SISTEMAS GESTORES DE BASES DE DATOS

¿Qué es un SGBD?

El SGBD es un conjunto coordinado de programas, procedimientos, lenguajes, etc., que suministra, tanto a los usuarios informáticos, como no informáticos y al Administrador, los medios necesarios para describir, recuperar y manipular los datos integrados en la BD, asegurando su confidencialidad y seguridad.

Las operaciones típicas que debe realizar un SGBD puede resumirse en aquellas que afectan a la totalidad de los datos y las que tienen lugar sobre registros concretos.

A) Sobre el conjunto de la base

- Creación
- Reestructuración
- Consulta a la totalidad

B) Sobre registros concretos

- Inserción
- Borrado
- Modificación

} Actualización

- Consulta selectiva

Objetivos de un SGBD

A. Independencia física.

Conseguir la independencia entre las estructuras de almacenamiento (representación física de los datos) y las estructuras lógicas de los datos (datos del mundo real).

La independencia física es la capacidad de modificar el esquema físico, tanto en la estructura del archivo como en las características de los campos, sin obligar a que se vuelvan a escribir los programas de aplicaciones.

Los cambios en la estructura física que se podrán realizar sin afectar, para nada, a la estructura lógica de los datos podrán ser: cambios en el tamaño de los bloques, longitud de los registros almacenados, cambiar los registros de longitud fija a variable, la posición relativa de los registros, la organización física, métodos de direccionamiento, índices, ubicación de los conjuntos de datos en diferentes volúmenes, etc.

Objetivos de un SGBD

B. Independencia lógica.

Que los cambios de las estructuras lógicas de datos no afecten a los programas que los utilizan.

La independencia lógica es la capacidad de variar el esquema conceptual sin necesidad de que se vuelvan a escribir de nuevo los programas de aplicaciones. Las modificaciones en el nivel conceptual son necesarias cuando se cambia la estructura lógica de la B.D.

Los cambios en aspectos lógicos podrán ser:

- En los campos. Cambios en el nombre, tipo, modo de cálculo (datos derivados), contraseñas, etc.
- En los registros. Pueden haber cambios en los nombres, introducción de nuevos campos, borrado de los mismos, alteración del orden en que aparecen los campos, etc.

Objetivos de un SGBD

C. Manipulación de los datos por los no informáticos.

Podrán manipular los datos por medio de lenguajes no sujetos a procedimientos, es decir, describiendo los datos que deban consultar o actualizar sin describir la forma en que hay que consultarlos o actualizarlos.

D. Eficacia de los accesos a los datos.

Que el SGBD proporcione un método de acceso lo más eficiente posible.

E. Administración coherente de datos.

Van a permitir un control eficaz de los datos, optimizar los accesos a los datos y la utilización de los medios informáticos y resolver los conflictos resultantes. Estas funciones se realizarán por un grupo de personas muy expertas (administrador de la B.D).

Objetivos de un SGBD

F. Redundancia mínima de los datos.

En los sistemas clásicos de ficheros no integrados, cada aplicación ejecuta sus archivos particulares lo que a menudo origina redundancia en los datos almacenados, con el consecuente aumento de los costes de almacenamiento, y un importante desperdicio de recursos para actualizar los mismos datos varias veces.

En las B.D los ficheros se comparten por las distintas aplicaciones. La administración debe velar por la no duplicación física de los datos.

Pero a pesar de todo, no es conveniente eliminar toda la redundancia, sino mantener a los diferentes archivos ligados entre sí por medio de un campo común. La redundancia debe controlarse para facilitar el acceso a los datos de forma eficiente y rápida.

Objetivos de un SGBD

G. Coherencia de los datos.

Las operaciones de actualización insertar, borrar y modificar registros se realizan con mucha frecuencia. Es necesario que el SGBD evite la inconsistencia de los datos, reduciendo al mínimo la redundancia y mantenimiento en todo momento la integridad de los datos.

H. Compartición de datos.

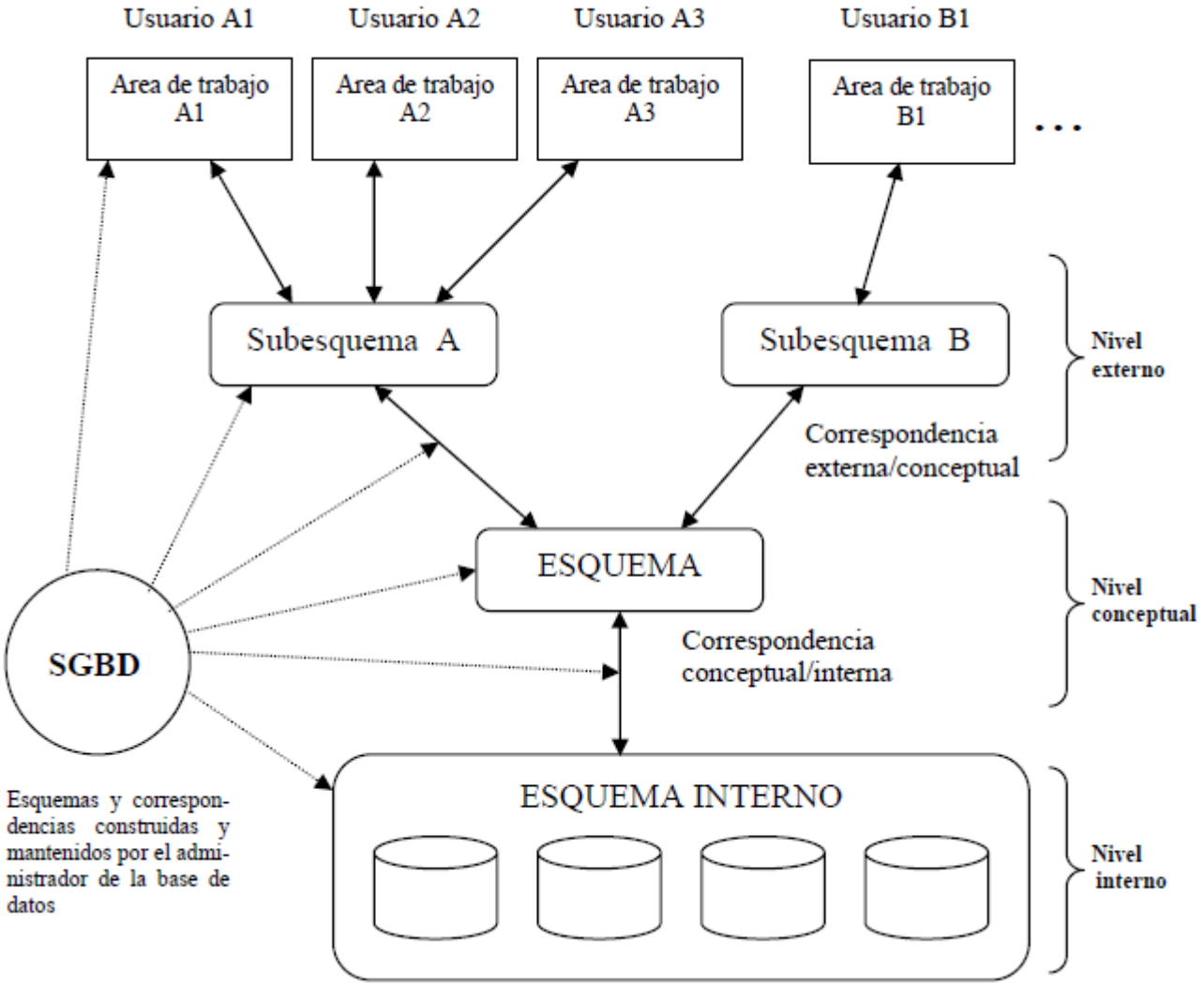
Permitir que distintas aplicaciones puedan compartir los datos simultáneamente.

I. Seguridad de los datos.

Debe existir mecanismos adecuados para controlar o retirar las autorizaciones de acceso a cualquier usuario para cualquier conjunto de datos.

Arquitectura de una B.D

(Modelo ANSI/X3/SPARC)



Arquitectura de una B.D

(Modelo ANSI/X3/SPARC)

El grupo de trabajo SPARC de la sección X3 del organismo de estándares ANSI, diseño un modelo en el que indicaba cómo debía funcionar un SGBD para asegurar la separación entre datos y aplicaciones. Este organismo definió tres niveles:

- **Nivel externo**. Es el más cercano a los usuarios. Representa la visión de cada usuario o programador de la base de datos. En el nivel externo se describe de forma individual un subconjunto de datos para un usuario o conjunto de usuarios. Los usuarios pueden trabajar con los archivos externos como si existiesen físicamente, aunque éstos en realidad no existen tal como son vistos por el usuario. Por cada tipo de usuario es necesario especificar un esquema externo, subesquema o vista externa, que describe un subconjunto de datos de la base de datos. Habrá usuarios que podrán acceder a más de un esquema externo, y un esquema externo podrá ser compartido por varios usuarios.
- **Nivel interno/físico**. Se refiere a la forma en la que realmente se almacena la información de la base de datos. El nivel interno se describe por medio de un esquema interno o vista interna. El administrador es el único que trabaja a nivel interno.
- **Nivel conceptual**. Es una representación de toda la información contenida en la BD en una forma más abstracta que la contenida en la estructura interna. El nivel conceptual se describe por medio de un esquema o vista conceptual de la BD.

Arquitectura de una B.D

(Modelo ANSI/X3/SPARC)

El SGBD debe poder garantizar la transferencia de los datos desde el nivel interno al nivel externo, a este proceso se llama transformación de datos o mapeo (data mapping).

Para ello existen dos niveles de correspondencia:

- **Correspondencia conceptual/interna**: Permite el paso de la vista conceptual a la vista interna, y viceversa. Especifica cómo se representan los registros y campos conceptuales en el nivel interno. Si se modifica la estructura interna de la base de datos, la correspondencia conceptual/interna deberá modificarse, para que no varíe el esquema conceptual. De este modo se conserva la independencia de los datos.
- **Correspondencia externa/conceptual**: Permite el paso de una vista externa específica a la vista conceptual, y viceversa.

Los subesquemas, el esquema conceptual, el esquema interno y las correspondencias conceptual/interna y externa/conceptual, las describe el administrador de la base de datos y quedan almacenados dentro del diccionario de la BD para futuras consultas del SGBD.

Cuando un usuario desea acceder a la base de datos, el SGBD examina el diccionario de datos para comprobar si la solicitud puede ser realizada y el modo de realizar las transformaciones pertinentes de los datos.

Otros niveles de abstracción

Hoy en día se definen más niveles de trabajo con las bases de datos. Estos niveles son (empezando desde el más cercano al usuario):

- **Nivel externo**. Sigue representando la vista que poseen los distintos usuarios de la base de datos. En realidad los esquemas de este nivel son los últimos que se crean y lo hacen los **desarrolladores** o programadores.
- **Nivel conceptual**. Actualmente se considera así al nivel que representa los primeros esquemas de la base de datos, que son aquellos que diseñan los **analistas** o **diseñadores**. Ejemplo de modelo que opera a este nivel es el **modelo Entidad/Relación**.
- **Nivel lógico**. Se acerca más a la física de la base de datos. En este nivel se hace referencia a estructuras de organización de información que varían según el tipo de SGBD que se utilice. Este nivel sigue siendo manejado por los analistas. En muchos casos (aunque ciertamente es peligroso) los diseñadores/as de la base de datos empiezan por este nivel saltándose el anterior. En la actualidad el **modelo relacional** sigue siendo el modelo más habitual para crear esquemas a nivel lógico.
- **Nivel interno**. Es el primero en el proceso de modelado de la base de datos que se realiza sobre el software gestor de la base de datos (teniendo en cuenta que lo externo, las aplicaciones, se crean más tarde). Usa el lenguaje de la base de datos para crear las estructuras de datos definidas en el nivel lógico. Este nivel lo maneja el a**dministrador de la base de datos** (o **DBA**).
- **Nivel físico**. Se refiere a como se organizarán los datos en el disco, en qué ordenadores se crea la base de datos, si es distribuida o no, sistema operativo necesario, estructura de directorios y archivos, configuración de servidores y sistema operativo, política de copia de seguridad, etc. Este nivel lo maneja el a**dministrador de la base de datos** (o **DBA**).

Componentes de un SGBD

Los SGBD se componen de:

- Lenguajes.
- El diccionario de datos.
- Mecanismos de seguridad e integridad.
- Factor humano.

Componentes de un SGBD

Los lenguajes que tenga un SGBD deben permitir:

- Crear la estructura de la base de datos, incluyendo todos los objetos que puede incluir la misma (tablas, vistas, usuarios, procedimientos, funciones, triggers, etc.). Ej: DDL
- Consultar y manipular la información almacenada en la base de datos. Ej.: DML
- Asignar privilegios a usuarios, confirmar o abortar transacciones, etc. Ej.: DCL.
- En algunos casos, también incluyen un lenguaje de cuarta generación (4GL) para RAD (desarrollo rápido de aplicaciones). Ej: Asistentes de Access, Oracle Developer Suite.

Componentes de un SGBD

El diccionario de datos contiene los metadatos (datos acerca de los datos) de la base de datos, esto es:

- La definición de todos los objetos existentes en la base de datos: tablas con sus columnas, vistas, procedimientos, triggers, índices, etc...
- La ubicación física de los objetos y el espacio asignado a los mismos.
- Los privilegios y roles asignados a los usuarios.
- Las restricciones de las tablas.
- Información de auditoría.
- Estadísticas de uso de la base de datos.
- Información del consumo de recursos actual.
- Etc.

Componentes de un SGBD

Un SGBD debe proporcionar utilidades que permitan:

- La realización de copias de seguridad de los datos y la restauración de las mismas.
- Garantizar la protección de los datos ante accesos no autorizados.
- Implantar restricciones de integridad de los datos para evitar daños accidentales de los datos.
- Recuperar la base de datos hasta un estado consistente en caso de error del sistema o cualquier otro imprevisto.
- Controlar el acceso concurrente de los usuarios para evitar errores de integridad.

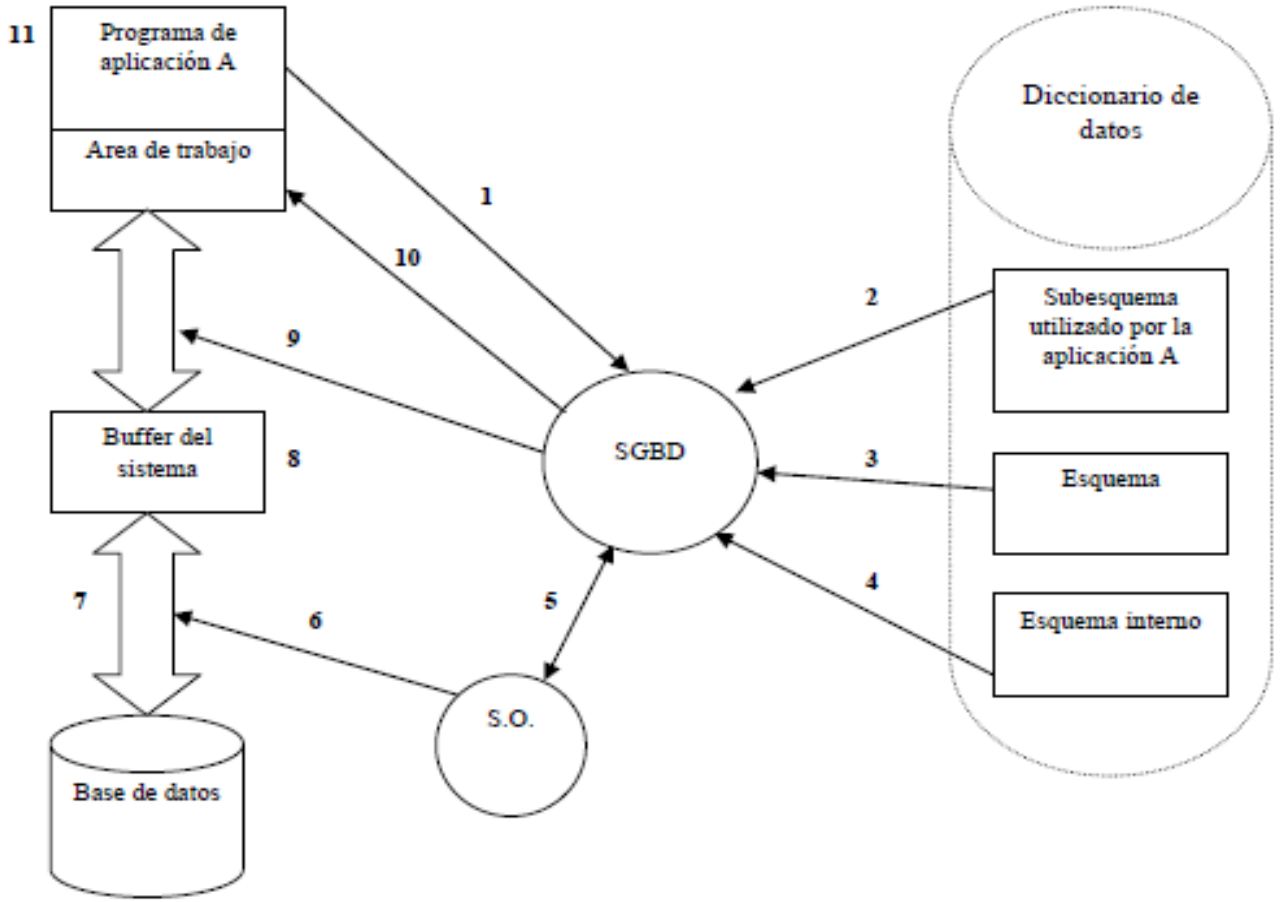
Componentes de un SGBD

Un SGBD siempre va a tener distintas categorías de usuarios:

- Usuarios finales: Podrán acceder a la información sobre la que le hayan sido concedidos privilegios.
- Programadores: Realizan aplicaciones sobre los objetos de la base de datos para facilitar su trabajo a los usuarios finales.
- Administradores o DBAs: Garantizan el correcto funcionamiento de la base de datos y gestionan todos sus recursos. Tienen el nivel más alto de privilegios y responsabilidades legales en caso de que los datos tengan algún tipo de protección. Su objetivo es que la base de datos está siempre disponible y con un rendimiento óptimo.

Funcionamiento de un SGBD

Supongamos el caso de un programa de aplicación, que lee un registro de una base de datos, para ello realiza una solicitud al SGBD, esencialmente se sigue la siguiente secuencia:



Funcionamiento de un SGBD

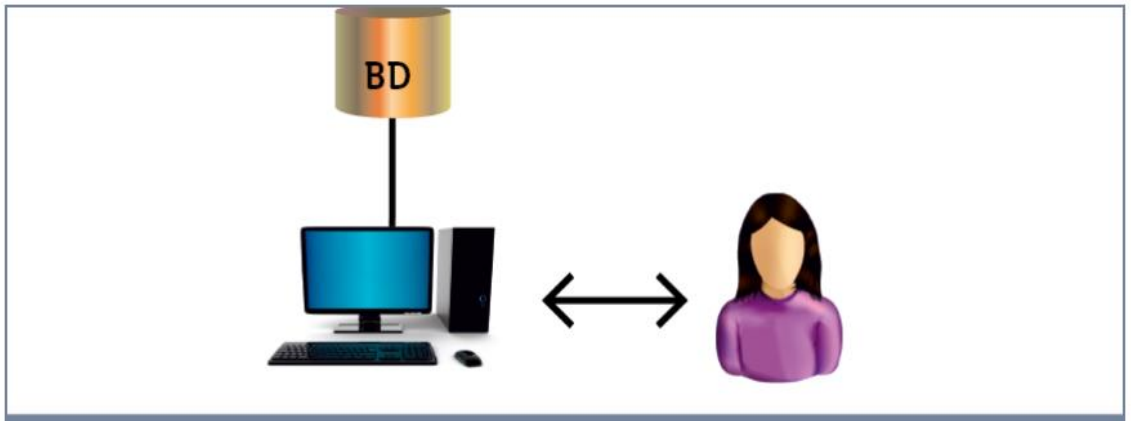
1. El programa de aplicación A pide al SGBD que lea un registro. Se proporciona la clave de acceso. El SGBD evalúa la capacidad del usuario para realizar la operación solicitada.
2. El SGBD obtiene el subesquema que utiliza el programa de aplicación A. Si no encuentra los datos solicitados, rechaza la solicitud.
3. El SGBD obtiene el esquema y determina qué tipo de datos lógicos necesita.
4. El SGBD examina el esquema interno y determina qué registro físico debe leer.
5. El SGBD ordena al S.O. que lea el registro pedido.
6. El S.O. interacciona con el dispositivo físico en el que se encuentran los datos.
7. Los datos pedidos se envían desde el dispositivo físico al buffer del sistema.
8. El SGBD analiza el esquema, el subesquema y las correspondencias externa/conceptual para realizar las transformaciones necesarias de los datos.
9. El SGBD transfiere los datos al área de trabajo del programa de aplicación A.
10. El SGBD informa al programa del éxito o fracaso de la operación de E/S, incluyendo cualquier indicación de error.
11. El programa puede ahora trabajar con los datos pedidos.

Tareas del DBA

- Decidir el SGBD idóneo, instalarlo y configurarlo inicialmente.
- Supervisar diseño lógico de la BD
- Realizar diseño físico de la BD: Estructura de almacenamiento
- Crear y mantener el esquema de la BD
- Crear y mantener cuentas de usuario
- Colaborar en la formación de usuarios y programadores
- Detectar y resolver problemas de rendimiento de la BD usando herramientas de monitorización
- Realizar copias de seguridad, migraciones, importaciones y exportaciones, auditorias de seguridad, etc...
- Recuperar instancias dañadas.

Modelos de explotación de B.D: monopuesto

- La base de datos se encuentra en una máquina y es explotada desde la misma máquina.
- Se trata de Sistemas Gestores instalados en una máquina desde la que se conectan los propios usuarios y administradores. Es decir, todo el sistema está en una sola máquina.
- Es un modelo que sólo se utiliza con bases de datos pequeñas y poca cantidad de conexiones. Es típico en SGBD de escritorio: Microsoft Access, LibreOffice Base.

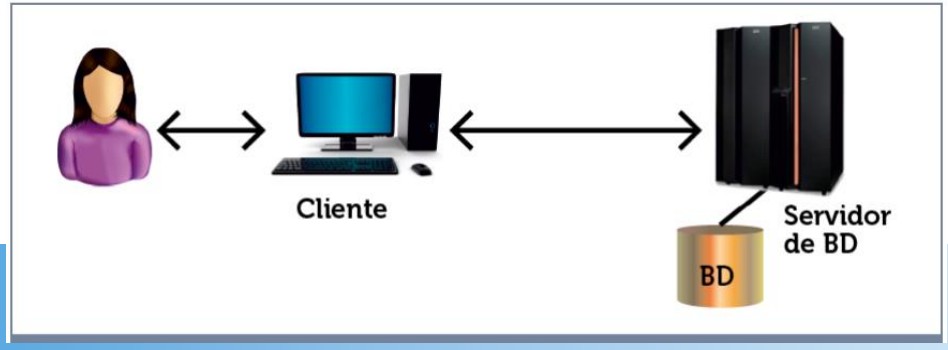


Modelos de explotación de B.D: cliente/servidor

La base de datos y el sistema gestor se alojan en un servidor, mientras que los clientes acceden desde máquinas distintas a través de la red (sea local o de área extensa).

Hay dos posibilidades:

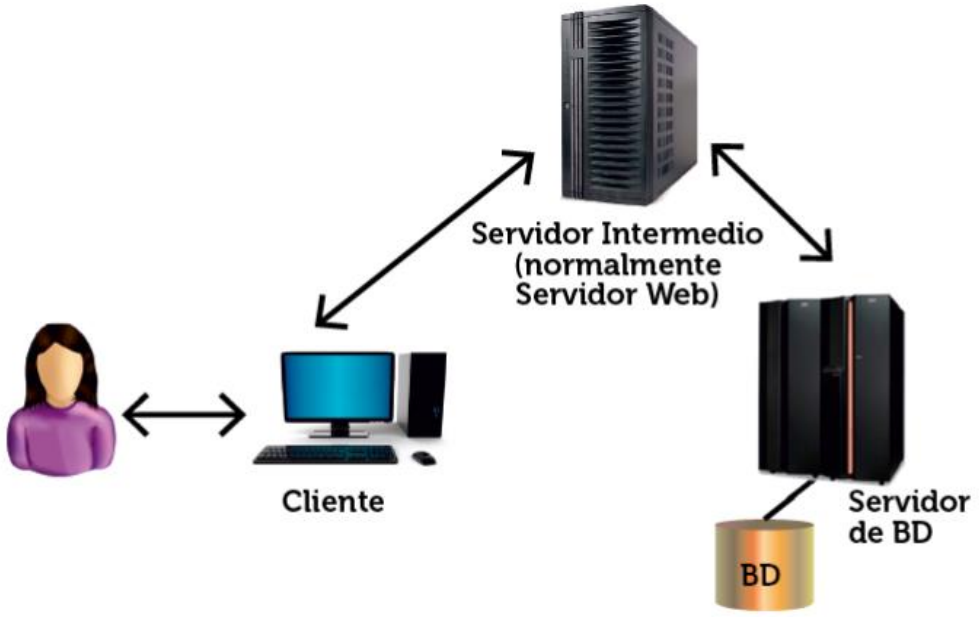
- Arquitectura cliente/servidor único. Un solo servidor gestiona la base de datos, todos los clientes se conectan a él para realizar las peticiones a la base de datos.
- Arquitectura cliente/multiservidor (grid de servidores). La base de datos se distribuye entre varios servidores. El cliente no sabe realmente a qué servidor se conecta; el software de control de comunicaciones se encargará de dirigir al usuario al servidor adecuado. De forma lógica, es como si se tratara de un solo servidor aunque físicamente sean muchos (el cliente no percibe que haya más de un servidor).



Modelos de explotación de B.D: 3 o más capas

Arquitectura de 3 o más capas

En este caso entre el cliente y el servidor hay al menos una capa intermedia (puede haber varias). Esa capa (o capas) se encarga de recoger las peticiones de los clientes y luego de comunicarse con el servidor (o servidores) de bases de datos para recibir la respuesta y enviarla al cliente.



Algunos SGBD comerciales

Oracle

Oracle es un sistema de gestión de base de datos desarrollado por la compañía Oracle, este sistema es de tipo objeto-relacional (ORDBMS - Object-Relational Data Base Management System), por el cual es uno de los gestores de bases de datos mas completo como: soporte de transacciones, estabilidad, escalabilidad y puede correr en los sistemas operativos GNU/LINUX, Windows, Mac y entre otros.

Microsoft SQL/Server

Es un sistema de gestión de base de datos relacional, desarrollado por la empresa Microsoft. El lenguaje de desarrollo utilizado (por línea de comandos o mediante la interfaz gráfica de Management Studio) es Transact-SQL (TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos (DML), crear tablas y definir relaciones entre ellas (DDL).

Es el claro competidor de Oracle, junto con éste ocupan la mayor cuota de mercado.

Algunos SGBD libres

- **MySQL**, es el sistema gestor de bases de datos relacional por excelencia. Es un SGBD multihilo y multiusuario utilizado en la gran parte de las páginas web actuales. Además es el más usado en aplicaciones creadas como software libre.
- **MariaDB,** es una derivación de MySQL que cuenta con la mayoría de características de éste e incluye varias extensiones.
- **PostGreSQL,** Este sistema gestor de base de datos relacional está orientado a objetos. La principal desventaja es la lentitud para la administración de bases de datos pequeñas ya que está optimizado para gestionar grandes volúmenes de datos.

Algunos SGBD NoSQL

- **MongoDB,** es un sistema de base de datos NoSQL, orientado a documentos y de código abierto.

  En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON (una especificación similar a JSON), haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

- **Redis**, es un motor de base de datos en memoria, basado en el almacenamiento en tablas de hashes (clave/valor) pero que opcionalmente puede ser usada como una base de datos durable o persistente. Está escrito en C.

- **Apache Cassandra**, es un sistema de base de datos NoSQL distribuido y basado en un modelo de almacenamiento de «clave-valor». De código abierto y escrita en Java. Permite grandes volúmenes de datos en forma distribuida. Por ejemplo, lo usa Twitter para su plataforma.