

Fundamentos Tecnológicos para el Tratamiento y
Análisis de Datos

Tema 1. Introducción a las bases de datos

Índice

Esquema

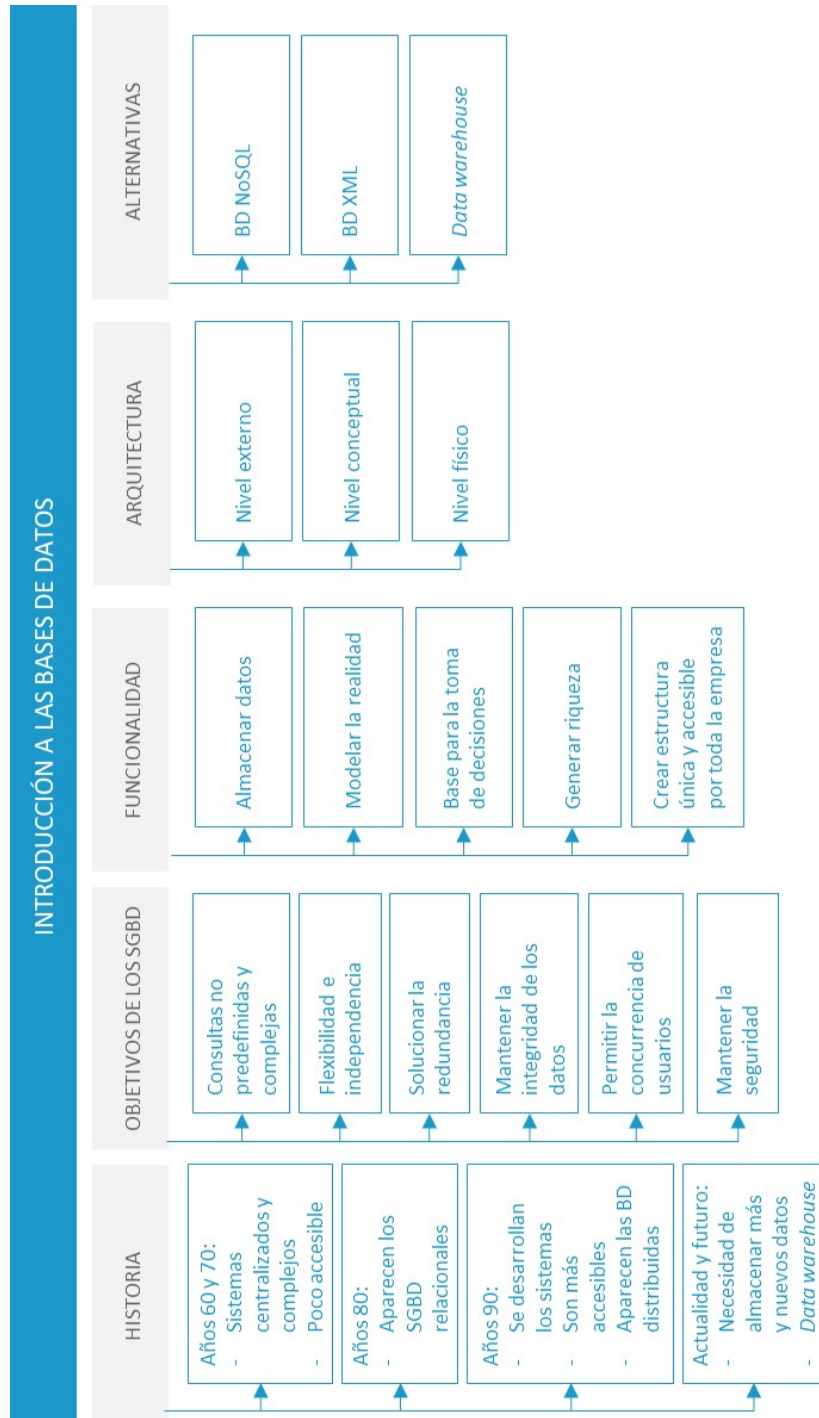
Ideas clave

- 1.1. Introducción y objetivos
- 1.2. Evolución histórica de las bases de datos y sus sistemas de gestión
- 1.3. Objetivos y funcionalidad de los SGBD
- 1.4. Arquitectura y modelos de los SGBD
- 1.5. Bases de datos vía tablas relacionadas (ejercicio práctico)
- 1.6. Alternativas y coste de implementación de las bases de datos en la empresa

A fondo

- Dispositivo de almacenamiento de datos
- Data warehouse

Test



1.1. Introducción y objetivos

Las empresas generan multitud de datos que pueden ser utilizados por los encargados de la inteligencia de negocio para llevar a cabo acciones y estrategias basadas en esos datos. La creciente fuente de información que dispone la empresa (datos de ventas, clientes, registros web, redes sociales etc.) hace necesario instalar un sistema de almacenamiento y gestión de datos en la empresa.

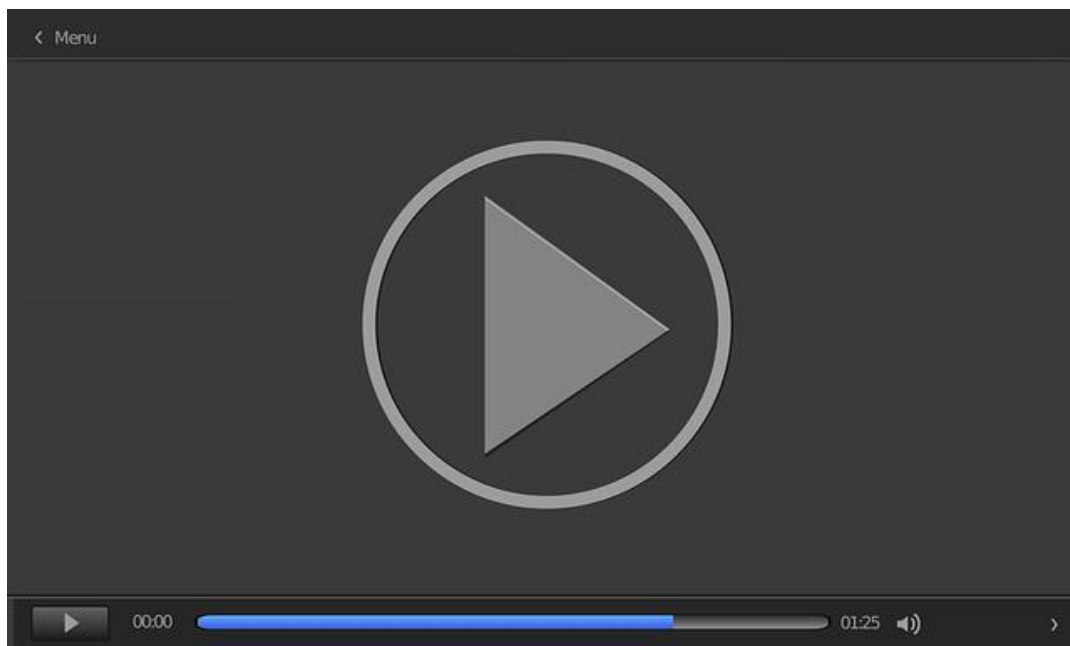
A la colección de datos relacionados entre sí con un significado implícito se le conoce como base de datos.

Esta base de datos (BD) representa algunos aspectos de la realidad de la empresa. Se diseña y construye para uno o varios propósitos específicos y está destinada a que un grupo de usuarios pueda utilizarla para obtener información y tomar decisiones. En resumen, una BD tiene una fuente de donde provienen los datos, que interaccionan entre sí de igual forma que lo hacen en la realidad, y una serie de usuarios interesados en su contenido.

Para conocer mejor qué son y para qué sirven las bases de datos, los objetivos del tema son:

- ▶ Conocer la evolución histórica de las bases de datos y sus sistemas de gestión, para entender las necesidades que han ido surgiendo.
- ▶ Entender cuáles son los objetivos a la hora de implantar una base de datos en la empresa.
- ▶ Saber cuáles son las distintas funcionalidades de una base de datos.
- ▶ Conocer la arquitectura de las bases de datos para entender su estructura.

- ▶ Entender los diferentes modelos que existen y su utilidad.
- ▶ Conocer las diferentes alternativas para almacenar datos que existen y los principales sistemas de gestión de bases de datos.



Introducción a las bases de datos

Accede al vídeo:

<https://unir.cloud.panopto.eu/Panopto/Pages/Embed.aspx?id=b51492dc-5a52-4036-b30d-b16c0095a888>

1.2. Evolución histórica de las bases de datos y sus sistemas de gestión

En los años setenta, con el inicio de los computadores y los sistemas de almacenamiento de archivos, se creaban ficheros cada vez que un programa necesitaba añadir una nueva aplicación. Esto hacía que los archivos estuvieran almacenados tantas veces como fuera necesario. A medida que se introdujo la comunicación, los terminales y los discos, se crearon programas que permitían a varios usuarios acceder a ficheros de forma simultánea.

No fue hasta que las aplicaciones comenzaron a aunarse, que se inició la integración de, por ejemplo, la aplicación de facturas con las de pedidos y almacenamiento de productos; comenzando a relacionar los ficheros con esas aplicaciones y eliminando su redundancia.

De manera lógica, el DNI o la dirección de los clientes que podían aparecer en diferentes ficheros debían estar ahora en un solo lugar. Este conjunto de ficheros que contenían información y estaban interrelacionados entre sí, que tenían estructuras complejas y eran compartidos por varios usuarios, recibieron en primer lugar el nombre de «*data banks*» y, posteriormente, «*data bases*» o lo que conocemos por bases de datos (BD).

Una BD es un conjunto estructurado de datos que representa entidades y sus interrelaciones.

El uso de este conjunto de ficheros por parte de las aplicaciones que necesitaban acceder a su información era bastante complejo y requería una arquitectura complicada. Por ello, a partir de la segunda mitad de los años setenta comenzaron a salir al mercado lo que se conoce como **sistemas de gestión de bases de datos**

(SGBD).

Un SGBD se encarga de gestionar las bases de datos y hacer de intermediario con el usuario de esta debido a su complejidad.

Décadas de los sesenta y setenta

En los primeros años, los SGBD estaban orientados a facilitar la utilización de grandes conjuntos de datos con interrelaciones concretas. Se utilizaban mayoritariamente en la industria del automóvil, la construcción de naves e industrias similares.

Estos sistemas estaban totalmente **centralizados** y necesitaban que su programador conociese todos los detalles del diseño físico. Además, cada vez que se realizaban cambios en las BD debían ser modificados; por lo que eran sistemas muy complejos de implementar en las empresas. El *hardware* que se utilizaba para mantener aquellas BD y los SGBD consistía en un gran computador para toda la empresa y una red de terminales sin inteligencia ni memoria.

IBM fue de las primeras compañías en crear un *software* de comunicación y gestión de transacciones y datos. Las aplicaciones típicas de este eran la reserva y compra de billetes a las compañías aéreas y ferroviarias y, poco más tarde, las cuentas de clientes de los bancos.

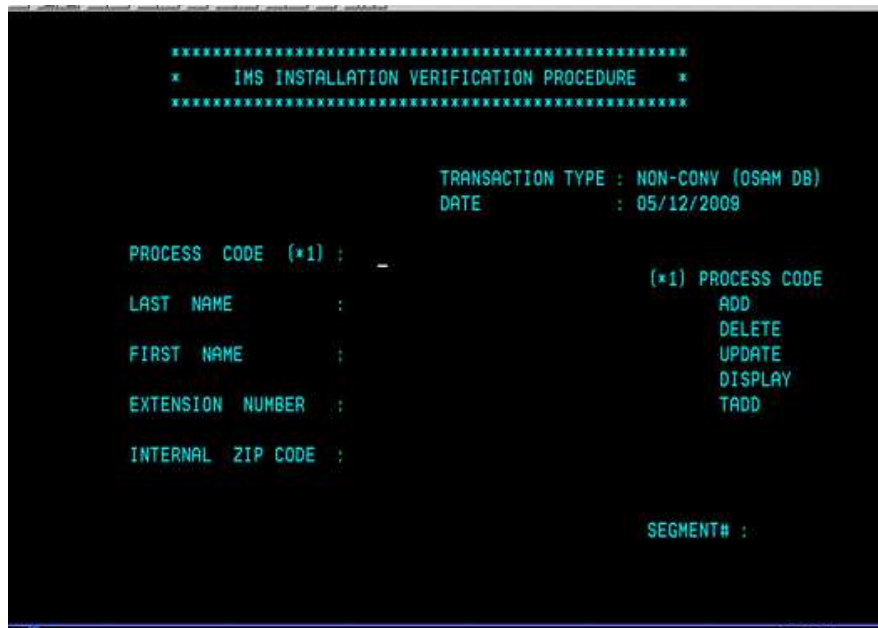


Figura 1. Pantalla de IMS de IBM.Fuente: <https://www.sutori.com/item/untitled-f9e4-e61a>

Década de los ochenta

La accesibilidad a computadores y la reducción de su tamaño y coste provocó que prácticamente todas las empresas e instituciones tuvieran uno. Este hecho exigía que los SGBD fueran más sencillos porque hasta la fecha solo podían ser utilizados por un personal muy cualificado.

En este punto aparecieron los **SGBD relacionales**, cuya filosofía se centra en utilizar un conjunto de tablas donde se almacenan los datos y en las relaciones que existen entre las diferentes tablas.

Ejemplo: Base de datos relacional

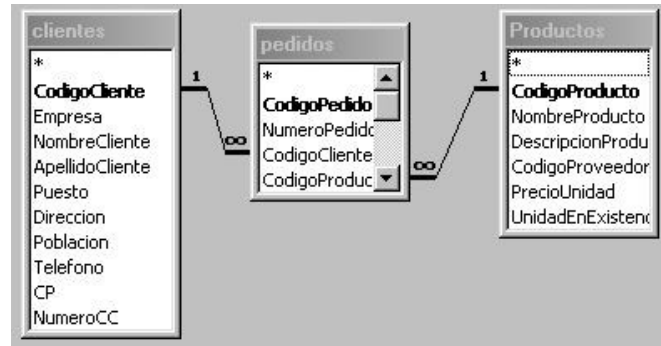


Figura 2. Base de datos relacional. Fuente:

<http://resumenbasededatos.blogspot.com/2015/03/base-de-datos-racionales.html>

Una base de datos relacional puede contener tres tablas, en este caso, dos de ellas con información de la empresa, clientes y productos y una tercera que contenga la relación que se establece entre las dos anteriores cuando un cliente compra un producto.

La aparición de estos SGBD relacionales supuso un gran avance a la hora de facilitar la programación de las aplicaciones y conseguir programas independientes de los aspectos físicos de las BD, que no tengan que ser actualizados con cada cambio que se produzca en esta. Además, se hizo común el uso del **lenguaje SQL**, un lenguaje de programación sencillo de utilizar, por el cual una persona se relaciona con el SGBD.

Década de los noventa

Al comenzar los años noventa, el uso de los SGBD ya se había generalizado, pero comenzó a surgir un nuevo problema debido a que todos los departamentos de las empresas usuarias disponían de varios computadores y numerosas BD y SGBD diferentes.

Fue entonces cuando surgió la necesidad de interrelacionar las diferentes

aplicaciones, los SGBD y BD de todos los departamentos de la empresa, para obtener una visión global de esta, y que se solucionó (y se soluciona) con la **homogeneidad** de los diferentes SGBD y la **estandarización del lenguaje SQL** de consulta.

Comenzó a aplicarse lo que se conoce como **base de datos distribuida**, donde diferentes BD son soportadas por un único SGBD. Sus ventajas están recogidas en la figura 3:

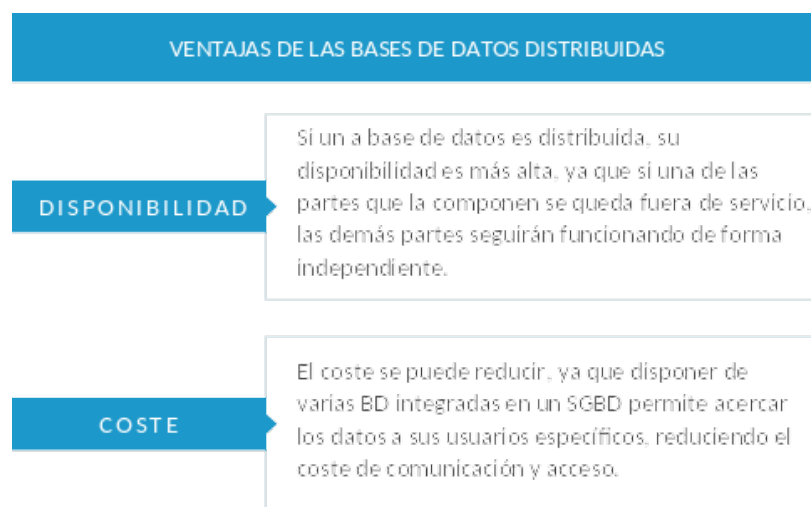


Figura 3. Ventajas de las bases de datos distribuidas.

En definitiva, cuando comenzó el siglo XXI, las BD y los SGBD funcionaban como recoge la figura 4, donde un usuario a través de un programa realiza una petición al servidor del SGBD y este, a su vez, al servidor de las distintas BD.

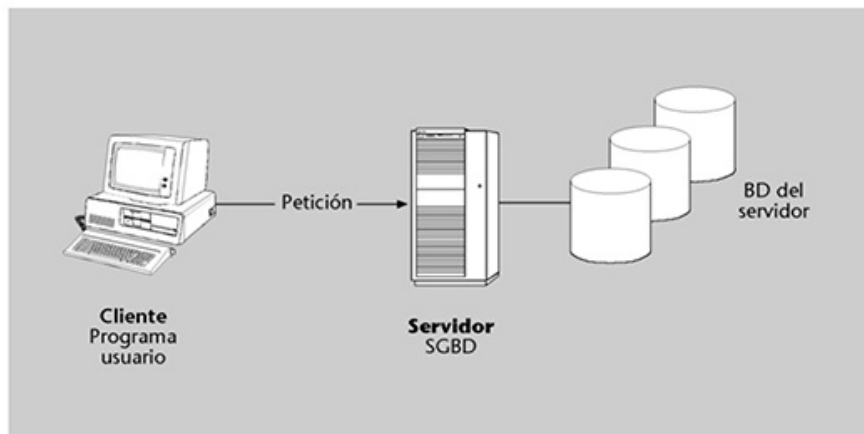


Figura 4. Proceso de consulta. Fuente: <https://app.emaze.com/@AZIRLQQQ#3>

Actualidad y futuro

Debido a las nuevas tecnologías relacionadas con lo multimedia, Internet o las web, las bases de datos continúan en plena transformación para poder albergar los datos que generan estas. En los años anteriores, los tipos de datos que se podían almacenar y gestionar en estas bases de datos eran muy limitados; las BD y los SGBD no estaban adaptados para almacenar otro tipo de dato más complejo, como puede ser el sonido o la imagen.

En la actualidad han comenzado a surgir nuevos lenguajes, BD y SGBD capaces de almacenar y gestionar este tipo de dato. Un ejemplo de esto son los *data warehouse*, cuya principal finalidad es almacenar información para realizar consultas y análisis de los diferentes departamentos de la empresa, como pueden ser los de análisis financiero o estratégico.

1.3. Objetivos y funcionalidad de los SGBD

Conocer cuál es la utilidad de los SGBD y sus posibles aplicaciones y posibilidades será útil a la hora de relacionarnos con ellos para extraer la información que necesitemos de estos.

Objetivos

Consultas no predefinidas y complejas

El objetivo principal de los SGBD es permitir que se realicen consultas específicas y complejas a la base de datos, el primero responderá automáticamente con la respuesta tras hacer la consulta en la BD. Veamos el siguiente ejemplo para entenderlo mejor:

Ejemplo: Consulta a un SGBD

Para realizar una campaña de publicidad se necesita conocer la información específica de los clientes de un determinado banco. Por ello, es necesario seleccionar a un tipo de cliente que cumpla ese requisito y así poderle enviar un correo ofreciéndole nuevos productos. En concreto, se realizará una consulta al SGBD para que extraiga la información que se le pide de los clientes:

- ▶ Se necesita conocer el nombre y la dirección de correo electrónico de todos aquellos clientes con una nómina domiciliada superior a 33 000€ anuales.
- ▶ Que, además, tengan hipoteca contratada con el banco.
- ▶ Y que hayan pedido un préstamo personal en 2019.

A través del lenguaje de consulta (generalmente SQL) se introduce el

código preciso para que el SGBD realice las operaciones necesarias, conecte la información contenida en la BD y extraiga una tabla con el nombre y correo electrónico de todas las personas que cumplan con esos requisitos.

Este proceso de consultas se hace de forma rápida y sencilla con el sistema de BD y SGBD. En el caso de que la información estuviera contenida en ficheros tradicionales, la forma de automatizar esta consulta habría sido mediante la creación de un programa a medida mucho más complejo y costoso.

Flexibilidad e independencia

Tener que interactuar directamente con una BD podría ser bastante complicado para el usuario final, ya que este debería conocer su estructura, la información contenida y, además, debería adaptarse a cada cambio que se produzca en esta.

Por ello, realizar todas las operaciones a través de un SGBD da independencia al usuario final, que no tiene que conocer absolutamente nada de la arquitectura de la BD para realizar las consultas. Además, cualquier variación que se produzca en esta no afecta al modo en el que el usuario final se comunica con el SGBD, ya que todo estará debidamente implementado en la relación que existe entre el SGBD y las BD.

Un ejemplo de flexibilidad e independencia que permiten los SGBD sería el siguiente:

Ejemplo: Flexibilidad e independencia

Imaginemos la situación del ejemplo anterior donde se realizaba una consulta con fines de marketing para un banco en la que el correo electrónico era importante con el fin de llevar a cabo una campaña. Imaginemos que en el departamento financiero de la empresa pretende realizar la misma consulta, pero su interés no reside en el correo electrónico, sino en el número de cuenta.

En este caso, el SGBD es flexible y permite hacer con la misma información consultas diferentes. Además, la independencia de la información contenida permite que se realicen cambios, por ejemplo, en el correo electrónico sin que esto influya en la consulta que desea realizar el departamento financiero.

Problemas de redundancia

El SGBD elimina la redundancia de los datos almacenados en las BD. En un sistema de almacenamiento de archivos tradicional, cada aplicación necesita sus propios archivos, por lo que varios archivos de diferentes aplicaciones pueden contener la misma información.

El SGBD se encarga de **optimizar el almacenamiento** para evitar en mayor medida la redundancia, aunque, en ocasiones y por distintos motivos, determinada información debe estar recogida simultáneamente varias veces. Uno de los principales motivos por los que sucede esto es para poder identificar datos que están relacionados entre sí.

Integridad de los datos

La integridad de los datos hace referencia a los **requisitos** y las **reglas** establecidas que tiene que cumplir la BD. Por ejemplo, para el caso de la información de los clientes del banco, el DNI es un requisito indispensable para poder identificarlos y no se podrá introducir información de ningún cliente a la base de datos si no se posee su DNI.

Además de este ejemplo, son muchas las posibilidades de integridad que se pueden crear con los SGBD y que son automatizadas cuando se quiera introducir o modificar información.

Concurrencia de usuarios

Uno de los objetivos también fundamentales de los SGBD es permitir que varios usuarios tengan **acceso simultáneo** a la misma BD. Cuando varios usuarios utilizan una, puede ocurrir que, cuando se está modificando el contenido, otro usuario pueda captar información sin esa modificación que se ha llevado a cabo.

Para ello, los SGBD utilizan el concepto de **transacción**, donde cualquier modificación que se realice en la base de datos se hará de forma íntegra y total, alterando todos los datos que se vean afectados.

Seguridad

Los SGBD permiten a través de técnicas de encriptación dar seguridad a la información contenida y generar **derechos de acceso** a los diferentes usuarios que puedan consultar la base de datos.

No todos los usuarios de la BD podrán realizar las mismas acciones ni acceder a la misma información. Por ejemplo, los trabajadores del departamento de marketing no podrán realizar ninguna modificación en el número de cuenta de los clientes de un banco o no podrá acceder a la información contable del banco.

Funcionalidad

Ya hemos visto cuáles son los objetivos de los SGBD, pero vamos a detallar cuál es su funcionalidad y cuáles son sus ventajas principales respecto a los tradicionales sistemas de almacenamiento en archivos. La tabla 1 recoge las principales funciones que cubre un SGBD:

FUNCIONES DE LAS BD Y LOS SGBD
Almacenar los datos de la empresa.
Modelar con datos la realidad de una empresa.
Servir como base de la toma de decisiones en la empresa.
Generar riqueza a partir de la información detallada .
Crear una estructura única y accesible de datos para la empresa.

Tabla 1. Funciones de las BD y los SGBD.

Como hemos mencionado anteriormente, los SGBD y las BD vienen a remplazar el tradicional uso de almacenar y gestionar datos través de ficheros que es más complejo, toma más tiempo y permite realizar menos funciones. La tabla 2 recoge las principales ventajas que posee los SGBD y las BD sobre los sistemas de ficheros.

VENTAJAS DE LAS BD FRENTE A LOS FICHEROS
La BD no solo son datos, también tiene una naturaleza descriptiva de la realidad de la empresa.
Independencia entre los programas y los datos.
Coherencia en los resultados de las consultas.
Mejor y mayor disponibilidad de los datos.
Eficiencia en la recogida, validación e introducción de los datos.
Reducción del espacio de almacenamiento al eliminar la redundancia.
Soporta el uso simultáneo de varios usuarios.

Tabla 2. Ventajas de las BD frente a los ficheros.

1.4. Arquitectura y modelos de los SGBD

Para conocer los SGBD es necesario conocer su arquitectura y los modelos de bases de datos que existen. Un SGBD tiene una estructura determinada que permite su uso de forma sencilla, además, estos son representaciones de la realidad a través de modelos que deben asimilarse lo máximo posible a la estructura que se esté representando.

Arquitectura

Niveles y esquemas

Los SGBD necesitan que les proporcionemos una definición de la estructura de la BD y sus principales componentes. Este esquema que se realiza es fundamental y permite **independizar** el SGBD de la BD, sin que modificaciones en esta última afecten al SGBD.

Las bases de datos siguen dos tipos de niveles:

- ▶ El nivel **lógico** que no muestra detalles de cómo se almacenan los datos; se accede físicamente a ellos y tan solo se muestran sus componentes principales:
 - Entidades (clientes, vendedores, productos).
 - Atributos (DNI, ID, precio).
- ▶ El nivel **físico** detalla la estructura física de la BD, por ejemplo, que índices habrá y que características tendrán, en qué espacio físico se agrupan los registros, el tamaño máximo de las diferentes tablas, etc.

La primera arquitectura que se creó, **ANSI/SPARC**, tenía tres niveles. El nivel lógico estaba dividido en dos: nivel externo y nivel conceptual; mientras que el tercer nivel era el interno o lo que hemos definido como nivel físico (véase figura 5):

- ▶ El nivel **externo** es aquel con el que los usuarios de los programas y aplicaciones interaccionan, se denomina esquema externo. Es una división del esquema conceptual donde solo se citarán aquellos atributos y entidades que sean de interés para la aplicación en concreto y los usuarios específicos.
- ▶ El nivel **conceptual** es donde se realiza la descripción lógica, básica y global y que sirve de referencia para el resto de los esquemas, se denomina esquema conceptual. Este corresponde al conjunto de necesidades de almacenamiento que posee la empresa en su totalidad.
- ▶ Por último, en el nivel **físico** se encuentra la descripción física de la base de datos, lo que se conoce como esquema interno. Este contendrá la descripción de las rutas, los accesos, la codificación de los datos o la gestión del espacio, entre otros.

Los **SGBD relacionales** utilizan una terminología diferente y los niveles no están separados de forma tan clara. En concreto, existe un solo esquema, aunque se detallan igualmente los tres niveles.

Estos sistemas relacionales son los más comunes y están basados solo en el mundo lógico, ya que la parte física se deja en manos de las compañías que crean los SGBD. Este es uno de los motivos por el que los sistemas relacionales son los más utilizados, ya que los usuarios del SGBD solo tienen que limitarse a estructurarlo desde un punto de vista lógico, describiendo cuál es el esquema necesario para satisfacer las necesidades de la empresa o entidad en cuestión.

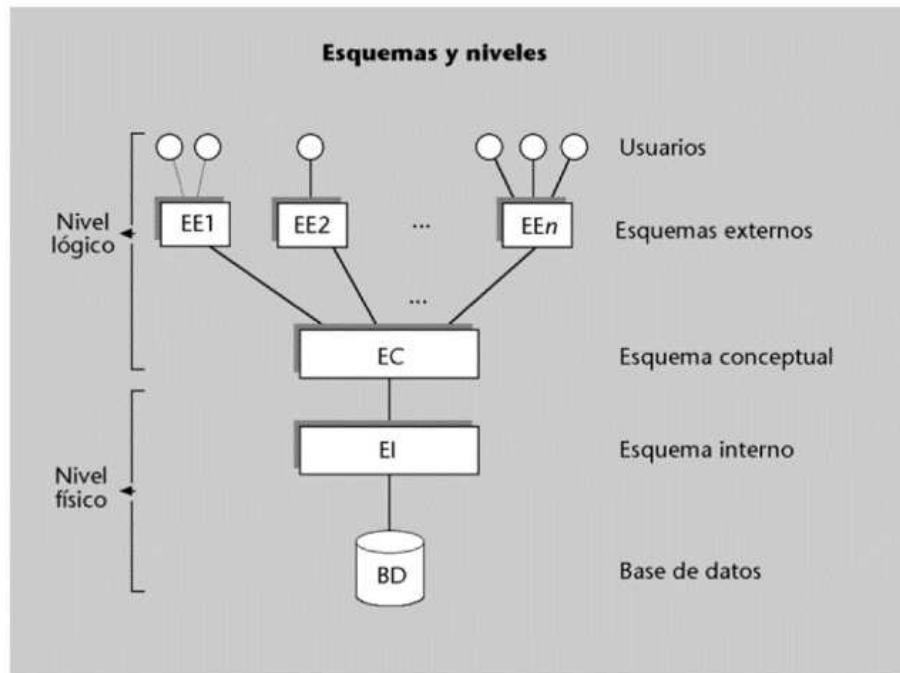


Figura 5. Esquema y niveles ANSI/SPARC.

Independencia y flujo de datos

Esta arquitectura que hemos descrito permite que exista independencia entre los niveles, es decir, el usuario final es totalmente independiente a los cambios que se produzcan en los otros niveles.

Tanto los cambios en el esquema conceptual como los cambios en el esquema interno solo afectarán a aquellas aplicaciones y usuarios que se relacionaban con la parte que ha sido modificada; lo que permite gestionar de forma más sencilla y dinámica el SGBD.

Por ejemplo, si eliminamos el apellido de los clientes de un banco afectará a los departamentos que operen con los apellidos de los clientes, mientras que los trabajadores del departamento financiero, que identifican a los clientes a través del número de cuenta y tienen capado el acceso a los nombres de los clientes, no se verán afectados.

Para terminar de conocer la arquitectura de la base de datos es necesario ordenar los pasos que ocurren cuando se realiza una **consulta** a la base de datos a través de la aplicación (véase figura 6).

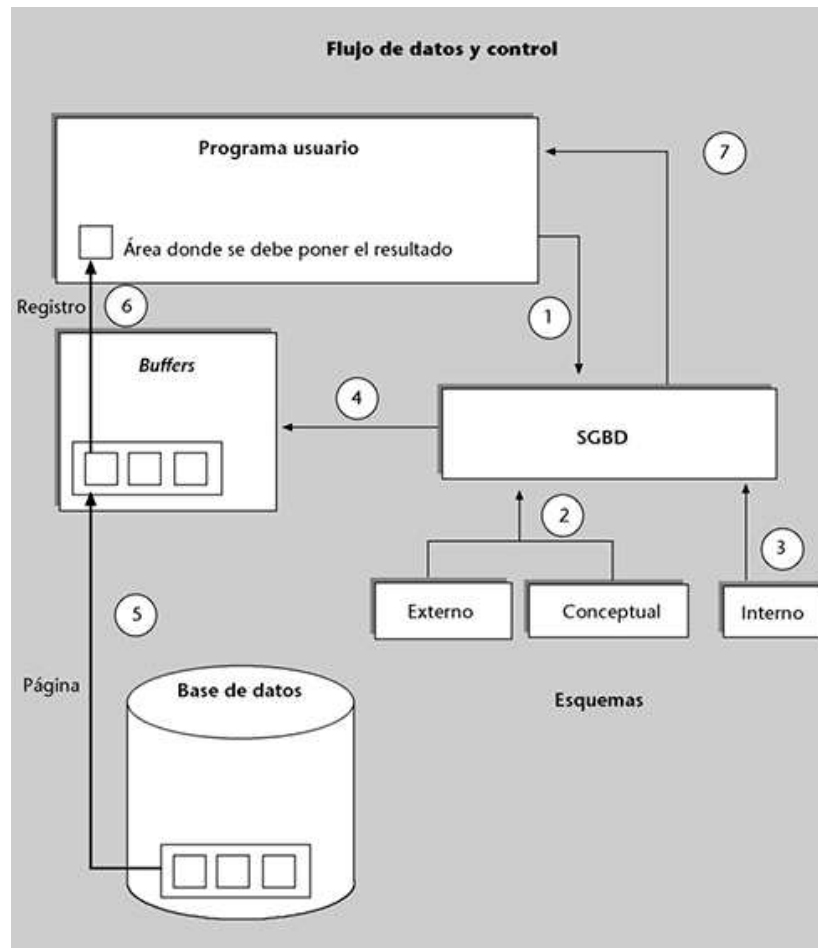


Figura 6. Proceso interno de consulta.

Para ello imagina que trabajas en el área de inteligencia de negocio para una cadena de tiendas de zapatillas; necesitas saber el precio de todas las zapatillas de color blanco que se han vendido en las tiendas de Extremadura y Andalucía desde la apertura de cada tienda con el objetivo de ver la distribución de precios que tienen en esa zona. Por tanto, a través de la aplicación disponible se realiza una consulta.

1. En primer lugar, al introducir la consulta el programa realiza una llamada al **SGBD (1)**

y este, basándose en el programa externo y conceptual, comprueba que todo es correcto, por ejemplo, a nivel sintáctico (que exista un producto llamado zapatillas y además tenga un atributo «color» y dentro de los colores exista el blanco, etc.) o que el usuario tenga permiso para conocer aquellos **campos (2)** que está solicitando.

2. Si la consulta es válida, el SGBD consultará el **nivel interno (3)**, cuál es el procedimiento a seguir para hacer efectiva esa consulta y dónde se sitúa esa información requerida.
3. Una vez que se comprueba esto, el SGBD consulta si la información solicitada ya se encuentra cargada en los *buffers* debido a que se haya realizado con anterioridad la **misma consulta (4)**. En caso de no ser así, deberá acceder al **disco (5)** donde se guarda la información para extraerla.
4. Por último, el SGBD aplica todas las transformaciones necesarias indicadas por el esquema externo y devuelve el **resultado al programa (6)**, también el control al programa y finaliza la operación de consulta (7).

Modelos

Las bases de datos sirven para almacenar datos que representan la realidad de las empresas y entidades. De hecho, la estructura de estas BD debe ser lógica con la realidad, por ejemplo, un cliente incluido en la base de datos solo podrá comprar productos de una tienda y nunca una operación podrá ser que un cliente compre un empleado que también se encuentre registrado.

Por tanto, tanto los elementos que conforman las BD como las relaciones entre ellos deben tener un **sentido lógico**, conforme a la realidad de la empresa u organización.

Los modelos son el conjunto de componentes y herramientas conceptuales que definen una BD, en concreto, proporcionan una estructura de datos, restricciones, reglas de integridad y un conjunto de operaciones que se pueden realizar con la BD.

Existen cuatro modelos que son los más utilizados a lo largo de la historia, que han ido apareciendo y perfeccionándose con el paso del tiempo.

Modelo jerárquico

Fue el primer modelo que apareció en el que su estructura estaba relacionada en forma de árbol como se muestra en la figura 7.

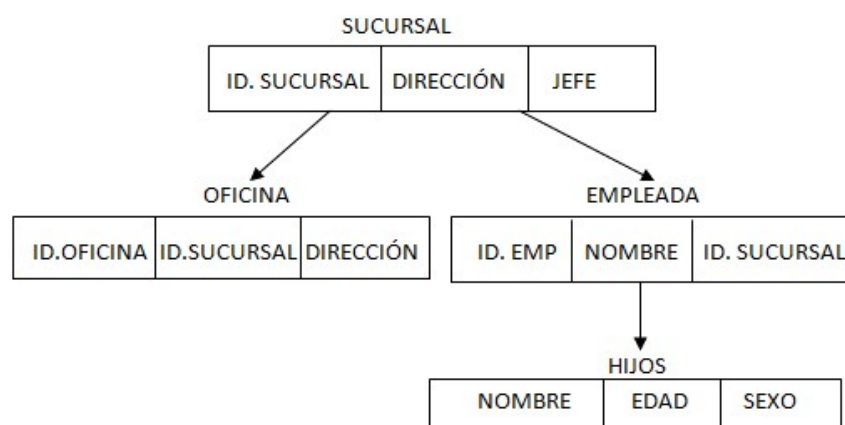


Figura 7. Modelo jerárquico.

Modelo en red

Más tarde, en los años setenta nacieron los modelos en red que son una derivación de los jerárquicos, en los que un registro puede proceder de varios superiores. En esta época nació el que es hoy en día es el modelo de bases de datos más utilizado, el modelo relacional y el lenguaje utilizado para interactuar con él, el SQL.

Modelo relacional

Este modelo se basa en el concepto de relación, donde una entidad (cliente, producto, vendedores) se relaciona con otra y esto se consigue a través de tablas.

Las **tablas** son la principal diferencia con los modelos anteriores ya que, hasta el modelo relacional, las estructuras de elementos de una BD constaban de dos elementos (los registros y las interacciones) mientras que los modelos relacionales solo constan de uno (la tabla).

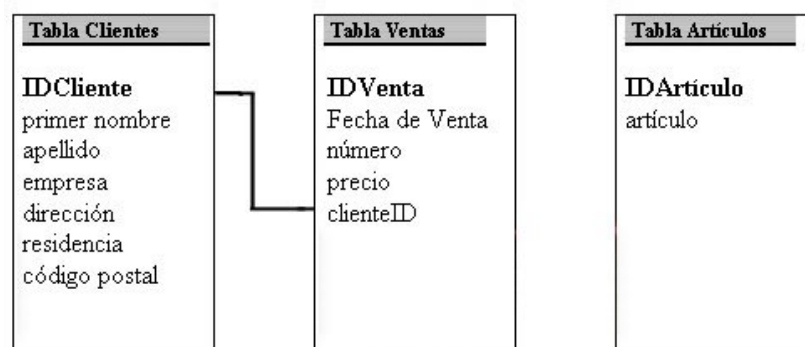


Figura 8. Modelo relacional.

Modelo orientado a objetos

También existen (son más recientes) los modelos relacionales con objetos, que son idénticos a los modelos relacionales, pero permiten la introducción de nuevos tipos de datos.

Los modelos vistos hasta ahora sirven para estructurar la BD y los SGBD en su parte interna y técnica, pero existen otros que proporcionan herramientas muy útiles para describir, a través de estructuras, el mundo real y las interrelaciones entre los diferentes agentes. Estos modelos son desarrollados para ocuparse de la parte lógica y, posteriormente, servir como referencia para la construcción de los modelos de BD.

El más utilizado es el **modelo entidad-relación**, que veremos en los próximos temas y servirá para transformar la realidad en un esquema lógico.

1.5. Bases de datos vía tablas relacionadas (ejercicio práctico)

En una base de datos operativa existen **multitud de tablas de diferente índole** que están interrelacionadas entre sí para asemejarse a la compleja realidad empresarial. Tal y como observamos en la figura 8, podemos percibir visualmente la posible relación entre las mismas. Para dicho fin introduciremos un concepto que veremos más detenidamente en los temas venideros: la **utilización de claves**.

En cada una de las tablas presentes en una base de datos existen diferentes claves. A aquellas que proporcionan información e identifican cada una de las filas de forma única se les denomina **claves primarias**. Por ejemplo, supongamos que tenemos una tabla de clientes compuesta por los siguientes atributos o columnas:

IDCliente	Nombre	Apellido	Empresa	Dirección	Residencia	CP
12	Manuel	Rodríguez	SQL SL	Calle 3	Madrid	28001
13	Manuel	López	SQL SL	Calle 2	Madrid	28002
14	José	López	NSQL SL	Calle 3	Barcelona	08001

Figura 9. Tabla Cliente.

Si analizamos detenidamente la tabla observamos que solo en dos columnas los valores son únicos («IDCliente» y «CP»). Argumentando de forma lógica, el «CP» no puede ser una clave primaria, dada la posibilidad de que dos o más clientes habiten en un mismo distrito. Esto deja a la vista la inherente capacidad de «IDCliente» de ser la clave primaria. Supongamos que se añade un nuevo cliente (por ejemplo, José López, que trabaja en SQL SL y que reside en calle 4 de Barcelona con CP 08001). A este se le añadiría en la tabla con el IDCliente 15, al tiempo que se definiría dicha fila o tupla de forma única.

En la figura 10 observamos que existe una **relación con la tabla de ventas** y precisamente se realiza a través de «IDCliente». La tabla de ventas tiene la siguiente estructura:

IDVenta	Fecha	Número	Precio	IDCliente
456	12/06/2017	42	50€	12
457	12/06/2017	41	50€	12
462	09/08/2018	42	50€	13

Figura 10. Tabla Ventas.

Podemos observar que, entre los atributos o columnas, el que identifica de forma única cada fila o tupla es «**IDVenta**», considerándola por tanto clave primaria de la tabla Ventas. Observamos que existe un «**IDCliente**», que en este caso tiene la característica de clave ajena y no de clave primaria, al no identificar las filas de forma única y estar presente en más de una fila; por ejemplo, en el caso del IDCliente 12. Por lo tanto, «IDCliente» es clave primaria en la tabla Cliente y clave ajena en la tabla Ventas, con lo que actúa de conector para la relación entre ambas tablas. Por ejemplo, observamos que la venta con ID 462 se realizó el 9 de agosto de 2018 por el cliente 13, Manuel López, que reside en Madrid.

En la última parte del tema se presentará una tarea referente a la creación de una base de datos vía tablas y, para ello, realizaremos a continuación un ejercicio de índole similar:

Un gran conglomerado de ventas de coches almacena en una base de datos información sobre fábricas, concesionarios, clientes y coches.

- ▶ Se desea almacenar la información que generan las siguientes circunstancias:
 - Los concesionarios obtienen coches de una sola fábrica.

- Los concesionarios venden coches.
- Los clientes compran un coche.
- ▶ Los datos que se almacenan en la base de datos son los siguientes:
 - De las fábricas se conoce su número («Nº»), nombre («Nombre») y provincia («Provincia»).
 - De los concesionarios se conoce el nombre («Nombre»), código identificativo («Cod»), ciudad («Ciudad») y director («Director»).
 - De los coches se conoce el código identificativo de la marca y modelo («IDMM»), nombre de la marca («NM»), el modelo, el color y el precio.
 - De los clientes se conoce el nombre («Nombre»), los apellidos («Apellidos»), el código postal («CP») y el número de cliente («NC»).

Por tanto, el contenido de las tablas queda de la siguiente forma:

FABRICAS (Nº; Nombre; Provincia).

CONCESIONARIO (Nombre; Cod; Ciudad; Director).

COCHES (IDMM; NM; Modelo; Color; Precio).

CLIENTES (Nombre; Apellidos; CP; NC).

Las pautas del presente ejercicio consisten en:

- ▶ Se deben seleccionar, en primer lugar, las claves primarias de cada tabla e incorporarlas como clave ajena en aquellas donde sea necesario para cumplir con las circunstancias anteriormente comentadas.

- ▶ Se crearán cuatro tablas: Fábrica, Concesionario, Coches y Clientes, de cardinalidad (número de tuplas) 4 y grado (número de atributos) correspondiente con datos inventados por el alumno, que no creen inconsistencias en la base de datos creada.
- ▶ Será necesario explicar el contenido de las tablas y el razonamiento de la elección de las claves primarias y su correspondiente uso como clave ajena.

Comencemos seleccionando cuáles son las claves primarias de las tablas propuestas. En referencia a las **fábricas**, ni el nombre ni las provincias pueden serlo, debido a que pueden estar alojadas en una misma provincia e incluso contener el mismo nombre. El valor numérico «**Nº**» es el idóneo para ser la clave primaria. Respecto a los concesionarios, estos pueden tener un director con el mismo nombre y apellidos; también puede existir más de un concesionario en una misma ciudad e, incluso, tener el mismo nombre. El atributo «**Cod**» incluye implícitamente la identificación de cada concesionario, lo cual le otorga el valor de clave primaria. Siguiendo con la tabla Coches, el atributo que incluye explícitamente la identificación de la marca y el modelo es «**IDMM**», con lo que esta es la clave primaria, puesto que una misma marca de coches puede tener más de un modelo, un mismo color e, incluso, un mismo precio. Finalmente, en la tabla Clientes, un cliente puede compartir nombre, apellido y código postal; por lo consiguiente, la forma única de identificarlos es a través de número de cliente («**NC**»).

A continuación, es necesario **dotar de valores** a las presentes tablas. En las siguientes figuras podemos apreciar una serie de valores íntegros y aplicables, los cuales cumplen con las restricciones asociadas y con las cardinalidades y grados correspondientes. Las columnas subrayadas en verde se corresponden con la clave primaria y las subrayadas en naranja, con la ajena.

Fábricas		
Cardinalidad 4; Grado 3		
Nº	Nombre	Provincia
54	Alfasa	Madrid
98	Mandasa	Barcelona
45	DiCar SL	Asturias
63	GIBatad	Navarra

Figura 11. Tabla Fábricas.

Concesionarios		Cardinalidad 4; Grado 3			
Nombre	Cod	Ciudad	Director	Nº	De la tabla fábricas
AutosGarrido	354	Sevilla	Muñoz	54	
MotorIbérica	498	Sevilla	González	63	
STR Motor	321	Málaga	Rodríguez	98	
Deluxe Cars	786	Huelva	Gómez	45	

Figura 12. Tabla Concesionarios.

Coches			Cardinalidad 6; Grado 5			
IDMM	NM	Modelo	Color	Precio	Cod	De la tabla Concesionarios
85	Taurus	Diesel	Verde	15000	354	
14	Rayo	Gasolina	Rojo	12000	786	
35	Lada	Híbrido	Celeste	25000	321	
24	Duster	Eléctrico	Gris	24500	321	

Figura 13. Tabla Coches.

Clientes			Cardinalidad 5; Grado 5		
Nombre	Apellidos	CP	NC	IDMM	De la tabla Coches
José	Gómez	21458	7896	35	
Francisco	Rodríguez	45235	4567	24	
Daniel	Rey	47899	1256	85	
David	Roldán	21005	3478	14	

Figura 14. Tabla Clientes.

Finalmente, para dotar de mayor poder explicativo a las relaciones existentes entre las tablas añadimos **una descripción por fila** de cada una de ellas. Por ejemplo, El cliente José Gomez (7896) ha comprado el coche Lada Híbrido (35), que fue vendido por el concesionario STR Motor (321), el cual adquiere coches en la fábrica Mandasa (98). El cliente Francisco Rodriguez (4567) ha comprado el coche Duster Eléctrico (24), que fue vendido por el concesionario STR Motor (321), el cual adquiere coches en la fábrica Mandasa (98). El cliente Daniel Rey (1256) ha comprado el coche Taurus Diesel (85), que fue vendido por el concesionario AutosGarrido (354), que adquiere coches en la fábrica Alfasa (54). Por último, el cliente David Roldan (3478) ha comprado el coche Rayo Gasolina (14), que fue vendido por el concesionario Deluxe Cars (786), que adquiere coches en la fábrica DiCar SL (45).

1.6. Alternativas y coste de implementación de las bases de datos en la empresa

Cuando se estudia la implantación de una base de datos o su reestructuración en una empresa se han de tener en cuenta varios factores que pueden ser determinantes a la hora de elegir entre las diferentes alternativas disponibles en el mercado.

- ▶ El primero de ellos es el **volumen de datos**, con el cual pueden cometerse dos errores: acumular pocos datos (tan solo los necesarios) o acumular demasiados, lo que puede dificultar la generación de valor a partir de estos, además, de su rendimiento y coste.
- ▶ El segundo factor es el **uso de los datos**. Hay que tener claro qué tipo de datos se quiere almacenar y cuál será su uso. Además, definir el acceso que los diferentes departamentos de la empresa tienen a los datos puede ser costoso.
- ▶ Por otro lado, también hay que tener en cuenta el **mantenimiento** y su **actualización**, ya que este añadirá un coste extra tanto en tiempo como en dinero para la empresa.

A la hora de gestionar datos, una empresa dispone de infinidad de recursos y alternativas con las que hacerlo. En la figura 9 se describen las principales alternativas para gestionar los datos de una empresa.

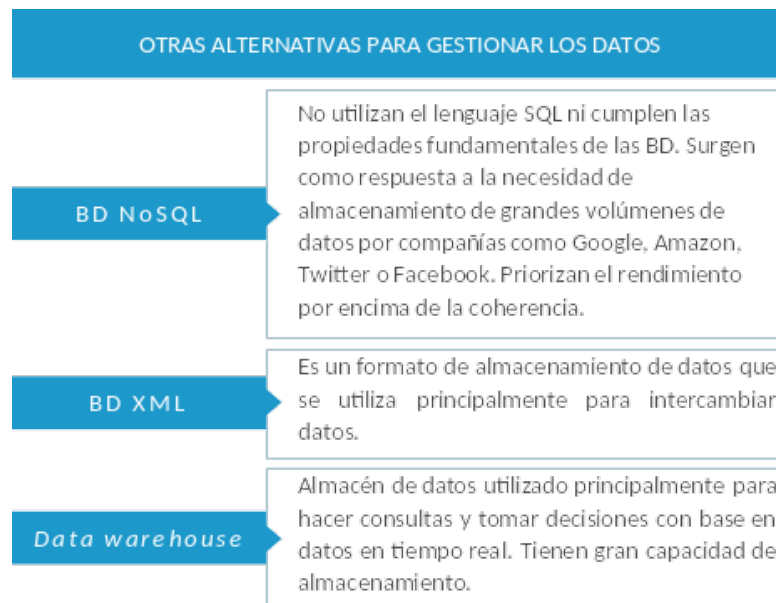


Figura 15. Alternativas para gestión de datos.

De todas las alternativas las más utilizadas son las bases de datos. En la figura 10 indicamos las más utilizadas en la actualidad. De entre los motivos por los que son las elegidas se encuentran los siguientes:

- ▶ Control de la redundancia.
- ▶ Restricción de los accesos no autorizados.
- ▶ Capacidad de realizar acciones usando reglas.
- ▶ Ofrecer diferentes interfaces de usuario.
- ▶ Integridad de datos.
- ▶ Facilidad de realizar copias de seguridad y recuperación.
- ▶ Mayor accesibilidad y capacidad de respuesta.
- ▶ Mayor productividad.



Figura 16. Bases de datos más utilizadas en la actualidad.

Dispositivo de almacenamiento de datos

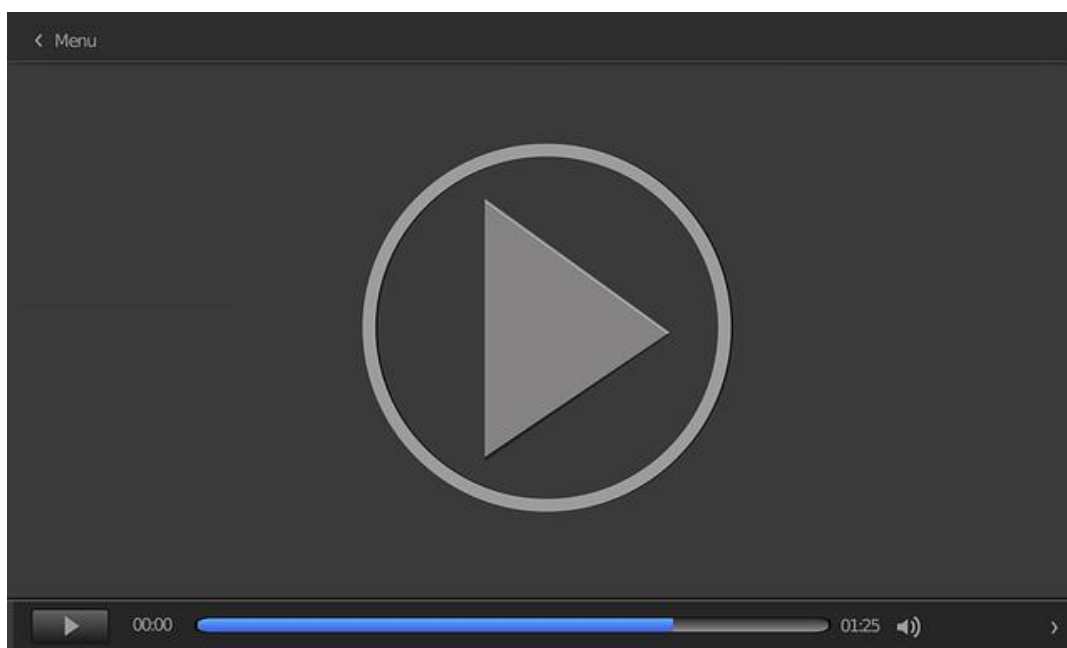
Dispositivo de almacenamiento de datos. (S. f.). En *Wikipedia* (actualizado el 22 de octubre de 2019, consultado el 28 de octubre de 2019). Recuperado de: https://es.wikipedia.org/wiki/Dispositivo_de_almacenamiento_de_datos

Las bases de datos son sistemas de almacenamiento de datos que van más allá de, simplemente, guardar información. En este enlace de Wikipedia encontrarás los elementos que existen y han existido para almacenar datos.

Data warehouse

Auribox Training. (15 de junio de 2017). *¿Qué es un Data Warehouse? | Business Intelligence* [Archivo de vídeo]. Recuperado de <https://youtu.be/jFsRdTcljeU>

Hemos visto los *data warehouse* pero, ¿cómo funcionan? En este enlace puedes ver un vídeo para conocer un poco más sobre ellos.



Accede al vídeo:

<https://www.youtube.com/embed/jFsRdTcljeU>

1. Una base de datos debe:
 - A. Tener relaciones y tablas.
 - B. Almacenar datos.
 - C. Estar dividida.
 - D. Contener números y palabras.

2. Cuando y qué avance permitió el desarrollo de aplicaciones más sencillas:
 - A. Los SGBD relacionales en los años ochenta.
 - B. Las bases de datos distribuidas en la década de los noventa.
 - C. Las bases de datos distribuidas en la década de los ochenta.
 - D. Las bases de datos centralizadas en la década de los setenta.

3. La concurrencia de usuarios permite:
 - A. Que un usuario pueda tener una base de datos.
 - B. Que varios usuarios puedan crear una base de datos de la misma fuente.
 - C. Que se pueda acceder de forma simultánea.
 - D. Que se pueda mantener la estructura de los usuarios.

4. ¿Por qué es importante establecer unos requisitos y reglas que tenga que tener una base de datos?
 - A. Para mantener la consistencia de los datos.
 - B. Para crear el nivel lógico.
 - C. Por seguridad.
 - D. Para evitar la redundancia.

5. Controlamos el problema de redundancia cuando:
- A. Permitimos a varios usuarios acceder de forma simultánea.
 - B. Establecemos protocolos de seguridad mediante encriptación.
 - C. Mantenemos la integridad de los datos.
 - D. Eliminamos información repetida.
6. Una base de datos sirve para:
- A. Tomar decisiones.
 - B. Modelar la realidad.
 - C. Generar riqueza.
 - D. Todas las respuestas anteriores son correctas.
7. El nivel lógico:
- A. Detalla la necesidad de seguridad.
 - B. No muestra detalles de cómo se almacenan los datos.
 - C. Muestra como interaccionarán los usuarios.
 - D. Todas las respuestas anteriores son correctas.
8. Los modelos de bases de datos sirven para:
- A. Almacenar los datos y dotar de un servicio a los usuarios.
 - B. Crear una estructura lógica de la realidad de la organización.
 - C. Crear una copia de la base de datos.
 - D. Ninguna de las respuestas anteriores es correcta.

9. Qué diferencia al modelo relacional del resto:
- A. Las tablas.
 - B. Los registros.
 - C. Las interacciones.
 - D. Todas las respuestas anteriores con incorrectas.
10. Una base de datos puede ser:
- A. Relacional.
 - B. NoSQL.
 - C. XML.
 - D. Todas las respuestas anteriores son correctas.