

# TEMA I

## Sistemas de Almacenamiento de la Información

### INDICE

1. Introducción
  1. Evolución de los Sistemas de Almacenamiento de la Información.
2. Ficheros
  1. Introducción: Concepto Sobre los Ficheros.
  2. Tipos de Ficheros
    1. Fichero Secuencial.
    2. Acceso Directo.
    3. Ficheros Indexados
3. Bases de Datos
  1. Introducción
  2. Independencia de los datos
  3. Modelos de Bases de Datos
    1. Modelo Entidad-Relación
    2. Modelo Jerárquico
    3. Modelo en red
    4. Modelo Relacional
    5. Modelo Orientado a Objeto
4. Sistemas Gestores de Bases de Datos
  1. Tipos
  2. Conceptos y Funciones
5. Arquitecturas de BD
  1. Arquitectura Cliente/Servidor
  2. BD Distribuidas

## 1. Introducción

¿Te has preguntado alguna vez dónde y de qué manera se almacenan y gestionan los datos que utilizamos diariamente? Si pensamos en cualquier acción de nuestra vida cotidiana, o si analizamos la mayoría de los ámbitos de actividad, nos encontramos que la utilización de las bases de datos está ampliamente extendida. Éstas, y los datos contenidos en ellas, serán imprescindibles para llevar a cabo multitud de acciones.

¿Crees que no es para tanto? Piensa en las siguientes situaciones:

- Cuando seleccionamos nuestro canal favorito en la TDT.
- Al utilizar la agenda del móvil para realizar una llamada telefónica.
- Cuando operamos en el cajero automático.
- Al solicitar un certificado en un organismo público.
- Cuando acudimos a la consulta del médico.
- Al inscribirnos en un curso, plataforma OnLine, etc.
- Si utilizas un GPS.
- Cuando reservamos unas localidades para un evento deportivo o espectáculo.
- Si consumimos ocio digital.
- Cuando consultamos cualquier información en Internet. (Bibliotecas, enciclopedias, museos, etc.)
- Al registrarte en una página de juegos OnLine, redes sociales o foros.
- Incluso, si tienes coche, puede ser que éste incorpore alguna base de datos.



Imagen 1

Suponemos que no es necesario que continuemos más para darnos cuenta de que casi todo lo que nos rodea, en alguna medida, está relacionado con los datos, su almacenamiento y su gestión. El gran volumen de datos que actualmente manejamos y sus innumerables posibilidades requieren de la existencia de técnicos perfectamente formados y capaces de trabajar con ellos.

### 1. Evolución de los Sistemas de Almacenamiento de la Información.

La mejor manera de entender el concepto de **bases de datos** es hacer una pequeña aproximación histórica a su evolución:

## Fichero manual

Hasta hace más o menos treinta años, para guardar y hacer un seguimiento de la información se hacía mediante soporte de papel. Se utilizaba un sistema de ficheros manual que se componía de un conjunto de carpetas etiquetadas cuyo contenido estaba relacionado y se guardaban en un armario o *archivo*.

Por ejemplo, en los colegios o **centros de enseñanza**, se guardaban así archivados los expedientes de cada uno de los alumnos con sus datos personales, notas de cada curso, etc. En una **consulta médica** se guardaban en carpetas las historias clínicas de los pacientes. Las **empresas** utilizaban *fichas* escritas con la información de los clientes o proveedores, etc.



Imagen 2

Este sistema podía ser útil cuando el volumen de datos manejado no era muy grande y se podía extraer la información que se necesitaba con cierta facilidad, pero a medida que el archivo manual aumentaba y que la información que se necesitaba era más compleja fue necesario sustituir este sistema por otro informatizado.

## Sistemas de ficheros

Las primeras aplicaciones que manejaban los datos utilizando el ordenador se concentraban en tareas propias de oficina como gestión de entradas y salidas de pedidos, nóminas, facturación, etc. Los datos necesarios se guardaban en ficheros en el ordenador y estas aplicaciones accedían a ellos para obtener los informes que se solicitaban de cara a la toma de decisiones en la empresa.

Teníamos por una parte los ficheros necesarios para contener los **datos** con una estructura determinada y por otro los **programas de aplicación** que accedían a estos datos para producir información.

Normalmente, los programas trabajaban de manera secuencial sobre un solo archivo maestro, que estaba almacenado en una cinta magnética (no en un disco duro), y generaban otro archivo como resultado.

Hasta no hace mucho tiempo el lenguaje de programación COBOL era el más extendido para implementar **aplicaciones de gestión** tanto **empresariales**, como para los **bancos**, etc; por su fiabilidad para trabajar con distintos tipos de ficheros.

## Sistemas de bases de datos

A partir de innovaciones tecnológicas como terminales, discos duros y redes de comunicación, los programas informáticos tuvieron que:

- Implementar la posibilidad de realizar consultas y actualizaciones de los mismos datos simultáneamente por parte de diferentes usuarios.
- Interrelacionar los archivos.
- Eliminar la redundancia, es decir, la repetición innecesaria de datos, que además de resultar ineficiente pone en riesgo su coherencia.

A raíz de estas necesidades, a finales de los años 60 surgen **las bases de datos**: Como conjuntos de archivos relacionados entre sí, compartidos por diferentes procesos y usuarios simultáneamente, y con estructuras complejas.

En una **base de datos** se almacenan datos y los programas que utilicen esos datos no se tienen que preocupar del almacenamiento físico de los mismos. Cualquier cambio en la estructura de los datos no afecta a los programas de aplicación que los utilicen.

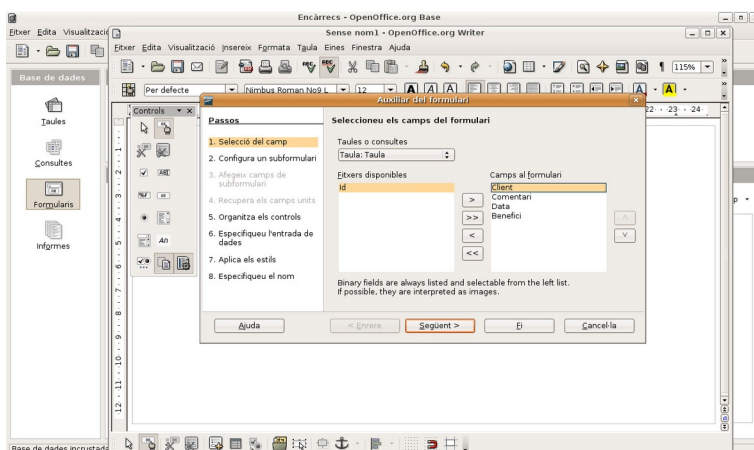


Imagen 3

Podremos utilizar una base de datos para cosas tan sencillas como mantener un registro de todos los **vídeos de nuestras películas**, nuestra **biblioteca personal** o la **música** que tenemos almacenada o tan complicadas como llevar toda la gestión de nuestra empresa.

## 2. Ficheros

### 1. Introducción: Concepto Sobre los Ficheros.

Al llevar a cabo una primera informatización, se pasó de tener los datos en formato papel a poder acceder a ellos de manera mucho más rápida a través del ordenador. En ese momento, la informática adaptó sus herramientas para que los elementos que el usuario maneja en el ordenador se parezcan a los que utilizaba manualmente. La información debía ser trasladada desde el papel al formato digital y por lo general, era necesario almacenarla para su posterior recuperación, consulta y procesamiento. El elemento que permitió llevar a cabo el almacenamiento de datos de forma permanente en dispositivos de memoria masiva fue **el fichero o archivo**.

La representación informática más frecuente en el ámbito de las BD es la representación tabular (*en forma de tabla*), la que se implementa habitualmente en archivos que se estructuran en **registros** y **campos**:

Los ficheros están formados por **registros** que contienen datos relativos a un mismo elemento u objeto (**entidad**). En el ejemplo del taller mecánico, supongamos que quieren almacenar los datos de los clientes que empiezan a ser habituales en un fichero de CLIENTES. De ellos interesa mantener, por ejemplo, el nombre y los apellidos, el teléfono y la dirección.

- En el ejemplo, un **registro** lo formarían todos los datos relativos a un cliente concreto: su nombre, sus apellidos, su teléfono y su dirección. (Por ejemplo, los datos de *Luis Gómez*)
- **Campo** o **atributo**: equivale a una columna de la representación tabular. Los **campos** o **atributos** son las propiedades. En el ejemplo, definiríamos un campo para guardar el nombre, en otro campo para el teléfono, etc.

Por definición, un **Fichero** es un conjunto de registros relacionados. En nuestro ejemplo, el fichero estaría formado por los datos de todos los clientes del taller:

FICHERO DE CLIENTES					
CAMPOS:	DNI	NOMBRE	APELLIDOS	DIRECCIÓN	TELEFONO
Registro 1	73456789K	Luis	Ruiz Gómez	C/ Pez, 45	678899090
Registro 2	67345890L	Maria	Pérez García	C/ Arriba 34, 1º C	654232323
Registro 3	123456789U	Jorge	Martinez	C/Alameda 34	666898789
Registro 4	23789654M	Mar	Pons Fons	C/ Paseo 2 3º A	646 234578

## 2. Tipos de Ficheros

El **acceso a un registro** es el procedimiento que se utiliza para seleccionarlo. Los tipos de acceso son:

- **Secuencial**: los registros se leen uno detrás de otro desde el principio del fichero hasta localizar el registro buscado o hasta el final del fichero.
- **Directo**: permite seleccionar a un registro sin tener que leer los anteriores, accediendo directamente a él mediante su clave.
- **Indexado**: para seleccionar un registro consultamos previamente de forma secuencial en una tabla que contiene la clave más alta y la dirección de comienzo de cada bloque de registros. Una vez localizado se utiliza el acceso directo a ese bloque de registros y, dentro del bloque, la lectura secuencial hasta localizarle.

- **Dinámico:** permite el acceso directo o por índice a un registro y a partir de ese se accede a los demás de forma secuencial.

### 1. Fichero Secuencial.

En un fichero con organización secuencial 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.

Este tipo de ficheros pueden utilizar dispositivos o soportes no direccionables o de acceso secuencial, como son las cintas magnéticas de almacenamiento de datos.

#### Las ventajas de los ficheros con organización secuencial son:

- Rapidez en el acceso a un bloque de registros contiguos
- No se desperdicia espacio en el dispositivo de almacenamiento porque no hay huecos
- Se pueden utilizar cualquier tipo de registros: de longitud fija, variable o indefinida.

#### Los inconvenientes de este tipo de organización son:

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

Para realizar una **consulta** hay que crear un proceso en el que se compare el valor del campo que se pretende localizar con el valor del mismo campo correspondiente a cada registro leído del fichero.

La **adición** de registros se realiza a continuación del último registro ya existente. No se pueden insertar nuevos registros.

No se pueden **eliminar** registros. Para eliminar un registro se marca de modo que no se muestre, pero el registro existe y ocupa espacio en el dispositivo del almacenamiento. (borrado lógico)

Para **mantener ordenado** y compactado el fichero, hay que crear un fichero nuevo a partir del existente.

Dirección de Memoria	Marca de Borrado	NOMBRE	APELLIDOS	DIRECCIÓN	TELEFONO
1200		Luis	Ruiz Gómez	C/ Pez, 45	678899090
1300	*	Maria	Pérez García	C/ Arriba 34, 1º C	654232323
1400		Jorge	Martinez	C/Alameda 34	666898789
1500		Mar	Pons Fons	C/ Paseo 2 3º A	646 234578

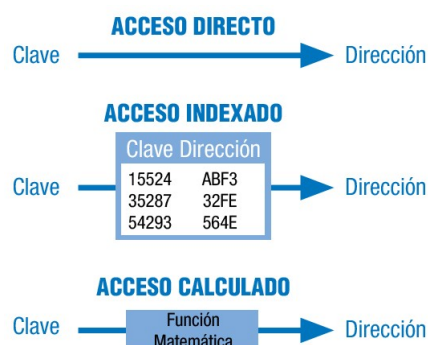
## 2. Acceso Directo.

En este tipo de ficheros se puede acceder a un registro a través de una **clave** que forma parte del registro como un campo más. Estos archivos deben almacenarse en dispositivos de memoria masiva de acceso directo, como por ejemplo los discos magnéticos.

**Campo clave:** campo que permite identificar y localizar un registro de manera ágil y organizada.

Cada uno de los registros se guarda en una posición física, que dependerá del espacio disponible en memoria masiva, de ahí que la distribución de los registros sea aleatoria dentro del soporte de almacenamiento. Para acceder a la posición física de un registro se utiliza una dirección o índice, no siendo necesario recorrer todo el fichero para encontrar un determinado registro.

A través de una transformación específica aplicada a la clave, se obtendrá la dirección física en la que se encuentra el registro. Según la forma de realizar esta transformación, existen diferentes modos de acceso:



En el acceso directo la clave coincide con la dirección, debiendo ser numérica y comprendida dentro del rango de valores de las direcciones. Es el método más rápido.

Otras características fundamentales de los ficheros de acceso directo o aleatorio son:

- Posicionamiento inmediato.
- Registros de longitud fija.
- Los registros se borran colocando un cero en la posición que ocupan.
- Permiten la utilización de algoritmos de compactación de huecos.
- Los archivos se crean con un tamaño definido, es decir, con un máximo de registros establecido durante la creación.
- Se usan cuando el acceso a los datos de un registro se hace siempre empleando la misma clave y la velocidad de acceso a un registro es lo que más nos importa.
- Permiten la actualización de los registros en el mismo fichero, sin necesidad de copiar el fichero.
- Permiten realizar procesos de actualización en tiempo real.



### 3. Ficheros Indexados

En esta organización los registros con los datos se graban en un fichero secuencialmente, 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.

Estos índices son similares a los de los libros. Si nos interesa leer un capítulo concreto podemos recurrir al índice que nos dice en que página comienza, y abrimos el libro por esa página, sin tener que mirar en todas las páginas anteriores para localizarlo.

En este tipo de ficheros se distinguen tres áreas:

<i>Área Primaria:</i>	<p>En esta área se escriben los registros ordenados ascendentemente por el valor de su clave. Esta área del fichero está dividida en segmentos. Cada segmento almacena “n” registros consecutivos y almacenados en posiciones contiguas. Es un área de organización secuencial, donde el acceso a cada registro se realiza en una doble operación:</p> <ul style="list-style-type: none"> <li>• <b>Acceso directo</b> al segmento donde se haya ubicado el registro buscado</li> <li>• <b>Acceso secuencial</b> posteriormente, a los registros del segmento, hasta localizar el registro buscado o alcanzar el final del segmento, en caso de que no se halle.</li> </ul>
<i>Área de Índices</i>	<p>Es creada por el sistema al mismo tiempo en que se almacenan los datos. En esta área los registros están formados por dos campos:</p> <ul style="list-style-type: none"> <li>• El valor del campo <b>clave del último registro</b> de un bloque o segmento. Los bloques están constituidos por un número fijo de registros consecutivos.</li> <li>• El segundo campo contiene la <b>dirección de comienzo</b> de cada uno de los segmentos en los que se halla dividida el área primaria</li> </ul>
<i>Área de Overflow</i>	<p>O área de excedentes. Es la zona destinada a contener los registros almacenados posteriormente a la creación del fichero, por lo que no han sido incluidos en el área primaria. Estos registros tendrán claves intermedias a las de los registros previamente almacenados en el área primaria</p>

- El **acceso a los registros** se realiza mediante una consulta secuencial al área de índices para determinar el segmento donde se encuentra el registro buscado.

Con el valor del segmento seleccionado, se recorren secuencialmente los registros de ese segmento. Si el registro no está comprendido en el segmento, se continúa la búsqueda de forma secuencial en el área de overflow hasta la localización del registro o hasta terminar de leer los registros de esa área.

- El **borrado**: la eliminación de los registros debe realizarse mediante marcas. Se generan huecos que realmente son posiciones de memoria ocupadas por registros marcados pero que no han sido eliminados

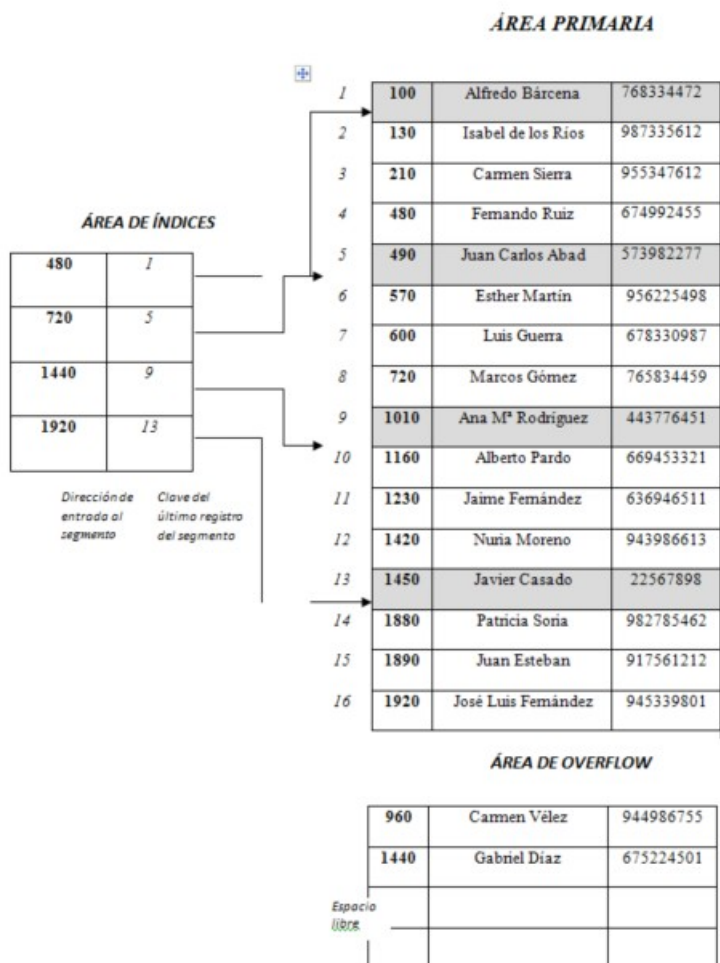


físicamente del fichero. La única posibilidad de eliminar estos huecos, es en futuras operaciones en las cuales necesitemos reorganizar el fichero.

- La **inserción** de registros se hace en el área de overflow. No está permitida la instrucción de nuevo registro en el área primaria después de la creación del fichero.
- En esta organización cuando el numero de registros borrados es grande, o las cadenas de desbordamientos son largas su utilización deja de ser eficiente, siendo necesario **reorganizar el archivo**.

Esto es un ejemplo de como se organiza un archivo secuencial indexado:

Esquema de un archivo con organización secuencial indexada



## Recomendación

En el siguiente [enlace](#) tenéis más información sobre Organización de Ficheros y Métodos de Acceso.

### 3. Bases de Datos

#### 1. Introducción

Como hemos visto anteriormente, los ficheros permiten organizar y memorizar conjuntos de datos del mismo tipo o naturaleza con una determinada estructura. Pero tienen serios inconvenientes: como información duplicada, incoherencia de datos, fallos de seguridad, etc.

Para solucionar estos problemas 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.

Una **base de datos** se define como una estructura de datos interrelacionados a los que pueden acceder varios usuarios simultáneamente.

Las **ventajas** que aportan las bases de datos sobre los sistemas de ficheros son:

- **Control sobre la redundancia de datos.** En los sistemas de bases de datos los datos se almacenan una única vez (puede haber excepciones controladas).
- **Consistencia de datos.** Eliminando o controlando las redundancias de datos se reduce en gran medida el riesgo de que haya inconsistencias. Si un dato está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez, y está disponible para todos los usuarios inmediatamente.
- **Acceso múltiple.** Diversos usuarios o aplicaciones podrán acceder a la base de datos, sin que existan problemas en el acceso o los 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.

Id_genero	Nombre	Id_pelicula	Título	Director	Año	Formato	Visionada
1	Ciencia-Ficción	1	Blade Runner	Ridley Scott	01/01/83	DVD	<input checked="" type="checkbox"/>
2	Aventuras	2	La Guerra de las Galaxias	George Lucas	01/01/77	VHS	<input checked="" type="checkbox"/>
3	Histórico	3	Indiana Jones en Busca del Arca Perdida	Steven Spielberg	01/01/81	DVD	<input checked="" type="checkbox"/>
4	Comedia	4	Millon Dolar Baby	Clint Eastwood	01/01/04	DVD	<input checked="" type="checkbox"/>
5	Drama	5	La Comunidad del Anillo	Peter Jackson	01/01/01	DVD	<input checked="" type="checkbox"/>
6	Thriller	6	Belleza Robada	Bernardo Bertolucci	01/01/96	VHS	<input checked="" type="checkbox"/>
7	Suspense	7	Gladiator	Ridley Scott	01/01/00	DVD	<input checked="" type="checkbox"/>
8	Terror	8	Rocky	John G. Avildsen	01/01/76	VHS	<input checked="" type="checkbox"/>
9	Fantasia	9	Los Lunes al Sol	Fernando León	01/01/02	DVD	<input type="checkbox"/>
		10	Mar Adentro	Alejandro Amenábar	01/01/04	DVD	<input type="checkbox"/>

Img06

- **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 la recuperación de los datos de forma fiable.
- **Independencia de datos:** Lo veremos en el siguiente apartado.

## 2. Independencia de los datos

Uno de los principales objetivos de las BD es proporcionar una **visión abstracta de los datos**. Con este fin, el sistema oculta a los usuarios ciertos detalles relativos al almacenamiento y mantenimiento de los datos. Por un lado se facilita el trabajo, pero también se garantizan ciertos aspectos en materia de seguridad.

En 1975, el comité ANSI-SPARC propuso una arquitectura de tres niveles para los SCBD cuyo objetivo principal era separar los programas de aplicación de la base de datos física. En esta arquitectura el esquema de una base de datos se define con 3 niveles de abstracción:

- Físico (interno)
- Conceptual
- Externo

### *Nivel Físico o Interno*

Describe cómo se almacenan físicamente las estructuras de datos. Este esquema especifica: 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, las rutas de acceso, etc.

**La descripción del nivel físico se realiza mediante un esquema interno, que es un conjunto de definiciones y reglas que permite definir las tablas y cómo se relacionan entre sí.**



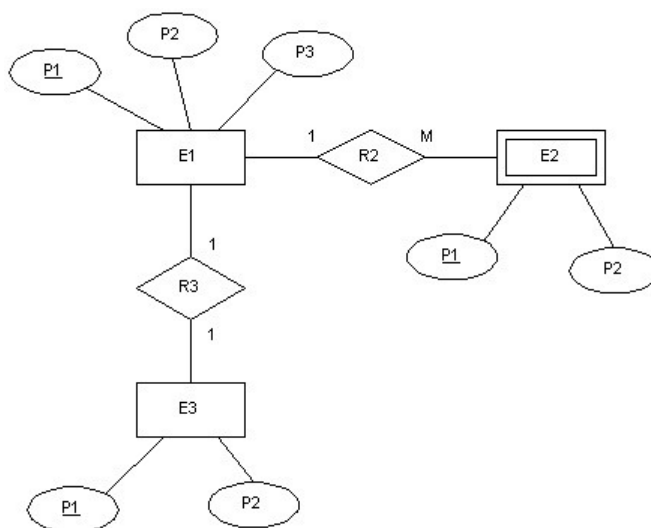
Los usuarios que trabajan a este nivel son los diseñadores de la base de datos o los Administradores. No es frecuente trabajar a este nivel. Sólo se hace cuando se necesitan optimizaciones en la estructuración de los datos a bajo nivel.

### *Nivel Conceptual*

O nivel lógico global. Describe la organización de los datos en la base de datos y las relaciones existentes entre ellos.

**La descripción de este nivel se realiza mediante un esquema conceptual, que permite definir las entidades, los atributos y sus propiedades, las relaciones, operaciones de los usuarios y las restricciones y reglas de validación.**

Este nivel se ocupa de la estructura organizacional de los datos sin ocuparse de las estructuras físicas de almacenamiento. A este nivel los usuarios que intervienen son los programadores y administradores de BD, encargados de crear las estructuras lógicas necesarias para guardar la información.



### *Nivel Externo*

O de usuario. Describe la base de datos como es percibida por los usuarios.

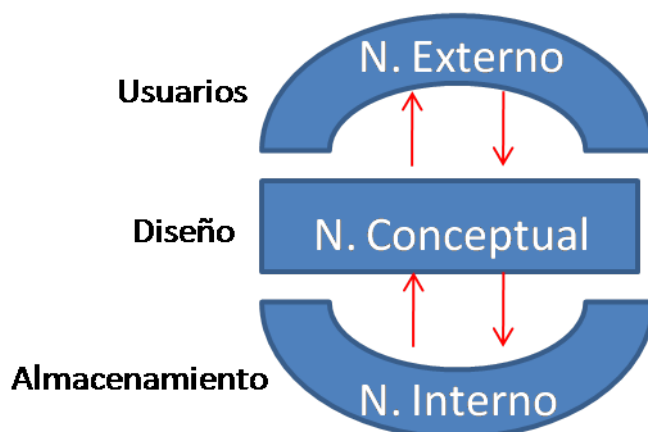
La mayoría de los usuarios no necesitan conocer toda la estructuración lógica de la BD con que trabajan. A los usuarios hay que ocultarles la complejidad interna de las bases de datos, de modo que les sea transparente.

Cada usuario verá una base de datos (esquema externo) distinto según sea el nivel de acceso que se le haya concedido, ya que tendrá acceso a aquéllos datos que necesite, a las relaciones que emplee y las restricciones de uso que se le hayan definido.

Los objetos a los que puede acceder un usuario o grupo de usuarios forman su nivel externo: tablas, vistas, formularios, informes, etc.

El nivel externo es la percepción de la base de datos por el usuario, de modo que hay tantos niveles externos distintos como grupos de usuarios haya en la base de datos.

Al esquema externo también se le llama **vista**.



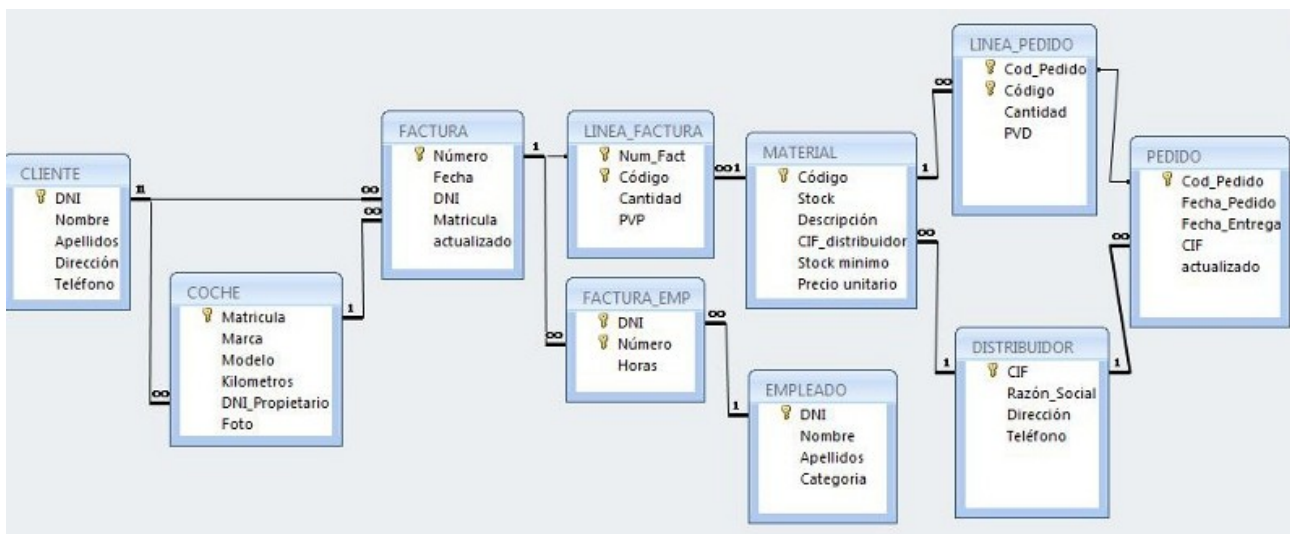
*Con la arquitectura a tres niveles* se introduce el concepto de **independencia de datos**. Se definen dos tipos de independencia:

- **Independencia lógica:** se refiere a la posibilidad de modificar el esquema conceptual de la base de datos sin tener que modificar los esquemas externos, ni los programas. Por ejemplo: si se borra una entidad las vistas que no se refieran a ella no se alteran.
- **Independencia física:** se refiere a la posibilidad de modificar el esquema interno sin tener que modificar ni el esquema conceptual ni los esquemas externos. Por ejemplo: se pueden reorganizar los archivos físicos o añadir nuevos archivos de datos para mejorar el rendimiento.

### 3. Modelos de Bases de Datos

Llamamos modelo a un instrumento que se aplica a una parcela del mundo real para obtener una estructura de datos a la que denominamos esquema. *Modelar* consiste en definir un mundo abstracto de forma que las conclusiones que se puedan sacar de él coincidan con las manifestaciones del mundo real. Un modelo de datos es un conjunto de herramientas conceptuales que permiten describir los datos, sus relaciones y las reglas de integridad que deben cumplir.

Este esquema se especifica durante el diseño, y no es de esperar que se modifique a menudo. Sin embargo, los datos que se almacenan en la base de datos pueden cambiar con mucha frecuencia: se insertan datos, se actualizan, etc.



Es muy importante que el esquema sea correcto y se debe tener muchísimo cuidado al diseñarlo.

En su **evolución** las bases de datos se han basado en 3 *modelos de datos*.

- Jerárquico.
- En red.
- Relacional.

Los primeros sistemas, introducidos en los años 60 se basaron en el **modelo jerárquico** que estructura todas las relaciones entre los datos como jerarquías. A finales de los 60 aparecieron los sistemas basados en el **modelo en red**. En estos dos modelos los datos se relacionaban con punteros físicos. El **modelo de red** sustituyó al **modelo jerárquico** porque podía representar relaciones M:N (muchos a muchos).

En 1970, Codd publicó un artículo que abrió una nueva perspectiva para los sistemas de gestión de la información argumentando que los datos deberían relacionarse mediante interrelaciones naturales y lógicas. Los datos se representan como tablas denominadas relaciones y se les aplica el cálculo y el álgebra relacional. Se denominó **modelo relacional** y es el estándar mas empleado en la actualidad.

El **modelo relacional** reemplazó al **modelo de red** porque permite representar la realidad de una forma más simple, ofrece mayor independencia de los datos y admite consultas utilizando un lenguaje relativamente fácil.

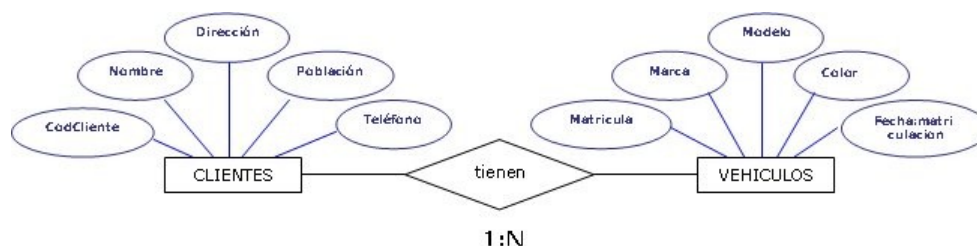
En estos momentos y a medida que las aplicaciones se van volviendo más complejas las bases de datos evolucionan en dos direcciones fundamentalmente: el **modelo de bases de datos orientadas a objetos** y el **modelo relacional ampliado** que incluye muchas características del modelo orientado a objetos. Ambos intentan recoger la mayor información semántica posible, pero desde orientaciones distintas: el modelo orientado a objetos se enfoca más a aplicaciones de ingeniería y científicas muy especializadas y el modelo relacional ampliado se dirige más a aplicaciones de negocios.

## 1. Modelo Entidad-Relación

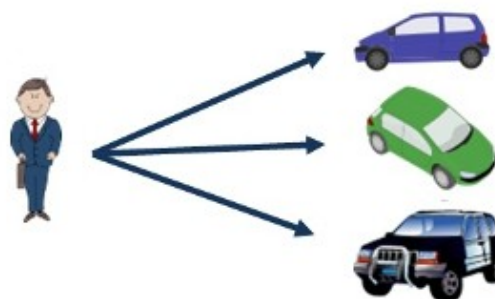
Este modelo fue propuesto por Peter Chen en 1976 para la representación conceptual de los problemas del mundo real. Es un modelo muy extendido y potente para la representación de los datos. Se simboliza haciendo uso de **grafos** y de **tablas**.

Se basa en una percepción del mundo real que consiste en una colección de objetos llamados **entidades** y las **relaciones** entre esos objetos. Cada entidad representa un objeto que se distingue de otros por tener un conjunto de **atributos** propio.

En el ejemplo tendríamos las **entidades** CLIENTES y VEHICULOS y la **relación** establecida entre esas dos entidades: tienen.



- Como **atributos** de la **entidad** CLIENTES tenemos: CodCliente, Nombre, Dirección, Población y Teléfono.
- Los **atributos** de la **entidad** VEHICULOS son: Matricula, Marca, Modelo, Color y Fecha matriculación.
- La **relación** tienen describe la asociación entre los datos de ambas entidades. Cuando la relación es **1:N** (uno a muchos) significa que un cliente podrá tener muchos vehículos (más de uno); mientras que cada vehículo es de un solo cliente.



**Un cliente puede tener varios vehículos. Cada vehículo es propiedad de un solo cliente.**

El modelado entidad-relación es una técnica para el modelado de datos utilizando diagramas entidad-relación. No es la única técnica pero sí la más utilizada. Mediante una serie de procedimientos se



puede pasar del modelo entidad-relación (E-R) a otros, como por ejemplo el **modelo relacional** que veremos posteriormente.

## 2. Modelo Jerárquico

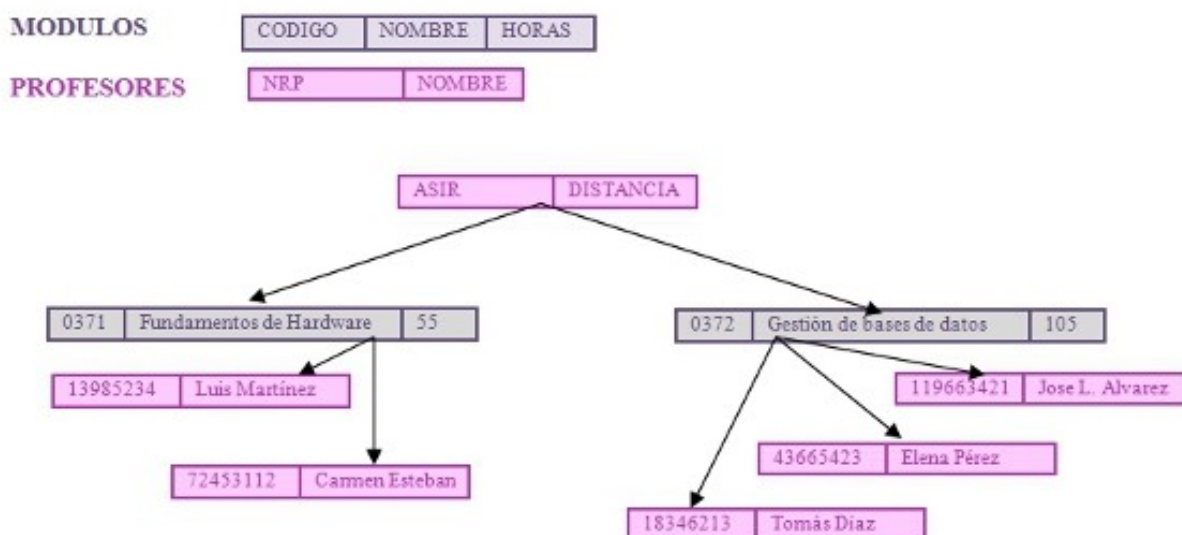
Se le llama también modelo en árbol, ya que utiliza para su representación una estructura de tipo árbol invertido.

Una base de datos jerárquica es un conjunto de registros lógicamente organizados con una estructura de árbol invertido. Dentro de la jerarquía el nivel superior se percibe como el "padre" de los registros situados debajo de él, de forma que:

- Cada padre puede tener muchos hijos
- Cada hijo sólo tiene un padre

Las **características** principales son:

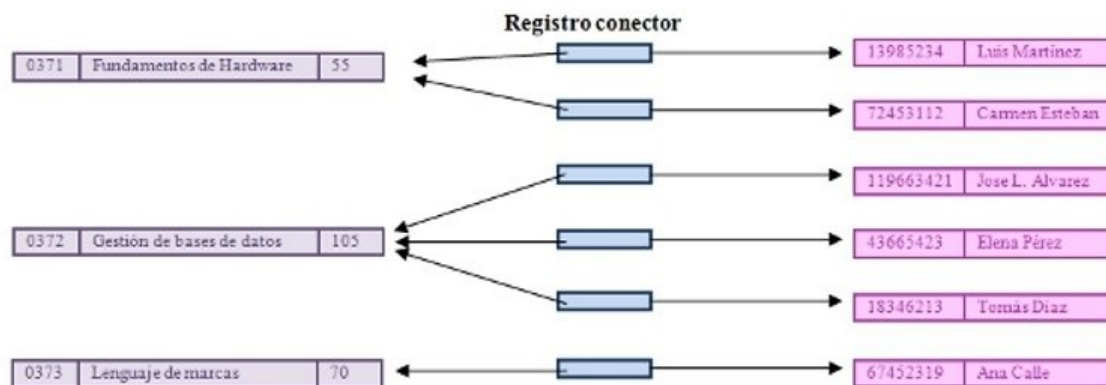
- Una colección de árboles forma una base de datos.
- A los registros se les denomina segmentos o **nodos**, que contienen atributos o **campos**.
- Los nodos están organizados en niveles. Cada nodo contiene los campos comunes a los nodos hijos, vinculados a él. Al nodo más alto en la jerarquía o estructura de árbol se le denomina **raíz**.
- Las relaciones entre registros se representan mediante **arcos o lazos**.



## 3. Modelo en red

Es un modelo de datos propio de los sistemas comerciales de los años 70. Fue creado para representar relaciones más complejas y eficientes que las del modelo jerárquico pero más flexible, ya que permitía que los nodos tuvieran más de un solo padre.





En este modelo se pueden representar perfectamente cualquier tipo de relación entre los datos, pero hace muy complicado su manejo y es por ello que no ha tenido tanta fortuna como su predecesor.

#### 4. Modelo Relacional

Fue desarrollado por Codd para IBM en los años 70 y actualmente es el modelo más utilizado para modelar problemas reales y administrar datos dinámicamente.

Su principal ventaja es que permite que el usuario y el diseñador operen en un entorno que se percibe como un **conjunto de tablas** y los detalles físicos complejos los maneja el sistema.

##### *Características:*

- Este modelo representa los datos y las relaciones entre ellos como una colección de **tablas**.
- De manera simple, una relación representa una tabla que no es más que un conjunto de **filas**, cada fila es un conjunto de **campos** y cada campo representa un valor que describe el mundo real.
- Las tablas son independientes pero se relacionan mediante un vínculo común.
- Proporciona una redundancia y una inconsistencia mínima.
- La independencia de datos de las aplicaciones y del dispositivo de almacenamiento.

**Ejemplo:** los datos de los CLIENTES se guardan en una tabla y los de los VEHICULOS en otra, de forma independiente. El vínculo entre las tablas CLIENTES y VEHICULOS es el CodCliente (código de cliente), que permite establecer que el cliente Raquel Marcos es propietario de dos vehículos: un Ford Focus y un Suzuki Vitara. En este ejemplo la relación es 1:N, un cliente puede tener muchos vehículos.

CLIENTES				
CodCliente	Nombre	Dirección	Población	Teléfono
1	Francisco Álvarez	C/ La Mata 9.	Alcántara	925767788
2	Raquel Marcos	C/ La Amapola 7.	Toledo	925998811
3	Carlos Revuelta	C/ La Arboleda 12.	Madrid	919090771
4	José María Cabello	C/ La plaza 72	Madrid	914409071
5	Jorge Peña	C/ Fresno, 23	Guadalajara	949788896
6	Dolores Manzano	C/ Autonomía, 8	Madrid	916767560

VEHICULOS					
Matricula	Marca	Modelo	Color	Fecha Matriculación	CodCliente
4534 FNG	Ford	Focus	Negro	14/04/2007	2
1203 CLL	Cotroën	C4	Magenta	23/08/2005	5
3367 GHB	Suzuki	Vitara	Cobalto	16/05/2009	2
1004 JLG	Kia	Rio	Rojo	02/07/2009	6
6709 BFG	Peugeot	206	Gris plata	12/10/2006	3

### Ventajas:

- Independencia estructural: los programadores, usuarios, diseñadores no necesitan conocer la ruta de acceso a los datos. Los cambios en la estructura de la base de datos no afectan a la capacidad de acceso a los datos.
- Simplicidad conceptual: debido a que el sistema se encarga del almacenamiento físico de los datos, los diseñadores se centran en la representación lógica de la base de datos.
- Facilidad para diseñar, administrar y utilizar la base de datos, debido a la independencia de los datos y estructural.
- Capacidad para hacer consultas de forma rápida y sencilla (mediante el lenguaje SQL)

### Desventajas:

- Requiere una elevada inversión en hardware y software para evitar que sea lento, aunque esto está cambiando gracias a la evolución de la capacidad del hardware y a las mejoras de los sistemas operativos.
- El diseño deficiente es bastante común debido a la facilidad de uso de esta herramienta para personas inexpertas. A medida que la base de datos crece, si el diseño es inapropiado, el sistema es más lento y se producen anomalías.
- Debido a la facilidad de uso, los usuarios finales a menudo crean subconjuntos de bases de datos que pueden producir datos inconsistentes.

## 5. Modelo Orientado a Objeto

Los modelos de bases de datos intentan representar cada vez con más fidelidad los problemas del mundo real que cada vez son más complejos. Uno de los modelos que se han desarrollado recientemente (años 90) es el modelo de bases de datos orientado a objetos.

Se denomina así porque su estructura básica es un **objeto**, que recoge tanto datos como sus relaciones. Un objeto se describe como un conjunto de hechos, pero incluye también información sobre la

relación que tienen los hechos dentro del objeto y con otros objetos; así como todas las operaciones que puedan ser realizadas en él.

El modelo de datos orientado a objetos está basado en los siguientes componentes:

- La estructura básica es un **objeto**: equivale a una entidad individual del modelo Entidad-Relación.
- Una **clase** es la definición del objeto. Esta definición explica los **métodos** y **atributos** que tiene. Las clases se organizan en una **jerarquía de clase** que se parece a un árbol invertido donde cada clase tiene solo un padre. Por ejemplo la clase cliente y la clase proveedor comparten una clase: persona.
- Un **método** representa una acción del mundo real. Por ejemplo: localizar el nombre de un cliente, cambiar el teléfono de un cliente o imprimir su dirección. Son equivalentes a los procedimientos en un lenguaje de programación. Definen el comportamiento de un objeto.
- La **herencia** es la capacidad de un objeto de heredar los atributos y los métodos de los objetos que están sobre él en una jerarquía de clase. Por ejemplo las clases cliente y proveedor, como subclasses de la clase persona heredarán los atributos de la clase persona.

Ejemplo de representación de una factura emitida a nuestros clientes.

FACTURA se representa incluyendo dentro del cuadro todos los atributos y relaciones con otros objetos.

En cada factura un cliente puede comprar varios artículos



#### 4. Sistemas Gestores de Bases de Datos

El Sistema gestor de la base de datos (SGBD) es una aplicación que permite a los usuarios definir, crear y mantener la Base de datos

Para que este sistema sea efectivo debe cumplir:

- Los datos pueden estar compartidos entre distintas personas, entre distintas localidades geográficas, etc.
- El uso de los datos debe estar controlado. El control lo facilita el SGBD y lo realizan los administradores.
- Los datos se integran de forma lógica, eliminando las redundancias y manteniendo la consistencia. La estructura lógica hace que se pueda mantener la consistencia entre muchos ficheros diferentes.



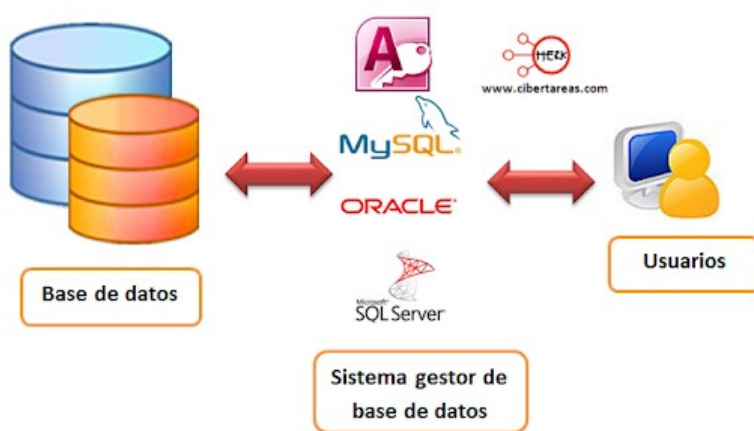
El SGBD permite a los usuarios la creación y el mantenimiento de una base de datos, facilitando la definición, construcción y manipulación de la información contenida en éstas.

- **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.
- 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.
- 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.

### 1. Tipos

¿Qué tipos de SGBD existen? Para responder a esta pregunta podemos realizar la siguiente clasificación, atendiendo a diferentes criterios:

- A. El primer criterio que se suele utilizar es **por el modelo lógico en que se basan**. Actualmente, el modelo lógico que más se utiliza es el **relacional**. Los modelos en red y jerárquico han quedado obsoletos. Otro de los modelos que más extensión está teniendo es el modelo **orientado a objetos**.



- B. El segundo criterio de clasificación se centra en el **número de usuarios** a los que da servicio el sistema:
- Monousuario:** sólo atienden a un usuario a la vez, y su principal uso se da en los ordenadores personales.
  - Multiusuario:** entre los que se encuentran la mayor parte de los SGBD, atienden a varios usuarios al mismo tiempo.
- C. El tercer criterio se basa en el **número de sitios en los que está distribuida la base de datos**:
- Centralizados:** sus datos se almacenan en un solo computador. 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.
  - Distribuidos (Homogéneos, Heterogéneos):** la base de datos real y el propio software del SGBD pueden estar distribuidos en varios sitios conectados por una red. Los sistemas 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 sistemas 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.

D. El cuarto criterio toma como referencia **el coste**. La mayor parte de los paquetes cuestan entre 10.000 y 100.000 euros. Los sistemas monousuario más económicos para microcomputadores cuestan entre 0 y 3.000 euros. En el otro extremo, los paquetes más completos cuestan más de 100.000 euros.

I. En este criterio también se podrían añadir los **SGBD libres/comerciales**.

E. El quinto, y último, criterio establece su clasificación **según el propósito**:

I. **Propósito General:** pueden ser utilizados para el tratamiento de cualquier tipo de base de datos y aplicación.

II. **Propósito Específico:** cuando el rendimiento es fundamental, se puede diseñar y construir un software de propósito especial para una aplicación específica, y este sistema no sirve para otras aplicaciones. Muchos sistemas de reservas de líneas aéreas son de propósito especial y pertenecen a la categoría de **sistemas de procesamiento de transacciones en línea**, que deben atender un gran número de transacciones concurrentes sin imponer excesivos retrasos.



## Recomendación

En este [enlace](#) encontrarás algunos de los SGBD disponibles en el mercado.

## 2. Conceptos y Funciones

Los SGBD son paquetes de software complejos que deben proporcionar una serie de servicios que permiten almacenar y explotar los datos de forma eficiente.

Los componentes y funciones principales son:

1. **Lenguajes de los SGBD:** todos los SGBD ofrecen lenguajes apropiados para realizar la comunicación entre los SGBD y los usuarios. Estos lenguajes se clasifican en:
  - **DDL (Lenguaje de definición de datos):** se utiliza para definir el esquema de la base de datos: los objetos de la base de datos, las estructuras de almacenamiento y las vistas de los distintos usuarios. Lo emplean los diseñadores de la base de datos y los administradores.
  - **DML (Lenguaje de manipulación de datos):** se utiliza para consultar y actualizar los datos de la base de datos. Los usuarios autorizados de un SGBD deben poder consultar, insertar, modificar o borrar datos en una base de datos.

La mayoría de los SGBD incorporan **lenguajes de cuarta generación** que permiten al usuario desarrollar aplicaciones con BD de forma fácil y rápida. Se denominan también *herramientas de*

desarrollo.

2. **El diccionario de datos:** es una descripción de la base de datos con todos los objetos que la forman: las definiciones de los datos y sus relaciones. Se dice también que contiene "metadatos" porque es información sobre los datos.

El diccionario proporciona información acerca de:

- La estructura lógica y física de la base de datos: nombre, descripción, contenido, organización y dónde se almacenan los datos del sistema.
- La definición de tablas, vistas, índices, funciones, procedimientos, etc.
- Los valores que toman las columnas de las tablas por defecto.
- Información que permite garantizar la integridad de los datos almacenados.
- Los privilegios y control de acceso de los usuarios.
- Normas que garanticen la seguridad de los datos.
- Estadísticas y auditorías de los accesos a los objetos, etc.



3. **Seguridad, redundancia e integridad de los datos:** son una serie de mecanismos que proporciona el SGBD para garantizar un acceso correcto, seguro y eficiente a los datos. Se hace mediante un componente software que se encarga de:
  - Garantizar que el acceso a los datos se permita solo a los usuarios autorizados.
  - Disponer de herramientas para planificar y realizar copias de seguridad y restauración.
  - Realizar los procedimientos necesarios para recuperar los datos tras un fallo o pérdida temporal.
  - Ofrecer mecanismos para implantar restricciones de integridad que los datos deberán cumplir.
  - Controlar el acceso concurrente de varios usuarios a los datos sin que se pierda la consistencia
  - Actualizar de manera correcta y automática todos los datos allí donde estén duplicados
4. **Los usuarios del SGBD:** existen distintos tipos de usuarios que acceden al sistema:
  - **Programadores:** son los responsables de la creación y ajuste de las aplicaciones que atacan a los datos. Emplean DDL, DML y cualquier lenguaje anfitrión.
  - **Usuarios expertos:** emplean las utilidades de la base de datos y DML para acceder a los datos y realizar sus propios procesos sobre los objetos para los que se les ha concedido permiso.
  - **Usuarios ocasionales:** utilizan programas de aplicación para acceder a las bases de datos, pero que solo pueden utilizar aquellos objetos para los que se les ha dado permiso de acceso.
  - **Diseñadores-Administradores:** los diseñadores planifican y desarrollan las bases de datos. Definen el esquema lógico y físico de la base de datos, optimizando el almacenamiento y generando la documentación de análisis necesaria para los programadores. En la mayoría de los



casos, cuando las bases de datos están creadas los diseñadores tienen la función de administradores. Los administradores de la base de datos gestionan la seguridad (usuarios- permisos), y la integridad de los datos asegurando que las transacciones sean correctas y no se pierdan datos. También se ocupan de crear las copias de seguridad. Tienen el máximo nivel de acceso. Utilizan fundamentalmente DDL.

Un objetivo fundamental de todo SGBD es posibilitar de manera eficiente el acceso simultáneo a la BD por parte de muchos usuarios. La **conurrencia de usuarios** en una BD consiste en el acceso simultáneo a la BD por parte de más de un usuario.

Para tratar correctamente los problemas derivados de la conurrencia de usuarios, los SGBD utilizan fundamentalmente, dos técnicas: las **transacciones** y los **bloqueos**.

5. **Herramientas de la base de datos:** todos los SGBD incluyen una serie de herramientas de administración que permiten a los administradores la gestión de la base de datos:

- Definir el esquema lógico y físico de la base de datos: crear, modificar y manipular
- Controlar la privacidad de los datos: gestión de usuarios y permisos
- Garantizar la independencia física y lógica de los datos

El SGBD transforma los datos que se introducen en las estructuras necesarias para guardarlos. Transforma las solicitudes lógicas en comandos que localizan y recuperan físicamente los datos, liberando así al usuario de esa tarea.

## 5. Arquitecturas de BD

La arquitectura de todo sistema de BD está muy condicionada también por las características del sistema informático sobre el que se ejecuta, y en especial por los siguientes aspectos:

- La conexión en red de diferentes computadoras.
- El procesamiento paralelo de consultas dentro de una misma computadora.
- La distribución de los datos en diferentes computadoras, incluso alejadas entre sí.

Estas innovaciones tecnológicas han permitido el desarrollo, a partir de los iniciales sistemas totalmente centralizados en una sola computadora, de diferentes arquitecturas de sistemas de BD más evolucionadas, y que permiten dar respuesta a una gran variedad de necesidades de los usuarios y de las organizaciones en que estos trabajan, tales como:

- Sistemas cliente-servidor.
- Sistemas distribuidos.



## 1. Arquitectura Cliente/Servidor

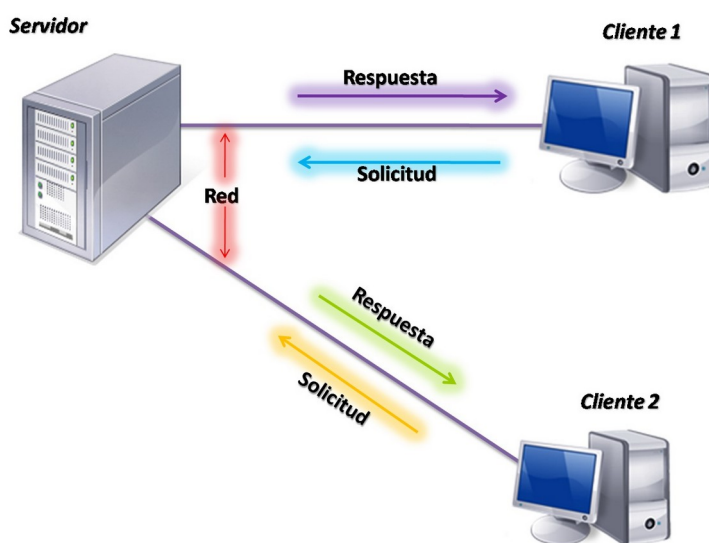
Inicialmente los sistemas de BD eran de tipo estrictamente centralizado, en el sentido que se ejecutaban sobre un único sistema informático, sin necesidad de interactuar con ningún otro. Pero el abaratamiento de los ordenadores personales y el desarrollo de sus capacidades, junto con la implantación de las redes y de Internet, ha hecho evolucionar los sistemas centralizados hacia arquitecturas de tipo **cliente-servidor** o **arquitectura en dos niveles**. Este sistema de bases de datos posee una estructura compuesta de dos partes:

1. El **servidor** permite llevar a cabo las funciones propias del SGBD, se puede decir que el servidor es en sí, el SGBD.
2. Un **cliente** es cada consumidor de recursos de la base de datos

Normalmente el cliente posee una interfaz de programación de aplicaciones (**API**), que es el encargado de enviar las consultas al servidor.

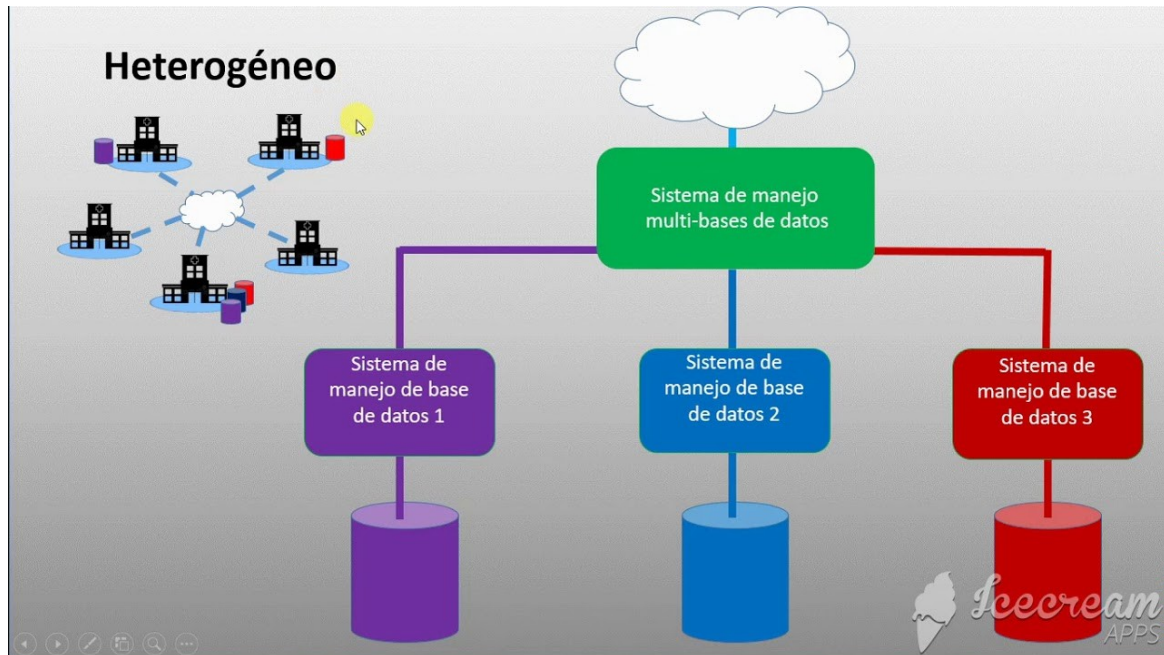
*Ventajas de los sistemas de bases de datos cliente/servidor:*

- La distribución los procesos entre el cliente y el servidor evita la disminución de velocidad de ejecución que ocurre en los procesos de datos centralizados, aunque puede generar cuellos de botella por la velocidad del tráfico de red.
- Empleo de ordenadores estándar para el servidor y los clientes, en lugar de grandes equipos.
- Son sistemas escalables y modulares, es decir, se puede aumentar la cantidad de servicios prestados por los ordenadores y reemplazarlos por equipos nuevos sin que sea afectado por el cambio de SGBD.
- Disponibilidad de herramientas de desarrollo de calidad: lenguajes de 4ª generación orientados a objetos y a procedimientos y entornos gráficos de desarrollo, herramientas de modelado de datos, etc.



## 2. BD Distribuidas

Cuando una aplicación cliente accede a datos de distintos servidores, se denomina **Bases de Datos Distribuidas**. En estos sistemas la base de datos está distribuida en más de un servidor. Los usuarios no tienen por qué conocer la ubicación física de los datos con los que trabajan y han de acceder simultáneamente a varios servidores.



Para el usuario (cliente) es indistinto si los orígenes de datos son únicos o múltiples, pero para el administrador es un trabajo extra considerable, ya que es necesario crear un sistema centralizado de administración de servidores y clientes o administrar de forma individualizada cada servidor para poder mantener en sus sitio usuarios, datos y aplicaciones, con controles de seguridad, acceso y concurrencia.

### Ventajas

- El acceso y procesamiento de los datos es más rápido ya que varios nodos comparten carga de trabajo.
- Desde una ubicación puede accederse a información alojada en diferentes lugares.
- Los costes son inferiores a los de las bases centralizadas
- Existe cierta tolerancia a fallos. Mediante la replicación, si un nodo deja de funcionar el sistema completo no deja de funcionar.
- Aunque los nodos están interconectados, tienen independencia local.

### Desventajas

- La probabilidad de violaciones de seguridad es creciente si no se toman las precauciones debidas.
- Mantenimiento y control puede resultar costoso.
- El intercambio de mensajes y el cómputo adicional necesario para conseguir la coordinación entre los distintos nodos constituyen una forma de sobrecarga que no surge en los sistemas centralizados.
- Dado que los datos pueden estar replicados, el control de concurrencia y los mecanismos de recuperación son mucho más complejos que en un sistema centralizado.