base de datos (o DBA).

- Nivel físico. Se refiere a como se organizarán los datos en el disco, en que ordenadores se crea la base de datos, si es distribuida o no, sistema operativo necesario, estructura de directorios y archivos, configuración de servidores y sistema operativo, política de copia de seguridad,...

La persona encargada de definir todo esto es el administrador de la base de datos.

# 7.Opciones de funcionamiento de un SGBD

## 7.1 SGBD monocapa

Se trata de Sistemas Gestores instalados en una máquina desde la que se conectan los propios usuarios y administradores. Es un modelo que solo se utiliza con bases de datos pequeñas y poca cantidad de conexiones.



## 7.2 SGBD de dos capas

Es el modelo tipo cliente/servidor. La base de datos y el sistema gestor se alojan en un servidor al cual se conectan los usuarios desde máquinas clientes. Un software de comunicaciones se encarga de permitir el acceso a través de la red. Los clientes deben instalar el software cliente de acceso según las instrucciones de configuración del administrador. Hay dos posibilidades:

- Arquitectura cliente/servidor único. Un solo servidor gestiona la base de datos todos los clientes se conectan a él para realizar las peticiones a la base de datos.
- Arquitectura cliente/multiservidor. La base de datos se distribuye entre varios servidores. El cliente no sabe realmente a que servidor se conecta, el software de control de comunicaciones se encargará de dirigirle al servidor adecuado. De forma lógica, es como si se tratara de un solo servidor aunque físicamente sean muchos.

## 7.3 SGBD de tres o más capas

En este caso entre el cliente y el servidor hay al menos una capa intermedia (puede haber varias). Esa capa (o capas) se encargan de recoger las peticiones de los clientes y luego de comunicarse con el servidor (o servidores) para recibir la respuesta y enviarla al cliente. El caso típico es que la capa intermedia sea un servidor web, que recibe las peticiones a través de páginas web; de este modo para conectarse a la base de datos, el usuario solo requiere un navegador web, que es un software muy habitual en cualquier máquina. Este modelo es el que más se está potenciando en la actualidad por motivos de seguridad y portabilidad de la base de datos.

# 8. Sistemas Gestores de Bases de Datos Comerciales

## 8.1 Licencias de software

El gurú del software libre, Richard Stallman considera al software propietario (software cuyo uso y explotación se rige por un contrato propio de la empresa), software privativo, puesto que dicho software no permite examinar el código fuente y, por lo tanto, impide modificar el mismo y adaptarlo a nuevas funcionalidades.

Por otro lado, él mismo define al software que sí permite este proceso, software libre. En cualquier caso ambos tipos de software no tienen por qué ser gratuitos, es decir la diferencia no es la gratuidad (aunque sí ocurre a menudo que el software de código abierto además suele ser gratuito) sino la libertad de utilizar el código fuente del software.

Una definición quizá menos tendenciosa es la que diferencia al software en: software de código abierto y software de código cerrado u oculto. Esta diferencia de software se debe a dos formas diferentes de hacer negocio con él; los defensores del código cerrado argumentan que es lógico protegerle para evitar copiar su tecnología por parte de la competencia e incluso por razones de seguridad del mismo al no poder asegurar su correcto funcionamiento ante modificaciones de terceros.

Los defensores del otro modelo están a favor de la versatilidad del código abierto que permite poder modificar el código por parte de miles de programadores en todo el mundo que pueden compartir dichas mejoras y así mejorar enormemente y de manera dinámica el producto.

## 8.2 SGBD relacionales de código cerrado

Normalmente las licencias de uso de Sistemas Gestor es de Bases de Datos con código cerrado usan licencias tipo CLUF o EULA (en inglés), acrónimo contrato de licencia de usuario final.

En estas licencias, el usuario firma unas condiciones de uso del software, entre las que siempre figuran el hecho de no poder distribuir libremente el mismo y que está restringido a unas condiciones de trabajo concretas (por ejemplo el hecho de que normalmente sólo se pueda utilizar cada licencia en una sola máquina o por parte de un solo usuario). Ejemplos de SGBD de este tipo son:

- Oracle. Propiedad de Oracle Corporation. Es el SGBD más veterano y más influyente ya que la mayoría de mejoras al SQL original se desarrollaron para este SGBD. Sigue siendo uno de los SGBD comerciales más utilizados. Presume de su gran estabilidad y escalabilidad, un control avanzado de transacciones y de sus lenguajes internos de manejo, especialmente famoso es su lenguaje procedimental PL/SQL. Es un SGBD multiplataforma, que tiene certificación para instalarse en Linux (aunque sólo con los compatibles con Red Hat y con condiciones muy concretas de instalación).
- DB2. Propiedad de IBM, es una de las bases de datos comerciales más populares. Desarrollada para Windows, UNIX y Linux. Implementa XML de manera nativa y dispone de amplias facilidades de migración de datos (especialmente desde Oracle) así como uso de transacciones avanzadas.

- SQL Server (de Microsoft). Originalmente basado en el código del SGBD SyBase que Microsoft compró al propietario de SyBase, ahora es un SGBD distinto distribuido sólo para Windows y que compite con los dos anteriores. Dispone de una gran escalabilidad, estabilidad, uso de transacciones, entorno gráfico avanzado y de éxito entre los programadores de la plataforma .NET (también de Microsoft por su compatibilidad con esta).

Las tres son de las bases de datos más utilizadas en la actualidad por su contrastada potencia. Ninguna de las tres cumple completamente los estándares y aportan sus propios lenguajes y forma de trabajo. Además las tres disponen de versiones gratuitas para uso personal con base de datos más pequeñas.



## 8.3 SGBD relacionales de código abierto

Los SGBD más conocidos de código abierto son:

- MySQL. Inicialmente creada por la empresa MySQL AB, posteriormente comprada por Sun Microsystems que, a su vez, fue comprada por Oracle. Ha sido considerada como la principal SGBD de la comunidad de programadores de código abierto y de hecho en Internet sigue siendo la principal base de datos asociada a una aplicación web. Mantiene su licencia de tipo GPL, pero posee una segunda licencia cerrada para opciones de trabajo más avanzadas. Es muy popular por su histórica asociación con PHP, por su buena estabilidad, gran escalabilidad, e incluso uso de transacciones y lenguaje procedimental; además de ser un producto con infinidad de plataformas posibles para su instalación.

- PostgreSQL. Versión de código abierto basada en el producto Ingres de la Universidad de Berkeley. Usa licencia de tipo MIT (del Instituto Tecnológico de Massachusetts) que es una de las más libres, permite su modificación, redistribución incluso hacia otro tipo de licencias del tipo que sean, sin usar en ningún momento copyright. Está considerado como el SGBD de código abierto más potente y, sobre todo, más fidedigno con los estándares. Posee uso de transacciones avanzadas, lenguaje procedimental, gran estabilidad y escalabilidad. Hoy en día está considerada como la más potente de las bases de datos de código abierto y a partir de su núcleo se han creado otros productos libres de software base de datos.

- Firebird. Se trata de un SGBD liberado del producto comercial Interbase que era propiedad de Borland y que fue liberado con una variante de la licencia que usa Mozilla (MPL) que, a su vez, se basa en la licencia BSD. Tiene soporte transaccional avanzado, buena estabilidad pero no es muy escalable.
- Apache Derby. Anteriormente, IBM Cloudscape, usa licencia Apache que es poco restrictiva con las redistribuciones. Está programa en Java y pensada para ser utilizada en ese mismo lenguaje.