

# UD 1

## Almacenamiento de la Información

### Ficheros: planos, indexados, acceso directo, ...

Los **ficheros** o **archivos** son estructuras de almacenamiento de información creadas por un sistema operativo. Cada fichero tiene un nombre (y en algunos sistemas operativos una extensión que indica el tipo de fichero, por ejemplo en Windows).

Un fichero no es más que un conjunto de bits cuyo significado puede ser muy variado: texto, imagen, configuración de una aplicación, ...

### Tipos de ficheros según su contenido

#### Ficheros de texto y binarios

- **Fichero de texto:** es un archivo en el cual todos sus bits codifican caracteres de texto (letras, números, signos de puntuación, ...). Existen diferentes codificaciones estándar de caracteres de texto como **ASCII extendido** (cada carácter se codifica con 8 bits), u otros como **Unicode** (que permite además codificar idiomas como el japonés o el chino, donde cada símbolo se codifica con 16 bits).
  - Por ejemplo los ficheros con extensión **.txt** creados por el bloc de Notas de Windows o con Pluma en Linux, o con cualquier otro editor de texto (se puede leer con cualquier editor de texto el fichero creado con otro editor de texto).
- **Fichero binario:** en general, cualquier fichero que no sea de texto, pueden ser de muchos tipos

¿Un fichero docx de Microsoft Office es un fichero de texto o de datos? Piensa si solamente contiene caracteres o también datos de formato. ¿Se puede leer con el Bloc de Notas?

#### Tipos de ficheros de texto

- **Fichero plano:** es un archivo de texto (cada línea del archivo de texto contiene un registro) con campos separados por delimitadores (como comas o tabulaciones).
  - Por ejemplo, los ficheros **.csv** (pueden importarse fácilmente a una hoja de cálculo)

```
Año,Marca,Modelo,Descripción,Precio
1997,Ford,E350,"ac, abs, moon",3000.00
1999,Chevyr,Venture,Extended Edition,4900.00
1999,Chevy,Venture,"Extended Edition, Very Large",5000.00
1996,Jeep,Grand Cherokee,"MUST SELL! air, moon roof, loaded",4799.00
```

- **Fichero de texto enriquecido:** además de los caracteres tiene también codificados formatos y estructuras, pero se pueden leer y editar con un editor de texto (lo cual no se puede hacer con un fichero generado con un procesador de texto, como MS Office)
  - Por ejemplo los archivos **.rtf**, **.ps**, o **.tex**
- **Ficheros de código fuente:** código fuente en un lenguaje de programación o en otro lenguaje informático
  - Ejemplos: ficheros **.c** en lenguaje de programación C, o ficheros **.java** en lenguaje de programación Java
  - Ejemplos: ficheros **.sql** con scripts en lenguaje SQL, o ficheros **.bash** con scripts de Linux
- **Fichero de código fuente web:** son una caso particular del tipo anterior
  - Ejemplos: ficheros de formato de páginas web en lenguajes de marcas como los archivos **.html**, **.css**, **.xml**
  - Ejemplo: ficheros **.php** en lenguaje de programación PHP
- **Ficheros de configuración:** los utiliza el sistema operativo o una aplicación para leer el valor ciertos parámetros
  - Ejemplos: ficheros **.conf**, **.ini**, **.inf**, ....

## Tipos de ficheros binarios

- **Ejecutable:** arrancan una aplicación en un sistema operativo (han sido creados con un lenguaje de programación)
  - Por ejemplo, los ficheros **.exe** en Windows
- **Documento generado con un procesador de texto:** además del contenido del fichero, contienen información de formato y estructura
  - Ejemplos: ficheros **.odt** (LibreOffice), **.docx** (MS Office)
- **Imagen:** existen múltiples formatos
  - Ejemplos: **.jpg**, **.png**, **.svg**, **.bmp**, ...
- **Comprimidos:** existen varios estándares
  - Ejemplos: **.zip**, **.rar**, **.tar**, **.gz**, ...
- **Multimedia:** existen múltiples estándares
  - Ejemplos: **.mp4**, **.avi**, **.mov**, , ...

# Tipos de ficheros según su modo de acceso

- Ficheros secuenciales:

Los registros se escriben sobre el dispositivo de almacenamiento en posiciones físicamente contiguas, sin dejar huecos entre ellos, en el mismo orden en que hayan sido introducidos.

- Para acceder al registro  $n$  hay que recorrer los  $n-1$  registros anteriores.
- El acceso es secuencial, la lectura siempre se realiza hacia delante.
- Ejemplo: los **ficheros planos**

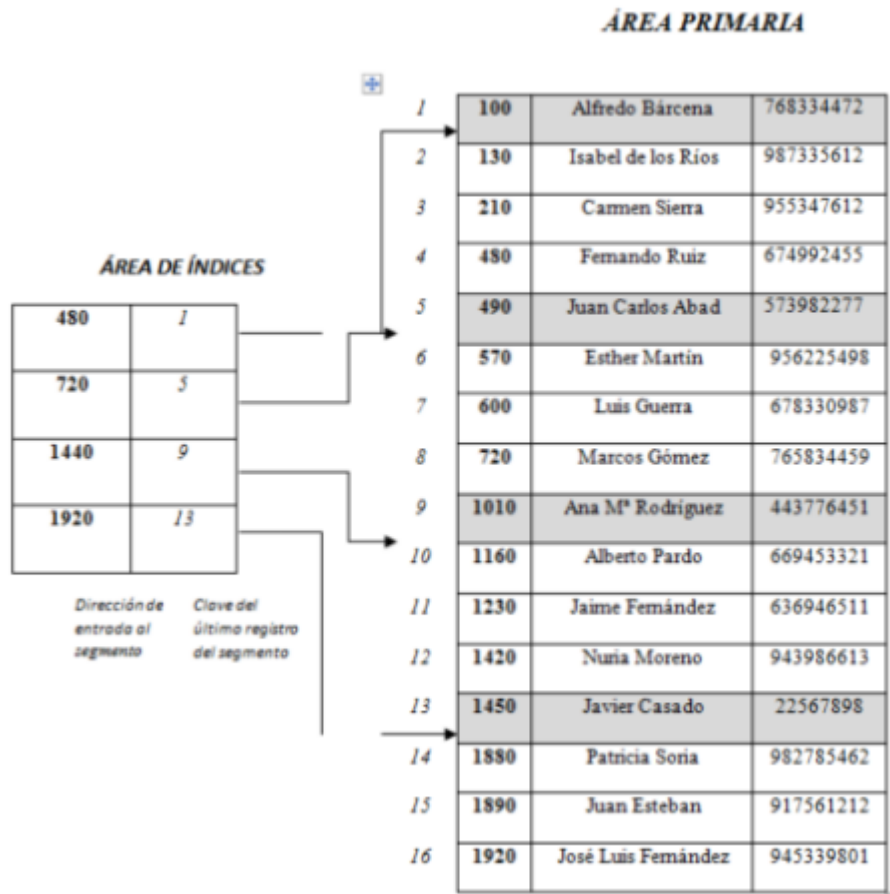
- Ficheros de acceso directo:

Se puede acceder a un registro a través de una clave que forma parte del registro como un campo más. La clave coincide con la dirección del registro en el fichero.

- La clave tiene que ser numérica y comprendida dentro del rango de valores de las direcciones.
- Es el método de acceso más rápido.

- Ficheros indexados:

Los registros con los datos se graban en **un fichero secuencial**, pero se pueden recuperar con acceso directo gracias a la utilización de **un fichero adicional**, llamado **de índices**, que contiene información de **la posición que ocupa cada registro** en el fichero de datos.



## Bases de Datos (BBDD). Conceptos, usos y tipos según el modelo de datos, la ubicación de la información

En los años 60 del siglo XX se llamó **bancos de datos** a los conjuntos complejos de **ficheros relacionados entre sí** y que eran usados por **varios procesos simultáneamente**. En los años 70 pasaron a llamarse **bases de datos** y así se les ha seguido llamando desde entonces.

El software de gestión de ficheros de los sistemas operativos tenía demasiados problemas:

1. No tenían previsto el tratamiento de las **relaciones entre ficheros** (que obligan a modificar un fichero cuando otro relacionado se ve modificado)
2. No permitían a **varios procesos** modificar el **mismo fichero simultáneamente**
3. Era muy difícil programar las aplicaciones para utilizar estos conjuntos de ficheros

Por estos motivos surgieron los **Sistemas Gestores de Bases de Datos** (SGBD).

Un **SGBD** puede definirse como un conjunto de sistemas de programas que permiten almacenar y examinar todo el conjunto de ficheros independientes que permiten modelar y manejar los datos de una organización (empresa, asociación, institución pública, ....).

Una base de datos (BDD) concreta modela objetos del mundo real y las relaciones entre diferentes objetos, de la siguiente manera:

- Cada **objeto** → Un **registro** (fila) en un **fichero**
- Cada **relación** entre dos objetos → un **enlace** entre los dos registros (uno en cada fichero)

## Conceptos

- **Dato**: fragmento de información concreta, tiene que ser de algún tipo
- **Tipo de Dato**: clase a la que pertenece cada dato. Ejemplos: datos numéricos, datos alfanuméricos, fecha, ...
- **Campo**: nombre de la columna en un registro de datos, cada campo tiene que ser de un tipo. Por ejemplo, el campo “Fecha de Nacimiento” tiene que ser del tipo fecha
- **Campo clave**: campo que identifica de manera única a cada registro. Por ejemplo el campo DNI de un registro de clientes.
- **Esquema**: definición de la estructura de una BDD. Contiene la definición de todos los elementos de una BDD, se puede conocer gracias al diccionario de metadatos
- **Consulta**: instrucción para extraer datos de una BDD
- **Vista**: consulta almacenada en la base de datos
- **Informe**: listado ordenado de campos y registros extraídos de la BDD en un formato fácil de leer
- **Guión o script**: conjunto de instrucciones para ejecutar de manera consecutiva sobre una BDD

## Usos de las BBDD

- Labores **administrativas**: registro de clientes, registro de productos, pedidos, facturas, ...
- Labores **contables**: registro de pagos, balances, patrimonio, ...
- Labores para **motores de búsqueda**: por ejemplo Google usa una BDD para almacenar toda la información conocida en Internet
- Labores **científicas**: por ejemplo la World Data Climate Center contiene una de las BDD mayores del mundo, (con información de datos climáticos)
- Labores de almacenamiento de **información de un sistema informático**: por ejemplo el registro interno de Windows, o la BDD de definiciones de virus de un antivirus
- **Bibliotecas**
- **Instituciones públicas**: censos, declaraciones de la renta, DNI, ...

## Tipos de BBDD según el modelo de datos

Los diferentes modelos de datos son diferentes abstracciones que permiten diferentes implementaciones eficientes de bases de datos.

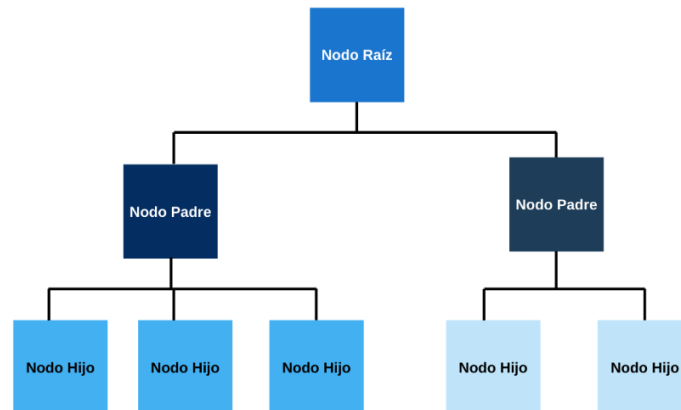
Algunos modelos usados para implementar bases de datos son:

- **Bases de datos jerárquicas**

Los datos se organizan en una estructura similar a un árbol (visto al revés) donde el nodo padre puede tener varios hijos.

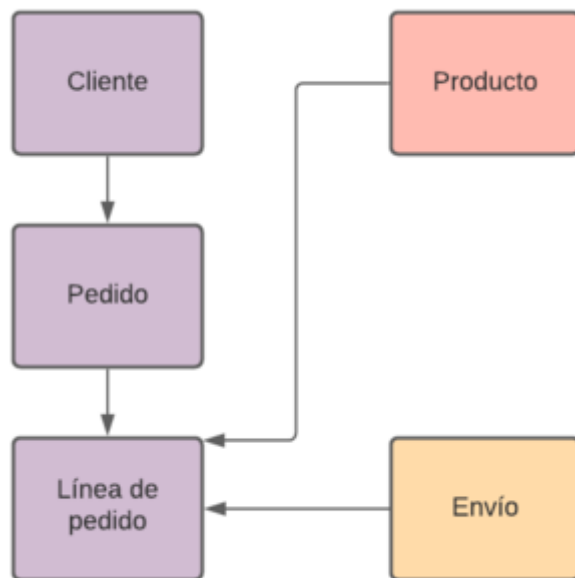
Principal inconveniente: mala gestión de la redundancia de datos, puesto que cuando un registro tiene relación con dos o más registros, debe almacenarse varias veces, puesto que un hijo no puede tener varios padres.

Ejemplo de SGBD jerárquico: Adabas



- Bases de datos en red

Cada nodo hijo puede tener varios nodos padres (lo cual evita la redundancia)

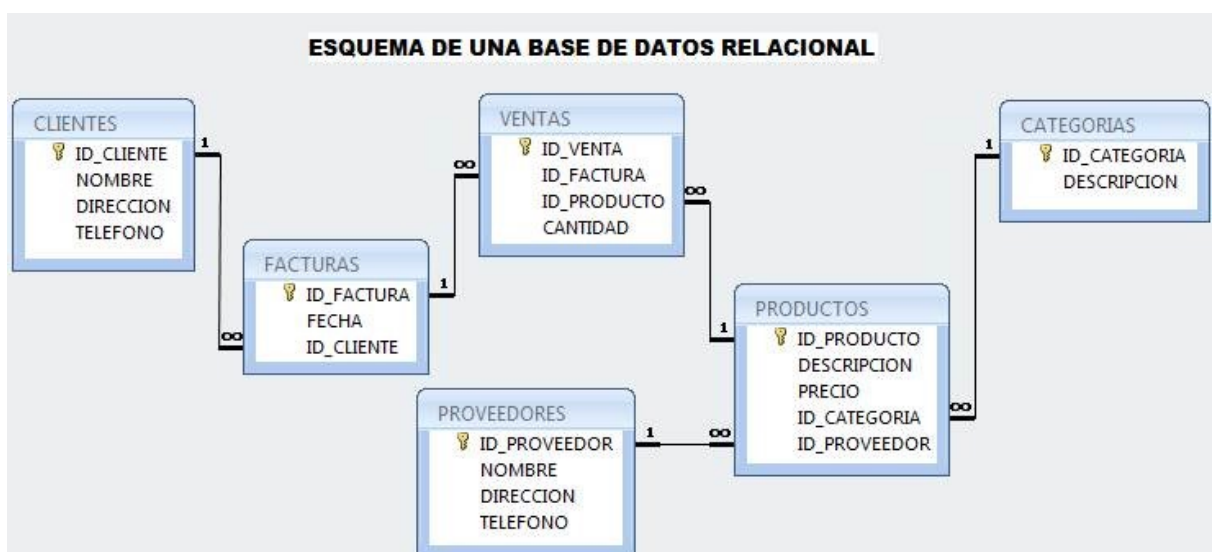


Principal inconveniente: resulta muy difícil administrar la información en la base de datos en red para un usuario final.

Ejemplo de SGBD en red: Raima Database Manager (RDM)

- Bases de datos relacionales

En este modelo la información se almacena en tablas donde cada registro (fila) de una tabla tiene diferentes campos (columnas) y permiten la interrelación entre diferentes tablas mediante claves (campos de una tabla que existen en otra tabla).



Los ficheros en los que se almacenan los datos de cada tabla no forman parte del modelo relacional. Es decir el usuario de la base de datos no tiene porqué conocer cómo se almacenan físicamente los datos, solamente necesita conocer las tablas y sus interrelaciones.

Es el **modelo de bases de datos más utilizado** desde los años 80 del siglo XX hasta hoy.

Los SGBD relacionales utilizan el **lenguaje SQL** para procesamiento de la información de una base de datos.

Ejemplo de SGBD relacional: MySQL

- **Bases de datos orientadas a objetos**

Permiten almacenar instancias de objetos de un lenguaje de programación que siga ese paradigma (Python, Java, C#, ...). Una característica de estas bases de datos es que las operaciones que se pueden realizar sobre los objetos almacenados también se almacenan y las puede definir el usuario de la base de datos.

Ejemplo de SGBD orientada a objetos: Zope Object Database (ZODB)

- **Bases de datos documentales**

Están constituidas por un conjunto de programas que almacenan, recuperan y gestionan datos de documentos o datos de algún modo estructurados (forman parte de las llamadas bases de datos NoSQL).

Están diseñadas alrededor de una noción abstracta de "Documento". Entre las codificaciones usadas en la actualidad se encuentran XML y JSON.

Ejemplo de SGBD documentales: MongoDB



# Sistemas gestores de bases de datos (SGBD): funciones, componentes y tipos

## Funciones de un SGBD

### 1. Definición de datos

Mediante un DDL (Lenguaje de Definición de Datos) el SGBD permite definir el esquema (diseño) de una base de datos: permite definir las estructuras y sus interrelaciones.

### 2. Manipulación de datos

Mediante un DML (Lenguaje de Manipulación de Datos) el SGBD permite:

- **Consultar** datos de la BDD
- **Insertar** nuevos datos en la BDD
- **Eliminar** datos de la BDD
- **Modificar** datos de la BDD

Los lenguajes DML pueden ser de dos tipos:

- **Procedimentales**: el usuario debe decir cómo adquirir los datos (además de qué datos va a utilizar). Ejemplo: el lenguaje NATURAL (usado por el SGDB Adabas)
- **No procedimentales**: el usuario solamente debe decir qué datos va a utilizar (pero no cómo adquirirlo). Ejemplo: SQL usado en las bases de datos relacionales

### 3. Seguridad de los datos

Mediante un DCL (Lenguaje de Control de Datos) se permite o se impide el acceso información a diferentes usuarios de la BDD

### 4. Gestión de transacciones

Mediante un TCL (Lenguaje de Control de Transacciones) se permite **confirmar** una secuencia de operaciones relacionadas sobre la base de datos, o bien anularlas en caso de que haya habido algún error en alguna.

### 5. Integridad de los datos

No se permiten operaciones que dejen datos incorrectos en la base de datos (por ejemplo se podría no permitir eliminar el registro de un cliente sin eliminar los registros de las compras realizadas por ese cliente)

## 6. Proporcionan un diccionario de metadatos

El diccionario de datos contiene el esquema de la base de datos: las tablas, campos de las tablas, interrelaciones entre tablas, usuarios, permisos, ...

## 7. Monitorizar el funcionamiento y el rendimiento de la base de datos

El SGBD garantiza herramientas estadísticas: registra operaciones efectuadas, consultas realizadas, operaciones fallidas y cualquier otra incidencia.

## 8. Permitir concurrencia

Varios usuarios pueden trabajar simultáneamente sobre un mismo conjunto de datos. El SGBD debe proporcionar mecanismos para arbitrar operaciones conflictivas de acceso o modificación del mismo dato por distintos usuarios en el mismo instante.

## 9. Independencia de los datos de las aplicaciones

Diferentes aplicaciones pueden acceder y modificar independientemente la información de la BDD.

## 10. Ofrecer conectividad con el exterior

Los SGBD proporcionan estándares de conectividad a las aplicaciones. Ejemplo: el protocolo ODBC.

## 11. Herramientas de salvaguarda y recuperación

Los SGBD proporcionan utilidades para hacer **copias de seguridad** de los datos y para **restaurar** una BDD a partir de una copia de seguridad.

## Componentes de un SGBD

Los componentes con los que debe contar un SGBD son:

- **Gestor de almacenamiento:** es el módulo que proporciona la interfaz entre los datos almacenados y los programas de aplicación y consultas solicitadas al sistema. Consta de:
  - **Gestor de transacciones:** asegura que la base de datos sea consistente ante fallos en el sistema.
  - **Gestor de archivos:** gestiona la reserva de espacio de disco y las estructuras de datos para representar los datos almacenados en el disco.
  - **Gestor de memoria intermedia:** se encarga de traer los datos desde el disco a la memoria principal y decidir qué datos se llevan a la memoria caché.
- **Gestor de consultas:** Está formado por:
  - **Intérprete del DDL:** interpreta el DDL y graba las definiciones en el Diccionario de Datos.
  - **Compilador del DML:** traduce las consultas a instrucciones de bajo nivel que puedan ser ejecutadas por el motor de consultas. También optimiza las consultas.
  - **Motor de evaluación de consultas:** ejecuta las instrucciones dadas por el Compilador DML.

## Tipos de SGBD

Por capacidad y potencia los SGBD se pueden clasificar en:

- **Gestores de bases de datos ofimáticas:** permiten gestionar bases de datos domésticas o de pequeños negocios. Ejemplo: Microsoft **Access** (forma parte del paquete ofimático Microsoft Office).
- **Gestores de bases de datos corporativas:** permiten gestionar bases de datos enormes (con muchos datos y transacciones) de grandes o medianas empresas. El SGBD se suele instalar en un servidor muy potente. Ejemplos: **Oracle** o **DB2**
- **Gestores de bases de datos medianas:** sirven para gestionar bases de datos intermedias entre las ofimáticas y las corporativas. El SGBD de este tipo más usado en la actualidad es **MySQL** que tiene licencia libre. Se usa mucho como parte de las soluciones XAMPP

## Bases de datos centralizadas y bases de datos distribuidas

### Bases de datos centralizadas

Una base de datos que reside en un solo nodo de una red informática es una **BDD centralizada**. El resto de nodos de la red pueden acceder a la BDD.

## Bases de datos distribuidas

Una BDD distribuida está formada por un conjunto de bases de datos lógicamente relacionadas e interconectadas entre ellas por una red informática.

Un usuario puede acceder a los datos de cualquier nodo de la red de manera transparente para él (sin saber a qué nodo está accediendo para obtener un dato concreto)