

## Unidad 1:

# Sistemas de almacenamiento de Información

# Unidad 1:

## Sistemas de almacenamiento de información

### Objetivos

- Analizar los sistemas lógicos de almacenamiento y sus funciones
- Identificar los tipos de bases de datos según el modelo de datos utilizado.
- Identificar los tipos de bases de datos según la ubicación de la información
- Valorar la utilidad de un sistema gestor de base de datos
- Describir la función de cada elemento de un sistema gestor de bases de datos
- Clasificar los sistemas gestores de bases de datos.

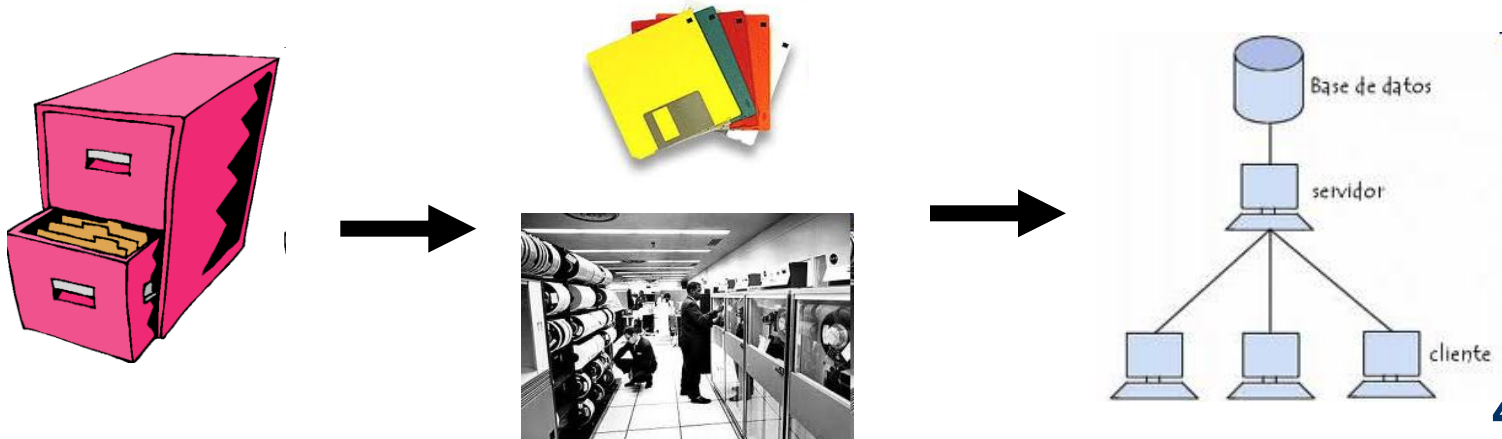
# Unidad 1:

## Almacenamiento de Información

- Introducción
- Ficheros.
- Base de Datos.
- SGBD. Funciones, componentes y tipos
- BD centralizadas y BD distribuidas

# Sistemas lógicos de almacenamiento

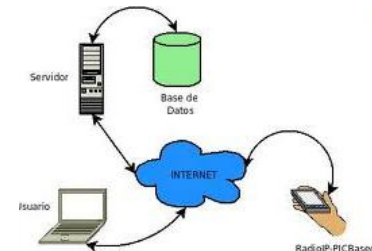
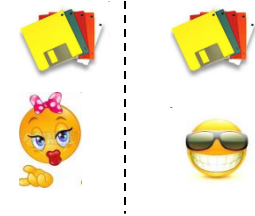
- Necesidad de manejar gran cantidad de datos de una forma fácil, rápida y sencilla
- Imprescindible para la toma de decisiones
- Necesidad de gestionar los datos



# Sistemas lógicos de almacenamiento

## Evolución de los sistemas de almacenamiento

- Fichero manual
- Sistema de Ficheros
- Sistema de BD



# Sistemas lógicos de almacenamiento

## Evolución de los sistemas de almacenamiento

- **Fichero manual:** conjunto de carpetas etiquetadas cuyo contenido estaba relacionado y se guardaban en un armario o archivador.
- **Sistema de Ficheros:** Los datos necesarios se guardaban en ficheros en el ordenador y las aplicaciones accedían a ellos para obtener los informes que se solicitaban de cara a la toma de decisiones en la empresa
- **Sistema de BD :** Una base de datos es un conjunto de datos relacionados entre sí, organizados y estructurados, con información referente a algo. Cualquier cambio en la estructura de los datos no afectará a los programas de aplicación que los utilicen. La información es tratado utilizando los SGBD

# Ficheros / Archivos

# Ficheros / Archivos

## Conceptos

- **¿Qué son?** :Conjunto de información relacionada con respecto a un mismo aspecto y que se guarda en dispositivos de almacenamiento masivo (memoria secundaria: discos duros, lápiz usb,....)

Ej. Ficheros de nuestro disco duro.





# Ficheros / Archivos

Conceptos: DATOS , CAMPO, REGISTRO, FICHERO

FICHERO DE CLIENTES

DNI	NOMBRE	APELLIDOS	DIRECCION	TELEFONO
73564765M	Javier	Barquín Arce	C/ Alta, 234	918342156
56558765W	Luis	Gómez de Miguel	Avda. de Castilla, 2A	956235567
13874521M	María Belén	Márquez Ruiz	C/ <u>Floranes</u> , 2	568732212
75675317R	Carmen	Rodríguez Mata	Paseo Pereda, 123	942665544

# Ficheros / Archivos

## Conceptos

**Datos:** información que necesitamos almacenar

*Ej: los datos personales de los clientes de una empresa, sin tratar.*

**Campo:** Es un carácter o conjunto de caracteres que tiene significado específico. Se utiliza para definir y guardar datos. Es la mínima unidad de información creada con sentido en sí misma.

*Ej: Definiríamos un campo para guardar el nombre, en otro campo para el teléfono, etc.*

# Ficheros / Archivos

## Conceptos

**Registro:** (*Registro Lógico*) Es un conjunto de campos lógicamente relacionados que describen una persona, lugar o cosa. Es también la unidad de tratamiento de los ficheros de datos.

*Ej: Un registro lo formarían todos los datos/campos relativos a un cliente: su nombre, sus apellidos, su teléfono y su dirección.*

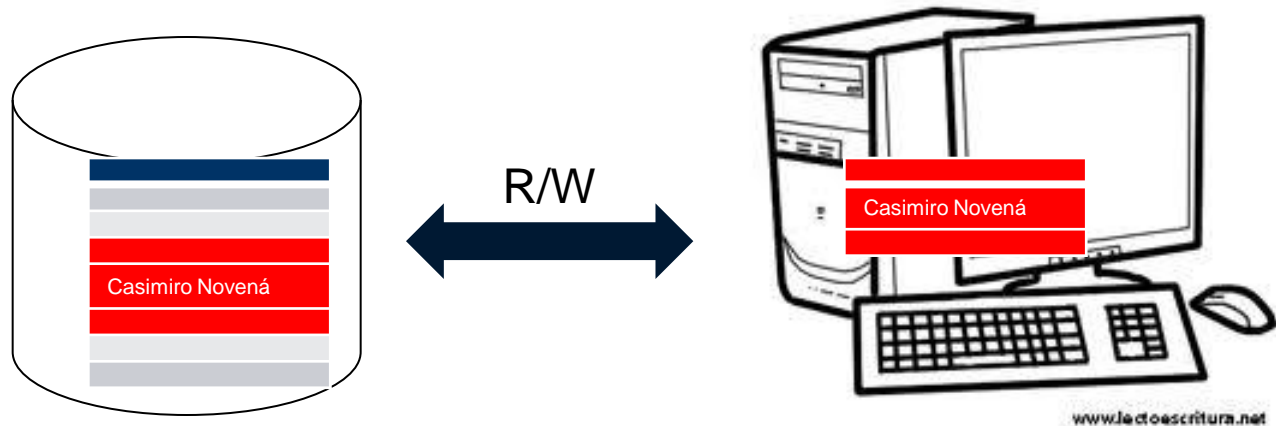
**Fichero:** Es un conjunto de registros relacionados.

*Ej: El fichero estaría formado por los datos de todos los clientes del taller.*

# Ficheros / Archivos

## Conceptos

**Registro Físico o bloque:** La cantidad de información que es transferida entre el soporte en el que se almacena el fichero, y la memoria principal del ordenador, en una sola operación de lectura/grabación; ya que es imposible tener todo el fichero en la memoria por su tamaño para procesarlo

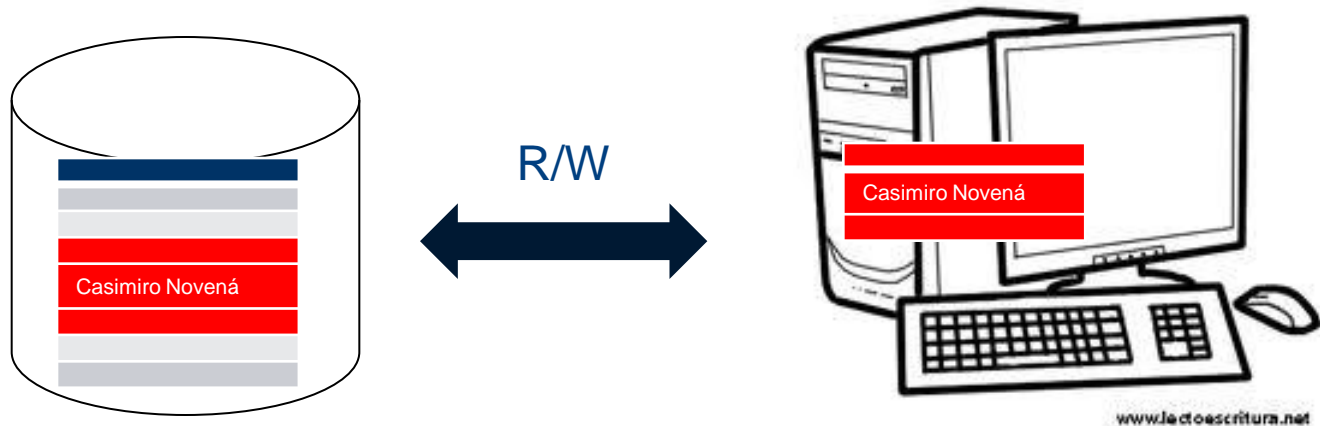


# Ficheros / Archivos

## Conceptos

**Factor de Blocaje o bloque de registros:** número de registros que entran en un bloque físico y que se transfieren en una operación de R/W

Ej. Factor de bloqueo 3



# Ficheros:

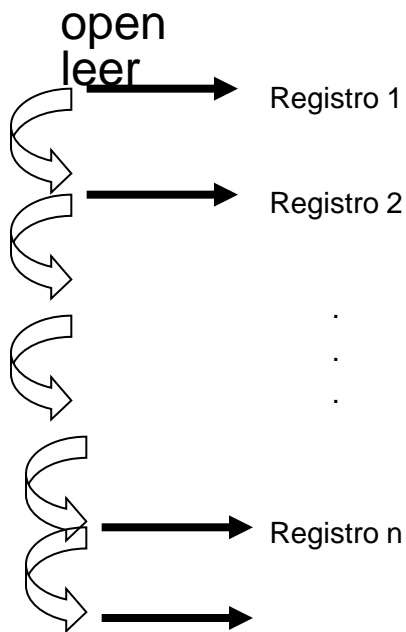
## Operaciones a realizar sobre ficheros

Las operaciones a realizar sobre un archivo o fichero:

- **Abrir (open).** Prepara el fichero para su proceso.
- **Cerrar (close).** Cierra el fichero impidiendo su proceso inmediato.
- **Leer (read).** Obtiene información del fichero.
- **Escribir (write).** Graba información en el fichero.
- **Posicionarse (seek).** Coloca el puntero de lectura en una posición concreta del mismo (no se puede realizar en todos los tipos de ficheros).
- **Fin de fichero (eof).** Indica si hemos llegado al final del fichero.

# Ficheros:

## Operaciones a realizar sobre ficheros

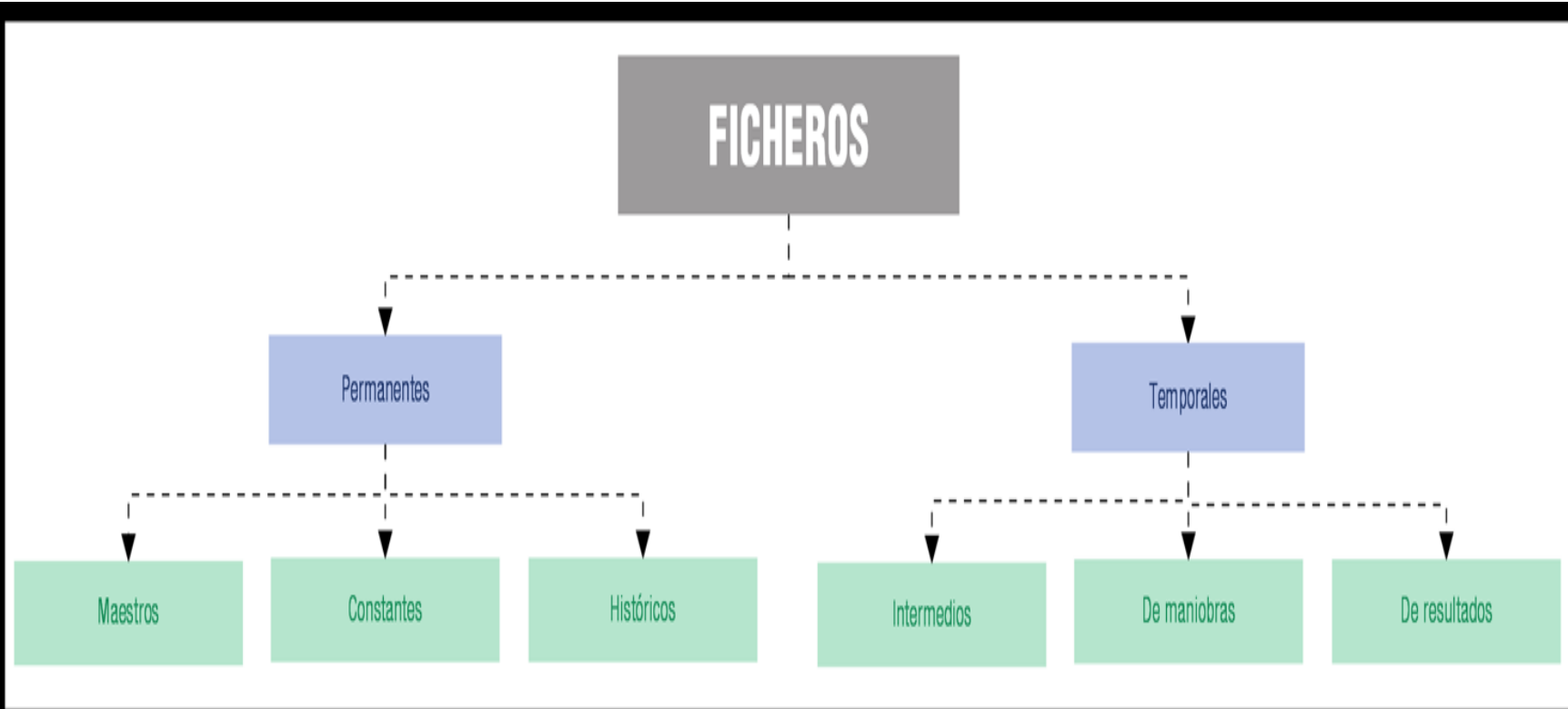


Campo1	Campo2	.....	Campo n
EOF ( End Of File)			

# Ficheros:

## Tipos de Ficheros

Según la función a desempeñar:





# Ficheros:

## Tipos de Ficheros

Según la función a desempeñar:

- **Permanente:** contienen información relevante para una aplicación
  - **Ficheros maestros:** contienen el estado actual de los datos que pueden modificarse desde la aplicación. Es la parte central de la aplicación, su núcleo. Podría ser un archivo con los datos de los usuarios de una plataforma educativa.
  - **Ficheros constantes:** son aquellos que incluyen datos fijos para la aplicación. No suelen ser modificados y se accede a ellos para realización de consultas. Podría ser un archivo con códigos postales.
  - **Ficheros históricos:** contienen datos que fueron considerados como actuales en un periodo o situación anterior. Se utilizan para la reconstrucción de situaciones. Podría ser un archivo con los usuarios que han sido dados de baja en la plataforma educativa.
- Temporales:

# Ficheros:

## Tipos de Ficheros

Según la función a desempeñar:

- Permanente
- **Temporales:** Se utilizan para almacenar información útil para una parte de la aplicación, no para toda ella. Son generados a partir de datos de ficheros permanentes. Tienen un corto periodo de existencia. Estos se subdividen en:
  - **Ficheros intermedios:** almacenan resultados de una aplicación que serán utilizados por otra.
  - **Ficheros de maniobras:** almacenan datos de una aplicación que no pueden ser mantenidos en memoria principal por falta de espacio.
  - **Ficheros de resultados:** almacenan datos que van a ser transferidos a un dispositivo de salida.

# Ficheros

## Clasificación de los dispositivos físicos

Los dispositivos físicos/soportes de información, según la forma de acceder a la información:

- **Dispositivos secuenciales**: la información se guarda en posiciones consecutivas, de forma que para acceder a un dato hay que recorrer los datos anteriores.

Ej. la cinta magnética



- **Dispositivos direccionables**: permiten el acceso directo a los datos.

ej. disco duro.



# Ficheros

## Clasificación según acceso al registro

**Acceso a un registro de un fichero:** procedimiento que se utiliza para seleccionarlo y está ligado al tipo de soporte donde esta la información

### Forma de acceso a los registros :

- Secuencial
- Acceso directo o aleatorio
- Indexados
- Dinámico

# Ficheros

## Clasificación según acceso al registro

Forma de acceso a los registros :

TIPO ACCESO	COMO SE HACE	TIPO DISPOSITIVO
<b><i>Secuencial</i></b>	los registros se leen uno detrás de otro desde el principio del fichero hasta localizar el registro buscado o hasta el final del fichero.	Secuenciales Direccionables
<b><i>acceso directo o aleatorio</i></b>	se accede a un registro sin tener que leer los anteriores, accediendo directamente a él mediante <u>su clave</u> .	Direccionables

# Ficheros

## Clasificación según acceso al registro

Forma de acceso a los registros :

TIPO ACCESO	COMO SE HACE	TIPO DISPOSITIVO
<b><i>Indexados</i></b>	consultamos previamente de forma secuencial en una tabla que <b>contiene la clave más alta y la dirección de comienzo de cada bloque de registros</b> . Una vez localizado se utiliza el acceso directo a ese bloque de registros y, dentro del bloque, la lectura secuencial hasta localizarle.	Direccionables
<b><i>Dinámico</i></b>	permite el acceso directo o por índice a un registro y a partir de ese se accede a los demás de forma secuencial.	Direccionables

# Ficheros Organización.....

- **Secuencial**
  - o Secuencial
  - o Secuencial encadenada
  - o Secuencial indexada
  - o Secuencial indexada encadenada
- **Relativa**
  - o Directa
  - o Aleatoria

# Ficheros

## 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 y grabados

Dirección de memoria	Marca de borrado	Nombre	Apellidos	Teléfono
1200		Alfredo	Bárcena	768334472
1300	*	Isabel	De los Ríos	987335612
1400		Carmen	Sierra	955347612
1500		Fernando	Ruiz	674992455
1600		Juan Carlos	Abad	573982277



# Ficheros

## Organización Secuencial

### Ventajas:

- Rapidez en el acceso a un bloque de registros contiguos
- No se desperdicia espacio en el dispositivo de almacenamiento porque no hay huecos
- No es necesario realizar operaciones de compactación del archivo para corregir los huecos entre registros, al no existir los mismos
- Se pueden utilizar cualquier tipo de registros: de longitud fija, variable o indefinida.

# Ficheros

## Organización Secuencial

Operaciones a realizar en los ficheros con organización secuencial

- **Acceder** a un registro:.....
- **Consultar** un registro:.....
- **Añadir** un registro:.....
- **Eliminar** un registro:.....
- **Compactar** el fichero:.....

# Ficheros

## Organización Secuencial

### Desventajas:

- Para **acceder** al registro  $n$  hay que recorrer los  $n-1$  registros anteriores. El acceso es secuencial.
- 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. El proceso puede ser lento.
- 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.

# Ficheros Organización.....

- **Secuencial**
  - o Secuencial
  - o Secuencial encadenada
  - o Secuencial indexada
  - o Secuencial indexada encadenada
- **Relativa**
  - o Directa
  - o Aleatoria

# Ficheros

## Organización Secuencial Encadenada

- En una variante de la organización secuencial
- Se gestionan mediante punteros (datos especiales que contienen la dirección de cada registro del fichero)
- Cada registro posee un puntero que indica la dirección del siguiente registro



- El puntero permite recorrer los datos en un orden concreto.

# Ficheros

## Organización Secuencial Encadenada

**fichero secuencial encadenado**

fichero secuencial encadenado				
puntero inicial				
7				
nº registro	clave	nombre	apellidos	sig.
1	123	Juan	Torres	8
2	34	Adela	Poyet	3
3	75	Martín	Romero	4
4	90	Abril	Sierra	9
5	108	Maite	Cascón	1
6	315	Rosa	Bueno	nulo
7	20	Pierre	Llontop	2
8	260	Mikel	Artetxe	6
9	102	Luis	Serrano	5

➔ SOPORTE DIRECCIONABLE

# Ficheros

## Organización Secuencial Encadenada

Operaciones:

- ***Consulta:.....***
- ***Adición / Inserción:.....***
- ***Eliminación:.....***
- ***Modificar:.....***

# Ficheros

## Organización Secuencial Encadenada

### Operaciones:

- **consulta** es secuencial, cada vez que se consulta un registro en él se lee la posición del registro siguiente en secuencia lógica.
- **adición** de registros se realiza al final, pues los registros se almacenan secuencialmente. Cuando se quiere **insertar** un registro en una dirección intermedia, el registro físicamente se añade al final, pero se modifican los punteros para mantener la secuencia lógica.
- **eliminación** de registros se efectúa modificando el puntero en el registro anterior, para que apunte al registro siguiente al que queremos borrar, de modo que el fichero mantiene su tamaño, ya que el borrado que se produce es borrado lógico, pero no físico.
- **modificar** un registro se reescribe sobre la información anterior



# Ficheros Organización.....

- **Secuencial**
  - Secuencial
  - Secuencial encadenada
  - **Secuencial indexada**
  - Secuencial indexada encadenada
- **Relativa**
  - Directa
  - Aleatoria

# Ficheros

## Organización Secuencial Indexada

Se utilizan 3 ficheros para almacenar los datos:

- **Fichero de datos con los registros** almacenados de forma secuencial y ordenado por un campo clave que forma parte del registro; pero que permite su acceso aleatorio.
- **Fichero de índices:** una tabla con punteros ordenados a los registros en función al campo clave.  
Para ello el fichero de datos se divide en bloques de registros (bloques de 10 registros o 15 registros o.....); cada registro de esta tabla de índices, contiene de forma ordenada :
  - El valor de la última clave de cada uno de los bloques
  - El puntero o posición donde se encuentra el primer registro de dicho bloque

# Ficheros

## Organización Secuencial Indexada

Se utilizan 3 ficheros para almacenar los datos:

- **Fichero de overflow o desbordamiento:** se guardan los nuevos registros que se van añadiendo (para no tener que ordenar el archivo principal cada vez que se añade un nuevo registro) este archivo está desordenado.

Esta zona sólo se utiliza si se busca un registro y no se encuentra en el archivo principal de datos. En ese caso se recorre todo el archivo de overflow hasta encontrarlo de forma secuencial.

# Ficheros

## Organización Secuencial Indexada

**fichero secuencial indexado**

**área de índices**

clave último elemento	primer índice
75	1
108	4
315	7

**área primaria**

nº registro	clave	nombre	apellidos
1	20	Pierre	Llontop
2	34	Adela	Poyet
3	75	Martín	Romero
4	90	Abril	Sierra
5	102	Luis	Serrano
6	108	Maite	Cascón
7	123	Juan	Torres
8	260	Mikel	Artetxe
9	315	Rosa	Bueno

**área desbordamiento**

clave	nombre	apellidos
190	Tomás	Heintz
314	Luis	Bayón

Bloque

Bloque

Bloque

# Ficheros

## Organización Secuencial Indexada

Operaciones sobre ficheros secuenciales indexados

*El acceso a los registros :...*

*El borrado :....*

*La inserción:...*

# Ficheros

## Organización Secuencial Indexada

### Operaciones sobre ficheros secuenciales indexados

#### 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.

# Ficheros

## Organización Secuencial Indexada

### Operaciones sobre ficheros secuenciales indexados

#### 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*** :

- se hace en el área de overflow.

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***.

# Ficheros

## Organización Secuencial Indexada

### Ventajas:

- El archivo está siempre ordenado en base a una clave
- La búsqueda de datos es rapidísima
- Permite la lectura secuencial (que además será en el orden de la clave)



# Ficheros

## Organización Secuencial Indexada

### Desventajas:

- Necesidad de reorganizar el archivo principal cada cierto tiempo: hay que reescribir de nuevo y de forma ordenada todo el archivo.
- La inserción de registros requiere más tiempo que en los casos anteriores al tener que reordenar los índices

# Ficheros Organización.....

- **Secuencial**
  - o Secuencial
  - o Secuencial encadenada
  - o Secuencial indexada
  - o Secuencial indexada encadenada
- **Relativa**
  - o Directa
  - o Aleatoria

# Ficheros

## Organización Indexada Encadenada

Utiliza **punteros** e **índices** de forma simultánea, lo que implica un aumento del espacio ocupado, pero proporciona una gran rapidez en la búsqueda de registros.

Está formado por:

- un fichero de índices equivalente al comentado en el caso anterior
- otro fichero con la información, pero que no esta ordenado físicamente como en el caso anterior por el campo clave, sino que la ordenación de la clave la establecerá el campo puntero, indicando cual es el siguiente registro a la clave actual
- Una zona de overflow para las inserciones nuevas

# Ficheros

## Organización Indexada Encadenada

**fichero secuencial indexo-encadenado**

**área de índices**

clave último elemento	primer índice
75	7
108	4
315	1

**área primaria**

nº registro	área primaria			
	clave	nombre	apellidos	sig.
1	123	Juan	Torres	8
2	34	Adela	Poyet	3
3	75	Martín	Romero	4
4	90	Abril	Sierra	9
5	108	Maite	Cascón	1
6	315	Rosa	Bueno	nulo
7	20	Pierre	Llontop	2
8	260	Mikel	Artetxe	6
9	102	Luis	Serrano	5

**área desbordamiento**

clave	nombre	apellidos
190	Tomás	Heintz
314	Luis	Bayón

# Ficheros

## Organización Indexada Encadenada

### Operaciones

Para **eliminar** registros se marcan, en lugar de ser borrados físicamente.

Las **adiciones** se realizan sobre la zona de overflow ya que no se pueden añadir registros en el área primaria una vez creado el fichero.

Para **acceder a un registro** se busca en el área de índices la dirección de inicio del bloque de registros a la que pertenece el registro buscado. Si no se encuentra en el área de índices, el último registro del bloque apuntará a un bloque de área de overflow, que se lee de forma secuencial. Si no se encuentra en el área de overflow, se terminará la búsqueda al acabar el área de overflow.

# Ficheros

## Organización Indexada Encadenada

### Operaciones

Si se desea **consultar** todo el fichero, el último registro de cada bloque del área de overflow tiene un puntero al primer registro del área siguiente y así se continuará la búsqueda hasta acabar de leer el fichero.

Los nuevos registros se **insertan** y quedan enlazados entre sí mediante punteros conservando el orden lógico que marca la clave o índice principal.

# Ficheros

## Organización Indexada Encadenada

### Ventajas

- Posee las mismas ventajas que los archivos secuenciales indexados
- mayor rapidez al reorganizar el fichero (sólo se modifican los punteros)

### Desventajas

- Requieren compactar los datos a menudo para reorganizar índices y eliminar el fichero de desbordamiento.

# Ficheros Organización.....

- **Secuencial**
  - o Secuencial
  - o Secuencial encadenada
  - o Secuencial indexada
  - o Secuencial indexada encadenada
- **Relativa**
  - o Directa
  - o Aleatoria



# Ficheros

## Organización Relativa

### Organización Relativa

- Se basan en la independencia entre el orden en el que se dan de alta los registros y la posición en que se graban en el soporte
- el almacenamiento físico de los registros se realiza mediante el empleo de una **CLAVE** que relaciona la posición del registro dentro del fichero y la posición de memoria donde está almacenado.

# Ficheros

## Organización Relativa

### Organización Relativa. Tipos

- **Relativa Directa:** se emplean claves numéricas, las posiciones físicas de almacenamiento coinciden con el valor de la clave.
- **Relativa Aleatoria:** emplean clave alfanumérica o bien numérica pero que se debe transformar, mediante algún tipo de algoritmo, obteniéndose un valor numérico entero que facilite la correspondencia directa entre la clave y la dirección de memoria.

# Ficheros Organización.....

- **Secuencial**
  - o Secuencial
  - o Secuencial encadenada
  - o Secuencial indexada
  - o Secuencial indexada encadenada
- **Relativa**
  - o Directa
  - o Aleatoria

# Ficheros

## Organización Relativa - Directa

### Organización Relativa Directa

- emplean claves numéricas: *las posiciones físicas de almacenamiento coinciden con el valor de la clave.*

Direcciones de memoria	Clave	Datos
1		
2	2	Cliente A
3		
4	4	Cliente B
5	5	Cliente C
6		
7		
8	8	Cliente D
9	9	Cliente E
10		

Espacios o huecos libres

# Ficheros

## Organización Relativa - Directa

### Ventajas

- Acceso directo a los registros. No se necesita un algoritmo de transformación
- Permite realizar operaciones de escritura y lectura simultáneamente, ya que primero se localiza el registro y luego se realiza la operación deseada: inserción, eliminación, consulta, modificación, etc.
- El acceso a los datos se realiza de dos formas diferentes:
  - Directamente, mediante la clave del registro
  - Secuencialmente, a partir del primer registro almacenado en el fichero, por lo que son muy rápidos en el tratamiento individual de registros.

# Ficheros

## Organización Relativa - Directa

### Desventajas

- Al realizar un acceso secuencial, en una consulta sobre todos los registros del fichero hay que recorrer todas las direcciones aunque estén vacías.
- Deja gran cantidad de posiciones libres de memoria dentro del fichero (huecos), debido a que las claves de los registros pueden indicar posiciones de almacenamiento no contiguas, lo que implica una falta de aprovechamiento del soporte del almacenamiento respecto al número real de registros almacenados.

# Ficheros

## Organización Relativa - Directa

### Desventajas

- Se pueden producir colisiones, ya que puede existir más de un registro con la misma clave. Esto causa errores.
- No se pueden almacenar registros cuya dirección de almacenamiento sea mayor que los límites máximos del fichero

# Ficheros Organización.....

- **Secuencial**
  - o Secuencial
  - o Secuencial encadenada
  - o Secuencial indexada
  - o Secuencial indexada encadenada
- **Relativa**
  - o Directa
  - o Aleatoria



# Ficheros

## Organización Relativa - Aleatoria

### Organización Relativa Aleatoria

- clave alfanumérica o bien numérica pero que se debe transformar, aplicando un algoritmo, obteniéndose un valor numérico entero que facilite la correspondencia directa entre la clave y la dirección de memoria.

Direcciones de memoria	Clave	Datos	
1			
2	AB85	Cliente C	
3			
4	DG49	Cliente A	
5	EH23	Cliente D	
6			
7			
8	BS12	Cliente F	
9	KL92	Cliente E	
10			
	FB43	Cliente B	

Espacios o huecos libres

Zona de overflow o sinónimos

# Ficheros

## Organización Relativa - Aleatoria

### Organización Relativa Aleatoria

El algoritmo de transformación o **hashing** debe cumplir las siguientes condiciones:

- Que sea **fácil de aplicar**, estableciendo una relación directa entre dirección lógica y dirección física
- Que deje el **mínimo número de huecos** posible, maximizando el espacio disponible en el dispositivo de almacenamiento.
- Que las claves de registros diferentes nos den direcciones diferentes. Producir el menor número de registros que con distintas claves creen las mismas direcciones de almacenamiento. Cuando a partir de dos o más claves diferentes se obtiene la misma dirección se dice que se producen **sinónimos** y que esos registros producen **colisiones**. En este caso solo uno de ellos puede ser almacenado en esa dirección y habrá que prever algún procedimiento para calcular la posición en que se tiene que grabar el otro registro. **Por ejemplo una zona de overflow**

# Ficheros

## Organización Relativa - Aleatoria

### Ventajas

- Acceso inmediato a los registros mediante su clave
- No es necesario ordenar el fichero
- Se pueden realizar operaciones de escritura y lectura a la vez
- Son muy rápidos en el tratamiento individual de registros.
- Se pueden realizar accesos secuenciales.

# Ficheros

## Organización Relativa - Aleatoria

### Desventajas

- Las consultas sobre todo el fichero son lentas
- El fichero contiene gran cantidad de huecos o espacios libres
- El algoritmo para la conversión de las claves ha de ser creados de modo que dejen el menor número de huecos libres y se genere el menor número de sinónimos.

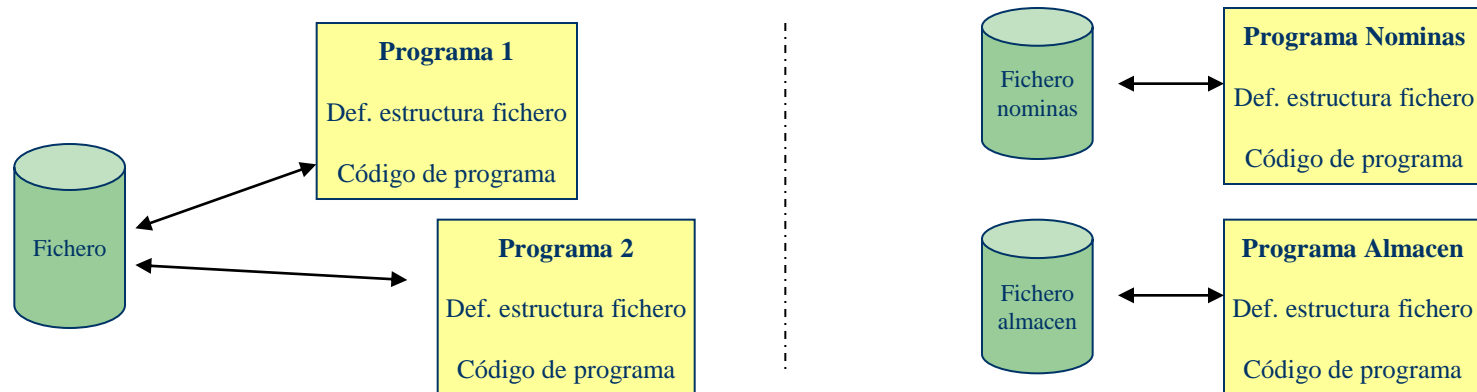
# Ficheros Inconvenientes

## Inconvenientes de los Ficheros

- Respecto a los ficheros
- Respecto a los datos

# Ficheros Inconvenientes

**Enfoque tradicional:** Son sistemas orientados al proceso



Ejemplo:

Nombre alumno	Dirección	Población	Código Postal
------------------	-----------	-----------	---------------

Nuevos campos ➔ Modificar todos los programas

Nombre alumno	Fcha Nto	Dirección	Población	Código Postal
------------------	----------	-----------	-----------	---------------

# Ficheros Inconvenientes

- Se da más importancia al tratamiento que reciben los datos que se almacenan en ficheros diseñados para una determinada aplicación
- Cada programa almacena y utiliza sus propios datos de forma un tanto caótica.

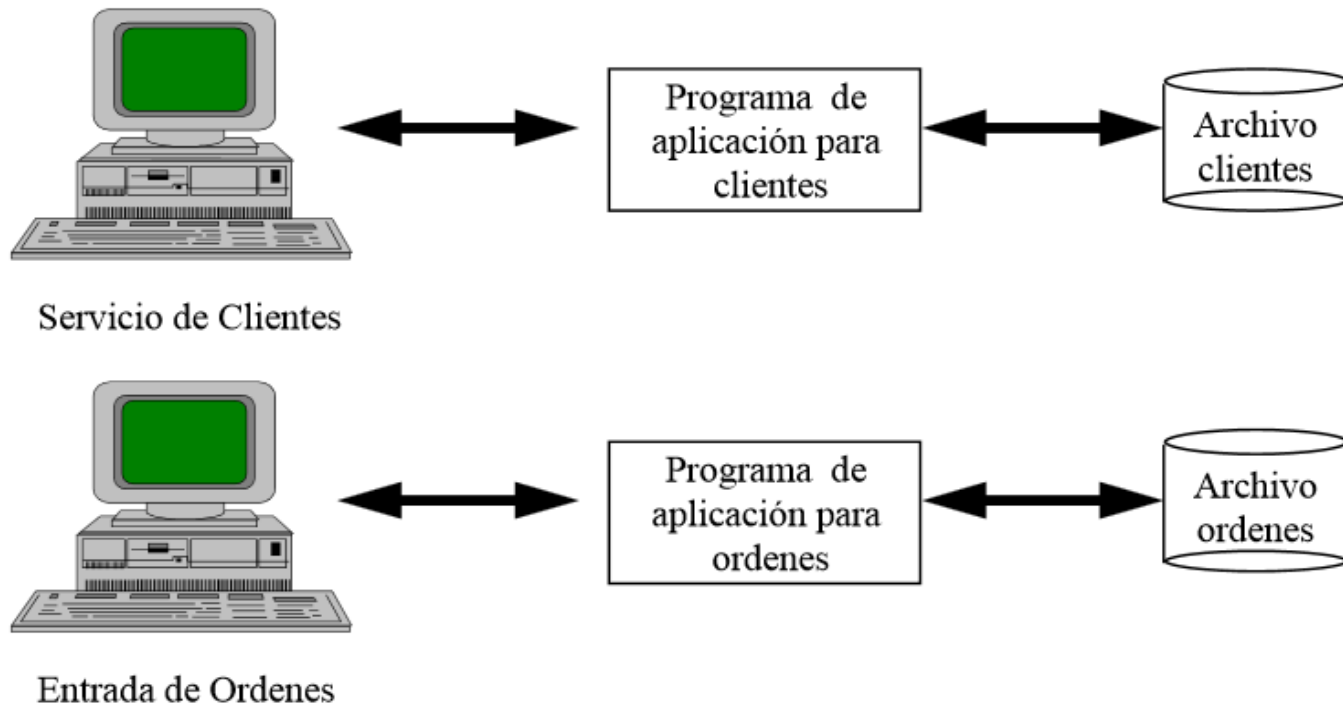
La **ventaja** de este sistema (la única ventaja), es que los procesos son independientes por lo que la modificación de uno no afectaba al resto.

Los inconvenientes pueden verse desde dos puntos de vista:

- Los ficheros
- Los datos

# Ficheros Inconvenientes

## Inconvenientes de los Ficheros





# Ficheros Inconvenientes

## Inconvenientes de los Ficheros

- Respecto a los ficheros
- Respecto a los datos

# Ficheros

## Inconvenientes respecto a FICHEROS

### ***Es necesario controlar:***

- ***Integridad semántica:*** Es un conjunto de restricciones, también llamadas Reglas de Validación, que permiten almacenar, o no, determinados valores de un objeto en los ficheros para evitar que se pierda la consistencia.
  - *Ej.: restricción para que el saldo de una CC no baje de un cifra determinada. Cada fichero puede tener diferentes reglas de validación. Es difícil crear programas que tengan en cuenta a la vez todas estas reglas, con lo que se produce información inconsistente para el sistema.*
- ***Control de autorizaciones:*** Trata de evitar que se produzcan accesos indebidos a los datos, para lo que a cada usuario se le da un identificador y una clave. En este caso al estar los elementos del sistema distribuidos sin organizar a lo largo del sistema, no se puede controlar el acceso a cada elemento del sistema.
  - *Ej.: No todo el personal del banco puede acceder a todos los datos. El departamento de nóminas sólo necesita acceder a los datos de los empleados y no a los de los clientes.*

# Ficheros

## Inconvenientes respecto a FICHEROS

- **Control de concurrencia:** Es el control del acceso simultáneo de varios usuarios a los mismos datos. Cuando varios usuarios acceden a la vez al mismo fichero para modificar información, si no hay un programa que controle el orden de acceso no habrá seguridad acerca de cuál de todas las modificaciones será guardada y en qué orden.

**Ej.:** cuando varios clientes acceden a una misma cuenta para retirar sus fondos. Si se puede acceder a los datos con distintos programas será muy difícil de coordinar.

# Ficheros Inconvenientes

## Inconvenientes de los Ficheros

- Respecto a los ficheros
- Respecto a los datos

# Ficheros Inconvenientes

## Inconvenientes de los Ficheros

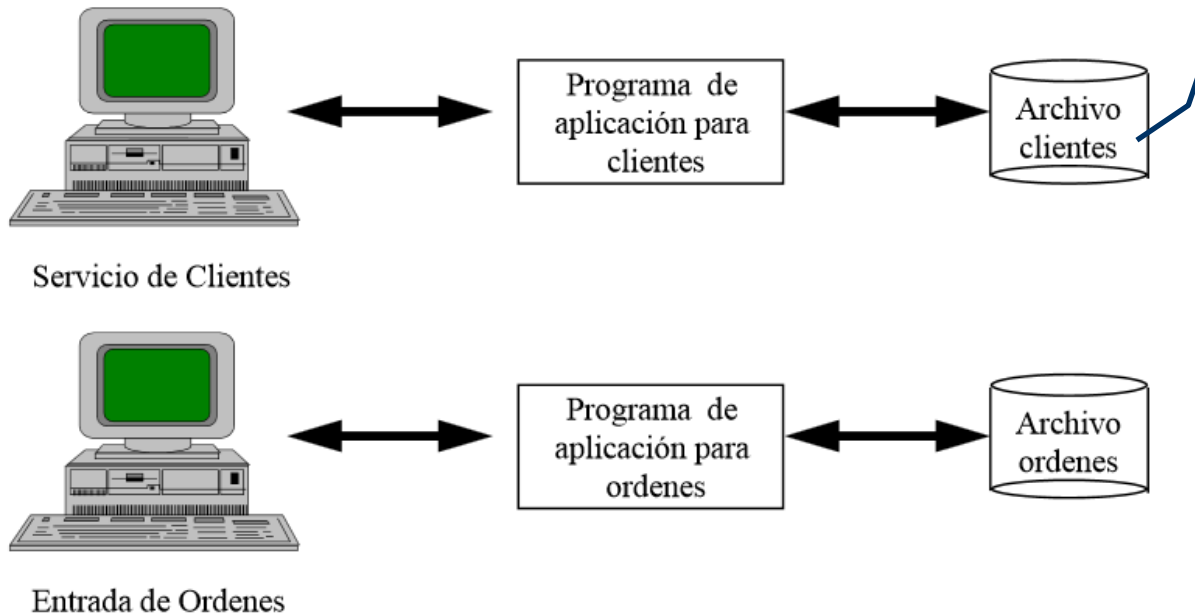
- Respecto a los datos

Hospital	Médico	Especialidad
Hospital Del Norte	Dr. Alonso	Traumatología
Hospital Del Norte	Dr. Alonso	Pediatría
Hospital Del Norte	Dr. González	Traumatología
Hospital Del Norte	Dr. González	Pediatría
Hospital Del Este	Dr. González	Neurología
Hospital Del Este	Dr. Bermúdez	Neurología
Hospital Del Sur	Dr. Alonso	Traumatología
Hospital Del Sur	Dr. Alonso	Pediatría
Hospital Del Sur	Dr. Alonso	Neurología
Hospital Del Sur	Dr. Bermúdez	Traumatología
Hospital Del Sur	Dr. Bermúdez	Pediatría
Hospital Del Sur	Dr. Bermúdez	Neurología

# Ficheros Inconvenientes

## Inconvenientes de los Ficheros

### – Respecto a los datos



Que pasa si un usuario viene al area de cliente y cambia su direcc. con este empleado/a???

# Ficheros

## Inconvenientes respecto a los DATOS

Se deben a su estructura física, a su modo de estar almacenados en diferentes archivos

- **Redundancia:** repetición innecesaria de información en varios ficheros.  
*Ej. Nombres y números de teléfono de los clientes del banco. Podrían aparecer duplicados, tanto en el archivo de clientes de cuentas corrientes como, por ejemplo, en el de cuentas de cheques.*
- **Inconsistencia:** Es información redundante en la que las copias de los datos de los distintos ficheros no concuerdan entre sí.  
*Ej: Cuando el teléfono de un cliente ha cambiado y no hemos reflejado el cambio en todos los ficheros que lo contienen.*

# Ficheros

## Inconvenientes respecto a los DATOS

*Ejemplo de un fichero*

Nss	Nombre	Puesto	Salario	email
111	Juan Perez	Jefe de Area	3000	juanp@ecn.es
111	Juan Perez	Jefe de Area	3000	jefe2@ecn.es
222	Jose Sanchez	Administrativo	1500	jsanchez@ecn.es
333	Ana Diaz	Administrativo	1500	adiaz@ecn.es
333	Ana Diaz	Administrativo	1500	ana32@gmail.com
...	...	...	...	...



# Ficheros

## Inconvenientes respecto a los DATOS

### *Ejemplo de Redundancia*

Nss	Nombre	Puesto	Salario	email
111	Juan Perez	Jefe de Area	3000	juanp@ecn.es
111	Juan Perez	Jefe de Area	3000	jefe2@ecn.es
222	Jose Sanchez	Administrativo	1500	jsanchez@ecn.es
333	Ana Diaz	Administrativo	1500	adiaz@ecn.es
333	Ana Diaz	Administrativo	1500	ana32@gmail.com
...	...	...	...	...

# Ficheros

## Inconvenientes respecto a los DATOS

*Ejemplo para evitar la Redundancia*

Nss	Nombre	Puesto	Salario
111	Juan Perez	Jefe de Area	3000
222	Jose Sanchez	Administrativo	1500
333	Ana Diaz	Administrativo	1500
...	...	...	...

Nss	email
111	juanp@ecn.es
111	jefe2@ecn.es
222	jsanchez@ecn.es
333	adiaz@ecn.es
333	ana32@gmail.com
...	...

*¿Se observa más redundancia en la tabla?*

# Ficheros

## Inconvenientes respecto a los DATOS

*Mas redunRedundancia*

Nss	Nombre	Puesto	Salario
111	Juan Perez	Jefe de Area	3000
222	Jose Sanchez	Administrativo	1500
333	Ana Diaz	Administrativo	1500
...	...	...	...

Nss	email
111	juanp@ecn.es
111	jefe2@ecn.es
222	jsanchez@ecn.es
333	adiaz@ecn.es
333	ana32@gmail.com
...	...

# Ficheros

## Inconvenientes respecto a los DATOS

### *Ejemplo para evitar la Redundancia*

Nss	Nombre	Puesto
111	Juan Perez	Jefe de Area
222	Jose Sanchez	Administrativo
333	Ana Diaz	Administrativo
...	...	...

Nss	email
111	juanp@ecn.es
111	jefe2@ecn.es
222	jsanchez@ecn.es
333	adiaz@ecn.es
333	ana32@gmail.com
...	...

*Hay redundancia ahora:*

**SI**, pero es la mínima y necesaria para poder representar la información de la tabla original

Puesto	Salario
Jefe de Area	3000
Administrativo	1500
...	...

# Ficheros

## Inconvenientes respecto a los DATOS

### *Ejemplo Inconsistencia de datos*

Nss	Nombre	Puesto	Salario
111	Juan Perez	Jefe de Area	3000
222	Jose Sanchez	Administrativo	1500
333	Ana Diaz	Administrativo	1500
...	...	...	...

Nss	email
111	juanp@ecn.es
111	jefe2@ecn.es
222	jsanchez@ecn.es
333	adiaz@ecn.es
333	ana32@gmail.com
<b>444</b>	josell@ecn.es
....	....

- ¿A quien pertenece el nss **444**? → **inconsistencia!!!**

# Ficheros

## Inconvenientes respecto a los DATOS

- **Aislamiento:** fragmentación de la información. Se produce cuando los datos referentes a un objeto se almacenan en distintos ficheros, siendo difícil obtener a la vez toda la información relativa al mismo objeto.

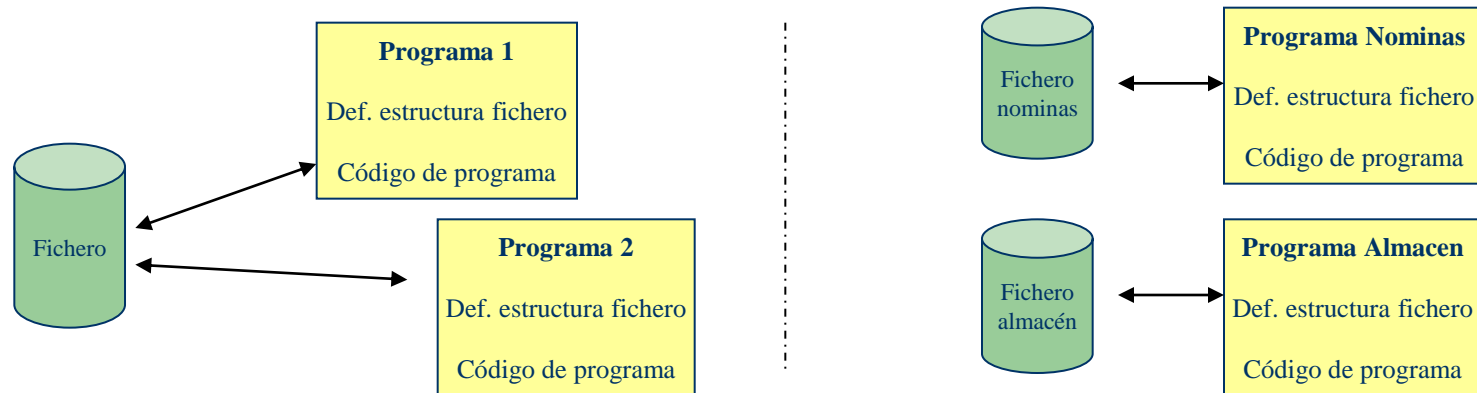
*Ej.: los datos personales de un cliente pueden estar almacenados en un fichero, mientras que se han creado otros ficheros para recoger los datos relativos a sus cuentas bancarias, los préstamos e hipotecas, etc.*

- **Dificultad de acceso a los datos:** Cada vez que se requiera una consulta no prevista inicialmente, hay que modificar el código de las aplicaciones o incluso crear una nueva aplicación.

*Ej: cuando queremos localizar los datos de los clientes que viven en la ciudad correspondiente a un código postal determinado. Tenemos un fichero y una aplicación que extrae los datos de todos los clientes pero no teníamos previsto obtener ese tipo de información. Será necesario escribir un nuevo programa.*

# Ficheros Inconvenientes

**Enfoque tradicional:** Son sistemas orientados al proceso y el principal inconveniente es que cada aplicación va ligada al fichero de datos que usa



Ejemplo:

Nombre alumno	Dirección	Población	Código Postal
------------------	-----------	-----------	---------------

Nuevos campos ➔ Modificar todos los programas

Nombre alumno	Fcha Nto	Dirección	Población	Código Postal
------------------	----------	-----------	-----------	---------------

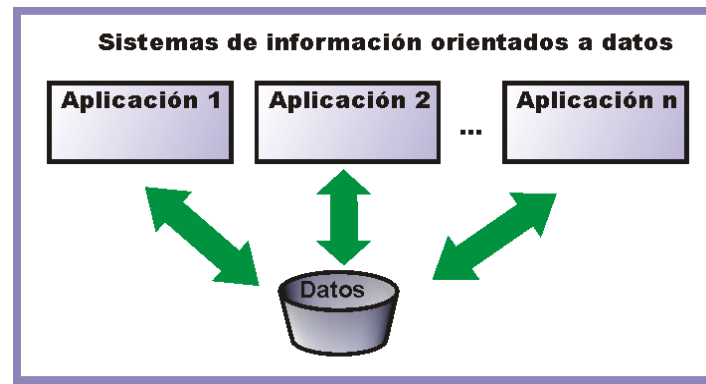
# Bases de Datos



# Base de Datos

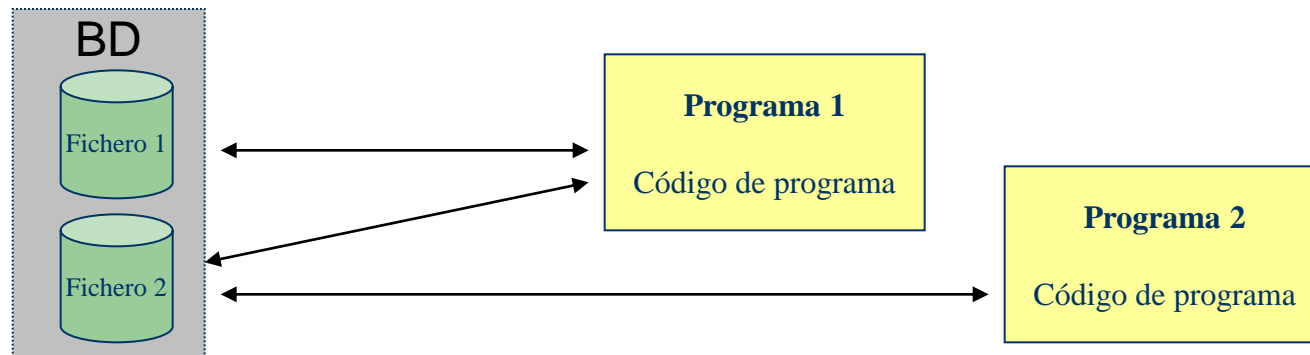
## Enfoque de BD: Sistemas orientados a datos

- los datos se centralizan en un repositorio común a todas las aplicaciones, que llamaremos **base de datos**.
- Este repositorio, en una única estructura lógica que es utilizable por todas las aplicaciones. A través de esa estructura se accede a los datos que son comunes a todas las aplicaciones.
- Cuando una aplicación modifica un dato, la modificación será visible para el resto de aplicaciones.



# Base de Datos

## Enfoque de BD: Sistemas orientados a datos



### Ejemplo Fichero 1:

Nombre alumno	Dirección	Población	Código Postal
------------------	-----------	-----------	---------------

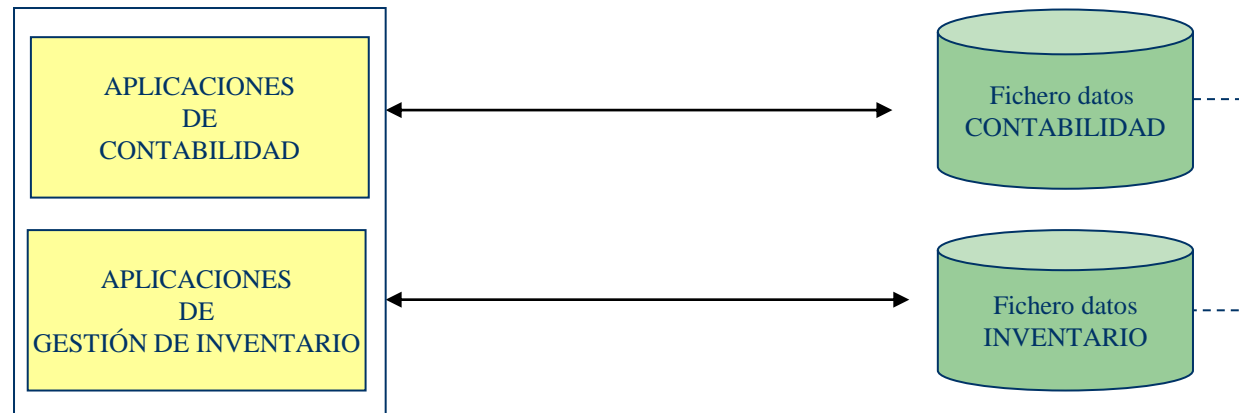
Nuevos campos en Fichero1 → SOLO afecta a los programas que van a trabajar con dichos campos

Nombre alumno	Fcha_Nto	Dirección	Población	Código_Postal
---------------	----------	-----------	-----------	---------------

# Base de Datos

Desde el punto de vista: **Sistemas de Transacciones**

Desarrollo Convencional de aplicaciones en una empresa con **ficheros**.

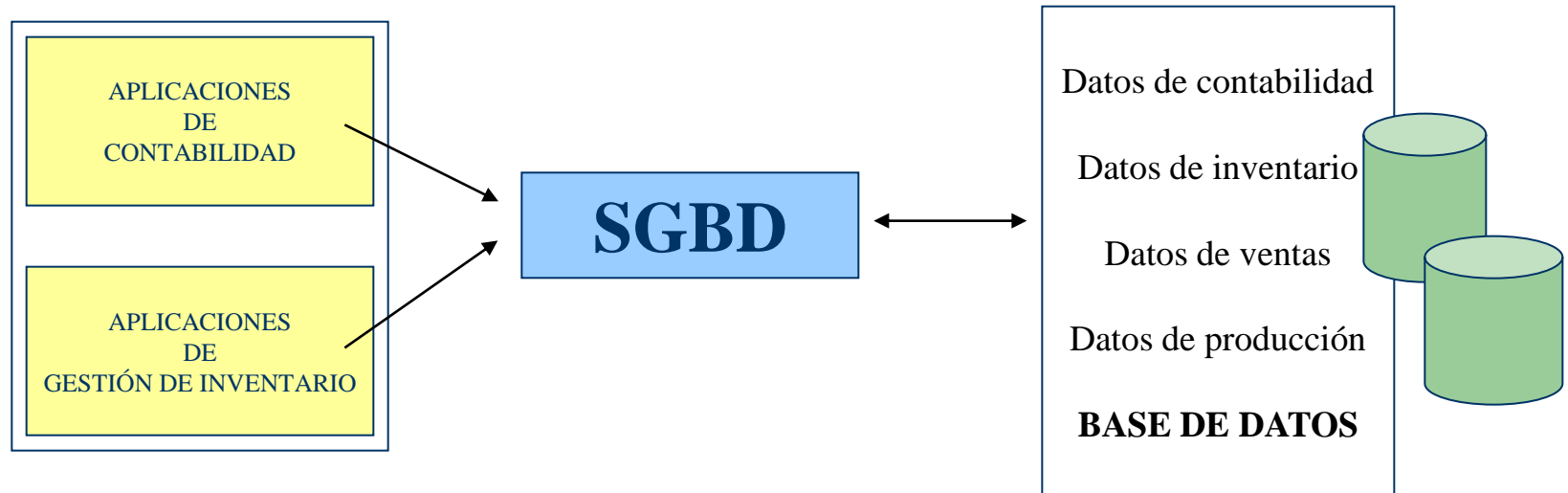


**!!!!!!!REDUNDANCIA e INCONSISTENCIA!!!!!!!!!!!**  
**en algún DATOS**

# Base de Datos

Desde el punto de vista: Sistemas de Transacciones

Desarrollo actual de aplicaciones en una empresa con **BD**:



# Base de Datos. Ventajas

## Ventajas de las BD:

- **Independencia de los datos, los programas y procesos.** Esto permite modificar los datos sin modificar el código de las aplicaciones.
- **Menor redundancia.** No hace falta tanta repetición de datos. Sólo se indica la forma en la que se relacionan los datos.
- **Integridad de los datos.** Mayor dificultad de perder los datos o de realizar incoherencias con ellos.
- **Mayor seguridad en los datos.** Al permitir limitar el acceso a los usuarios. Cada tipo de usuario podrá acceder a unas cosas.

# Base de Datos. Ventajas

## Ventajas de las BD:

- **Datos más documentados.** Gracias a los **metadatos** que permiten describir la información de la base de datos.
- **Acceso a los datos más eficiente.** La organización de los datos produce un resultado más óptimo en rendimiento.
- **Menor espacio de almacenamiento.** Gracias a una mejor estructuración de los datos.
- **Acceso simultáneo a los datos.** Es más fácil controlar el acceso de usuarios de forma concurrente.
- **Lenguaje de consulta:** Utiliza un lenguaje que permite a los usuarios interrogar y explotar la BD sin necesidad de hacer programas a medida

# Base de Datos. Inconvenientes

## Los principales inconvenientes de BD:

- **Instalación costosa.** El control y administración de bases de datos requiere de un software y hardware poderoso
- **Requiere personal cualificado.** Debido a la dificultad de manejo de este tipo de sistemas.
- **Implantación larga y difícil.** Debido a los puntos anteriores. La adaptación del personal es mucho más complicada y lleva bastante tiempo.

# Introducción

## Definición de BASE DE DATOS

Un conjunto de datos interrelacionados y almacenados sin redundancias perjudiciales o innecesarias, los cuales se caracterizan por:

- Servir a una o varias aplicaciones de la mejor manera posible
- Existir independencia entre los datos y los programas que los manejan



# Introducción

## Tipos de datos que almacenan las BD

- **Los datos del usuario:** Datos de las aplicaciones de usuarios  
Valores de los campos: Nombre, apellidos, codigo\_postal,.....
- **Los datos del sistema:** Datos que la base necesita para gestionarse a sí misma  
Datos de usuario: nombre\_usr, contraseña, privilegios, espacio asignado,...  
Datos de archivos: identificación, tamaño, ubicación

## Objetivos de la organización de las BD

Una BD, independientemente de la organización debe cumplir los siguientes objetivos:

- Versatilidad en la información
- Atender las peticiones de datos adecuándolas al uso que se haga de ellos (p.e.: Proceso en lotes frente a petición en tiempo real)
- Redundancia mínima para evitar la inconsistencia y siempre que no suponga complejidad y bajo rendimiento
- Capacidad de acceso rápido
- Integridad de los datos cumpliendo las restricciones definidas
- Seguridad (ante fuego, robo, capacidad de recuperación ante fallos)
- Privacidad de acceso a los datos
- Capacidad de adaptación a nuevas situaciones (físicas y lógicas) y a situaciones transitorias

## Ventajas e Inconvenientes de las BD

### Ventajas:

- Redundancia mínima y controlada
- Actualización única
- Independencia de los datos con respecto a los programas
- Confidencialidad y seguridad controlada

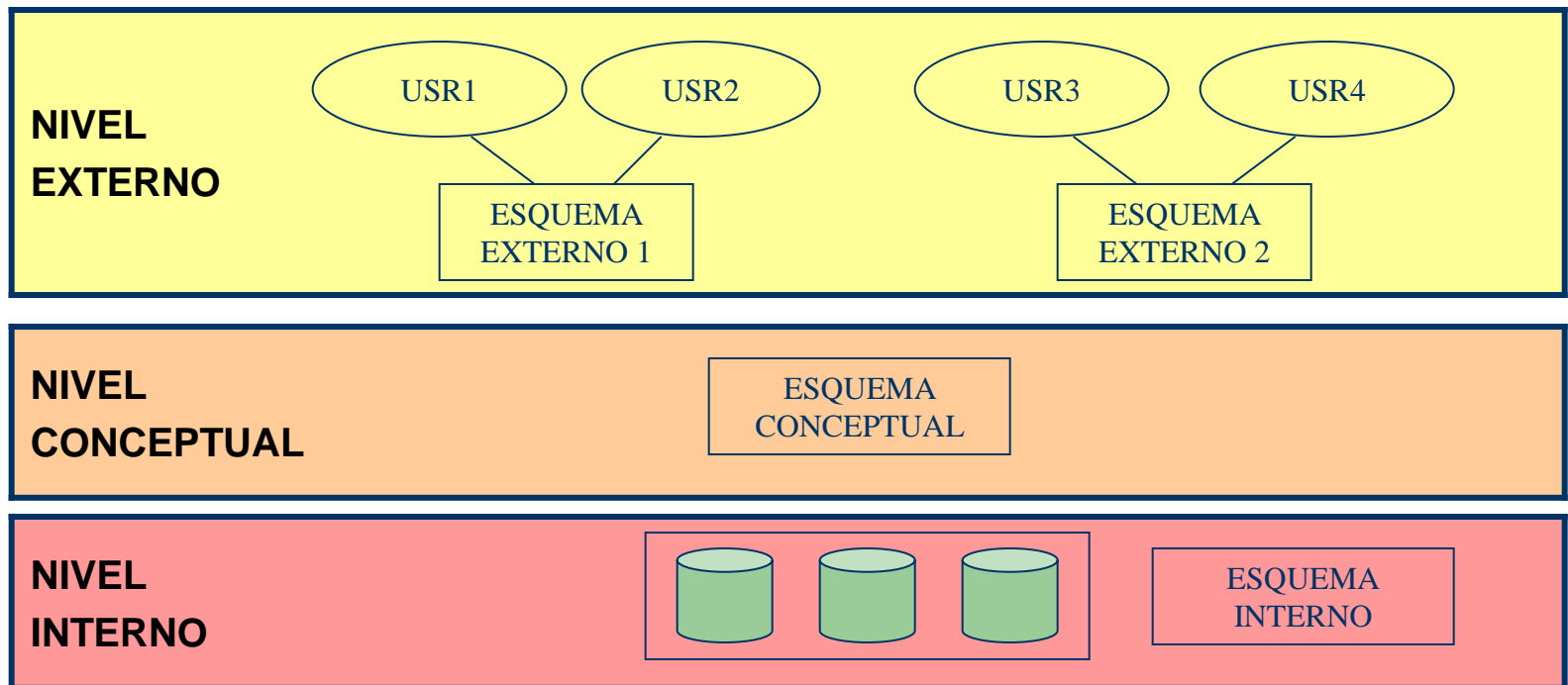
### Inconvenientes:

- Mayor tamaño
- Software más complejo y costoso
- Hardware con alto rendimiento
- Personal cualificado para su diseño y mantenimiento

# Arquitectura de una BD

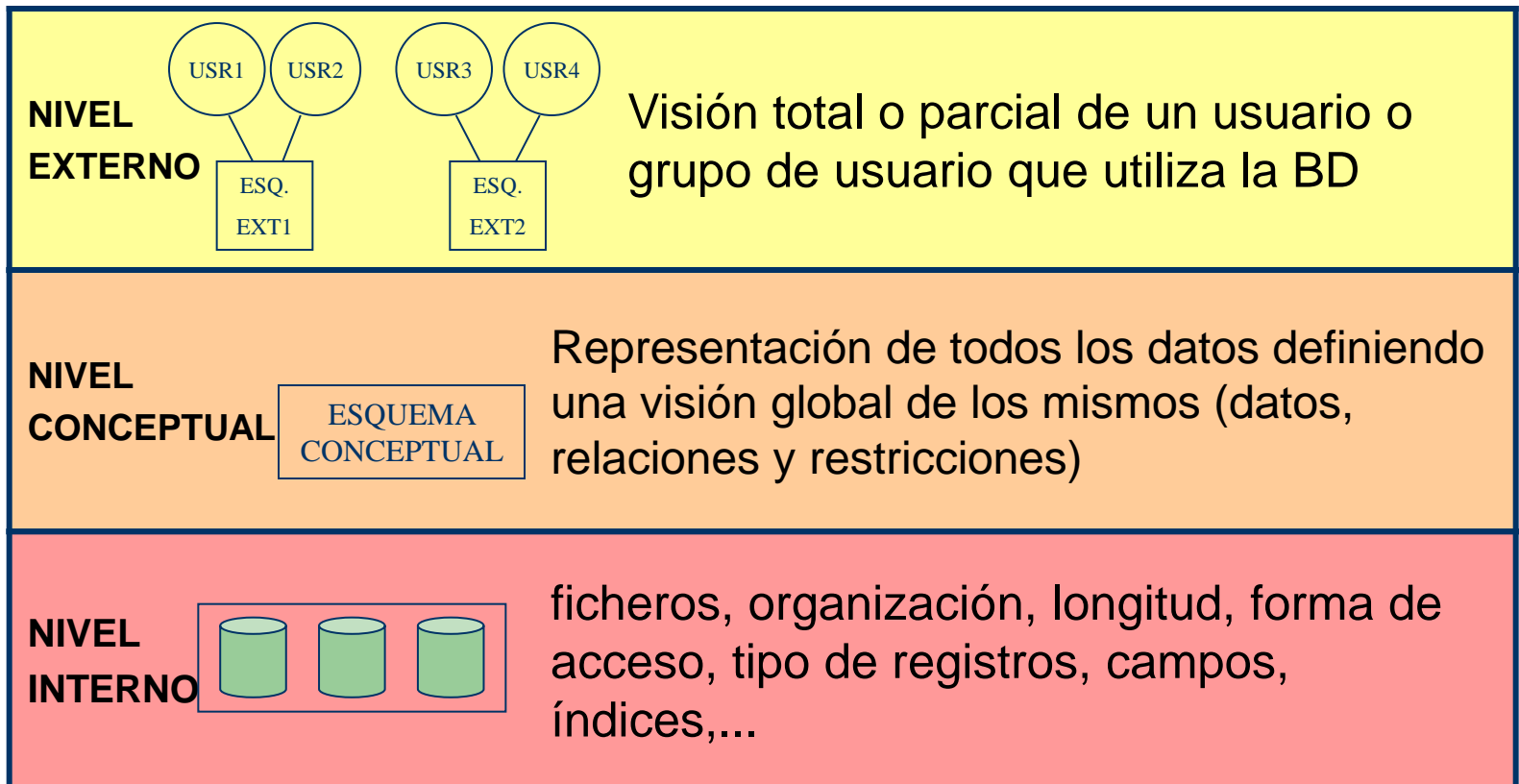
# Arquitectura de las BD

Niveles de abstracción en la descripción de una BD propuesta por ANSI-SPARC: Pretende separar los programas de aplicación de la base de datos física



# Arquitectura de las BD

Niveles de abstracción en la descripción de una BD:



# Arquitectura de las BD

## Nivel Externo:

- Visión total o parcial de cada uno de los usuario o grupo de usuario que utiliza la BD

## Nivel Conceptual:

- Descripción del esquema conceptual: Representación de todos los datos definiendo una visión global de los mismos. (datos, relaciones y restricciones) permite definir las entidades, los atributos y sus propiedades, las relaciones, operaciones de los usuarios y las restricciones y reglas de validación

## Nivel Interno o físico:

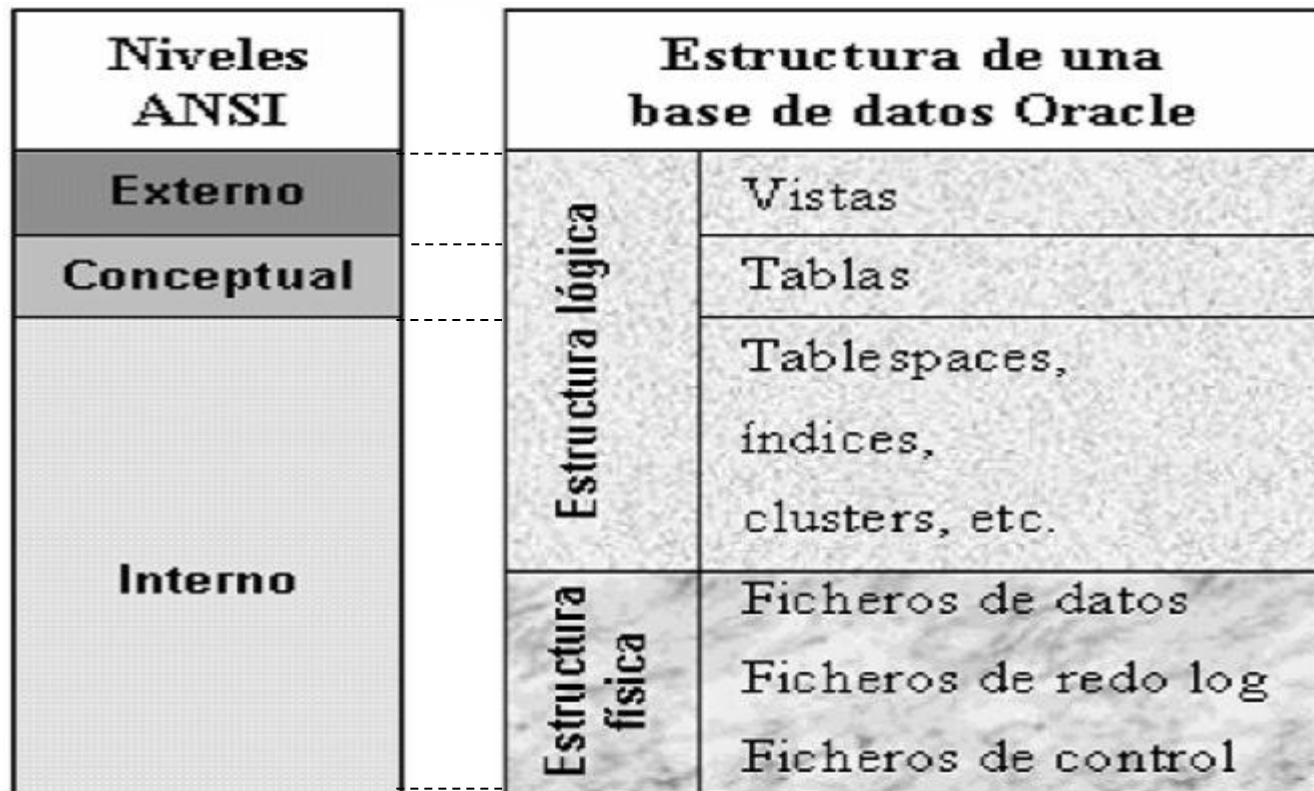
- Descripción del esquema interno: ficheros, organización, longitud, forma de acceso, tipo de registros, campos, índices,...

## Arquitectura de las BD

Nivel Externo	Visión parcial de la BD que utiliza el usuario
Nivel Conceptual	Definición de tablas, columnas y restricciones
Nivel Interno	Almacenamiento físico de los datos



# Arquitectura de las BD



Los niveles ANSI y la estructura de la base de datos Oracle.

## Arquitectura de las BD

- 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*

# Ejemplo: Arquit. de una BD de una empresa

## Esquema Externo:

- Esquema1: Departamento ventas

- Consulta ventas del mes

Num_vendedor	nom_vend	producto	cantidad	importe
.....	.....	.....	.....	.....

- Consulta de ventas de productos

Cod_producto	nom_prod	cantidad	importe
.....	.....	.....	.....

- Esquema 2: Departamento Almacén

- Consulta de productos en almacén

Cod_producto	nom_prod	cantidad	precio_unidad
.....	.....	.....	.....

## Ejemplo: Arquit. de una BD de una empresa

**Esquema Externo:** Los datos que ve el usuario

VEHÍCULO		
Matrícula:	Modelo:	FechaMatriculación:
4534 FNG	Focus	14/04/07
Marca:	Color:	CodCliente:
Ford	Negro	3

REPARACION		PROPIETARIO
IdReparacion:	FechaEntrada:	CodCliente:
0	05/05/18	4
Avería:		Nombre:
Cambio amortiguadores		Raquel Marcos
FechaSalida:		Dirección:
12/05/18		C/ La Amapala 7.
HorasEmpleadas:	Reparado:	Población:
3,00	<input checked="" type="checkbox"/>	Toledo
Observaciones:		Teléfono:
		91598811

# Ejemplo: Arquit. de una BD de una empresa

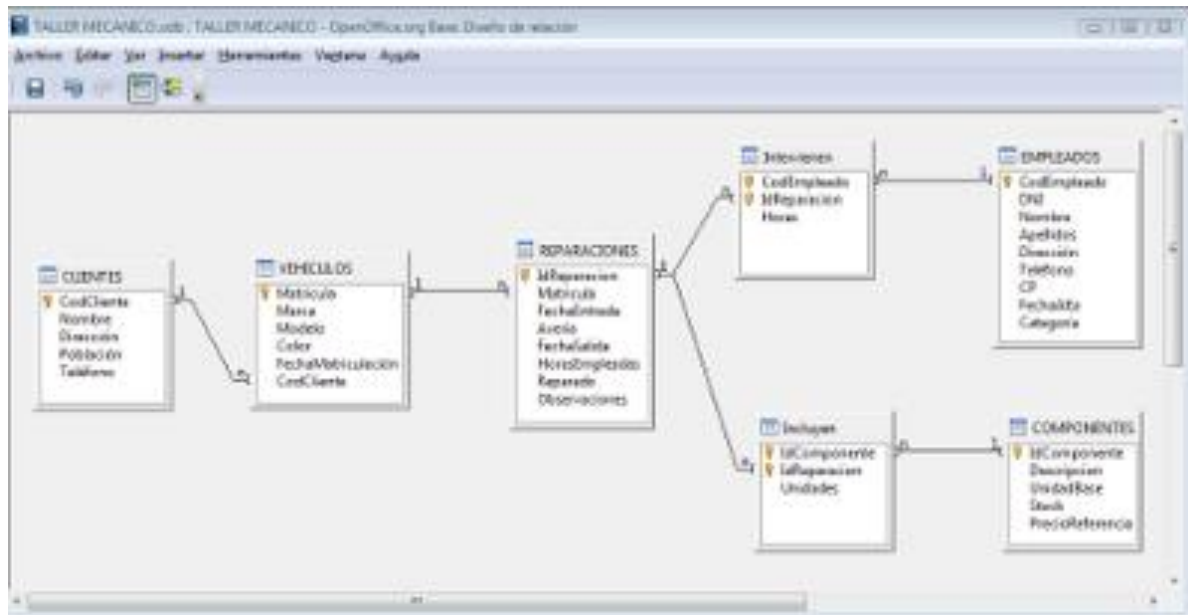
**Esquema Conceptual:** Estructura general de las tablas de la BD

(tablas)

VENDEDORES(num\_vend,nombre, apellidos, dirección,...)

PRODUCTOS (cod\_prod,nom\_prod,descrip, precio\_unidad, stock, cod\_prov...)

VENTAS(num\_vend,cod\_prod,cantidad,...)



## Ejemplo: Arquit. de una BD de una empresa

**Esquema Interno:** Como se almacenan físicamente en el ordenador  
(tablas, relaciones, tipo de los campos, longitud, validaciones,....)

Archivo VENDEDORES: num\_vend, nombre, apellidos, dirección,....

Archivo PRODUCTOS: cod\_prod, nom\_prod, descrip, precio\_unidad, stock, cod\_prov...

Archivo VENTAS: num\_vend, cod\_prod, cantidad,....

Archivo índices para VENDEDORES: clave cod\_vendedor

Archivo índices para PRODUCTOS: clave cod\_prod



## Ejemplo: Listado de nota de alumnos

### Esquema Externo

Vista según el usuario

- Esquema1: Listado de notas de alumnos con los siguientes datos: Curso, Nombre, Asignatura y Nota

Curso	Nombre	Nombre Asignatura	Nota
1	Ana	PLE	6
1	Ana	Sist. Informáticos Multiusuario y en red	8
2	Rosa	Desarr. Aplicaciones en entornos de 4ª Gener.	3
2	Juan	Desarr. Aplicaciones en entornos de 4ª Gener.	4
1	Alicia	PLE	6
1	Alicia	Sist. Informáticos Multiusuario y en red	8

## Ejemplo:

### Esquema Conceptual (1/4)

Definición de todas las tablas, columnas y Restricciones

**Tabla Alumnos: (num\_matric,nombre,curso,direccion,poblacion)**

Num_matric	Nombre	Curso	Direccion	Poblacion
11111	Ana	1	.....	.....
11110	Rosa	2	.....	.....
11122	Juan	2	.....	.....
23445	Alicia	1	.....	.....



## Ejemplo:

### Esquema Conceptual (2/4)

Definición de todas las tablas, columnas y Restricciones

**Tabla ASIGNATURAS: (**codigo\_asig**, nombre\_asig)**

<b>Codigo_asig</b>	Nombre Asignatura
1	Desarr. Aplicaciones en entornos de 4ª Gener.
2	PLE
3	Sist. Informáticos Multiusuario y en red

## Ejemplo:

### Esquema Conceptual (3/4)

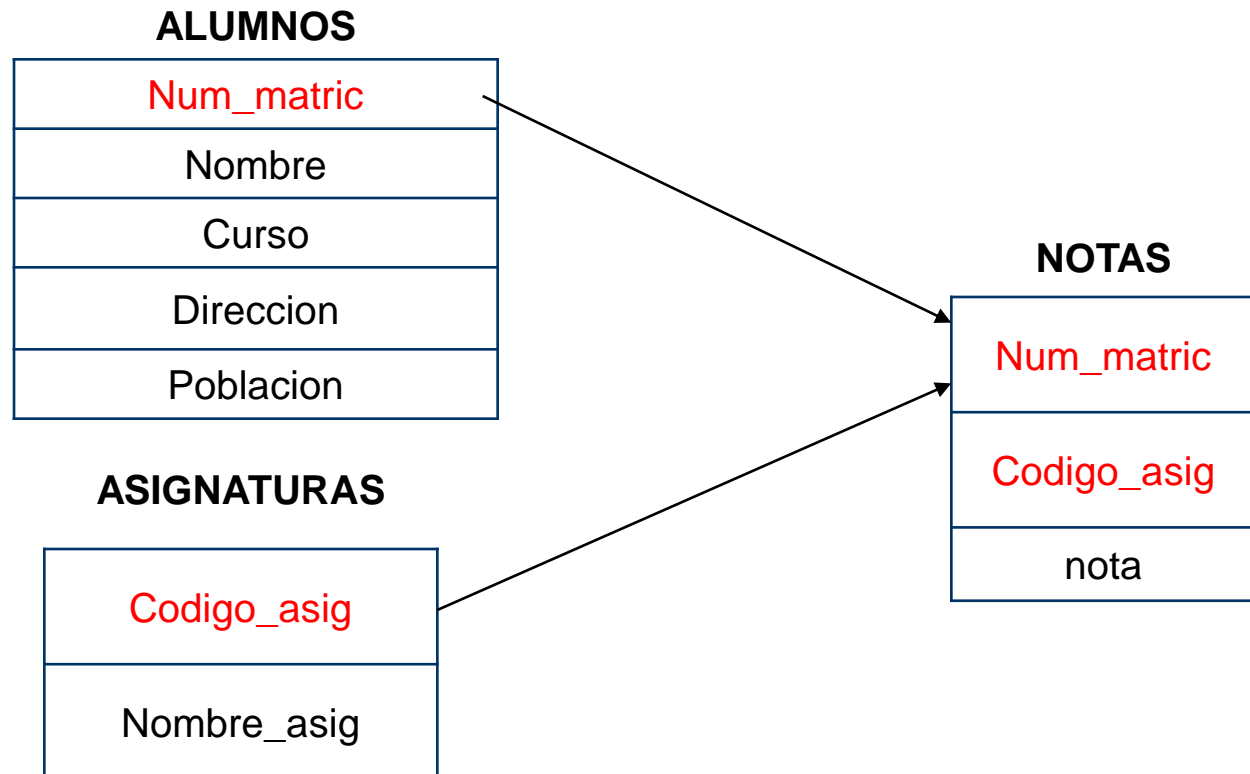
Definición de todas las tablas, columnas y Restricciones

**Tabla NOTAS: (num\_matric,codigo\_asig,nota)**

Num_matric	Codigo_asig	nota
11111	2	6
11111	3	8
11110	1	3
11122	1	4
23445	2	6
23445	3	8

## Ejemplo:

### Esquema Conceptual (4/4)



## Ejemplo:

### Esquema Interno

#### Almacenamiento físico de los datos

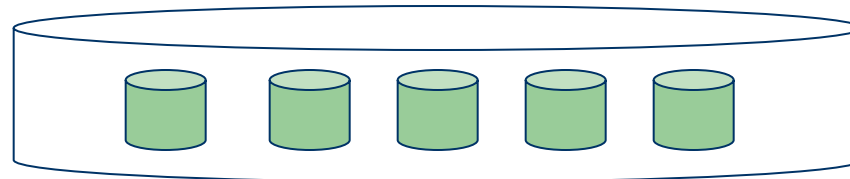
Archivo ALUMNOS: num\_matric,nombre,curso,dirección,poblacion

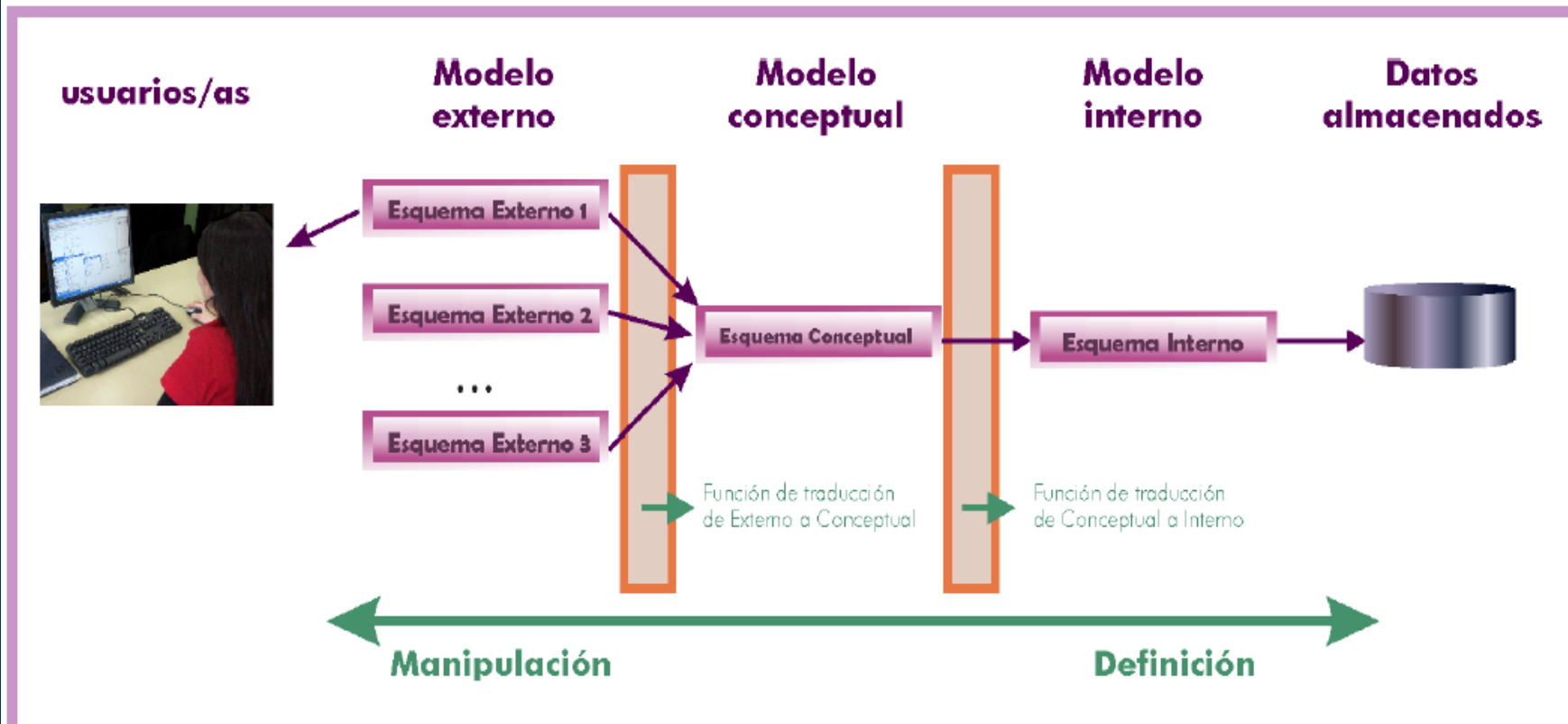
Archivo ASIGNATURAS: codigo\_asig,nombre\_asig

Archivo NOTAS: num\_matric,codigo\_asig,nota

Archivo índices ALUMNOS: clave num\_matric

Archivo índices ASIGNATURAS: clave codigo\_asig



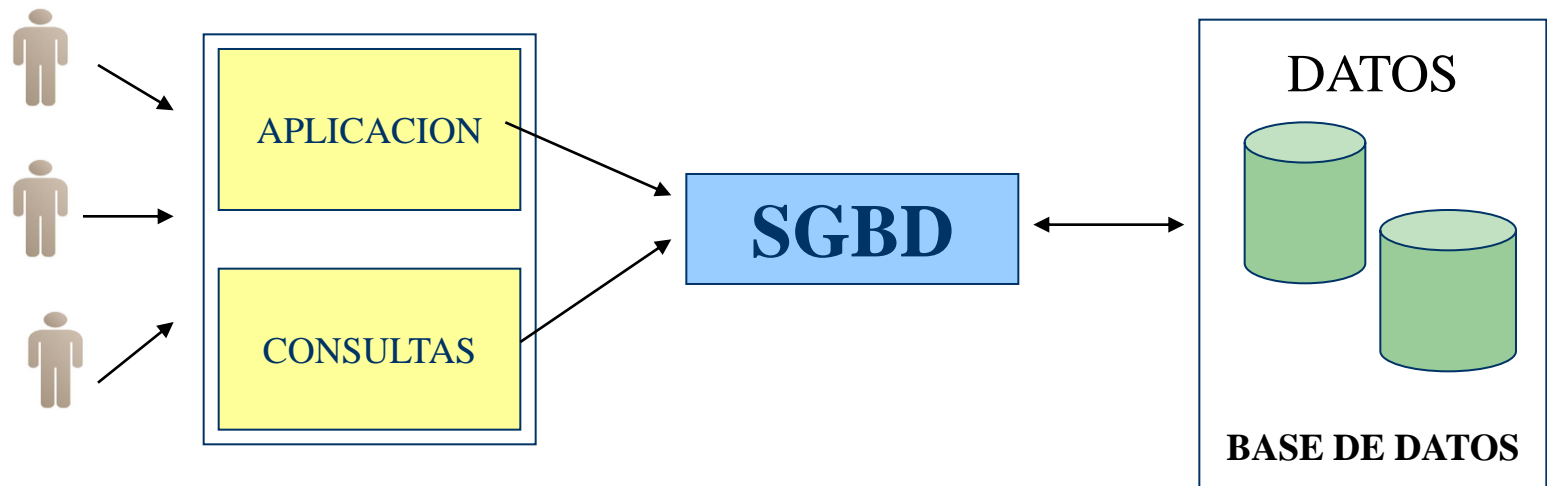


# **Sistemas Gestores de Base de Datos SGBD**

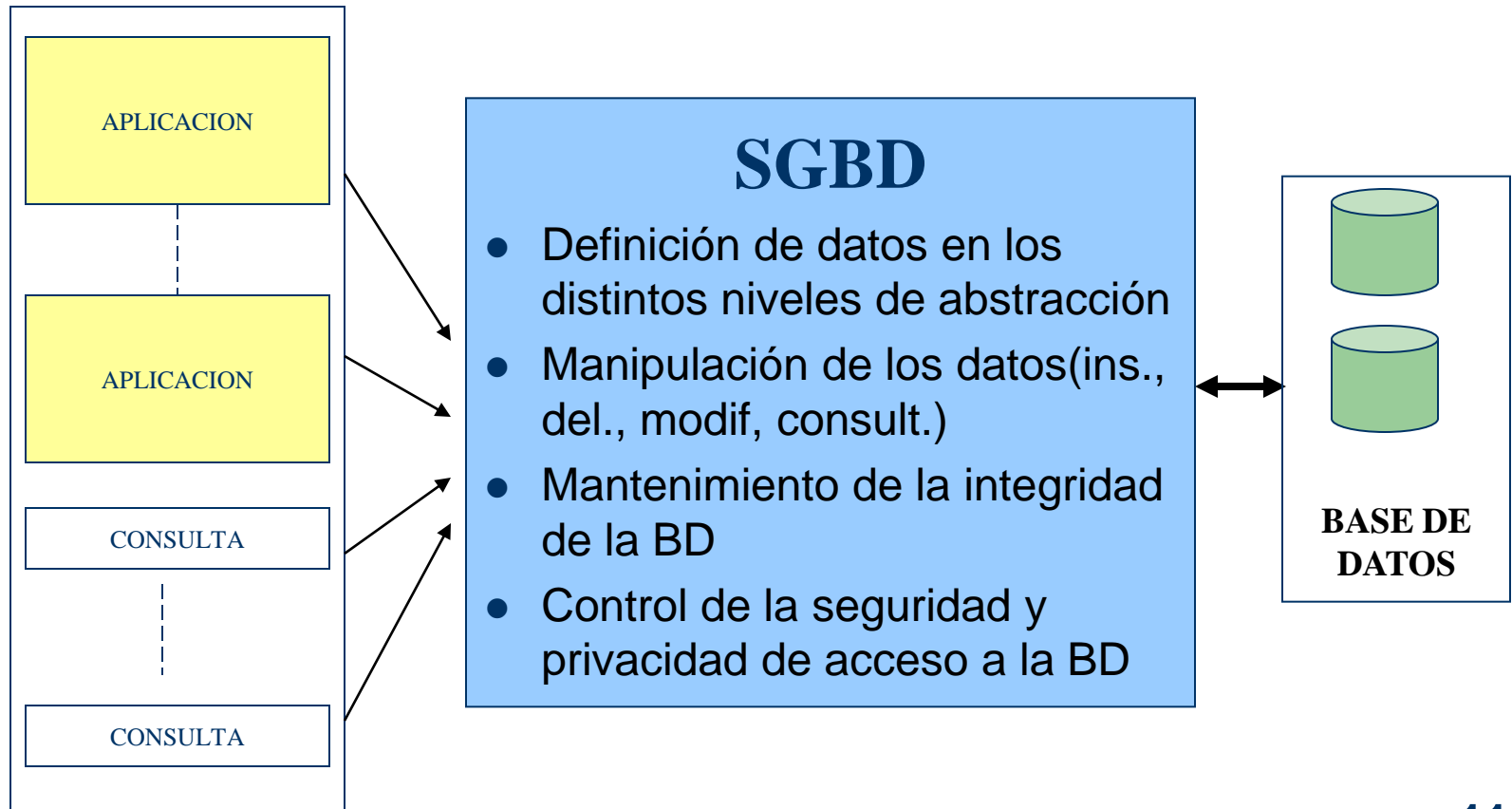
# Sistemas Gestores de Bases de Datos

## SGBD:

Software de gestión: Conjunto de programas que permiten administrar y gestionar la información de una BD.



# Sistemas Gestores de Bases de Datos





## Funciones de un SGBD

Las funciones básicas de un SGBD son:

- Descripción de los datos: DDL (Lenguaje Definición de Datos)
- Manipulación de los datos: DML (Leng. Manipulación de datos)
- Transformación de los datos entre los distintos niveles de esq. externo  $\leftrightarrow$  esq. conceptual  $\leftrightarrow$  esq. Interno; entre la representación de los datos para el usuario y como se almacenan en el equipo
- Protección de los datos:
  - Seguridad: tanto lógica como física
  - Integridad
  - Confidencialidad en el acceso
- Utilidades de administración

## Herramientas de un SGBD

El SGBD debe de proporcionar herramientas para:

- **Herramientas para la estructura de la base de datos.** Así como para la creación y especificación de los datos.
- **Herramientas para administrar y crear la estructura física** requerida en las unidades de almacenamiento.
- **Herramientas para la manipulación de los datos de las bases de datos,** para añadir, modificar, suprimir o consultar datos.
- **Herramientas de recuperación** en caso de desastre

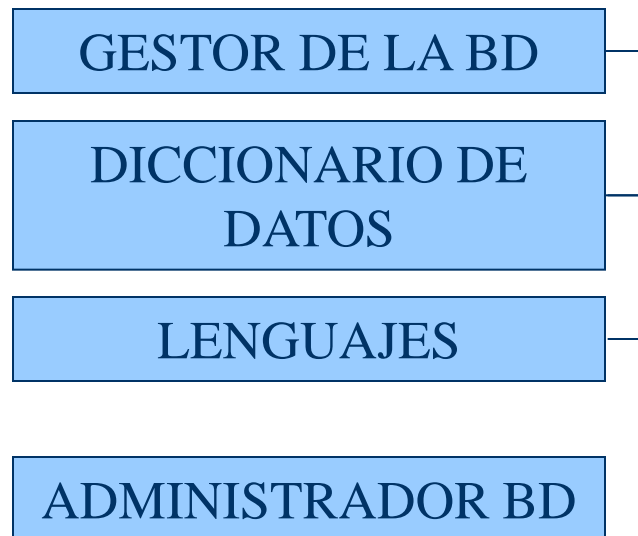
## Herramientas de un SGBD

El SGBD debe de proporcionar herramientas para:

- ***Herramientas para la creación de copias de seguridad***
- ***Herramientas para la gestión de la comunicación de la base de datos***
- ***Herramientas para la creación de aplicaciones*** que utilicen esquemas externos de los datos
- ***Herramientas de instalación de la base de datos***
- ***Herramientas para la exportación e importación de datos***

# Sistemas Gestores de Bases de Datos

## Componentes de un SGBD



# Sistemas Gestores de Bases de Datos

## Gestor de la BD:

Software transparente al usuario que garantiza la privacidad, seguridad, integridad y el acceso a los datos, además de la comunicación con el SO.

Interfaz entre los datos, los programas y los usuarios

Diccionario de la Base Datos

Lenguajes

Administrador de BD

# Sistemas Gestores de Bases de Datos

## Gestor de la BD

### Diccionario de la Base Datos:

Donde se encuentra la descripción de toda la BD: estructura de los datos, relaciones y la gestión e implementación de la BD. Contendrá:

- Descripción de los distintos niveles de abstracción de la BD y sus transformaciones (*no es lo mismo como se representa la información en el ordenador a como la interpreta o ve el usuario*)
- Restricciones de los datos
- Acceso a los datos
- Cuentas de usuario y sus privilegios
- Esquemas externos de cada programa, sus usuarios y autorizaciones

## Lenguajes

## Administrador de BD

# Sistemas Gestores de Bases de Datos

Gestor de la BD

Diccionario de la Base de Datos

Lenguajes:

Proporciona lenguajes para definir y manipular los datos de la base de datos a los administradores y los usuarios.

- DDL: Lenguaje Definición de Datos
- DML: Lenguaje Manipulación de Datos o Lenguaje de Consulta DQL
- DCL: Lenguaje Control de Datos (Administrador)

Administrador de BD

# Sistemas Gestores de Bases de Datos

Gestor de la BD

Diccionario de la Base de Datos

Lenguajes

## Administrador de BD: (DBA)

Responsable de la seguridad, del control y del mantenimiento de la BD. Sus tareas:

- Definir el esquema lógico
- Definir el esquema físico: estructura almacenamiento y método de acceso
- Subesquemas o visión del usuario de la BD
- Asignación de privilegios
- Seguridad de los datos en caso de fallos
- Mantenimiento del esquema lógico y físico de la BD



## Diccionario de Datos

- **Directorio de datos o catálogo:** son las especificaciones necesarias para pasar la representación externa de los datos a su representación interna
- **Diccionario de datos:** contiene información sobre los datos almacenados en la BD desde el punto de vista de los usuarios, de forma que estos puedan comprender el significado: descripción, estructuras, consideraciones de seguridad,...

Los SGBD relacionales almacenan información sobre los datos para ser almacenadas tanto por el SGBD como por el usuario, por lo que este directorio/diccionario de datos que implementan los sistemas relacionales se conoce como **CATALOGO**

# Usuarios de un SGBD

## Tipos de usuarios

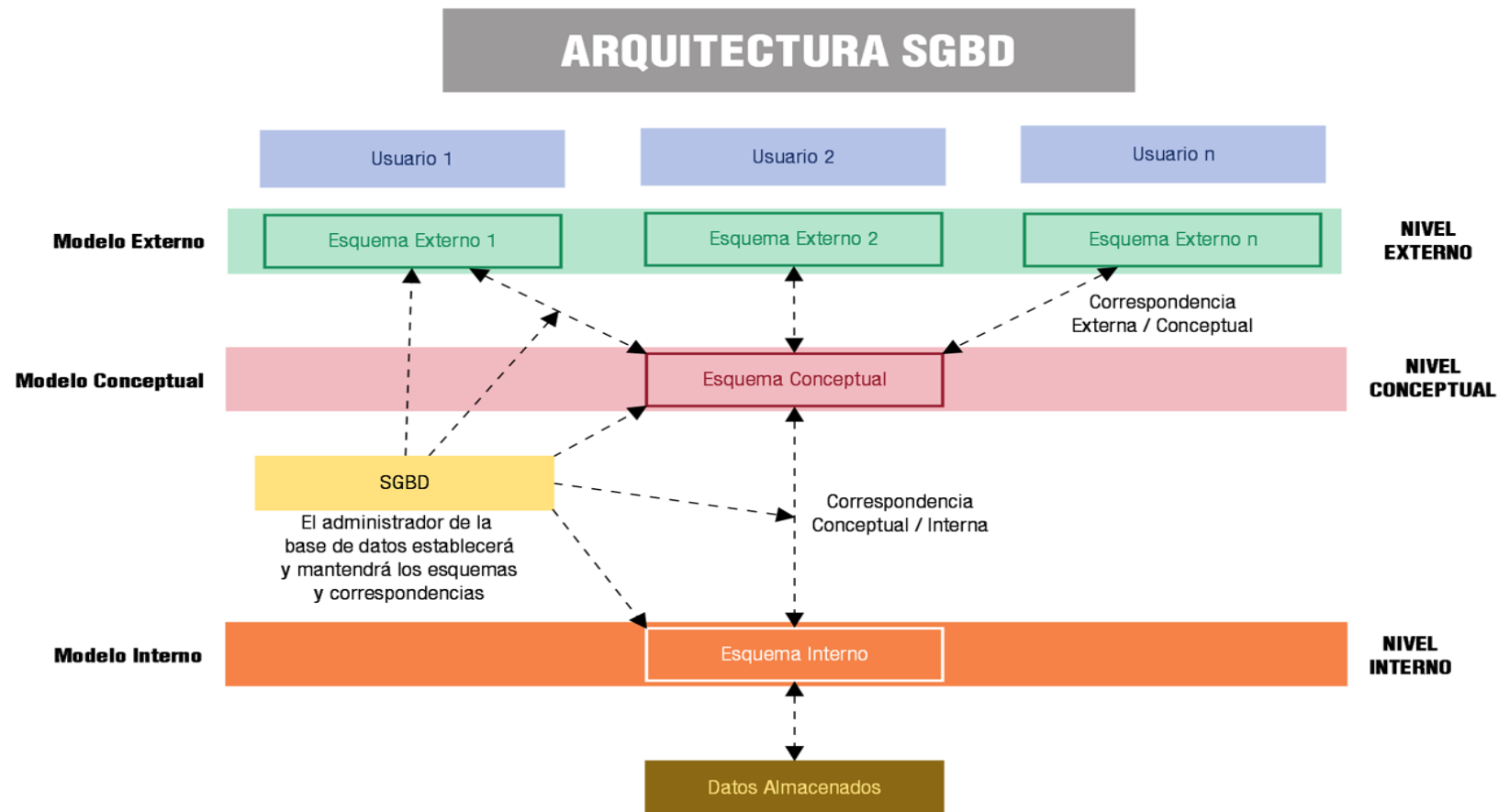
- **Programadores:** Son los responsables de la creación y ajuste de las aplicaciones que atacan a los datos. Emplean DDL, DML y cualquier lenguaje anfitrión.
- **Usuarios expertos:** Emplean las utilidades de la base de datos y el DML para acceder a los datos y realizar sus propios procesos sobre los objetos para los que se les ha concedido permiso.
- **Usuarios ocasionales:** que 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.

# Usuarios de un SGBD

## Tipos de usuarios

- **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. 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 y 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

# Arquitectura de un SGBD



# Modelos de BD

# Modelos de Bases de Datos

Podemos establecer distintos tipos de modelos de BD en función a:

- Modelos lógicos en el que se basan
- Al número de usuarios al que dan servicios
- Al número de sitios en los que está distribuida la BD
- Según propósito para el que van a ser usadas

# Modelos de Bases de Datos

Podemos establecer distintos tipos de modelos de BD en función a:

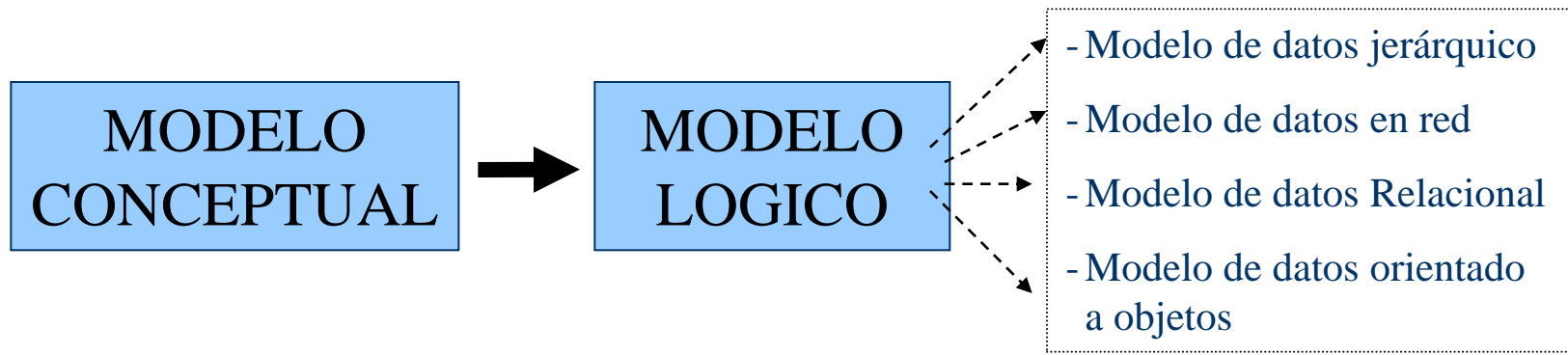
- Modelos lógicos en el que se basan
- Al número de usuarios al que dan servicios
- Al número de sitios en los que está distribuida la BD
- Según propósito para el que van a ser usadas

# Modelos de Bases de Datos

## En función del modelo lógico

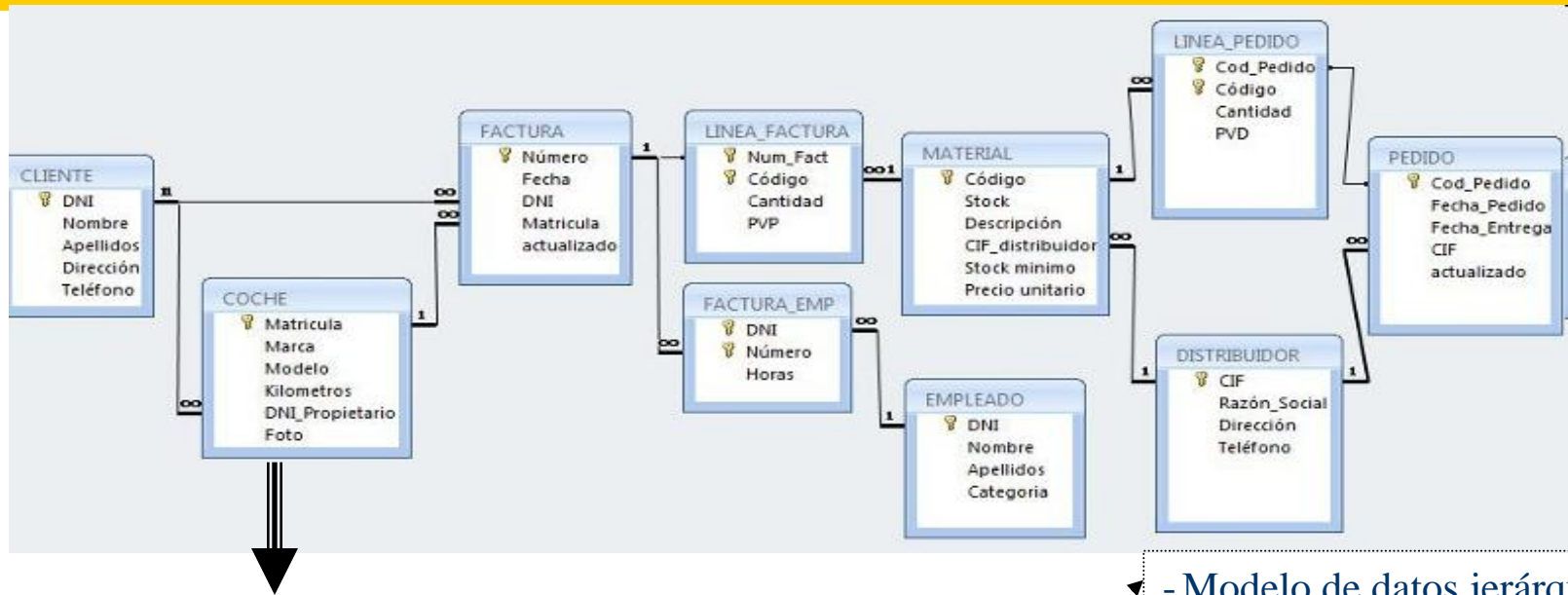
*Modelar* consiste en definir un mundo abstracto de forma que las conclusiones que se puedan sacar de él coincidan con las manifestaciones del mundo real.

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.





# Modelos de Bases de Datos En función del modelo lógico



**MODELO  
CONCEPTUAL**



**MODELO  
LOGICO**

- Modelo de datos jerárquico
- Modelo de datos en red
- Modelo de datos Relacional
- Modelo de datos orientado a objetos

# Modelos de Bases de Datos

## En función del modelo lógico

### MODELO DE DATOS JERÁRQUICO

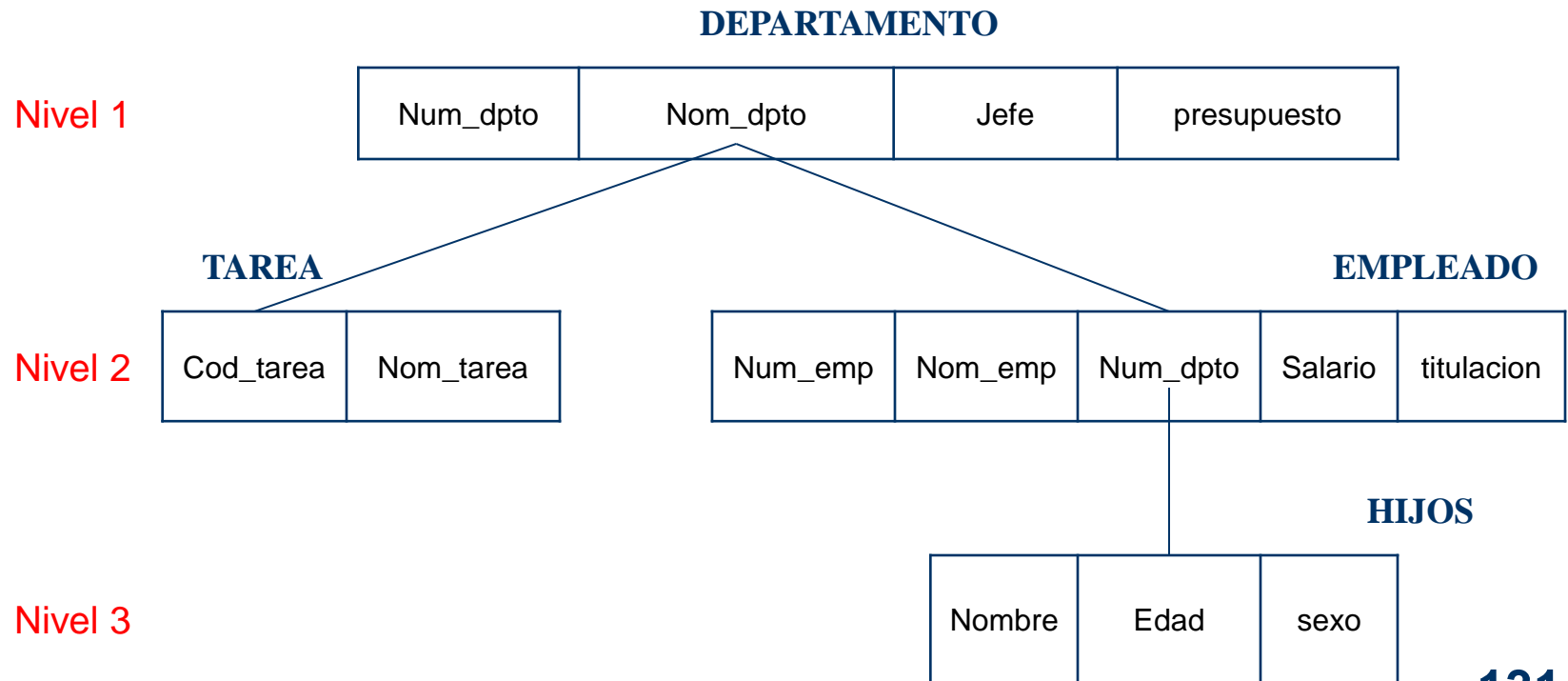
- Estructura arborescente
- El árbol está compuesto por una jerarquía de **NODOS**, distribuido en niveles, donde existe un nodo **RAIZ** del que parte toda la jerarquía
- **NODO** = registro llamado **SEGMENTO** con sus **CAMPOS**

Ejemplos: **IMS** de IBM y **SYSTEM 2000** de INTEL

# Modelos de Bases de Datos

## En función del modelo lógico

### MODELO DE DATOS JERÁRQUICO. Ejemplo



# Modelos de Bases de Datos

## En función del modelo lógico

### MODELO DE DATOS JERÁRQUICO

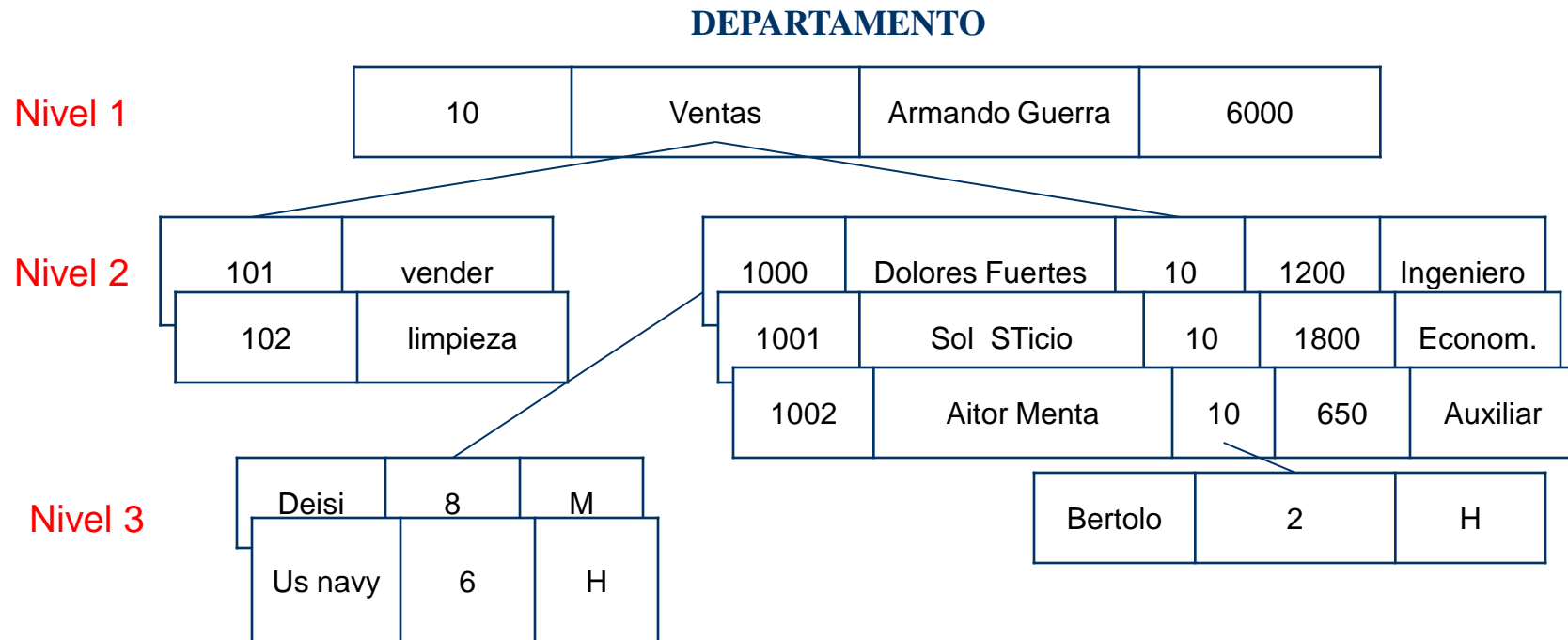
Características de este modelo:

- Solo existen relaciones 1:M
- No se permite más de una relación entre dos segmentos
- No se admiten relaciones del tipo M:N
- No se permite que un segmento HIJO tenga más de un PADRE
- El árbol se recorre en un cierto orden (arriba → abajo)
- El acceso a cualquier segmento siempre se realiza a partir del raíz

# Modelos de Bases de Datos

## En función del modelo lógico

### MODELO DE DATOS JERÁRQUICO. Ejemplo



# Modelos de Bases de Datos

## En función del modelo lógico

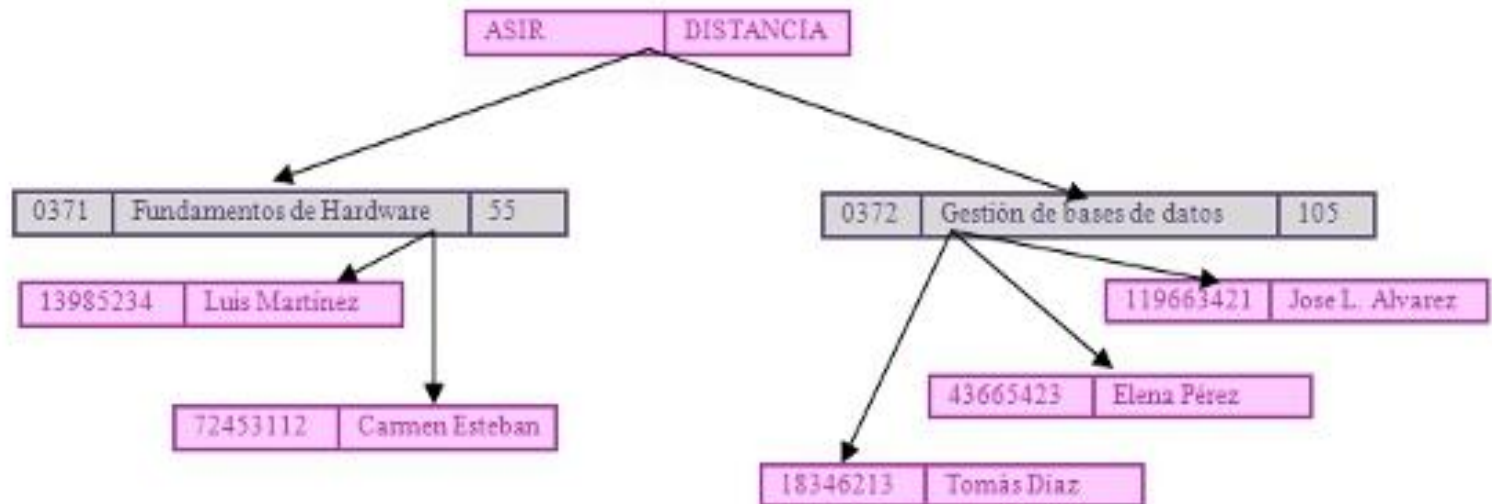
### MODELO DE DATOS JERÁRQUICO. Ejemplo

MODULOS

CODIGO	NOMBRE	HORAS
--------	--------	-------

PROFESORES

NRP	NOMBRE
-----	--------



# Modelos de Bases de Datos En función del modelo lógico

## MODELO DE DATOS EN RED

- Estructura en red (PLEX)
- Las entidades o tablas se representan en NODOS
- Las relaciones son líneas que unen los nodos
- Cualquier entidad se puede vincular con otra (punteros)
- Es posible describirla en términos padres e hijos, pero un hijo puede tener varios padres

# Modelos de Bases de Datos

## En función del modelo lógico

### MODELO DE DATOS EN RED.

#### Ejemplo de este modelo: DBTG de CODASYL

- Tipo de Registro: representa un nodo
- Elementos: es un campo de datos (DNI)
- Agregados de datos: Conjunto de campos dentro de un mismo campo (Ej. fecha)
- Conjunto: Relaciona dos tipos de registros (propietario y miembro). Admite relaciones 1:M y 1:1



# Modelos de Bases de Datos

## En función del modelo lógico

### MODELO DE DATOS EN RED.

#### Ejemplo de este modelo: DBTG de CODASYL

Entre ellos destaca: DMS 1100 (UNIVAC), DBOMP (IBM)

##### PROVEEDOR

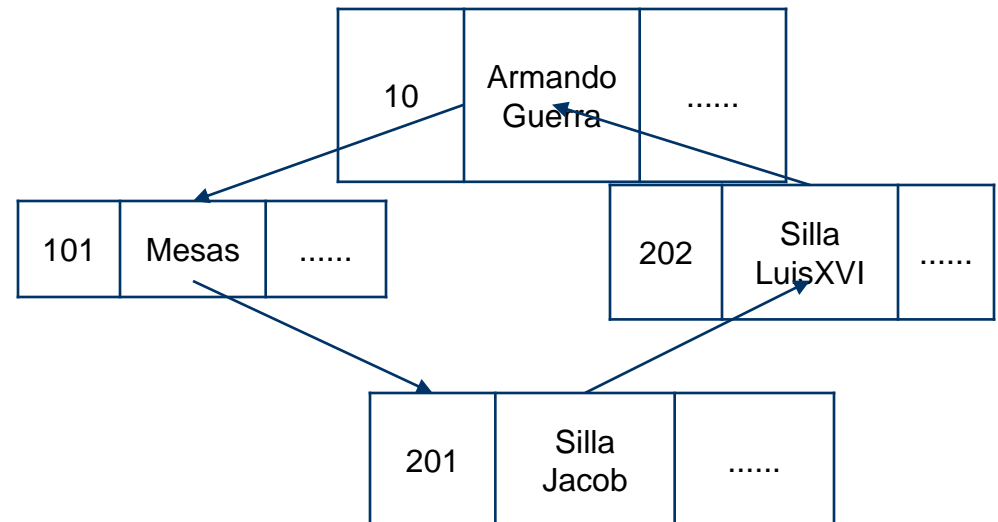
Cod	Nom_prov	.....
-----	----------	-------

Propietario

##### ARTICULO

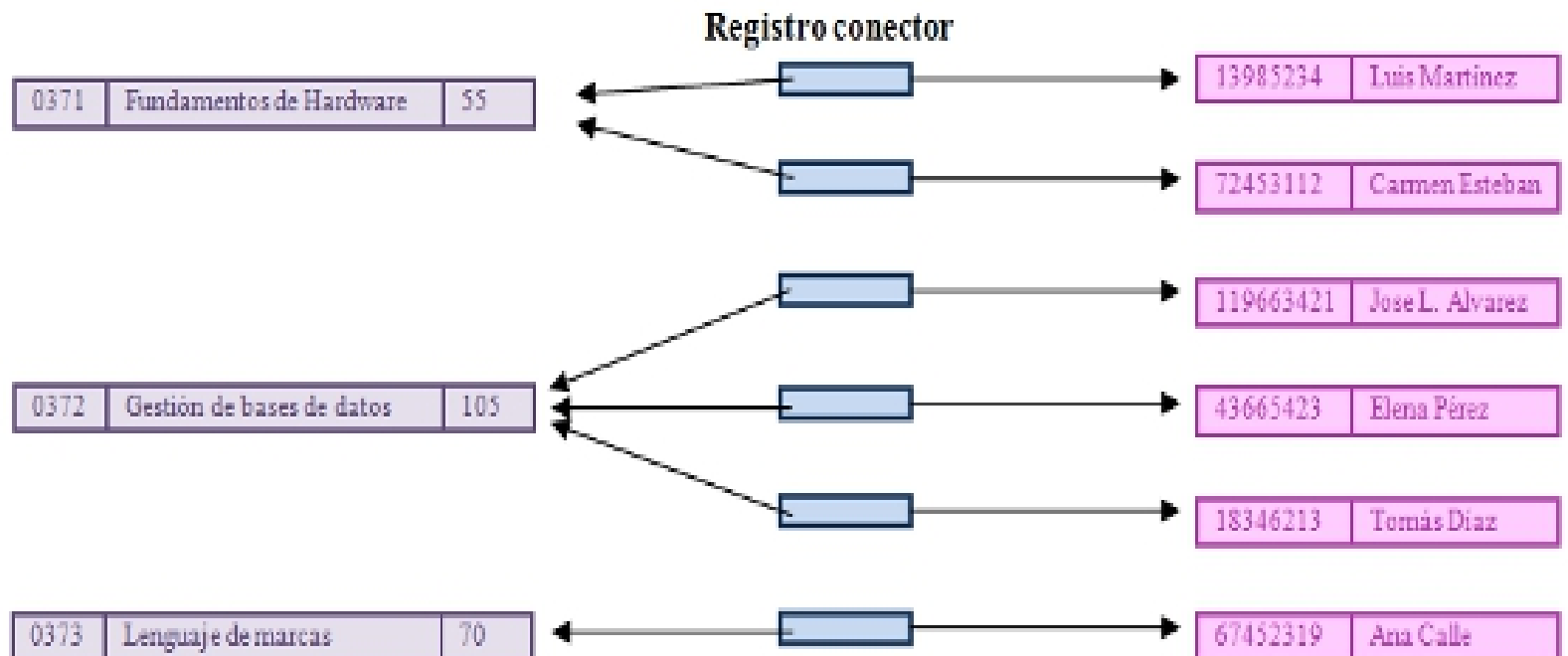
Cod	Nom_art	.....
-----	---------	-------

Miembro



# Modelos de Bases de Datos En función del modelo lógico

## MODELO DE DATOS EN RED



# Modelos de Bases de Datos

## En función del modelo lógico

### MODELO DE DATOS RELACIONAL

- Utiliza tablas bidimensionales para la representación lógica de los datos y las relaciones
- Se apoya en la teoría de matemáticas relacional
- Se trata de un modelo lógico que referencia la representación lógica de información y no es directamente aplicable a la representación física

# Modelos de Bases de Datos

## En función del modelo lógico

### MODELO DE DATOS RELACIONAL

#### Ventajas:

- Representación sencilla del conjunto de datos de la BD (modelo conceptual)
- Fácil de entender y usar por un usuario
- Los usuarios no necesitan saber donde se encuentran físicamente los datos
- Independencia del modelo conceptual y los programas de aplicación que utilizan los datos

# Modelos de Bases de Datos

## En función del modelo lógico

### MODELO DE DATOS RELACIONAL

#### 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.

# Modelos de Bases de Datos En función del modelo lógico

## MODELO DE DATOS RELACIONAL

Resumen del modelo:

- El elemento principal es la *relación*
- Cada relación se representa mediante una *tabla*
- Es el modelo más utilizado actualmente

Ejemplos: ORACLE, INFOMIX, DBANS IV, MySQL....

# Modelos de Bases de Datos

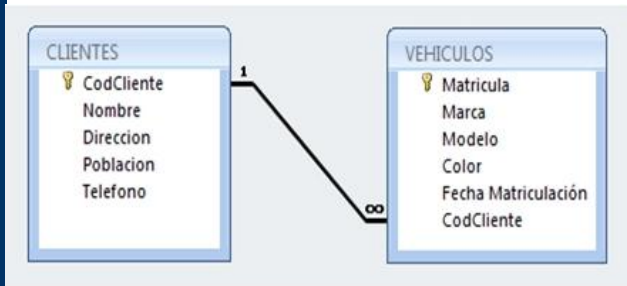
## En función del modelo lógico

### 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

# Modelos de Bases de Datos

## En función del modelo lógico



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 Sabello	C/ La plaza 72	Madrid	914409071
5	Jorge Peña	C/ Fresnedo, 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



# Modelos de Bases de Datos

## En función del modelo lógico

### MODELO ORIENTADO A OBJETOS

- 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

# Modelos de Bases de Datos

## En función del modelo lógico

### MODELO ORIENTADO A OBJETOS. Componentes

- Los **objetos** del modelo: equivale a una entidad individual del modelo E-R.
- Los **atributos** que describen las propiedades de ese objeto.
- Los objetos que comparten características similares se agrupan en **clases**.
- Una **clase** es un conjunto de objetos similares con estructura (atributos) y comportamiento (métodos) compartidos. Se podría comparar a una entidad del modelo E-R pero se diferencian en que una clase contiene una serie de procedimientos llamados *métodos*.

# Modelos de Bases de Datos

## En función del modelo lógico

### MODELO ORIENTADO A OBJETOS. Componentes

- 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.
- 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
- 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.

# Modelos de Bases de Datos

## En función del modelo lógico

### MODELO ORIENTADO A OBJETOS. Ejemplo

- Ejemplo de representación de una factura emitida a nuestros clientes.

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

En cada factura un cliente puede comprar varios artículos



# Modelos de Bases de Datos

## En función del modelo lógico

### **MODELO ORIENTADO A OBJETOS. Ventajas**

- Agrega contenido semántico. En el ejemplo anterior dentro del objeto factura se incluyen las relaciones entre el cliente y la factura y entre la factura y los artículos.
- La representación visual facilita la comprensión de relaciones complejas dentro de los objetos y entre ellos.
- Mantiene la integridad de la base de datos al implementar la herencia entre objetos.
- Independencia estructural de los datos ya que son objetos autónomos.

# Modelos de Bases de Datos

## En función del modelo lógico

### **MODELO ORIENTADO A OBJETOS. Desventajas**

- No existen estándares de modelo de datos orientados a objetos. Sobre todo método de acceso a datos estándar.
- El método de acceso se parece al jerárquico y en red.
- El modelado y ejecución es difícil debido a que tienen mucho contenido semántico y no hay estándares.
- La complejidad y elevados requerimientos del sistema hace que las transacciones sean lentas.

# Modelos de Bases de Datos

Podemos establecer distintos tipos de modelos de BD en función a:

- Modelos lógicos en el que se basan
- Al número de usuarios al que dan servicios
- Al número de sitios en los que está distribuida la BD
- Según propósito para el que van a ser usadas

# Modelos de Bases de Datos

En función del número de usuarios al 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.



# Modelos de Bases de Datos

Podemos establecer distintos tipos de modelos de BD en función a:

- Modelos lógicos en el que se basan
- Al número de usuarios al que dan servicios
- Al número de sitios en los que está distribuida la BD
- Según propósito para el que van a ser usadas

# Modelos de Bases de Datos

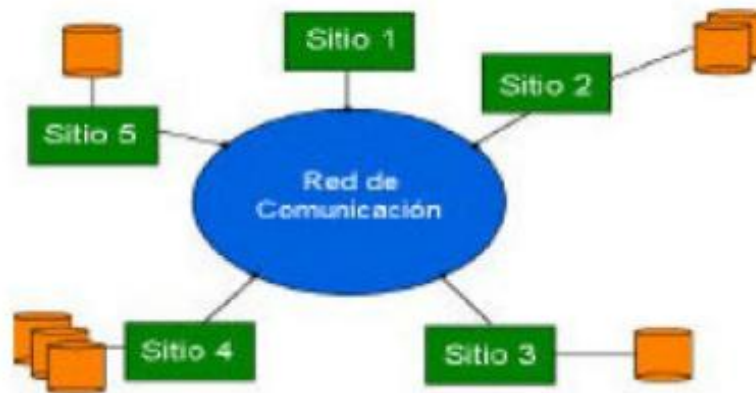
En función del número de sitios en los que está distribuida la BD

- **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.

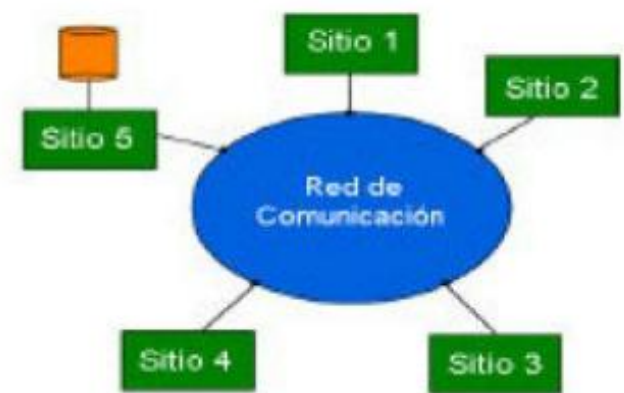
# Modelos de Bases de Datos

En función del número de sitios en los que está distribuida la BD

distribuida



centralizada



# Modelos de Bases de Datos

Podemos establecer distintos tipos de modelos de BD en función a:

- Modelos lógicos en el que se basan
- Al número de usuarios al que dan servicios
- Al número de sitios en los que está distribuida la BD
- Según propósito para el que van a ser usadas

# Modelos de Bases de Datos

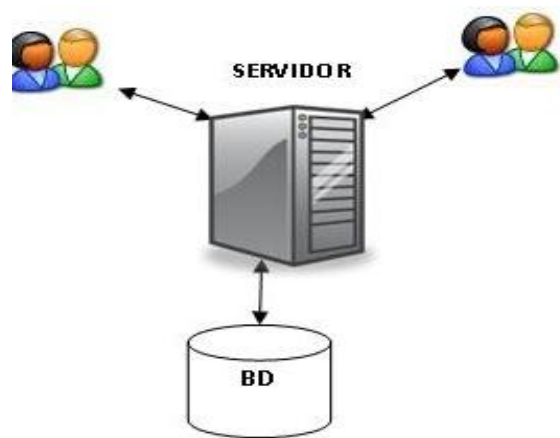
En función al propósito para la que han sido creadas

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

**Ejemplo:** 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.

# Modelos de BD Centralizadas

- El SGBD está implantado en una sola plataforma u ordenador desde donde se gestiona directamente, de modo centralizado, la totalidad de los recursos.
- Es la arquitectura de los centros de proceso de datos tradicionales.
- Se basa en tecnologías sencillas, muy experimentadas y de gran robustez.



# Modelos de BD Centralizadas

## Ventajas e inconvenientes de las bases de datos centralizadas

Ventajas	Inconvenientes
<ul style="list-style-type: none"> <li>- Se evita la redundancia debido a la posibilidad de inconsistencias y al desperdicio de espacio.</li> <li>- Se evita la inconsistencia. Ya que si un hecho específico se representa por una sola entrada, la no-concordancia de datos no puede ocurrir.</li> <li>- La seguridad se centraliza.</li> <li>- Puede conservarse la integridad.</li> <li>- El procesamiento de los datos ofrece un mejor rendimiento.</li> <li>- Mantenimiento más barato. Mejor uso de los recursos y menores recursos humanos.</li> </ul>	<ul style="list-style-type: none"> <li>- Un mainframe en comparación de un sistema distribuido no tiene mayor poder de cómputo.</li> <li>- Cuando un sistema de bases de datos centralizado falla, se pierde toda disponibilidad de procesamiento y sobre todo de información confiada al sistema.</li> <li>- En caso de un desastre o catástrofe, la recuperación es difícil de sincronizar.</li> <li>- Las cargas de trabajo no se pueden difundir entre varias computadoras, ya que los trabajos siempre se ejecutarán en la misma máquina.</li> <li>- Los departamentos de sistemas retienen el control de toda la organización.</li> <li>- Los sistemas centralizados requieren un mantenimiento central de datos.</li> </ul>

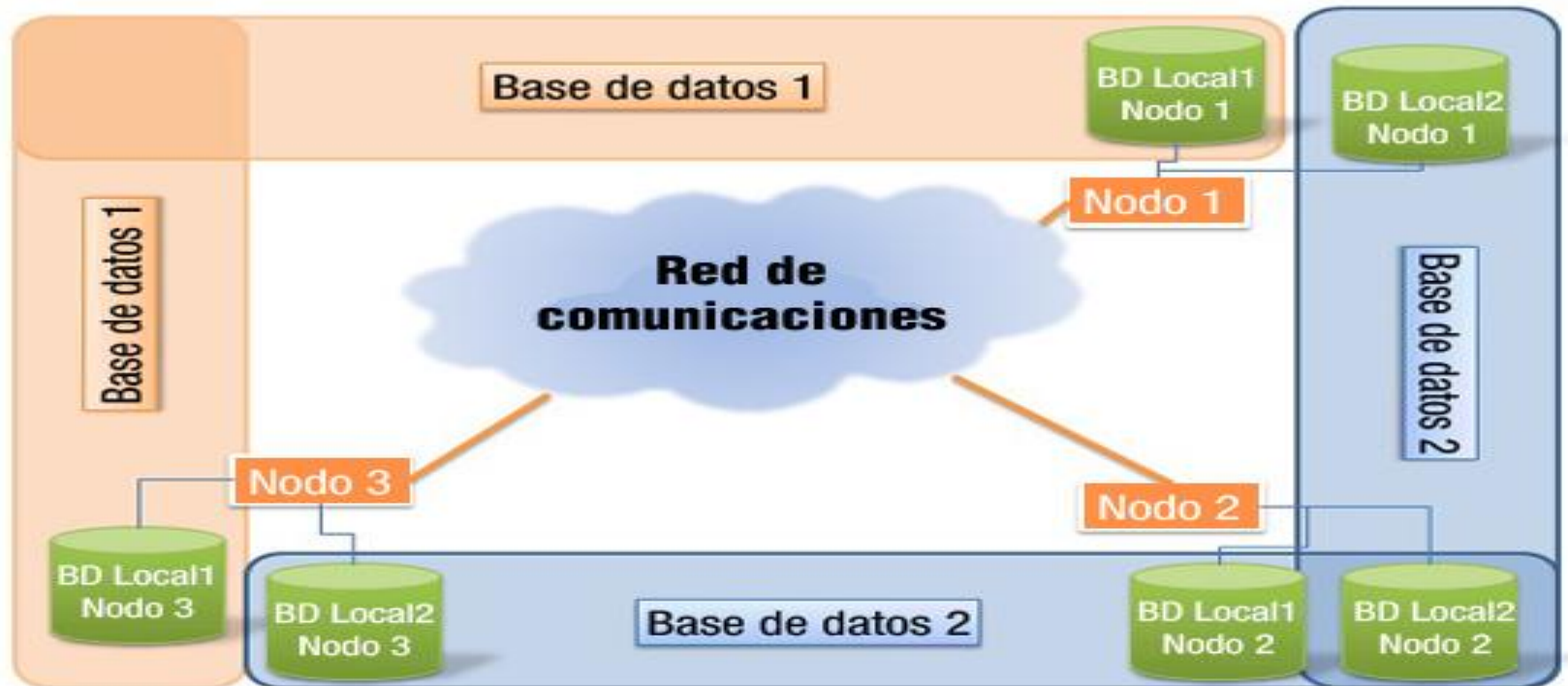
# Modelos de BD Distribuidas

- **Base de datos distribuida (BDD):** es un conjunto de múltiples bases de datos lógicamente relacionadas las cuales se encuentran distribuidas entre diferentes nodos interconectados por una red de comunicaciones.
- **Sistema de bases de datos distribuida (SBDD):** es un sistema en el cual múltiples sitios de bases de datos están ligados por un sistema de comunicaciones, de tal forma que, un usuario en cualquier sitio puede acceder los datos en cualquier parte de la red exactamente como si los datos estuvieran almacenados en su sitio propio.
- **Sistema gestor de bases de datos distribuida (SGBDD):** es aquel que se encarga del manejo de la BDD y proporciona un mecanismo de acceso que hace que la distribución sea transparente a los usuarios. El término transparente significa que la aplicación trabajaría, desde un punto de vista lógico, como si un solo SGBD ejecutado en una sola máquina, administrara esos datos.



# Modelos de BD Distribuidas

## BASES DE DATOS RELACIONALES DISTRIBUIDAS, HOMOGÉNEAS Y ALTAMENTE INTEGRADAS



# Modelos de BD Distribuidas

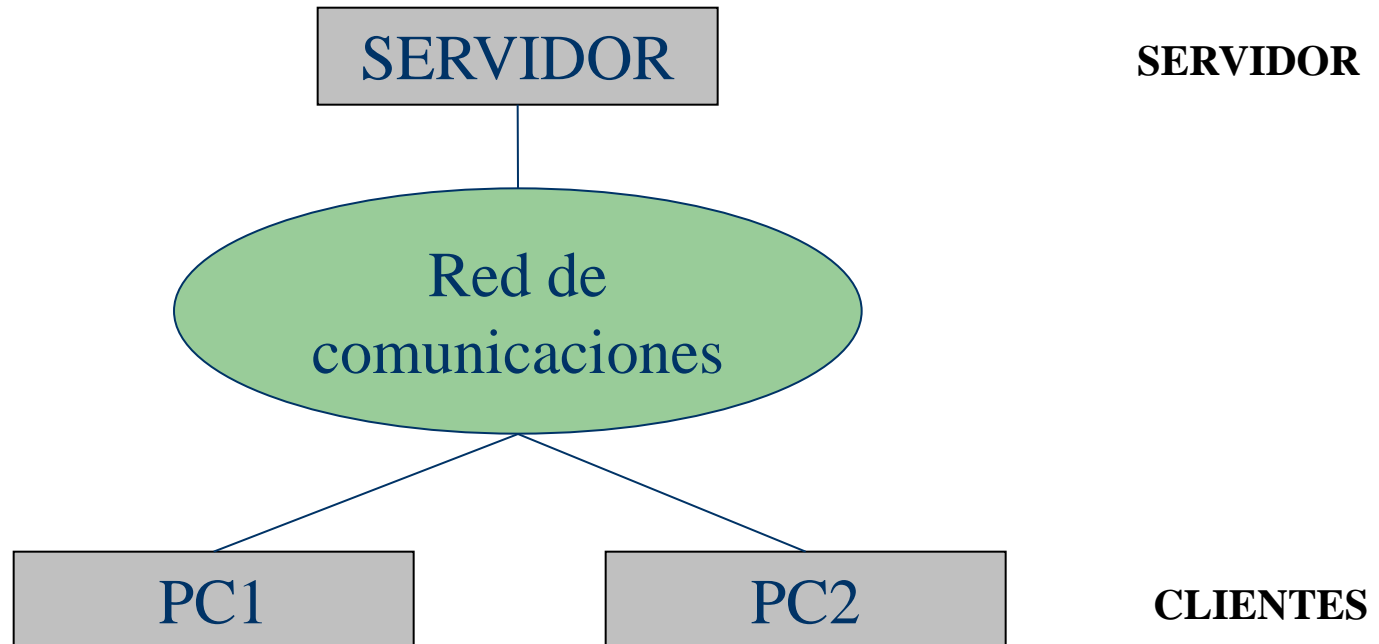
## Ventajas e inconvenientes de las bases de datos distribuidas

Ventajas	Inconvenientes
<ul style="list-style-type: none"> <li>- El acceso y procesamiento de los datos es más rápido ya que varios nodos comparten carga de trabajo.</li> <li>- Desde una ubicación puede accederse a información alojada en diferentes lugares.</li> <li>- Los costes son inferiores a los de las bases centralizadas.</li> <li>- Si un nodo deja de funcionar el sistema completo no deja de funcionar.</li> <li>- Permiten la incorporación de nodos de forma flexible y fácil.</li> <li>- Aunque los nodos están interconectados, tienen independencia local.</li> </ul>	<ul style="list-style-type: none"> <li>- La seguridad es necesario incrementarla</li> <li>- Complejidad añadida para garantizar la coordinación apropiada entre los nodos.</li> <li>- La inversión inicial es menor, pero el mantenimiento y control puede resultar costoso.</li> <li>- 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.</li> <li>- El intercambio de mensajes y conseguir la coordinación entre los distintos nodos son una forma de sobrecarga que no hay en las centralizados.</li> <li>- Dada la complejidad del procesamiento entre nodos es difícil asegurar la corrección de los algoritmos, el funcionamiento correcto durante un fallo o la recuperación.</li> </ul>

# Arquitectura Cliente-Servidor

# Arquitectura Cliente/Servidor

**Objetivo de un sistema BD:** facilitar el desarrollo y ejecución de aplicaciones. Un sistema de BD esta formado por dos partes



## Arquitectura Cliente/Servidor

Componentes del Software para una arquitectura cliente/servidor:

- Software de gestión de datos (servidor)
- Software de interacción con el usuario y presentación (cliente)
- Software de desarrollo de aplicaciones (cliente)

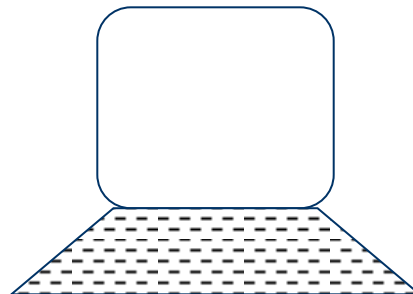
Además:

- SO en red tanto en cliente como en servidor
- Aplicaciones de BD
- Aplicaciones de comunicaciones

## Arquitectura Cliente/Servidor

Configuraciones de una arquitectura cliente/servidor:

- BASADA EN ANFITRIÓN:  
el cliente y el servidor son la misma máquina

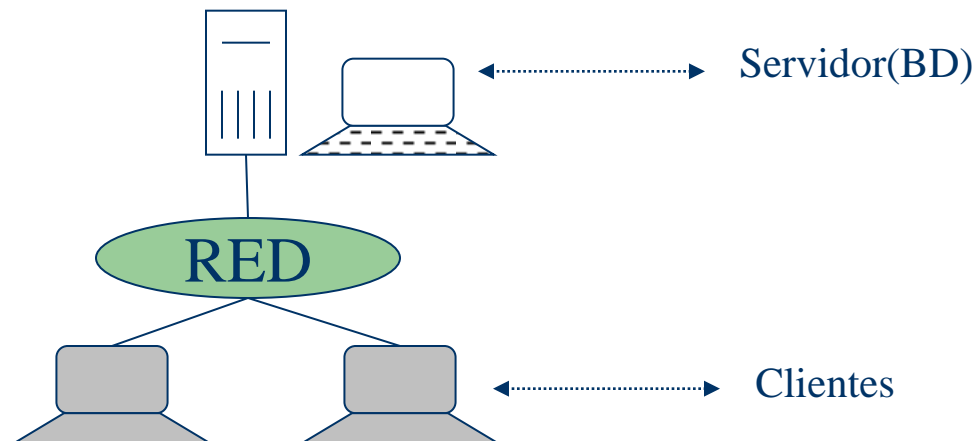


Cliente y servidor

# Arquitectura Cliente/Servidor

## Configuraciones de una arquitectura cliente/servidor:

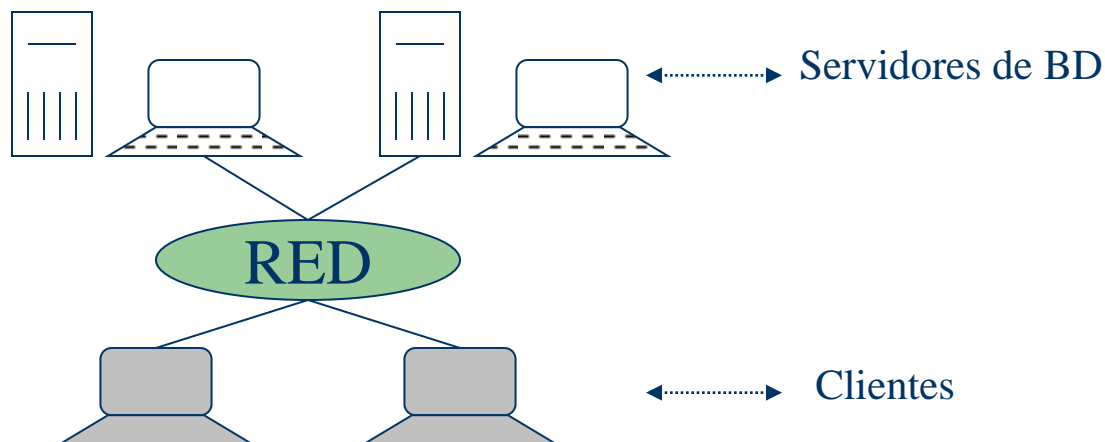
- CLIENTE/SERVIDOR:
  - BD esta en el Servidor
  - Usuarios acceden a la BD desde su PC a través del cliente y con una aplicación que se le instalará en su equipo
  - El acceso se realiza a través de una red



# Arquitectura Cliente/Servidor

## Configuraciones de una arquitectura cliente/servidor:

- PROCESAMIENTO DISTRIBUIDO:
  - BD esta repartida en más de un Servidor
  - Usuarios acceden a la BD desde su PC a través del cliente sin necesidad de conocer donde están físicamente los datos
  - El acceso se realiza a través de una red

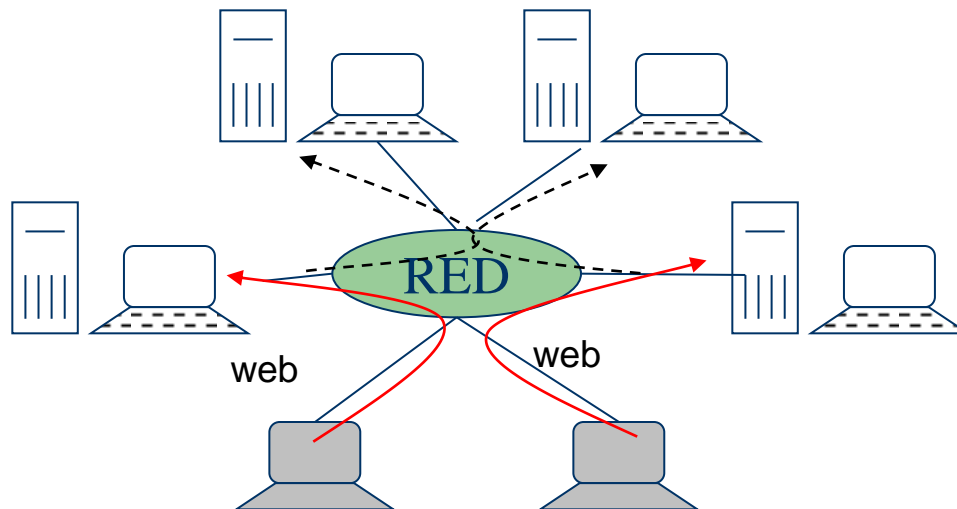




# Arquitectura Cliente/Servidor

## Configuraciones de una arquitectura cliente/servidor:

- Basada en servidores de aplicaciones:



Servidores de BD

Servidores de aplicación

Clientes

## Arquitectura Cliente/Servidor

### Configuraciones de una arquitectura cliente/servidor:

- Basada en servidores de aplicaciones:
  - Permite el uso de aplicaciones en redes WAN e Internet.
  - Las aplicaciones se instalan en los servidores de aplicación
  - a través de internet por el navegador, los clientes solicitan información a los servidores de aplicación, sin que requieran ninguna administración
  - Los servidores de aplicación son los que realizan la gestión y proporcionan al cliente a través de su navegador la información solicitada