

Bases de datos I

Unidad 1

Introducción - Conceptos Generales



Prof: Trini

Comunicación:

- Correo:
trinidad.Latorre@docentes.unpaz.edu.ar

- Campus.

Cronograma

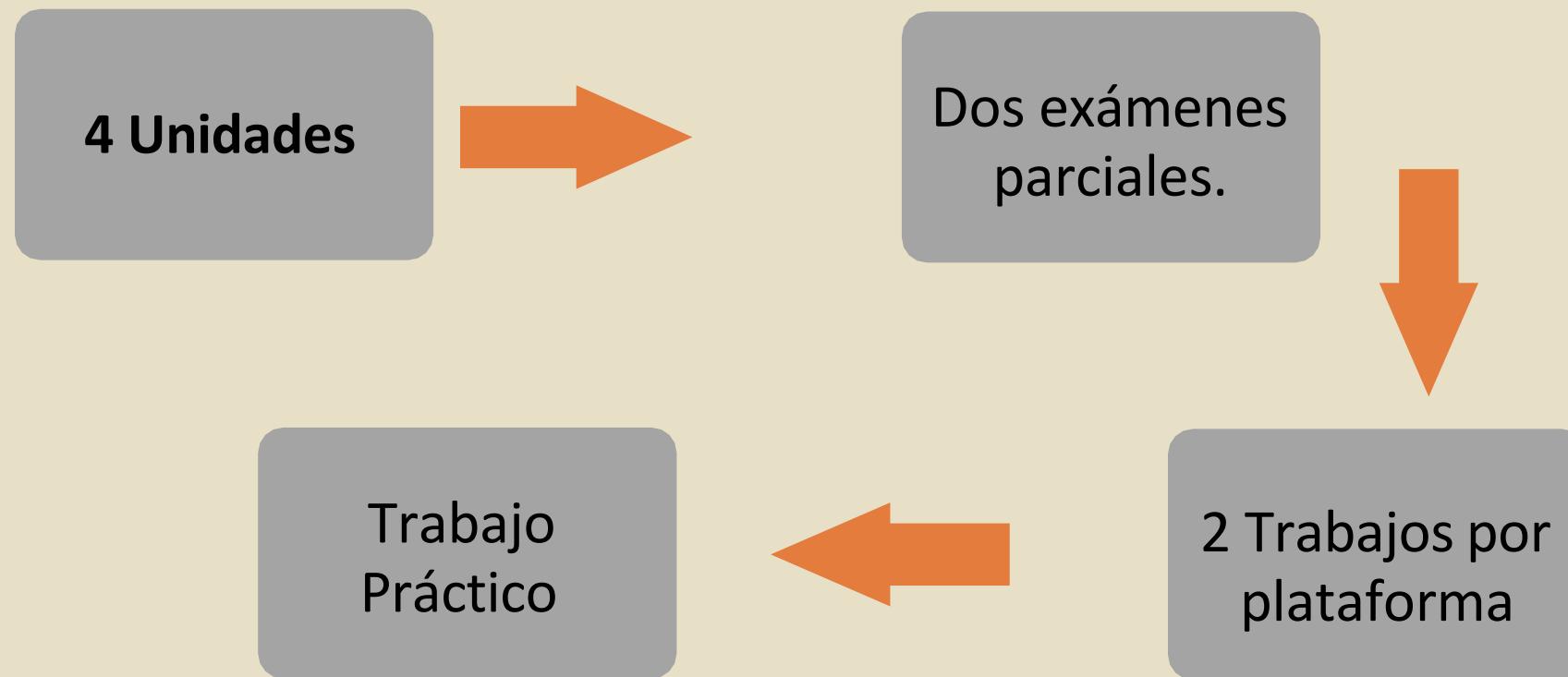
| Semana | Fecha | Tema | Modalidad |
|--------|----------------------|---|----------------------|
| 1 | 11/8/2025 | PARO DOCENTE | Presencial |
| 2 | 18/8/2025 | U1 - Presentación de la UC: Bases de Datos | Virtual - sincrónica |
| 3 | 25/8/2025 | U2 - Modelo Entidad Relación | Presencial |
| 4 | 1/9/2025 | U2 - Modelo Entidad Relación | Virtual - sincrónica |
| 5 | 8/9/2025 | U3 - Modelo Relacional | Presencial |
| 6 | 15/9/2025 | U3 - Lenguajes de Consulta Formales | Virtual - sincrónica |
| 7 | 22/9/2025 | Primer Examen Parcial | Presencial |
| 8 | 29/9/2025 | U4 - Lenguajes de Consulta Estructurado | Virtual - sincrónica |
| 9 | 6/10/2025 | U4 - Lenguajes de Consulta Estructurado | Presencial |
| 10 | 13/10/2025 | U4 - Lenguajes de Consulta Estructurado | Virtual - sincrónica |
| 11 | 20/10/2025 | Día del Municipio de José C. Paz | ASUETO |
| 12 | 27/10/2025 | Recuperatorio Primer Parcial | Virtual - sincrónica |
| 13 | 3/11/2025 | Segundo Examen Parcial / Entrega de TP | Presencial |
| 14 | 10/11/2025 | U4 - Lenguajes de Consulta Estructurado | Virtual - sincrónica |
| 15 | 17/11/2025 | Recuperatorio Primer Parcial | Presencial |
| 16 | 24/11/2025 | Cierre | Virtual - sincrónica |
| | 1/12/2025 | Examen Integrador | Presencial |
| | 15-20/dic '25 | Examen Final | Presencial |



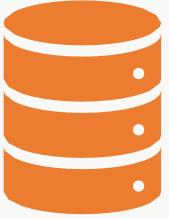
Objetivos de BD1

- Adquirir los conceptos básicos de la teoría de los sistemas de bases de datos. Diferenciar un sistema de Base de Datos de un Sistema de Archivos.
- Realizar el diseño conceptual de un Sistema de Información.
- Diseñar y modelar sistemas de bases de datos relacionales.
- Realizar consultas mediante la utilización de Algebra Relacional.
- Implementar consultas en SQL.

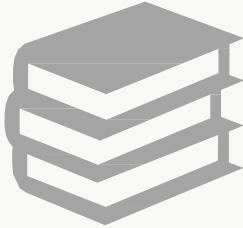
Organización de la materia



Material Bibliográfico



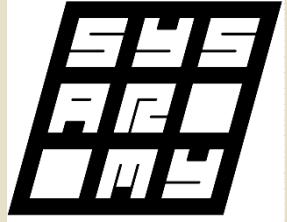
Fundamentos de bases de datos. Sexta Edición. Abraham Silberschatz, Henry F. Korth y S. Sudarshan. McGraw Hill 2014



Introducción a los sistemas de bases de datos. Séptima edición. C.J. Date. Prentice Hall (2001)



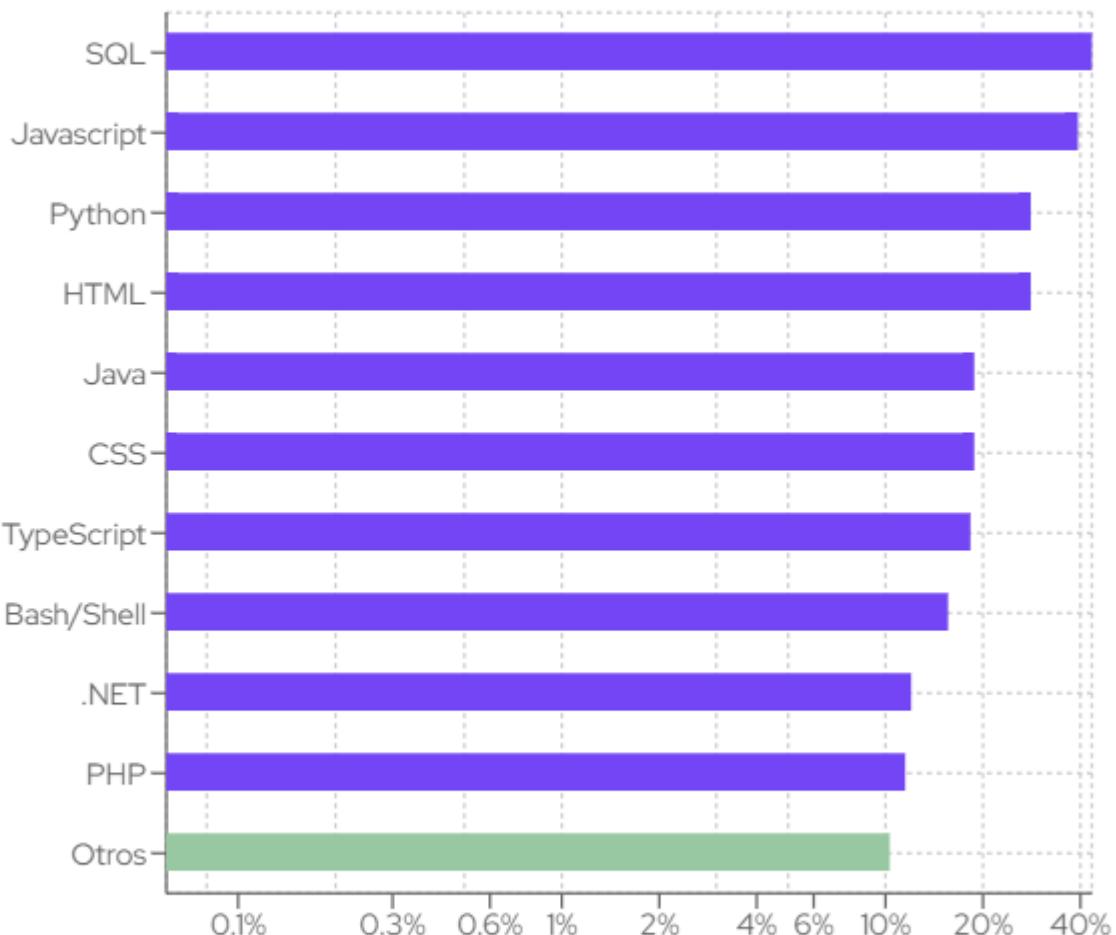
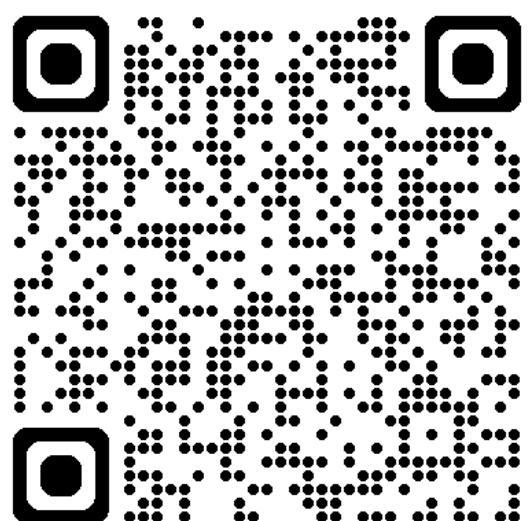
Apunte de la materia



Lenguajes de Programación

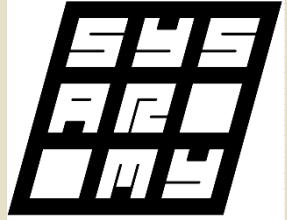
[ver más](#)

<https://sueldos.openqube.io/encuesta-sueldos-2025.01/>

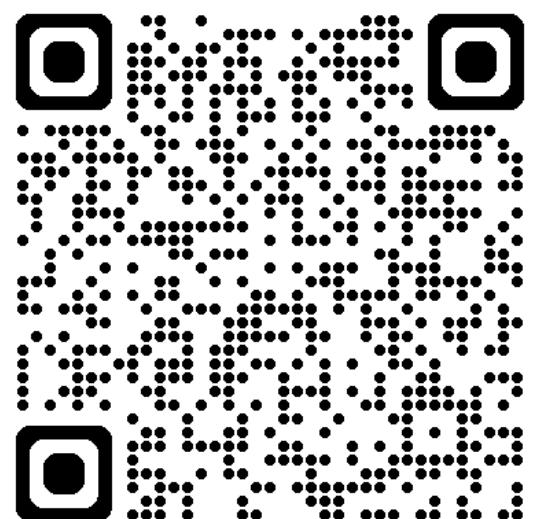


Lenguajes de programación más utilizadas entre los participantes

Los lenguajes de programación no son excluyentes, es decir puede haber más de uno por persona relevada. Los valores son porcentuales sobre el total de participantes.

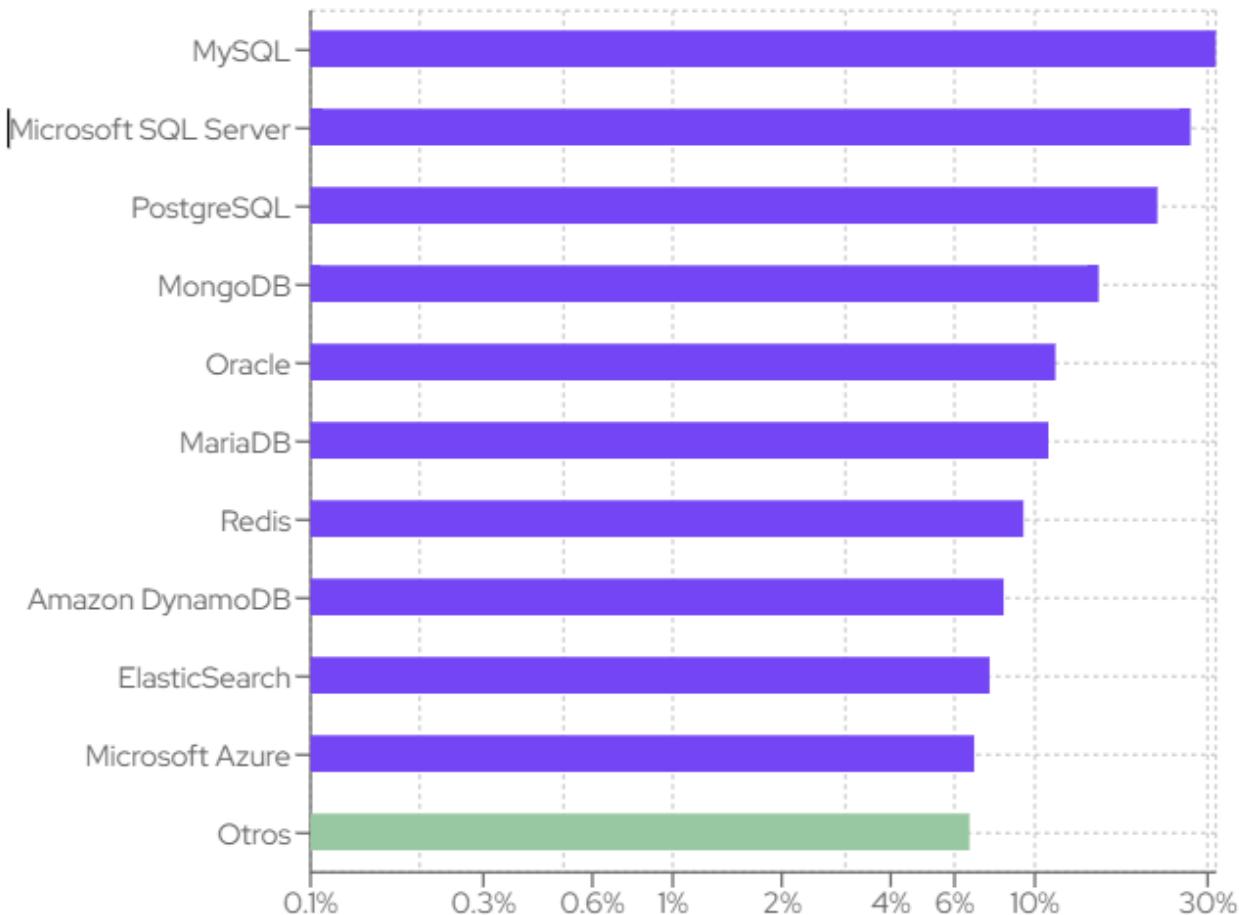


<https://sueldos.openqube.io/encuesta-sueldos-2025.01/>



Bases de Datos

[ver más](#)



Bases de datos más utilizadas entre los participantes

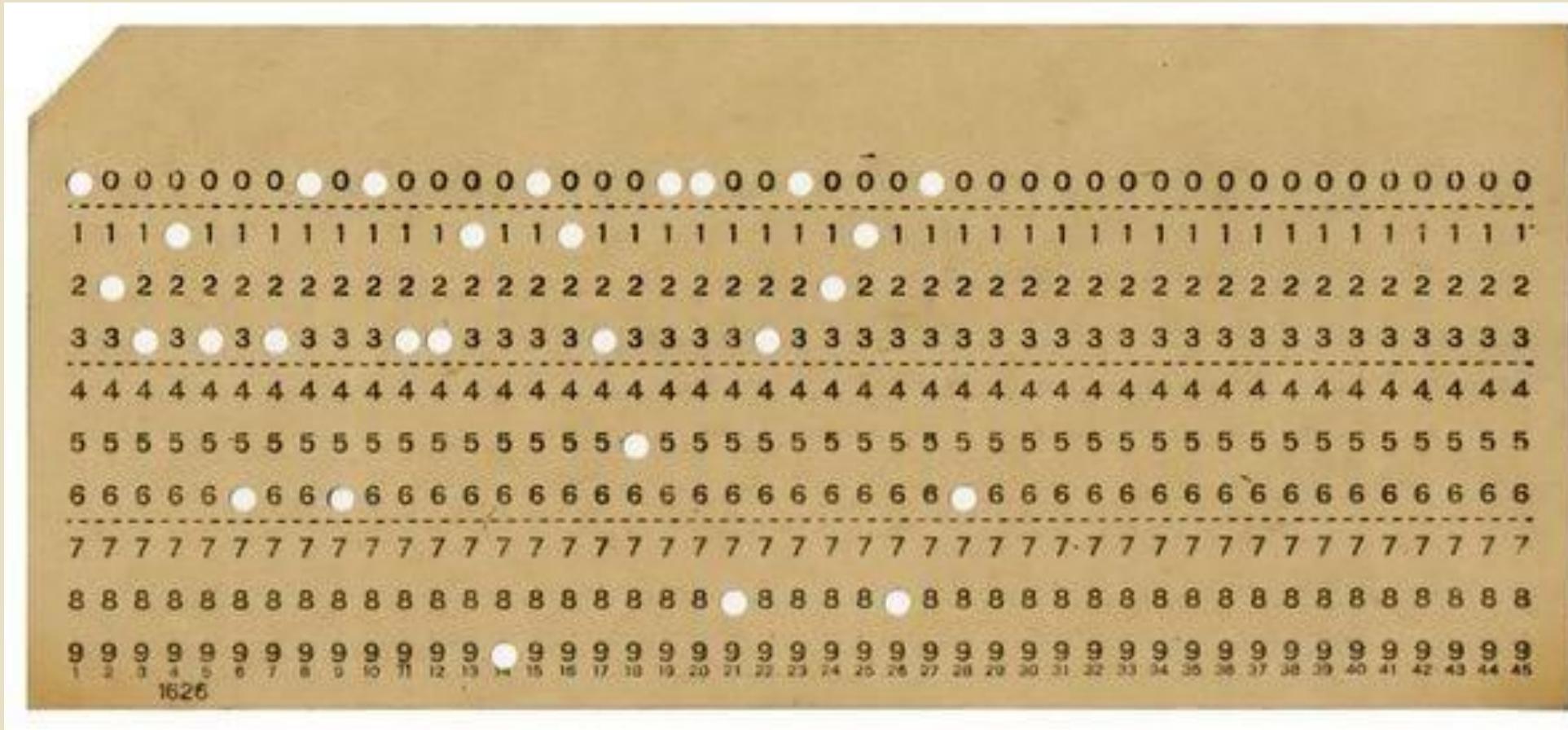
Las bases de datos no son excluyentes, es decir puede haber más de una por persona relevada. Los valores son porcentuales sobre el total de participantes.

Bases de Datos

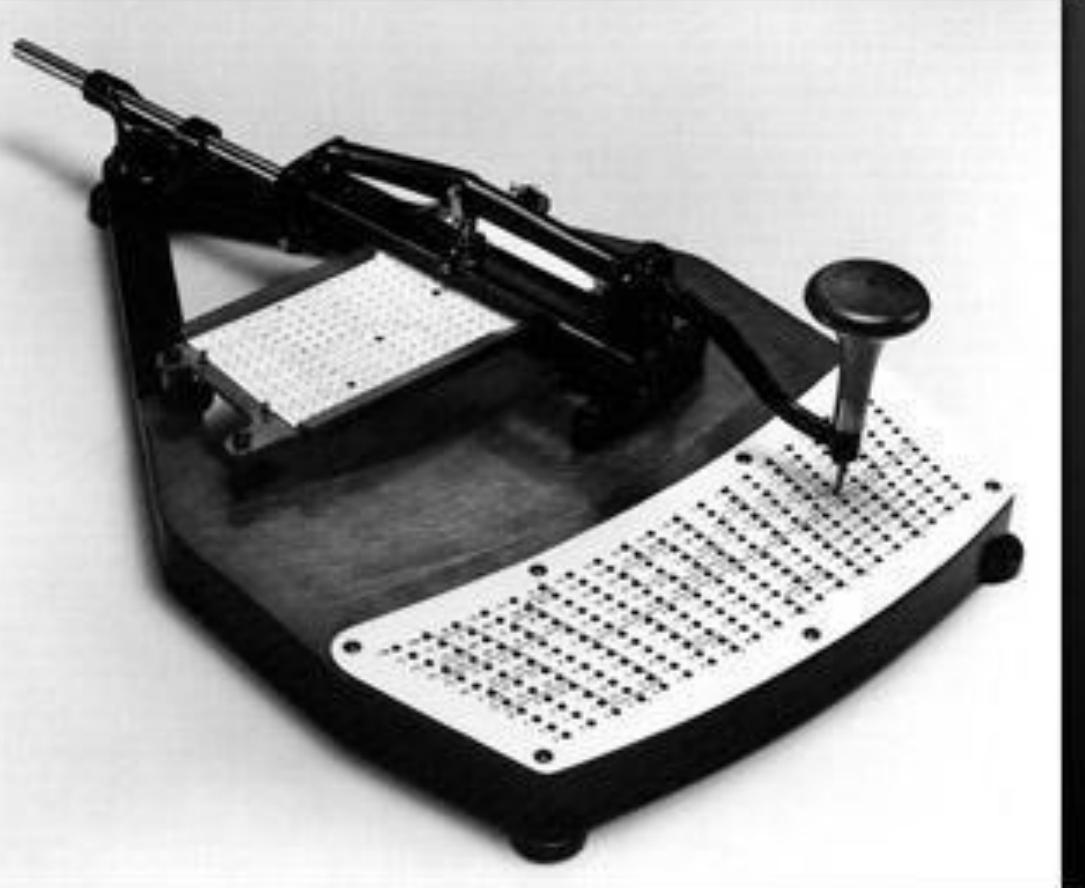
- ¿Alguna vez nos encontramos con una?
- ¿Que contacto tuvimos?
- ¿Trabajamos con alguna cotidianamente?



¿Qué es esto?



Lector y escritor de tarjetas perforadas



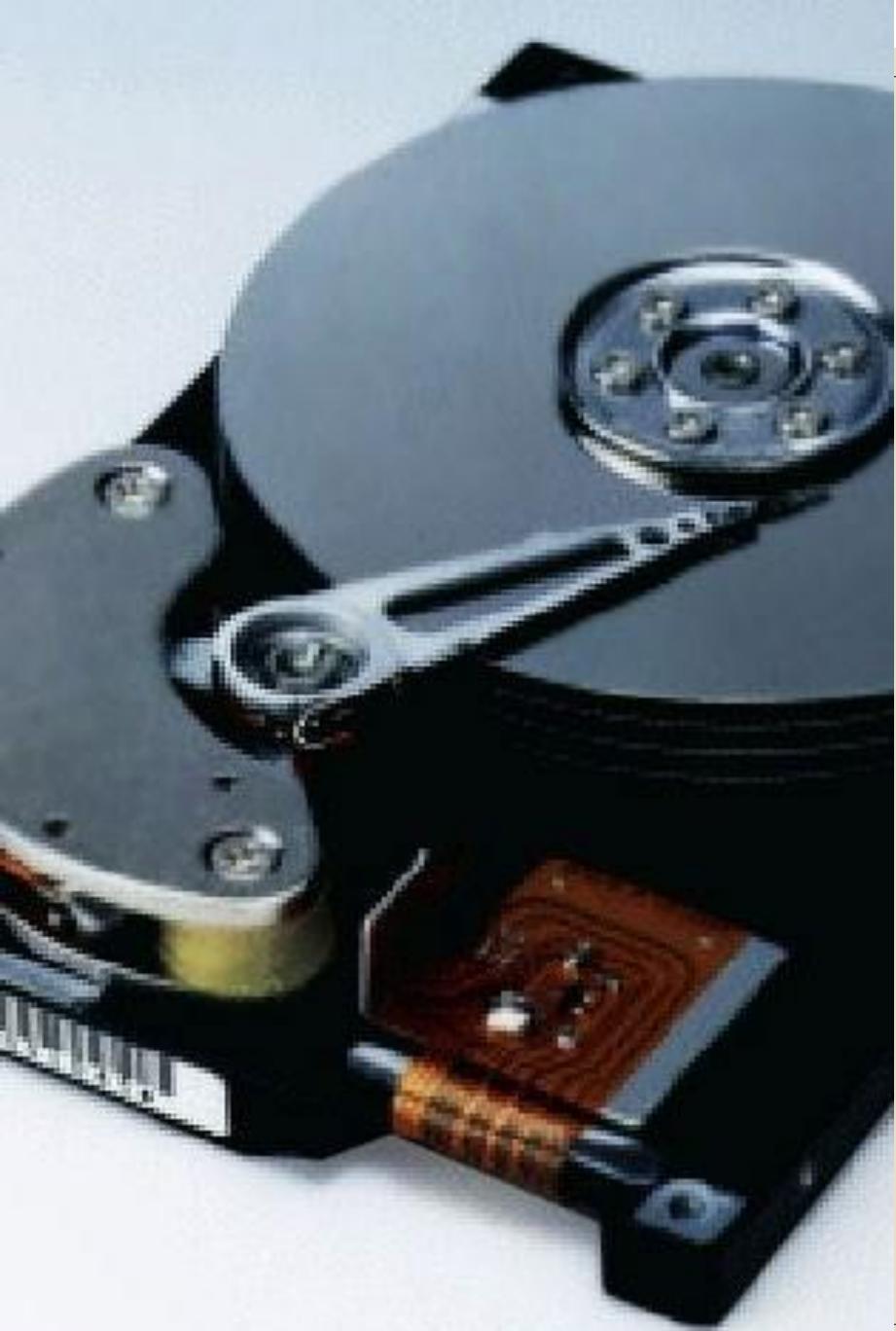






Un poco de historia

- Antes de los 70:
 - Aplicaciones batch. Monotarea
 - Archivos de movimientos en cintas magnéticas (acceso secuencial)
 - Información redundante (para minimizar accesos)



Un poco de historia

- Con los discos y las redes, los archivos comenzaron a consultarse de manera on-line por distintos usuarios.
- Luego surgió la necesidad de poder actualizar también on-line
- Integración de aplicaciones => Integración de archivos => Eliminar redundancia
- Data Banks: primeros archivos interrelacionados, con estructuras complejas y compartidos (a inicios de los 70 se los comenzó a llamar Data Bases). Primeros modelos en red y jerárquicos

Un poco de historia

- El software de gestión de archivos era demasiado elemental (no contemplaba relaciones, actualizaciones simultáneas)
- Mediados de los 70: surgen los Data Base Management Systems (SGBD)



Definición

- Una base de datos de un Sistema de Información es la representación integrada de los conjuntos de entidades instancia correspondientes a las diferentes entidades tipo del Sistema de Información y de sus interrelaciones.
- Esta representación informática (o conjunto estructurado de datos) debe poder ser utilizada de forma compartida por muchos usuarios de distintos tipos.
- En otras palabras, una base de datos es un conjunto estructurado de datos que representa entidades y sus interrelaciones



Base de Datos

- Colección de datos interrelacionados archivados juntos sin redundancias perjudiciales o innecesarias. Su finalidad es la de servir a una o más aplicaciones de la mejor manera posible. Los datos están almacenados de modo que resulten independiente de los programas que los usan. Se emplean métodos bien determinados para incluir nuevos datos y para modificar o extraer los datos almacenados.



Base de Datos

- Conjunto de datos operativos almacenados a los que acceden los sistemas de aplicación de una empresa.



Archivos vs. Bases de Datos

| Archivos | | Bases de Datos |
|-----------------|--|---|
| Entidades Tipo | tienen registros de una sola entidad tipo | tienen datos de varias entidades tipo |
| Interrelaciones | el sistema no interrelaciona archivos | el sistema tiene previstas herramientas para interrelacionar entidades |
| Redundancia | se crean archivos a la medida de cada aplicación, con todos los datos necesarios, aunque algunos sean redundantes respecto de otros archivos | todas las aplicaciones trabajan con la misma Base de Datos y la integración de los datos es básica, de modo que se evita la redundancia |
| Usuarios | sirven para un solo usuario o una sola aplicación. Dan una sola visión del mundo real | es compartida por muchos usuarios de distintos tipos. Ofrece varias visiones del mundo real |

Evolución de los SGBD

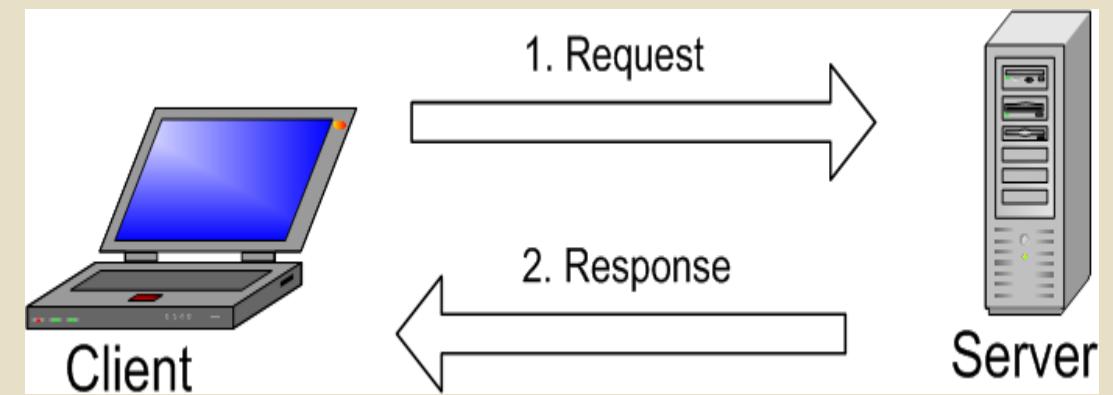
- 60-70: Sistemas Centralizados (Mainframe)
- Terminales con teclado, conectadas mediante línea telefónica
- SGBD ligados a software de comunicaciones y gestión de transacciones
- Programación en Cobol o PL/I, con subrutinas especializadas y complejas para tratar las BD (requerían conocimiento físico)
- Años 80: SGBD relacionales
- La aparición de los SGBD relacionales supone un avance importante para facilitar la programación de aplicaciones con BD y para conseguir que los programas sean independientes de los aspectos físicos de la BD
- 86: Se estandariza el SQL

Evolución de los SGBD

- Años 90: Distribución, C/S y 4GL
- Numerosas bases (departamentales) y varios SGBD en la empresa
- La necesidad de tener una visión global de la empresa y de interrelacionar diferentes aplicaciones que utilizan BD diferentes, junto con la facilidad que dan las redes para la intercomunicación entre computadoras, ha conducido a los SGBD actuales, que permiten que un programa pueda trabajar con diferentes BD como si se tratase de una sola. Es lo que se conoce como base de datos distribuida.

C/S

- La tecnología que se utiliza habitualmente para distribuir datos es la que se conoce como entorno (o arquitectura) cliente/servidor (C/S). Todos los SGBD relacionales del mercado han sido adaptados a este entorno.
- La idea del C/S es sencilla. Dos procesos diferentes, que se ejecutan en un mismo sistema o en sistemas separados, actúan de forma que uno tiene el papel de cliente o peticionario de un servicio, y el otro el de servidor o proveedor del servicio.





Tendencias actuales

- Hoy día, los SGBD relacionales están en plena transformación para adaptarse a tres tecnologías de éxito reciente, fuertemente relacionadas: la multimedia, la de orientación a objetos (OO) e Internet y la web.
- Manejos de grandes volúmenes de información sin estructura (Big Data)

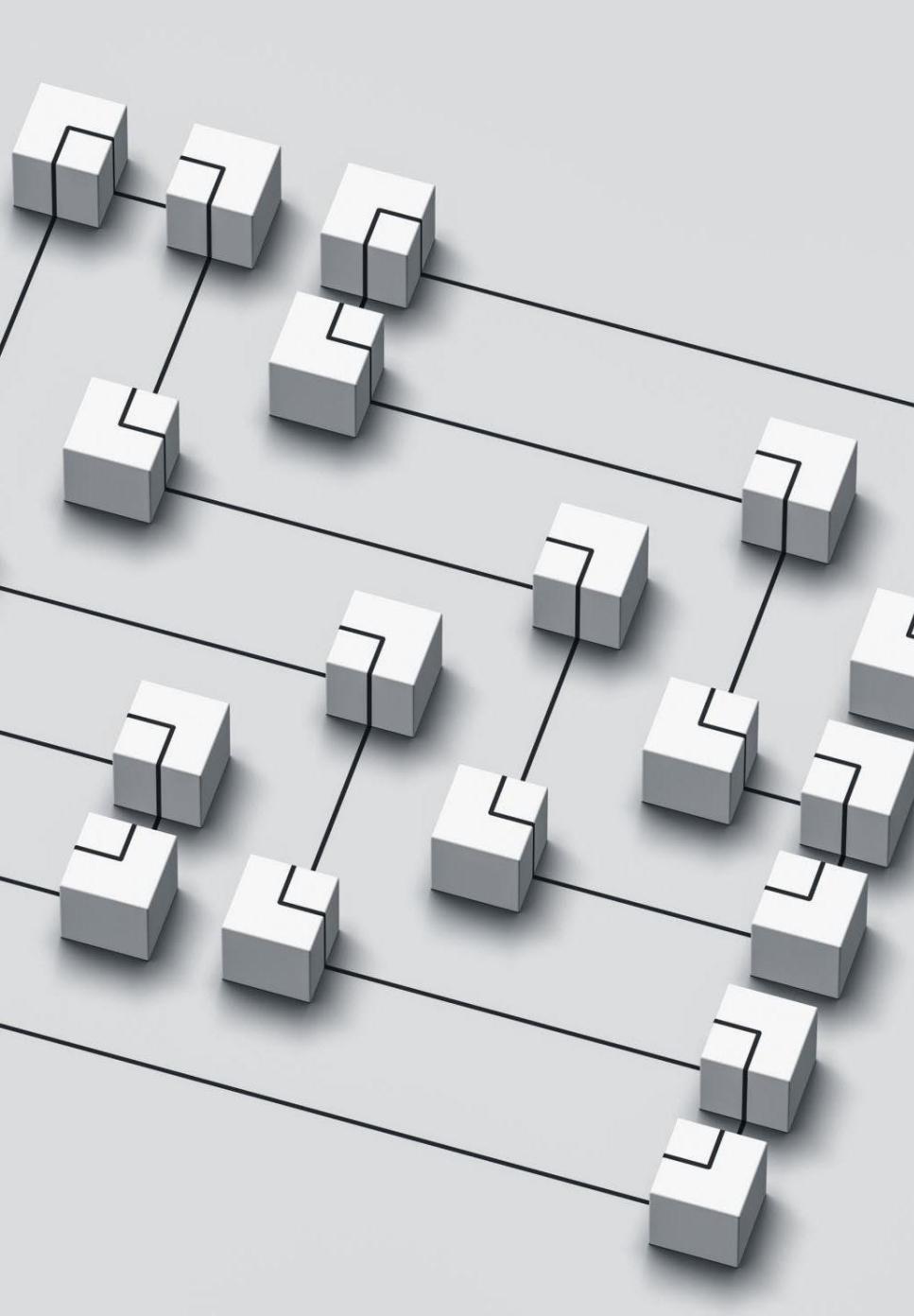
Objetivos y servicios de los SGBD

- Consultas no predefinidas y complejas (Ad hoc)
- Flexibilidad e Independencia (abstraerse de la implementación física)
- Problemas de redundancia
 - Inconsistencia
 - El SGBD debe permitir que el diseñador defina datos redundantes, pero entonces tendría que ser el mismo SGBD el que hiciese automáticamente la actualización de los datos en todos los lugares donde estuviesen repetidos.
- Integridad de los datos
 - Integridad de entidad / Integridad referencial / Reglas
- Conurrencia de usuarios
 - Transacciones *
- Seguridad



Concurrencia de usuarios

- Un objetivo fundamental de los SGBD es permitir que varios usuarios puedan acceder concurrentemente a la misma Base de Datos.



Transacción

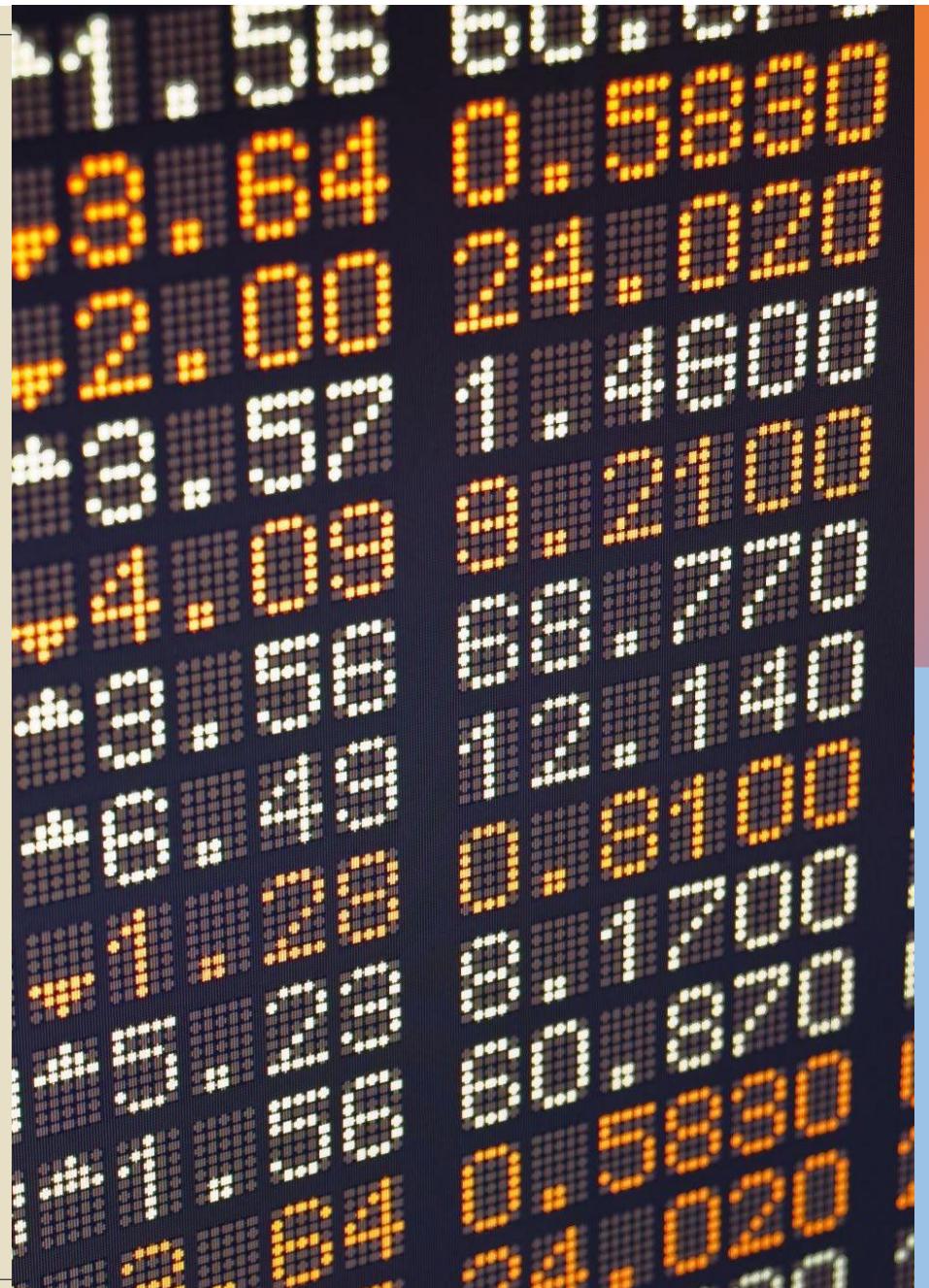
- Denominamos transacción de Base de Datos o, simplemente, transacción un conjunto de operaciones simples que se ejecutan como una unidad.
- Los SGBD deben conseguir que el conjunto de operaciones de una transacción nunca se ejecute parcialmente.
- O se ejecutan todas, o no se ejecuta ninguna.

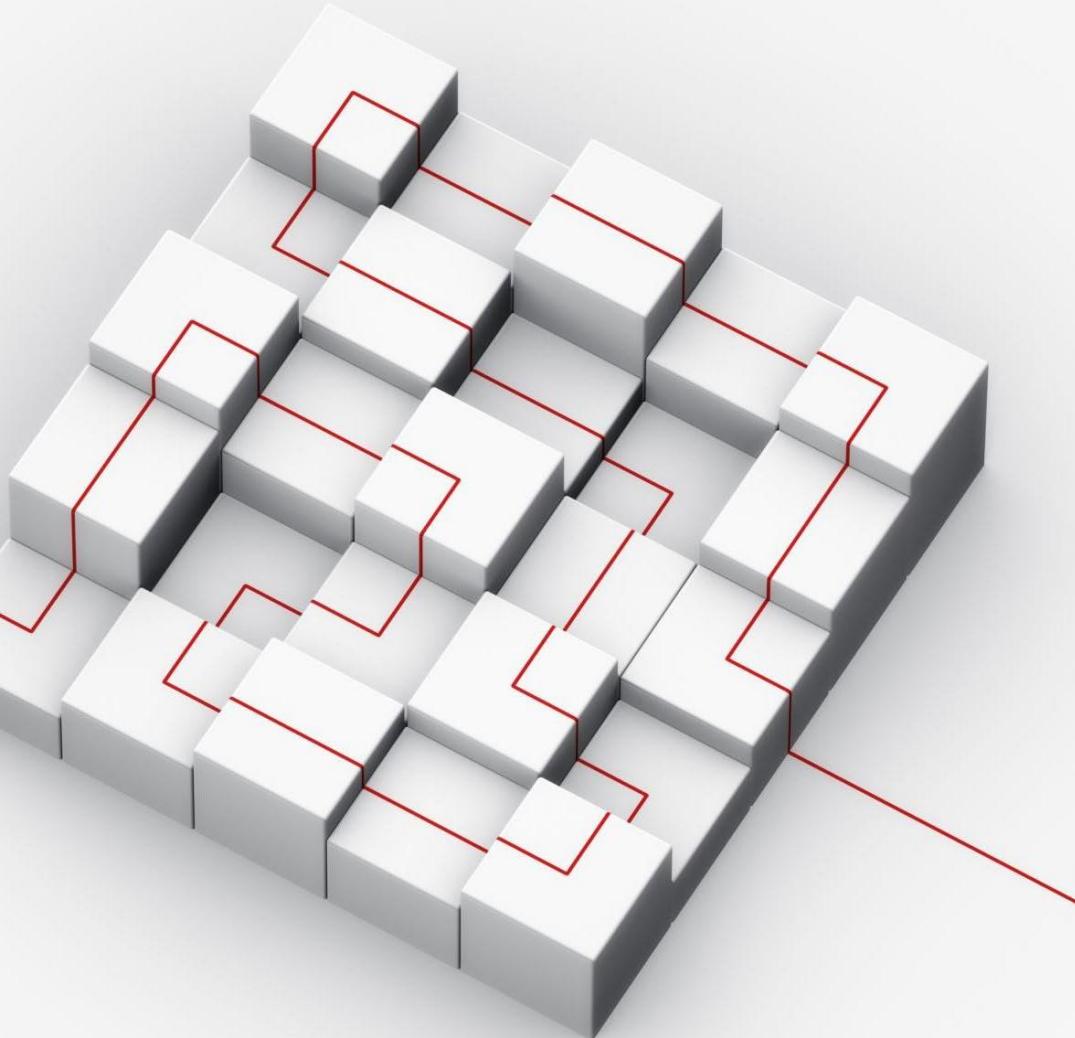
Propiedades ACID

- Atomicidad.
- Consistencia.
- Aislamiento. (Isolation)
- Durabilidad.

Arquitectura de los SGBD

- Esquemas y Niveles
 - Los SGBD necesitan que les demos una descripción o definición de la BD. Esta descripción recibe el nombre de esquema de la BD, y los SGBD la tendrán continuamente a su alcance.
 - Nivel Lógico: oculta los detalles de cómo se almacenan los datos, cómo se mantienen y cómo se accede físicamente a ellos. En este nivel sólo se habla de entidades, atributos y reglas de integridad
 - Nivel Físico: índices, en qué discos se almacenan los datos, tamaño de página, etc.





ANSI/SPARC

- De acuerdo con la arquitectura ANSI/SPARC, debía haber tres niveles de esquemas (tres niveles de abstracción). La idea básica de ANSI/SPARC consistía en descomponer el nivel lógico en dos: el nivel externo y el nivel conceptual. Denominábamos nivel interno lo que aquí hemos denominado nivel físico
 - Nivel externo se sitúan las diferentes visiones lógicas que los procesos usuarios (programas de aplicación y usuarios directos) tendrán de las partes de la Base de Datos que utilizarán. Estas visiones se denominan esquemas externos.
 - Nivel conceptual hay una sola descripción lógica básica, única y global, que denominamos esquema conceptual, y que sirve de referencia para el resto de los esquemas.
 - Nivel físico hay una sola descripción física, que denominamos esquema interno.

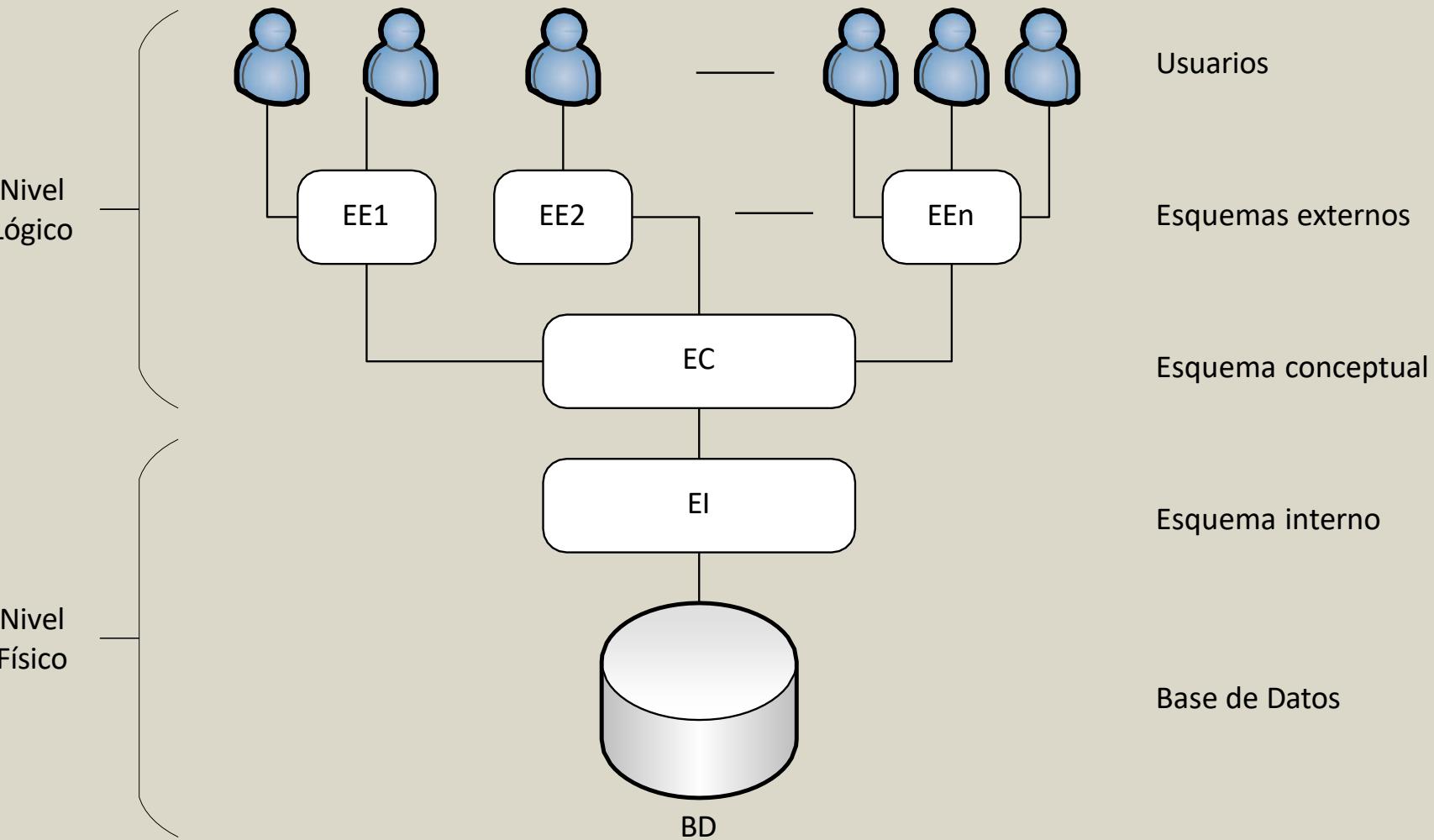
ANSI/SPARC

- Al definir un esquema externo, se citarán sólo aquellos atributos y aquellas entidades que interesen; los podremos redenominar, podremos definir datos derivados o redefinir una entidad para que las aplicaciones que utilizan este esquema externo crean que son dos, definir combinaciones de entidades para que parezcan una sola, etc.
- En el esquema conceptual se describirán las entidades tipo, sus atributos, las interrelaciones y las restricciones o reglas de integridad.
- El esquema interno o físico contendrá la descripción de la organización física de la Base de Datos: caminos de acceso (índices, hashing, punteros, etc.), codificación de los datos, gestión del espacio, tamaño de la página, etc.
- Para crear una Base de Datos hace falta definir previamente su esquema conceptual, definir como mínimo un esquema externo y, de forma eventual, definir su esquema interno. Si este último esquema no se define, el mismo SGBD tendrá que decidir los detalles de la organización física. El SGBD se encargará de hacer las correspondencias (mappings) entre los tres niveles de esquemas

ESQUEMAS Y NIVELES EN LOS SGBD RELACIONALES

- Se utiliza una terminología ligeramente diferente. No se separan de forma clara tres niveles de descripción. Se habla de un solo esquema –schema–, pero en su interior se incluyen descripciones de los tres niveles. En el schema se describen los elementos de aquello que en la arquitectura ANSI/SPARC se denomina esquema conceptual (entidades tipo, atributos y restricciones) y las vistas –view–, que corresponden aproximadamente a los esquemas externos.
- El estándar ANSI-ISO de SQL no habla en absoluto del mundo físico o interno; lo deja en manos de los SGBD relacionales del mercado. Sin embargo, estos SGBD proporcionan la posibilidad de incluir dentro del schema descripciones de estructuras y características físicas (índice, tablespace, cluster, espacios para excesos, etc.)

Esquemas y niveles



Independencia de los Datos

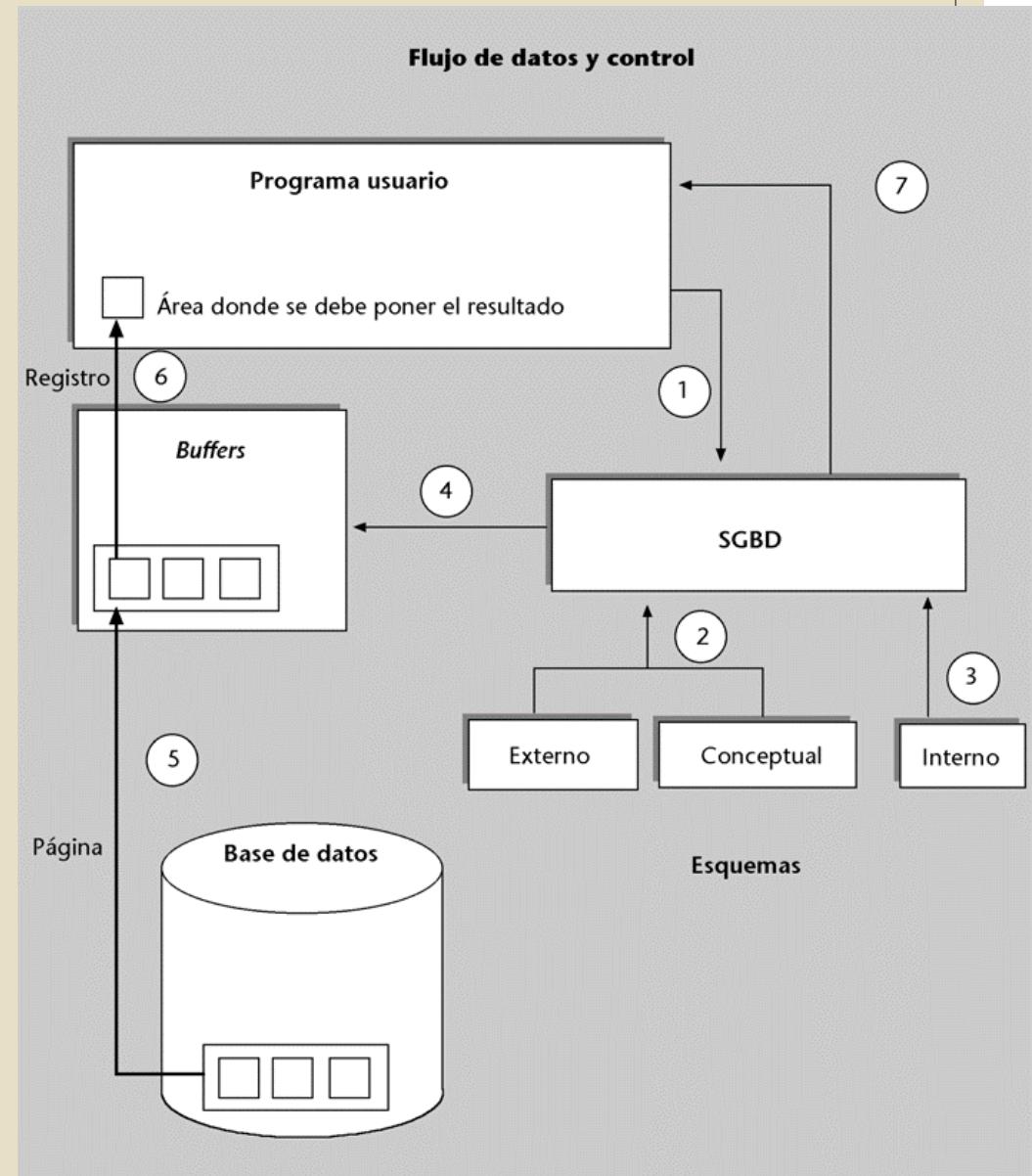
- Hay independencia física cuando los cambios en la organización física de la Base de Datos no afectan al mundo exterior (es decir, los programas usuarios o los usuarios directos).
- Hay independencia lógica cuando los usuarios no se ven afectados por los cambios en el nivel lógico.

Flujo de datos y control

- el SGBD, con la ayuda del SO, lee páginas (bloques) de los soportes donde está almacenada la Base de Datos física, y las lleva a un área de buffers o memorias caché en la memoria principal.
- El SGBD pasa registros desde los buffers hacia el área de trabajo del mismo programa

Flujo de datos y control

- Empieza con una llamada (1) del programa al SGBD. Se verifica que la sintaxis, autorización para hacerla, etc. El SGBD se basa (2) en el esquema externo con el que trabaja el programa y en el esquema conceptual.
- El SGBD determina, consultando el esquema interno (3), qué mecanismo debe seguir para responderla. Hay varias formas y diferentes caminos para responder a una consulta. Supongamos que ha elegido aplicar un hashing al valor del DNI, que es el parámetro de la consulta, y el resultado es la dirección de la página donde se encuentra (entre muchos otros) el registro del alumno buscado.
- El SGBD comprobará (4) si esta página ya se encuentra en el área de los buffers (tal vez como resultado de una consulta anterior de este usuario o de otro). Si no está, el SGBD, con la ayuda del SO, la busca en disco y la carga en los buffers (5). Si ya está, se ahorra el acceso a disco.
- La página deseada ya está en la memoria principal. El SGBD extrae, de entre los distintos registros que la página puede contener, el registro buscado, e interpreta la codificación y el resultado según lo que diga el esquema interno.
- El SGBD aplica a los datos las eventuales transformaciones lógicas que implica el esquema externo (tal vez cortando la dirección por la derecha) y las lleva al área de trabajo del programa (6).
- El SGBD retorna el control al programa (7) y da por terminada la ejecución de la consulta.



Modelos de BD

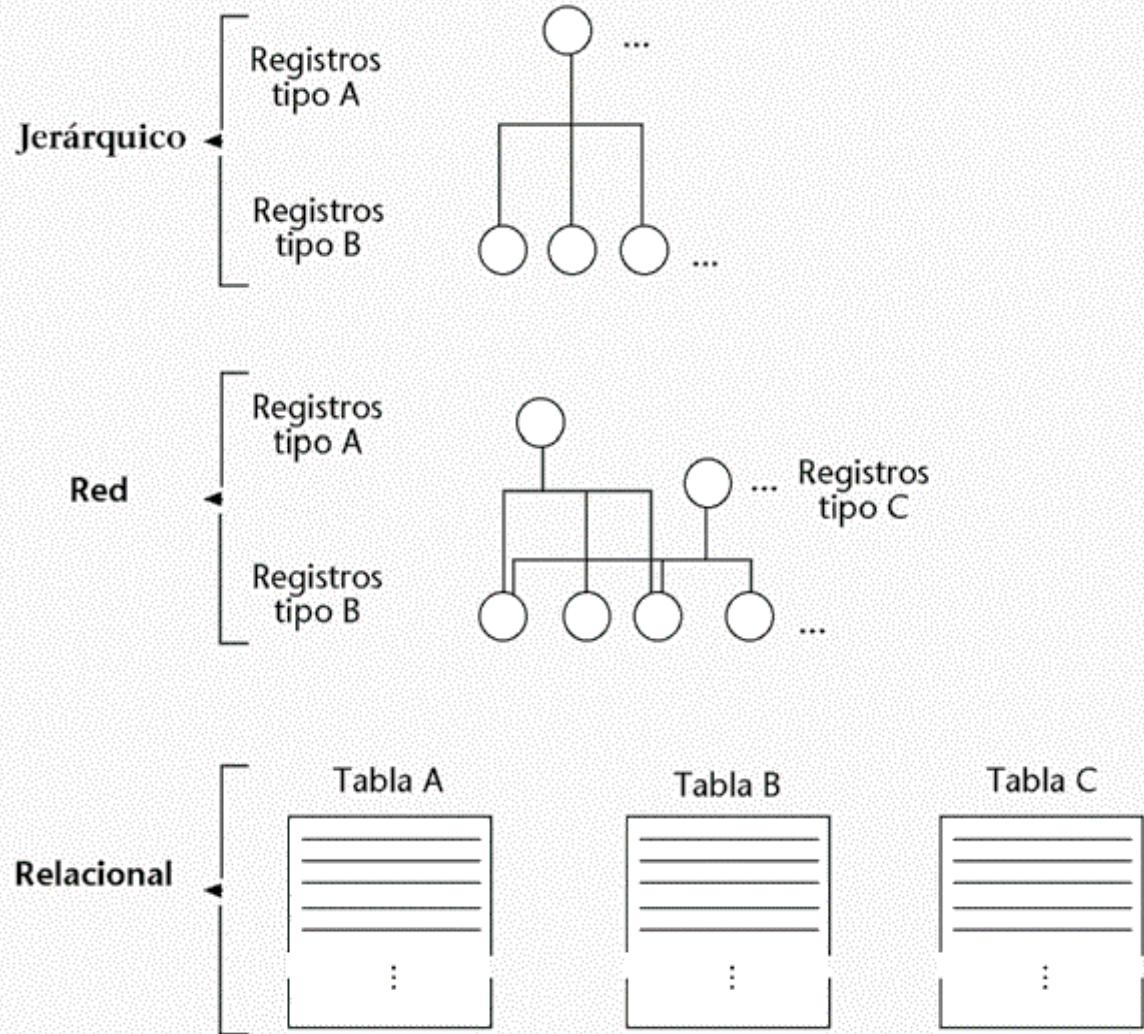
- Una BD es un modelo de la realidad
- En SGBD relacional el componente fundamental es la tabla
- Hay otros tipos de modelo de base de datos
 - Jerárquico
 - En Red
 - Relacional con objetos
- Todo modelo brinda
 - Estructuras (tablas, árboles, etc.)
 - Restricciones (reglas): dominios, claves, etc.
 - Operaciones



Evolución de los modelos

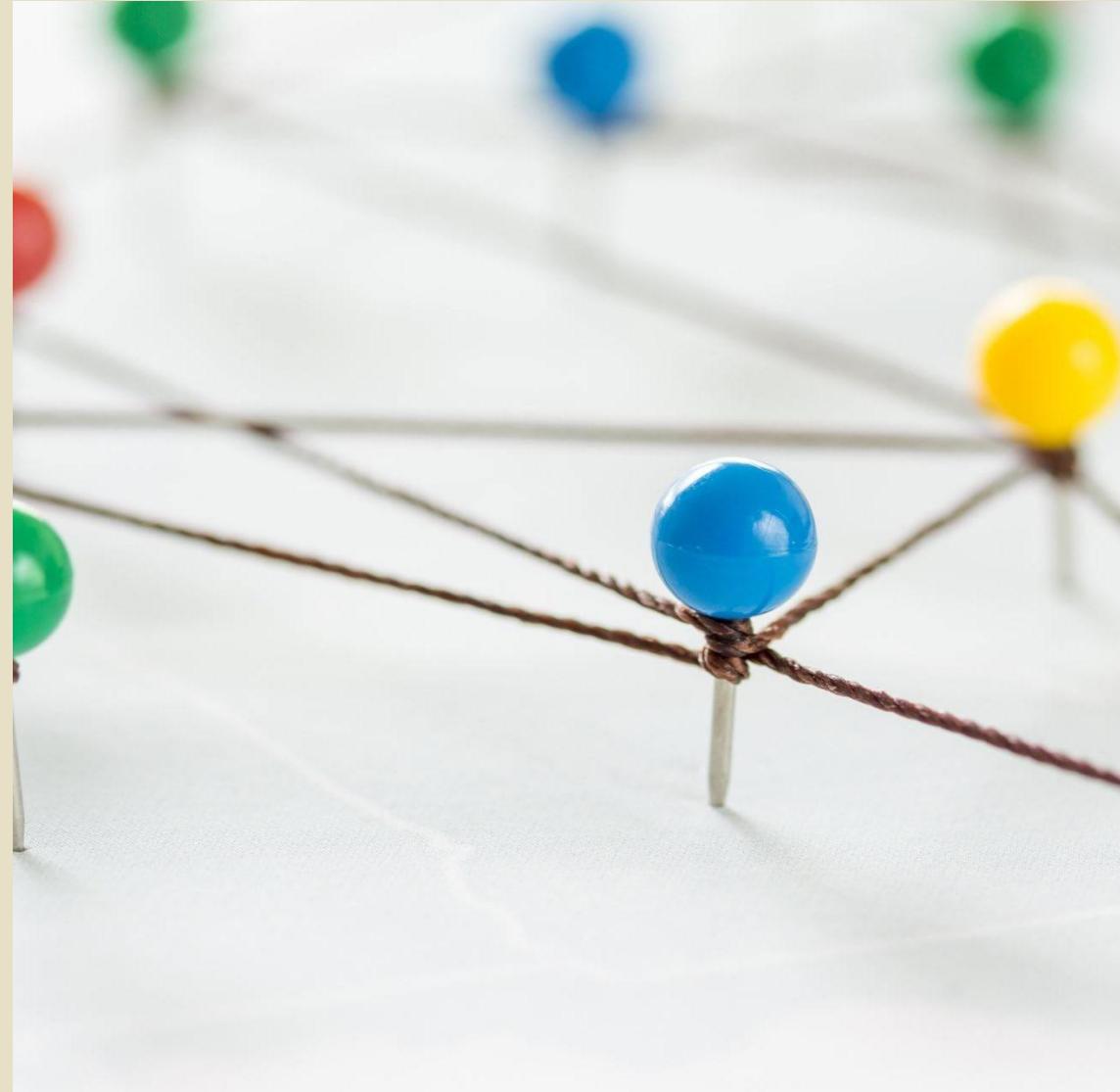
- Modelos prerelacionales
 - Años 60: Modelo jerárquico
 - Estructura: Registros interrelacionados en forma de árboles
 - IMD/DL1 de IBM
 - Principios 70: Modelo en red
 - Hay registros e interrelaciones, pero un registro no está limitado a ser “hijo” de un solo tipo de registro.
 - Comité CODASYL-DBTG propuso un estándar

Modelos de BD

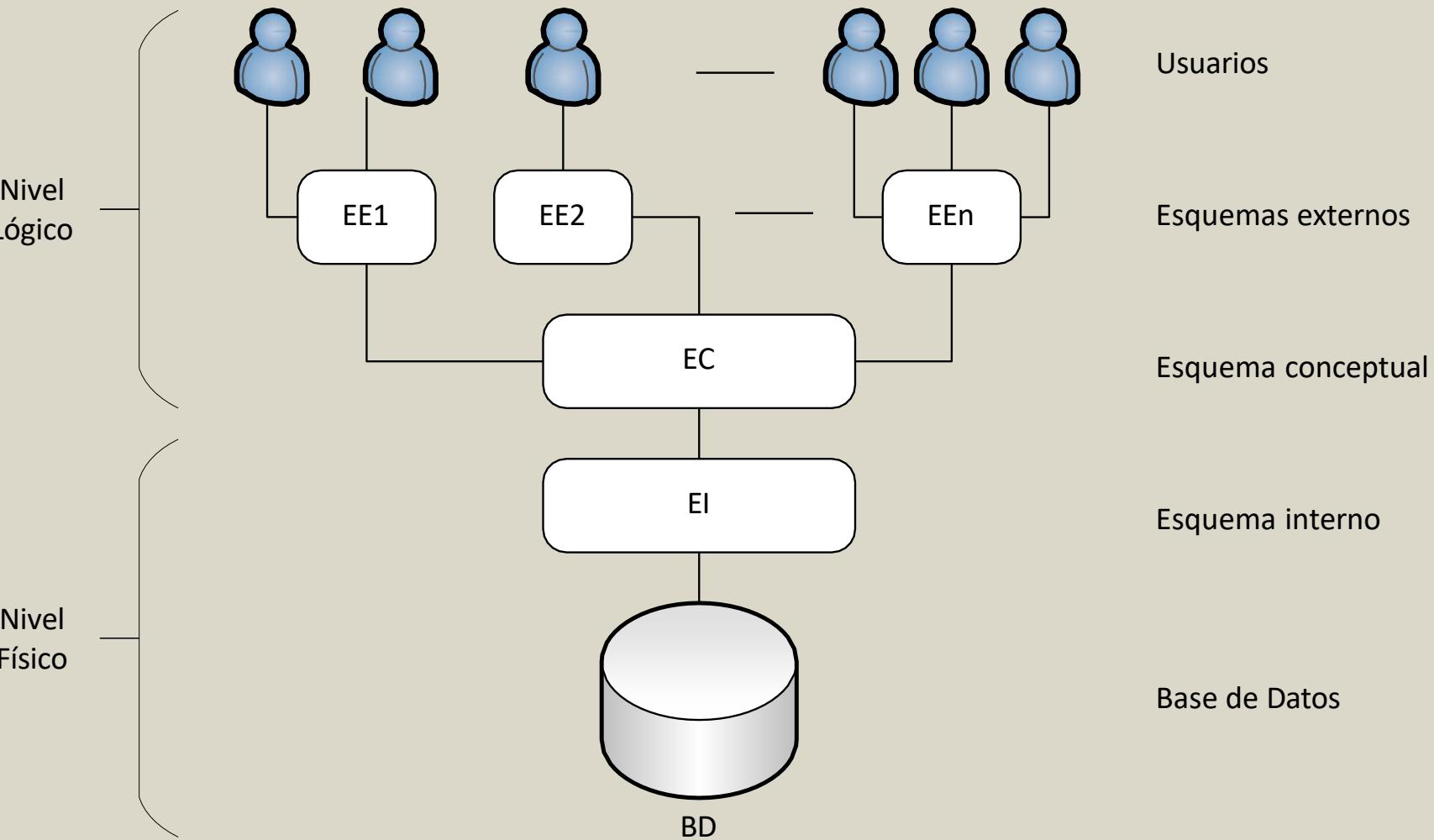


Evolución de los modelos

- Años 80: aparecen SGBD basados en el modelo relacional propuesto por Codd en 1969, con SQL como lenguaje nativo
 - Estructura: tablas (relaciones)
 - Se limita al nivel lógico (independencia física)
- En la actualidad: Surgen modelos relacionales con objetos
 - Ampliación del modelo relacional.
 - Añade posibilidad de que los tipos de datos sean tipos de datos abstractos (TAD)
 - Acerca los sistemas relacionales al paradigma de la OO



Esquemas y niveles

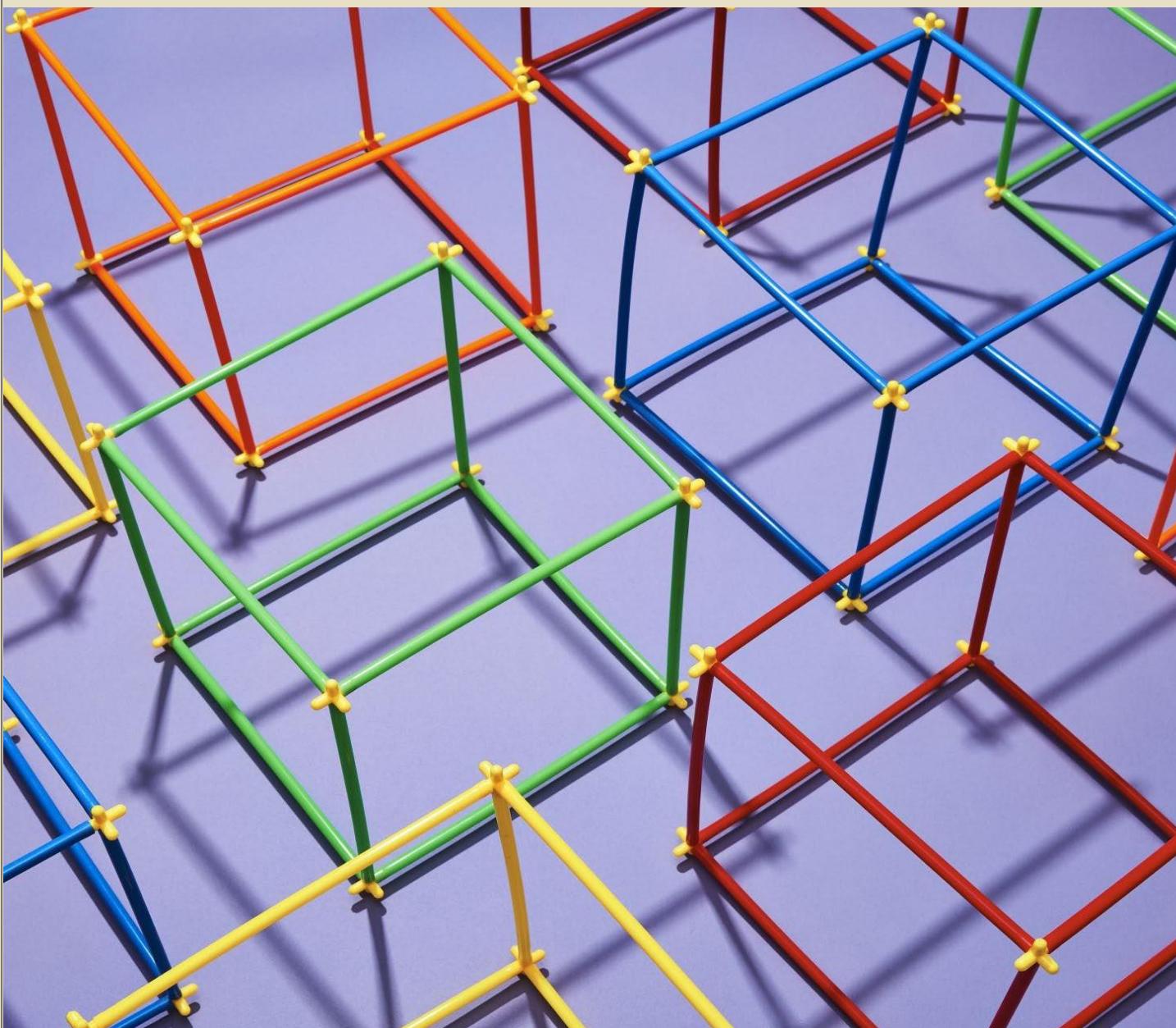


Archivos, estructuras y Operaciones básicas

- Un **archivo** es una colección de registros semejantes, guardados en dispositivos de almacenamiento secundario de la computadora (Wiederhold, 1983).
- Un **archivo** es una estructura de datos que recopila, en un dispositivo de almacenamiento secundario de una computadora, una colección de elementos del mismo tipo (De Giusti, 2001).
- Un **archivo** es una colección de registros que abarcan entidades con un aspecto común y originadas para algún propósito particular.

Administración de archivos

- Los archivos residen en **memoria secundaria** (por ejemplo, disco rígido)
- Los archivos son manejados por algoritmos.
 - Archivo físico: reside en la memoria secundaria y es administrado (ubicación, tipos de operaciones disponibles) por el sistema operativo.
 - Archivo lógico: es el archivo utilizado desde el algoritmo. Cuando un algoritmo necesita operar con un archivo, genera una conexión con el sistema operativo.



Tipos de archivos – Acceso a información

- Secuencial
- Secuencial indexado
- Directo

Proceso de búsqueda

- Considerar la cantidad de accesos a disco, y el proceso de verificación (comparación)
 - Cantidad de accesos (operación sobre memoria secundaria)
 - Cantidad de comparaciones (operación sobre memoria principal)



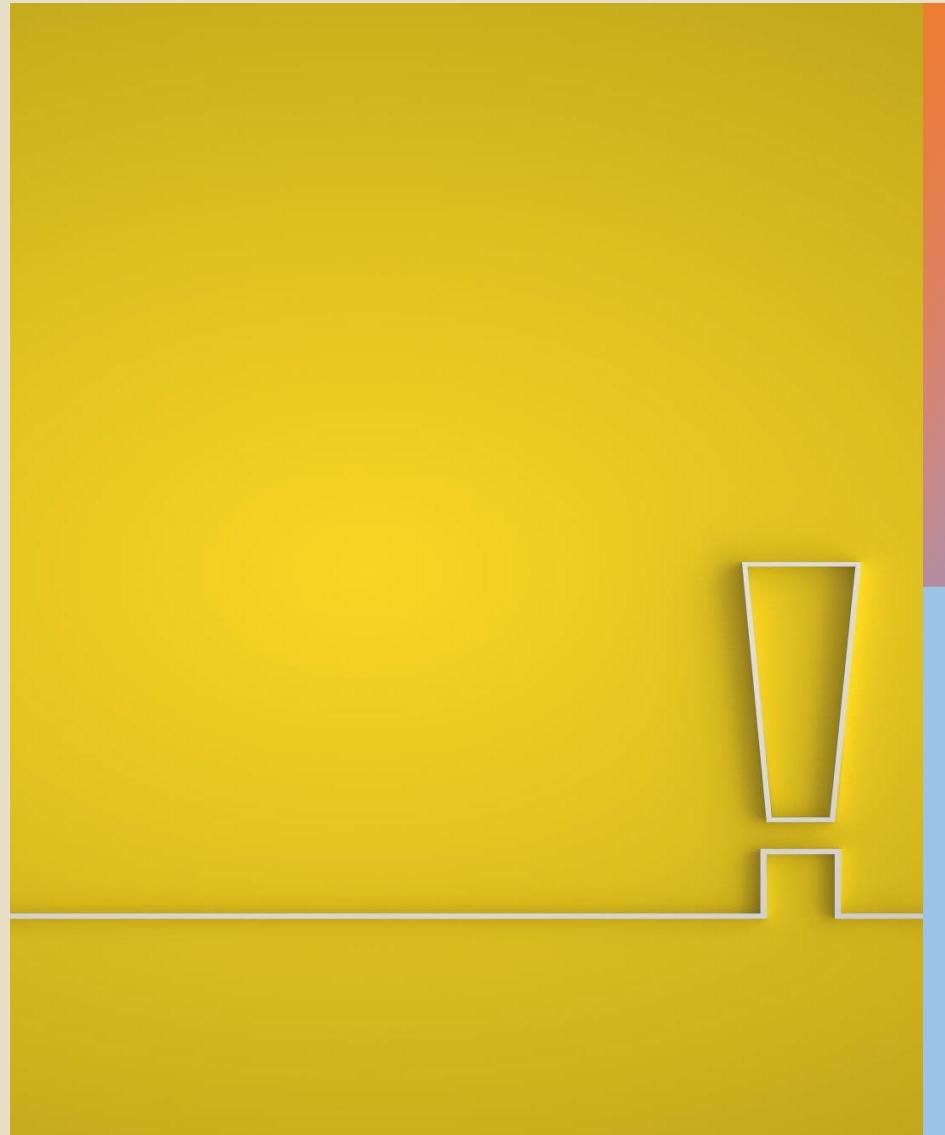
Secuencial

- Archivo sin orden preestablecido
- Para acceder un registro determinado, se deben visitar todos los registros previos en el orden que están almacenados
- La búsqueda se detiene en el registro que tiene el dato, o en el final del archive si el resultado no es exitoso.
 - Mejor caso -> 1 acceso
 - Peor caso -> N accesos
 - -----
 - Promedio -> $N/2$ accesos



Archivo físicamente ordenado

- El argumento de búsqueda coincide con el criterio de ordenación
- El proceso de búsqueda se detiene cuando el dato es “mayor” al buscado (no necesito ir hasta el final del archivo)
- La búsqueda sigue siendo secuencial y de orden N





Archivo físicamente ordenado con registros de longitud fija

- La primera comparación es contra el registro del medio del archivo ($NRR=N/2$)
- Si el registro no contiene el dato buscado, se descarta la mitad mayor o menor según corresponda, y se busca el registro del medio de la mitad que queda.
- Este proceso se repite hasta encontrar el valor buscado.

Archivo físicamente ordenado con registros de longitud fija

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 3 | 4 | 5 | 8 | 11 | 15 | 18 | 21 | 25 | 29 | 30 | 40 | 43 | 45 | 56 | 67 | 87 | 88 | 90 |
|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|

Supongamos que tenemos 20 registros, y estamos buscando el elemento con valor 15.

Se evalúa primero la posición 10, que tiene el elemento 25

| | | | | | | | | |
|---|---|---|---|---|----|----|----|----|
| 1 | 3 | 4 | 5 | 8 | 11 | 15 | 18 | 21 |
|---|---|---|---|---|----|----|----|----|

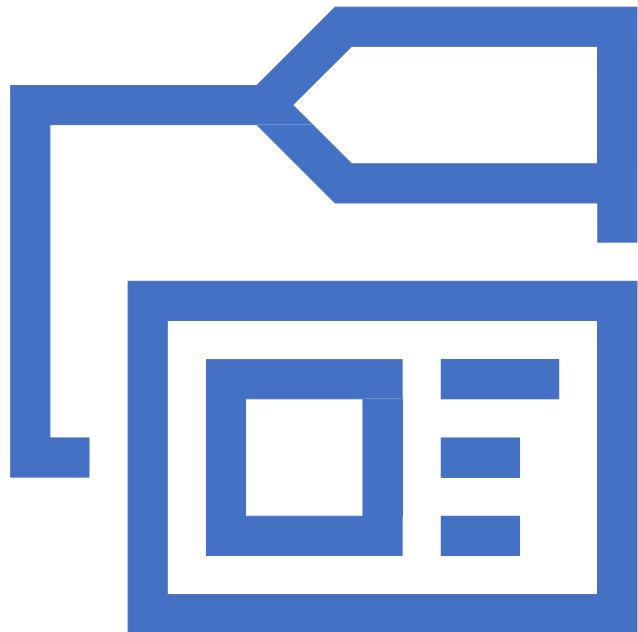
Se evalúa ahora la posición 5, que tiene el elemento 8

| | | | |
|----|----|----|----|
| 11 | 15 | 18 | 21 |
|----|----|----|----|

Se evalúa ahora la posición 3, que tiene el elemento 18

| | |
|----|----|
| 11 | 15 |
|----|----|

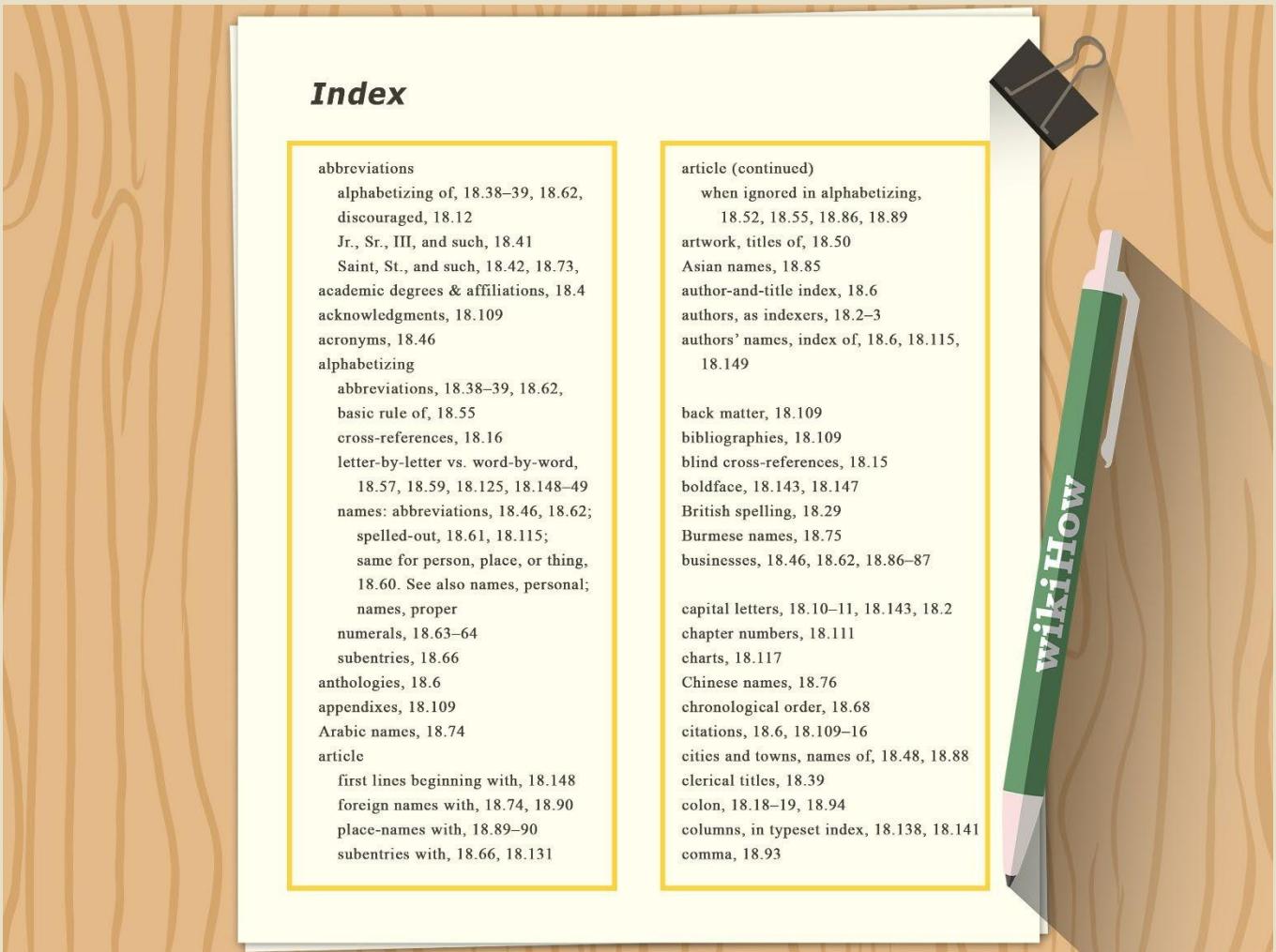
Se evalúa ahora la posición 2, que tiene el elemento buscado



Búsqueda Binaria

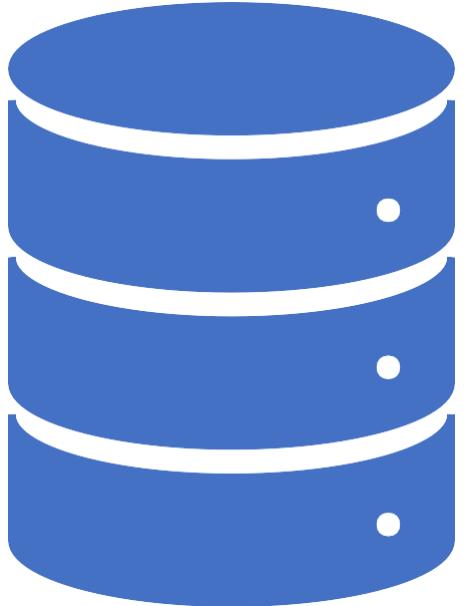
- En general, en un archivo de N registros se realiza a lo sumo $\log_2(N)+1$ comparaciones y en promedio $(\log_2(N)+1)/2$
- Por lo tanto, el orden es de $\log_2(N)$, una mejora sustancial con respecto al secuencial
- Como contra, hay que mantener el archivo ordenado, y el tamaño fijo puede implicar desperdicio de espacio.

Índice



2. m. En un libro u otra publicación, lista ordenada de los capítulos, artículos, materias, voces, etc., en él contenidos, con indicación del lugar donde aparecen.

SIN.: [glosario](#), [nomenclátor](#).

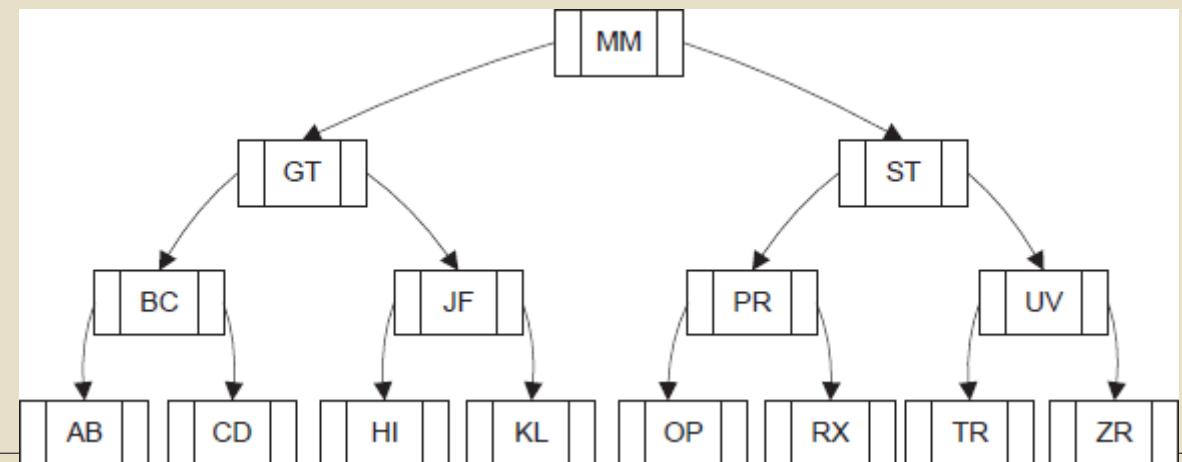


Índice

- Un índice es una estructura de datos adicional que permite agilizar el acceso a la información almacenada en un archivo. En dicha estructura se almacenan las claves de los registros del archivo, junto a la referencia de acceso a cada registro asociado a la clave. Es necesario que las claves permanezcan ordenadas.
- Esta estructura de datos es otro archivo con registros de longitud fija, independientemente de la estructura del archivo original. La característica fundamental de un índice es que posibilita imponer orden en un archivo sin que realmente este se reacomode.
- El índice es un nuevo archivo ordenado, con registros de longitud fija, con la diferencia de que contiene solo un par de datos del archivo original, y, por ende, es mucho más pequeño. En el peor caso, se plantea de nuevo la situación de ordenar este nuevo archivo en memoria secundaria. Una vez disponible el índice, la búsqueda de un dato se realiza primero en el índice (a partir de la clave), de allí se obtiene la dirección efectiva del archivo de datos y luego se accede directamente a la información buscada.

Árboles

- Las estructuras tipo árbol presentan algunas mejoras tanto para la búsqueda como para el mantenimiento del orden de la información
- Un **árbol binario** es una estructura de datos dinámica no lineal, en la cual cada nodo puede tener a lo sumo dos hijos.
- La estructura de datos árbol binario en general tiene sentido cuando está ordenado.



Árbol binario

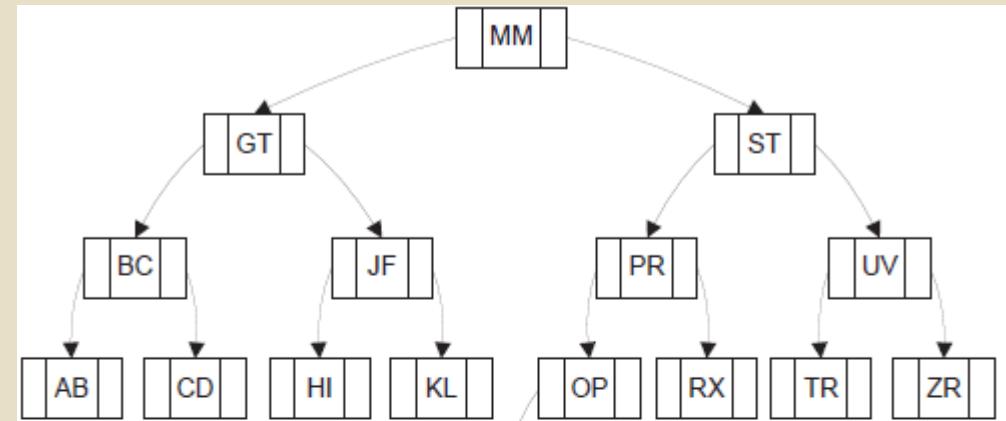
- La búsqueda en este tipo de estructuras se realiza a partir del nodo raíz y se recorre, explorando hacia los nodos hoja.
- Se chequea un nodo; si es el deseado, la búsqueda finaliza o, en su defecto, se decide si la búsqueda continúa a izquierda o a derecha descartando la mitad de los elementos restantes.
- La búsqueda de información en un árbol binario es de orden logarítmico. Encontrar un elemento es del orden $\log_2 (N)$, siendo N la cantidad de elementos distribuidos en el árbol.

Árbol binario

- Una ventaja de la organización mediante árboles binarios está dada en la inserción de nuevos elementos.
- Mientras que un archivo se desordena cuando se agrega un nuevo dato, si la organización se realiza con la política de árbol binario, la operatoria resulta más sencilla en términos de complejidad computacional.

Problema con los árboles binarios

- ¿Qué pasa si al árbol de ejemplo que vimos antes le agregamos las claves NI, OC, RN, OA NZ?
- El árbol queda desbalanceado
- La performance de búsqueda ya no puede considerarse de orden logarítmico



Árboles AVL

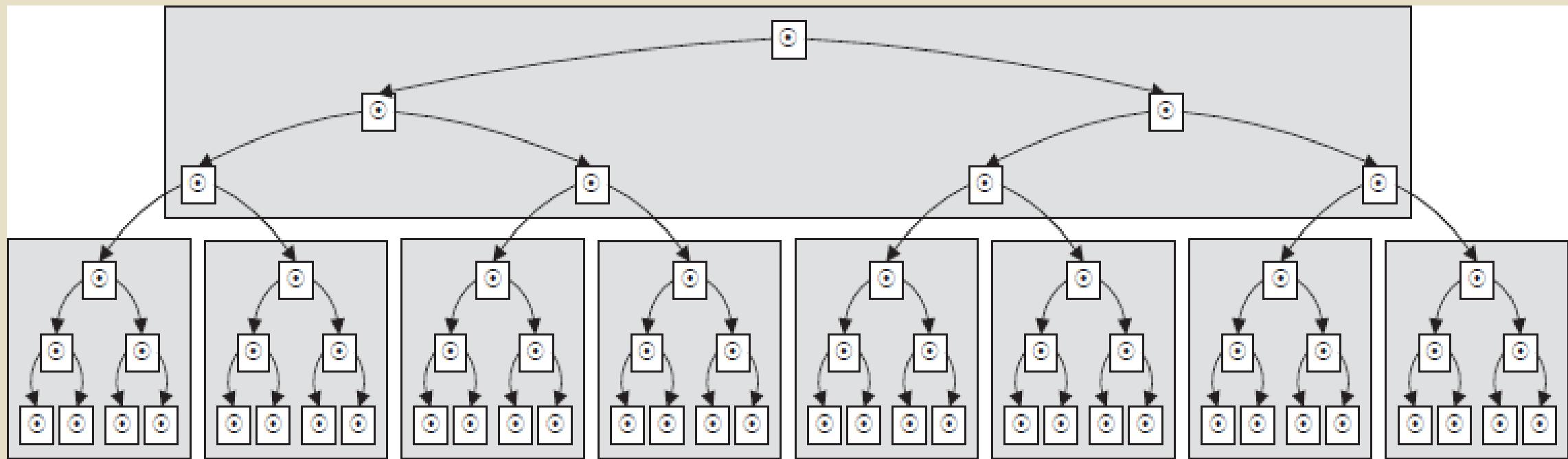
- Los árboles balanceados en altura son árboles binarios cuya construcción se determina respetando un precepto muy simple: la diferencia entre el camino más corto y el más largo entre un nodo terminal y la raíz no puede diferir en más que un determinado delta, y dicho delta es el nivel de balanceo en altura del árbol.
- Otra definición posible plantea que un árbol balanceado en altura es aquel en el que la diferencia máxima entre las alturas de cualquiera de dos subárboles que tienen raíz común no supera un delta determinado.

Árboles AVL

- Es un árbol balanceado en altura donde el delta determinado es uno, es decir, el máximo desbalanceo posible es uno.
- En el caso del árbol anterior, al insertar la clave NI, el árbol aún se mantiene balanceado como AVL(1). Sin embargo, cuando se intenta insertar OC, se viola el precepto establecido. Se debe llevar a cabo, en esta situación, un rebalanceo de este árbol. Se debe comprender que, si bien este algoritmo es relativamente sencillo de implementar, los costos computacionales de acceso a disco aumentan considerablemente, por lo que su implantación deja de ser viable.
- Entonces, se dispone de una estructura capaz de mantener el balanceo acotado, pero asumiendo mayores costos en las operaciones de inserción y borrado. Por lo tanto, los árboles binarios y los AVL no representan una solución viable para los índices del archivo de datos.

Paginación de árboles binarios

- Se divide en páginas, es decir, se pagina, y cada página contiene un conjunto de nodos, los cuales están ubicados en direcciones físicas cercanas

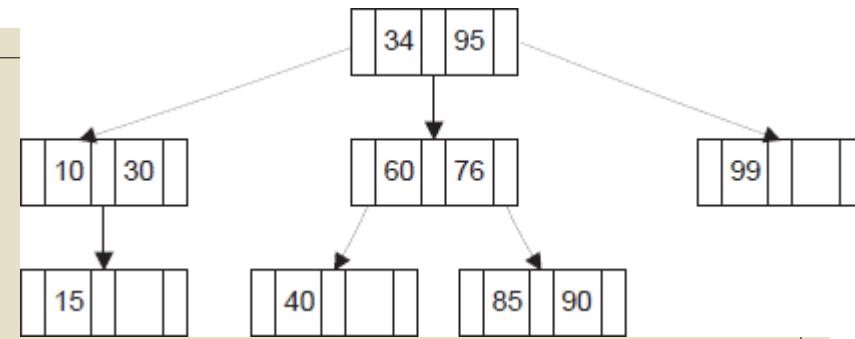


Paginación de árboles binarios

- Al dividir un árbol binario en páginas, es posible realizar búsquedas más rápidas de datos almacenados en memoria secundaria.
- En el ejemplo, cada página contiene siete elementos, pero es solo para presentar la situación. En una situación más realista, una página puede contener 255 o 511 nodos del árbol, con lo cual la altura relativa decrece rápidamente, permitiendo recuperar un elemento del árbol en muchos menos accesos que los planteados originalmente ($\log_2(N)$).
- Para analizar la performance resultante, deberían tenerse en cuenta la cantidad de nodos que caben en una página.
- Suponiendo que en un buffer caben 255 elementos, el tamaño de cada página sería entonces de 255 nodos, resultando la performance final de búsqueda del orden de $\log_{256}(N)$, es decir, $\log_k+1(N)$, siendo N la cantidad de claves del archivo y K la cantidad de nodos por página.

Árboles multicamino

- Anteriormente se definió a un árbol binario como una estructura de datos en la cual cada nodo puede tener cero, uno o dos hijos. Siguiendo esta línea se podría definir a un árbol ternario como aquel en el cual cada nodo puede tener cero, uno, dos o tres hijos.
- En la imagen, en el nodo raíz hay dos elementos, el 34 y el 95; a la izquierda del 34, como primer puntero se referencia a un nodo que contiene los elementos menores que 34. Los valores comprendidos entre 34 y 95 se referencian por un puntero existente entre ambos, en tanto que a la derecha del 95 se apunta hacia el nodo que contiene los elementos mayores.



Árboles multicamino

- Un árbol multicamino es una estructura de datos en la cual cada nodo puede contener k elementos y $k+1$ hijos.
- Se define el concepto de orden de un árbol multicamino como la máxima cantidad de descendientes posibles de un nodo.
- Un árbol multicamino representa otra forma de resolver el concepto de página vertido anteriormente. En este caso, el orden del árbol dependerá del tamaño de la página y de los elementos que se coloquen en ella.

Árboles B (balanceados)

- Los árboles B son árboles multicamino con una construcción especial que permite mantenerlos balanceados a bajo costo.
- Un árbol B de orden M posee las siguientes propiedades básicas:
 1. Cada nodo del árbol puede contener, como máximo, M descendientes y M-1 elementos.
 2. la raíz no posee descendientes directos o tiene al menos dos.
 3. Un nodo con x descendientes directos contiene x-1 elementos.
 4. Los nodos terminales (hojas) tienen, como mínimo, $[M/2] - 1$ elementos, y como máximo, M-1 elementos.
 5. los nodos que no son terminales ni raíz tienen, como mínimo, $[M / 2]$ elementos.
 6. Todos los nodos terminales se encuentran al mismo nivel.

Otros árboles B

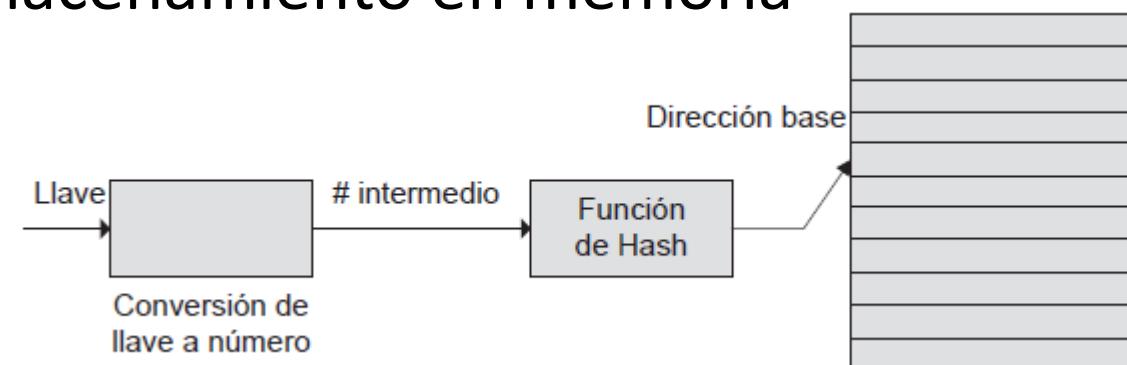
- Existen además las estructuras B* (B Estrella) y B+, que son variantes de los árboles B que se utilizan para mejorar aún más el rendimiento en ciertos escenarios
- Los árboles B* permiten la reorganización de nodos internos sin afectar la estructura de búsqueda
- Los árboles B+ tienen un diseño que permite inserciones y eliminaciones eficientes de elementos

Dispersión (hashing)

- Los archivos directos son almacenados en disco en diferentes formas respecto de los archivos serie.
- Debe permitir una rápida recuperación de la información contenida en el archivo, que es el objetivo principal que se persigue con un archivo directo.
- El mecanismo que trata de asegurar una recuperación rápida de registros, en un solo acceso promedio, lleva el nombre de dispersión o hashing.

Conceptos de dispersión

- Técnica para generar una dirección base única para una clave dada. La dispersión se usa cuando se requiere acceso rápido mediante una clave.
- Técnica que convierte la clave asociada a un registro de datos en un número aleatorio, el cual posteriormente es utilizado para determinar dónde se almacena dicho registro.
- Técnica de almacenamiento y recuperación que usa una función para mapear registros en direcciones de almacenamiento en memoria secundaria.



Conceptos de dispersión

- La técnica de dispersión presenta una serie de atributos. Estos pueden resumirse en los siguientes:
 - No se requiere almacenamiento adicional. Esto significa que cuando se elige la opción de dispersión como método de organización de archivos, es el archivo de datos el que resulta disperso. Esta dispersión se genera a partir de la clave o llave primaria del archivo, y el registro que contiene la información relacionada con la clave es ubicado en el espacio resultante de aplicar la función de hash. Así, no es necesario tener una estructura auxiliar que actúe como soporte para poder acceder rápidamente a la información.
 - Facilita la inserción y eliminación rápida de registros en el archivo.
 - Localiza registros dentro del archivo con un solo acceso a disco. Como corolario del punto anterior, otra ventaja del método de dispersión consiste en ubicar cada elemento de datos, en promedio, con un acceso a disco.

Limitaciones de la dispersión

- No es posible aplicar la técnica de dispersión en archivos con registros de longitud variable. Esto se debe a que cada dirección física obtenida debe tener capacidad para almacenar un registro de tamaño conocido.
- No es posible obtener un orden lógico de los datos. Utilizando índices como metodología de acceso a datos, no solo la búsqueda es eficiente, sino que además presenta la característica de mantener los registros ordenados. Bajo la técnica de hashing o dispersión no es posible preservar la propiedad de orden. Los registros son esparcidos en el archivo de datos, de acuerdo con la dirección que se obtiene con el uso de la función de hash.
- No es posible tratar con claves duplicadas. Así, no es aplicable la función de hash sobre una clave secundaria. Si se analiza con detalle dicha situación, se podrá observar a qué se debe este impedimento. Una clave secundaria puede repetirse; por lo tanto, dos registros diferentes con la misma clave secundaria, aplicando la función de hash, tendrían como resultado la misma dirección de memoria.

Tipos de dispersión

- Se denomina **hashing con espacio de direccionamiento estático** a aquella política donde el espacio disponible para dispersar los registros de un archivo de datos está fijado previamente. Así, la función de hash aplicada a una clave da como resultado una dirección física posible dentro del espacio disponible para el archivo.
- Se denomina **hashing con espacio de direccionamiento dinámico** a aquella política donde el espacio disponible para dispersar los registros de un archivo de datos aumenta o disminuye en función de las necesidades de espacio que en cada momento tiene el archivo. Así, la función de hash aplicada a una clave da como resultado un valor intermedio, que será utilizado para obtener una dirección física posible para el archivo. Estas direcciones físicas no están establecidas a priori y son generadas de manera dinámica

Función de hash

- Una **función de hash** o **dispersión** es una función que transforma un valor, que representa una llave primaria de un registro, en otro valor dentro de un determinado rango, que se utiliza como dirección física de acceso para insertar un registro en un archivo de datos.



Modelos semánticos y funcionales

- Se utilizan en etapa de análisis y diseño
- Permiten describir estructuras de información del mundo real, la semántica y las interrelaciones
- Se utilizan herramientas CASE (Computer Aided Software Engineering)
- El más utilizado es el ER (Entity-relationship)

Lenguajes y usuarios

- Usuarios informáticos expertos:
procesos complejos – lenguajes
complejos
 - Usuarios no informáticos ocasionales:
consultas – lenguaje sencillo
 - Usuarios no informáticos dedicados:
lenguajes eficientes y compactos
-
- SQL es el más utilizado en bases
relacionales



Lenguajes

- Hay lenguajes especializados en la escritura de esquemas; es decir, en la descripción de la Base de Datos. Se conocen genéricamente como DDL o data definition language. Incluso hay lenguajes específicos para esquemas internos, lenguajes para esquemas conceptuales y lenguajes para esquemas externos.

A blackboard filled with mathematical calculations, formulas, and diagrams. The board includes:

- A circle with center c and radius r .
- A right triangle with legs a and b , and hypotenuse c .
- An equation $\sqrt{a^2 + b^2} = c$.
- A diagram showing a shaded area with a boundary line labeled $x^2 + y^2 = r^2$.
- A system of equations:
$$\begin{cases} xy = c \\ cx - cy = ab \\ \pi = c \end{cases}$$
- An equation $B_1 = \frac{2x+3}{y} + \frac{a^2+b^2}{c} + \vec{x} \cdot \vec{s}$.
- An equation $new = 58^2 + nov (x^2 + 34x -$.
- A diagram with $x \rightarrow 2$ and $y = 14!$.
- An equation $\sum N_{50} \cdot x - \frac{1}{2} [964 + xg +$.
- A diagram of a circle with radius $r=4$.
- An equation $\beta = 9 + x^2 + y^2$.

Lenguajes

- Otros lenguajes están especializados en la utilización de la Base de Datos (consultas y mantenimiento). Se conocen como DML o data management language.
- Sin embargo, lo más frecuente es que el mismo lenguaje disponga de construcciones para las dos funciones, DDL y DML.

DML

- Lenguajes muy declarativos (o implícitos), con los que se especifica qué se quiere hacer sin explicar cómo se debe hacer.
- Lenguajes más explícitos o procedimentales, que nos exigen conocer más cuestiones del funcionamiento del SGBD para detallar paso a paso cómo se deben realizar las operaciones (lo que se denomina navegar por la Base de Datos).

The image shows a blackboard covered with handwritten mathematical work. At the top right, there's a diagram of a rectangle divided into four quadrants with labels \overrightarrow{P} , \overrightarrow{Q} , \overrightarrow{R} , and \overrightarrow{S} . Below it, a formula is given: $\sqrt{2456.96} = b + \frac{c}{b}$. To the left, there's a formula involving a function $D(x) = a + b + 4.31447$. Further down, there's a circle with radius c and a shaded sector. A system of equations is shown: $\begin{cases} xy = c \\ cx - cy = 35^2 \\ \pi r^2 = c \end{cases}$. On the left, there's a fraction $\frac{24+3x}{x} + \frac{d^2+3^2}{c} + \frac{\vec{x}}{\vec{y}}$. To the right, there's a complex expression involving $x^4 + 34x^2 + 1 = 58^2 + n^{av}$. Below that, there's a summation formula: $\sum_{x=2}^{u=14!} N^{50} \cdot x - \frac{1}{2} [964 + xg + \dots]$. At the bottom, there's a diagram of a triangle with vertices labeled A , B , and C , with a point P inside. A formula for β is given: $\beta = 9 + x^2 + y^2$.

DDL

- Declarativos por naturaleza

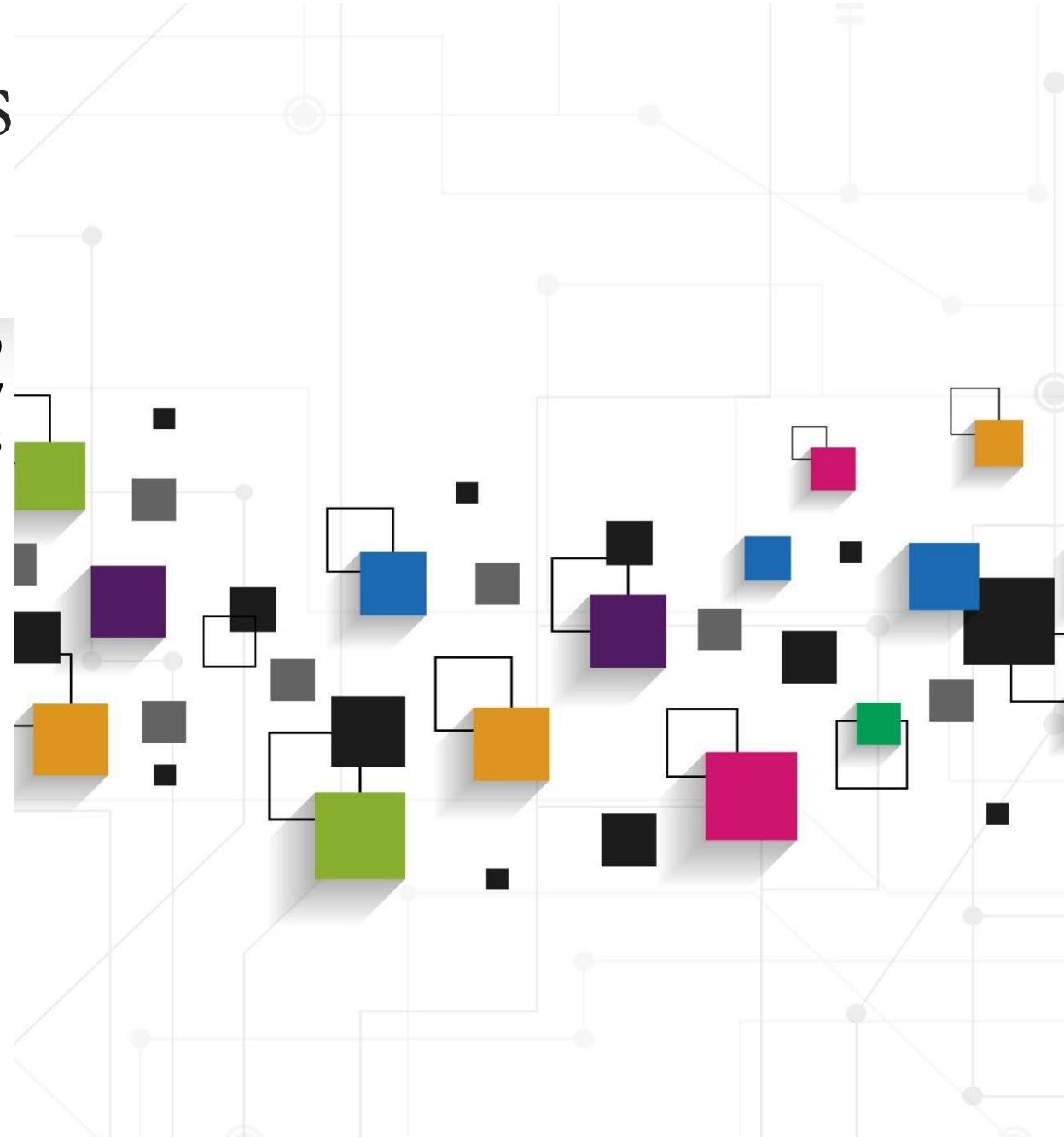
SQL

- Es básicamente declarativo, pero tiene posibilidades procedimentales



Herramientas Visuales

- Permiten usar las Bases de Datos siguiendo el estilo de diálogos con ventanas, iconos y ratón, puesto de moda por las aplicaciones Windows.
- No sólo son útiles a los usuarios no informáticos, sino que facilitan mucho el trabajo a los usuarios informáticos: permiten consultar y actualizar la Base de Datos, así como definirla y actualizar su definición con mucha facilidad y claridad.
- Por ejemplo: SSMS (SQL Server Management Studio), MySQL Workbench



Acceso a BD con lenguajes

- No suelen tener instrucciones para acceso a BD
 - Llamadas a funciones: librerías de funciones especializadas en BD (por ejemplo, las librerías ODBC). Incluir llamadas a las funciones deseadas dentro del programa escrito con el lenguaje habitual. Las funciones serán las que se encargarán de enviar las instrucciones (generalmente en SQL) en tiempo de ejecución al SGBD.
 - Lenguaje hospedado: Incluir directamente las instrucciones del lenguaje de Bases de Datos en nuestro programa. Exige utilizar un precompilador especializado que acepte en nuestro lenguaje de programación habitual las instrucciones del lenguaje de BD. Entonces SQL es el lenguaje hospedado o incorporado (embedded), y nuestro lenguaje de programación (Pascal, C, Cobol, etc.) es el lenguaje anfitrión (host).

Administración de BD

- Hay un tipo de usuario especial: el que realiza tareas de administración y control de la BD.
- Lleva a cabo una serie de funciones centralizadas de gestión y administración, para asegurar que la explotación de la BD es la correcta.
- Este conjunto de funciones se conoce con el nombre de administración de Bases de Datos (ABD), y los usuarios que hacen este tipo especial de trabajo se denominan administradores de Bases de Datos. (DBA)



Tareas del DBA

- Mantenimiento, administración y control de los esquemas. Comunicación de los cambios a los usuarios.
- Asegurar la máxima disponibilidad de los datos; por ejemplo, haciendo copias (back-ups), administrando diarios (journals o logs), reconstruyendo la Base de Datos, etc.
- Resolución de emergencias.
- Vigilancia de la integridad y de la calidad de los datos.
- Diseño físico, estrategia de caminos de acceso y reestructuraciones.
- Control del rendimiento y decisiones relativas a las modificaciones en los esquemas y/o en los parámetros del SGBD y del SO, para mejorarlo.
- Normativa y asesoramiento a los programadores y a los usuarios finales sobre la utilización de la Base de Datos.
- Control y administración de la seguridad: autorizaciones, restricciones, etc.

The background of the slide features a complex, abstract digital structure composed of numerous small, glowing blue and green dots arranged in a grid-like pattern, resembling binary code or a network. This structure is set against a dark blue gradient background with several larger, semi-transparent circular bokeh effects in shades of red, green, and blue.

Sistemas de Gestión de Base de Datos



Ranking de db-engines (Relacional)

include secondary database models

166 systems in ranking, February 2025

| Rank | | | DBMS | Database Model | Score | | |
|----------|----------|----------|----------------------|-------------------------|----------|----------|----------|
| Feb 2025 | Jan 2025 | Feb 2024 | | | Feb 2025 | Jan 2025 | Feb 2024 |
| 1. | 1. | 1. | Oracle | Relational, Multi-model | 1254.82 | -3.93 | +13.38 |
| 2. | 2. | 2. | MySQL | Relational, Multi-model | 999.99 | +1.84 | -106.67 |
| 3. | 3. | 3. | Microsoft SQL Server | Relational, Multi-model | 786.87 | -11.69 | -66.70 |
| 4. | 4. | 4. | PostgreSQL | Relational, Multi-model | 659.62 | -3.79 | +30.21 |
| 5. | 5. | ↑ 6. | Snowflake | Relational | 155.58 | +1.68 | +28.13 |
| 6. | 6. | ↓ 5. | IBM Db2 | Relational, Multi-model | 125.44 | +2.46 | -6.79 |
| 7. | 7. | 7. | SQLite | Relational | 113.82 | +7.13 | -3.47 |
| 8. | 8. | 8. | Microsoft Access | Relational | 96.54 | +3.84 | -16.63 |
| 9. | 9. | ↑ 11. | Databricks | Multi-model | 90.03 | +2.19 | +13.13 |
| 10. | 10. | ↓ 9. | MariaDB | Relational, Multi-model | 89.50 | +3.92 | -7.73 |

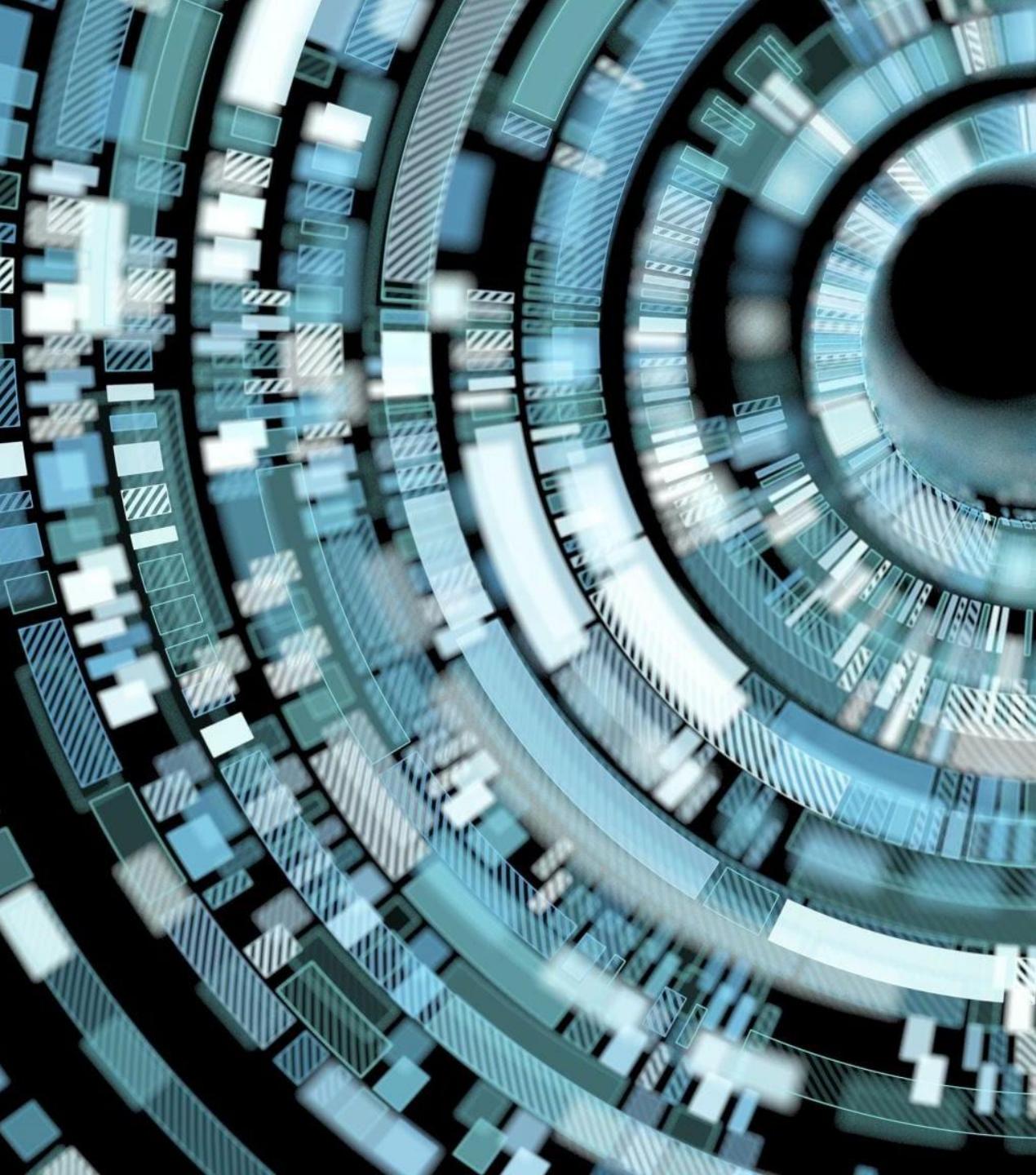
Ranking db-engines (Completo)

424 systems in ranking, February 2025

| Rank | | | DBMS | Database Model | Score | | |
|----------|----------|----------|------------------------------|---|----------|----------|----------|
| Feb 2025 | Jan 2025 | Feb 2024 | | | Feb 2025 | Jan 2025 | Feb 2024 |
| 1. | 1. | 1. | Oracle | Relational, Multi-model i | 1254.82 | -3.93 | +13.38 |
| 2. | 2. | 2. | MySQL | Relational, Multi-model i | 999.99 | +1.84 | -106.67 |
| 3. | 3. | 3. | Microsoft SQL Server | Relational, Multi-model i | 786.87 | -11.69 | -66.70 |
| 4. | 4. | 4. | PostgreSQL + | Relational, Multi-model i | 659.62 | -3.79 | +30.21 |
| 5. | 5. | 5. | MongoDB + | Document, Multi-model i | 396.63 | -5.87 | -23.73 |
| 6. | ↑ 7. | 6. | Redis | Key-value, Multi-model i | 157.91 | +4.55 | -2.80 |
| 7. | ↓ 6. | ↑ 9. | Snowflake | Relational | 155.58 | +1.68 | +28.13 |
| 8. | 8. | ↓ 7. | Elasticsearch | Multi-model i | 134.63 | -0.29 | -1.11 |
| 9. | 9. | ↓ 8. | IBM Db2 | Relational, Multi-model i | 125.44 | +2.46 | -6.79 |
| 10. | 10. | 10. | SQLite | Relational | 113.82 | +7.13 | -3.47 |

Presente y Futuro

- HTAP (Hybrid Transactional/Analytical Processing)
- Cloud Databases
- IA y Machine Learning
- DataOps
- Bases de datos distribuídas



SGBD

Microsoft SQL Server

- Es un producto de Microsoft
- Programado mayormente en C++, y algunas partes en C.
- Funciona en Windows y Linux

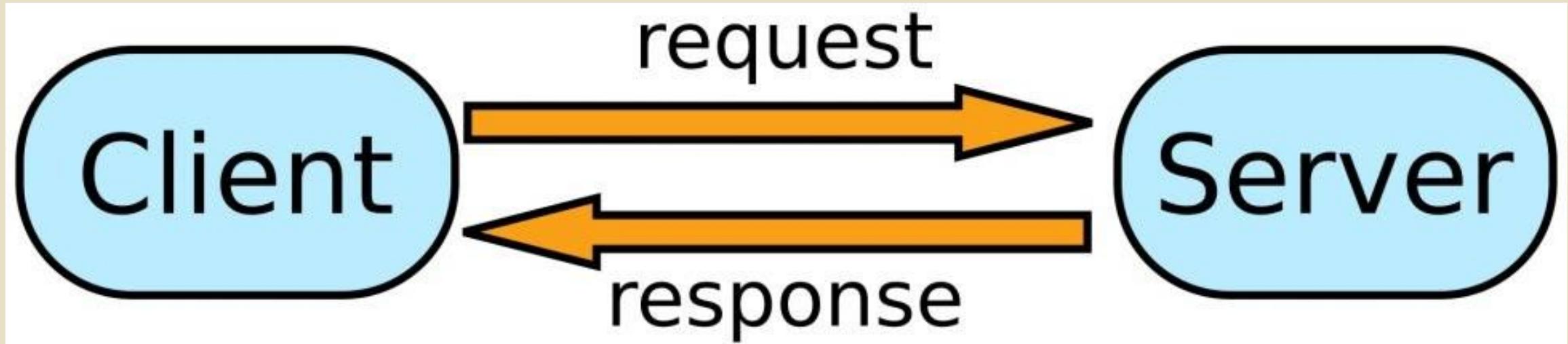


MySQL

- Es un producto de Oracle, pero es open source
- Programado en C y C++
- Funciona en múltiples plataformas



Cliente / Servidor





MS SQL Server

- Cuenta con dos opciones para poder trabajar sin problemas de licencia
 - Express
 - Developer

SQL Server 2022 Express



- Sistema Operativo
 - Windows Server 2016, Windows Server 2019, Windows Server 2022, Windows 10, Windows 11
 - Procesador: Intel - compatible processor with a minimum speed of 1 GHz or a faster processor
 - RAM: 512MB
 - Disco: 4.2 GB
 - Linux
 - Ubuntu 20.04 o 22.04, Red Hat Enterprise Linux 8.x o 9.x Server, o Suse Linux Enterprise Server v15
 - Procesador: 2GHz, 2 cores, x64-compatible
 - RAM: 2GB
 - Disco: 6GB
- Docker
 - Developer Edition

<https://www.microsoft.com/en-gb/download/details.aspx?id=104781>

MySQL 8.4.4 LTS

Community Server



- Sistema Operativo:
 - Oracle Linux / Red Hat / CentOS
 - Oracle Solaris
 - Ubuntu
 - SUSE
 - Debian
 - Microsoft Windows Server (2012 R2 a 2022)
 - Microsoft Windows (10 u 11)
 - Apple (macOS 11 o 12)
 - Varios Linux

<https://dev.mysql.com/downloads/mysql/>



SQL Server / Clientes

- Oficiales de Microsoft
 - SQL Server Management Studio
 - Azure Data Studio
 - SQL Server Data Tools (Visual Studio)
 - Visual Studio Code (con extensión MSSQL)
 - sqlcmd
- Otros
 - dbForge Studio for SQL Server
 - Navicat for SQL Server
 - EMS SQL Management Studio for SQL Server
 - DataGrip

<https://aka.ms/ssms>

MySQL / Clientes



Oficial de Oracle

- MySQL Workbench

Otros

- BeeKeeper Studio
- Dbeaver
- Azure Data Studio (con Complemento)
- Visual Studio Code (con Complemento)
- DataGrip by JetBrains
- HeidiSQL
- dbForge Studio for MySQL
- Navicat for MySQL
- SQuirreL SQL

<https://dev.mysql.com/downloads/workbench/>

Buena semana!!!

ATENCIÓN a la comunicación...