

TEMA 1. ALMACENAMIENTO DE LA INFORMACIÓN (Parte II)

1. INTRODUCCIÓN

2. SISTEMAS CLÁSICOS DE GESTIÓN DE FICHEROS

2.1. CONCEPTOS BÁSICOS

2.2. CLASIFICACIÓN DE LOS FICHEROS SEGÚN SU ORGANIZACIÓN

2.3. MODOS DE ACCESOS DE LOS FICHEROS SEGÚN SU ORGANIZACIÓN

2.4. CLASIFICACIÓN DE LOS FICHEROS SEGÚN SU USO

2.5. CLASIFICACIÓN DE LOS FICHEROS EN LA ACTUALIDAD

3. SISTEMAS GESTORES DE BASES DE DATOS

3.1. CONCEPTO DE BASES DE DATOS

3.2. CONCEPTO DE SGBD

3.3. COMPONENTES DE UN SISTEMA DE BASE DE DATOS

3.4. CARACTERÍSTICAS DEL DISEÑO DE BASES DE DATOS

3.5. ARQUITECTURA EN TRES NIVELES O REFERENCIA ANSI-SPARC

3.6. PRINCIPALES FUNCIONES DE UN SGBD

3.7. DISEÑO DE UNA BASE DE DATOS

3.8. TIPOS DE SGBD

3.9. BASES DE DATOS CENTRALIZADA Y BASES DE DATOS DISTRIBUIDAS.
FRAGMENTACIÓN.

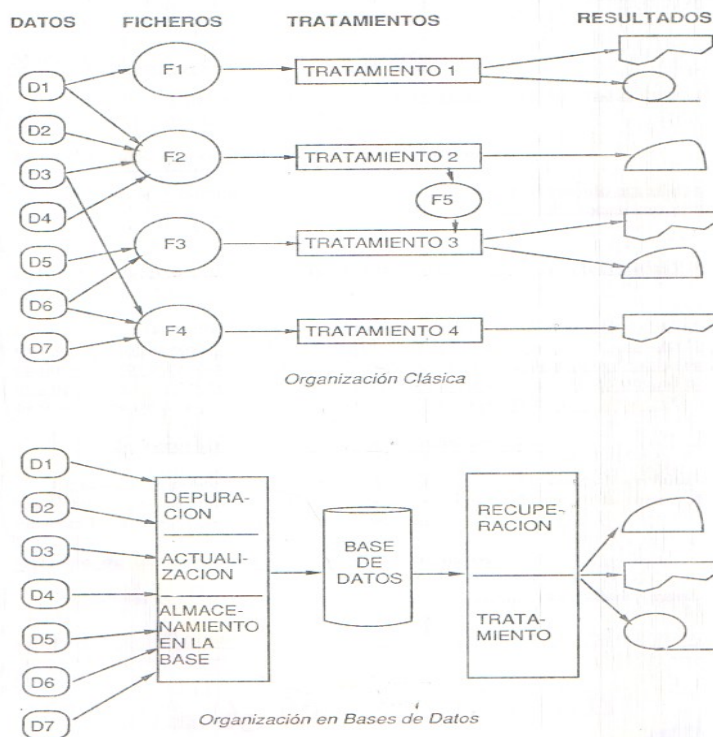
3. SISTEMAS GESTORES DE BASES DE DATOS

3.1. CONCEPTO DE BASE DE DATOS

Colecciones de datos que están lógicamente relacionados entre sí, que tiene una definición y una descripción comunes y que están estructurados de una forma particular. Podemos decir que una Base de Datos es también, un modelo del mundo real, y como tal, debe poder servir para toda una gama de empleos y aplicaciones.

Fuente: wikipedia

Una **base de datos** o **banco de datos** (en ocasiones abreviada con la sigla *BD* o con la abreviatura *b.d.*) es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.



3.2. CONCEPTO DE SISTEMA GESTOR DE BASES DE DATOS (SGBD)

Los sistemas de gestión de bases de datos (en inglés database management system, abreviado DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

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

3.3. COMPONENTES DE UN SISTEMA DE BASE DE DATOS

Los componentes o elementos de un sistema de base de datos son los siguientes:

✓ **DATOS**

✓ **LA METABASE.**

- El Diccionario de Datos y Catálogo de Variables. Contienen información sobre las tablas almacenadas, usuarios, índices, etc.

✓ **EL LOGICAL.**

- El Sistema de Gestión de la Base de Datos (SGBD).
- El Interfaz de Comunicaciones.
- Los programas y lenguajes para tratamientos específicos.

✓ **EL ADMINISTRADOR**

En una empresa existe:

Un administrador de datos (AD): Es la persona que toma las decisiones respecto a la información en la empresa. Suele ser el gerente.

Y el Administrador de la Base de Datos (ABD): Es quien proporciona el apoyo técnico para poner en práctica las decisiones.

No tiene porque ser una única persona (depende del tamaño y tipo de entidad). Puede contar con: Programadores y técnicos de administración del sistema.

El ABD es el responsable de mantener el SGBD en correcto funcionamiento. **Su objetivo debe ser que el servidor se encuentre permanentemente en disposición de proveer a los usuarios de la información que en cualquier momento requieran.**

Como funciones del ABD podemos destacar las siguientes:

- ✓ Instalación del SGBD.
- ✓ Proporcionar espacios para almacenamientos y planificar futuros requerimientos de espacios. (3)
- ✓ Definir el esquema interno. (3)
- ✓ Definir el esquema conceptual. (3)
- ✓ Definir esquema externos. (3)
- ✓ Modificación en caso necesario. (3)
- ✓ Creación y eliminación de usuarios. (1)
- ✓ Definir autorizaciones de acceso a la BD. (1)
- ✓ Definir derechos sobre objetos (tablas, vistas, etc..) de la BD. (1)
- ✓ Vincularse con los usuarios (ayudarles).
- ✓ Planificación: copias de seguridad. (2)
- ✓ Realización de copias de seguridad. (2)
- ✓ Recuperación en caso necesario. (2)
- ✓ Mantenimiento de datos almacenados en almacenes secundarios. (2)

Teniendo en cuenta la lista anterior, podemos agrupar las funciones en aquellas relacionadas con la **creación, borrado y modificación de objetos** (marcadas con un 3), y las relacionadas con **seguridad de los datos**, que abarca tanto, las tareas marcadas con (1), que se refieren a la seguridad de acceso a los datos (confidencialidad), como a las tareas marcadas con (2), que se refieren a la gestión de copias de seguridad.

✓

LOS USUARIOS

- Los Informáticos o No Informáticos

3.4. CARACTERÍSTICAS DEL DISEÑO DE BASES DE DATOS

CONTROL DE REDUNDANCIAS: En las Bases de Datos se pretende eliminar las redundancias, pero muchas veces se admite cierto nivel de redundancia con el objeto, por ejemplo, para reducir los tiempos de acceso. Entonces se habla de ***Redundancia Controlada o Redundancia Mínima.***

INDEPENDENCIA: Se refiere a que los datos y los programas de aplicación deben ser independientes, de modo que puedan ser modificados sin que se afecten mutuamente.

TÉCNICAS PARA LA SEGURIDAD DE LOS DATOS

Es **tarea del ABD** asegurar que se lleven a cabo estas técnicas.

Garantizar la seguridad de los datos implica:

Impedir los accesos indebidos y para ello existen *técnicas de control de accesos*. Existen tres niveles: físico, sistema operativo y SGBD.

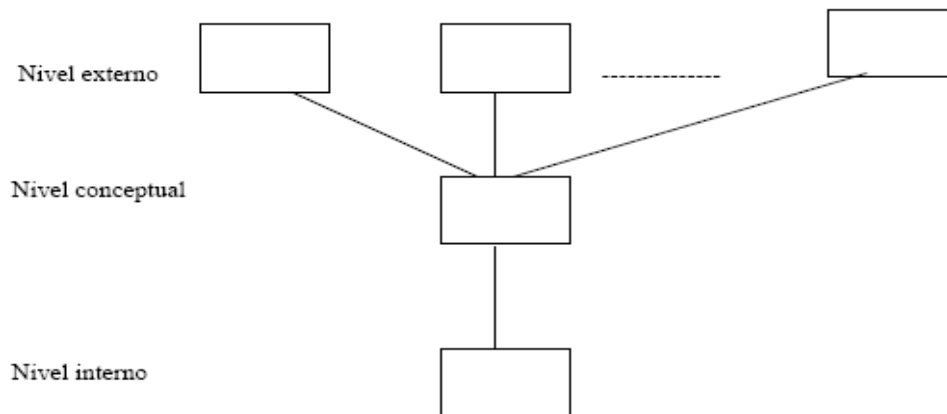
Conservar la INTEGRIDAD: El objetivo es evitar que haya errores en la información almacenada en la base de datos o en los resultados de los procesos sobre ella. Existen dos tipos de *técnicas para conseguir y conservar la integridad de las Bases de Datos*:

- ✓ ***Técnicas Preventivas:*** Orientadas a **evitar** que la Base de Datos pierda integridad. Por ejemplo, mediante la verificación previa de los datos de entrada. (Reglas de integridad: de la clave, de la entidad, referencial y semántica)
- ✓ ***Técnicas Curativas:*** Que se encargan de reconstruir la integridad perdida.

3.5. ARQUITECTURA EN TRES NIVELES O REFERENCIA ANSI-SPARC

El objetivo de presentar una arquitectura es **establecer un marco que resulta bastante útil para describir los conceptos generales de Bases de Datos y explicar la estructura de sistemas específicos**. Vamos a ajustarnos a la **arquitectura ANSI/SPARC**.

Esta arquitectura se divide en tres niveles, denominados nivel interno, conceptual y externo.



EL NIVEL INTERNO es el más cercano al almacenamiento físico, es el que se ocupa de la **forma como se almacenan físicamente los datos**. En este nivel se **describen los archivos que contienen la información**, su organización, ubicación, forma de acceso a sus registros, los tipos de registros, su longitud, los campos que lo componen, **los índices**, etc. Este nivel **se describe mediante el ESQUEMA INTERNO**. Representa la forma en la que están almacenados los datos. **Esta visión sólo la requiere el administrador**. El administrador la necesita para poder gestionar más eficientemente la base de datos. En este esquema es donde aparecen las unidades de disco, archivos y carpetas del sistema.

EL NIVEL CONCEPTUAL es la representación de los datos que intervienen en el problema. No se tiene en cuenta ni la organización física ni los métodos de acceso. Este nivel **se describe mediante el ESQUEMA CONCEPTUAL** o simplemente **ESQUEMA**. Realmente cuando se habla del **diseño de la base de datos**, se suele referir a este esquema. El esquema conceptual **lo realiza el diseñador (analista) de la base de datos**.

EL NIVEL EXTERNO es el más cercano a los usuarios, es decir, es el que se ocupa de la **forma en la que los usuarios individuales perciben los datos**. Este nivel **se describe mediante el ESQUEMA EXTERNO o SUBESQUEMA**. Se trata de la visión de los datos que poseen los usuarios finales. Esa visión es la que obtienen a través de las aplicaciones. Las aplicaciones creadas por los **desarrolladores o programadores** abstraen la realidad conceptual de modo que el usuario no conoce las relaciones entre los datos, como tampoco conoce todos los datos que realmente se almacenan.

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

Para una Base de Datos específica hay **un único esquema interno** y **un único esquema conceptual**, pero puede haber **varios esquemas externos** cada uno definido para uno o varios usuarios.

En el siguiente ejemplo se muestra el nivel conceptual de una sencilla base de datos de personal, el nivel interno correspondiente, y dos vistas externas, una para un usuario PL/I y otra para un usuario de COBOL. (El ejemplo es hipotético y se han omitido muchos detalles superfluos).

<u>Externo (PL/I)</u>	<u>Externo (COBOL)</u>
DCL 1 EMPP, 2 EMP # CHAR(6) 3 SAL FIXED BIN (31);	01 EMPC, 02 NUMEMP PIC X(6). 02 NUMDEP PIC X(4).
<u>Conceptual</u>	
EMPLEADO	
NUMERO_EMPLEADO	CARÁCTER (6)
NUMERO_DEPARTAMENTO	CARÁCTER (4)
SALARIO	NUMERICO (5)
<u>Interno</u>	
EMP_ALMAC	
PREFIJO	TIPO = BYTE (6), DESPLAZAMIENTO = 0
EMP#	TIPO = BYTE (6), DESPLAZAMIENTO = 6, INDICE = IEMP
DEPTO#	TIPO = BYTE (4), DESPLAZAMIENTO = 12
PAGA	TIPO = PALABRA, DESPLAZAMIENTO = 16

Interpretemos el ejemplo de la siguiente manera:

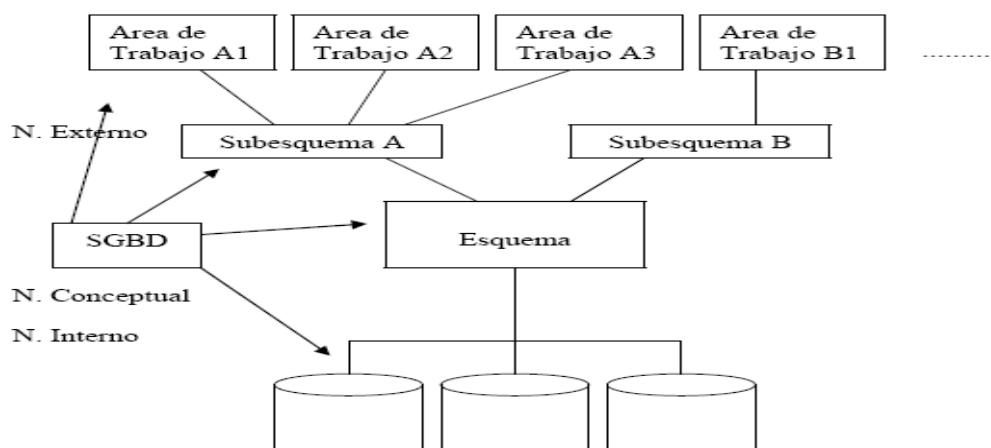
En el nivel conceptual, la base de datos contiene información relativa a un tipo de entidad llamada EMPLEADO. Cada EMPLEADO tiene un NÚMERO_EMPLEADO (seis caracteres), un NÚMERO_DEPARTAMENTO (cuatro caracteres), y un SALARIO (cinco dígitos decimales).

En el nivel interno, los empleados se representan mediante un tipo de registro llamado EMP_ALMAC. EMP_ALMAC contiene cuatro campos almacenados: un prefijo de seis bytes (el cual contiene información de control como banderas o apuntadores), y tres campos de datos correspondientes a las tres propiedades de los empleados.

Además los registros EMP_ALMAC están indexados según el campo EMP# mediante un índice llamado IEMP.

El usuario PL/I percibe una vista externa de la base de datos, en la cual cada empleado está representado por un registro de PL/I con dos campos puesto que los números de departamentos no interesan a este usuario

De manera similar, el usuario de COBOL percibe una vista externa en la que cada empleado está representado por un registro de COBOL, el cual contiene también dos campos, omitiéndose en esta ocasión los salarios.



3.6. DISEÑO DE UNA BASE DE DATOS

Consta de tres pasos:

1. Diseño Conceptual. Cuyo objetivo es obtener un **Modelo Conceptual** que nos de una representación de los recursos de información de la empresa, con independencia de los usuarios, así como del software o el hardware disponible. (Modelo E/R)

2. Diseño Lógico. Cuyo objetivo es transformar el modelo conceptual anterior en un **Modelo Lógico** totalmente adaptado al tipo de SGBD que se vaya a utilizar. (Será el modelo relacional) (Diagrama de Estructura de Datos o DED)

3. Diseño Físico. Su objetivo es conseguir un **Modelo Físico** de la Base de Datos adaptando el modelo lógico obtenido al SGBD propio de nuestra instalación.

3.7. PRINCIPALES FUNCIONES DE UN SGBD

DESCRIPCIÓN

Permite al **ABD** especificar los **elementos de la BD** que la integran, su **estructura**, y las **relaciones que existen entre ellos**, así como los controles a efectuar antes de **autorizar el acceso a la BD**.

Esta función es llevada a cabo fundamentalmente por el **Lenguaje de Definición de Datos (LDD)** propio de cada SGBD. Debe suministrar los medios para definir las tres estructuras de datos (conceptual, interna y externa).

MANIPULACIÓN

Permite a los **usuarios** de la Base de Datos **añadir, buscar, suprimir y modificar los datos** de la misma. Se lleva acabo mediante un **Lenguaje de Manipulación de Datos (LMD)**. Algunos SGBD llevan incorporados lenguajes de manipulación de datos orientados al usuario no informático que pueden usarse, en general en modo conversacional.

UTILIZACIÓN

Reúne todos los **interfaces necesarios** a los diferentes usuarios para conectarse con la Base de Datos.

3.8. TIPOS DE SGBD

Los SGBD se pueden clasificar de muchas formas, en este apartado vamos a clasificarlas según dos criterios: según la capacidad y potencia del propio gestor, y según la base de datos que gestionan.

Clasificación de los SGBD según la capacidad y potencia del gestor de bases de datos.

SGBD Ofimáticas: Son aquellas que almacenan y manipulan pequeñas bases de datos para uso doméstico y pequeñas empresas. Por ejemplo, Microsoft Access es una base de datos ofimática.

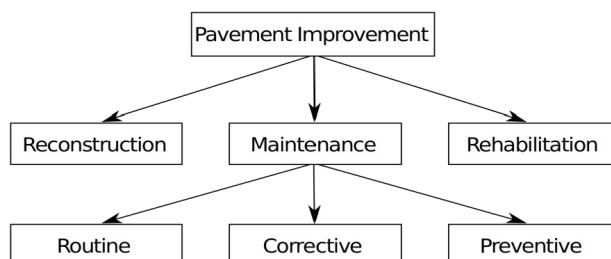
SGBD Corporativas: Son aquellas que pueden almacenar y manipular grandes bases de datos de forma rápida y eficiente, para grandes y medianas empresas. Por ejemplo, Oracle, SQL Server o MySQL.

Clasificación de los SGBD según la base de datos que gestionan (modelo de base de datos)

SGBD Jerárquico (fuente Wikipedia): En un modelo jerárquico, los datos están organizados en una estructura arbórea, lo que implica que cada registro sólo tiene un padre. Las estructuras jerárquicas fueron usadas extensamente en los primeros sistemas de gestión de datos, como el Sistema IMS por IBM, y ahora se usan para describir la estructura de documentos XML. Esta estructura permite relaciones 1:N entre los datos, y es muy eficiente para describir muchas relaciones del mundo real: tablas de contenido, ordenamiento de párrafos y cualquier tipo de información anidada.

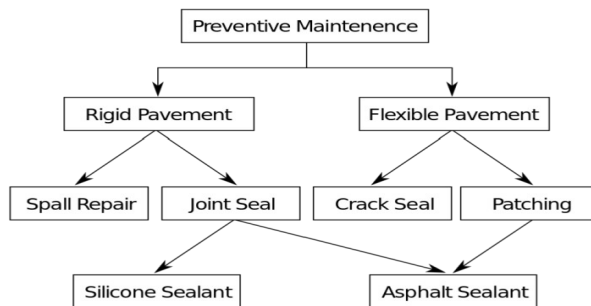
En la relación Padre-hijo: El hijo sólo puede tener un padre pero un padre puede tener múltiples hijos. Los padres e hijos están unidos por enlaces. Todo nodo tendrá una lista de enlaces a sus hijos.

Hierarchical Model



SGBD en Red (fuente Wikipedia): El modelo de red expande la estructura jerárquica, permitiendo relaciones N:N en una estructura tipo árbol que permite múltiples padres. Antes de la llegada del modelo relacional, el modelo en red era el más popular para las bases de datos. Entre los sistemas más populares que tienen arquitectura en red se encuentran Total e IDMS. IDMS en 1980 adoptó el modelo relacional y SQL, manteniendo además sus herramientas y lenguajes originales.

Network Model



SGBD Relacional: Este modelo se basa fundamentalmente en establecer relaciones o vínculos entre los datos, imaginando tablas. Este modelo se comenzó a usar en 1970, y ha ido sufriendo una serie de transformaciones hasta convertirse, hoy en día, en el **modelo más utilizado** para administrar bases de datos. **Todos los SGBD relacionales utilizan SQL (Structured Query Language)**

Entre los sistemas relacionales más conocidos y utilizados en la actualidad podemos destacar: Oracle, MySQL, MariaDB, SqlServer, PostgreSQL, SQLite.

SGBD Orientados a Objetos (fuente Wikipedia): En estos sistemas la información se representa mediante objetos. Para realizar el modelado se crea un Diagrama de Clases. Cabe destacar, **ObjectDB** que es una base de datos orientada a objetos para java, y a **Zope Object Database (ZODB)** que es una base de datos orientada a objetos para Python.

SGBD Objeto – Relacional: De forma simplificada podemos decir que este tipo de SGBD son sistemas relacionales con la peculiaridad de poder almacenar objetos. El SGBD Oracle y PostgreSQL permiten este comportamiento.

SGBD NoSQL: No requiere de estructuras de datos fijas como tablas. Estos sistemas empezaron a crecer con Twitter, Facebook, Google y Amazon.

Se utilizan en sistemas distribuidos y que por tanto, deben estar siempre disponibles y operativos gestionando un gran volumen de datos.

Entre los principales SGBD NoSQL en la actualidad encontramos: MongoDB, Firebase, Amazon DynamoDB y Cassandra, este último lo utilizan Facebook, Twitter, Instagram, Spotify o Netflix.

3.9. BASES DE DATOS CENTRALIZADA Y BASES DE DATOS DISTRIBUIDAS. FRAGMENTACIÓN

Idea Previa: Para hacernos una idea previa podemos considerar las *bases de datos centralizadas* como **una o varias bases de datos en un solo ordenador** (servidor). Mientras que las *bases de datos distribuidas* constaran de **varias bases de datos en varios ordenadores**.

Definición de Bases de Datos Distribuidas:

Es una colección de datos (base de datos) construida sobre una red y que pertenecen, lógicamente, a un solo sistema distribuido, la cual cumple entre otras las siguientes condiciones:

- La información de la base de datos esta almacenada físicamente en diferentes sitios de la red.
- En cada sitio de la red, la parte de la información, se constituye como una base de datos en sí misma.
- Las bases de datos locales tienen sus propios usuarios locales y sus propios SGBD.
- Existe un gestor global que permite que los usuarios puedan acceder a los datos desde cualquier punto de la red, como si lo hicieran con los datos de su base de datos local, es decir, para el usuario, no debe existir diferencia en trabajar con datos locales o datos de otros sitios de la red.

En consecuencia, la base de datos distribuida, es como una unidad virtual, cuyas partes se almacenan físicamente en varias bases de datos "reales" distintas, ubicadas en diferentes sitios.

Ejemplo de base de datos distribuida:

Considere un banco que tiene tres sucursales, en cada sucursal, un ordenador controla las terminales de la misma y el sistema de cuentas. Cada computador con su sistema de cuentas local en cada sucursal constituye un "sitio" de la BDD; las computadoras están conectadas por la red. Durante las operaciones normales, las aplicaciones en las terminales de la sucursal necesitan sólo acceder la base de datos de la misma. Como sólo acceden a la misma red local, se les llaman aplicaciones locales.

Desde el punto de vista tecnológico, aparentemente lo importante es la existencia de algunas transacciones que acceden a información en más de una sucursal. Estas transacciones son llamadas transacciones globales o transacciones distribuidas.

La existencia de transacciones globales será considerada como una característica que nos ayude a discriminar entre las BDD (Bases de Datos Distribuidas) y un conjunto de base de datos locales.

Una típica transacción global sería una transferencia de fondos de una sucursal a otra. Esta aplicación requiere de actualizar datos en dos sucursales diferentes y asegurarse de la real actualización en ambos sitios o en ninguno. Asegurar el buen funcionamiento de aplicaciones globales es una tarea difícil.

Algunas ventajas de las BDD

- Mejora de rendimiento: *Pues los datos serán almacenados y usados donde son generados*, lo cual permitirá distribuir la complejidad del sistema en los diferentes sitios de la red, optimizando la labor.
- **Mejora de fiabilidad y disponibilidad:** ***El fallo de uno o varios lugares o el de un enlace de comunicación no implica la inoperatividad total del sistema***, incluso si tenemos datos duplicados puede que exista una disponibilidad total de los servicios.
- Flexibilidad: *Permite acceso local y remoto de forma transparente.*
- Disponibilidad: *Pueden estar los datos duplicados con lo que varias personas pueden acceder simultáneamente de forma eficiente.* El inconveniente es que el sistema administrador de base de datos debe preocuparse de la consistencia de los mismos.

Entre los **inconvenientes de las BDD** podemos destacar la **mayor complejidad**, que juega en contra de este tipo de sistemas, pues muchas veces *se traduce en altos* gastos de construcción y mantenimiento. Esto se da por la gran cantidad de componentes Hardware y muchas aplicaciones susceptibles de fallar. Por ejemplo, el control de concurrencia y recuperación de fallos, requiere de personal muy especializado y por tal costoso.

Distribución de los datos

Una de las decisiones más importantes que el diseñador de bases de datos distribuidas debe tomar es el **posicionamiento de los datos en el sistema** y el esquema bajo el cuál lo desea hacer. Para esto existen cuatro alternativas principales: **centralizada, replicada, fragmentada, e híbrida**.

Centralizada

Si los datos están centralizados (por ejemplo, un servidor corriendo 2 máquinas virtuales), realmente tenemos un modelo similar al del Cliente/Servidor en el sentido que la BDD está **centralizada en un lugar y los usuarios están distribuidos**. Este modelo solo brinda la ventaja de tener el *procesamiento distribuido* ya que en sentido de disponibilidad y fiabilidad de los datos no se gana nada.

Replicadas

El esquema de BDD de replicación consiste en que **cada nodo debe tener su copia completa de la base de datos**. Es fácil ver que este esquema tiene un alto costo en el almacenamiento de la información. Debido a que **la actualización de los datos debe ser realizada en todas las copias**, también tiene un alto costo de escritura, pero todo esto vale la pena si tenemos un **sistema en el que se va a escribir pocas veces y leer muchas**, y dónde la **disponibilidad y fiabilidad de los datos** sea de máxima importancia.

Particionadas o Fragmentada

Este modelo consiste en que **solo hay una copia de cada elemento**, pero la información está distribuida a través de los nodos. En cada nodo se aloja uno o más fragmentos disjuntos de la base de datos. Como los fragmentos no se replican esto **disminuye el costo de almacenamiento**, pero también **sacrifica la disponibilidad y fiabilidad** de los datos.

Híbrida

Este esquema simplemente representa la **combinación del esquema de partición (fragmentación) y replicación**. Se particiona la relación y a la vez los fragmentos están selectivamente replicados a través del sistema de BDD.

