



# Tema 1

Sistemas de almacenamiento  
de la información

Pelayo Vega Valdés

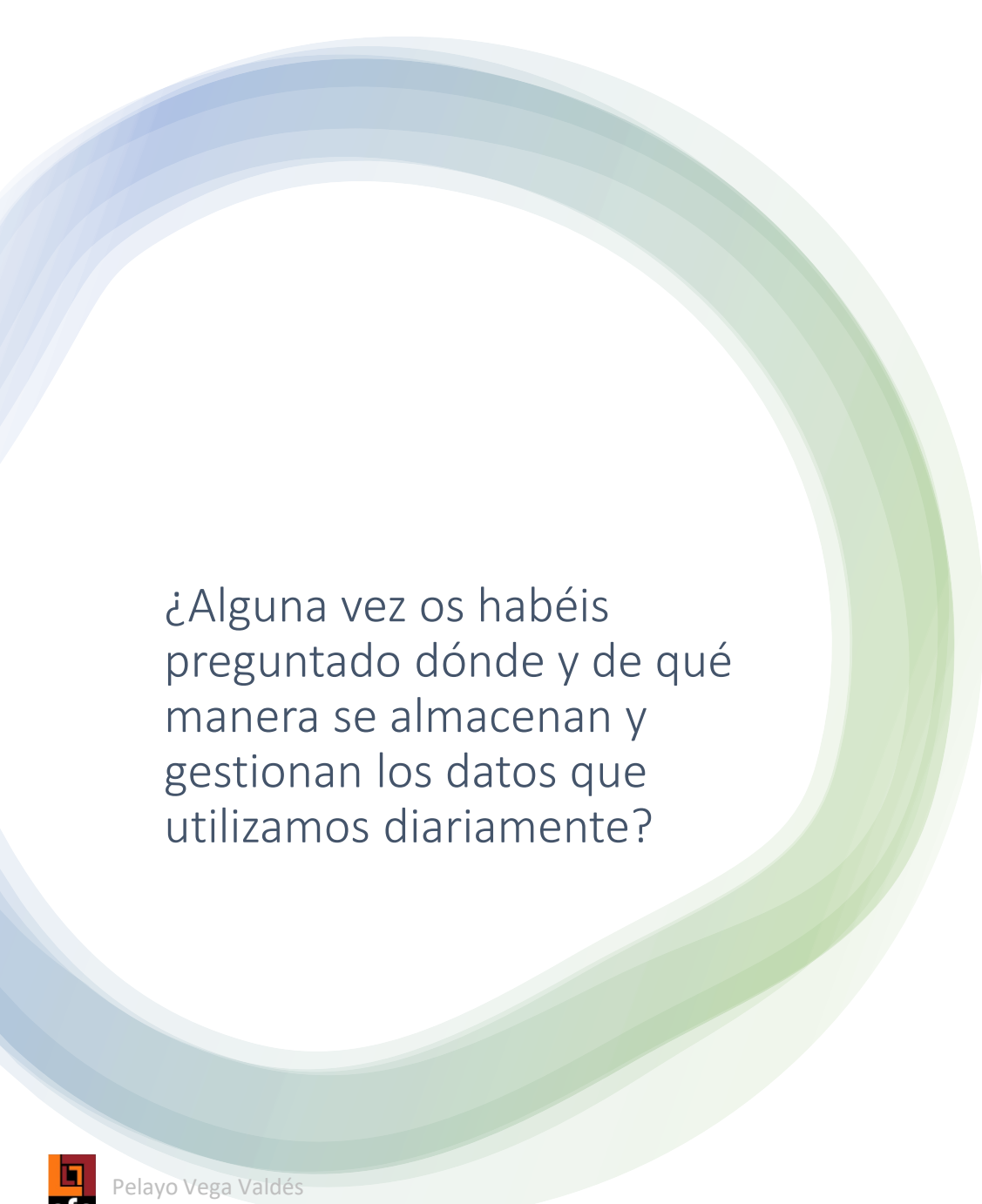
# Objetivos

- Comprender la problemática derivada del almacenamiento óptimo de la información.
- Tomar cierta perspectiva histórica sobre el desarrollo de los sistemas de información.
- Asimilar las características de los distintos tipos de ficheros.
- Describir los distintos modelos de bases de datos.
- Familiarizarse con los componentes y funciones de los sistemas gestores de bases de datos.

# Índice

1. Introducción.
2. Ficheros.
  1. Tipos de fichero según su estructura de almacenamiento.
  2. Tipos de soporte de almacenamiento.
3. Bases de datos.
  1. Definición.
  2. Modelos de bases de datos
  3. Tipos de bases de datos.
4. Sistema gestores de bases de datos.
  1. Componentes del SGBD.
  2. Funciones del SGBD





¿Alguna vez os habéis preguntado dónde y de qué manera se almacenan y gestionan los datos que utilizamos diariamente?





Si nos paramos a pensar en cualquier acción de nuestro día a día, nos encontramos que la utilización de las bases de datos está ampliamente extendida.

# Actividad

Pensar 5 acciones cada uno de vosotros donde utilizéis una base de datos “inconscientemente”



# 1.1. Introducción

Según la RAE, informática es el “Conjunto de conocimientos científicos y técnicos que hacen posible el tratamiento de la información por medio de ordenadores”.

Definimos sistema de información, como el conjunto de procedimientos y funciones dirigidos a la recogida, elaboración, evaluación, almacenamiento, recuperación, condensación y distribución de informaciones dentro de una organización.

Antes de que surgieran las bases de datos el procesamiento automatizado de información se hacía mediante **ficheros**.

Los ficheros se diseñaban a medida para cada sistema de información, sin que existiera un formato común.

El problema de esta organización, es que no contemplaba la gestión de la información a medio o largo plazo y una organización disponía de varias aplicaciones que, en algunos casos, trataban la misma información.

El software utilizado por el departamento de recursos humanos debía gestionar un fichero de datos con empleados, mientras que la aplicación de contabilidad de la misma empresa mantenía otro fichero con los mismos datos pero de otra forma.

Esto generaba los siguientes **problemas**:

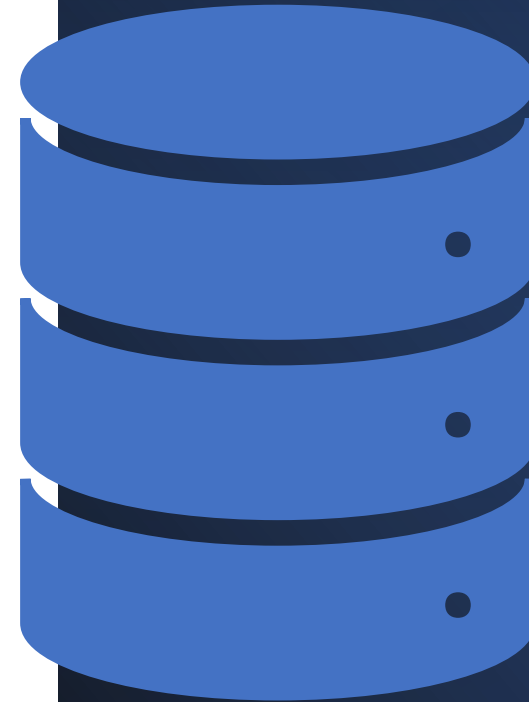
- Redundancia de los datos (duplicidad innecesaria de información).
- Mal aprovechamiento del espacio de almacenamiento.
- Aumento en el tiempo de proceso.
- Inconsistencia de información debida a la redundancia (si un dato cambia en el fichero de una aplicación, no cambiaba en los demás).
- Aislamiento de la información (imposibilidad de transferencia a otros programas a no ser que se desarrollara un software de migración específico).

Había, en definitiva, una gran falta de flexibilidad originada en la dependencia total de la estructura física de los datos.



# Índice

1. Introducción.
2. [Ficheros.](#)
  1. Tipos de fichero según su estructura de almacenamiento.
  2. Tipos de soporte de almacenamiento.
3. Bases de datos.
  1. Definición.
  2. Modelos de bases de datos
  3. Tipos de bases de datos.
4. Sistema gestores de bases de datos.
  1. Componentes del SGBD.
  2. Funciones del SGBD.



## 2. Ficheros

Las aplicaciones gestoras de bases de datos se encargan de configurar una estructura óptima de almacenamiento de información con mínima intervención por parte del usuario.

No obstante, es interesante contemplar la perspectiva histórica con una breve descripción teórica sobre la organización y tipos de ficheros.



## 2.2.1. Tipos de ficheros según su estructura de almacenamiento



### **Texto Plano.**

Secuenciales

De acceso directo o aleatorio

Indexados



### **Binarios.**

Almacenan secuencias de caracteres correspondientes a una codificación determinada (ASCII, Unicode, etc.). Son estándares de codificación de caracteres.

Son legibles mediante un software de edición de textos como: bloc de notas de Windows o el Vi de Linux.

Ejemplos:

- Ficheros de texto con extensión txt.
- Los CSV (comma-separated values) de valores separados por comas.
- Archivos html de las páginas web.
- Lenguajes de marcas xml.
- ...

# Texto plano

La información se escribe en posiciones físicamente contiguas. Para acceder a un dato hay que recorrer todos los anteriores.

```
00789521T#Paula#Sanz#González#619554687$50687452Y#José Luis#García#Viñals#  
667859621$38546998X#Javier#Peinado#Martín#666932541$09653801B#Ruth#Lázaro#  
Cardenal#689330247%
```

Para cada contacto se ha decidido estructurar la información en 5 datos independientes: NIF, nombre, primer apellidos, segundo apellido y número de teléfono.

Observa que, el programador ha decidido utilizar la # para separar los datos, el \$ como separador de contactos y el % como marca de fin de fichero.

## Texto plano: Ficheros secuenciales

# Actividad

Crear un fichero secuencial, con distintos caracteres del ejemplo anterior (# \$ %)

Dale el ejemplo a tu compañero de mesa

¿Qué ocurre? ¿lo entiende?



Cada línea de contenido se organiza con unos tamaños fijos de datos. Se puede acceder directamente al principio de cada línea.

00789521TPaula	Sanz	González	619554687
50687452YJosé Luis	García	Viñals	667859621
38546998XJavier	Peinado	Martín	666932541
09653801BRuth	Lázaro	Cardenal	689330247

En esta ocasión cada contacto ocupa una línea del fichero (al final de cada línea el Sistema mete uno o dos caracteres invisibles para el usuario) y cada dato ocupa una longitud fija, utilice o no la totalidad de los caracteres.

Como todos los datos ocupan la misma longitud podemos acceder a cualquier posición ¿Cómo?

## Texto plano: Ficheros de acceso directo o aleatorio

00789521TPaula	Sanz	González	619554687
50687452YJosé Luis	García	Viñals	667859621
38546998XJavier	Peinado	Martín	666932541
09653801BRuth	Lázaro	Cardenal	689330247

Sabemos que la longitud de la línea es 65 (63 de la longitud de los datos y vamos a poner 2 que mete al final el Sistema operativo invisibles).

Si por ejemplo queremos acceder a la tercera posición, tendremos que restar uno a su posición (siempre empezamos en la posición 0), por tanto 2 (3-1=2) y multiplicar por la longitud de la línea.

$$2 \times 65 = 130$$

Por tanto la posición del tercer cliente estará en la posición 131.

La contrapartida a esta facilidad de acceso es que el fichero crece considerablemente en cuanto al tipo secuencial

## Texto plano: Ficheros de acceso directo o aleatorio

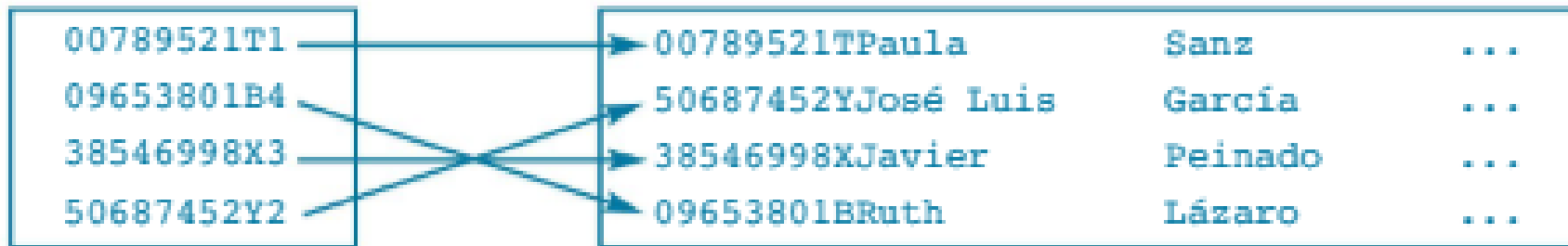


Generalmente en un ficheros de acceso aleatorio la información se guarda en el orden que se da de alta, a los clientes en este caso.

Incluso aunque se siga algún criterio para insertarlos de manera ordenada, en ocasiones es necesario ordenar los datos con algún otro criterio.

Por ejemplo, suponer que en el caso anterior se quiere crear un listado de los clientes ordenados por NIF.

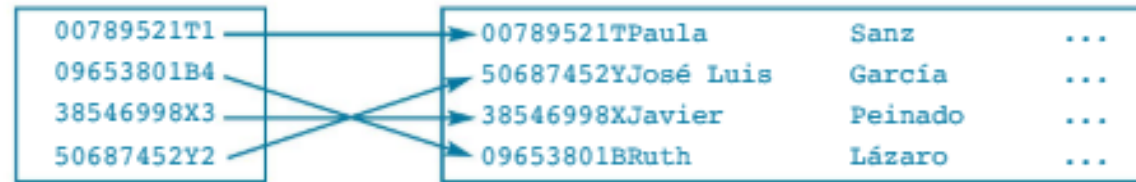
Para dar solución a este problema se crea la organización indexada, que consiste en la existencia de uno o varios ficheros adjuntos que ordenan el dato (llamado clave) por el que se desea ordenar el fichero y lo relacionan con la localización de la línea correspondiente.



## Texto plano: Ficheros indexados

Tras los NIF se ha añadido un número que indica el número de línea del fichero principal donde se encuentra la información asociada.

El sistema operativo si quisiera encontrar un cliente, recorrería el fichero de índices y al final de cada línea leería el valor para encontrar el cliente.



Otro ejemplo, ordenado por apellido



Aunque estemos utilizando un número para acceder a la posición, en un sistema real no lo marca un número, sino un puntero a la posición de memoria del ordenador.

## Ficheros indexados

# Actividad

Con los datos del ejemplo anterior, crear un fichero indexado en clave por nombre



Contienen información codificada en binario para su procesamiento por parte de las aplicaciones. Su contenido resulta ilegible para un editor de textos.

Ejemplo:

- Archivo ejecutable: exe.
- Documentos de aplicaciones: pdf, docx, xlsx, pptx, etc.
- Ficheros de imágenes, audios, vídeos: jpg, gif, mp3, avi, etc.
- Archivos de sistema: dll.

# Binarios

# Actividad

Buscar en el equipo de trabajo, archivos de tipo secuencial y archivos de tipo binario.  
Ejemplo: fichero.doc y fichero.html

Abre los archivos con el bloc de notas y observa lo que ves



## 2.2.2. Tipos de soporte de almacenamiento

De acuerdo a la organización física de los datos, diferenciamos entre dos tipos de soportes:

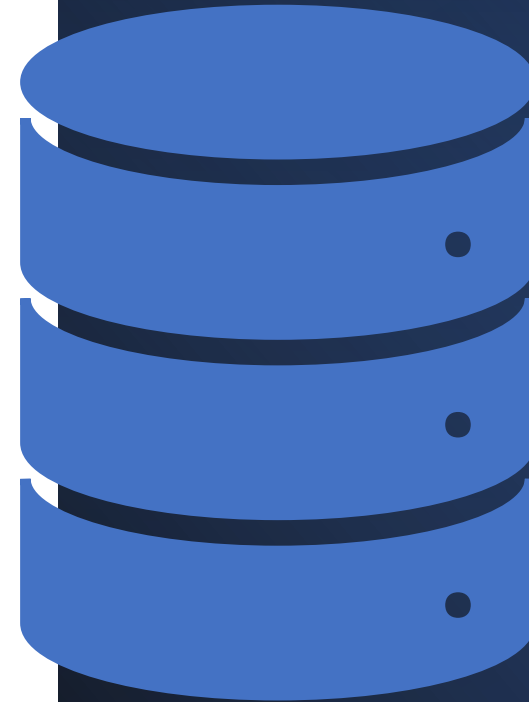
- Secuencial: para acceder a un dato hay que recorrer todo el contenido del soporte previo a dicho dato. Ejemplo: las cintas magnéticas de IBM.
- Direccional: se puede acceder directamente a un dato sin tener que recorrer todos los elementos anteriores. Ejemplo: disco duro.



En un soporte direccional se puede implementar un acceso secuencial, directo o indexado, mientras que en un soporte secuencial solo se podrá implementar un acceso secuencial.

# Índice

1. Introducción.
2. Ficheros.
  1. Tipos de fichero según su estructura de almacenamiento.
  2. Tipos de soporte de almacenamiento.
3. Bases de datos.
  1. Definición.
  2. Modelos de bases de datos
  3. Tipos de bases de datos.
4. Sistema gestores de bases de datos.
  1. Componentes del SGBD.
  2. Funciones del SGBD.



### 3. Bases de Datos

Como se vio en el capítulo anterior, los **ficheros** permiten organizar y memorizar conjuntos de datos del mismo tipo o naturaleza con una determinada estructura, siendo un medio para el almacenamiento de los datos o resultados de una aplicación específica. Pero si las aplicaciones, al ser diseñadas, deben depender directamente de sus ficheros o archivos, se pierde independencia y surgen serios inconvenientes: como **información duplicada, incoherencia de datos, fallos de seguridad**, etc.

Estos problemas debían ser solucionados, es cuando aparece el concepto de **base de datos**. Una base de datos permitirá reunir toda la información relacionada en un único sistema de almacenamiento, pudiendo cualquier aplicación utilizarla de manera independiente y ofreciendo una mejora en el tratamiento de la información, así como una evolución para el desarrollo de aplicaciones.

A finales de los setenta, la aparición de nuevas tecnologías de manejo de datos a través de los sistemas de bases de datos supuso un considerable cambio. Los sistemas basados en ficheros separados dieron paso a la utilización de sistemas gestores de bases de datos, que son sistemas software centralizados o distribuidos que ofrecen facilidades para la definición de bases de datos, selección de estructuras de datos y búsqueda de forma interactiva o mediante lenguajes de programación.

Llegados a este punto, te preguntarás... ¿Qué es una base de datos?



## 3.1. ¿Qué es una base de datos?

**Base de datos:** Es una colección de datos relacionados lógicamente entre sí, con una definición y descripción comunes y que están estructurados de una determinada manera. Es un conjunto estructurado de datos que representa entidades y sus interrelaciones, almacenados con la mínima redundancia y posibilitando el acceso a ellos eficientemente por parte de varias aplicaciones y usuarios.

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.

# Elementos de una base de datos

Una base de datos constará de los siguientes elementos:

- **Entidades:** objeto real o abstracto con características diferenciadoras de otros, del que se almacena información en la base de datos. En una base de datos de un taller mecánico, posibles entidades podrían ser: mecánico, persona, coche, local, etc.
- **Atributos:** son los datos que se almacenan de la entidad. Cualquier propiedad o característica de una entidad puede ser atributo. Continuando con nuestro ejemplo, podrían ser atributos de la entidad coche por ejemplo: marca, color, modelo, matrícula, etc.
- **Registros:** donde se almacena la información de cada entidad. Es un conjunto de atributos que contienen los datos que pertenecen a una misma repetición de entidad. En nuestro ejemplo, un registro podría ser: 9873DHB, Renault, Megane, Gris.
- **Campos:** donde se almacenan los atributos de cada registro. Teniendo en cuenta el ejemplo anterior, un campo podría ser el valor Renault.

# Actividad

Enumerar las entidades de un supermercado, los atributos de uno de ellos y escribir un registro de la entidad elegida



# Ventajas que ofrece el uso de una base de datos

- **Acceso múltiple:** Diversos usuarios o aplicaciones podrán acceder a la base de datos, sin que existan problemas en el acceso a datos.
- **Utilización múltiple:** Cada uno de los usuarios o aplicaciones podrán disponer de una visión particular de la estructura de la base de datos, de tal manera que cada uno de ellos accederá sólo a la parte que realmente le corresponde.
- **Flexibilidad:** La forma de acceder a la información puede ser establecida de diferentes maneras, ofreciendo tiempos de respuesta muy reducidos.
- **Confidencialidad y seguridad:** El control del acceso a los datos podrá ser establecido para que unos usuarios o aplicaciones puedan acceder a unos datos y a otros no, impidiendo a los usuarios no autorizados la utilización de la base de datos.
- **Protección contra fallos:** En caso de errores en la información, existen mecanismos bien definidos que permiten recuperar los datos de forma fiable.
- **Independencia física:** Un cambio de soporte físico de los datos (por ejemplo, el tipo de disco), no afectaría a la base de datos o a las aplicaciones que acceden a ellos.
- **Independencia lógica:** Los cambios realizados en la base de datos no afectan a las aplicaciones que la usan.
- **Redundancia:** Los datos se almacenan, por lo general, una única vez. Aunque si es necesario podríamos repetir información de manera controlada.
- **Interfaz de alto nivel:** Mediante la utilización de lenguajes de alto nivel puede utilizarse la base de datos de manera sencilla y cómoda.
- **Consulta directa:** Existen herramientas para poder acceder a los datos interactivamente.

## 3.2. Modelos o tipos de bases de datos

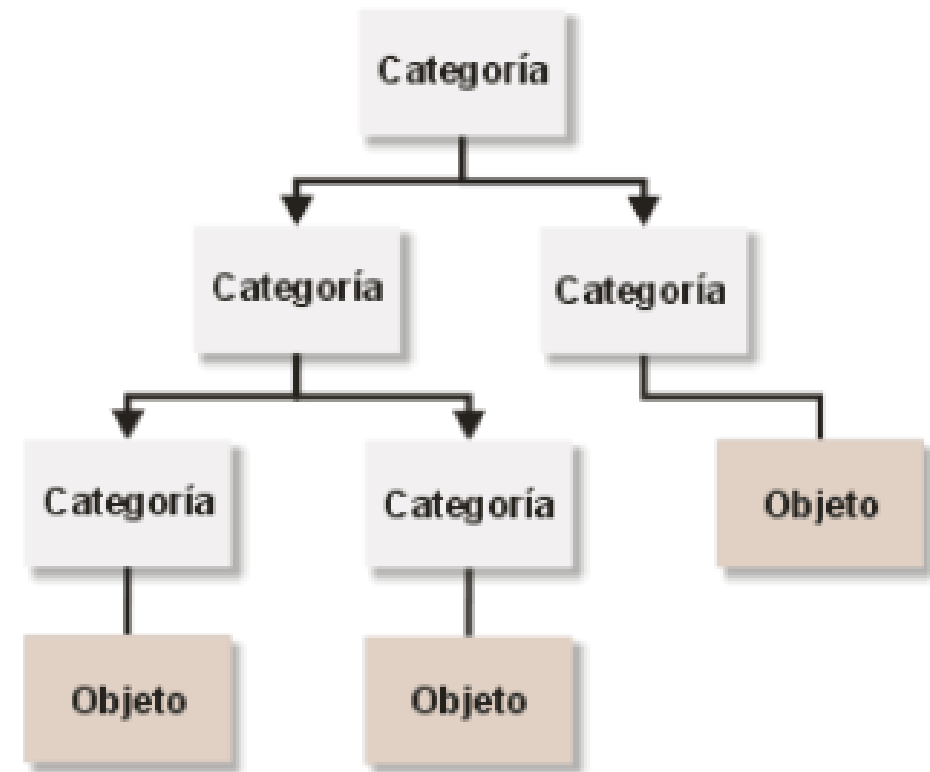
Un modelo de bases de datos es la arquitectura mediante la que se almacena e interrelaciona la información que se va a gestionar.

La clasificación tradicional de las bases de datos establece tres modelos de bases de datos: jerárquico, en red y relacional aunque en la actualidad el modelo de bases de datos más extendido es el **relacional**.

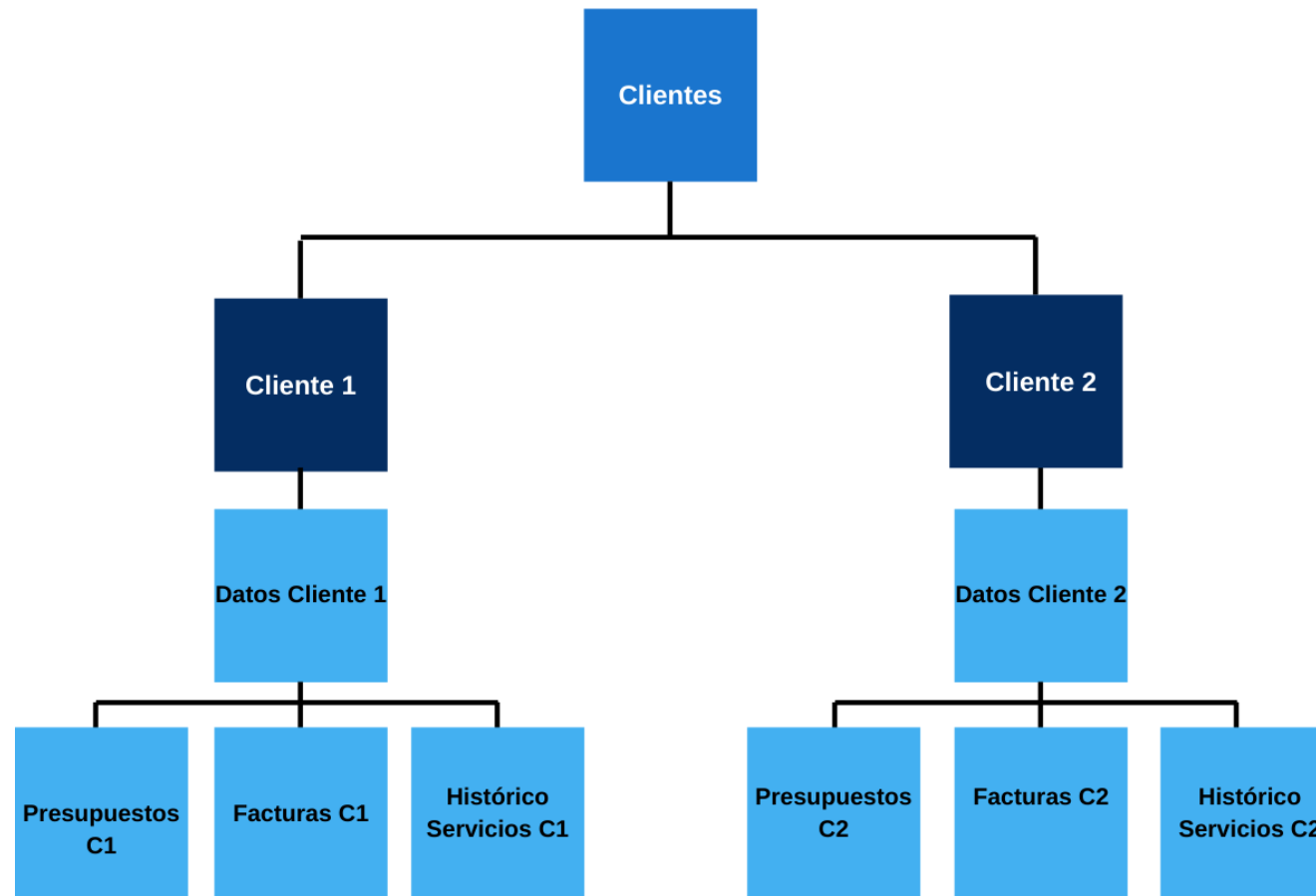
# Modelo Jerárquico

El modelo jerárquico organiza los datos en una estructura de árbol, en la que cada registro tiene un único elemento o raíz. Los registros del mismo nivel se clasifican en un orden específico. Ese orden se usa a manera de orden físico para almacenar la base de datos. El modelo es bueno para describir muchas relaciones del mundo real.

Este modelo fue usado principalmente por los Sistemas de Gestión de Información de IBM en las décadas de 1960 y 1970, pero raramente se los ve en la actualidad debido a ciertas ineficiencias operativas.



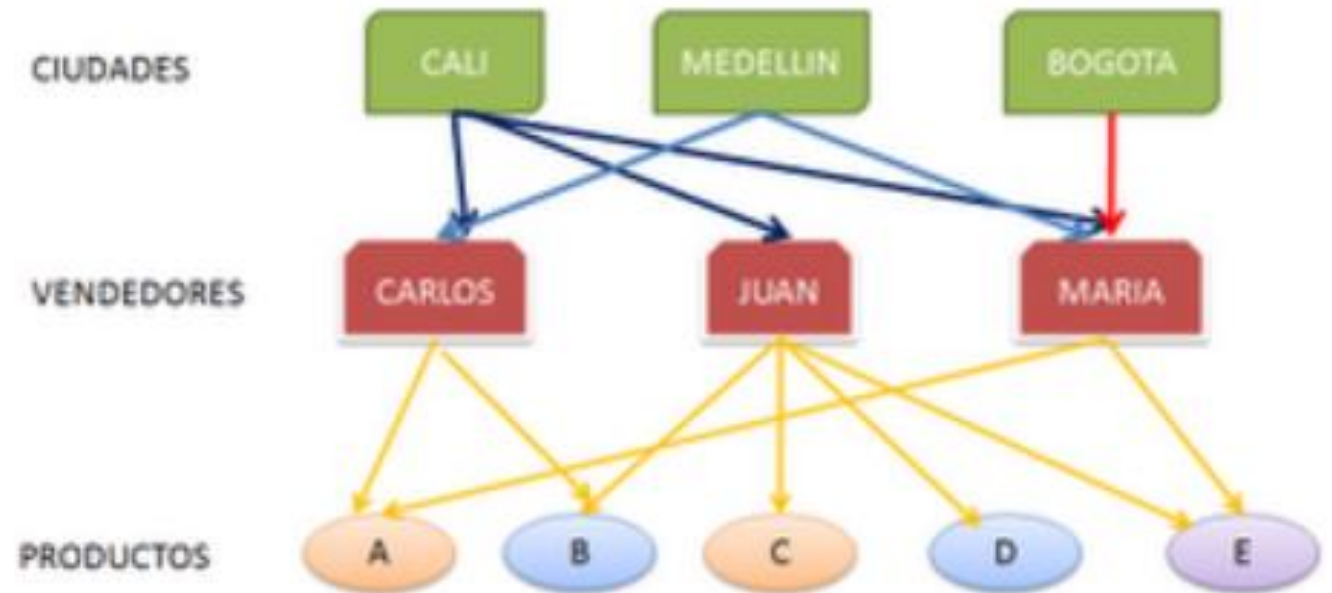
# Modelo Jerárquico



# Modelo en Red

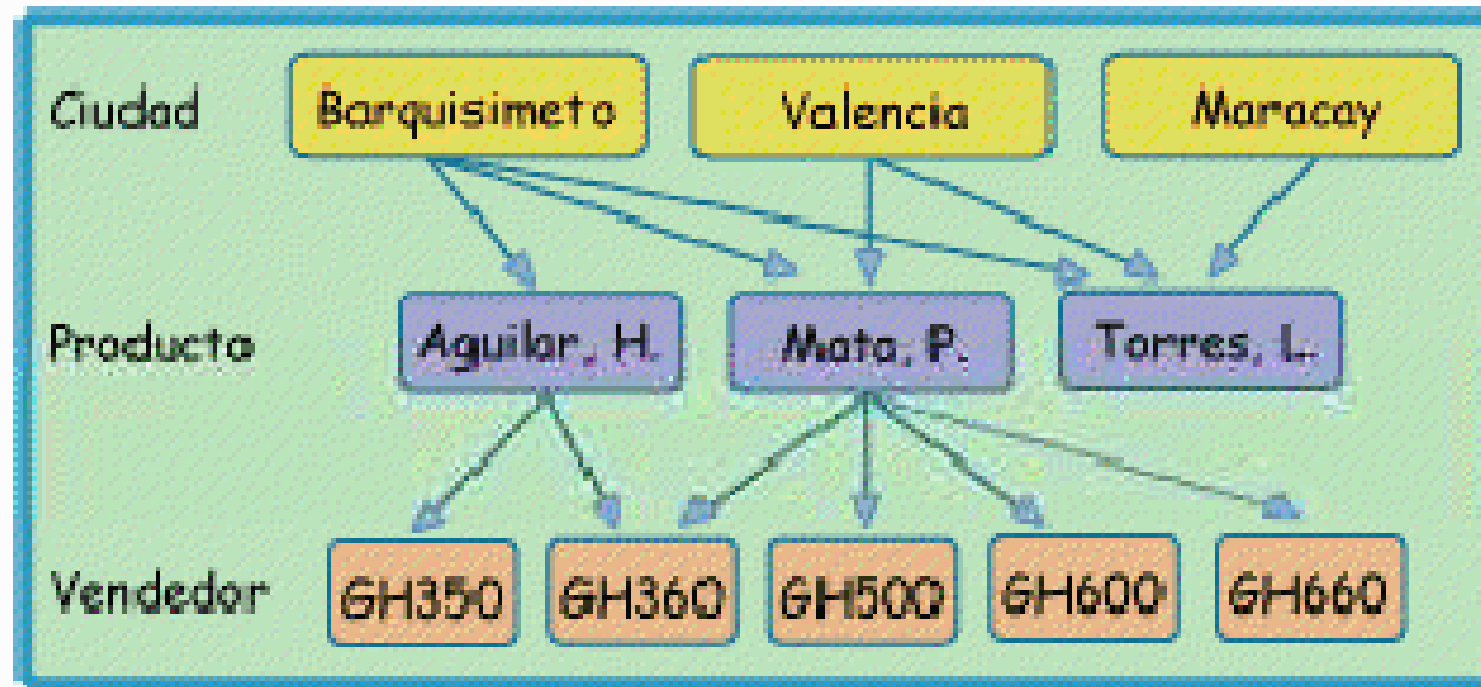
El modelo de red se basa en el modelo jerárquico, permitiendo relaciones de muchos a muchos entre registros vinculados, lo que implica registros principales múltiples. Basado en la teoría matemática de conjuntos, el modelo se construye con conjuntos de registros relacionados. Cada conjunto consiste de un registro propietario o principal y uno o más registros miembros o secundarios. Un registro puede ser miembro o secundario en múltiples conjuntos, permitiendo que este modelo represente relaciones complejas.

Fue muy popular en la década de 1970 después de que fue definido formalmente por la Conference on Data Systems Languages (CODASYL).





# Modelo en Red



# Modelo Relacional

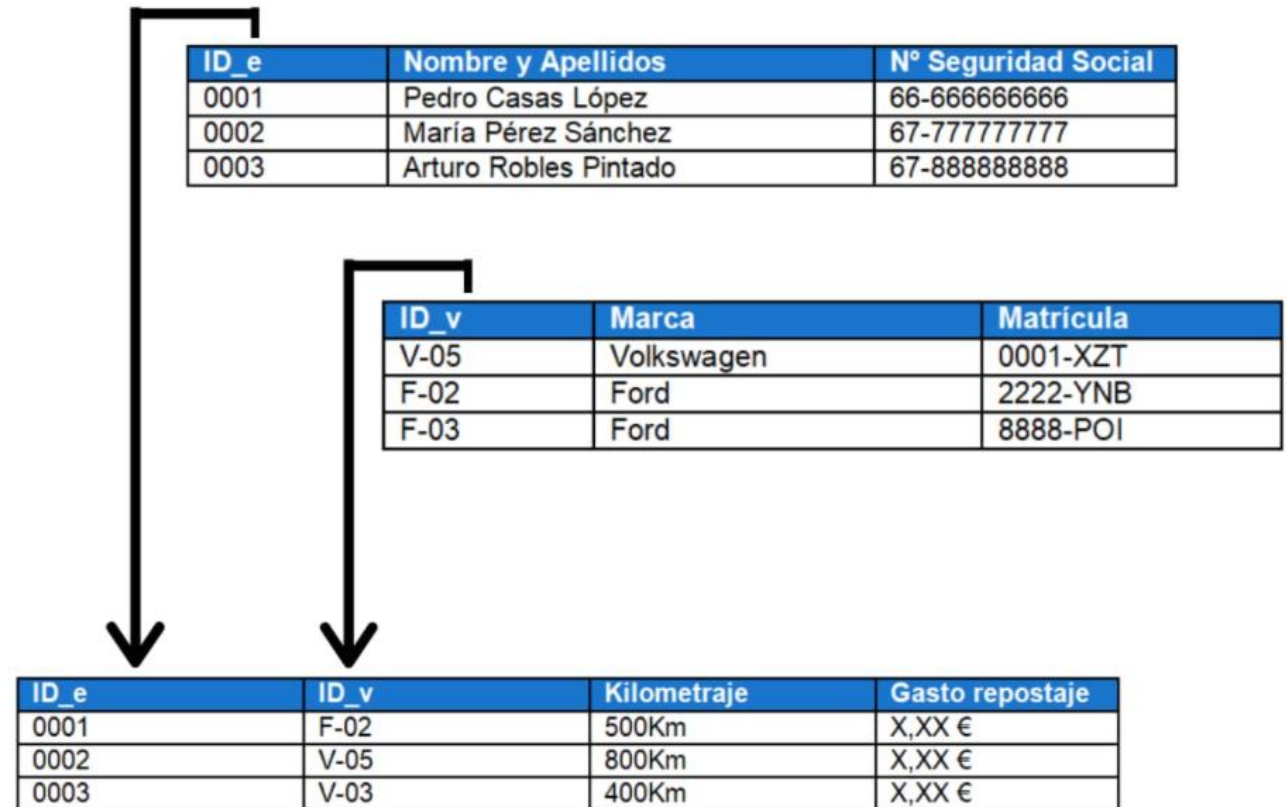
Siendo el modelo más común, el modelo relacional ordena los datos en tablas, también conocidas como relaciones, cada una de las cuales se compone de columnas y filas. Cada columna enumera un atributo de la entidad en cuestión, por ejemplo, precio, código postal o fecha de nacimiento. En conjunto, a los atributos en una relación se los llama dominio. Se elige un atributo particular o combinación de atributos como clave primaria, a la cual se puede hacer referencia en otras tablas, en donde se la denomina clave externa.

Cada fila, también denominada tupla, incluye datos sobre una instancia específica de la entidad en cuestión, por ejemplo, un empleado específico.

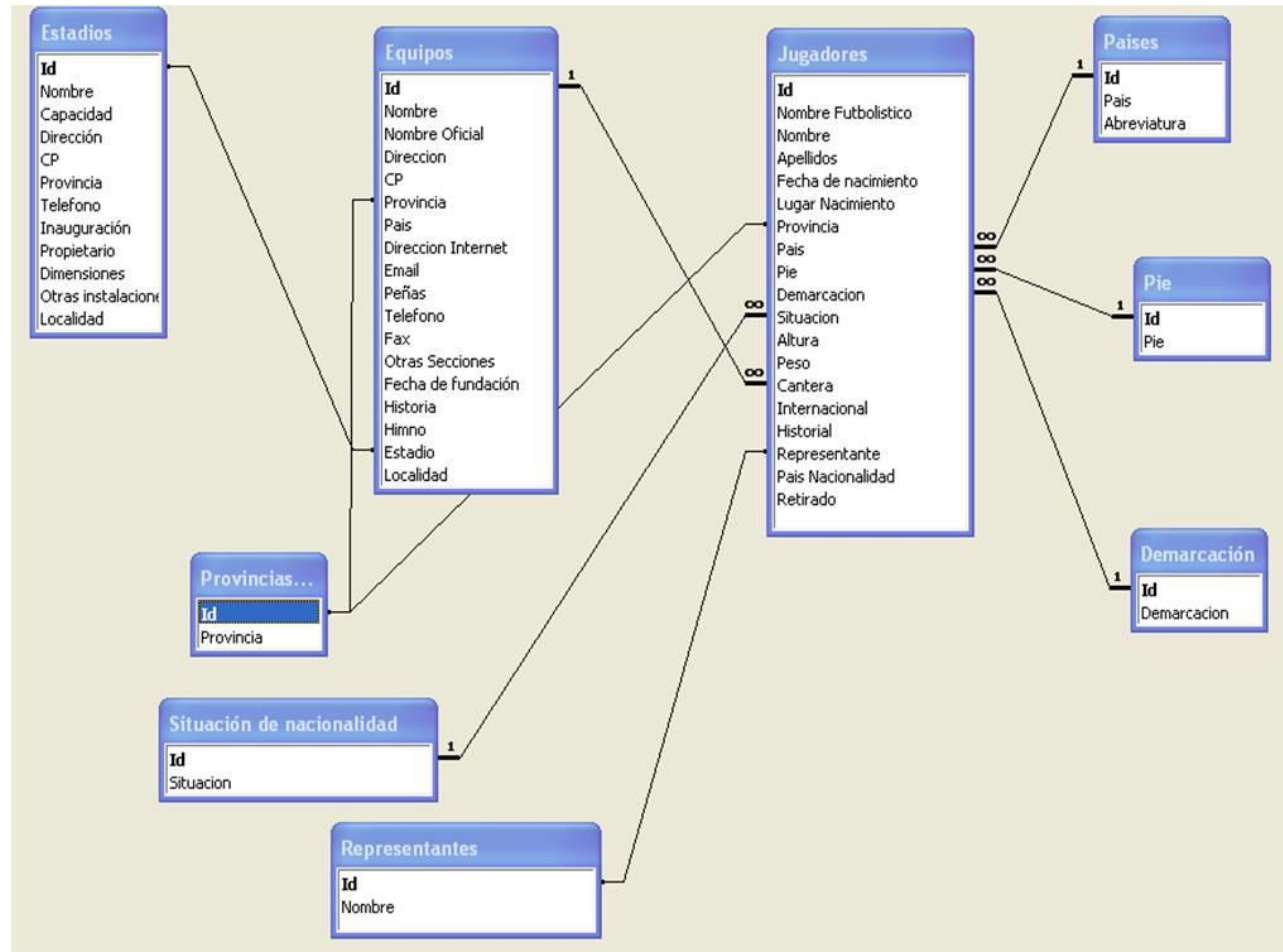
El modelo también representa los tipos de relaciones entre esas tablas, incluidas las relaciones uno a uno, uno a muchos y muchos a muchos.

Dentro de la base de datos, las tablas se pueden normalizar, es decir, hacer que cumplan las reglas de normalización que hacen a la base de datos flexible, adaptable y escalable. Al estar normalizada, cada porción de los datos es atómica, es decir, está dividida en partes útiles lo más pequeñas posibles.

Las bases relacionales generalmente se escriben en Lenguaje de Consulta Estructurado (SQL, por sus siglas en inglés). El modelo fue introducido por E. F. Codd en 1970.



# Modelo Relacional



# Modelo Orientado a Objetos

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. Las operaciones de cada clase se especifican en términos de procedimientos predefinidos denominados **métodos**.

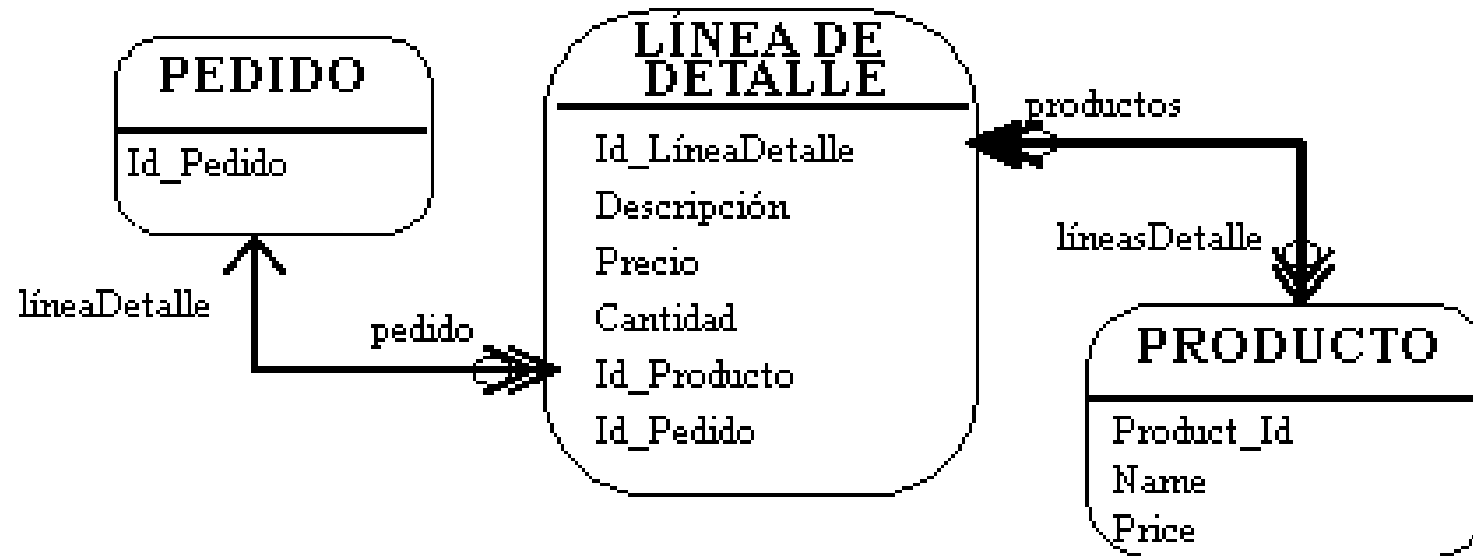
Algunos sistemas existentes en el mercado, basados en el modelo relacional, han sufrido evoluciones incorporando conceptos orientados a objetos. A estos modelos se les conoce como sistemas **objeto-relacionales**.

El objetivo del modelo orientado a objetos es cubrir las limitaciones del modelo relacional.

Gracias a este modelo se incorporan mejoras como la herencia entre tablas, los tipos definidos por el usuario, disparadores almacenables en la base de datos (triggers), soporte multimedia, etc.

Desde la aparición de la programación orientada a objetos (POO u OOP) se empezó a pensar en bases de datos adaptadas a estos lenguajes. Este modelo es considerado como el fundamento de las bases de datos de tercera generación, siendo consideradas las bases de datos en red como la primera y las bases de datos relacionales como la segunda generación. Aunque no han reemplazado a las bases de datos relacionales, si son el tipo de base de datos que más está creciendo en los últimos años.

# Modelo Orientado a Objetos



# Actividad

Buscar una representación gráfica de cada uno de los modelos que no estén en los apuntes



## 3.3. Tipos de bases de datos

Ya hemos visto una forma de clasificar las bases de datos, por modelos. Ahora vamos a ver otras clasificaciones en función de otros criterios.

Según su contenido:

- Bases de datos con contenido actual.
- Directorios.
- Bases de datos documentales.

Según su uso:

- Bases de datos individual.
- Base de datos compartida.
- Base de datos de acceso público.
- Base de datos propietarias o bancos de datos.

Según su variabilidad de la información:

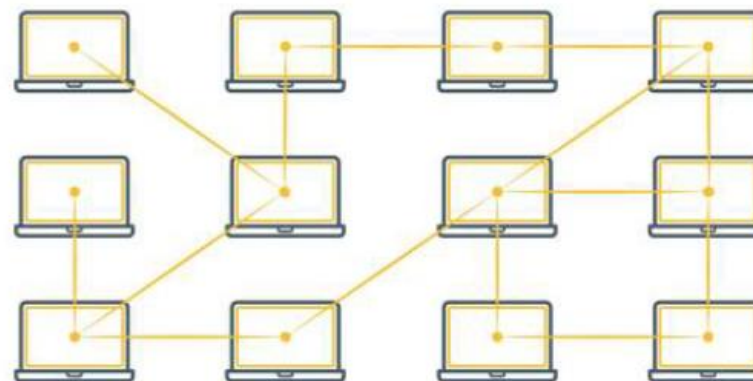
- Bases de datos estáticas.
- Bases de datos dinámicas.

## Según el organismo productor:

- Bases de datos de organismos públicos y de la administración.
- Bases de datos de instituciones sin ánimo de lucro.
- Bases de datos de entidades comerciales o privadas.
- Bases de datos realizadas por cooperación en red.

## Según la localización de la información:

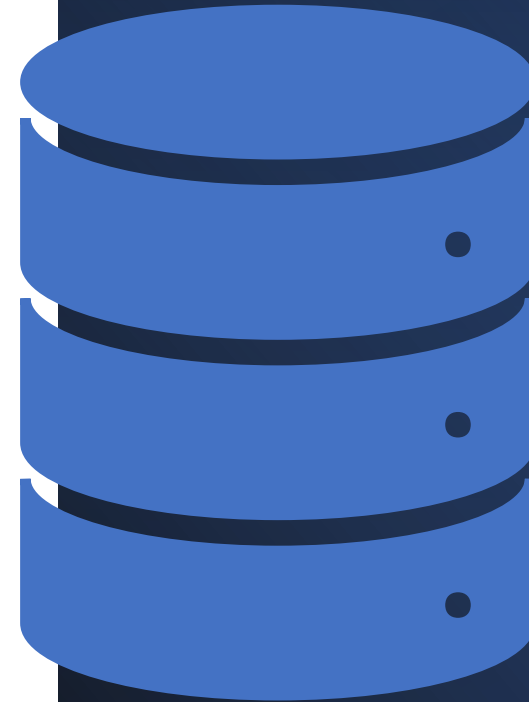
- Bases de datos centralizada.
- Bases de datos distribuidas.





# Índice

1. Introducción.
2. Ficheros.
  1. Tipos de fichero según su estructura de almacenamiento.
  2. Tipos de soporte de almacenamiento.
3. Bases de datos.
  1. Definición.
  2. Modelos de bases de datos
  3. Tipos de bases de datos.
4. [Sistema gestores de bases de datos.](#)
  1. Componentes del SGBD.
  2. Funciones del SGBD.



## 4. Sistemas de gestión de bases de datos

Para poder tratar la información contenida en las bases de datos se utilizan los sistemas gestores de bases de datos o SGBD, también llamados DBMD (DataBase Management System), que ofrecen un conjunto de programas que permiten acceder y gestionar dichos datos por parte de los usuarios.

**Sistema Gestor de Base de Datos:** Conjunto coordinado de programas, procedimientos, lenguajes, etc., que suministra, tanto a los usuarios no informáticos, como a los analistas programadores, o al administrador, los medios necesarios para describir y manipular los datos contenidos en la base de datos, manteniendo su integridad, confidencialidad y seguridad.

El SGBD permite a los usuarios la creación y el mantenimiento de una base de datos, facilitando la definición, construcción y manipulación de la información contenida en éstas.

Definir una base de datos consistirá en especificar los tipos de datos, las estructuras y las restricciones que los datos han de cumplir a la hora de almacenarse en dicha base.

Por otro lado, la construcción de la base será el proceso de almacenamiento de datos concretos en algún medio o soporte de almacenamiento que esté supervisado por el SGBD.

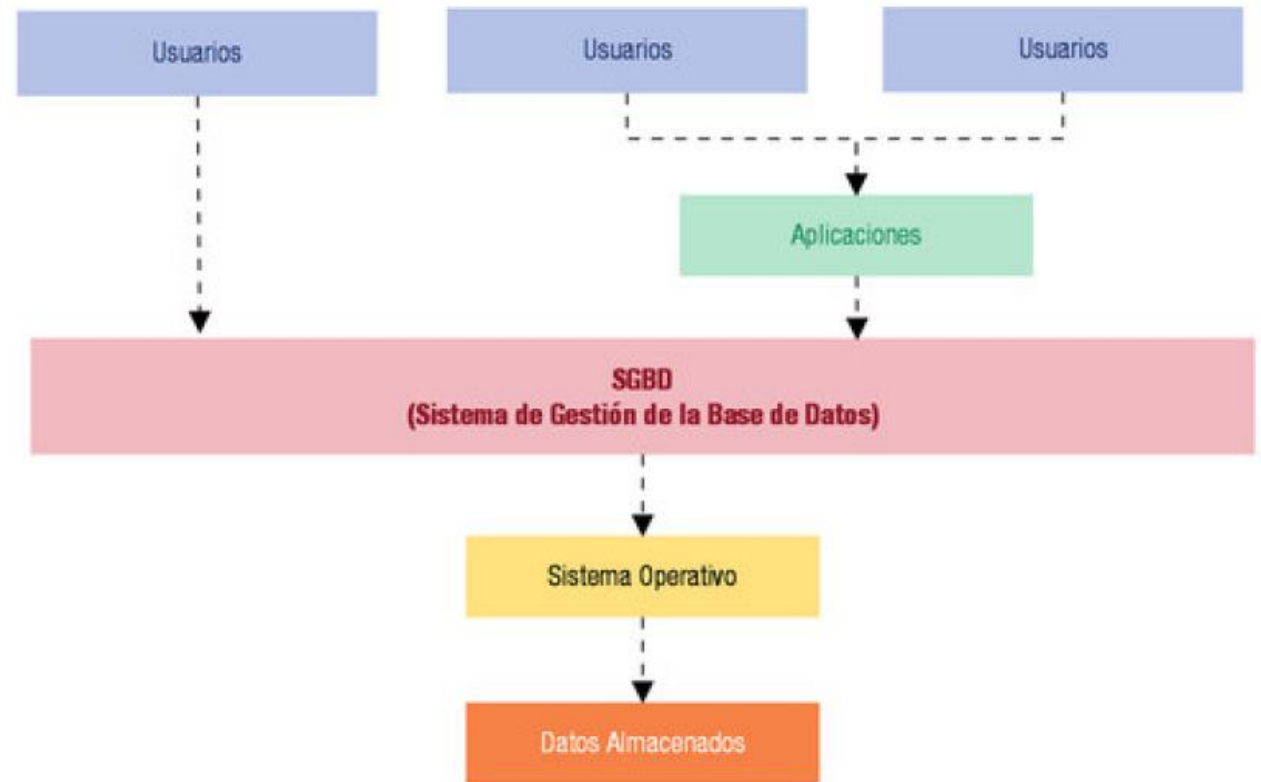
Finalmente, la manipulación de la base de datos incluirá la posibilidad de realización de consultas para recuperar información específica, la actualización de los datos y la generación de informes a partir de su contenido.

# Ventajas del uso de las bases de datos

- Proporcionan al usuario una visión abstracta de los datos, ocultando parte de la complejidad relacionada con cómo se almacenan y mantienen los datos.
- Ofrecen Independencia física, es decir, la visión que tiene de la información el usuario, y la manipulación de los datos almacenados en la Base de Datos, es independiente de cómo estén almacenados físicamente.
- Disminuyen la redundancia y la inconsistencia de datos.
- Aseguran la integridad de los datos.
- Facilitan el acceso a los datos, aportando rapidez y evitando la pérdida de datos.
- Aumentan la seguridad y privacidad de los datos.
- Mejoran la eficiencia.
- Permiten compartir datos y accesos concurrentes.
- Facilitan el intercambio de datos entre distintos sistemas.
- Incorporan mecanismos de copias de seguridad y recuperación para restablecer la información en caso de fallos.

# Interacciones de las bases de datos con otros componentes

El SGBD interacciona con otros elementos software existentes en el sistema, concretamente con el sistema operativo. Los datos almacenados de forma estructurada en la base de datos son utilizados indistintamente por otras aplicaciones, será el SGBD quien ofrecerá una serie de facilidades a éstas para el acceso y manipulación de la información, basándose en las funciones y métodos propios del sistema operativo



# Actividad

Buscar los sistemas de gestión de bases de datos más populares.



# Índice

1. Introducción.
2. Ficheros.
  1. Tipos de fichero según su estructura de almacenamiento.
  2. Tipos de soporte de almacenamiento.
3. Bases de datos.
  1. Definición.
  2. Modelos de bases de datos
  3. Tipos de bases de datos.
4. Sistema gestores de bases de datos.
  1. [Funciones del SGBD.](#)
  2. Componentes del SGBD.



# Funciones de la base de datos

Un SGBD desarrolla tres funciones fundamentales como son las de descripción, manipulación y utilización de los datos



Descripción  
**DDL**

Manipulación  
**DML**

Utilización de  
los datos  
**DCL**



# Descripción

**Función de descripción o definición:** Permite al diseñador de la base de datos crear las estructuras apropiadas para integrar adecuadamente los datos.

Esta función se realiza mediante el **lenguaje de descripción de datos** o DDL (Data Definition Language).

Mediante ese lenguaje: se definen las estructuras de datos, se definen las relaciones entre los datos y se definen las reglas (restricciones) que han de cumplir los datos.



# Descripción

CREATE, se usa para crear una base de datos, tabla, vistas, etc.

ALTER, se utiliza para modificar la estructura, por ejemplo añadir o borrar columnas de una tabla.

DROP, con esta sentencia, podemos eliminar los objetos de la estructura, por ejemplo un índice o una secuencia.

# Manipulación

**Función de manipulación:** permite a los usuarios de la base buscar, añadir, suprimir o modificar los datos de la misma, 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 – Data Manipulation Language) que facilita los instrumentos necesarios para la realización de estas tareas.

También se encarga 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.

Por manipulación de datos entenderemos:

- La recuperación de información almacenada en la base de datos, lo que se conoce como **consultas**.
- La inserción de información nueva en la base de datos.
- El borrado de información de la base de datos.
- La modificación de información almacenada en la base de datos.



# Manipulación

SELECT, esta sentencia se utiliza para realizar consultas sobre los datos.

INSERT, con esta instrucción podemos insertar los valores en una base de datos.

UPDATE, sirve para modificar los valores de uno o varios registros.

DELETE, se utiliza para eliminar las filas de una tabla.

# Control y utilización de los datos

**Función de control:** permite al administrador de la base de datos establecer mecanismos de protección de las diferentes visiones de los datos asociadas a cada usuario, proporcionando elementos de creación y modificación de dichos usuarios. Adicionalmente, incorpora sistemas para la creación de copias de seguridad, carga de ficheros, auditoría, protección de ataques, configuración del sistema, etc. El lenguaje que implementa esta función es el **lenguaje de control de datos o DCL** (Data Control Language).



Utilización de los  
datos  
**DCL**

# Control y utilización de los datos

GRANT, permite otorgar permisos.

REVOKE, elimina los permisos que previamente se han concedido.

Pregunta, ¿Y a través de qué  
lenguaje podremos  
desarrollar estas funciones  
sobre la base de datos?



# SQL

Lo haremos utilizando el **Lenguaje Estructurado de Consultas (SQL: Structured Query Language)**.

Este lenguaje proporciona sentencias para realizar operaciones de DDL, DML y DCL.

SQL fue publicado por el ANSI en 1986 (American National Standard Institute) y ha ido evolucionando a lo largo del tiempo. Además, los SGBD suelen proporcionar otras herramientas que complementan a estos lenguajes como generadores de formularios, informes, interfaces gráficas, generadores de aplicaciones, etc.





# Tipos de datos en SQL

Algunos de los tipos de datos básicos de SQL son:

Varchar: Recibe cadena de palabras compuestas de letras, números y caracteres especiales.

Int es el principal tipo de datos de valores enteros de SQL Server. Con números enteros con o sin signo

Date: una fecha de calendario que contiene el año (de cuatro cifras), el mes y el día.

Time: La hora del día en horas minutos segundos (el valor predeterminado es 0).

# Actividad

Para ir familiarizándonos con las funciones de las bases de datos, buscar en internet una sentencia básica de cada tipo para ver su sintaxis



# Índice

1. Introducción.
2. Ficheros.
  1. Tipos de fichero según su estructura de almacenamiento.
  2. Tipos de soporte de almacenamiento.
3. Bases de datos.
  1. Definición.
  2. Modelos de bases de datos
  3. Tipos de bases de datos.
4. Sistema gestores de bases de datos.
  1. Funciones del SGBD.
  2. [Componentes del SGBD.](#)



# Componentes de una base de datos

Una vez descritas las funciones que un SGBD debe llevar a cabo, imaginarás que un SGBD es un paquete de software complejo que ha de proporcionar servicios relacionados con el almacenamiento y la explotación de los datos de forma eficiente. Para ello, cuenta con una serie de componentes que se detallan a continuación:

- **Lenguaje de la base de datos** → DDL, DML y DCL.
- **El diccionario de los datos** → Descripción de los datos almacenados. Se trata de información útil para los programadores de aplicaciones. Es el lugar donde se deposita la información sobre la totalidad de los datos que forman la base de datos. Contiene las características lógicas de las estructuras que almacenan los datos, su nombre, descripción, contenido y organización.
- **El gestor de la base de datos** → Es la parte de software encargada de garantizar el correcto, seguro, íntegro y eficiente acceso y almacenamiento de los datos. Es un intermediario entre el usuario y los datos. Es el encargado de garantizar la privacidad, seguridad e integridad de los datos, controlando los accesos concurrentes e interactuando con el sistema operativo.
- **Usuarios de la base de datos** → existen distintos perfiles de usuario.
  - Administrador de la base de datos.
  - Usuarios de la base de datos.
- **Herramientas de la base de datos** → Son un conjunto de aplicaciones que permiten a los administradores la gestión de la base de datos, de los usuarios y permisos, generadores de formularios, informes, interfaces gráficas, generadores de aplicaciones, etc.

# Actividad

Realiza una comparativa de los distintos SGBD actuales en el mercado, con sus características, ventajas e inconvenientes.



# Fin

# Actividad Entregable

Resumen del tema con los conceptos que consideres más importantes.

Subir al campus virtual un archivo PDF, nombrado de la siguiente manera:

**“DAW\_DNI\_nombre\_apellido1\_apellido2.pdf” o**

**“DAM\_DNI\_nombre\_apellido1\_apellido2.pdf”**



# Actividad Entregable

Elaborar un examen tipo test de 10 preguntas relacionadas con el tema 1.

No se permite repetir preguntas vistas en el test realizado en clase con la aplicación Kahoot.

Las preguntas pueden ser bien de teoría o relacionada con las prácticas con alguna pregunta de algún diagrama sencillo E/R visto o no en clase.

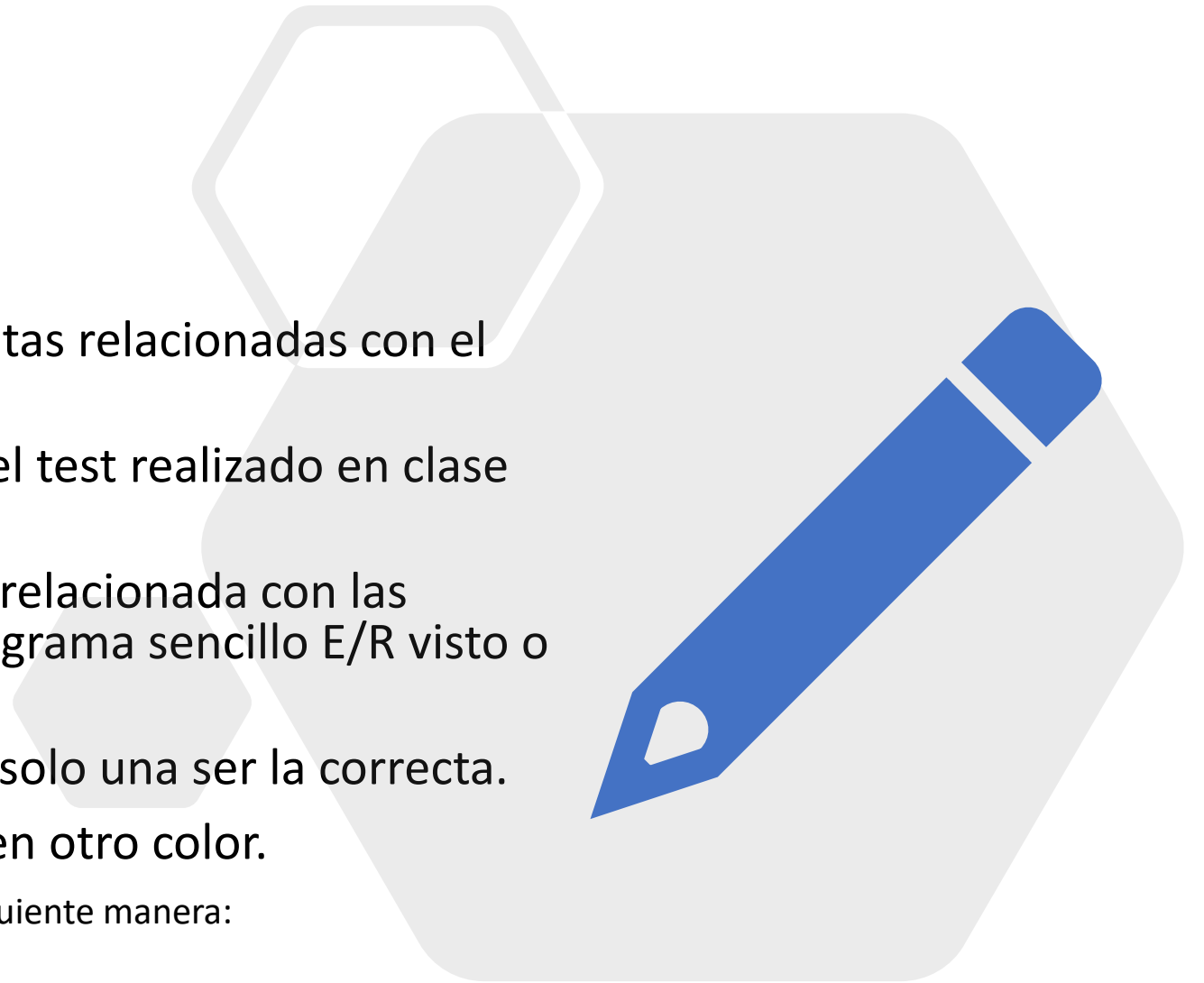
Las preguntas han de tener 4 respuestas y solo una ser la correcta.

Marcar la respuesta correcta en negrita o en otro color.

Subir al campus virtual un archivo PDF, nombrado de la siguiente manera:

**“DAW\_DNI\_nombre\_apellido1\_apellido2.pdf”** o

**“DAM\_DNI\_nombre\_apellido1\_apellido2.pdf”**





# Kahoot!

## REPASO