

TEMARIO

- 1.1 Introducción
- 1.2 Sistemas de Información
- 1.3 Los Sistemas de Gestión de Bases de Datos
- 1.4 Los niveles de abstracción de una Base de Datos
- 1.5 Componentes de los SGBD
- 1.6 Funcionamientos del SGBD
- 1.7 Arquitectura de los SGBD

INTRODUCCIÓN A LOS SISTEMAS DE BASES DE DATOS

Tal como dice el título de este capítulo, la idea fundamental es mostrar y explicarte los elementos básicos de un Sistema de Bases de Datos y cómo estos se relacionan entre sí.

En primer lugar, explicaremos los conceptos de Bases de Datos y Sistemas de Gestión de Bases de Datos (SGBD), para luego dar lugar a la arquitectura, niveles de abstracción y funcionamiento de los SGBD. A continuación describiremos los tipos de SGBD y sus características principales.

Con todo lo anterior, creemos que podrás responder preguntas del tipo ¿Qué es una Base de Datos? ¿Qué es un SGBD?, etc.

1.1 Introducción

El mundo actual tiene cada vez más una mayor demanda de datos para generar información relevante. Esta demanda siempre ha sido patente en empresas y sociedades, pero durante los últimos años, ésta ha aumentado exponencialmente debido al acceso multitudinario a Internet.

El propio nombre Informática hace referencia al hecho de ser una ciencia que trabaja con información. Desde los inicios de la creación de computadores, 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. Desde el inicio de la informática se ha reconocido al dato como al elemento fundamental de trabajo en un computador. Por ello se han realizado numerosos estudios y aplicaciones para mejorar la gestión que desde las computadoras se realiza de los datos.

Inicialmente los datos que se necesitaban almacenar y gestionar eran pocos, pero éstos han ido creciendo. En la actualidad las aplicaciones de Internet han producido grandes sistemas de información que incluso para poder gestionarlas requieren decenas de máquinas haciendo la información accesible desde cualquier parte del planeta y en un tiempo muy rápido. Eso ha requerido que la ciencia de las bases de datos esté en continua renovación para hacer frente a esas enormes necesidades.

Antes de la aparición del computador, el tiempo requerido para manipular datos era enorme. Sin embargo el proceso de aprendizaje era relativamente sencillo ya que se usaban elementos que el usuario reconocía perfectamente. Por esa razón, la informática adaptó sus herramientas para que los elementos que el usuario maneja en el computador se parezcan a los que utilizaba manualmente.

Es así como nace el concepto de Bases de Datos como herramienta informática.

Se considera Bases de Datos como una colección o depósito de datos integrados, almacenados en soporte secundario (no volátil) y con redundancia controlada.

Los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de ellos, y su definición (estructura de la base de datos) única y almacenada junto con los datos, se ha de apoyar en un modelo de datos, el cual ha de permitir captar las interrelaciones y las restricciones existentes en el mundo real. Los procedimientos de actualización y recuperación, comunes y bien determinados, facilitarán la seguridad del conjunto de los datos. (Piattini, 1999)

1.2 Sistemas de Información

1.2.1 La Empresa como sistema

Según la RAE, la definición de sistema es “Conjunto de cosas que ordenadamente relacionadas entre sí contribuyen a un determinado objeto”.

El cliente fundamental del profesional de la informática es la empresa. La empresa se puede entender como un sistema formado por diversos objetos: el capital, los recursos humanos, los inmuebles, los servicios que presta, etc.

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

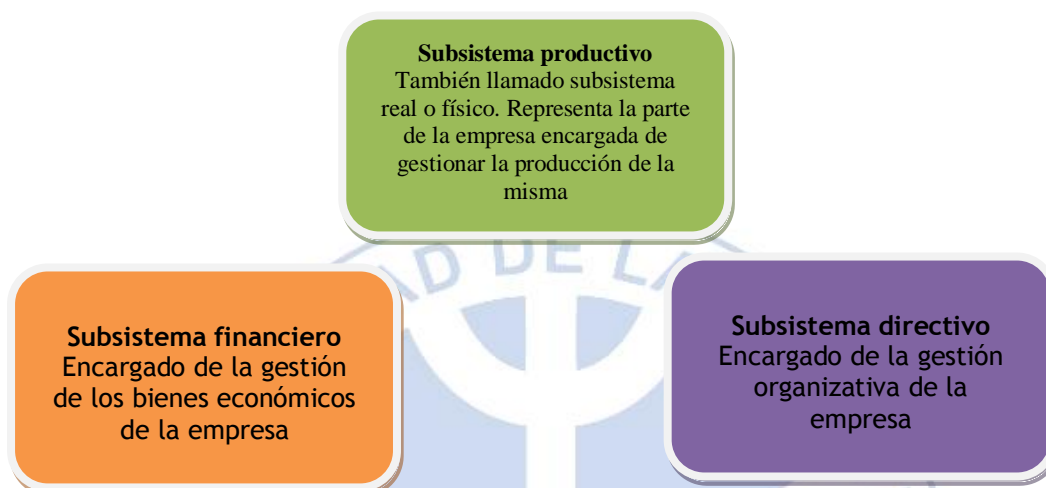


Figura nº1: Subsistemas que conforman el Sistema de la Empresa

Hay que hacer notar que cada subsistema se asocia a un departamento concreto de la empresa.

1.2.2 Los Sistemas de Información

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

Realmente un SI sólo incluye la información que nos interesa de la empresa y los elementos necesarios para gestionar esa información.

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

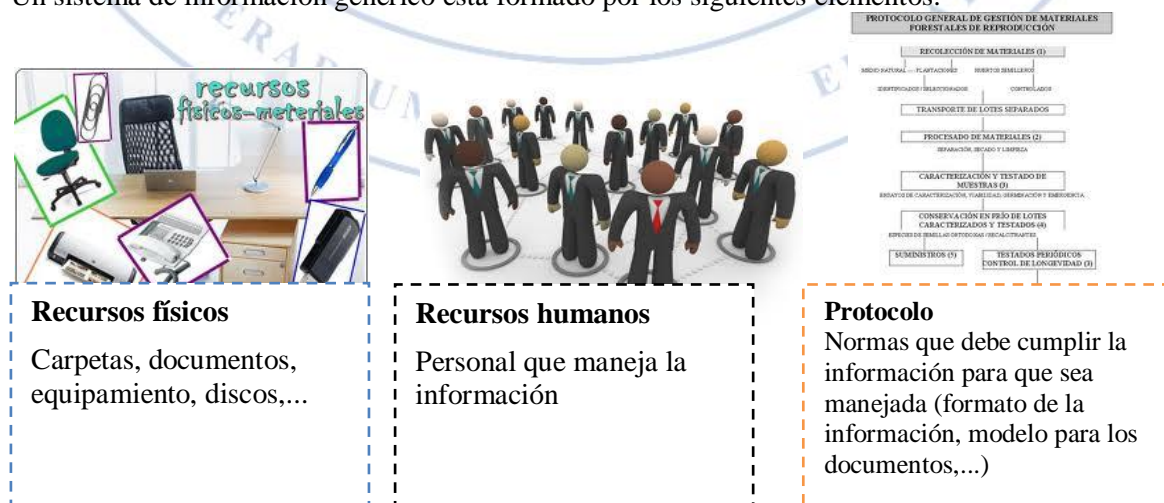


Figura nº2: Elementos de un Sistema de Información Genérico

Las empresas necesitan implantar estos SI debido a la competencia que las obliga a gestionar de la forma más eficiente sus datos para una mayor calidad en la organización de las actividades de los subsistemas empresariales.

En el caso de una gestión electrónica de la información, o un SI 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.

1.2.3 Tipos de Sistemas de Información

En la evolución de los SI ha habido dos puntos determinantes, que han formado los dos tipos fundamentales de SI.

1. Sistemas de Información orientados al proceso

En estos 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 computador (a veces en archivos binarios, o en hojas de cálculo, o incluso en archivos de texto).

Cada programa almacena y utiliza sus propios datos de forma un tanto caótica. 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. Pero tiene grandes inconvenientes:

- **Datos redundantes.** Ya que se repiten continuamente
- **Datos inconsistentes.** 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.
- **Costo de almacenamiento elevado.** Al almacenarse varias veces el mismo dato, se requiere más espacio en los discos. Luego se agotarán antes.
- **Difícil acceso a los datos.** Cada vez que se requiera una consulta no prevista inicialmente, hay que modificar el código de las aplicaciones o incluso crear una nueva aplicación.
- **Dependencia de los datos a nivel físico.** Para poder saber cómo se almacenan los datos, es decir qué estructura se utiliza de los mismos, necesitamos ver el código de la aplicación; es decir el código y los datos no son independientes.
- **Tiempos de procesamiento elevados.** Al no poder optimizar el espacio de almacenamiento.
- **Dificultad para el acceso simultáneo a los datos.** Es casi imposible de conseguir ya que se utilizan archivos que no admiten esta posibilidad. Dos usuarios no pueden acceder a los datos de forma concurrente. Si dos o más usuarios acceden concurrentemente a los datos pueden obtener información errónea.

- **Dificultad para administrar la seguridad del sistema.** Ya que cada aplicación se crea independientemente; es por tanto muy difícil establecer criterios de seguridad uniformes.

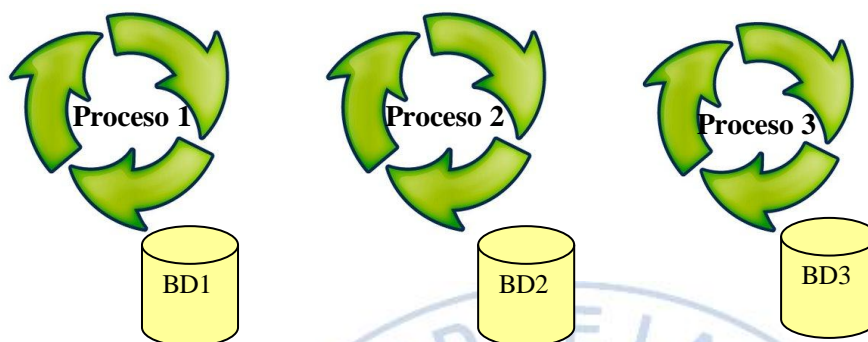


Figura n°3: Sistemas de Información orientados al proceso

A estos sistemas se les llama sistemas de gestión de archivos. 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.

2. Sistemas de Información orientados a los datos, Bases de datos

En este tipo de sistemas los datos se centralizan en una base de datos común a todas las aplicaciones. Estos serán los sistemas que estudiaremos en este curso.

En esos sistemas los datos se almacenan en una única estructura lógica que es utilizable por las aplicaciones. A través de esa estructura se accede a los datos que son comunes a todas las aplicaciones.

Cuando una aplicación modifica un dato, dicha modificación será visible para el resto de aplicaciones.

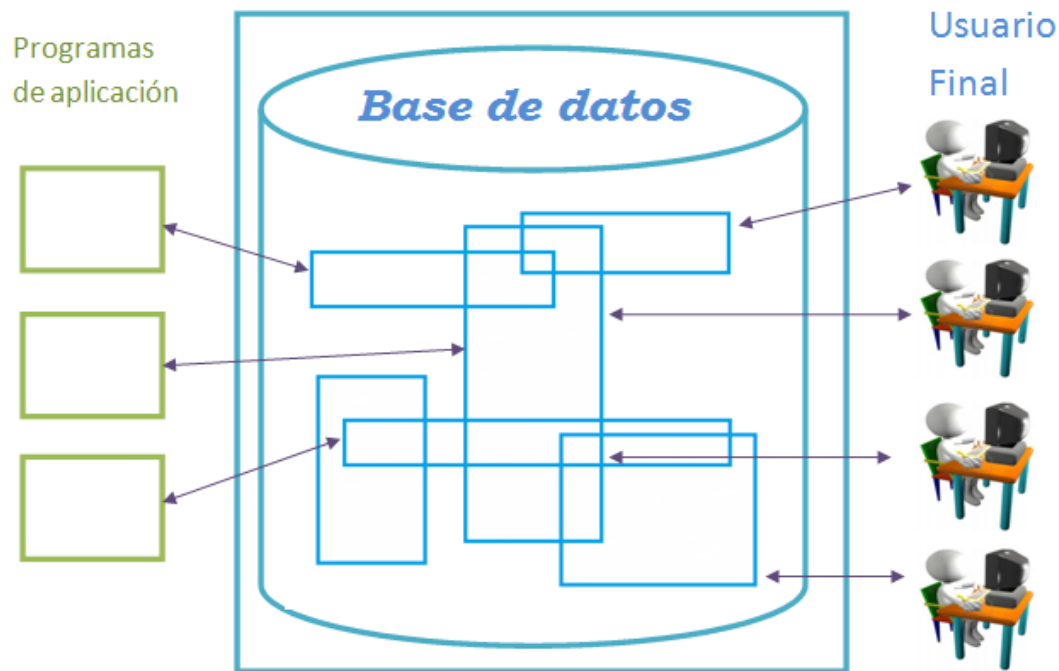


Figura nº4: Sistemas de Información orientados a datos

Ejemplo

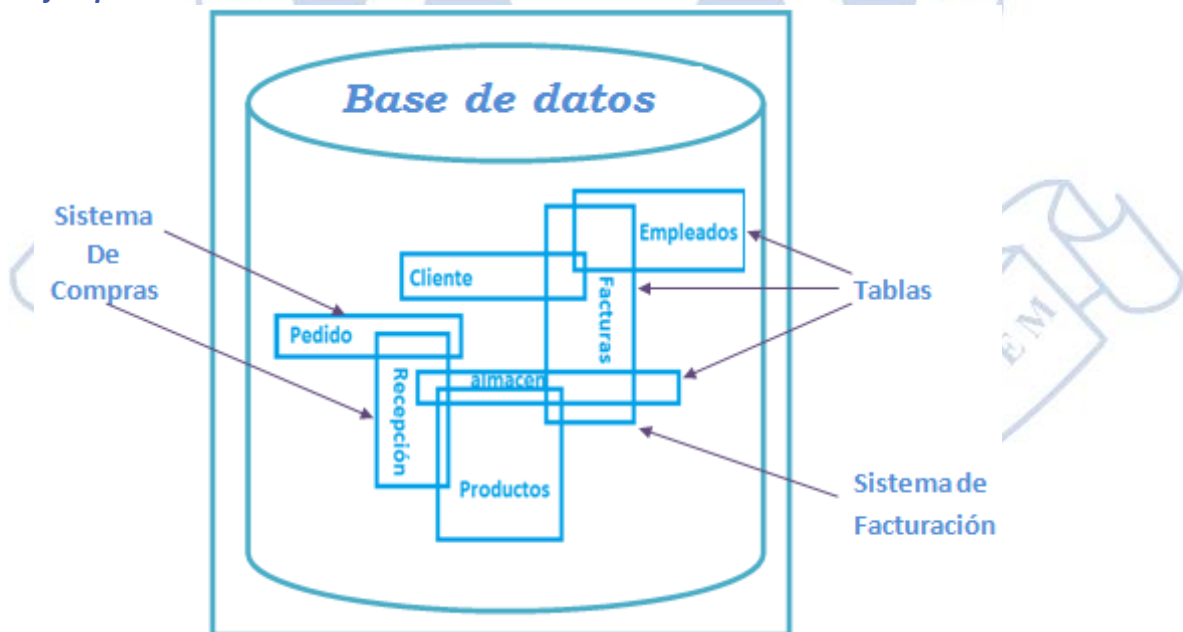


Figura nº5: Ejemplo de Sistemas de Información orientados a datos

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 los datos para los cuales está autorizado.
- **Datos más documentados.** Gracias a los metadatos que permiten describir la información contenida en 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 y más seguro controlar el acceso de usuarios de forma concurrente, dado que el sistema proporciona mecanismos que secuencializan las operaciones sobre la base de datos.

Desventajas

- **Instalación costosa.** El control y administración de bases de datos requiere de un software y hardware poderoso, que puede resultar de costo elevado.
- **Requiere personal cualificado.** Debido a la dificultad de manejo de este tipo de sistemas y de su administración.
- **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.

Resumen

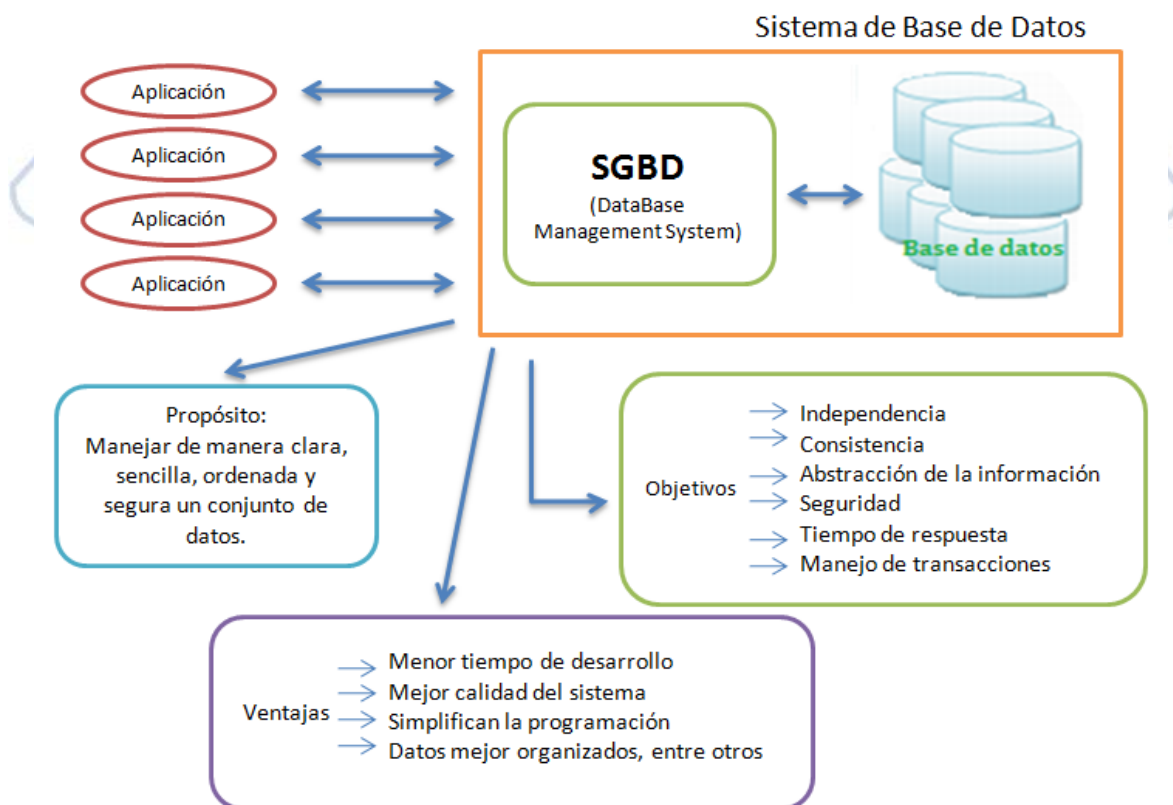


Figura nº6: Sistemas de Bases de Datos

1.3. Los Sistemas de Gestión de Bases de Datos

Una Base de Datos se define como un “conjunto de datos relacionados entre sí”

Los conceptos relevantes en esta definición son “datos” y “relacionados”.

“Datos”

- Conjunto de hechos relevantes que pueden ser registrados de algún modo, y que cuentan con un significado implícito.
- Reflejan situaciones del mundo real y cambios en esas situaciones.

Por ejemplo, los datos del empleado que son significativos para una empresa pueden ser: Rut, nombre, dirección, ciudad, fecha de nacimiento, título profesional, sueldo base, Fondo de pensiones, Salud, etc.

Así entendemos que estos datos pueden cambiar, por ejemplo: el empleado puede cambiar de dirección, o puede cambiar de sistema de salud, etc.

Otro ejemplo son los datos de los proyectos, como: código de identificación, nombre, fecha de inicio, fecha final, empleado que lo dirige, etc.

“Relacionados”:

- Debe existir homogeneidad en la colección de datos que conforma una BD.
- No se trata de un conjunto seleccionado de forma aleatoria.
- Los datos se recopilan y registran con una finalidad.
- Los datos deben ser relevantes con respecto a esa finalidad.

Por ejemplo, los datos del empleado son relevantes para una organización ya que necesita conocer el título profesional para asignarle el sueldo base, o el fondo de pensiones que contrató para calcular el porcentaje de descuento mensual, o conocer la fecha de nacimiento para enviarle un saludo de cumpleaños, etc. Además el empleado estará relacionado con otros objetos dentro de la organización, ya que éste trabaja en un cierto departamento, y tiene asignados algunos proyectos.

Otro concepto relevante es el Sistema de Gestión de Bases de Datos o SGBD (aunque se suele utilizar más a menudo las siglas DBMS procedentes del inglés, Data Base Management System) es el software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos.

Sistema de Gestión de Bases de Datos:

Conjunto de programas de propósito general, que proporcionan funcionalidades *horizontales* para facilitar la gestión de la información contenida en una base de datos.

Los SGBD proporcionan un conjunto coordinado de programas, procedimientos y lenguajes que permiten a los distintos usuarios realizar acciones sobre los datos. Las acciones que puedan llevar a cabo dependerán del tipo de usuario que se le asigne, por ejemplo, un usuario puede leer datos y no modificarlos, otro usuario puede leer pero también insertar nuevos registros, incluso puede eliminar datos; otro usuario puede manipular todos los datos pero además crear nuevas tablas, modificarlas y eliminarlas. Este último usuario es el Administrador de la Base de Datos, conocido como DBA.

Las formas de acceder a una base de datos son variadas. Una de las formas es por medio de los programas de aplicación. Por ejemplo, un alumno puede acceder a las calificaciones de una asignatura por medio de la Intranet de la Universidad. En este caso, la aplicación es la Intranet, y las calificaciones están almacenadas en una base de datos. ¿Pero para que está el SGBD?. Observamos en la figura nº7 que está justo en medio de los dos, de las aplicaciones y de la base de datos, ya que es el software que recibe la solicitud de la aplicación (solicita las calificaciones para un determinado alumno), obtiene los datos necesarios desde la base de datos y finalmente los envía a la aplicación para que los muestre al alumno.

Otra forma es por medio de un usuario que tiene permiso para acceder a la base de datos de manera directa. Por ejemplo, el DBA utiliza un nombre de usuario y contraseña para acceder a un motor de bases de datos y manipular desde allí todos los datos y bases de datos de la organización. Por lo general utiliza software especializados para administrar la base de datos. Por ejemplo, en el caso de poseer Mysql como motor de bases de datos, el DBA puede utilizar Mysql Front como SGBD con el fin de administrar la base de datos de una manera más simple y visual (ver figura nº7 como referencia).



Figura nº7: Integración de los Sistemas de Bases de Datos

El éxito del SGBD reside en mantener la seguridad e integridad de los datos. Lógicamente tiene que proporcionar herramientas a los distintos usuarios. Entre las herramientas que proporciona están:

- **Herramientas para la creación y especificación de los datos.** Así como la estructura de la base 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, eliminar, actualizar o consultar datos.
- **Herramientas de recuperación** en caso de desastre
- **Herramientas para la creación de copias de seguridad**
- **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**

La seguridad se refiere a elementos que se deben agregar para evitar accesos no deseados a los datos y que éstos puedan ser modificados o eliminados. El nivel más básico de seguridad viene

dado por la definición de cuentas de usuarios y sus permisos de acceso a los datos. En un nivel superior, tenemos el uso de firewall, réplica de la base de datos, encriptación de datos, entre otros.

Por otro lado, la integridad de los datos se refiere a que deben almacenarse de manera coherente. Por ejemplo, no podríamos agregar los datos de un producto a la base de datos si no se conoce la marca, o no podríamos agregar los datos de un empleado si no está vinculado a un departamento de la organización, o no podemos agregar un proyecto si no tiene asignado un empleado responsable de él.

1.4 Niveles de Abstracción de una BD

1.4.1 Introducción

En cualquier SI 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.

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 todo SI digital, los usuarios ven los datos desde las aplicaciones creadas por los programadores. A ese nivel se manejan formularios, informes en pantalla o en papel,...

Pero la realidad física de esos datos, tal cual se almacenan en los discos queda oculta a los usuarios. Esa forma de ver la base de datos está reservada a los administradores. Es el nivel físico el que permite ver la base de datos en función de cómo realmente se están almacenando en el computador, en qué carpeta, qué archivos se usan,...

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 (eléctricos, de albañilería, de tuberías de agua,...). Con lo cual una base de datos requiere diseñar al menos tres esquemas (en realidad son más). La figura nº8 presenta estos tres niveles de manera ordenada.

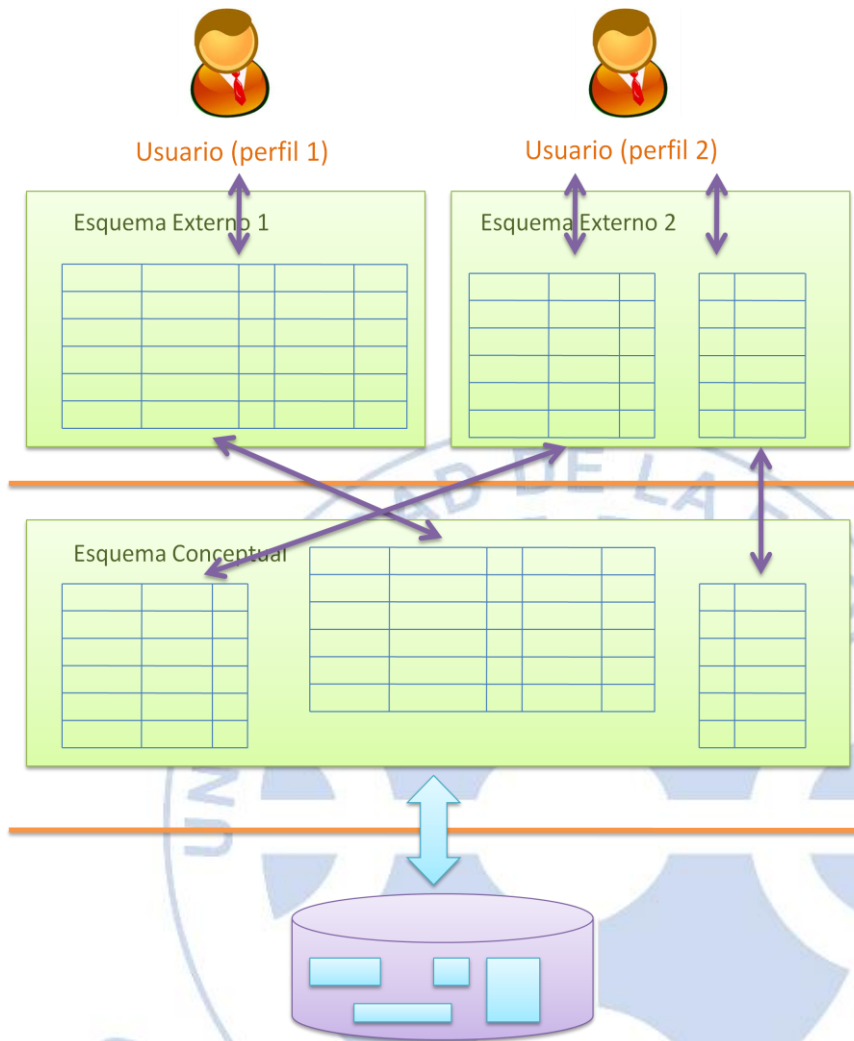


Figura nº8: Niveles de abstracción de los datos.

1.4.2 Esquema Físico

Representa la forma en la que están almacenados los datos. Esta visión sólo la requiere el administrador/a de la base de datos o DBA. El DBA la necesita para poder gestionar más eficientemente la base de datos.

En este esquema se habla de archivos, directorios o carpetas, unidades de disco, servidores,...

1.4.3 Esquema Conceptual

Se trata de un esquema teórico de los datos en el que figuran organizados en estructuras reconocibles del mundo real y en el que también aparece la forma de relacionarse los datos. Este esquema es el paso que permite modelar un problema real a su forma correspondiente en el computador.

Este esquema es la base de todos los demás. Como se verá más adelante, es el primer paso a realizar al diseñar una base de datos. En definitiva es el plano o modelo general de la base de datos.

El esquema conceptual lo realizan diseñadores/as o analistas.

1.4.4 Esquema Externo

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. Las aplicaciones creadas por los desarrolladores abstraen la realidad conceptual de modo que el usuario no conoce las relaciones entre los datos, como tampoco conoce dónde realmente se están almacenando los datos.

Los esquemas externos los realizan los programadores según las indicaciones formales de los y las analistas.

Realmente cada aplicación produce un esquema externo o vista de usuario diferente (aunque algunos pueden coincidir). El conjunto de todas las vistas de usuario es lo que se denomina esquema externo global.

1.5. Componentes de los SGBD

Los SGBD tienen que realizar tres tipos de funciones para ser considerados válidos.

■ Función de descripción o definición:

Permite al diseñador de la base de datos crear las estructuras apropiadas para integrar adecuadamente los datos. Se dice que esta función es la que permite definir las tres estructuras de la base de datos (relacionadas con los tres niveles de abstracción).

Estructura interna
Estructura conceptual
Estructura externa

Realmente esta función trabaja con metadatos. Los **metadatos** corresponden a la información de la base de datos que sirve para describir a los datos. por ejemplo, Sánchez Rodríguez y Crespo son datos; pero Primer Apellido es un metadato. Otro ejemplo de datos es decir que la base de datos contiene Alumnos o que el RUT lo forman 9 caracteres de los cuales los 8 primeros son números y el noveno un carácter alfanumérico.

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:

1. **Se definen las estructuras de datos** (por ejemplo, la tabla alumno contendrá los siguientes datos: Rut de tipo carácter, nombre de tipo carácter y edad de tipo numérico).
2. **Se definen las relaciones entre los datos** (por ejemplo, los alumnos están relacionados con las carreras a las que pertenecen).
3. **Se definen las reglas que han de cumplir los datos** (por ejemplo, al eliminar un alumno, se eliminan también las asignaturas que está cursando actualmente).

■ Función de manipulación:

Permite modificar y utilizar los datos de la base de datos. Se realiza mediante un lenguaje de modificación de datos o **DML**. Mediante ese lenguaje se puede:

1. **Añadir datos**
2. **Eliminar datos**

3. **Modificar datos**
4. **Buscar datos**

■ **Función de control:**

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; o bien usar ciertos recursos concretos de la base de datos y prohibir otros.

Es decir simplemente permite controlar la seguridad de la base de datos. El lenguaje que implementa esta función es el lenguaje de control de datos o DCL.

1.6. Funcionamiento del SGBD

El esquema siguiente presenta el funcionamiento típico de un SGBD:

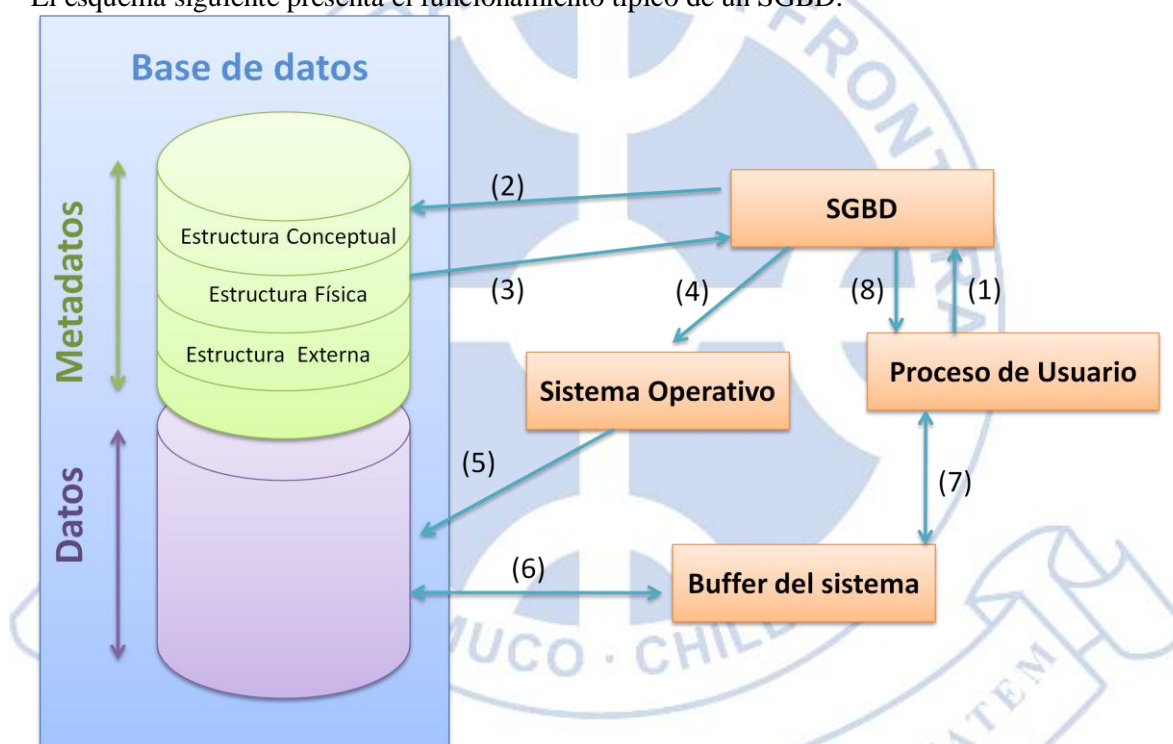


Figura n°9: Funcionamiento del SGBD

El esquema anterior reproduce la comunicación entre un proceso de usuario que desea acceder a los datos y el SGBD:

- 1) El proceso lanzado por el usuario llama al SGBD indicando la porción de la base de datos que se desea tratar.
- 2) El SGBD traduce la llamada a términos del esquema lógico de la base de datos. Accede al esquema lógico comprobando derechos de acceso y la traducción física (normalmente los metadatos se guardan en una zona de memoria global y no en el disco) .
- 3) El SGBD obtiene el esquema físico .
- 4) El SGBD traduce la llamada a los métodos de acceso del Sistema Operativo que permiten acceder realmente a los datos requeridos .

- 5) El Sistema Operativo accede a los datos tras traducir las órdenes dadas por el SGBD
- 6) Los datos pasan del disco a una memoria intermedia o buffer. En ese buffer se almacenarán los datos según se vayan recibiendo
- 7) Los datos pasan del buffer al área de trabajo del usuario (ATU) del proceso del usuario. Los pasos 6 y 7 se repiten hasta que se envíe toda la información al proceso de usuario.
- 8) 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.

1.7. Arquitectura de los SGBD

Es uno de los aspectos que todavía sigue pendiente. Desde la aparición de los primeros gestores de base de datos se intentó llegar a un acuerdo para que hubiera una estructura común para todos ellos, a fin de que el aprendizaje y manejo de este software fuera más provechoso y eficiente.

El acuerdo nunca se ha conseguido del todo, no hay estándares aceptados del todo. Aunque sí hay unas cuantas propuestas de estándares que sí funcionan como tales.

1.7.1 Organismos de Estandarización:

ISO (International Organization for Standardization). Es un organismo internacional de definición de estándares de gran prestigio.

IEC (International Electrotechnical Commission). Organismo de definición de normas en ambientes electrónicos. Es la parte, en definitiva de ISO, dedicada a la creación de estándares.

JTC 1 (Joint Technical Committee). Comité parte de IEC dedicado a la tecnología de la información (informática). En el campo de las bases de datos, el subcomité SC 21 (en el que participan otros organismos nacionales, como el español AENOR) posee un grupo de trabajo llamado WG 3 que se dedica a las bases de datos. Este grupo de trabajo es el que define la estandarización del lenguaje SQL entre otras cuestiones.

ANSI (American National Standards Institute) es un organismo científico de Estados Unidos que ha definido diversos estándares en el campo de las bases de datos. X3 es la parte de ANSI encargada de los estándares en el mundo de la electrónica. Finalmente SPARC, System Planning and Repairs Committee, comité de planificación de sistemas y reparaciones es una subsección de X3 encargada de los estándares en Sistemas Informáticos en especial del campo de las bases de datos. Su logro fundamental ha sido definir un modelo de referencia para las bases de datos (que se estudiará posteriormente).

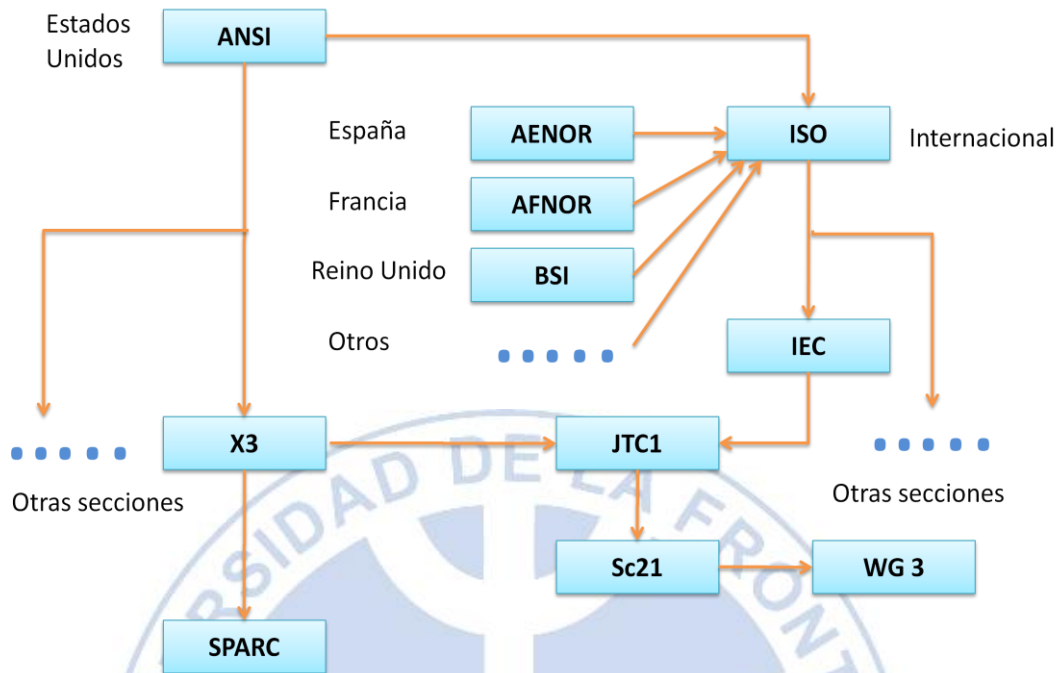


Figura nº10: Relaciones entre los estándares

En la actualidad ANSI para Estados Unidos e ISO para todo el mundo son nombres equivalentes en cuanto a estandarización de bases de datos, puesto que se habla ya de un único modelo de sistema de bases de datos.

El organismo ANSI ha marcado la referencia para la construcción de SGBD. El modelo definido por el grupo de trabajo SPARC se basa en estudios anteriores en los que se definían tres niveles de abstracción necesarios para gestionar una base de datos. ANSI profundiza más en esta idea y define cómo debe ser el proceso de creación y utilización de estos niveles.

En el modelo ANSI se indica que hay tres modelos: externo, conceptual e interno. Se entiende por modelo, el conjunto de normas que permiten crear esquemas (diseños de la base de datos). Los esquemas externos reflejan la información preparada para el usuario final, el esquema conceptual refleja los datos y relaciones de la base de datos y el esquema interno la preparación de los datos para ser almacenados.

El esquema conceptual contiene la información lógica de la base de datos. Su estructuración y las relaciones que hay entre los datos.

El esquema interno contiene información sobre cómo están almacenados los datos en disco. Es el esquema más cercano a la organización real de los datos.

En definitiva el modelo ANSI es una propuesta teórica sobre cómo debe funcionar un sistema gestor de bases de datos (sin duda, la propuesta más importante).

La Figura nº11 presenta la propuesta de ANSI.

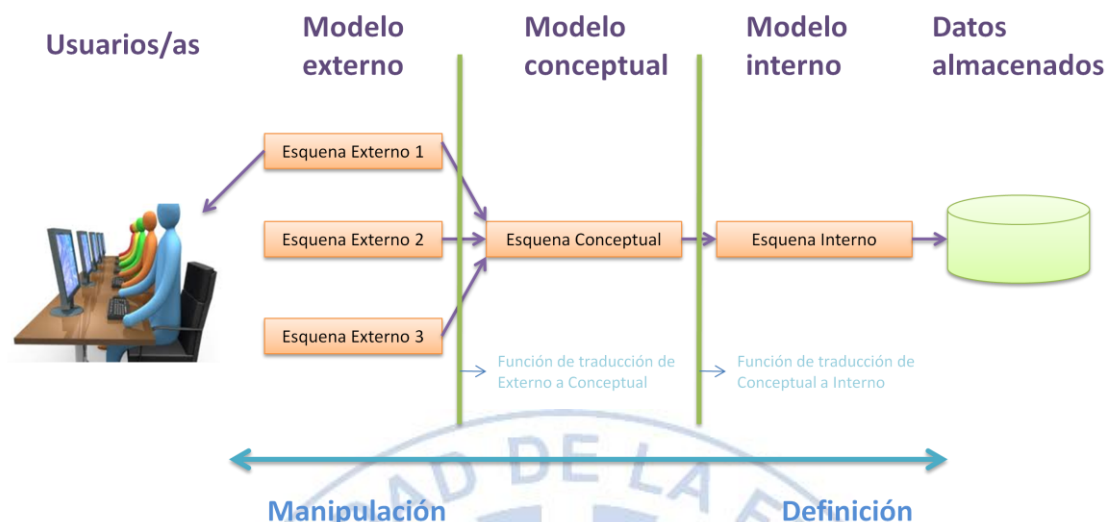


Figura nº11: Propuesta del estándar ANSI.

1.7.2. Tipos De SGBD

Como se ha visto en los apartados anteriores, cada SGBD puede utilizar un modelo diferente para los datos. Por lo que hay modelos conceptuales diferentes según qué 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 (referido a cualquier SGBD de ese tipo) y el modelo propiamente interno (aplicable sólo a un SGBD en particular). De hecho en la práctica al definir las bases de datos desde el mundo real hasta llegar a los datos físicos se pasa por los siguientes esquemas:

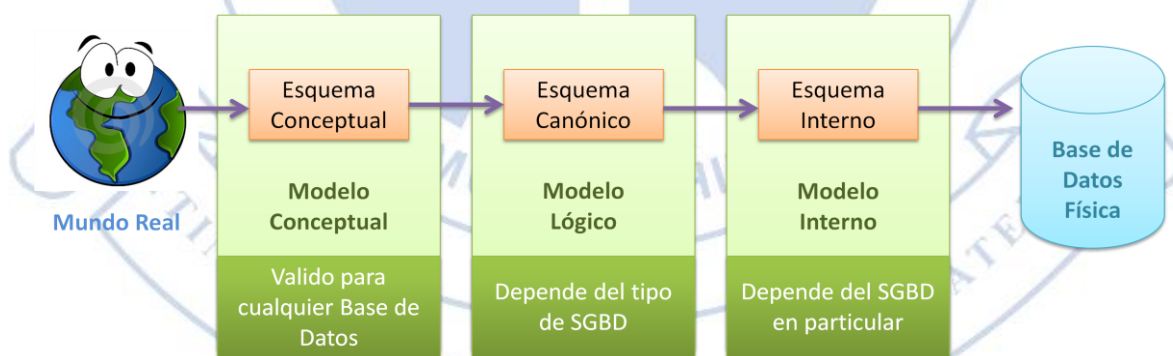


Figura nº12: Tipos de esquemas.

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 SGBD 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 computador. 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:

■ Modelo Jerárquico

Era utilizado por los primeros SGBD, desde que IBM lo definió para su IMS (Information Management System, Sistema Administrador de Información) en 1970. Se le llama también modelo en árbol debido a que utiliza una estructura en árbol para organizar los datos.

La información se organiza con un jerarquía en la que la relación entre las entidades de este modelo siempre es del tipo padre / hijo. De esta forma hay una serie de nodos que contendrán atributos y que se relacionarán con nodos hijos de forma que puede haber más de un hijo para el mismo padre (pero un hijo sólo tiene un padre).

Los datos de este modelo se almacenan en estructuras lógicas llamadas segmentos. Los segmentos se relacionan entre sí utilizando arcos.

La forma visual de este modelo es de árbol invertido, en la parte superior están los padres y en la inferior los hijos.

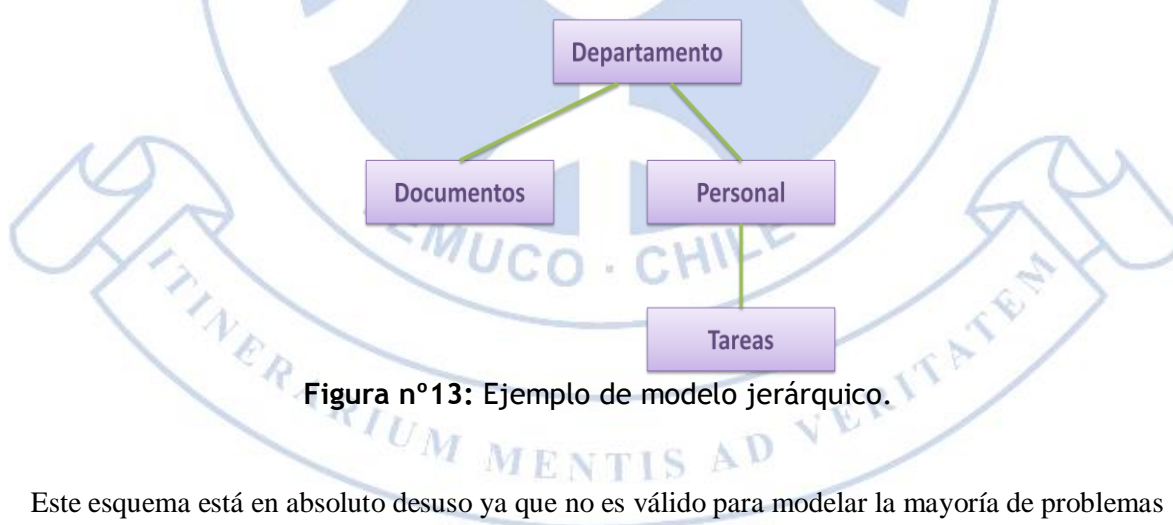


Figura n°13: Ejemplo de modelo jerárquico.

Este esquema está en absoluto desuso ya que no es válido para modelar la mayoría de problemas de bases de datos.

■ Modelo en Red (CODASYL)

Es un modelo que tuvo una gran aceptación (aunque apenas se utiliza actualmente). En especial se hizo popular la forma definida por Codasyl a principios de los 70 que se ha convertido en el modelo en red más utilizado.

El modelo en red organiza la información en registros (también llamados nodos) y enlaces. En los registros se almacenan los datos, mientras que los enlaces permiten relacionar estos datos. Las bases de datos en red son parecidas a las jerárquicas sólo que en ellas puede haber más de un padre.

En este modelo se pueden representar perfectamente cualquier tipo de relación entre los datos (aunque el Codasyl restringía un poco las relaciones posibles), pero hace muy complicado su manejo.

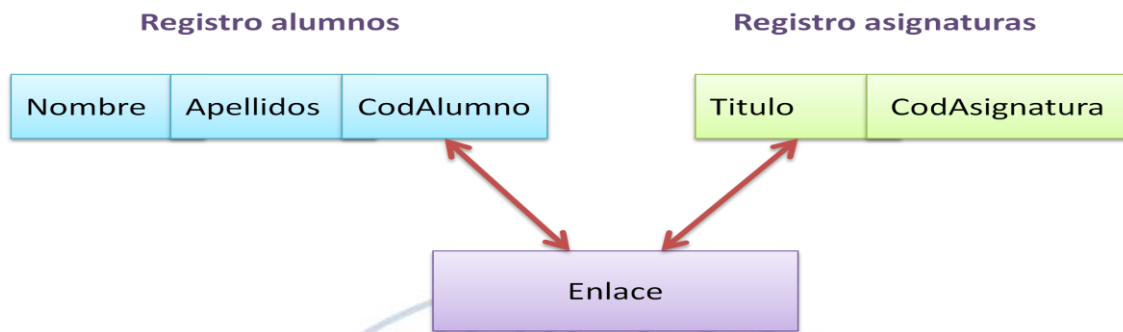


Figura nº14: Ejemplo de modelo en red.

■ Modelo Relacional

En este modelo los datos se organizan en tablas cuyos datos se relacionan. Es el modelo más popular y se describe con más detalle en capítulos siguientes de este libro, ya que será el que usaremos para implementar una base de datos en un motor de bases de datos comercial.

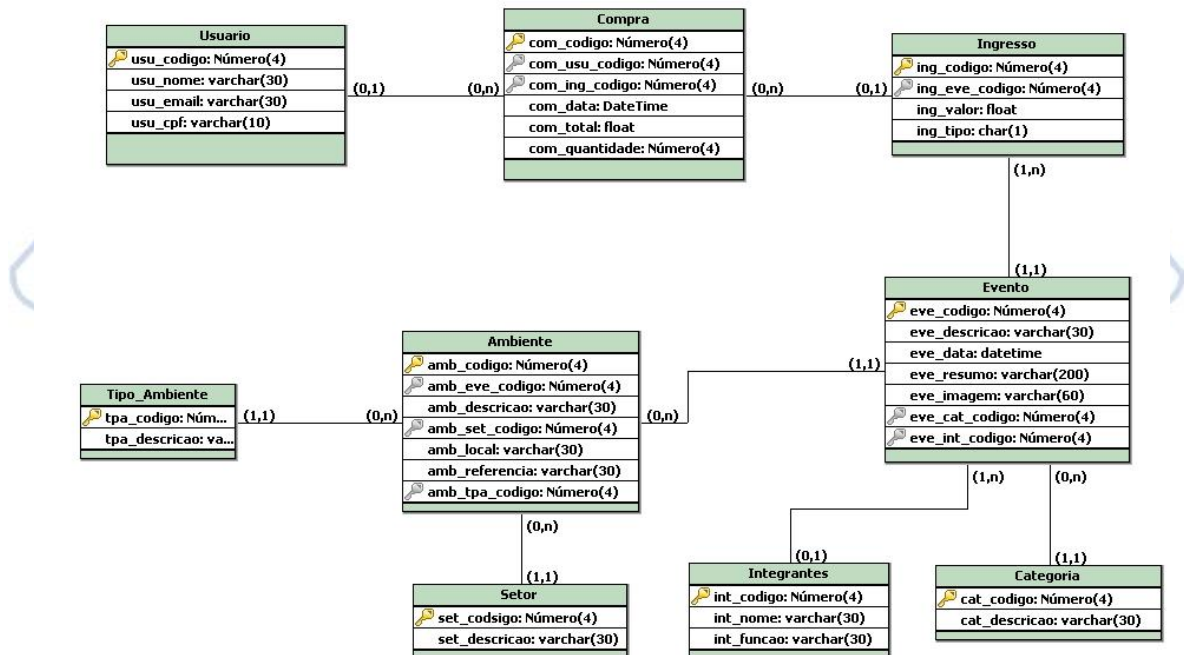


Figura nº15: Ejemplo de modelo Relacional.

■ Modelo de Bases de Datos Orientadas a Objetos

Desde la aparición de la programación orientada a objetos (POO) se empezó a pensar en bases de datos adaptadas a estos lenguajes. La programación orientada a objetos permite cohesionar datos y procedimientos, haciendo que se diseñen estructuras que poseen datos (atributos) en las

que se definen los procedimientos (operaciones) que pueden realizar con los datos. En las bases orientadas a objetos se utiliza esta misma idea.

A través de este concepto se intenta que estas bases de datos consigan arreglar las limitaciones de las relacionales. Por ejemplo el problema de la herencia (el hecho de que no se puedan realizar relaciones de herencia entre las tablas), tipos definidos por el usuario, disparadores (triggers) almacenables en la base de datos, soporte multimedia...

Se supone que son las bases de datos de tercera generación (la primera fue las bases de datos en red y la segunda las relacionales), lo que significa que el futuro parece estar a favor de estas bases de datos. Pero siguen sin reemplazar a las relacionales, aunque son el tipo de base de datos que más ha estado creciendo en los últimos años.

Su modelo conceptual se suele diseñar en UML (Lenguaje de Modelado Unificado) y el lógico actualmente en ODMG (Grupo de administración de objetos de datos, organismo que intenta crear estándares para este modelo).

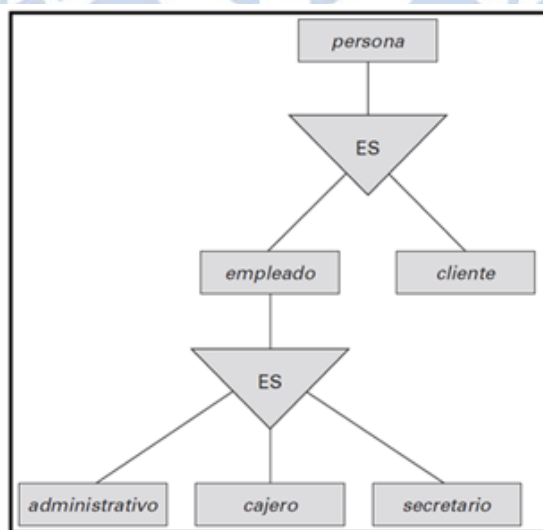


Figura n°16: Ejemplo de modelo de Datos Orientado a Objetos.

■ Bases De Datos Objetos - Relacionales

Tratan de ser un híbrido entre el modelo relacional y el orientado a objetos. El problema de las bases de datos orientadas a objetos es que requieren reinvertir capital y esfuerzos de nuevo para convertir las bases de datos relacionales en bases de datos orientadas a objetos. En las bases de datos objeto relacionales se intenta conseguir una compatibilidad relacional dando la posibilidad de integrar mejoras de la orientación a objetos.

Estas bases de datos se basan en el estándar SQL 99. En ese estándar se añade a las bases relacionales la posibilidad de almacenar procedimientos de usuario, triggers, tipos definidos por el usuario, consultas recursivas, bases de datos OLAP, tipos LOB,...

Las últimas versiones de la mayoría de las clásicas grandes bases de datos relacionales (Oracle, SQL Server, Informix, ...) son objeto relacionales.

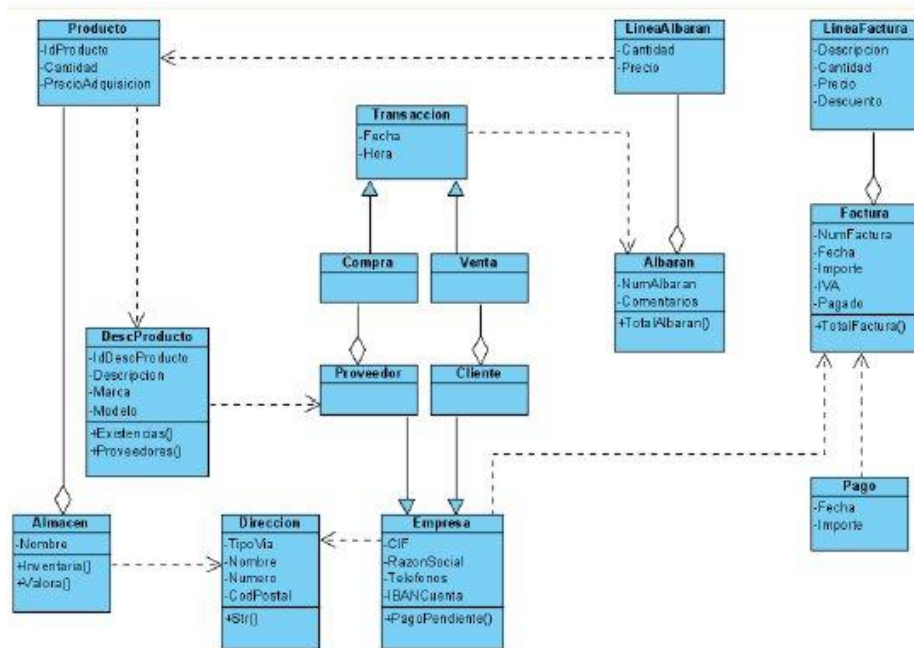


Figura n°17: Ejemplo de modelo de Datos Objeto Relacional.

■ Bases De Datos NOSQL

Bajo este nombre se agrupan las bases de datos (con arquitecturas muy diversas) pensadas para grabar los datos de manera veloz para así poder atender a miles y miles de peticiones. Es decir, es el modelo de las bases de datos que se utilizan en los grandes servicios de Internet (como twitter, Facebook, Amazon,...).

La idea es que los datos apenas necesitan validarse y relacionarse y lo importante es la disponibilidad de la propia base de datos. El nombre NoSQL, hace referencia a que este modelo de bases de datos rompe con el lenguaje SQL (el lenguaje de las bases de datos relacionales, las bases de datos dominantes hasta la actualidad) para poder manipular los datos con lenguajes de otro tipo.

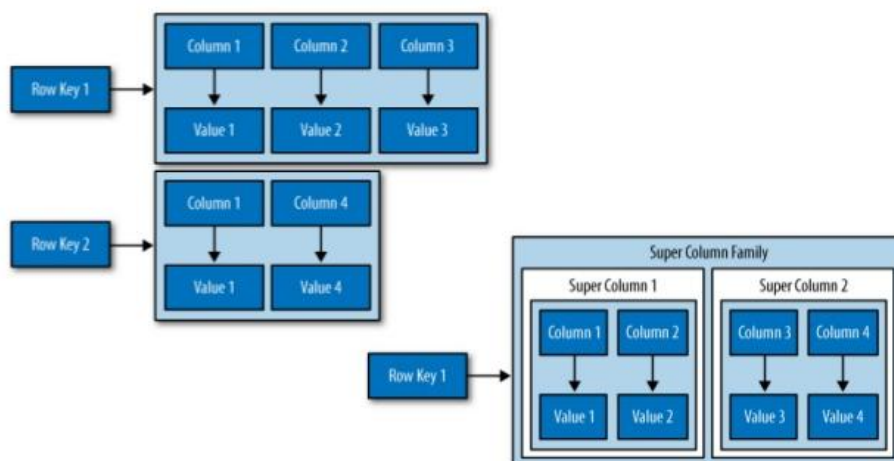


Figura n°18: Ejemplo de modelo de Datos NoSql.

1.7.3. Arquitectura del Sistema de Bases de Datos

La arquitectura de un sistema de base de datos está influenciada en gran medida por el sistema informático subyacente en el que se ejecuta el sistema de base de datos.

En la arquitectura de un sistema de base de datos se reflejan aspectos como la conexión en red, el paralelismo y la distribución. A continuación te describimos las más importantes:

■ Arquitectura Centralizada

Los sistemas de BD centralizados se ejecutan en un único sistema informático sin interactuar con ningún otro computador.

Hay dos formas de utilizar las computadoras: como sistema monousuario (computadores personales y estaciones de trabajo) o multiusuario (tenemos más discos y más memoria; disponemos de varias CPU o Unidad Central de Procesos y trabajamos con un Sistema Operativo multiusuario). Éste último se encarga de dar servicio a un gran número de usuarios que están conectados al sistema a través de terminales.

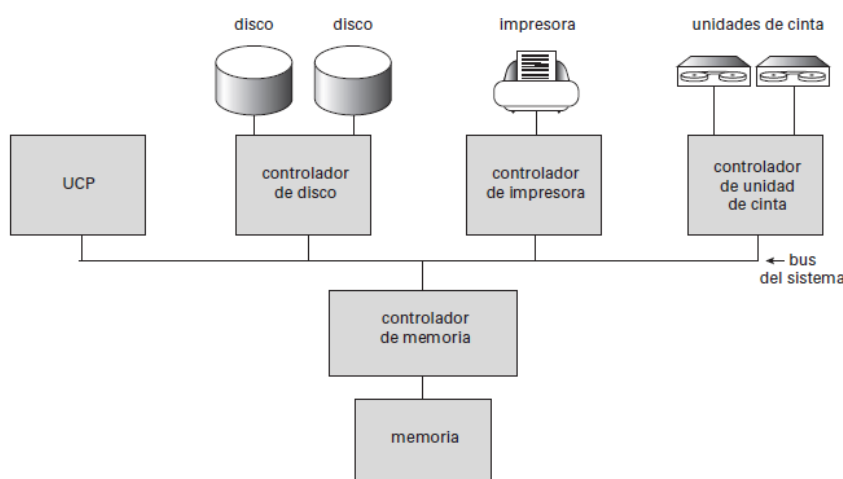


Figura n°19: Ejemplo de sistema informático centralizado.

Los sistemas de BD que operan sobre sistemas monousuarios no tienen control de concurrencia. Las facilidades de recuperación o no existen o son primitivas. La mayoría de estos sistemas no admiten SQL y el lenguaje de consulta es muy simple.

Aunque los computadores tienen varios procesadores, utilizan paralelismo de grano grueso donde unos pocos procesadores comparten la misma memoria principal. Las BD que se ejecutan en estas máquinas no dividen una consulta simple entre los distintos procesadores. Lo que hacen es ejecutar cada consulta en un único procesador produciéndose la concurrencia de varias consultas.

■ Arquitectura Cliente-Servidor

Las computadoras personales son ahora más rápidas, potentes y baratas distanciándose de una arquitectura centralizada. Los terminales han sido sustituidos por las computadoras personales. Es por ello que los sistemas centralizados actúan hoy como sistemas servidores que satisfacen las peticiones generadas por los sistemas clientes, es decir, sistemas cliente-servidor.

Cualquier red de área local puede ser considerada como un sistema cliente-servidor, desde el momento en que el cliente solicita servicios como datos, archivos o imprimir desde el servidor.

Los sistemas cliente-servidor no están limitados a aplicaciones de bases de datos. Cualquier aplicación que tenga una interfaz de usuario (front-end, sección frontal o parte cliente) que se ejecute localmente en el cliente y un proceso que se ejecute en el servidor (back-end, sección posterior, o sistema subyacente) es cliente-servidor.

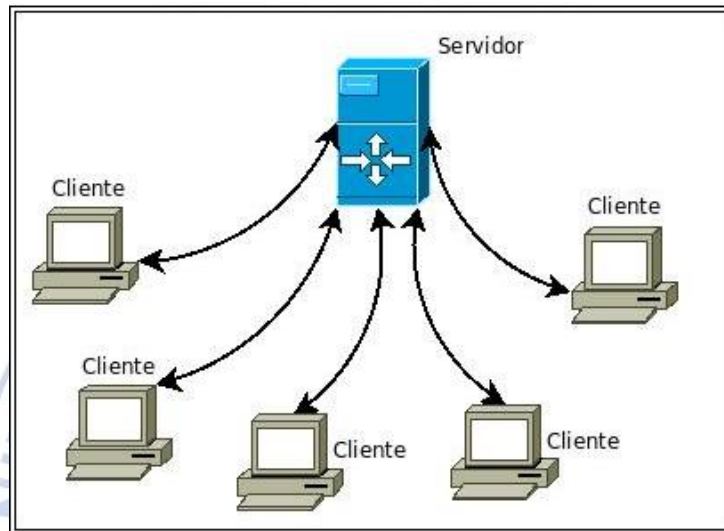


Figura nº20: Ejemplo de sistema informático cliente-servidor.

Hay varios tipos de arquitecturas cliente-servidor. La arquitectura en 2 capas se utiliza para describir los sistemas cliente-servidor en donde el cliente solicita recursos y el servidor responde directamente a la solicitud, con sus propios recursos. Esto significa que el servidor no requiere otra aplicación para proporcionar parte del servicio. La siguiente figura muestra un ejemplo de ello.

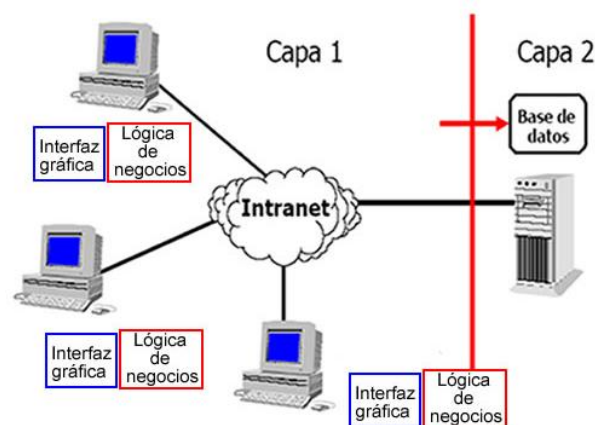


Figura nº21: Ejemplo de arquitectura cliente-servidor de 2 capas.

La arquitectura de 3 capas surgió para superar las limitaciones de la arquitectura de 2 capas. En esta arquitectura, existe un nivel intermediario. Esto significa que la arquitectura generalmente está compartida por:

- Un cliente, es decir, el computador que solicita los recursos, equipado con una interfaz de usuario (generalmente un navegador Web) para la presentación
- El servidor de aplicaciones (también denominado software intermedio), cuya tarea es proporcionar los recursos solicitados, pero que requiere de otro servidor para hacerlo
- El servidor de datos, que proporciona al servidor de aplicaciones los datos que requiere.

La figura siguiente presenta un ejemplo:

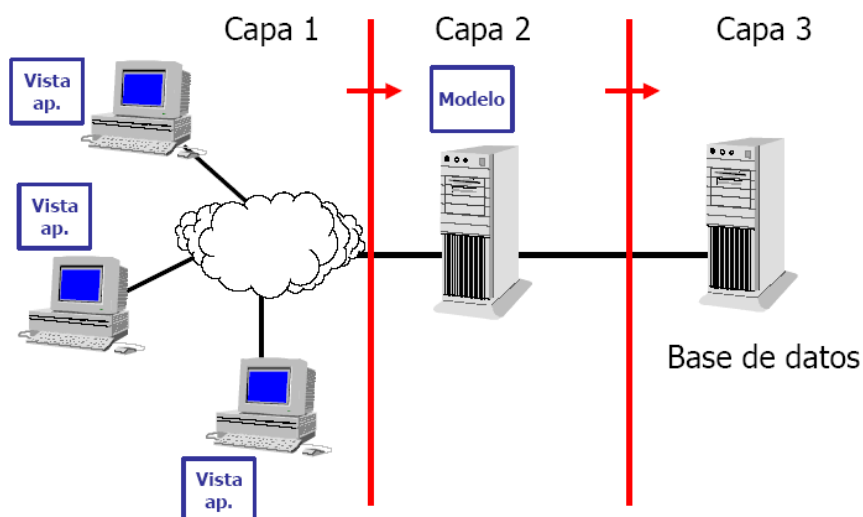


Figura n°22: Ejemplo de arquitectura cliente-servidor de 3 capas.

En definitiva, la arquitectura de 3 capas es usada cuando se necesita un diseño cliente-servidor que proporcione, en comparación con la arquitectura de 2 capas, incrementar el rendimiento, flexibilidad, mantenibilidad, reusabilidad y escalabilidad mientras se esconde la complejidad del procesamiento distribuido al usuario.

■ Arquitectura Distribuida

Los computadores de un sistema distribuido se comunican entre sí a través de diversos medios de comunicación, tales como cables de alta velocidad o líneas telefónicas. Ver figura n°23.

Los procesadores de un sistema distribuido pueden variar en cuanto su tamaño y función. Pueden incluir microcomputadores pequeños, estaciones de trabajo y sistemas de computadores grandes de aplicación general. Estos procesadores reciben diferentes nombres, tales como localidades, nodos o computadores.

Por otro lado, un sistema distribuido de bases de datos consiste en un conjunto de localidades, cada uno de las cuales puede participar en la ejecución de transacciones que accedan a datos de una o varias localidades. La diferencia principal entre los sistemas de base de datos centralizados y distribuidos es que, en los primeros, los datos residen en una sola localidad, mientras que, en los últimos, se encuentran en varias localidades. Observa en la figura n°23 que la base de datos se encuentra distribuida en varios países, Japón, México, entre otros. En este caso la base de datos organizacional se encuentra dividida en varios servidores, incluso, puede existir réplica de datos si se requiere.

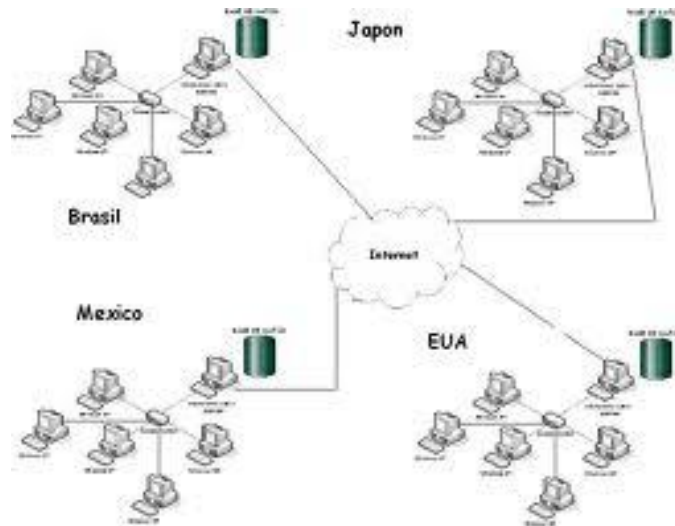


Figura nº23: Ejemplo de arquitectura distribuida.

Resumen

En este capítulo te presentamos una introducción a los conceptos fundamentales de un Sistema de Bases de Datos y cómo estos se relacionan entre sí.

En primer lugar, te explicaremos los conceptos de Bases de Datos y Sistemas de Gestión de Bases de Datos (SGBD), para luego dar lugar a los niveles de abstracción y funcionamiento de los SGBD, las diversas arquitecturas y sus características principales.

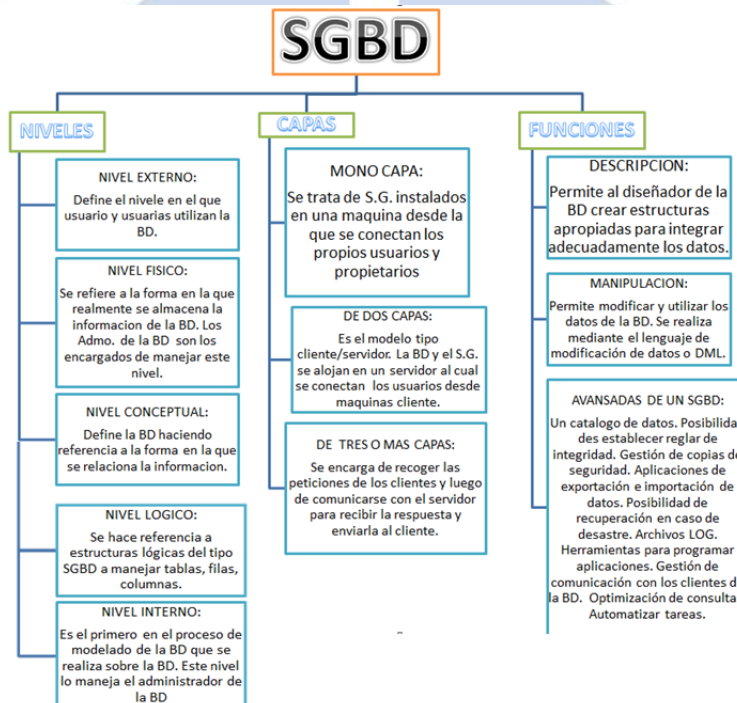


Figura nº24: Resumen SGBD.