1. Introducción

El término bases de datos fue escuchado por primera vez en un simposio celebrado en California en 1963.

Desde el punto de vista informático, una base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Por su parte, un sistema de Gestión de Bases de datos es un tipo de software muy especifico dedicado a servir de interfaz entre la base de datos, el usuario

2. Orígenes

Los orígenes de las bases de datos se remontan a la Antigüedad donde ya existían bibliotecas y toda clase de registros. Además también se utilizaban para recoger información sobre las cosechas y censos. Sin embargo, su búsqueda era lenta y poco eficaz y no se contaba con la ayuda de máquinas que pudiesen reemplazar el trabajo manual.

En 1884 Herman Hollerith creó la máquina automática de tarjetas perforadas,

siendo nombrado así el primer ingeniero estadístico de la historia.

Ante esta situación, Hollerith comenzó a trabajar en el diseño de una maquina tabuladora o censadora, basada en tarjetas perforadas.

Posteriormente, en la década de los cincuenta se da origen a las cintas magnéticas, para automatizar la información y hacer respaldos.

Las aplicaciones informáticas de los años sesenta acostumbraban a darse totalmente por lotes (batch) y estaban pensadas para una tarea muy específica relacionada con muy pocas entidades tipo.

Cada aplicación (una o varias cadenas de programas) utilizaba ficheros de movimientos para actualizar (creando una copia nueva) y/o para consultar uno o dos ficheros maestros o, excepcionalmente, más de dos.

Más adelante fue surgiendo la necesidad de hacer las actualizaciones también on-line.

A medida que se integraban las aplicaciones, se tuvieron que interrelacionar sus ficheros y fue necesario eliminar la redundancia.

El nuevo conjunto de ficheros se debía diseñar de modo que estuviesen interrelacionados;

Estos conjuntos de ficheros interrelacionados, con estructuras complejas y compartidas por varios

procesos de forma simultánea (unos on-line y otros por lotes), recibieron al principio el nombre de Data Banks, y después, a inicios de los años setenta, el de Data Bases.

 la segunda mitad de los años setenta, fue saliendo al mercado software más sofisticado: los **Data Base Management Systems**, que aquí denominamos **Sistemas de Gestión de BD**(**SGBD**).

En otras palabras, una base de datos es un conjunto estructurado de datos que representa entidades y sus interrelaciones.

**Década de los 60.**

Posteriormente en la época de los sesenta, las computadoras bajaron los precios para que las compañías privadas las pudiesen adquirir; dando paso a que se popularizara el uso de los discos,

 partir de este soporte se podía consultar la información directamente, sin tener que saber la ubicación exacta de los datos.

En esta misma época se dio inicio a las primeras generaciones de bases de datos de red y las bases de datos jerárquicas, ya que era posible guardar estructuras de datos en listas y arboles.

se llevo a cabo el **desarrollo del IDS desarrollado por Charles Bachman** (**que formaba parte de la CODASYL**) supuso la creación de un nuevo tipo de sistema de bases de datos conocido como modelo en red que permitió la creación de un estándar en los sistemas de bases de datos gracias a la creación de nuevos lenguajes de sistemas de información.

**CODASYL** (**Conference on Data Systems Languages**) era un consorcio de industrias informáticas que tenían como **objetivo la regularización de un lenguaje de programación** estándar que pudiera ser utilizado en multitud de ordenadores.

El Data Base / Data Comincations

IBM denominaba así (DB/DC) al software de comunicaciones y de gestión de transacciones de datos. Las aplicaciones típicas eran las reservas/compras de billetes a las compañías aéreas y de ferrocarriles. Más tarde, el mismo, se comenzará a utilizar en el mundo del mercado financiero y bursátil. У

**4. Década de los 70 – Sistemas Centralizados.**

 Estaban orientados a facilitar la utilización de grandes conjuntos de datos en los que las interrelaciones eran complejas. Estos sistemas trabajaban exclusivamente por lotes (batch).

Al aparecer los terminales de teclado, conectados al ordenador central (Mainframes) mediante una línea telefónica, se empiezan a construir grandes aplicaciones on-line transaccionales (OLTP).

Los SGBD estaban íntimamente ligados al software de comunicaciones y de gestión de transacciones.

Los SGBD estaban íntimamente ligados al software de comunicaciones y de gestión de transacciones.

Aunque para escribir los programas de aplicación se utilizaban lenguajes de alto nivel como Cobol o PL/I, se disponía también de instrucciones y de subrutinas especializadas para tratar las BD que

requerían que el programador conociese muchos detalles del diseño físico, y que hacían que la programación fuese muy compleja.

Por lo que respecta a la **década de los setenta, Edgar Frank Codd,** científico informático ingles conocido por sus aportaciones a la teoría de bases de datos relacionales, **definió el modelo relacional** a la par que publicó una serie de reglas para los sistemas de datos relacionales a través de su artículo “**Un modelo relacional de datos para grandes bancos de datos compartidos**”.

Este hecho dio paso al nacimiento de la segunda generación de los Sistemas Gestores de Bases de Datos.

Como consecuencia de esto, durante la década de 1970, Lawrence J. Ellison, más conocido como **Larry Ellison**, a partir del trabajo de Edgar F. Codd sobre los sistemas de bases de datos relacionales, desarrolló el Relational Software System, o lo que es lo mismo, lo que actualmente se conoce como **Oracle Corporation**, desarrollando así un sistema de gestión de bases de datos relacional con el mismo nombre que dicha compañía.

Ingres se desarrolló en la UBC en los años de 1974 a 1977.

ingres utilizaba un lenguaje de consulta, llamado QUEL, dando pie a la creación de sistemas como

Ingres Corporación, MS SQL Server, Sybase, PACE Wang, y Britton Lee-. Por su parte, el Sistema R utilizó el lenguaje de consulta Secuela, el cual ha contribuido al desarrollo de SQL / DS, DB2, Allbase, Oracle y SQL Non-Stop.

 en la época de los ochenta también se desarrollará el **SQL** (**Structured Query Language**)

 permite efectuar consultas y hacer cambios sobre la base de datos de forma sencilla;

ORACLE es considerado como uno de los sistemas de bases de datos más

completos que existen en el mundo

actualmente sufre la competencia del SQL Server de la compañía Microsoft y de la oferta de otros Sistemas Administradores de Bases de Datos Relacionales con licencia libre como es el caso de PostgreSQL, MySQL o Firebird que aparecerían posteriormente en la década de 19

**Década de los 80 – SGBD Relacionales.**

Por su parte, a principios de los años ochenta comenzó el auge de la comercialización de los sistemas relacionales, y SQL comenzó a ser el estándar de la industria, ya que las bases de datos relacionales con su sistema de tablas (compuesta por filas y columnas) pudieron competir con las bases jerárquicas y de red

Los SGBD de los años sesenta y setenta (IMS de IBM, IDS de Bull, DMS de Univac, etc.) eran sistemas totalmente centralizados, como corresponde a los sistemas operativos de aquellos años, y al hardware para el que estaban hechos:

Los Ordenadores personales

Durante los años ochenta aparecen y se extienden muy rápidamente los ordenadores personales. También surge software para equipos monousuario (por ejemplo dBase, Access), con los cuales es muy fácil crear y utilizar conjuntos de datos, que se denominan personal data bases. Note que el hecho de dominar SGBD a estos primeros sistemas para PC es un poco forzado, ya que no aceptaban estructuras complejas ni interrelaciones, ni podían ser utilizados en una red que sirviese simultáneamente a muchos usuarios de diferentes tipos. Sin embargo, algunos, con el tiempo, se han ido convirtiendo en auténticos SGBD.

En la década de los años 80’, se desarrolló el SQL (Structured Query Language), un lenguaje de consultas que permite consultar, valga la redundancia, con el fin de recuperar información de una base de datos y a su vez, hacer cambios sobre esa misma base, de forma sencilla. Permitía analizar gran cantidad de información y especificar varios tipos de operaciones con la misma información

SQL comenzó a ser el modelo estándar de las industrias, con su base de datos bajo un sistema de tablas

**Década de los 90: Distribución, C/S y 4GL**

Al acabar la década de los ochenta, los SGBD relacionales ya se utilizaban prácticamente en todas las empresas

base de datos distribuida.

La necesidad de tener una visión global de la empresa y de interrelacionar diferentes aplicaciones que utilizan BD diferentes, junto con la facilidad quedan las redes para la intercomunicación entre ordenadores, 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 se conoce como base de datos distribuida.

Esta distribución ideal se consigue cuando las diferentes BD son soportadas por una misma marca de SGBD, es decir, cuando hay homogeneidad.

 marcas diferentes pueden darse servicio unos a otros y colaborar para dar servicio a un programa de aplicación.

Las razones básicas por las que interesa esta distribución son las siguientes:

**Disponibilidad**

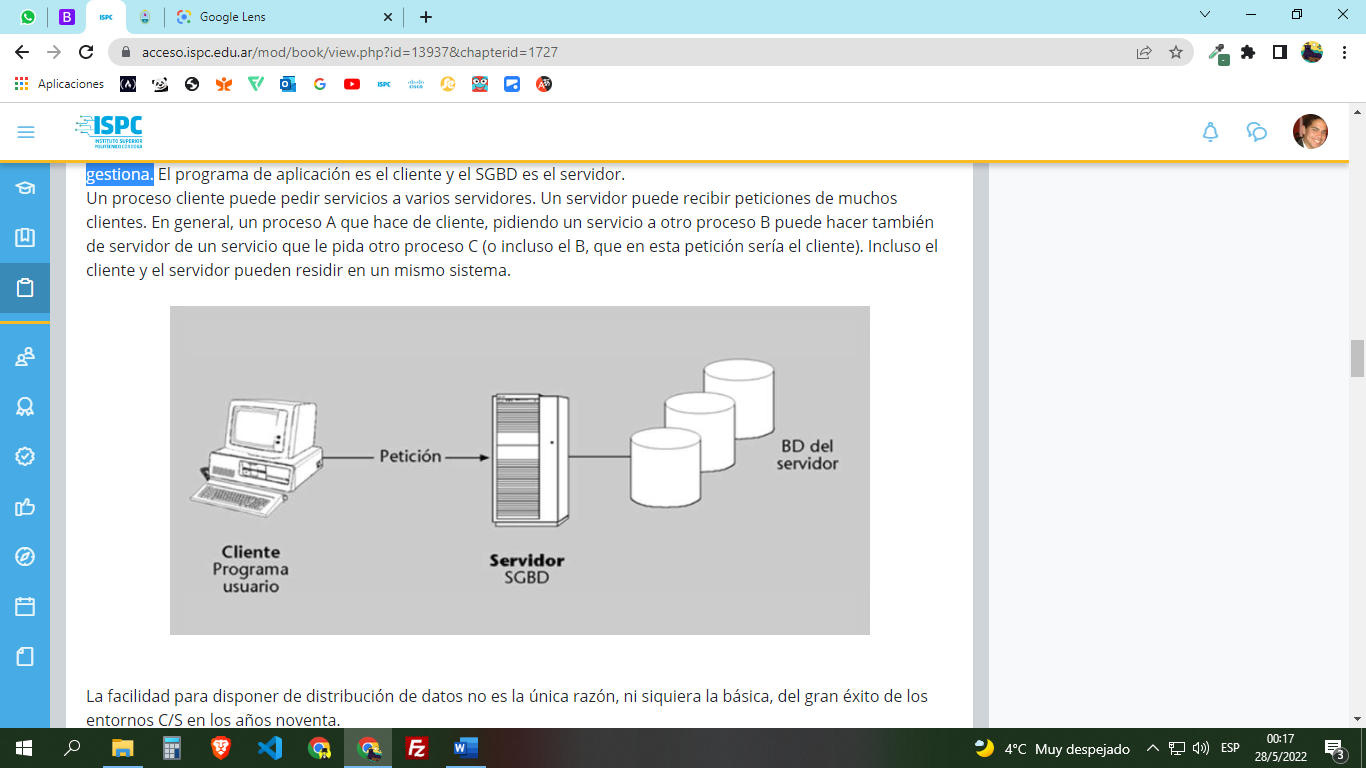
                           La disponibilidad de un sistema con una BD distribuida puede ser más alta, porque si queda fuera de servicio uno de los sistemas, los demás seguirán funcionando. Si los datos residentes en el sistema no disponible están replicados en otro sistema, continuarán estando disponibles. En caso contrario sólo estarán disponibles los datos de los demás sistemas.

**Coste**

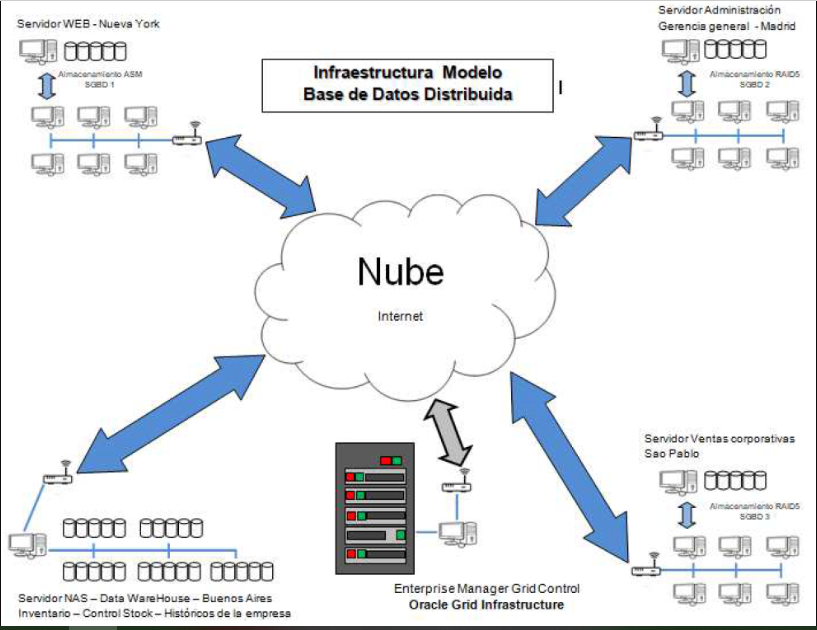
                           Una BD distribuida puede reducir el coste. En el caso de un sistema centralizado, todos los equipos usuarios, que pueden estar distribuidos por distintas y lejanas áreas geográficas, están conectados al sistema central por medio de líneas de comunicación.

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 peticionando un servicio y el otro el del servidor o proveedor del servicio.



El éxito de las BD, incluso en sistemas personales, ha llevado a la aparición de los Fourth Generation Languages (4GL), lenguajes muy fáciles y potentes, especializados en el desarrollo de aplicaciones fundamentadas en BD.



**Tendencias actuales**

Los tipos de datos que se pueden definir en los SGBD relacionales de los años ochenta y noventa son muy limitados.

Necesitan tipos complejos que el desarrollador pueda definir a medida de la aplicación.

En definitiva, se necesitan tipos abstractos de datos: **TAD**.

**Esto nos lleva a la orientación a objetos** (OO). El éxito de la OO al final de los años ochenta, en **el**

**desarrollo de software básico,** en las aplicaciones de ingeniería industrial y en la construcción de

interfaces gráficas con los usuarios, ha hecho que durante la década de los noventa se extendiese en prácticamente todos los campos de la informática. En los SI se inicia también la adopción, tímida de momento, de la OO.

La utilización de lenguajes como C++ o Java requiere que los SGBD relacionales se adapten a ellos con interfaces adecuadas.

La rápida adopción de la web a los SI hace que los SGBD incorporen recursos para ser servidores de páginas web, como por ejemplo la inclusión de SQL en guiones HTML, SQL incorporado en Java, etc.

Durante estos últimos años se ha empezado a extender un tipo de aplicación de las BD denominado Data Warehouse, o almacén de datos, que también produce algunos cambios en los SGBD relacionales del mercado.

Los datos de este gran almacén, el Data Warehouse, se obtienen por una replicación más o menos elaborada de las que hay en las BD que se utilizan en el trabajo cotidiano de la empresa.

Actualmente, los SGBD se adaptan a este tipo de aplicación, incorporando, por ejemplo, herramientas como las siguientes:

**La creación y el mantenimiento de réplicas, con una cierta elaboración de los datos.**

**La consolidación de datos de orígenes diferentes.**

**La creación de estructuras físicas que soporten eficientemente el análisis multidimensional.**

**Gestores de Bases de Datos**

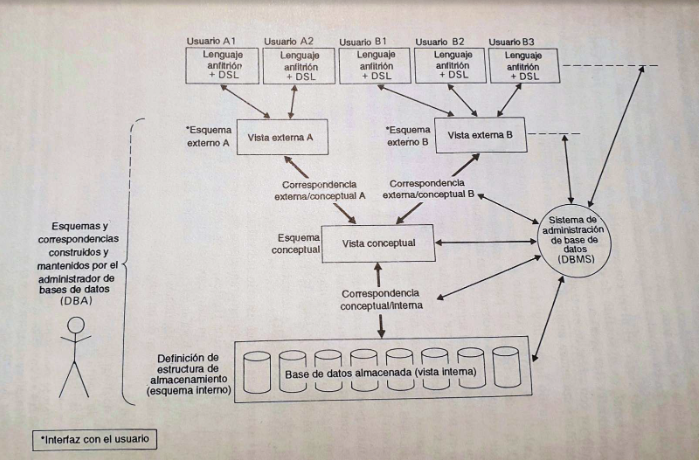
Hoy día la mayoría de las bases de datos se presentan en formato digital, gracias a los avances

tecnológicos en la informática y la electrónica.

Los gestores de bases de datos, Database Management System o DBMS (SGBD) **son programas que permiten almacenar y luego acceder a los datos de forma estructurada y rápida.**

*Una base de datos es un sistema compuesto por un conjunto de datos, los cuales están almacenados en discos, a los que se accede directamente y un conjunto de programas que regulen o manejen ese conjunto de datos.*

*Mientras que un sistema de Gestión de Bases de Datos es un software que sirve de interfaz entre la base de datos, el usuario y las aplicaciones que se utilizan.*



**Los mejores gestores de base de datos**

El principal lenguaje de base de datos y el más utilizado desde que se conoce la programación de

gestión, es el Structured Query Language (SQL). Este, de consulta estructurada, facilita el acceso a la gestión de las bases de datos relaciones,

**Oracle**

             Es de los más confiables sistemas de gestión de base de datos relacional, además del más usado. Es propiedad de Oracle Corporation y fue desarrollado en 1977.

Se accede directamente a los objetos, a través del lenguaje de consulta SQL, es muy utilizado en las empresas, con un componente de red que permite la comunicación a través de las redes.

**SQL Server**

             En competencia directa a Oracle, está SQL Server de Microsoft. Los dos ocupan gran parte del mercado en el sector de base de datos. Son muy parecidos en algunas de sus características y funciones, aunque tienen sus marcadas diferencias.

SQL Server se ejecuta en Transact-SQL, esto es un grupo de programas que pueden añadir características al programa, como tratamiento de errores y excepciones, extracción de datos de la web en forma directa, procesamiento de datos, uso de distintos lenguajes de programación

**Gestores de base de datos de acceso libre**

            Dos de los principales y más utilizados gestores de pago, que son de acceso libre (Open Source) son los siguientes:

**MySQL**

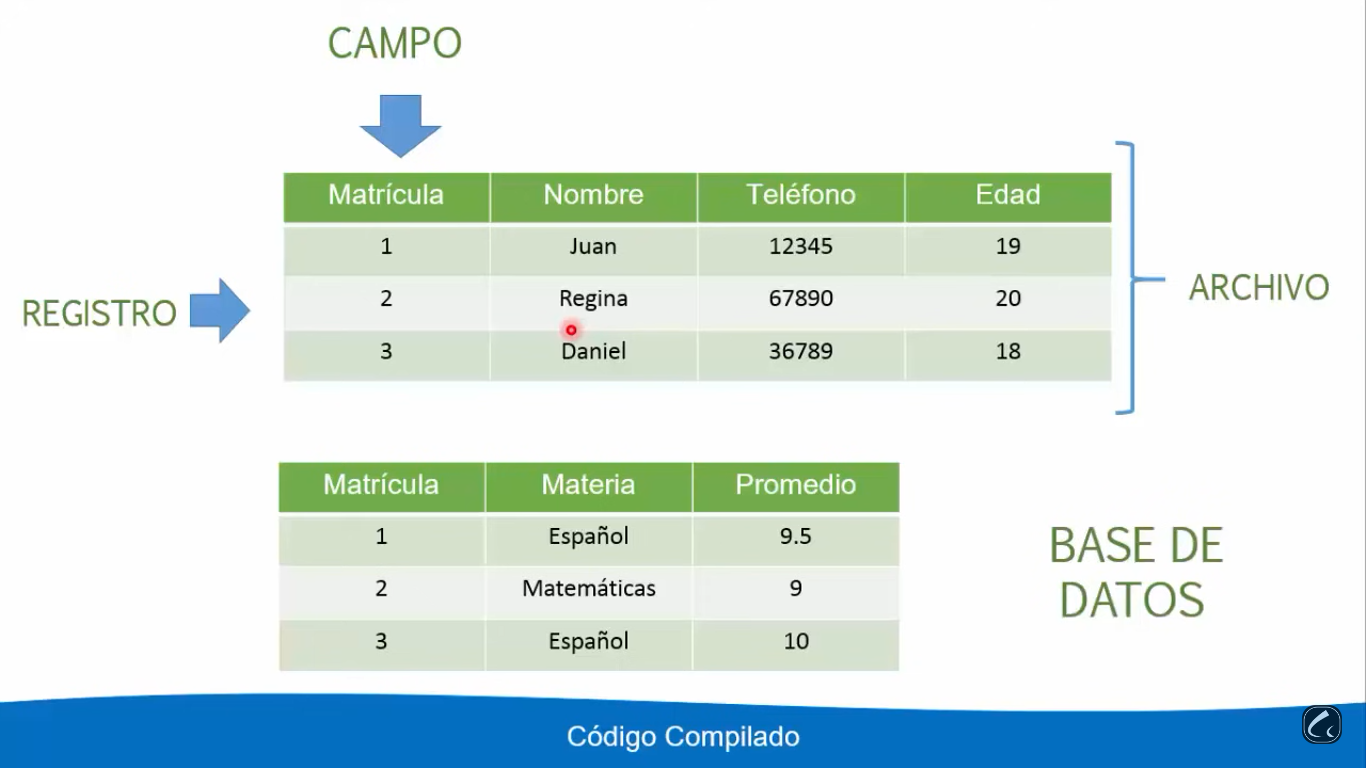
           Este es de simple instalación y actúa de lado del cliente o servidor, es de código abierto y tiene licencia comercial disponible. Pertenece a Oracle Corporation y gestiona las bases de datos relacionales Se usa principalmente para el desarrollo Web.

**Fire Bird**

            De gran potencia y muy sencillo a la vez, este sistema de gestión de base de datos relacional SQL, es uno de los mejores gestores Open Source (Código abierto) o libres. Es compatible con Windows y Linux.

### Introducción a las Bases de Datos (video 1)

En este capítulo nos introducimos a las base de datos, comenzamos con conceptos básicos pero claves, como lo es el concepto de dato, para poco a poco ir profundizando.



Datos y Archivos.

La necesidad de gestionar datosEl propio nombre Informática hace referencia al hecho de ser una ciencia que trabaja con información. Desde los albores de la creación de ordenadores, la información se ha considerado como uno de los pilares de las computadoras digitales. Por ello las bases de datos son una de las aplicaciones más antiguas de la informática. En informática se conoce como dato a cualquier elemento informativo que tenga relevancia para el sistema.

La escritura fue la herramienta que permitió al ser humano poder gestionar bases cada vez más grandes de datos

Para poder almacenar datos y cada vez más datos, el ser humano ideó nuevas herramientas archivos, cajones, carpetas y fichas en las que se almacenaban los datos.

Por esa razón, la informática adaptó sus herramientas para que los elementos que el usuario maneja en el ordenador se parezcan a los que utilizaba manualmente. Así en informática se sigue hablado de ficheros, formularios, carpetas, directorios,....

## Sistemas de Información.

definición: **“Conjunto de cosas que ordenadamente relacionadas entre sí contribuyen a un determinado objeto”** .

El sistema completo que forma la empresa, por otra parte, se suele dividir en los siguientes subsistemas:

**Subsistema productivo**. Representa la parte de la empresa encargada de gestionar la producción de la misma.

**Subsistema financiero**. Encargado de la gestión de los bienes económicos de la empresa

**Subsistema directivo.** Encargado de la gestión organizativa de la empresa

Los sistemas que aglutinan los elementos que intervienen para gestionar la información que manejan los subsistemas empresariales es lo que se conoce como Sistemas de Información. Se suele utilizar las siglas **SI o IS** (de Information Server) para referirse a ello).

Un sistema de información genérico está formado por los siguientes elementos:

**Recursos físicos.** Carpetas, documentos, equipamiento, discos,...

**Recursos humanos**. Personal que maneja la información

**Protocolo**. Normas que debe cumplir la información para que sea manejada (formato de la información, modelo para los documentos,...)



#### Componentes de un sistema de información electrónico

**sistema de información electrónico** los componentes son:

**Datos**. Se trata de la información relevante que almacena y gestiona el sistema de información. Ejemplos de datos son: Sánchez, 12764569F, Calle Mayo 5, Azul…

**Hardware**. Equipamiento físico que se utiliza para gestionar los datos. cada uno de los dispositivos electrónicos que permiten el funcionamiento del sistema de información.

**Software**. Aplicaciones informáticas que se encargan de la gestión de la base de datos y de las herramientas que facilitan su uso.

**Recursos humanos**. Personal que maneja el sistema de información.

# Archivos.

Los datos deben de ser almacenados en componentes de almacenamiento permanente, lo que se conoce como **memoria secundaria**. En esas memorias, los datos se estructuran en archivos (también llamados acros)

**. Un fichero es una secuencia de números binarios que organiza información relacionada a un mismo aspecto**.

En general sobre los archivos se pueden realizar las siguientes operaciones:

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

Cuando los ficheros almacenan datos, se dice que constan de registros. Cada registro contiene datos relativos a un mismo elemento u objeto. Por ejemplo en un fichero de personas, cada registro contiene datos de una persona. Si el archivo contiene datos de 1000 personas, constará de 1000 registros. A continuación se explican los tipos más habituales de ficheros.

# Ficheros secuenciales.

En estos ficheros, los datos se organizan secuencialmente en el orden en el que fueron grabados. Para leer los últimos datos hay que leer los anteriores. Es decir leer el registro número nueve, implica leer previamente los ocho anteriores.

Ventajas

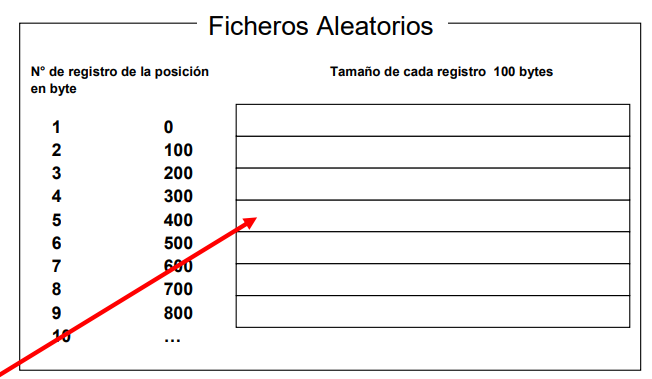
* Rápidos para obtener registros contiguos de una base de datos
* No hay huecos en el archivo al grabarse los datos seguidos, datos más compactos.

Desventajas

* Consultas muy lentas al tener que leer todos los datos anteriores al dato que queremos leer
* Algoritmos de lectura y escritura más complejos
* No se pueden eliminar registros del fichero(se pueden señalar)
* El borrado provoca archivos que no son compactos
* La ordenación de los datos requiere volver a crearle de nuevo

# Ficheros de acceso directo o aleatorio.

Se puede leer una posición concreta del fichero, con saber la posición (normalmente en bytes) del dato a leer. Cuando se almacenan registros, posicionarnos en el quinto registro se haría de golpe, lo único necesitamos saber el tamaño del registro, que en este tipo de ficheros debe de ser el mismo. Suponiendo que cada registro ocupa 100 bytes, el quinto registro comienza en la posición 400. Lo que se hace es colocar el llamado puntero de archivo en esa posición y después leer.



Acceso directo al 5 fichero

**Ventajas**

* Acceso rápido al no tener que leer los datos anteriores
* La modificación de datos es más sencilla
* Permiten acceso secuencial
* Permiten leer y escribir a la vez
* Aptos para organizaciones relativas directas, en las que la clave del registro se relaciona con su posición en el archivo

**Desventajas**

* Salvo en archivos relativos directos, no es apto por sí mismo para usar en bases de datos, ya que los datos se organizan en base a una clave
* No se pueden borrar datos (sí marcar para borrado, pero generarán huecos)
* Las consultas sobre multitud de registros son más lentas que en el caso anterior.

# Ficheros secuenciales encadenados.

Son ficheros secuenciales gestionados mediante punteros, datos especiales que contienen la dirección de cada registro del fichero. Cada registro posee ese puntero que indica la dirección del siguiente registro y que se puede modificar en cualquier momento. El puntero permite recorrer los datos en un orden concreto. Cuando aparece un nuevo registro, se añade al final del archivo, pero los punteros se reordenan para que se mantenga el orden.



**Ventajas**

* El fichero mantiene el orden en el que se añadieron los registros y un segundo orden en base a una clave
* La ordenación no requiere reorganizar todo el fichero, sino sólo modificar los punteros
* Las mismas ventajas que el acceso secuencial
* En esta caso sí se borran los registros y al reorganizar, se perderán definitivamente

**Desventajas**

* No se borran los registros, sino que se marcan para ser ignorados. Por lo que se malgasta espacio
* Añadir registros o modificar las claves son operaciones que requieren recalcular los punteros

# Ficheros secuenciales indexados.

Se utilizan dos ficheros para los datos, uno posee los registros almacenados de forma secuencial, pero que permite su acceso aleatorio. El otro posee una tabla con punteros a la posición ordenada de los registros. Ese segundo fichero es el índice, una tabla con la ordenación deseada para los registros y la posición que ocupan en el archivo.

Existe un archivo llamado de desbordamiento u overflow en el que se colocan los nuevos registros que se van añadiendo (para no tener que ordenar el archivo principal cada vez que se añade un nuevo registro) este archivo está desordenado. Se utiliza sólo si se busca un registro y no se encuentra en el archivo principal. En ese caso se recorre todo el archivo de overflow hasta encontrarlo.

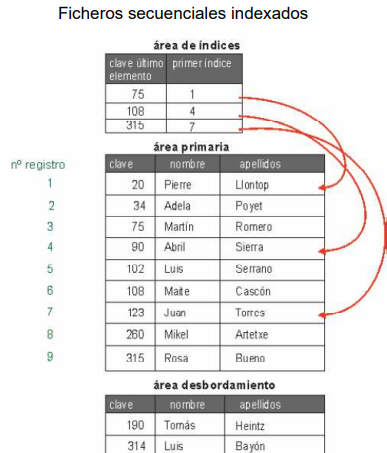
**Ventajas**

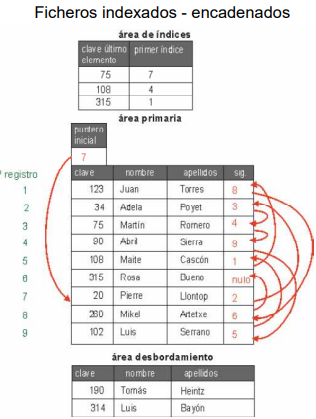
* El archivo está siempre ordenado en base a una clave
* La búsqueda de datos es rapidísima
* Permite la lectura secuencial (que además será en el orden de la clave)
* El borrado de registros es posible (aunque más problemático que en el caso anterior)

**Desventajas**

* Para un uso óptimo hay que reorganizar el archivo principal y esta operación es muy costosa ya que hay que reescribir de nuevo y de forma ordenada todo el archivo.
* La adición de registros requiere más tiempo que en los casos anteriores al tener que reordenar los índices

# Ficheros indexado – encadenados.

Utiliza punteros e índices, es una variante encadenada del caso anterior. Hay un fichero de índices equivalente al comentado en el caso anterior y otro fichero de tipo encadenado con punteros a los siguientes registros.

Cuando se añaden registros se añaden en un tercer registro llamado de desbordamiento u overflow. En ese archivo los datos se almacenan secuencialmente, se accede a ellos si se busca un dato y no se encuentra en la tabla de índices. 

**Ventajas**

* Posee las mismas ventajas que los archivos secuenciales

indexados, además de una mayor rapidez al reorganizar

el fichero (sólo se modifican los punteros)

**Desventajas**

* Requieren compactar los datos a menudo para

reorganizar índices y quitar el fichero de desbordamiento.

. Operaciones relacionadas con uso de ficheros en bases de datos.

Borrado y recuperación de registros

Algunos de los tipos de ficheros vistos anteriormente no admiten el borrado real de datos, sino que sólo permiten añadir un dato que indica si el registro está borrado o no.

En otros casos los datos antes de ser eliminados del todo pasan a un fichero especial (conocido como papelera) en el que se mantienen durante cierto tiempo para su posible recuperación.

Fragmentación y compactación de datos

La fragmentación en un archivo hace referencia a la posibilidad de que éste tenga huecos interiores debido a borrado de datos u a otras causas. Causa los siguientes **problemas**:

* La fragmentación en un archivo hace referencia a la posibilidad de que éste tenga huecos interiores debido a borrado de datos u a otras causas. Causa los siguientes problemas:
* Lentitud en las operaciones de lectura y escritura del fichero

Por ello se requiere **compactar** los datos. Esta técnica permite eliminar los huecos interiores a un archivo. Las formas de realizarla son:

* Reescribir el archivo para eliminar los huecos. Es la mejor, pero lógicamente es la más lenta al requerir releer y reorganizar todo el contenido del fichero
* provechar huecos. De forma que los nuevos registros se inserten en esos huecos. Esta técnica suele requerir un paso previo para reorganizar esos huecos.

Compresión de datos

En muchos casos para ahorrar espacio de almacenamiento, se utilizan técnicas de compresión de datos. La ventaja es que los datos ocupan menos espacio y la desventaja es que al manipular los datos hay que descomprimirlos lo que hace que la manipulación de los datos sea lenta.

Cifrado de datos

Otra de las opciones habituales sobre ficheros de datos es utilizar técnicas de cifrado para proteger los ficheros en caso de que alguien no autorizado se haga con el fichero. Para leer un fichero de datos, haría falta descifrar el fichero. Para descifrar necesitamos una clave o bien aplicar métodos de descifrado; lógicamente cuanto mejor sea la técnica de cifrado, más difícil será descifrar los datos mediante la fuerza bruta.

### 4. Sistemas Gestores de Bases de Datos

Tipos de Sistemas de Información

Sistemas de Información orientados a procesos.

Antes de aparecer los SGBD (década de los setenta), la información se trataba y se gestionaba utilizando los típicos sistemas de gestión de archivos que iban soportados sobre un sistema operativo. Éstos consistían en un conjunto de programas que definían y trabajaban sus propios datos.

En estos sistemas de información se crean diversas aplicaciones (software) para gestionar diferentes aspectos del sistema. Cada aplicación realiza unas determinadas operaciones. Los datos de dichas aplicaciones se almacenan en archivos digitales dentro de las unidades de almacenamiento del ordenador

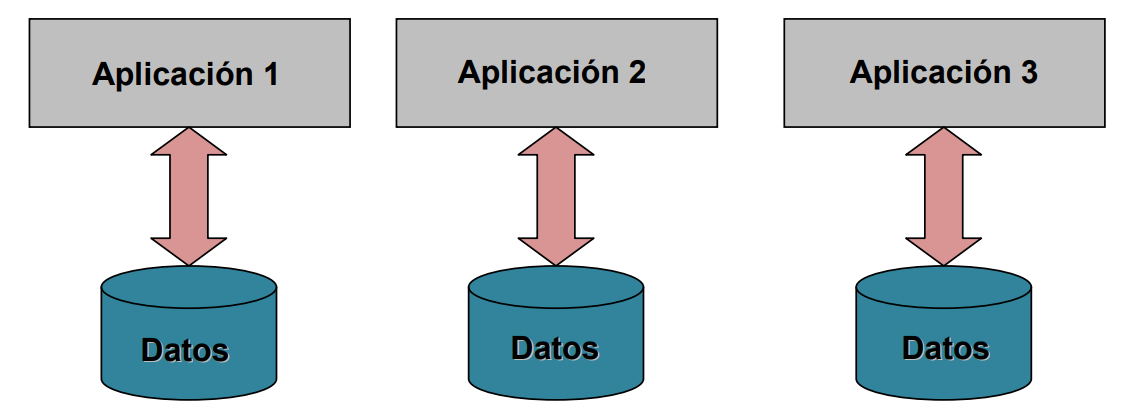
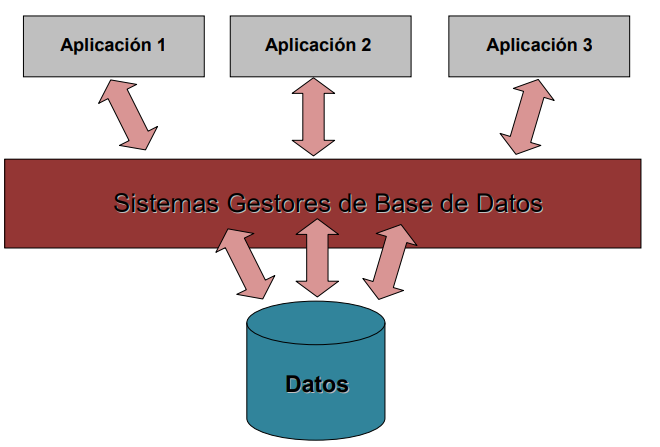
La ventaja de este sistema (la única ventaja), es que los procesos son independientes por lo que la modificación de uno no afectaba al resto.

Esto supone un gran inconveniente a la hora de tratar grandes volúmenes de información. Surge así la **idea de separar los datos contenidos en los archivos de los programas que los manipulan**, es decir, que se pueda modificar la estructura de los datos de los archivos sin que por ello se tengan que modificar los programas con los que trabajan

Inconvenientes de un sistema de gestión de archivos

* **Redundancia e inconsistencia de los datos**: se produce porque los archivos son creados por distintos programas y van cambiando a lo largo del tiempo, es decir, pueden tener distintos formatos y los datos pueden estar duplicados
* **Dependencia de los datos física-lógica:** Cualquier cambio en esa estructura implica al programador identificar, modificar y probar todos los programas que manipulan esos archivos.
* **Dificultad para tener acceso a los datos**: o. Lo que se trata de probar es que los entornos convencionales de procesamiento de archivos no permiten recuperar los datos necesarios de una forma conveniente y eficiente.
* **Separación y aislamiento de los datos:** es decir, al estar repartidos en varios archivos, y tener diferentes formatos, es difícil escribir. Antes se deberían sincronizar todos los archivos para que los datos coincidiesen.
* **Dificultad para el acceso concurrente**: pues en un sistema de gestión de archivos es complicado que los usuarios actualicen los datos simultáneamente. Las actualizaciones concurrentes pueden dar por resultado datos inconsistentes
* **Dependencia de la estructura del archivo con el lenguaje de programación:** pues la estructura se define dentro de los programas. Esto implica que los formatos de los archivos sean incompatibles. La incompatibilidad entre archivos generados por distintos lenguajes hace que los datos sean difíciles de procesar.
* **Problemas en la seguridad de los datos:** Resulta difícil implantar restricciones de seguridad pues las aplicaciones se van añadiendo al sistema según se van necesitando.
* **Problemas de integridad de datos (datos inconsistentes):** es decir, los valores almacenados en los archivos deben cumplir con restricciones de consistencia. Ya que un proceso cambia sus datos y no el resto. Por lo que el mismo dato puede tener valores distintos según qué aplicación acceda a él.

A estos sistemas se les llama sistemas de gestión de ficheros. Se consideran también así a los sistemas que utilizan programas ofimáticos (como Word o Excel por ejemplo) para gestionar sus datos (muchas pequeñas empresas utilizan esta forma de administrar sus datos). De hecho estos sistemas producen los mismos (si no más) problemas.



Sistemas de Información orientados a los datos. Bases de Datos.

Todos estos inconvenientes hacen posible el fomento y desarrollo de SGBD. El objetivo primordial de un gestor es proporcionar eficiencia y seguridad a la hora de extraer o almacenar información en las BD.

Los sistemas gestores de BBDD están diseñados para gestionar grandes bloques de información, que implica tanto la definición de estructuras para el almacenamiento como de mecanismos para la gestión de la información. En este tipo de sistemas los datos se centralizan en una base de datos común a todas las aplicaciones.

Una BD es un gran almacén de datos que se define una sola vez; los datos pueden ser accedidos de forma simultánea por varios usuarios;

**Ventajas**

* Independencia de los datos y los programas y procesos. Esto permite modificar los datos sin modificar el código de las aplicaciones.
* Menor redundancia. No hace falta tanta repetición de datos. Sólo se indica la forma en la que se relacionan los datos.
* Integridad de los datos. Mayor dificultad de perder los datos o de realizar incoherencias con ellos.
* Mayor seguridad en los datos. Al permitir limitar el acceso a los usuarios. Cada tipo de usuario podrá acceder a unas cosas.
* Datos más documentados. Gracias a los metadatos que permiten describir la información de la base de datos.
* Acceso a los datos más eficiente. La organización de los datos produce un resultado más óptimo en rendimiento.
* Menor espacio de almacenamiento. Gracias a una mejor estructuración de los datos.
* Acceso simultáneo a los datos. Es más fácil controlar el acceso de usuarios de forma concurrente.

**Desventajas**

* Instalación costosa. El control y administración de bases de datos requiere de un software y hardware poderoso.
* Requiere personal cualificado. Debido a la dificultad de manejo de este tipo de sistemas.
* Implantación larga y difícil. Debido a los puntos anteriores. La adaptación del personal es mucho más complicada y lleva bastante tiempo.
* Ausencia de estándares reales. Lo cual significa una excesiva dependencia hacia los sistemas comerciales del mercado. Aunque, hoy en día, una buena parte de esta tecnología está aceptada como estándar de hecho.

. Arquitectura de los Sistemas Gestores de Bases de Datos

SGBD :es el software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos.

El éxito del SGBD reside en mantener la seguridad e integridad de los datos.

herramientas que proporciona:

* Herramientas para la creación y especificación de los datos: así como la estructura de la base de datos. Especificación de la estructura, el tipo de los datos, las restricciones y relaciones entre ellos mediante lenguajes de definición de datos. Toda esta información se almacena en el diccionario de datos, el SGBD proporcionará mecanismos para la gestión del diccionario de datos.
* Herramientas para administrar y crear la estructura física: requerida en las unidades de almacenamiento.
* Herramientas para la manipulación de los datos: de las bases de datos, para añadir, modificar, suprimir o consultar datos.
* Herramientas de recuperación: en caso de desastre.
* Herramientas para la creación de copias de seguridad: para restablecer la información en caso de fallos en el sistema.
* Herramientas para la gestión de la comunicación: de la base de datos.
* Herramientas para la creación de aplicaciones: que utilicen esquemas externos de los datos.
* Herramientas de instalación: de la base de datos.
* Herramientas para la exportación e importación: de datos.

# Niveles de abstracción de una base de datos.

Introducción En cualquier sistema de información se considera que se pueden observar los datos desde dos puntos de vista:

**Nivel externo**: Esta es la visión de los datos que poseen los usuarios del Sistema de Información.

**Nivel físico**: Esta es la forma en la que realmente están almacenados los datos

Realmente la base de datos es la misma, pero se la puede observar desde estos dos puntos de vista. Al igual que una casa se la pueda observar pensando en los materiales concretos con los que se construye o bien pensando en ella con el plano en papel.

En el caso de los Sistemas de Base de datos, se añade un tercer nivel, un tercer punto de vista, es el **nivel conceptual**. Ese nivel se sitúa entre el físico y el externo.

En cada nivel se manejan esquemas de la base de datos, al igual que al construir una casa, los distintos profesionales manejan distintos tipos de planos

En 1975, el comité ANSI-SPARC propuso una arquitectura de tres niveles para los SGBD cuyo objetivo principal era el de separar los programas de aplicación de la BD física.

**Nivel interno o físico**: el más cercano al almacenamiento físico, es decir, tal y como están almacenados en el ordenador Esta visión sólo la requiere el administrador/a. El administrador la necesita para poder gestionar más eficientemente la base de datos

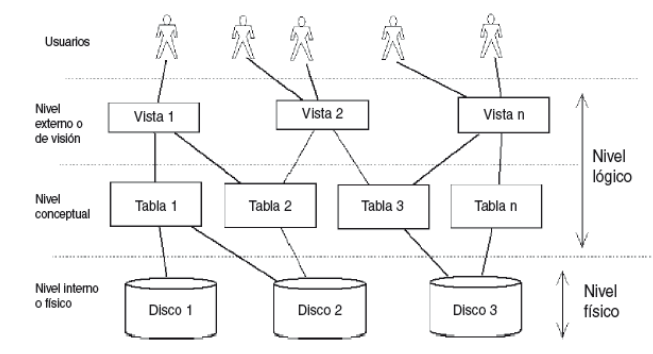
**Nivel externo o de visión**: es el más cercano a los usuarios

**En realidad son varios. Se trata de la visión de los datos que poseen los usuarios y usuarias finales. Esa visión es la que obtienen a través de las aplicaciones**

el usuario no conoce las relaciones entre los datos, como tampoco conoce dónde realmente se están almacenando los datos

Los esquemas externos los realizan las programadoras/es según las indicaciones formales de los y las analistas. Realmente cada aplicación produce un esquema externo diferente (aunque algunos pueden coincidir) o vista de usuario. El conjunto de todas las vistas de usuario es lo que se denomina **esquema externo global**.

**Nivel conceptual:** describe la estructura de toda la BD para un grupo de usuarios mediante un esquema conceptual. Este esquema describe las entidades, atributos, relaciones, operaciones de los usuarios y restricciones, ocultando los detalles de las estructuras físicas de almacenamiento. Representa la información contenida en la BD. Este esquema es el paso que permite modelar un problema real a su forma correspondiente en el ordenador

****

. Componentes de los Sistemas Gestores de Bases de Datos

**Lenguajes de los SGBD**

Todos los SGBD ofrecen lenguajes e interfaces apropiadas para cada tipo de usuario: administradores, diseñadores, programadores de aplicaciones y usuarios finales.

Los lenguajes del SGBD se clasifican en:

Lenguaje de definición de datos (LDD o DDL): se utiliza para especificar el esquema de la BD, las vistas de los usuarios y las estructuras de almacenamiento. Es el que define el esquema conceptual y el esquema interno. Lo utilizan los diseñadores y los administradores de la BD.permite definir las tres estructuras de la base de datos

* Estructura interna
* Estructura conceptual
* Estructura externa

Realmente esta función trabajo con metadatos. Los metadatos es la información de la base de datos que realmente sirve para describir a los datos

**La función de definición sirve pues para crear, eliminar o modificar metadatos. Para ello permite usar un lenguaje de descripción de datos o DDL.** Mediante ese lenguaje:

• Se definen las estructuras de datos

• Se definen las relaciones entre los datos

• Se definen las reglas que han de cumplir los datos

Lenguaje de manipulación de datos (LMD o DML) se utilizan para leer y actualizar los datos de la BD. **Es el utilizado por los usuarios para realizar consultas, inserciones, eliminaciones y modificaciones.**

Mediante ese lenguaje se puede:

• Añadir datos

• Eliminar datos

• Modificar datos

• Buscar datos

Lenguaje de control de datos (LCD o DCL): Mediante esta función **los administradores poseen mecanismos para proteger los datos**; de modo que se permite a cada usuario ver ciertos datos y otros no

**Es decir simplemente permite controlar la seguridad de la base de datos**

**No procedurales** son los lenguajes declarativos. En muchos SGBD se pueden introducir interactivamente instrucciones del LMD desde un terminal, también pueden ir embebidas en un lenguaje de programación de alto nivel. Estos lenguajes permiten especificar los datos a obtener en una consulta, o los datos a modificar, mediante sentencias sencillas. Las BD relacionales utilizan lenguajes no procedurales como SQL

La mayoría de los SGBD comerciales incluyen lenguajes de cuarta generación (4GL) que permiten al usuario desarrollar aplicaciones de forma fácil y rápida, también se les llama herramientas de desarrollo.

# Recursos humanos de las bases de datos.

Habíamos seleccionado cuatro tipos de usuarios (administradores/as, desarrolladores, diseñadores/as y usuarios/as)

**Informáticos**

Lógicamente son los profesionales que definen y preparan la base de datos. Pueden ser:

Directivos/as. Organizadores y coordinadores del proyecto a desarrollar y máximos responsables del mismo

Analistas. Son los encargados de controlar el desarrollo de la base de datos aprobada por la dirección. Normalmente son además los diseñadores de la base de datos

Administradores/as de las bases de datos. Encargados de crear el esquema interno de la base de datos

Desarrolladores/as o programadores/as. Encargados de la realización de las aplicaciones de usuario de la base de datos.

Equipo de mantenimiento. Encargados de dar soporte a los usuarios en el trabajo diario

**Usuarios**

Expertos/as. Utilizan el lenguaje de manipulación de datos (DML) para acceder a la base de datos.

Habituales. Utilizan las aplicaciones creadas por los desarrolladores para consultar y actualizar los datos

Ocasionales. Son usuarios que utilizan un acceso mínimo a la base de datos a través de una aplicación que permite consultar ciertos datos. Serían por ejemplo los usuarios que consultan el horario de trenes a través de Internet.

**El Administrador de la Base de Datos DBA**

El DBA tiene una gran responsabilidad ya que posee el máximo nivel de privilegios. Será el encargado de crear los usuarios que se conectarán a la BD. El objetivo principal de un DBA es garantizar que la BD cumple los fines previstos por la organización

* Instalar SGBD en el sistema informático.
* Crear las BBDD que se vayan a gestionar.
* Crear y mantener el esquema de la BD.
* Crear y mantener las cuentas de usuario de la BD.
* Arrancar y parar SGBD, y cargar las BBDD con las que se ha de trabajar.
* Colaborar con el administrador del S.O. en las tareas de ubicación, dimensionado y control de los archivos y espacios de disco ocupados por el SGBD.
* Colaborar en las tareas de formación de usuarios.
* Establecer estándares de uso, políticas de acceso y protocolos de trabajo diario para los usuarios de la BD.
* Suministrar la información necesaria sobre la BD a los equipos de análisis y programación de aplicaciones.

Efectuar tareas de explotación como:

* Vigilar el trabajo diario colaborando en la información y resolución de las dudas de los usuarios de la BD
* Controlar en tiempo real los accesos, tasas de uso, cargas en los servidores, anomalías, etcétera.
* Llegado el caso, reorganizar la BD.
* Efectuar las copias de seguridad periódicas de la BD.
* Restaurar la BD después de un incidente material a partir de las copias de seguridad.
* Estudiar las auditorías del sistema para detectar anomalías, intentos de violación de la seguridad, etcétera.
* Ajustar y optimizar la BD mediante el ajuste de sus parámetros, y con ayuda de las herramientas de monitorización y de las estadísticas del sistema.
* En su gestión diaria, el DBA suele utilizar una serie de herramientas de administración de la BD.
* Con el paso del tiempo, estas herramientas han adquirido sofisticadas prestaciones y facilitan en gran medida la realización de trabajos que, hasta no hace demasiado, requerían de arduos esfuerzos por parte de los administradores.

# Estructura multicapa.

El proceso que realiza un SGBD está en realidad formado por varias capas que actúan como interfaces entre el usuario y los datos. Fue el propio organismo ANSI

Este modelo toma como objeto principal al usuario habitual de la base de datos y modela el funcionamiento de la base de datos en una sucesión de capas cuya finalidad es ocultar y proteger la parte interna de las bases de datos.

**Facilidades de usuario**

Son las herramientas que proporciona el SGBD a los usuarios para permitir un acceso más sencillo a los datos. Actúan de interfaz entre el usuario y la base de datos, y son el único elemento que maneja el usuario. Son, en definitiva, las páginas web y las aplicaciones con las que los usuarios manejan la base de datos. Permite abstraer la realidad de la base de datos a las usuarias y usuarios, mostrando la información de una forma más humana.

**Capa de acceso a datos**

La capa de acceso a datos es la que permite comunicar a las aplicaciones de usuario con el diccionario de datos. Es un software (un driver o controlador en realidad) que se encarga traducir las peticiones del usuario para que lleguen de forma correcta a la base de datos y ésta pueda responder de forma adecuada.

**Diccionario de datos**

El diccionario de datos es el lugar donde se deposita información acerca de todos los datos que forman la BD. Es una guía en la que se describe la BD y los objetos que la forman

En una BD relacional, el diccionario de datos proporciona información acerca de

La estructura lógica y física de la BD.

* Las definiciones de todos los objetos de la BD: tablas, vistas, índices, disparadores, procedimientos,
* funciones, etcétera.
* El espacio asignado y utilizado por los objetos.
* Los valores por defecto de las columnas de las tablas.
* Información acerca de las restricciones de integridad.
* Los privilegios y roles otorgados a los usuarios.
* Auditoría de información, como los accesos a los objetos

Un diccionario de datos debe cumplir las siguientes características:

* Debe soportar las descripciones de los modelos conceptual, lógico, interno y externo de la BD.
* Debe estar integrado dentro del SGBD.
* Debe apoyar la transferencia eficiente de información al SGDB. La conexión entre los modelos

interno y externo debe ser realizada en tiempo de ejecución.

* Debe comenzar con la reorganización de versiones de producción de la BD. Además debe reflejar

los cambios en la descripción de la BD. Cualquier cambio a la descripción de programas ha de ser

reflejado automáticamente en la librería de descripción de programas con la ayuda del

diccionario de datos.

* Debe estar almacenado en un medio de almacenamiento con acceso directo para la fácil
* recuperación de información.

**Núcleo**

El núcleo de la base de datos es la encargada de traducir todas las instrucciones requeridas y prepararlas para su correcta interpretación por parte del sistema. Realiza la traducción física de las peticiones.

**Sistema operativo**

Es una capa externa al software SGBD pero es la única capa que realmente accede a los datos en sí. En realidad los SGBD no acceden directamente al disco, sino que piden al Sistema Operativo que lo haga.

# Funcionamiento del SGBD.

En esta arquitectura describe los datos a tres niveles de abstracción. En realidad los únicos datos que existen están a nivel físico almacenados en discos u otros dispositivos. Los SGBD basados en esta arquitectura permiten que cada grupo de usuarios haga referencia a su propio esquema externo.

El SGBD debe de transformar cualquier petición de usuario (esquema externo) a una petición expresada en términos de esquema conceptual

el SGBD es capaz de interpretar una solicitud de datos y realiza los siguientes pasos:

1. El usuario solicita unos datos y crea una consulta.
2. La Aplicación del usuario convierte esta consulta en un proceso realizado por el cliente del SGBD.
3. La consulta viaja a través de un medio (red).
4. El SGBD verifica y acepta el esquema externo para ese usuario. Se convierte el proceso del usuario en un “Proceso de Servidor” interno.
5. Transforma la solicitud al esquema conceptual.
6. Verifica y acepta el esquema conceptual.
7. El proceso lanzado por el usuario llama al SGBD indicando la porción de la base de datos que se desea tratar.
8. Transforma la solicitud al esquema físico o interno. El SGBD obtiene el esquema físico.
9. El SGBD traduce la llamada a los métodos de acceso del Sistema Operativo que permiten acceder realmente a los datos requeridos
10. Selecciona la o las tablas implicadas en la consulta y ejecuta la consulta. El Sistema Operativo accede a los datos tras traducir las órdenes dadas por el SGBD
11. Transforma del esquema interno al conceptual, y del conceptual al externo. Los datos pasan del disco a una memoria intermedia o buffer. En ese buffer se almacenarán los datos según se vayan recibiendo
12. ) Los datos pasan del buffer al área de trabajo del usuario (ATU) del proceso del usuario. Los pasos se repiten hasta que se envíe toda la información al proceso de usuario.
13. En el caso de que haya errores en cualquier momento del proceso, el SGBD devuelve indicadores en los que manifiesta si ha habido errores o advertencias a tener en cuenta. Esto se indica al área de comunicaciones del proceso de usuario. Si las indicaciones son satisfactorias, los datos de la ATU serán utilizables por el proceso de usuario.
14. Finalmente, el usuario ve los datos solicitados.

Para una BD específica sólo hay un esquema interno y uno conceptual, pero puede haber varios esquemas externos definidos para uno o para varios usuarios.

# Formas de ejecución de un SGBD

en realidad hay tres posibilidades de funcionamiento:

**SGBDs monocapa**: Es la más sencilla, pero la que tiene menos escalabilidad (posibilidad de crecer). El Sistema Gestor se instala en una máquina y los usuarios acceden directamente a esa máquina y ese Sistema Gestor. En estos sistemas no se accede de forma remota a la base de datos

**SGBDs bicapa**: Estructura clásica, la base de datos y su SGBD están en un servidor al cual acceden los clientes. El cliente posee software que permite al usuario enviar instrucciones al SGBD en el servidor y recibir los resultados de estas instrucciones. Para ello el software cliente y el servidor deben utilizar software de comunicaciones en red. Hay dos posibilidades:.

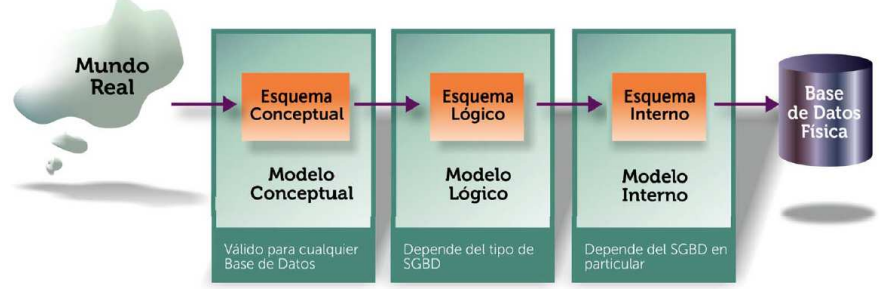
Estructura Cliente-Servidor. La base de datos está en un solo servidor al que acceden los clientes (incluso simultáneamente). ¬

Cliente Multi-servidor. En este caso los clientes acceden a un conjunto de servidores que distribuyen la base de datos.

**SGBD de tres o más capas:** Es una estructura de tipo cliente/servidor, pero en la que hay al menos una capa intermedia entre las dos. Esa capa se suele encargar de procesar las peticiones y enviarlas al SGBD con el que se comunica. Un ejemplo habitual es que la tercer capa sea un servidor web que evita que el cliente se conecte directamente al SGBD. Ese servidor web se encarga de traducir lo que el cliente realiza a una forma entendible por la base de datos.

# Tipos de Sistemas Gestores de Bases de Datos

**Introducción**

**** Como se ha visto en los apartados anteriores, resulta que cada SGBD puede utilizar un modelo diferente para los datos. Por lo que hay modelos conceptuales diferentes según que SGBD utilicemos

No obstante existen modelos lógicos comunes, ya que hay SGBD de diferentes tipos. En la realidad el modelo ANSI se modifica para que existan dos modelos internos

el modelo lógico y el modelo propiamente interno

Por lo tanto la diferencia entre los distintos SGBD está en que proporcionan diferentes modelos lógicos.

**Diferencias entre el modelo lógico y el conceptual**

* El modelo conceptual es independiente del DBMS que se vaya a utilizar. El lógico depende de un tipo de

SGBD en particular.

* El modelo lógico está más cerca del modelo físico, el que utiliza internamente el ordenador.
* El modelo conceptual es el más cercano al usuario, el lógico es el encargado de establecer el paso entre
* el modelo conceptual y el modelo físico del sistema.

Algunos ejemplos de modelos conceptuales son:

**Modelo Entidad Relación**

**Modelo RM/T**

**Modelo UML**

Ejemplos de modelos lógicos son:

**Modelo relacional**

**Modelo Codasyl**

**Modelo Jerárquico**

A continuación se comentarán los modelos lógicos más importantes.