

¿Cuál es el elemento fundamental para una empresa?  
¿Por qué?



# Bases de Datos

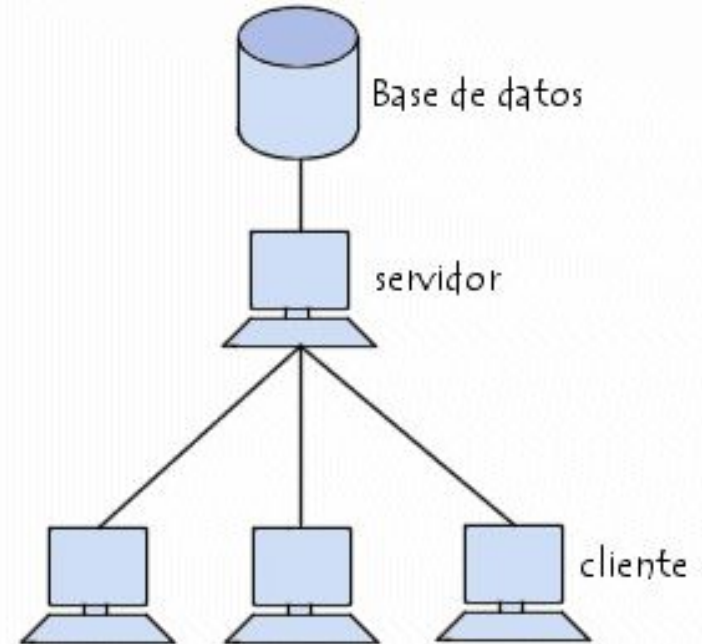
Introducción



¿Qué es una base de datos?

# ¿Qué es una base de datos?

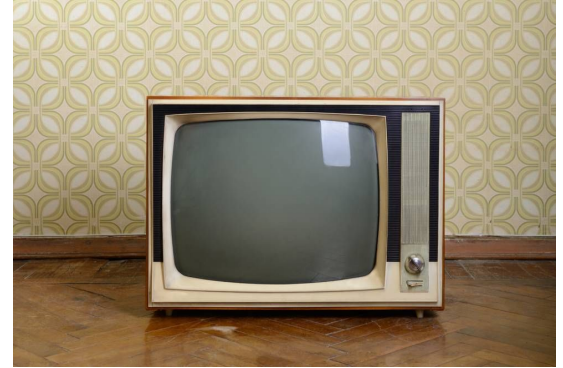
Una **base de datos** es un “almacén” **que** nos permite guardar grandes cantidades de información de forma organizada para **que** luego podamos encontrar y utilizar fácilmente.



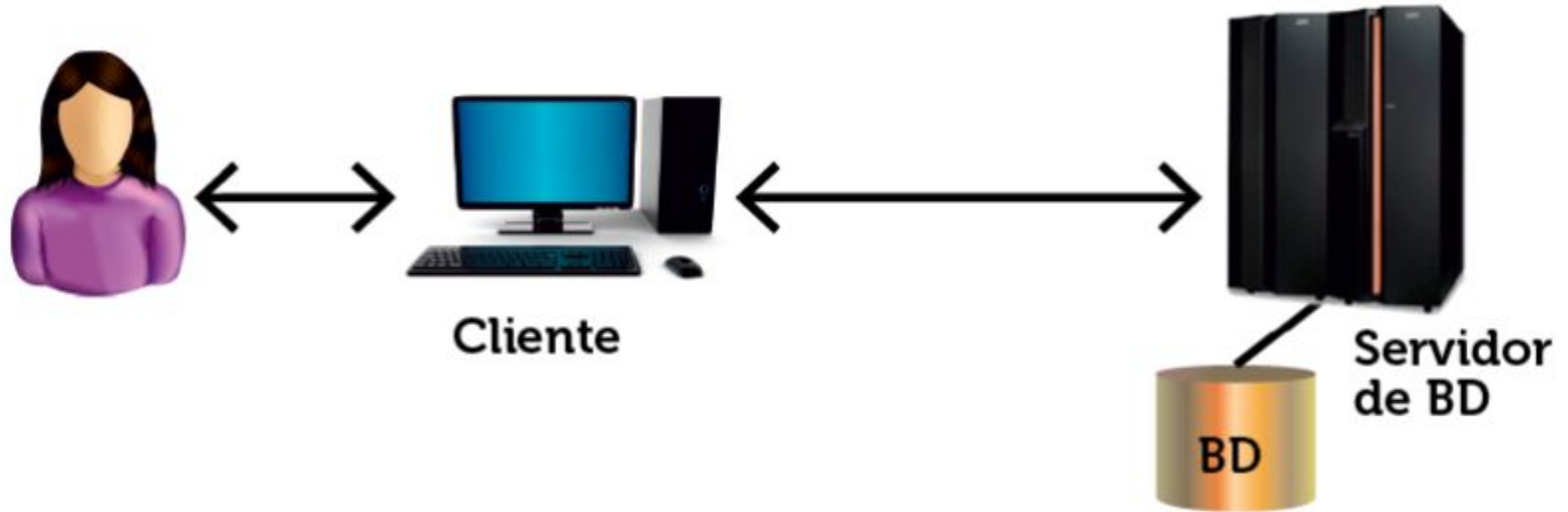
¿Cuándo usamos una base de datos?

# Ejemplos de bases de datos

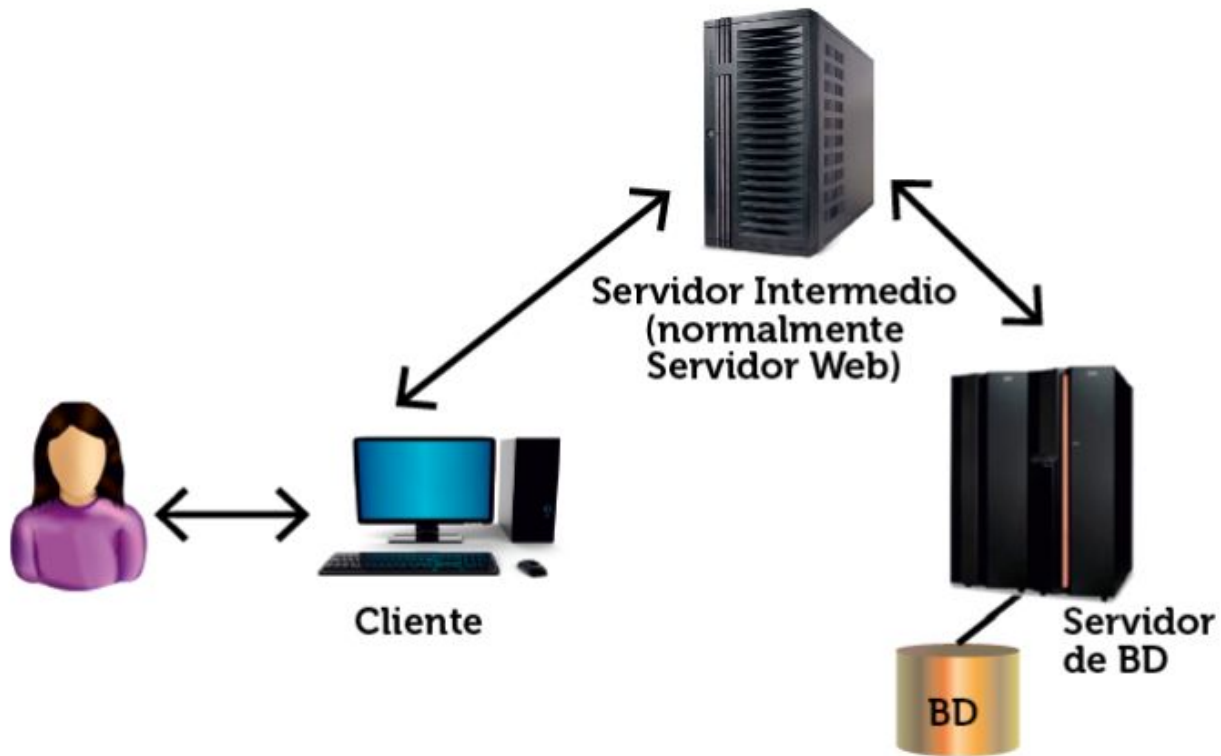
- Cuando elegimos un canal en la tele
- Al utilizar navegar por la agenda del móvil
- Cuando operamos en el cajero automático
- Cuando acudimos a la consulta del médico
- Al inscribirnos en un curso
- Si utilizas un GPS
- Cuando cogemos entradas para un concierto
- Cuando consultamos cualquier información en Internet
- Al hacernos una cuenta en Steam



# Función de la BD



# Función de la BD





# Tecnologías

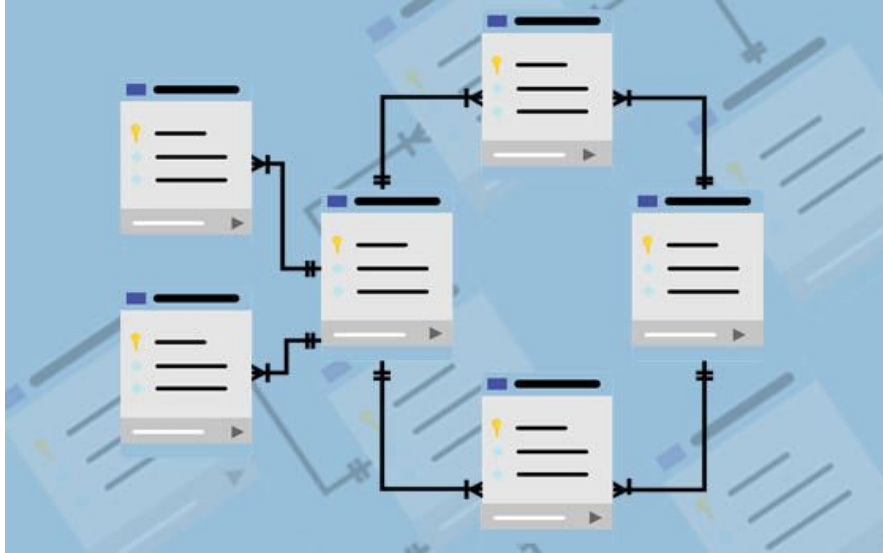
- **SQL:** Structured Query Language / Lenguaje de Consulta Estructurada:
  - **Lenguaje declarativo** de acceso a bases de datos que combina construcciones del álgebra relacional y el cálculo relacional.
  - Originalmente desarrollado en los '70 por IBM en su Research Laboratory de San José a partir del cálculo de predicados creado por Codd.
  - **Lenguaje estándar**
  - **MySQL**
- **NoSQL:**
  - **Not-Only-SQL** (*No-Sólo-SQL*)
  - Diseñadas para almacenar grandes cantidades de información
  - Menos restricciones
  - **MongoDB**

# SQL: Características



- **El Lenguaje de Definición de Datos (LDD)**
  - Proporciona comandos para la creación, borrado y modificación de esquemas relacionales
- **El Lenguaje de Manipulación de Datos (LMD)**
  - Basado en el álgebra relacional y el cálculo relacional permite realizar consultas y adicionalmente insertar, borrar y actualizar de tuplas
  - Ejecutado en una consola interactiva
  - Embebido dentro de un lenguaje de programación de propósito general
- Definición de vistas
- Autorización: Definición de usuarios y privilegios
- Integridad de datos
- Control de Transacciones

# Niveles de Diseño




- **Estructura:** Crear la estructura de **tablas** y sus relaciones
- **Información:** Insertar, actualizar, buscar y eliminar la información contenida dentro de las tablas

\*Nada es inamovible

# Parte 1: Tablas



Tabla 'film':

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo
1	<b>id</b> 	int(11)			No
2	<b>title</b>	varchar(100)	latin1_swedish_ci		No
3	<b>director</b>	varchar(100)	latin1_swedish_ci		No
4	<b>year</b>	varchar(4)	latin1_swedish_ci		No
5	<b>description</b>	varchar(500)	latin1_swedish_ci		No

# Parte 2: Información

Contenido de la tabla 'film':

id	title	director	year	description
1	Pulp Fiction	Quentin Tarantino	1994	Jules y Vincent son dos asesinos a sueldo que trab...
2	Terminator 2: El Juicio Final	James Cameron	1991	Las máquinas, en su intento fallido por matar a la...
3	E.T.	Steven Spielberg	1982	A troubled child summons the courage to help a fri...
4	Zodiac	David Fincher	2007	In the late 1960s/early 1970s, a San Francisco car...

# Las tablas contienen la información:

- Las tablas son el continente
- La información es el contenido

id	title	director	year	description
1	Pulp Fiction	Quentin Tarantino	1994	Jules y Vincent son dos asesinos a sueldo que trab...
2	Terminator 2: El Juicio Final	James Cameron	1991	Las máquinas, en su intento fallido por matar a la...
3	E.T.	Steven Spielberg	1982	A troubled child summons the courage to help a fri...
4	Zodiac	David Fincher	2007	In the late 1960s/early 1970s, a San Francisco car...

# Tablas

# SQL: Tablas


Sentencia SQL para la creación de la Tabla 'film':

```
CREATE TABLE `film` (  
    id int NOT NULL,  
    title varchar(100),  
    director varchar(100),  
    year varchar(4),  
    description varchar(500),  
    PRIMARY KEY (ID)  
)
```



# SQL: Tablas

Tabla 'film' creada (sin información, sólo estructura):

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo
1	<b>id</b> 	int(11)			No
2	<b>title</b>	varchar(100)	latin1_swedish_ci		No
3	<b>director</b>	varchar(100)	latin1_swedish_ci		No
4	<b>year</b>	varchar(4)	latin1_swedish_ci		No
5	<b>description</b>	varchar(500)	latin1_swedish_ci		No

# SQL: Tablas

Operaciones sobre la estructura:

- **CREATE TABLE:** Crear tablas, campos de índices
- **ALTER TABLE:** Modificar tablas al agregar o cambiar la definición de los campos campos.
- **DROP TABLE:** Eliminar tablas e índices



Información (datos)



# SQL: Información

Contenido de la Tabla 'film':

id	title	director	year	description
1	Pulp Fiction	Quentin Tarantino	1994	Jules y Vincent son dos asesinos a sueldo que trab...
2	Terminator 2: El Juicio Final	James Cameron	1991	Las máquinas, en su intento fallido por matar a la...
3	E.T.	Steven Spielberg	1982	A troubled child summons the courage to help a fri...
4	Zodiac	David Fincher	2007	In the late 1960s/early 1970s, a San Francisco car...

# SQL: Información

Operaciones sobre la información:

- **SELECT**: buscar entre los datos
- **INSERT**: añadir nuevos datos
- **UPDATE**: actualizar los datos existentes
- **DELETE**: eliminar todos o parte de los datos



# SQL: Tablas y Relaciones

actors.sql

id	first_name	last_name	gender	film_count
933	Lewis	Abermathy	M	1
2547	Andrew	Adamson	M	1
2700	William	Addy	M	1
2898	Seth (I)	Adkins	M	1
2925	Charles (I)	Adler	M	1
3226	Casey	Affleck	M	1
4306	Shigekazu	Aida	M	1
4856	Julliet	Akinyi	M	1
6005	Henri	Alciatore	M	1
6990	Dean	Alexandrou	M	1
7124	Jim	Alfonso	M	1
7817	Carl (I)	Allen	M	1
7979	Ivan (I)	Allen	M	1
8161	Richard (XV)	Allen	M	1
8409	Anthony	Allison	M	1
8412	Bill (I)	Allison	M	1
8426	Frankie J.	Allison	M	1
9184	Steve	Altes	M	1
9275	Walter	Altman	M	1
11489	Dion	Anderson	M	1
11547	George (IV)	Anderson	M	1
11590	J. Todd	Anderson	M	1
11865	Scott G.	Anderson	M	1
12043	Kris	Andersson	M	1
12508	David (I)	Andrews	M	2
12987	Jack (I)	Angel	M	1
13553	Joey	Ansah	M	1

movies.sql

id	name	year	rank
10920	Aliens	1986	8.2
17173	Animal House	1978	7.5
18979	Apollo 13	1995	7.5
30959	Batman Begins	2005	NULL
46169	Braveheart	1995	8.3
109093	Fargo	1996	8.2
111813	Few Good Men, A	1992	7.5
112290	Fight Club	1999	8.5
116907	Footloose	1984	5.8
124110	Garden State	2004	8.3
130128	Godfather, The	1972	9
147603	Hollow Man	2000	5.3
167324	JFK	1991	7.8
176711	Kill Bill: Vol. 1	2003	8.4
176712	Kill Bill: Vol. 2	2004	8.2
192017	Little Mermaid, The	1989	7.3
194874	Lost in Translation	2003	8
207992	Matrix, The	1999	8.5
210511	Memento	2000	8.7
224842	Mystic River	2003	8.1
237431	O Brother, Where Art Thou?	2000	7.8
238072	Ocean's Eleven	2001	7.5
238695	Office Space	1999	7.6
254943	Pi	1998	7.5
256630	Pirates of the Caribbean	2003	NULL
257264	Planes, Trains & Automobiles	1987	7.2
267038	Pulp Fiction	1994	8.7

roles.sql

actor_id	movie_id	role
16844	10920	Lydecker
36641	10920	Russ Jorden
42278	10920	Cpl. Dwayne Hicks
144260	10920	Doctor
204493	10920	Timmy Jorden
204719	10920	Bishop
213646	10920	Lt. Gorman
240277	10920	Pvt. Spunkmeyer
272557	10920	Power Loader Operator
305705	10920	Sgt. Apone
306790	10920	Van Leuwen
310030	10920	Al Simpson, Colony Officer
366173	10920	Private Hudson
378476	10920	Insurance Man
394516	10920	Carter Burke
405572	10920	Pvt. Drake
408215	10920	Pvt. Frost
452549	10920	Pvt. Wierzbowski
474375	10920	Pvt. Crowe
476429	10920	Alien Warrior
583693	10920	Cocooned Woman
583735	10920	ECA Representative
596318	10920	Ann Jorden
635153	10920	Pvt. Vasquez
651551	10920	Rebecca 'Newt' Jorden
653997	10920	Cpl. Mira Ferro
661430	10920	Amanda Ripley



# Bases de Datos

Almacenamiento de la Información



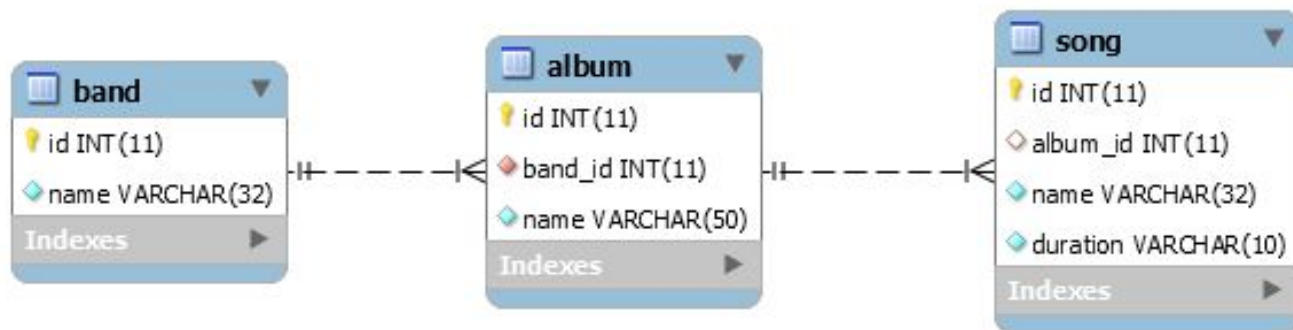
# Utilidad

- Necesidad de manejar grandes cantidades de datos
- Esos datos deben poder ser accedidos por diferentes personas/sistemas simultáneamente: concurrencia
- Existen muchos tipos de bases de datos
  - Sistemas propietarios
  - Archivos (más simples)





# Nomenclatura



# Nomenclatura

**Campos o Atributos**

<b>Nombre</b>	<b>Nacionalidad</b>	<b>Institución</b>
Sofia	España	O.N.U.
John	EE. UU.	O.M.S.
Jacob	Francia	N.A.S.A.
Sebastian	Chila	UNASUR

**Registros o Tuplas**

# Nomenclatura

The diagram shows a database table with two columns: 'IdEquipo' and 'Nombre'. The table contains five rows of data. A red box highlights the entire table structure, labeled 'Campo' (Field). Another red box highlights the second row, labeled 'Registro' (Record). A red arrow points from the cell containing 'Imparables' to the label 'Dato' (Data).

IdEquipo	Nombre
0	LocosDelBalón
1	18 Fútbol Club
2	Imparables
3	Siniviedo FC
4	Los mejores
5	Luchadores

**Campo**

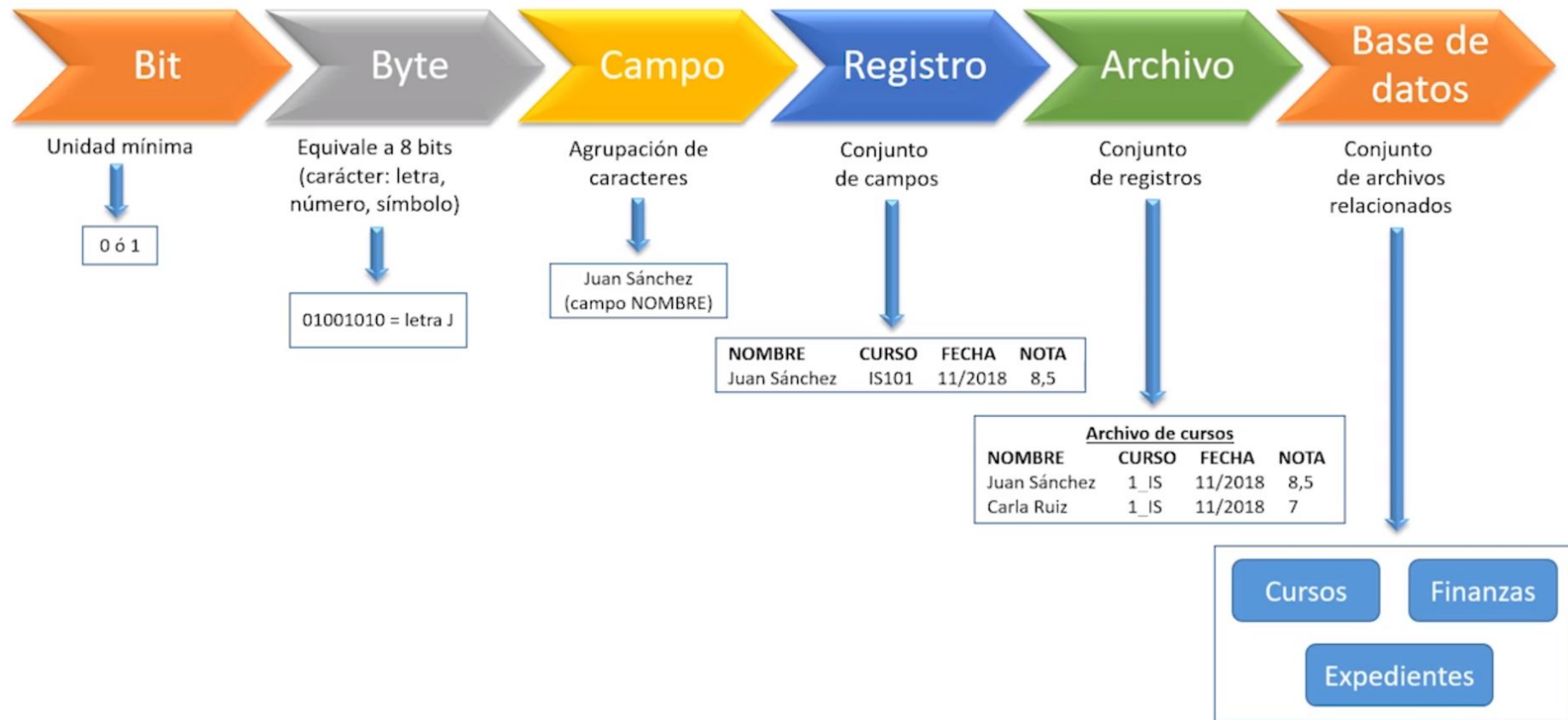
**Registro**

**Dato**

# Jerarquía de datos

- Para poder ser manejados de forma más adecuada, la información se agrupa en conjuntos más pequeños.
- Existen 6 Niveles de Jerarquía:

BIT 0, 1	El bit es la unidad más pequeña de información. Su valor es 0 o 1.
BYTE (8 bits) 11000001 = A	8 bits se combinan para formar un caracter (byte).
CAMPO Pedro Gómez	Un campo es el dato al que se hace referencia dentro de un sistema de información. Está formado por un grupo de bytes.
REGISTRO Pedro Gómez, Gran Vía 32, etc.	Los campos se agrupan para formar registros. Los datos de los registros forman parte de una entidad común.
ARCHIVO Pedro Gómez, Gran Vía 32, etc.; María Pérez, Avda. España 2, etc	Los archivos son conjuntos de registros de un mismo tipo.
BASE DE DATOS Empleados, clientes, usuarios, etc.	Las bases de datos son conjuntos de archivos que representan la información de un sistema.



# Sistemas de Información (SI)

- Las bases de datos de una empresa, son parte del SI de esa empresa
- Están formados por:
  - Las personas
  - Los datos
  - Las actividades que realizan
  - Recursos (ordenadores, papeles, teclados, lápices)



# Sistemas de Información (SI)

Realizan 4 actividades básicas



1. **ENTRADA**: introducción de datos en el sistema.
2. **PROCESAMIENTO** O TRANSFORMACIÓN DE LOS DATOS: usar herramientas para relacionar, resumir o concluir los datos. Los datos una vez procesados son almacenados.
3. **SALIDA** DE LA INFORMACIÓN.
4. **RETROALIMENTACIÓN**: los resultados obtenidos se ingresan de nuevo en el sistema para ser procesados de nuevo, esto sirve para corregir la etapa de entrada.

# Sistemas de Información (SI)

Los sistemas de información cumplen tres objetivos básicos dentro de las organizaciones:

1. **Automatizar los procesos operativos:** su función principal consiste en procesar transacciones como pagos, cobros, entradas, salidas, etc.
2. **Proporcionar información para facilitar la toma de decisiones:** sirven para facilitar el seguimiento, control y toma de decisiones por parte de los gerentes. Utiliza información tanto externa como interna.
3. **Conseguir ventajas competitivas a través de su implantación y uso.**



# Almacenamiento: Archivos / Ficheros:

- Los datos se almacenan en archivos tradicionalmente
- Les dan persistencia a los datos
- En principio una aplicación usará un único archivo (pero pueden ser varios)
- Los archivos se pueden transportar de un equipo a otro
- La capacidad de almacenamiento no tiene límite

```
<note>  
  <date>2017-11-08</date>  
  <hour>08:30</hour>  
  <to>Raj</to>  
  <from>Ravi</from>  
  <body>Meeting at 8am.</body>  
</note>
```

# Almacenamiento: Archivos / Ficheros:

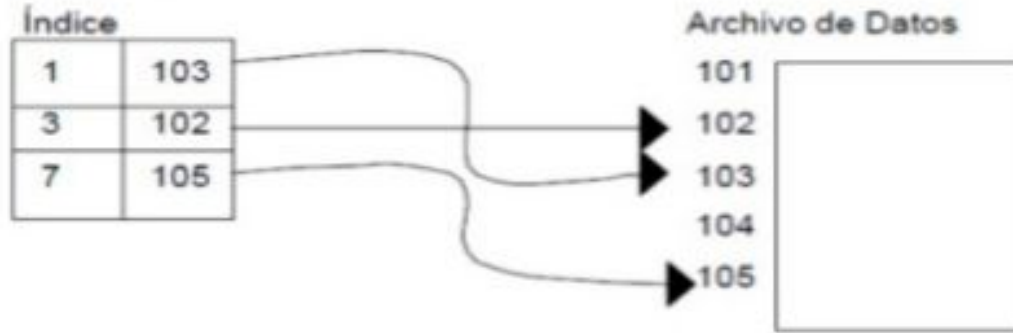
- Está formado por registros lógicos, y cada registro tiene diferentes campos
- Cada campo es un apartado de un dato
- El acceso es más lento (no hay un motor)
- Cuanto mayor sea el archivo, más tardará en consultarse
- Puede dar redundancias (un mismo dato almacenado varias veces)
- Puede dar contradicciones

ID	First Name	Last Name	Email	Year of Birth
1	Peter	Lee	plee@university.edu	1992
2	Jonathan	Edwards	jedwards@university.edu	1994
3	Marilyn	Johnson	mjohnson@university.edu	1993
6	Joe	Kim	jkim@university.edu	1992
12	Haley	Martinez	hmartinez@university.edu	1993
14	John	Mfume	jmfume@university.edu	1991
15	David	Letty	dletty@university.edu	1995

# Acceso a la Información: Índices

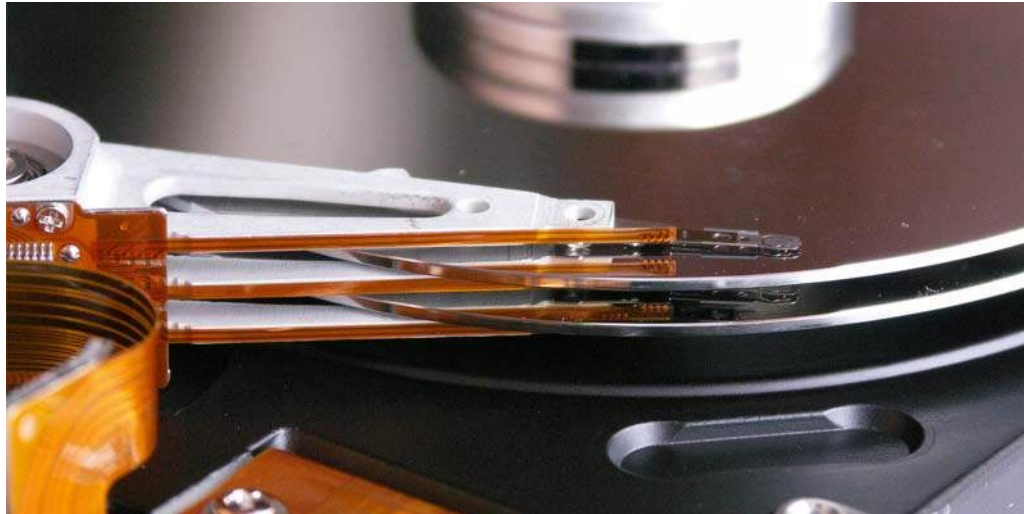
- Se crea un índice ordenado que guarda la información del orden en el que están guardados los registros dentro del archivo.
- Para acceder a los registros, hay que buscar secuencialmente la clave en el índice que se ha creado previamente.
- Una vez se ha localizado el índice, este lleva asociado la dirección real del registro en el archivo. Con esta dirección se puede acceder al registro directamente.
- **Ventaja:** permite el acceso directo a los registros.
- **Inconveniente:** necesita más espacio de almacenamiento ya que necesita guardar la tabla de índices.

# Acceso a la Información: Índices



# Soporte Direccional: Disco

Los discos son soportes direccionables donde se almacenan archivos. Los discos se dividen en unidades mínimas llamadas bloques. Además, permiten la reescritura de dichos bloques. Por otro lado, permiten el acceso directo a cualquier bloque de información.





# Bases de Datos

Sistemas Gestores de Bases de  
Datos (SGBDs)



# La BD y el SGBD

- BD: conjunto de datos almacenados de forma estructurada y organizada en un soporte de informático y que es manejado por un ordenador.
- La información
  - se almacena en archivos independientes integrados en la base de datos
  - puede ser compartida por distintos usuarios al mismo tiempo
  - puede ser utilizada para diferentes fines



# ¿Cómo logramos todo esto? El SGBD

Antes de la aparición de las aplicaciones informáticas, las empresas tenían como únicas herramientas de gestión de datos a los cajones, carpetas y fichas en las que se almacenaban los datos. En este proceso manual, el tipo requerido para manipular estos datos era enorme. Sin embargo el proceso de aprendizaje era relativamente sencillo ya que se usaban elementos que el usuario reconocía perfectamente.

Con la aparición de las nuevas tecnologías, ha aparecido el dato digital como elemento pivotal de las comunicaciones digitales, por lo que es necesario una forma para gestionar todos esos datos, una forma de almacenar, insertar, buscar y eliminar datos que sea sencilla y asequible para todo el mundo.

De la necesidad de poder manejar los datos de forma sencilla salen los sistemas gestores de bases de datos.



# Sistemas Gestores de Bases de Datos (SGBDs)

- Es lo que utilizamos para interactuar con la BD
- Son los programas que permiten la creación, administración y gestión de la información en una base de datos.
  - Maneja grandes volúmenes de información.
  - Nos proporciona:
    - Un motor de consultas
    - Una interfaz para interactuar con las tablas y sus datos



# SGBDs Comerciales

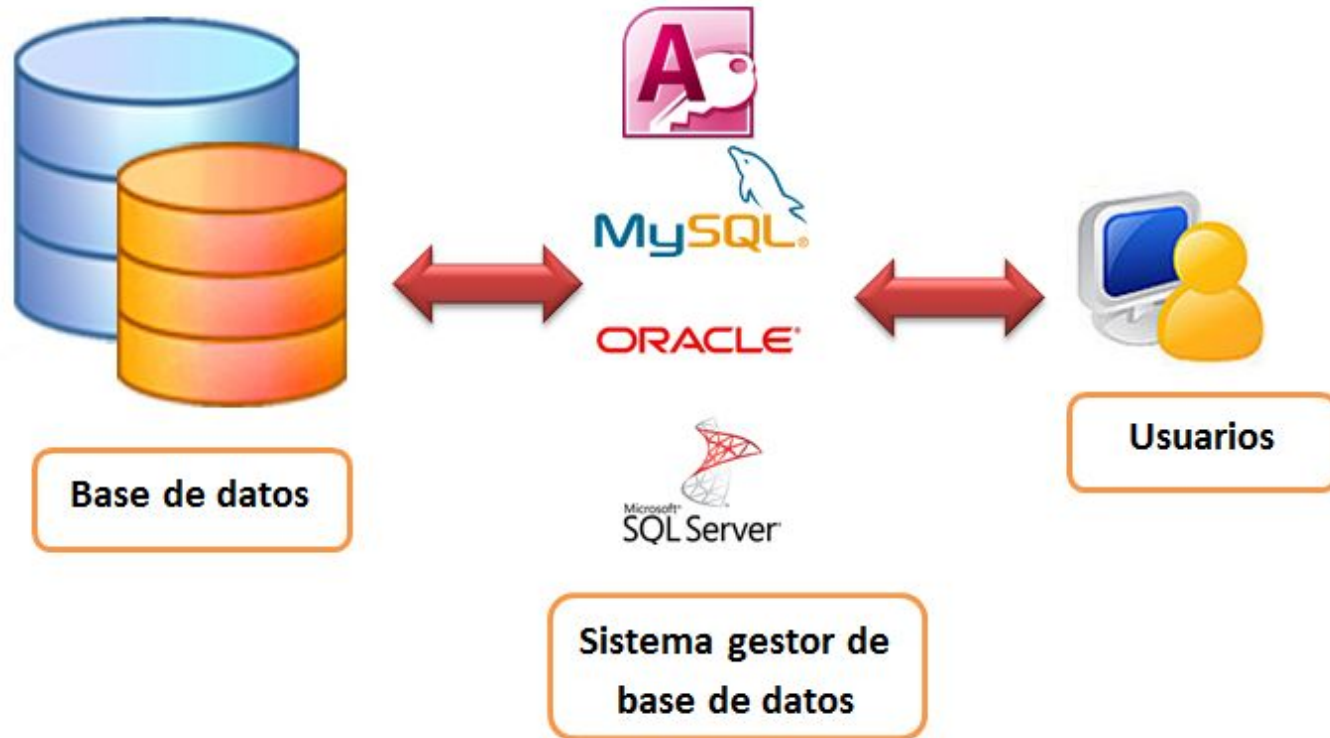
ORACLE®



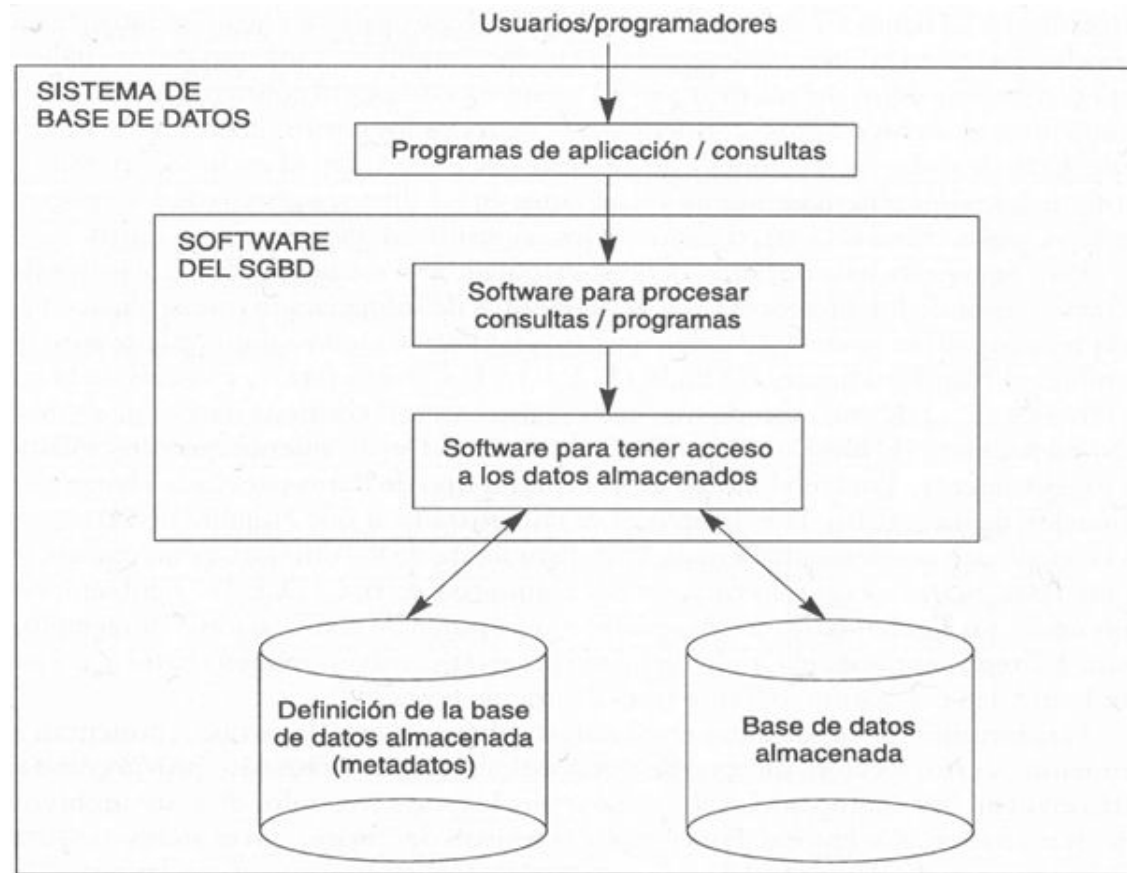
# SGBD: ¿Cómo funciona?

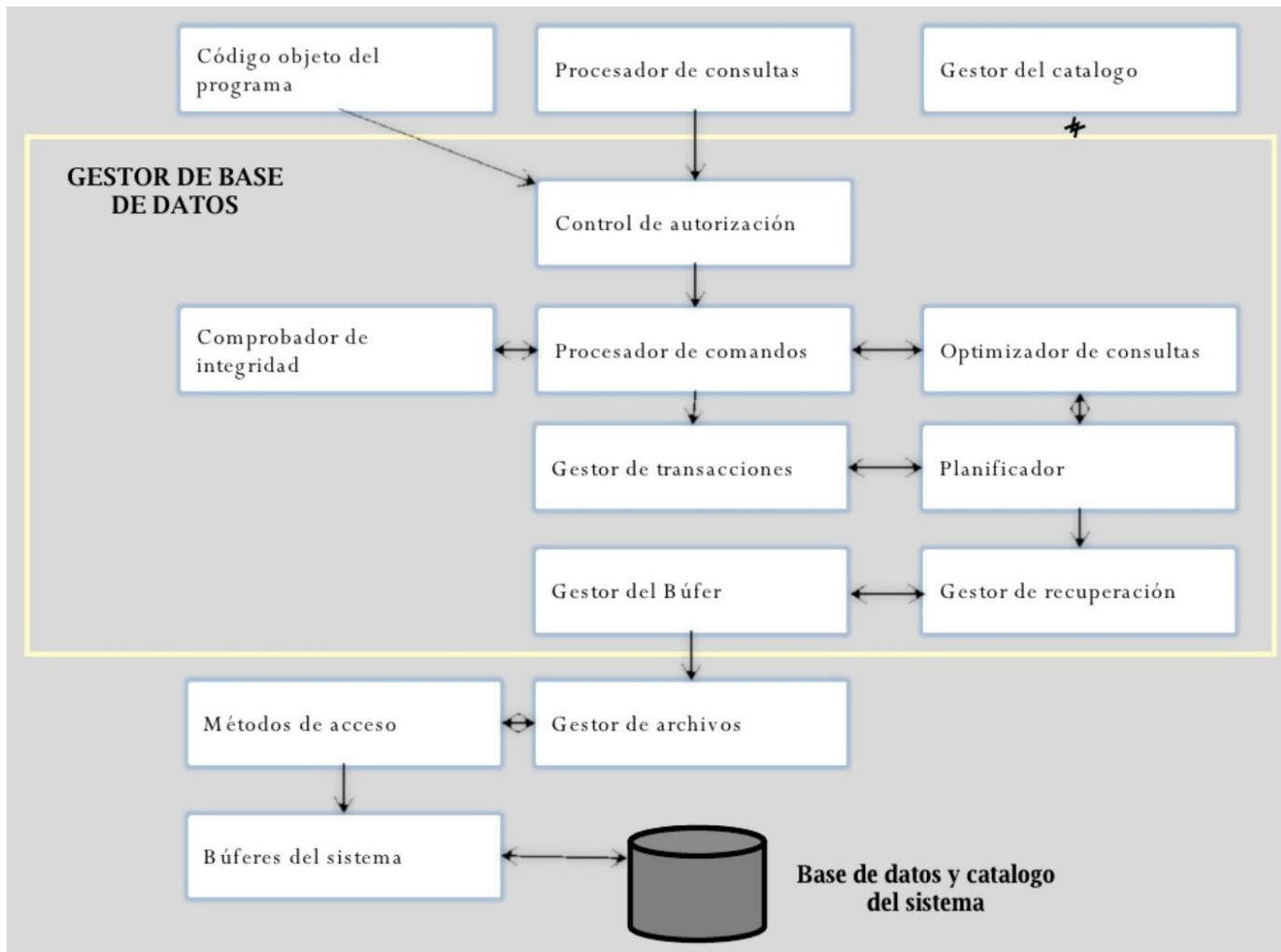
- En los SGBD, **ni los programadores, ni los usuarios necesitan saber cómo está distribuida y organizada la información**
- Quien maneja la información es un conjunto de **procedimientos** que forman parte del SGBD.
- Una de las funciones de estos procedimientos es proporcionar **diferentes niveles de abstracción de la información dependiendo del usuario que la esté manejando.**
- **Para la mayoría de los usuarios, se ocultan los detalles de la forma y lugar en el que están almacenados los datos y cuáles son los procedimientos de recuperación y actualización de la información.**
- Un SGBD también **se encarga de conectar e implementar los distintos niveles de la arquitectura** de la base de datos.

# SGBDs



# SGBD





**Control de  
autorización**

Comprueba que el usuario tiene la autorización para llevar a cabo la operación

**Procesador de  
comandos**

Luego de comprobar la autorización, se pasa el control al procesador de comandos

**Comprobador de  
integridad**

Verifica que la operación solicitada satisface todas restricciones de integridad necesarias

**Optimizador de  
consultas**

Determina una estrategia optima para la ejecución de las consultas

**Gestor de  
transacciones**

Realiza el procesamiento requerido para las operaciones que recibe de las transacciones

**Planificador**

Garantiza que las operaciones concurrentes en la base de datos puedan llevarse a cabo sin entrar en conflicto unas con otras.

**Gestor de recuperación**

Garantiza que la base de datos permanezca en un estado coherente cuando se produzcan fallos.

**Gestor de búfer**

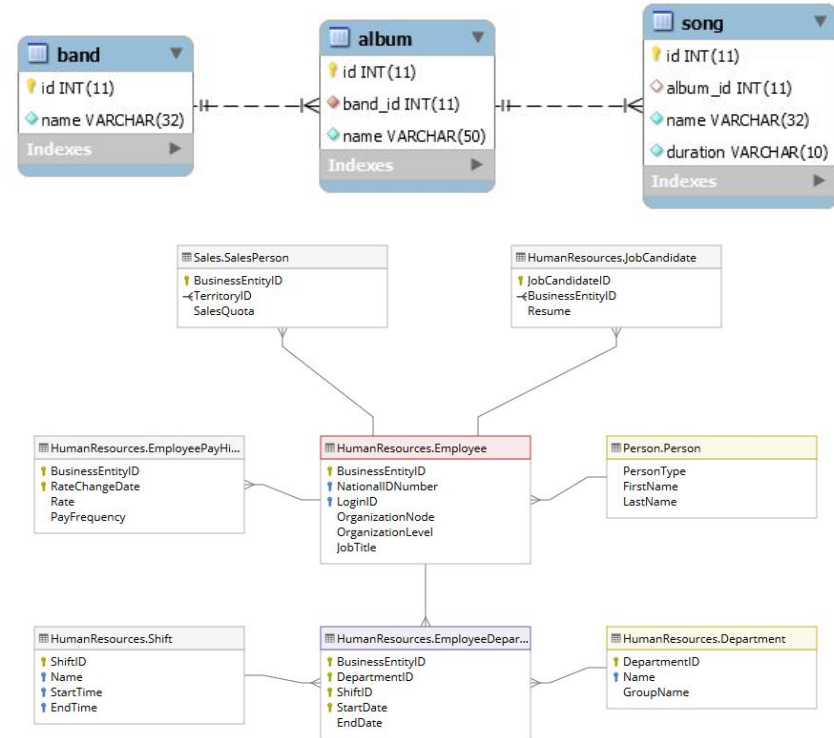
Responsable de la transferencia de datos entre la memoria principal y el almacenamiento secundario.

# Definición: Esquema de una base de datos

Describe la estructura de una base de datos, en un lenguaje formal soportado por un sistema de gestión de base de datos (DBMS).

En una base de datos relacional, el esquema define:

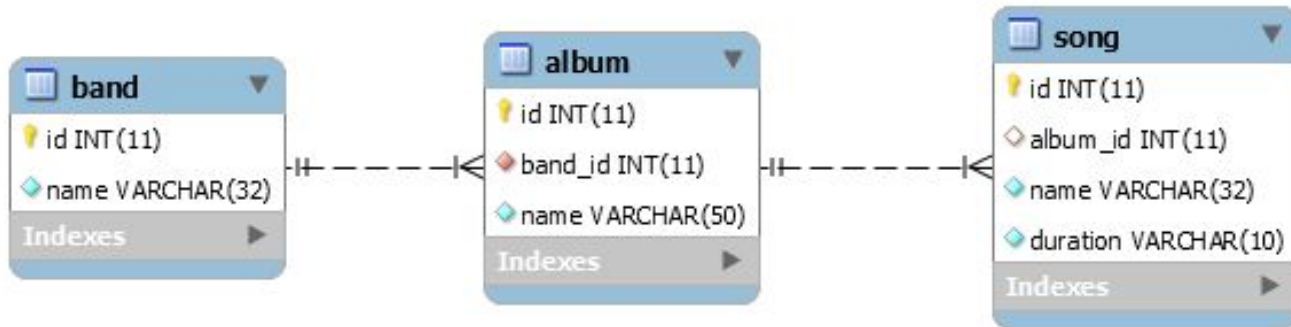
- Tablas
- Los campos de cada tabla
- Las relaciones entre cada campo y cada tabla.





# SGBD: Propósito

- Debe existir una **separación entre los datos y los programas** de aplicación **que los utilizan**.
- Permitir a los usuarios el manejo de múltiples **vistas**.
- Facilitar el uso de un catálogo para **almacenar el esquema de la base de datos**.





# Sistema Gestor de Bases de Datos (SGBD)



# SGBD: Características

**INDEPENDENCIA DE DATOS:** cualquier programa puede acceder a ellos, no dependen de un programa en concreto.

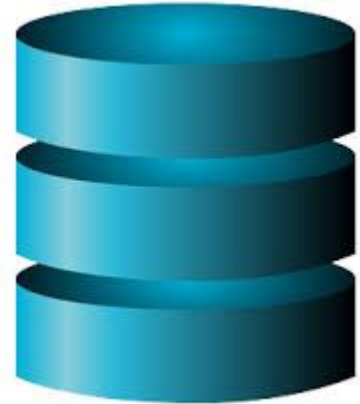
**REDUCCIÓN DE REDUNDANCIA:** la redundancia es la duplicación de datos. Si reducimos al máximo la redundancia, conseguimos un mayor aprovechamiento del espacio y evitamos la inconsistencia de datos (evita datos contradictorios).

**SEGURIDAD:** un buen SGBD debe permitir tener un control sobre la seguridad de los datos.

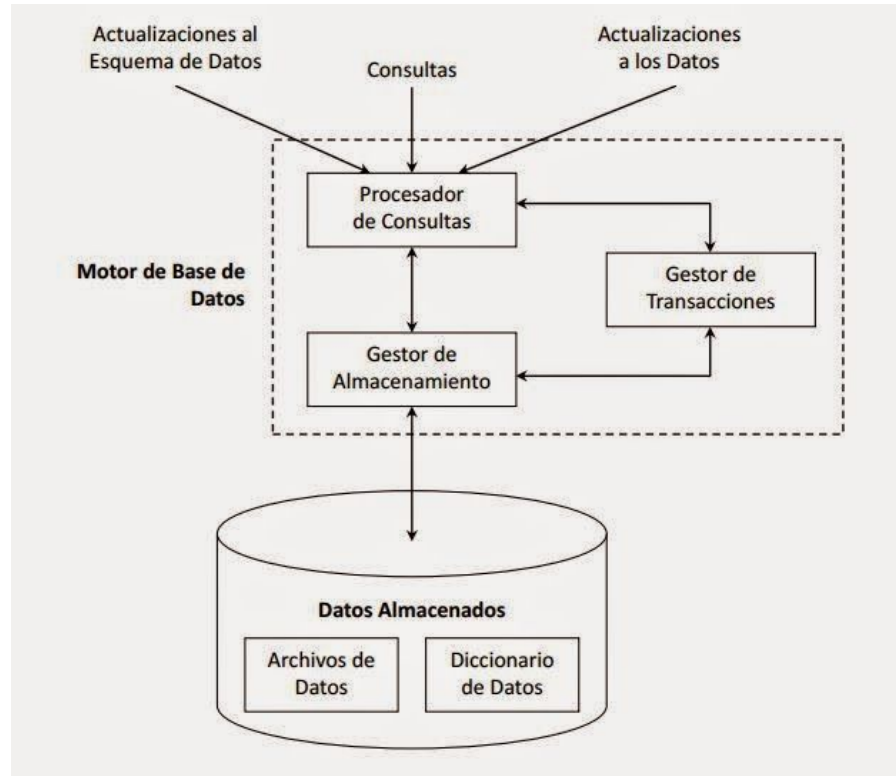
# SGBD: Sistema de Control de la Base de Datos (SCBD)

Es el encargado de transformar los distintos datos requeridos por los programas en registros físicos a leer.

Realiza la petición al sistema y si la información está disponible, la transfiere al área de trabajo del programa que la solicitó.



# SGBD: Motor de consultas



# SGBD: Admin del Sistema (SysAdmin)

Es el encargado de **organizar y controlar los recursos del sistema**. Las funciones del administrador del sistema son las siguientes:

- Utilizando el **lenguaje de definición del SGBD**, definir el esquema conceptual de la base de datos.
- Gestionar los **permisos** de acceso a los usuarios, controlando de esta manera los distintos accesos que se realizan a la base de datos.
- Ante un posible **fallo de la base de datos**, definir las estrategias de recuperación de la información



# SGBD: Tareas del SysAdmin

- **Configurar e instalar el hardware en el que se instalará el servidor o servidores de bases de datos.** Implica decidir qué hardware es el óptimo e implementar memoria, discos e instalación de red necesaria. A veces hay un segundo administrador (el del sistema) que se encarga de ello pero basándose en las necesidades expuestas por el DBA.
- **Configurar e instalar el Sistema Operativo.** También la gestión mínima del sistema operativo para que la base de datos funcione correcta y rápidamente. Nuevamente, puede ser una tarea coordinada con un administrador del sistema.
- **Instalación y mantenimiento del SGBD.** Seleccionando la más adecuada forma de instalación y configurando lo necesario para su óptimo rendimiento, acorde con las necesidades. También es parte de esta tarea el control de las actualizaciones del sistema de bases de datos que sean necesarias.
- **Crear y configurar las base de datos.** Creación de la estructura interna de la base de datos (tablas, usuarios, permisos, vistas,...). Es otra de las tareas más habitualmente relacionadas con el DBA y la primera fase (y la más crítica) en la administración de una base de datos.
- **Control de los usuarios y los permisos.** En definitiva establecer las políticas de seguridad ante accesos no cualificados, tan imprescindibles en toda base de datos.
- **Gestión de la seguridad.** Complemente la tarea anterior para proteger al sistema ante ataques de terceros o condiciones que comprometan la estabilidad del sistema.
- **Monitorizar y optimizar el rendimiento de la base de datos.** Un DBA debe detectar los cuellos de botella del sistema y actuar en consecuencia. Esto incluye optimizar las instrucciones, crear estructura de acceso veloces o mejorar la infraestructura hardware y software.
- **Realizar tareas de copia de seguridad y recuperación.** Quizá la tarea más crítica. Consiste en realizar acciones para en caso de catástrofe poder recuperar todos los datos

# SGBD: Objetivos: BD rápida, eficaz y polivalente

- Los datos deben ser independientes de los programas.
- Reducir los datos repetidos a los mínimos posibles (redundancia mínima de la información).
- Suministrar herramientas de seguimiento para las operaciones realizadas sobre la base de datos y la información sobre los usuarios que las realizan (fecha, hora de conexión, usuarios, operaciones realizadas en cada sesión, los datos que han sido modificados, etc.).
- Proporcionar mecanismos de búsqueda y obtención de información versátiles y sencillos a los usuarios



# SGBD: Objetivos: BD rápida, eficaz y polivalente

- Proteger la base de datos contra los accesos no autorizados.
- Controlar que la información de la base de datos sea íntegra. Por tanto, hay que controlar los posibles fallos de hardware, inserción de datos incorrectos o inválidos, procesos no finalizados, etc.
- Dotar de mecanismos que permitan realizar copias de seguridad de la información



# SGBD: Objetivos: BD rápida, eficaz y polivalente

- Conseguir tiempos de respuesta óptimos:
  - El tiempo de respuesta es el periodo de **tiempo que transcurre desde que el usuario termina de realizar una consulta al sistema hasta que comienza a recibir una respuesta.**
- Mecanismos que sean capaces de solucionar los problemas que pueda plantear la concurrencia de usuarios.
  - Si la base de datos está **compartida por varios usuarios** que trabajan en distintas terminales, puede darse el caso de que intenten actualizar la información de forma concurrente (en el mismo instante de tiempo) sobre el mismo registro.

# SGBD: Componentes: 1 - Módulo del Núcleo

Este módulo incluye un conjunto de programas que se encargan de coordinar y controlar todo el funcionamiento del SGBD.

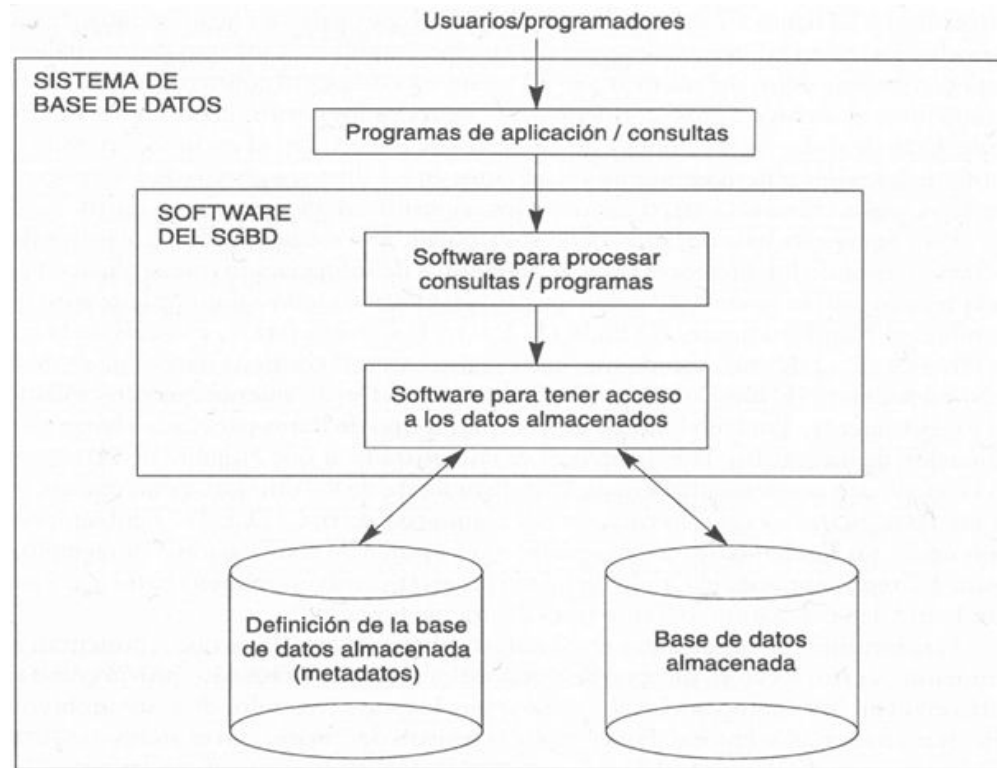
- Son programas transparentes al usuario.
- Controlan la integridad y seguridad.
- Facilitan la independencia de los datos.
- Gestionan el diccionario de datos.
- Implementan las funciones de comunicación entre **niveles**.
- Proporcionan el soporte necesario para los programas de utilidad y los lenguajes.

# SGBD: Componentes: 2 - Lenguajes

- SGBD proporciona diversos lenguajes que permiten definir y manejar los datos de la base de datos.
- Cada SGBD tiene su propia estructura y organización, pero todos tienen una característica común y es que ofrecen al administrador y a los usuarios dos lenguajes:
  - **DDL** (lenguaje de descripción de datos): se utiliza para definir el esquema conceptual y los subesquemas externos de la base de datos. También se utiliza para establecer parte de la seguridad de la base de datos, ya que permite asignar derechos sobre las operaciones que pueden realizar los usuarios. utiliza para crear tablas y definir tipos.
  - **DML** (lenguaje de manipulación de datos): se utiliza para gestionar la información de la base de datos. Este lenguaje permite añadir, eliminar, modificar registros y recuperar información de forma estructura de la base de datos. Se utiliza para insertar registros y realizar consultas.

# SGBD

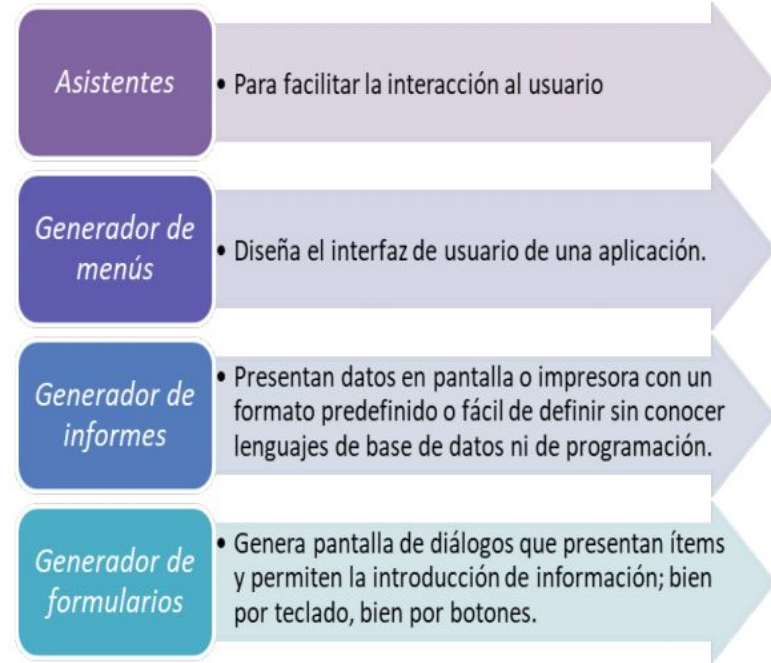
Los SGBD tienen la capacidad de procesar peticiones de manipulación de datos que **se envían desde programas escritos en otros lenguajes de programación.**



# SGBD: Componentes: 3 - Utilidades

Los SGBD incluyen aplicaciones que facilitan el trabajo tanto a usuarios como a programadores.

- Incluyen una interfaz que es fácil de comprender.
- Su diseño se basa en menús que permiten guiar al usuario para conseguir su objetivo final.



# SGBD: Componentes: 4- Diccionario de Datos

- Es un almacén integrado en el que se guarda toda la información referente a la descripción, gestión e implantación de la base de datos.
- El diccionario de datos contiene la descripción de los esquemas interno, conceptual y externo, las tablas de usuarios con sus respectivos permisos, los programas de aplicación utilizados, las operaciones que realizan los usuarios en el sistema, la información de otros recursos implicados en el sistema, etc.
- Están estructurados en capas y si la complejidad de la base de datos lo requiere, dichas capas se pueden subdividir recursivamente.
- Se divide en 3 capas

# SGBD Comerciales



# MySQL



- Desarrollado en los años 90
- De uso muy muy común
- Sirve para cualquier lenguaje
- Todos los SOs. El más común es Linux
- Muy buen rendimiento
- Software de código abierto
- Tiene múltiples motores de almacenamiento.
- No puedes cancelar una consulta en medio de la ejecución
- Usa la licencia pública general GNU, por lo que su uso es totalmente gratuito
- Todo tipo de empresas

# Oracle

- Oracle Corp.
- Versión de pago muy superior a MySQL
- Años 80
- Ecosistema Java
- El más complejo, el más completo y el más potente
- Más seguridad
- Ecosistema Oracle para empresas
- Oracle soporta y usa XML
- Más para grandes empresas
- Soporte con muchos S.O.s: Windows, Linux, Solaris...
- Para grandes empresas



# SQL Server



- Microsoft
- Transact-SQL (TSQL)
- Orientado a .NET
- Muy completo
- Fácil de usar. Muchos menús
- De pago
- 120-180 días de prueba
- Complejidad y eficiencia
- Tiene un motor de almacenamiento propio de Microsoft
- Bastantes opciones de seguridad
- Permite cancelar una consulta en medio de su ejecución
- Requiere ecosistema windows (Windows Server)
- Se puede instalar en Linux
- Muy caro
- Requiere comprar licencias para múltiples Bases de Datos
- Pequeñas y medianas empresas

# Arquitectura ANSI



# Independencia de Datos

Permite que se pueda modificar el nivel de un sistema sin que se vean afectados otros niveles, concretamente el nivel superior al modificado.

1- **INDEPENDENCIA LÓGICA:** podemos modificar el esquema conceptual sin alterar el resto de esquemas (externo, interno); es decir, nos permite ampliar o reducir la base de datos sin que el resto de niveles se vean afectados.

Por ejemplo, añadiendo o eliminando un registro, sin que cambie la estructura.

2- **INDEPENDENCIA FÍSICA:** podemos modificar el esquema interno y que el resto de niveles no se vean afectados (externo, conceptual).

- Podremos crear una estructura de acceso adicional sin que afecte a los usuarios
- Esto permite una separación entre el hardware (estructuras de almacenamiento) y las aplicaciones.

# Independencia de Datos

Es más fácil lograr la independencia física que la lógica.

Si contamos con un SGBD en **tres capas**, hay que incluir la información sobre cómo asociar las peticiones de los usuarios con los distintos niveles: esto se realiza mediante la ampliación del catálogo o diccionario del SGBD.

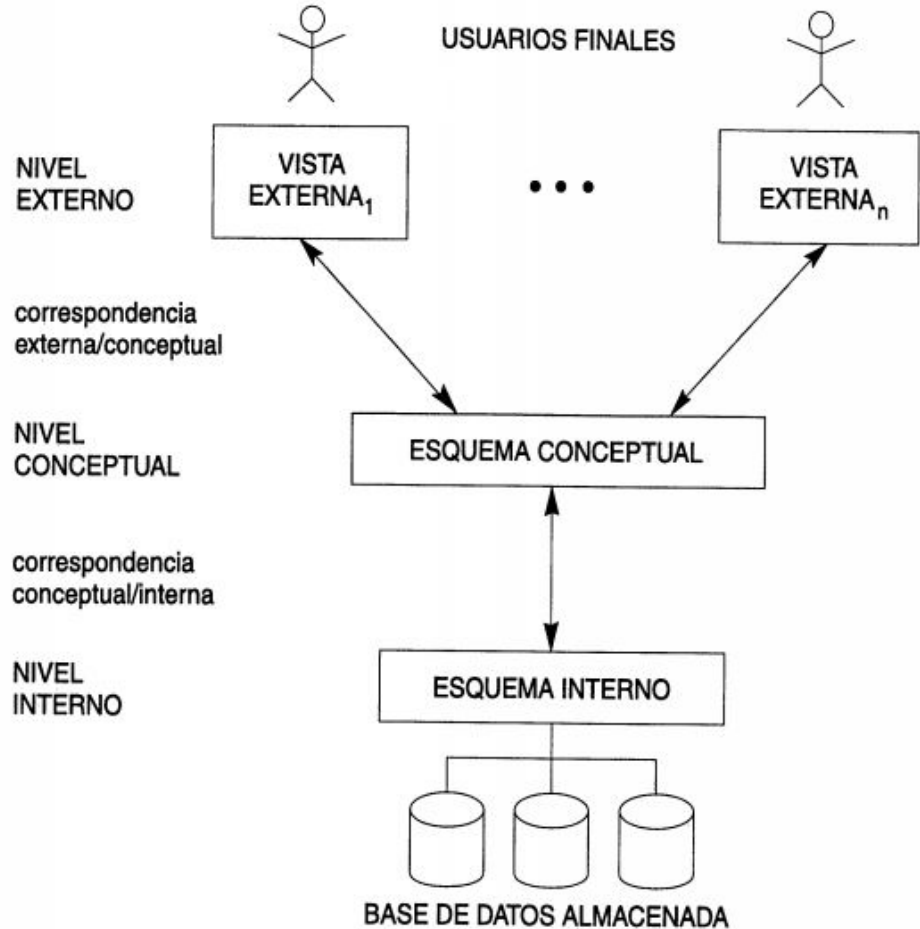
Si tenemos una arquitectura con tres esquemas la independencia de datos, tanto de forma física como lógica es más sencilla. Los tres niveles de esta estructura son:

- Nivel externo
- Nivel conceptual
- Nivel interno

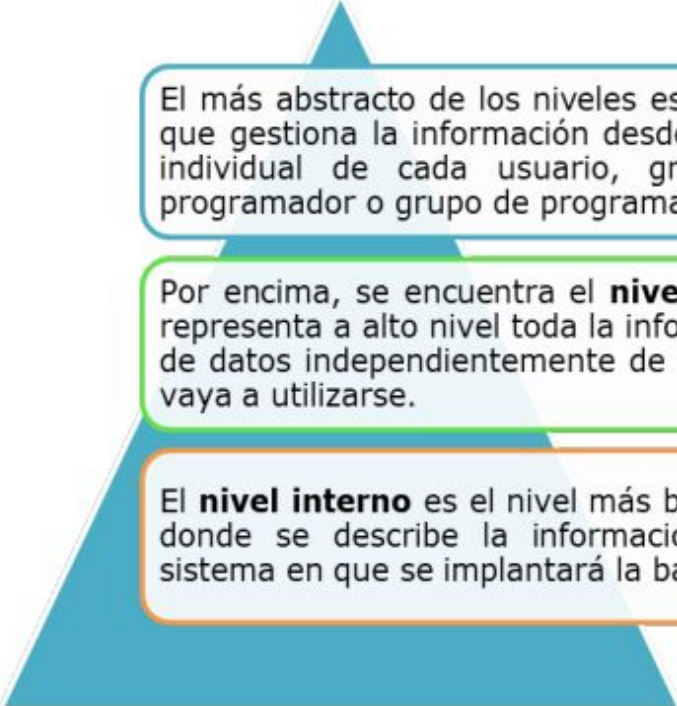
# Arquitectura ANSI

Propuesta a tres niveles, muchos autores dividen el conceptual en dos ya que no existe una herramienta general que permita definir el modelo conceptual.

- No siempre son 3 niveles, las dos capas bajas se pueden mezclar.



# ANSI: Niveles de Abstracción



El más abstracto de los niveles es el **nivel externo**, que gestiona la información desde el punto de vista individual de cada usuario, grupo de usuarios, programador o grupo de programadores.

Por encima, se encuentra el **nivel conceptual**, que representa a alto nivel toda la información de la base de datos independientemente de la máquina en que vaya a utilizarse.

El **nivel interno** es el nivel más bajo de abstracción, donde se describe la información en función del sistema en que se implantará la base de datos.



# ANSI: Niveles de Abstracción

**1- NIVEL INTERNO/FÍSICO:** Mediante el esquema interno se describe la estructura física de la base de datos. Aquí se especifican todos los detalles del almacenamiento y los métodos de acceso:

- Los archivos que contienen la información y su organización.
- Cómo acceder a los registros.
- De qué tipo y qué longitud tiene cada registro.

# ANSI: Niveles de Abstracción

**2- NIVEL CONCEPTUAL:** Este nivel se encuentra asociado al esquema conceptual y es donde se definen todos los datos que intervendrán en el sistema.

Esta información se obtiene a partir de los requerimientos de los usuarios y es independiente de la forma o lugar en donde se van a almacenar y recuperar los datos.

Incluye datos elementales (campos), datos compuestos (registros), la relación entre ambos, las reglas que rigen el funcionamiento de la empresa, etc.

En este nivel, se describe la estructura de toda la base de datos de una empresa u organización.

En este esquema, se ocultan los detalles del almacenamiento y se centra en describir entidades, atributos, relaciones, operaciones de los usuarios y restricciones.

# ANSI: Niveles de Abstracción

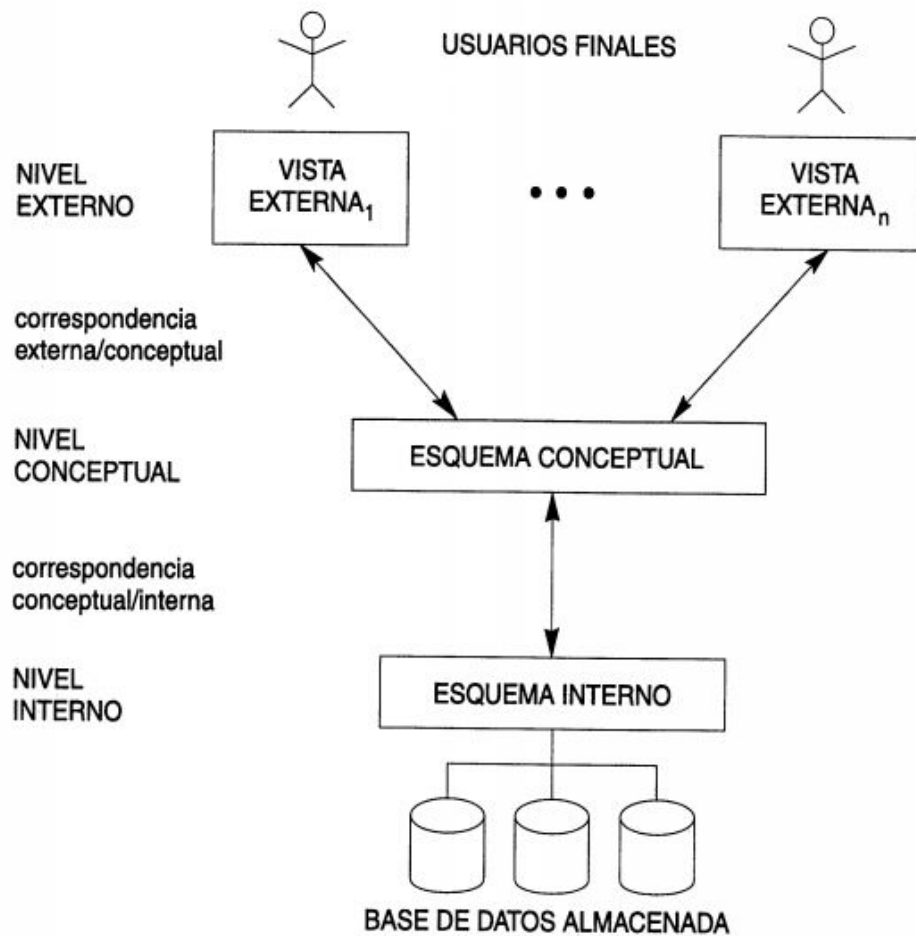
El esquema conceptual contiene:

- **REGISTROS:** representan las entidades en el mundo real. Por ejemplo: empleados, departamentos, etc.
- **CAMPOS:** atributos de los registros. Por ejemplo, para la entidad empleado: nombre, dirección, NIF, sueldo, etc.
- **RELACIONES ENTRE ENTIDADES:** por ejemplo: un empleado pertenece a un departamento, un departamento está formado por varios empleados, etc.
- **VERIFICACIONES DE INTEGRIDAD:** por ejemplo: un empleado no puede pertenecer a un departamento que no existe, no se puede asignar a un departamento un usuario inexistente, etc...

# ANSI: Niveles de Abstracción

**3- NIVEL EXTERNO:** Este nivel, es el conjunto de percepciones individuales de la base de datos.

- SUBESQUEMA o **VISTA**: cada visión individual.
- Una vista puede ser **compartida por varios usuarios** y cada usuario tendrá la posibilidad de acceder a varias vistas.
- Al crear una vista, es posible mezclar los campos de distintos registros, omitir campos o cambiar su orden, añadir campos que se puedan calcular a partir de lo descrito en el esquema conceptual, etc.
- En este nivel, se describen varios esquemas externos o vistas de usuario.
- Cada esquema externo describe la parte de base de datos que interesa a un grupo determinado de usuarios y **oculta el resto de la información**.
- Para una base de datos concreta, solo existe un esquema interno y un esquema conceptual, pero puede muchos esquemas externos, para uno o varios usuarios.

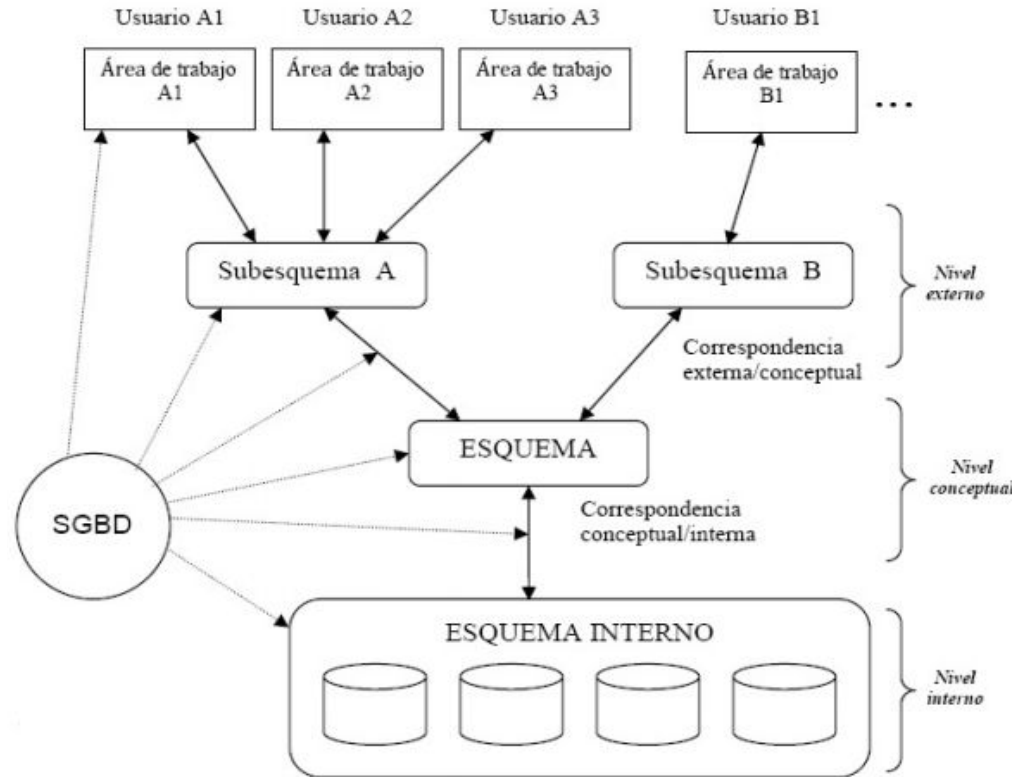


# ¿Cómo funciona una petición?

## CORRESPONDENCIA o

**TRANSFORMACIÓN:** se encarga de transformar de un nivel a otro las peticiones y los resultados de las consultas.

Nivel externo	Nivel conceptual	Nivel interno
Usuarios	Diseño	Almacenamiento





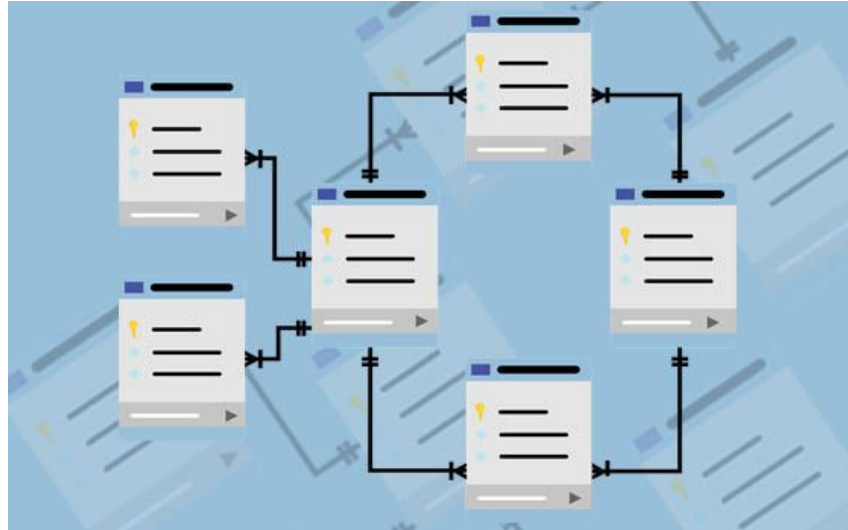
# Bases de Datos

Modelos de Bases de Datos



# Modelos de BD

- Habrá diferentes tipos de BD en función de cómo se almacene la info
- Cada tipo de base de datos tiene su propio modelo de datos
- El modelo de datos indica de qué manera están estructurados los datos





# Modelos de BD

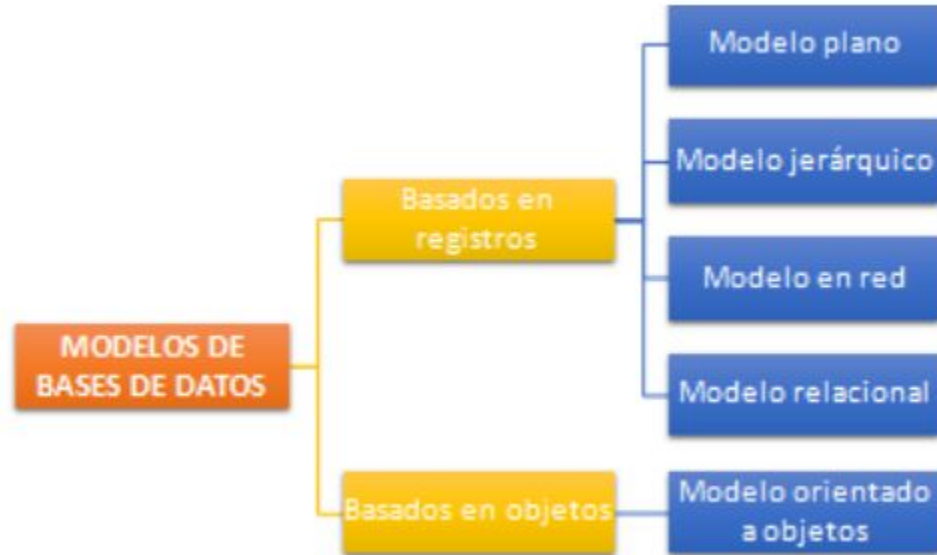
## 1) Modelos basados en **registros**:

- Describen los datos a nivel conceptual y físico
- La BD datos está estructurada en registros de distintos tipos
- Cada tipo de registro define un número de campos o atributos
- Cada campo define una longitud determinada.

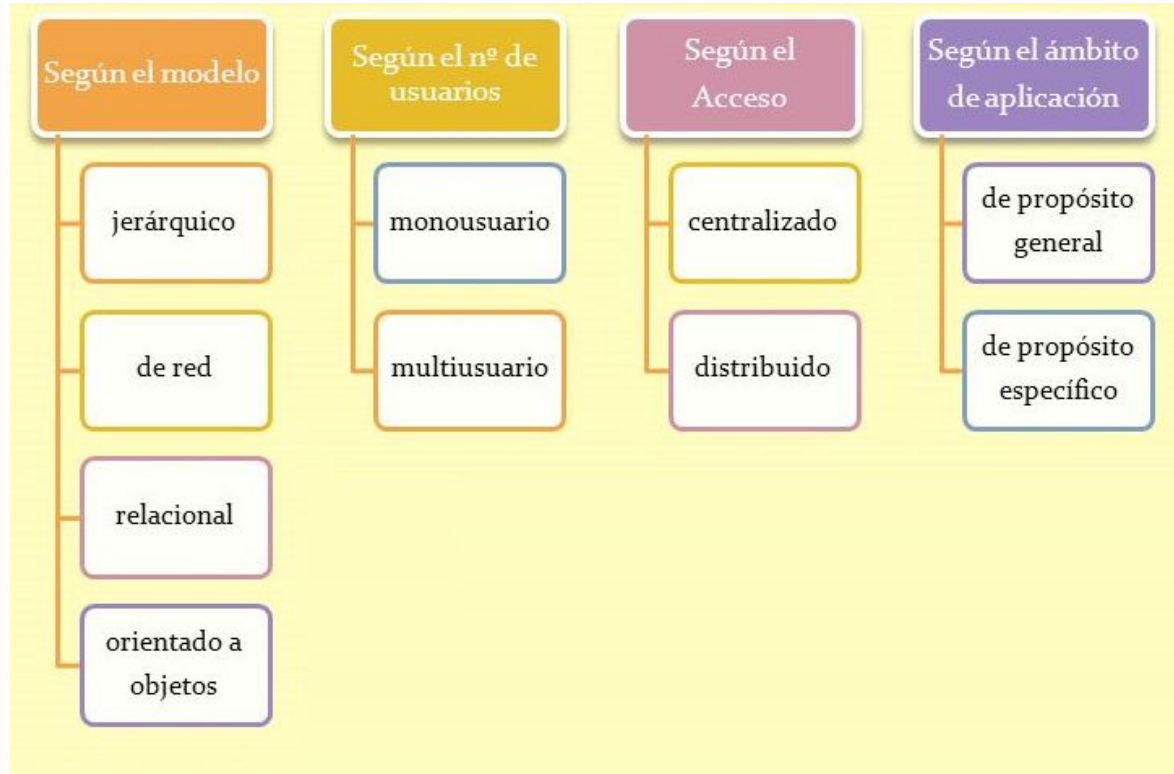
## 2) Modelos basados en **objetos**:

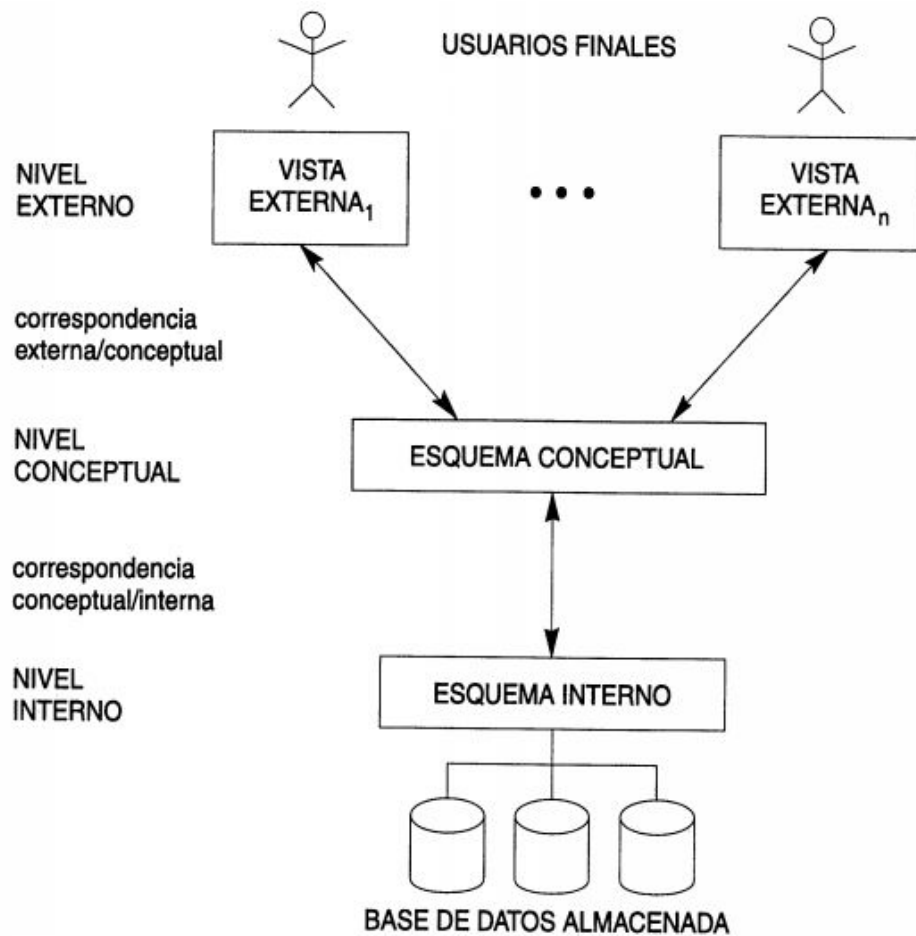
- Permiten describir los datos en los niveles conceptual y de vista.
- Permiten una estructuración bastante flexible de los datos y permiten especificar claramente las restricciones.
- Los dos modelos que vamos a tratar se caracterizan porque favorecen el diseño de la BD, son los que mejor se prestan para hacer el paso de modelo conceptual al diseño lógico.

# Modelos de BD



# Modelos de BD





# Modelo Relacional



# Modelo Relacional

- Creado por Codd
- Este modelo surgió como una alternativa a las bases de datos jerárquicas y en red.
- Lo que pretendía era una **mayor flexibilidad** y rigor en el tratamiento de los datos.
- El modelo relacional es el más popular.
- La base de datos relacional está **compuesta por tablas y relaciones**
- En este modelo los datos se representan mediante tablas, donde cada tabla tiene varias columnas con nombres únicos y una columna de una tabla representa una relación entre un conjunto de valores. Las relaciones entre los datos se modelan usando las tablas.

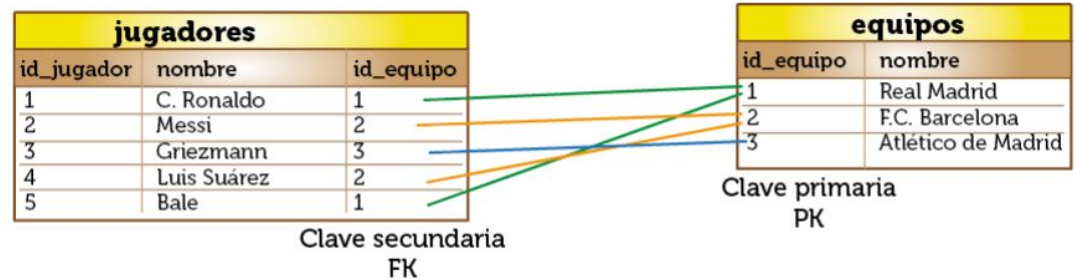
# Modelo Relacional

- La tabla es una estructura bidimensional, compuesta por filas y columnas donde las filas contienen a los registros y las columnas contienen a los campos.
- Los registros son todos del mismo tipo y cada uno de estos está compuesto por un número fijo de campos.
- Si tenemos varias tablas relacionadas, deben tener un campo común (clave primaria) que se utiliza para enlazar la información almacenada en ambas tablas.



# Modelo Relacional

- No pueden contener campos repetidos.
- No permiten registros duplicados.
- Existe un campo, clave primaria, que identifica de forma única al registro.
- Permite distintos tipos de relaciones: uno a uno, uno a muchos y muchos a muchos.





# Modelo Relacional



# Modelo Relacional

Ventajas de una base de datos relacional:

- Es mucho más flexible que los modelos jerárquicos y en red.
- Se puede restringir el acceso a los datos según las necesidades dando así mayor seguridad.
- Aporta mayor flexibilidad de acceso a la información.
- Evita los datos duplicados, esto reduce la necesidad de memoria y aumenta la eficacia.

Las bases de datos relacionales se gestionan mediante un Sistema Gestor de Bases de Datos Relacionales (SGBDR) y normalmente utiliza como lenguaje de programación de la base de datos **SQL**.

# Nomenclatura

**Campos o Atributos**

**Registros o Tuplas**

Nombre	Nacionalidad	Institución
Sofia	España	O.N.U.
John	EE. UU.	O.M.S.
Jacob	Francia	N.A.S.A.
Sebastian	Chila	UNASUR

IdEquipo	Nombre
0	LocosDelBalón
1	18 Futbol Club
2	Imparables
3	SinVieado FC
4	Los mejores
5	Luchadores
< Campo auto r	

**Registro**

**Dato**

**Campo**

# Ejemplo Modelo Relacional

Nombre	Departamento	Sueldo
Pedro	Sistemas	3000
Juan	Comercial	2500
Raúl	RRHH	1500
Alberto	Contabilidad	2000



Empleados (Juan, Comercial, 2500)

- Cada campo es un dominio de datos (dominio sueldo, dominio dpto)
- Los datos en conjunto están relacionados
- Hay una relación única de valores de cada dominio (Juan - Comercial - 2500)
- Cada tupla es una relación de dominios

# Restricciones del Modelo Relacional

- **Restricciones de dominio:** el valor de cada atributo debe ser un valor simple, dentro del conjunto de valores:
  - La edad no puede ser -50, 300 o 14-15-16.
  - Va a poder ser sólo un número natural entre 0 y 130
- **Restricciones de clave:** Cada conjunto de valores (tupla) debe ser diferente
  - Forman una **superclave**: un valor de atributos único en su conjunto
  - Si cogemos un atributo cualquiera sí que se va a poder repetir (dos nombres iguales)
- **Restricciones de Integridad:**
  - **Integridad de entidades:** Ninguna clave primaria puede tener NULL como valor
  - **Integridad referencial** (entre relaciones): para poder establecer una relación de un tupla de una tabla con otra tupla de otra tabla, deben existir datos en ambas

# Ejemplo

Datos de estudiantes:

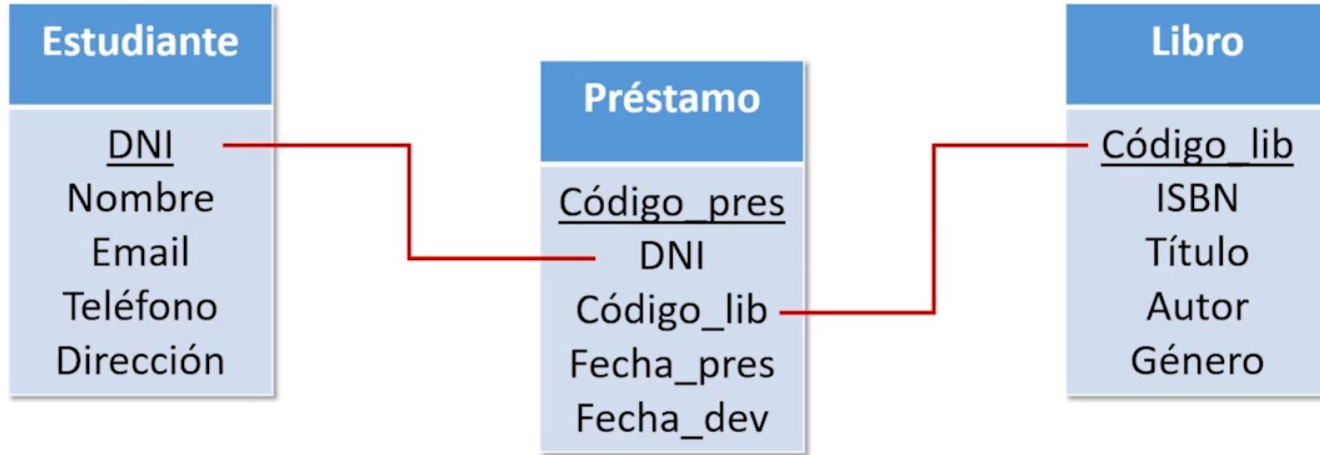
<u>DNI</u>	Nombre	Email	Teléfono	Dirección
------------	--------	-------	----------	-----------

Datos de libros:

<u>Código_lib</u>	ISBN	Título	Autor	Género
-------------------	------	--------	-------	--------

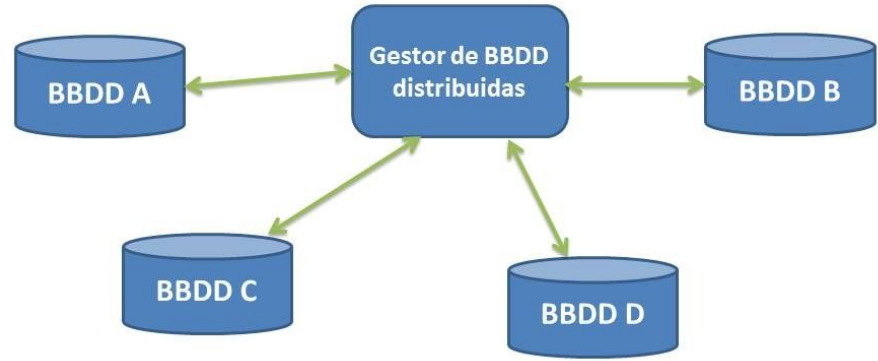
Préstamo (relaciona estudiantes con libros):

<u>Código_pres</u>	DNI	Código_lib	Fecha_pres	Fecha_dev
--------------------	-----	------------	------------	-----------



## Restricciones de integridad:

- **Entidades:** El DNI y el Código del libro no podrán ser NULL
- **Referencial:** Un estudiante no podrá coger prestado un libro que no esté en la BD



# Bases de Datos Centralizadas y Distribuidas

# Base de Datos Centralizada

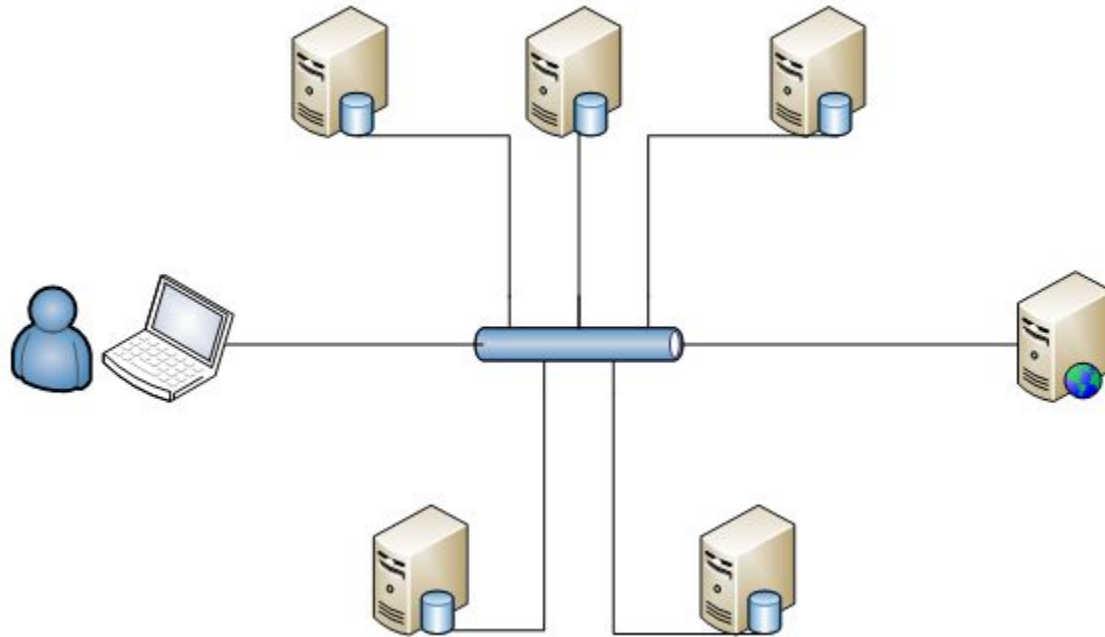


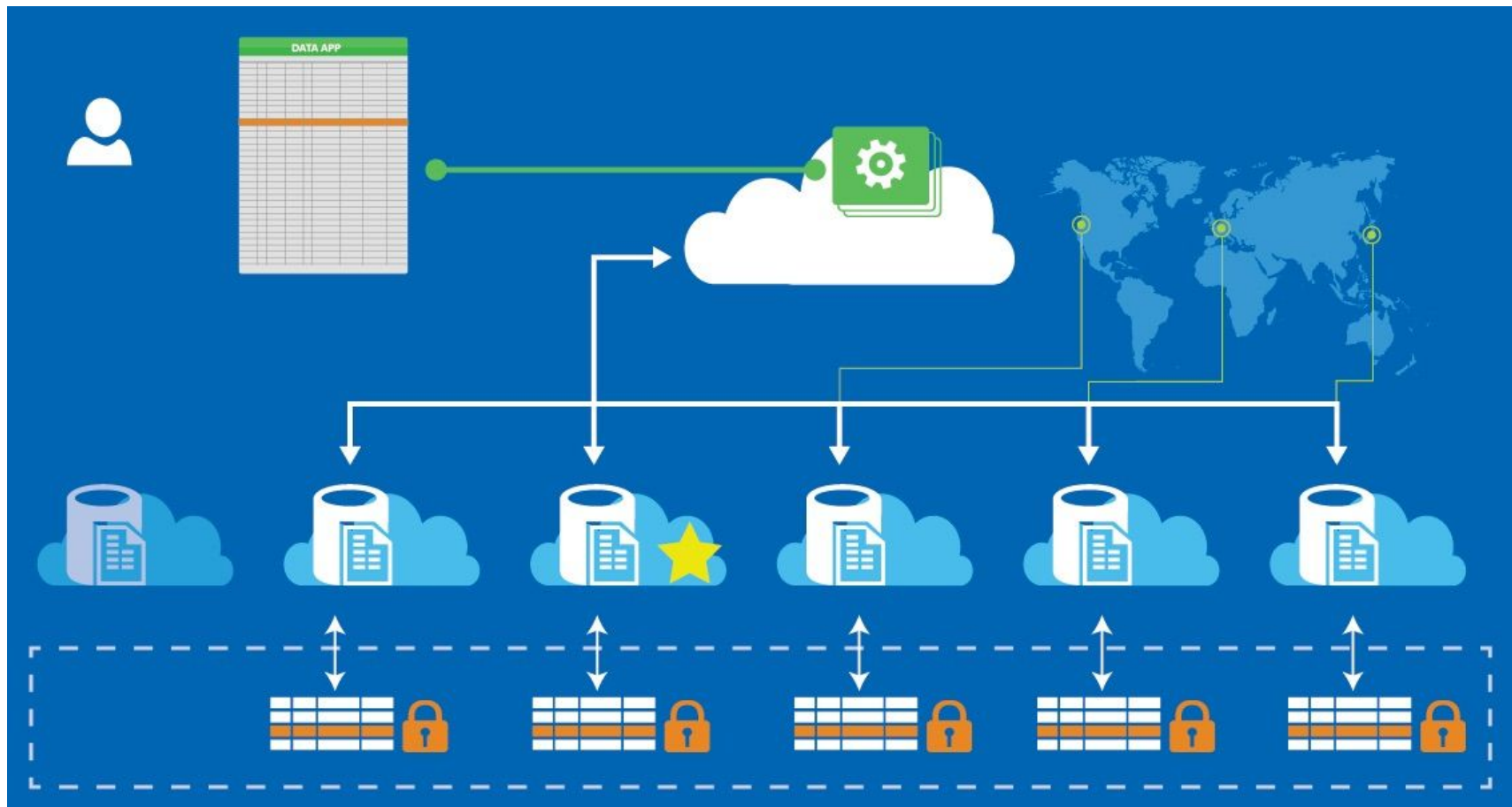


# Base de Datos Distribuida (BDD)

- Colección de datos distribuidos en diferentes nodos de una red de ordenadores que pueden estar distribuidas geográficamente a nivel mundial
- Está estructurada como una serie de bases de datos interconectadas entre sí
- Se comunican por Internet
- Están gestionadas por un **Sistema de Gestión de Bases de Datos Distribuidas (DDBMS)** de tal forma que todo el conjunto parezca como **una única base de datos** para los usuarios.
- Cada nodo de la red es autónomo, puede ejecutar sus propias aplicaciones locales o globales.
- Este nodo requiere acceso a los datos, ubicados en diferentes sitios, por tanto se necesita un sistema global de comunicación.

# Base de Datos Distribuida (BDD)

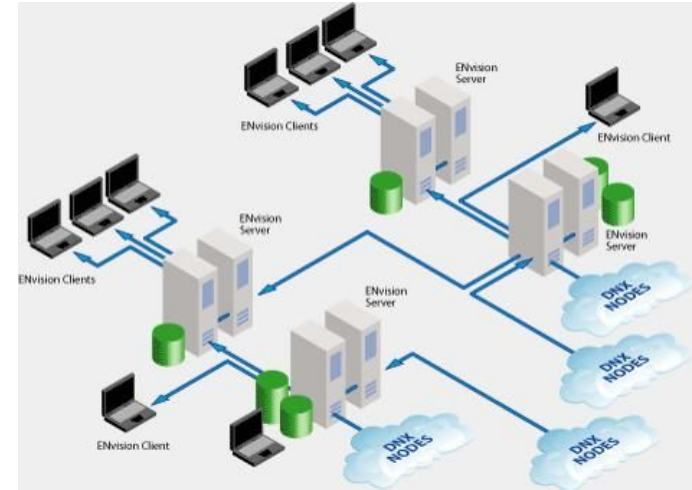




# Sistemas de Bases de Datos Distribuidas:

## Características

- Representan de forma más natural la estructura geográfica descentralizada de cualquier organización.
- Aumentan la disponibilidad de los datos.
- Reducen el tráfico de comunicación.
- Abaratan los costos en equipamiento y en la infraestructura de comunicaciones de las redes de ordenadores.



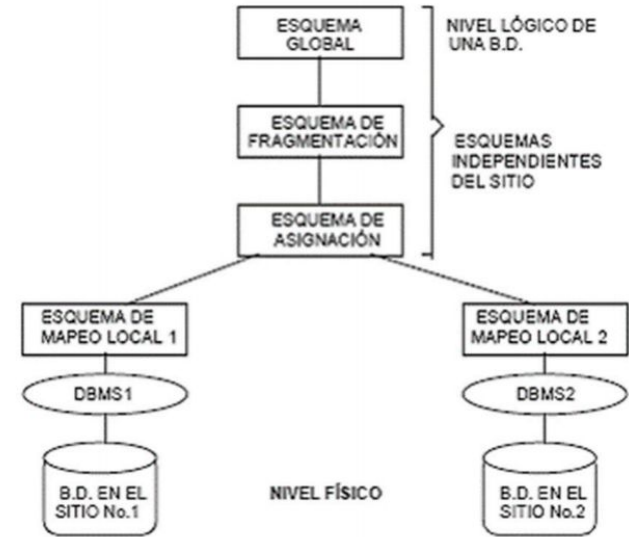
# Estrategias de Diseño de BDD: ENFOQUE TOP-DOWN: FRAGMENTACIÓN:

- Este enfoque es de arriba hacia abajo (top-down) y es el más apropiado para **aplicaciones nuevas** y para sistema homogéneos.
- En este enfoque partimos del análisis de requerimientos, posteriormente se realiza el diseño conceptual y las vistas para los usuarios.
- Partiendo de estos datos, se define el esquema conceptual global y los esquemas externos que van a ser necesarios.
- Después se realiza el diseño de la fragmentación de la base de datos y la localización de cada uno de los fragmentos en los diferentes sitios, además se crean las imágenes físicas.
- Finalmente, este proceso se completa, ejecutando en cada sitio el diseño físico de los datos.



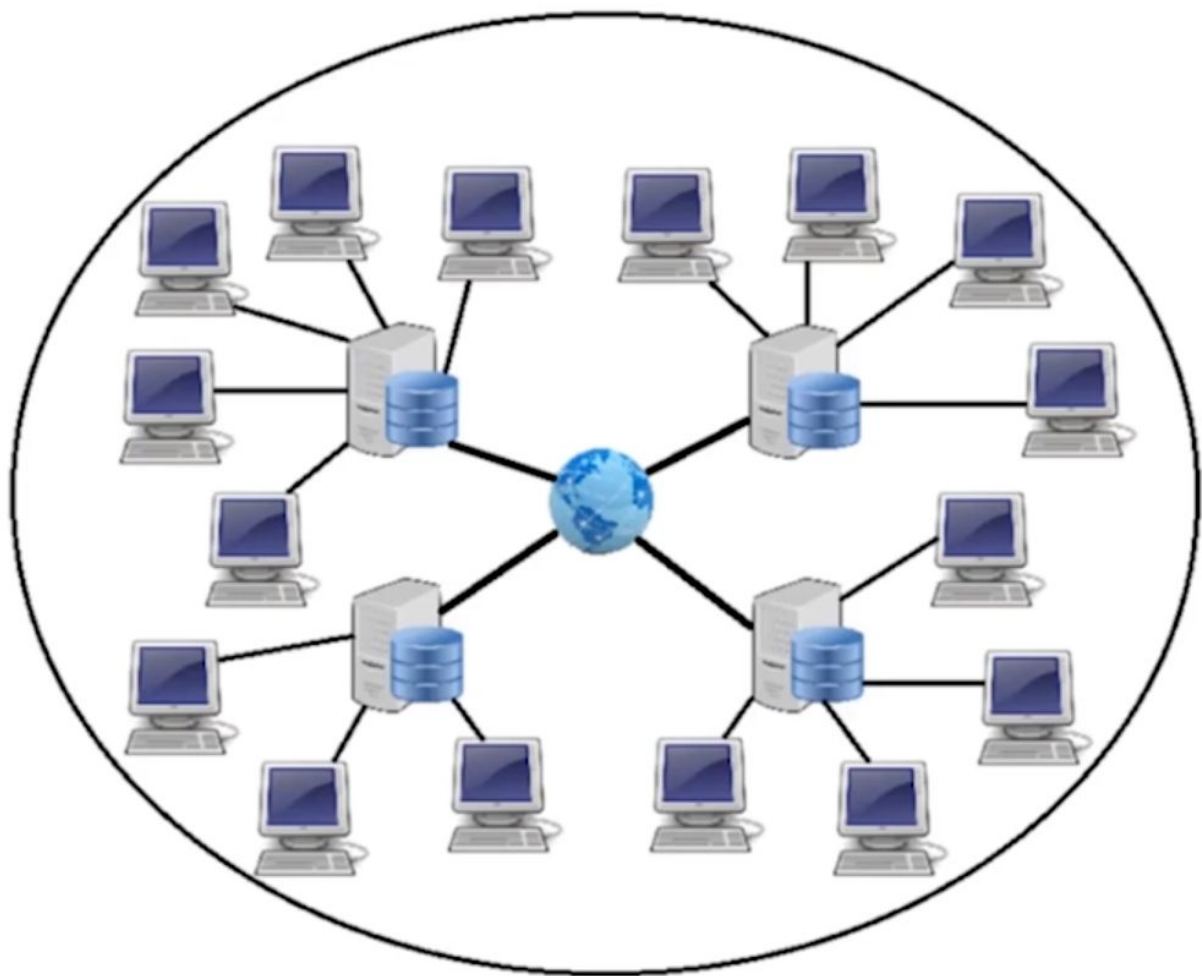
# Estrategias de Diseño de BDD: ENFOQUE BOTTOM-UP: INTEGRACIÓN DE BDs:

- Este enfoque es de abajo hacia arriba
- Se suele utilizar en bases de datos existentes para crear bases de datos distribuidas.
- Lo primero que hay que hacer es seleccionar un modelo de bases de datos común para que describa el esquema global de la base de datos.
- Posteriormente hay que traducir cada esquema local al modelo de datos común.
- El último paso es realizar una integración del esquema local en un esquema global común.
- **Muerte mortal**



# Sistemas de Bases de Datos Distribuidas: Ventajas

- El acceso a los datos es más rápido, ya que los datos se localizan más cercanos al lugar donde se van a utilizar.
- Procesamiento de datos más rápido, ya que en el procesamiento de una carga de trabajo pueden intervenir varios nodos, además, si es necesario, se pueden añadir nuevos nodos al proceso de forma fácil y rápida.
- Si falla un nodo, la probabilidad de que este fallo afecte al sistema es muy baja. Además existe autonomía e independencia entre los nodos.
- Tiene una mayor tolerancia a los fallos, sobre todo si se utilizan mecanismos de replicación de la información, ya que de esta forma el funcionamiento del sistema no depende de un único lugar.
- Menor coste en comunicaciones, ya que si los datos más utilizados se localizan en el lugar donde se usan más, se reduce el coste en las comunicaciones para la manipulación de datos.
- Son amigables al usuario, ya que su estructura interna es totalmente transparente para él.





# Sistemas de Bases de Datos Distribuidas: Desventajas

- El control y la manipulación de los datos es más complicado.
- Frente a fallos impredecibles de componentes hardware o software es complejo asegurar la integridad de los datos (Los datos son íntegros cuando son consistentes, válidos y exactos)
- Tanto la concurrencia como los mecanismos de recuperación son mucho más complejos, ya que existe la posibilidad de que los datos estén replicados.
- Necesitan un software complejo y caro para su funcionamiento.
- Puede haber una sobrecarga de procesamiento, ya que incluso con operaciones simples, se pueden requerir muchos cálculos y comunicaciones para lograr que los datos sean uniformes en todo el sistema.
- La capacidad de respuesta de un sistema distribuido depende principalmente de que se haya realizado una distribución adecuada de los datos. Si no se ha realizado una distribución adecuada, los tiempos de respuesta serán muy lentos
- No existen estándares que ayuden a los usuarios a convertir bases de datos centralizadas en bases de datos distribuidas (Muerte Mortal)

# Fragmentación

Es la descomposición o **partición de una tabla en pedazos** llamados fragmentos.

Cada fragmento **se guarda en sitio diferente**, tiene como objetivo buscar alternativas para dividir una las tablas o instancias en otras más pequeñas.

El mayor reto es identificar la fragmentación ideal para los datos, además de resolver los grados de fragmentación.

Los datos pueden almacenarse en localidades donde son utilizados con mayor frecuencia, de tal manera que la mayor parte de las operaciones sean sólo locales lo cual reducirá el tráfico en la red.

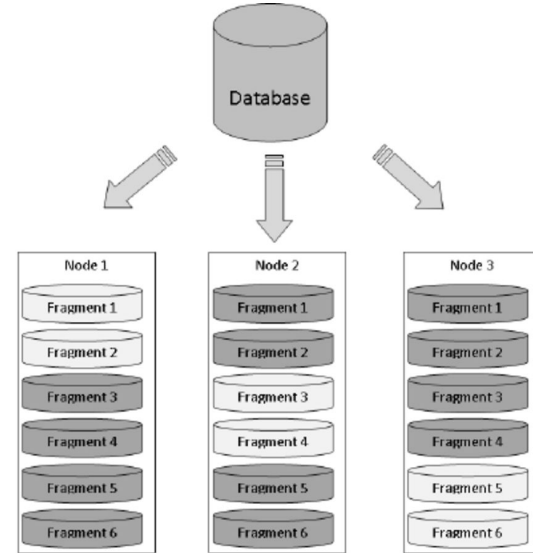
**Fragmentación Óptima:** produce un esquema de división que hace muy pequeño el tiempo de ejecución de las aplicaciones que usan esos fragmentos

- Habrá que tener en cuenta cómo se va a usar para diseñar los fragmentos

# Diseño de una BDD

Para crear una BDD habrá que hacer el diseño de la distribución, teniendo en cuenta:

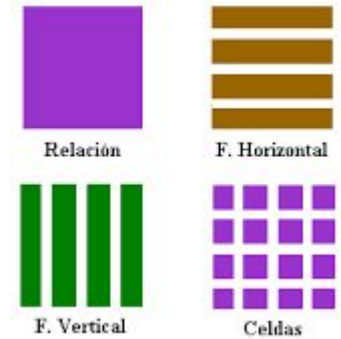
- **FRAGMENTACIÓN:** decidir en qué partes se divide la base de datos y cómo.
- **ASIGNACIÓN:** decidir dónde se ubica cada parte dividida.
- **REPLICACIÓN:** decidir si se va a realizar la replicación de datos, es decir, mantener una copia de los fragmentos en diferentes sitios.



# Fragmentación y Localización en Sist Distribuidos

La distribución de la base de datos requiere determinar por una parte la fragmentación y por otra la localización de los distintos fragmentos. Existen tres tipos de fragmentaciones:

- **FRAGMENTACIÓN HORIZONTAL:**
  - divide la información en subconjuntos
  - Cada fragmento se sitúa en un nodo.
  - Para reconstruir toda la información sólo habrá que unir los subconjuntos.
- **FRAGMENTACIÓN VERTICAL:**
  - divide la información mediante descomposiciones.
  - Cada fragmento incluye la clave primaria de la tabla para poder acceder a toda la información.
- **FRAGMENTACIÓN HÍBRIDA:**
  - es una combinación de las anteriores
  - divide la información tanto de forma horizontal como vertical.



# Replicación

- Consiste en **copiar de forma exacta en otra ubicación una instancia** de la BD.
- Se utiliza en entornos distribuidos donde una sola base de datos tiene que ser utilizada y actualizada en varios lugares de forma simultánea.
  - **Replicación Instantánea:** los datos de un servidor son simplemente copiados a otro servidor o a *otra base de datos dentro del mismo servidor*. Al copiarse todo no necesitas un control de cambios. Se suele utilizar cuando los datos cambian con muy poca frecuencia.
  - **Replicación Transaccional:** primero se envía una copia completa de la base de datos y luego se van enviando de forma periódica (mayor o menor) las actualizaciones de los datos que cambian. Se utiliza cuando necesitas que todos los nodos con todas las instancias de la BD tengan los mismos datos a los pocos segundos de realizarse un cambio.
  - **Replicación de mezcla:** los datos de dos o más bases de datos se combinan en una sola base de datos. En primer lugar se envía una copia completa de la base de datos. Luego el Sistema de Gestión de Base de Datos va comprobando los cambios que van apareciendo en los distintos nodos y a una hora programada o a petición los datos se sincronizan. Es sobre todo útil cuando cada nodo suele utilizar solo los datos que se actualizan allí pero que por circunstancias necesita tener también los datos de los otros sitios.

# Replicación



## VENTAJAS

- Mayor prestación al usuario, ya que los datos son locales.
- Mayor disponibilidad, ya que los datos son siempre accesibles.

## DESVENTAJAS

- Se deben propagar las actualizaciones por todas las réplicas.

# BDSs: Tipos

## **HOMOGÉNEAS:**

- Si todos los sitios de la BDD tienen el mismo SGBD, HW y SO.
- Todos los sitios son conscientes de la existencia de los demás sitios y cooperan para procesar las distintas solicitudes.
- Cada sitio local mantiene el mismo esquema y SGBD.

## **HETEROGÉNEAS:**

- Cada sitio puede tener un SGBD, HW o SO distinto y esquemas diferentes.
- Algunos sitios puede ser que no conozcan a los otros
- En este caso, puede que sólo se ofrezcan facilidades limitadas para la cooperación en el procesamiento de las transacciones

# SGDBs para BDSs

Es el sistema que se utiliza para gestionar la base de datos distribuida de forma transparente para el usuario

- **Acceder a sitios remotos y transmitir consultas y datos** a través de diferentes sitios mediante una red de comunicación.
- Almacenar el esquema de distribución y réplica de datos. Este esquema se almacena en el **catálogo del sistema**.
- Establecer **estrategias** de ejecución de las consultas y transacciones que acceden a los datos de más de un sitio.
- Decidir **qué copia de los datos replicados se debe utilizar**.
- Mantener la **consistencia** de las copias de los datos replicados.
- Realizar la **recuperación del sistema ante los fallos** que puedan surgir.



# Reglas de los SGDBs

1. Regla 0: **“Desde el punto de vista del usuario, un sistema distribuido será igual que un sistema centralizado”.**
2. **AUTONOMÍA LOCAL:** todos los sitios de un sistema distribuido deben ser autónomos.
  - La autonomía local implica que hay: propietario, administración, responsabilidad, integración y representación locales.
3. **NO DEPENDENCIA DE UN SITIO CENTRAL:** no existe un único sitio central.
  - La centralidad implica: cuello de botella y vulnerabilidad.
4. **OPERACIONES CONTINUAS:** realiza operaciones continuas de adición de elementos y de actualización de versiones.
5. **INDEPENDENCIA DE LA LOCALIZACIÓN:** el usuario desconoce la ubicación física de los datos, trabajan como si todos los datos estuvieran en local.
6. **INDEPENDENCIA DE LA FRAGMENTACIÓN:** esto simplifica las aplicaciones de los usuarios y sus actividades en la terminal.

# Reglas de los SGDBs

7. **INDEPENDENCIA DE LA RÉPLICA:** la gestión de las réplicas (creación y destrucción) debe ser transparente al usuario.
8. **PROCESAMIENTO DISTRIBUIDO DE CONSULTAS:** los sistemas distribuidos incluyen herramientas de consulta muy eficientes.
9. **MANEJO DISTRIBUIDO DE LAS TRANSACCIONES:** incluye el control de recuperación y el control de concurrencia.
10. **INDEPENDENCIA DE EQUIPOS:** la ejecución será la misma independientemente de las características del equipo donde se ejecute.
11. **INDEPENDENCIA DEL SISTEMA OPERATIVO:** la ejecución será independiente del sistema operativo del equipo donde se ejecute.
12. **INDEPENDENCIA RESPECTO A LA RED:** el SGBDD deberá manejar distintos tipos de redes de comunicaciones sin afectar al usuario.
13. **INDEPENDENCIA RESPECTO AL SGBD:** se pueden manejar distintas copias del SGBD siempre que todos los SGBD de cada sitio utilicen la misma interfaz.



# Bases de Datos No Relacionales (NoSQL)

Not only SQL OR ~~SQL~~

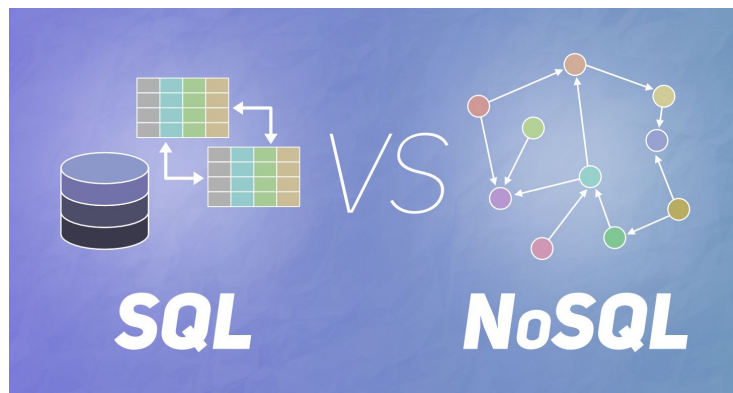
# Base de Datos No Relacional (NoSQL)

Es una base de datos que **no utiliza el esquema habitual** de una base de datos relacional, es decir **tablas formadas por filas y columnas**.

Estas bases de datos utilizan un **modelo de almacenamiento no relacional** que está optimizado para los requisitos específicos del tipo de datos que almacena. Se denominan NoSQL, ya que no utilizan el lenguaje SQL en sus consultas.

Los modelos de datos son más flexibles

Para datos **no estructurados**



# Características de las BD NoSQL

- Son **más rápidas que las bases de datos relacionales** ya que tienen capacidad para gestionar una gran cantidad de peticiones que requieren muchas operaciones por segundo.
- Son idóneas para **aplicaciones que necesitan almacenar o acceder a grandes cantidades de información** en poco tiempo.
- Tienen una **capacidad de almacenamiento mayor** que las bases de datos relacionales.
- Son mucho más **flexibles**, ya que permiten el almacenamiento de diferentes tipos de información.

# Ventajas del NoSQL

- No necesitan máquinas con grandes recursos, ya que no necesitan prácticamente procesamientos, por tanto pueden utilizar **máquinas de bajo coste y con pocas prestaciones**.
- Para mejorar el rendimiento, sólo es necesario añadir más nodos e indicar al sistema que están disponibles, por tanto la **escalabilidad** es horizontal.
- Utilizan una **estructura distribuida** que permite manejar gran cantidad de datos.
- El **procesamiento es mucho más sencillos** que en las bases de datos centralizadas, por tanto se evitan los cuellos de botella que genera el exceso de ejecución de sentencias.

# Desventajas del NoSQL

- **No son totalmente compatibles con instrucciones SQL** de bases de datos relaciones.
- **Poca estandarización**, aunque ya hay algunas muy extendidas
- No son multiplataforma, **no todos los sistemas pueden utilizarlas.**
- Su **administración es complicada** ya que las herramientas no son muy accesibles o se realiza desde consola

# SQL vs NoSQL

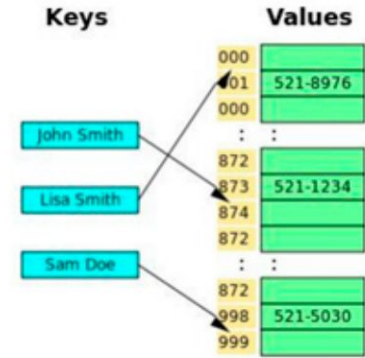
BASES DE DATOS RELACIONALES (SQL)	BASES DE DATOS NoSQL
Utilizan un lenguaje estándar SQL.	Utilizan APIs de comunicación diferentes.
Esquemas rígidos definidos previamente.	Esquema flexible, se genera según se añaden nuevos datos.
Garantiza las propiedades: atomicidad, consistencia, aislamiento y durabilidad.	No garantiza: atomicidad, consistencia, aislamiento y durabilidad.
Modelo de datos único.	Suelen tener varios modelos de datos.
Orientadas a filas.	Suelen ser distribuidas.



# NoSQL: Clasificación según como se almacena la info

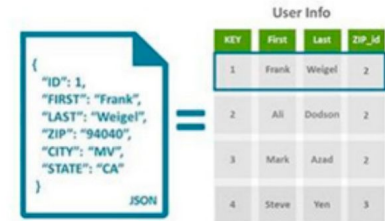
## - BASES DE DATOS CLAVE-VALOR:

- es el modelo más popular y más sencillo.
- Cada elemento está identificado por una clave única, esto permite recuperar la información rápidamente.
- Son muy eficientes tanto en escrituras como en lecturas.



## - BASES DE DATOS DOCUMENTALES:

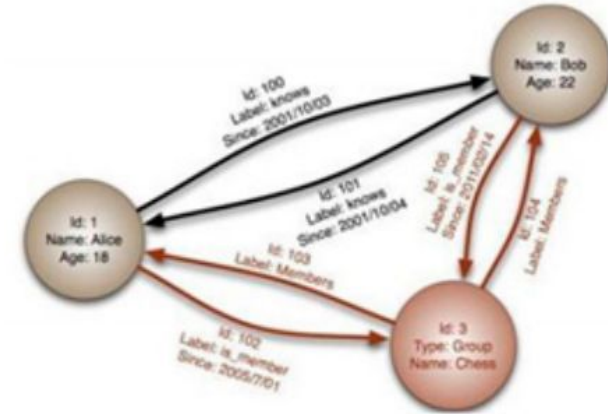
- la información se almacena como un documento y se utiliza una clave única para cada registro.
- Permite búsquedas por clave-valor y búsquedas avanzadas sobre el contenido del documento.
- Son las más versátiles y se pueden utilizar en multitud de proyectos incluso en muchos que funcionarían en bases de datos relacionales



# NoSQL: Clasificación según como se almacena la info

## - BASES DE DATOS EN GRAFO:

- la información se presenta como los nodos de un grafo y las relaciones como las aristas del grafo.
- Se puede utilizar la teoría de grafos para recorrer la base de datos.
- Para un correcto funcionamiento de esta BD la información debe estar normalizada, de tal forma que cada tabla tenga una sola columna y cada relación dos.
- Esta BD ofrece una navegación mucho más eficiente que el modelo relacional.



## - BASES DE DATOS ORIENTADAS A OBJETOS: la información se representa mediante objetos.

Clase	Objetos	Atributos/datos
Empleado	Juan Pérez	Edad: 25
		Puesto: Psicóloga social
		Salario: 8000
	María Suárez	Edad: 23
		Puesto: Pedagoga
		Salario: 15 000