

SISTEMAS DE INFORMACIÓN: FICHEROS y BASES DE DATOS

Javier Fernández Rivera - www.aurea.es

INTRODUCCIÓN

SI: Sistema de información.

SG: Sistema gestor o de gestión.

BD: Base de datos, en ingles veremos DB.

SGBD: Sistema gestor de base de datos.

Un sistema de información "SI" : Es un conjunto de aplicaciones o programas, junto con un sistema gestor (que gestionara la información) y unos usuarios (que interactuaran con ella).

Un sistema gestor "SG": Es un conjunto de programas, aplicaciones, utilidades (software) que permite la implantación acceso y mantenimiento de la base de datos.

SI = SG + BD/FI + USERS

Con lo que SI (sistema de información) es igual al SG (sistema gestor), mas el soporte BD/FI base de datos o ficheros, y mas los usuarios que interactuaran con la información.

Importancia de los datos en el sistema de información

Para almacenar la información dentro del sistema informático de una empresa, se deben definir dos características:

1.- **Diseño lógico**; consiste en el diseño a nivel de usuario, resultante para los usuarios, aquel sistema que los usuarios ven y manejan para gestionar los datos.

2.- **Diseño físico**; consiste en el diseño a nivel del programador, define como van a ir almacenados los datos en los discos.

En el diseño físico podemos distinguir dos tipos de sistemas:

1.- Un sistema de fichero de datos

2.- Un sistema DB (base de datos).

La diferencia entre uno u otro sistema versa en que:

Los sistemas basados en ficheros de datos almacenan dichos datos de forma independiente y sin ningún tipo de relación entre ellos, ni de unidad física.

Los sistemas gestores de DB, almacenan todos sus datos con relaciones entre si de forma lógica y en algunos casos física.

El uso de uno u otro sistema depende de la fecha en la que el sistema informático fue implantado en la empresa. Es mejor una base de datos a un sistema de ficheros, pero antes esta tecnología no existía y los bancos fueron de las primeras empresas en informatizarse. Algunos bancos pues siguen usando el método de datos por ficheros.

En los años 60/70 se usaban ficheros de datos a los que se accedía a traves de lenguajes de programación como cobol, C, Basic. A mediados de los años 70 se crean las primeras DBs que evolucionan hasta las actuales SQLserver, ORACLE, Informix, DB2, SyBASE.

En microinformatica (access, DBASE, PARADOX, FOXPRO).

El diseño lógico de una base de datos consiste en un conjunto de información referente a personas, animales o cosas, que constituye interés para una empresa y que están organizados de forma coherente y sin redundancias.

Ahora bien según en donde este almacenada esa información podemos distinguir entre:

- **Sistemas de ficheros:** La información se encuentra almacenada en ficheros.
- **Sistemas de bases de datos:** La información se encuentra almacenada en una base de datos.

Tanto uno como otro sistema constan de 3 partes.

| PARTES | S. de ficheros | S. de base de datos |
|---------------------|--|--|
| 1. Usuarios | Los usuarios del sistema (admins,opers, users) | Los usuarios del sistema (admins,opers, users) |
| 2. Programas | Programa/s para interactuar con los ficheros | Gestores de bases de datos (MySQL) |
| 3. Datos | Ficheros (registros, campos) "datos" | Base de datos (DB) "datos" |

Usuarios: Conjunto de usuarios del sistema gestor.

El administrador del sistema: administra el sistema, crea cuentas de operadores, mantiene los ficheros, etc.

Los operadores del sistema: son aquellos que tienen autorización para interactuar con la información del sistema, añadir más información, quitar la ya existente, etc.

Resto de usuarios: Simplemente pueden ver la información.

Programas: Es el software capaz de interactuar con los datos. Estos programas funcionarán como interfaz entre el usuario que mete los datos a través de él y los soportes de almacenamiento, ya sea un fichero o una base de datos.

Cuanto más crecen las necesidades de la empresa, mejoran al mismo tiempo dichos programas.

Datos: Es el soporte lógico donde se encuentran almacenada la información que le pasan los programas. Si trabajamos con un sistema gestor de bases de datos será una base de datos, si se trata de un sistema de ficheros serán ficheros.

Problemas de respectivos sistemas de información

Sistema de bases de datos

1. **La semántica** (el sentido de las cosas), haber si tenemos dentro del sistema gestor de ficheros un programa que se encarga de meter clientes, este debe testear si el DNI del cliente es correcto, al igual que otros datos como la fecha. Sería absurdo dejar que el usuario metiera 35 de septiembre.
2. **Autorización de usuarios:** Lógicamente en nuestro sistema gestor de ficheros no queremos que entre cualquier persona, con lo que debemos hacer algún tipo de autorización. El programa ahora pide login (Nombre de usuario) y password (contraseña), pero observamos que al tener varios departamentos en nuestra empresa sería bueno que cada departamento solo pudiera acceder a unos datos concretos, con lo cual el programa ahora debe gestionar más de una cuenta de usuario. Ahora supongamos que hay algunos usuarios que tienen autorización para acceder al sistema desde más de un departamento, y que el usuario es dado de baja en la empresa y se le quita acceso a 3 departamentos pero no a todos los que podía acceder, este usuario puede pues acceder a un 4º departamento y cargarse toda la información. Se debe tener cuidado con eso.
3. **Concurrencia:** No existe un control de concurrencia de acceso a fichero. Si acceden desde varios programas no pueden controlar la concurrencia.
4. **Tiempo de proceso:** Es mucho mayor el tiempo de proceso y el retardo en devolver la información de este sistema en comparación con un sistema de bases de datos, que por el contrario es mucho más rápido.
5. **Peor gestión lógica:** En un sistema de ficheros los datos introducidos son tratados todos por igual, y de forma independiente, mientras que en una base de datos son relacionados entre sí, y su gestión se debe según a la semántica. En un sistema de ficheros los datos no relación alguna, mientras que en un sistema de bases de datos, los datos tienen relación lógica y en algunos casos físicas.

Sistema de ficheros

1. **La redundancia:** Típico problema informático, tener datos o información que se repite innecesariamente y que causa conflicto y errores en todo el sistema. Por ejemplo tenemos en un listado de clientes una cuenta de cliente repetida 2 veces, con el nombre de "Javier Fernández rivera". Esto puede provocar muchos errores.
2. **La inconsistencia:** Serían datos redundantes pero diferentes. En el caso anterior todos los datos de las 2 cuentas repetidas eran iguales; la edad, dirección, etc. En este caso si la dirección o edad es distinta, estaríamos hablando de la inconsistencia que también plantea graves problemas.
3. **El aislamiento:** Serían datos almacenados a los que es imposible acceder. Tenemos por ejemplo una cuenta de cliente. Dicho cliente (pepito) tiene 34 años, a la hora de realizar una búsqueda por edad (entre 30 y 35 años) aparecen todos los clientes con esa edad menos "pepito". Estaríamos ante un aislamiento.
4. **Independencia** tanto física como lógica de los datos.
5. **Se necesita personal especializado:** los conocimientos que resultan imprescindibles para un correcto y eficaz uso de la DB. Tanto para el diseño y administración de la DB significan la necesidad de personal especializado.
6. **Instalación:** la implantación de un Sistema de base de datos puede llevar consigo un coste elevado, sin rentabilidad a corto plazo y larga instalación e implantación.

La elección de uno u otro sistema depende de la fecha de su implantación, anteriormente se solían usar sistemas de ficheros (no existían las bases de datos), una vez su aparición, las nuevas empresas pasan a implantar sistema de bases de datos.

Las empresas que disponen desde hace años de su sistema de ficheros pueden no pasarse a una base de datos, dicho volcado de información sería una tarea ardua y puede que bastante costosa.

En los años 60/70 se usaban ficheros de datos a los que se accedía a través de lenguajes de programación como cobol, C, Basic. A mediados de los años 70 se crean las primeras DBs que evolucionan hasta las actuales SQLserver, ORACLE, Informix, DB2, SyBASE.

Nosotros prestaremos mas importancia a los SI basados en bases de datos

Sistemas de información basados en ficheros

Dentro de este sistema de información agruparemos y almacenaremos la información en ficheros. A medida que crecen las necesidades de información en la empresa se van creando nuevos programas y aplicaciones para acceder a nuevos datos o insertarlos junto a los ya existentes.

Además los diferentes departamentos de una empresa pueden compartir los mismos datos o usar datos diferentes y cada departamento va a tener sus propias aplicaciones informáticas.

En este tipo de SI se puede dar que llegue el momento de crear un programa nuevo para gestionar los datos en los ficheros, esto sería una tarea dificultosa, ya que los datos contenidos en los ficheros tienen un formato o sintaxis determinada y pueden estar hechos en un lenguaje de programación distinto al que vallamos a usar.

Con lo que esto podía provocar problemas de aislamiento o redundancia de los datos.

Sistemas de información basados en bases de datos

La aparición de las bases de datos se produce en los años 70 y en el año 1963 tuvo lugar en "Santa Mónica" (EEUU) un congreso en cuyo titulo aparecían las siglas DB (data base "base de datos"), en este congreso se propuso que no fuera admitido universalmente.

En 1967 el grupo CODASYL (empresa de estandarizaciones) decide cambiar su nombre y pasa a llamarse DBTASK group.

Existen múltiples definiciones de DBs en el año 75. Todas las definiciones de DBs coincidían en que:

- Una DB es un conjunto, colección, o deposito de datos almacenados en un soporte de información no volátil.
- Los datos están interrelacionados y estructurados de acuerdo con un modelo capaz de recoger el máximo contenido semántico. Por ejemplo, si tuviéramos una base de datos de "informática básica" podríamos formar un árbol con la posible estructuración (software, hardware, etc). Una vez se van metiendo los datos en esa base de datos todos estarán pues relacionados entre si por esa estructura previamente fijada.

- 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 (Estructurada 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 restricciones en el mundo real. Los procedimientos de actualización y recuperación, comunes y bien determinados, facilitarán la seguridad del conjunto de los datos.

Otras definiciones dadas son:

- Colección de datos correspondientes a las diferentes perspectivas de un sistema de información (de una empresa o institución), existentes en algún soporte de tipo físico (normalmente de acceso directo), agrupados en una organización integrada y centralizada en la que figuran no sólo los datos en sí, sino también las relaciones existentes entre ellos, y de forma que se minimiza la redundancia y se maximiza la independencia de los datos de las aplicaciones que los requieren." (GUILERA, 1993: 377)
- Una base de datos es una colección de datos estructurados según un modelo que refleje las relaciones y restricciones existentes en el mundo real. Los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de éstas, y su definición y descripción han de ser únicas estando almacenadas junto a los mismos. Por último, los tratamientos que sufran estos datos tendrán que conservar la integridad y seguridad de éstos." (MOTA, CELMA y CASAMAYOR, 1994: 9)

Un sistema de bases de datos debe cumplir

1. **Independencia de los datos:** Las aplicaciones no deben verse afectadas por cambios en la estructura de los datos
2. **Integridad de los datos:** Los datos deben cumplir ciertas restricciones que aseguren la correcta introducción, modificación y borrado de los mismos. La redundancia de los datos debe ser controlada de forma que no existan, duplicaciones perjudiciales, ni innecesarias, y que las redundancias físicas convenientes muchas veces en caso de existir sean tratadas automáticamente por el mismo sistema y que puedan existir inconsistencias. Por tanto, un dato se actualizará lógicamente por el usuario de forma única, y el sistema se preocupará de cambiar físicamente todos aquellos campos en los que el dato estuviese repetido en caso de existir redundancia física; Es lo que se denomina también redundancia controlada por el sistema.
3. **Seguridad:** Establecer diferentes niveles de acceso a los datos a diferentes tipos de usuarios.

Arquitectura de una base de datos o Modelo de abstracción

Las bases de datos usan una arquitectura de 3 niveles por normativa general y popularizada. Dicha arquitectura de 3 niveles dispone de las siguientes capas:

Capa 1: Nivel interno ("almacenamiento"), es el nivel mas bajo dentro del modelo de abstracción. En este nivel se define el como se almacenan los datos en los distintos soportes de almacenamiento (discos, cintas, etc), así como también su método de acceso.

Capa 2: Nivel conceptual ("Diseño" y "Datos"), es el nivel medio, en este nivel se definen los datos y las relaciones entre ellos, así como también su organización y visualización.

Capa 3: Nivel externo ("Usuarios"), es el nivel superior, en este nivel se dan las vistas, visualizaciones de los datos, este nivel seria una prolongación del nivel anterior (conceptual) a la que los usuarios tienen acceso.

El Sistema de Gestor de Bases de Datos

SGBD es el conjunto coordinado de programas, lenguajes y procedimientos que permite la implantación, acceso y mantenimiento de la base de datos.

El SGBD funciona básicamente como una interfaz (elemento de enlace o comunicación) entre el usuario y la base de datos.

El SGBD, junto con la base de datos y con los usuarios, constituye el Sistema de la Base de Datos.

Antiguamente las bases de datos solo se daban en grandes computadores, y en algunos casos mainframes. Pero la microinformática creció y con ella un gran abanico de aplicaciones para gestionar las bases de datos, este conjunto de aplicaciones terminaron desembocando en lo que hoy en día se conoce como SGBD.

Con lo que se consiguió que las bases de datos llegaran a la microinformática, y con ello a particulares o usuarios capaces de gestionar su propia base de datos, a programadores capaces de crear bases de datos, etc.

El diseño lógico de una base de datos consiste en un conjunto de información referente a personas, animales o cosas, que constituye interés para una empresa y que están organizados de forma coherente y sin redundancias.

Las operaciones que realiza un SGBD son

1. Operaciones que afectan a toda la DB
 - a. La creación
 - b. La estructuración
 - c. Consultas globales
2. Operaciones que afectan a datos en concreto
 - a. Actualizaciones: Añadir, Eliminar, Modificar
 - b. Consulta selectiva

Las funciones esenciales de un SGDB son

1. Función de descripción
2. Función de manipulación
3. Función de control

El SGBD incorpora como herramienta fundamental dos lenguajes, para la definición y la manipulación de los datos.

1.- **La función de descripción:** Es la que define o describe la estructuración de todos los elementos o datos que integrara la DB, las interrelaciones, las vistas lógicas del usuario, la integridad.

Esta función es realizada (programada) en un lenguaje DDL (lenguaje de definición de datos) propio del SGDB.

2.- **La función de manipulación:** Recogen a todas aquellas funciones necesarias para la manipulación e interacción de los datos de una DB. Estas operaciones se agrupan en:

1. Operaciones de consulta
 - a. Selectiva
 - b. Global
2. Operaciones de actualización
 - a. Añadir
 - b. Eliminar
 - c. Modificar

Esta función es realizada (programada) en un lenguaje DML (lenguaje de manipulación de datos), que facilita las funciones necesarias para realizar estas operaciones.

El lenguaje DML puede ser huésped, o anfitrión.

DML huésped: Este lenguaje incrusta en el lenguaje anfitrión ciertas operaciones, comandos o funciones, para la manipulación de los datos.

Necesita de un lenguaje anfitrión, sobre el cual apoyarse.

DML Anfitrión: Es un lenguaje principal a partir del cual se desarrolla la interactividad necesaria con la DB. No necesita de inclusiones de ningún lenguaje huésped.

3.- **Función de control:** Es la función que determinar el interface (elemento comunicacional) entre la DB y el personal. Esta interface ira dirigida principalmente a los usuarios que desean interactuar con la DB, y al administrador.

Esta función es llevada a cabo por los programadores y analistas. Tienen que realizar las aplicaciones necesarias para comunicar a los usuarios con la DB, y los programas que el administrador necesite para administrar mejor la DB. Deben poner especial atención a la seguridad y confidencialidad.

4.- **Teleproceso:** Comunicación a distancia. Mas que una función, seria una capacidad de un SGDB.

Lenguajes del SGDB

Los SGDB han de cumplir diferentes funciones por lo que es necesario disponer de diferentes tipos de lenguajes unos orientados hacia la función y otros lenguajes orientados a los diferentes tipos de usuarios. Así pues, podemos distinguir entre:

1. **Lenguajes de definición:** Lenguaje que crea o define la estructura de la DB.
2. **Lenguajes de manipulación:** Lenguaje que manipular los datos de la DB.
 - a. Aplicaciones formalizadas: Muy usuales (consultas, añadir datos, eliminarlos, etc).
 - b. Aplicaciones no formalizadas: Poco usuales (consultas muy puntuales, etc).

- El lenguaje de definición de la estructura lógica en el nivel de abstracción ha de ser un lenguaje anfitrión.
- El lenguaje de definición de la estructura externa, puede ser anfitrión o huésped.

La interacción entre el SGDB y el sistema operativo (SO)

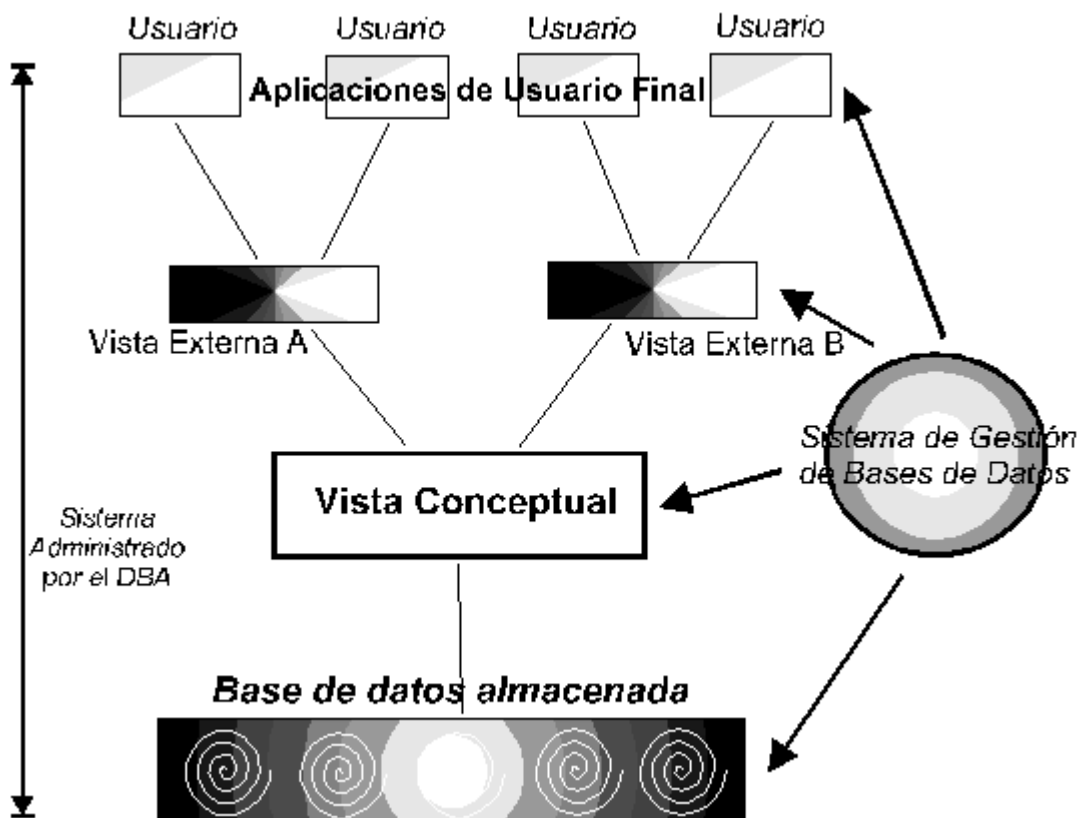
El SGDB esta muy relacionado con el SO, aunque cada SGDB, dependiendo de su diseño y de la plataforma (Linux, Windows, macOS, etc) en la que trabaje va a tener unas características propias y un modo de funcionamiento específico.

Se recomienda que se use un SGDB multiplataforma, esto es un SGDB que funcione en varios sistemas operativos, cuantos mas mejor. Y digo esto, porque imaginémosnos que la empresa decide migrar de windows a linux, pero que su SGDB solo funciona en windows. Deberíamos programar esa migración (tarea muy tediosa). Así que mas vale prevenir.

En la actualidad existen muchas empresas que están migrando a Linux, abandonando así SOs windows con tecnología NT. Una de las razones de esto, es el elevado coste de licencia del SO windows, mientras que por el contrario Linux es totalmente gratuito (sin contar manuales). Al fin y al cabo las empresas buscan rentabilidad, si deben elegir entre dos SO sean los que sean, a iguales calidades siempre se decantarán por el de menor precio (por lo normal y lógico).

Una ventaja que nos presenta el SGDB frente al sistema gestor de ficheros, es que este último, lanza trabajos al SO de forma independiente. Esto es, supongamos que tenemos un sistema gestor de ficheros, y los usuarios interactúan con la información mediante aplicaciones, lógicamente cada aplicación funciona de forma independiente a las demás, ahora imaginemos que la empresa tienen 100 trabajadores y que coincide que 74 de ellos lanzan trabajos concurrentemente (a la vez) al SO. Este es muy probable que se ralentice o que llegue a colgarse.

En el caso de las DB aunque los 74 usuarios lanzen trabajos estos van, desde las aplicaciones al SGDB y este administra y mantiene en cola a los trabajos, evitando así la saturación del SO. Incluso en algunos casos el SO incluye al SGDB como parte de él, como un módulo adjunto.



Los usuarios

Personal de una DB

El distinto personal, que usa o tienen algún tipo de relación con la DB se podría clasificar en dos grupos

1. Informáticos
2. Usuarios

Informáticos: Tienen como misión la creación, implantación y mantenimiento de la DB. También deben crear programas aparte dirigidos a los usuarios finales con la finalidad de hacerles más fácil su trabajo con la DB.

1.- Los informáticos

1. Administradores: Tienen la misión de administrar la DB, entendiendo por administrar a, gestionar los datos de la DB, dar el visto bueno al trabajo de los programadores y analistas, dirigir y coordinar todo el proceso de creación de la DB, vigilar y mantener la confidencialidad de los datos, etc.
2. Diseñadores: Son los encargados de realizar el diseño o estructura de la DB, en una primera aproximación. Dos clases de diseñadores.
 - a. Diseñadores lógicos: Deben determinar que tipos de datos y de que tamaño, han de estar contenidos en la DB.
 - i. Deben plasmar el punto de vista del conjunto de los usuarios en las estructuras lógicas más adecuadas.
 - ii. Para realizar este trabajo deben entrevistarse con los distintos futuros usuarios de la DB y recoger información relevante.
 - b. Diseñadores físicos: Deben obtener del diseño lógico de la DB, la mejor optimización posible. Deben buscar las formas o estructuras mejores para que la máquina tarde el menor tiempo posible en dar respuesta a una consulta de la DB. Esta tarea también puede realizarla el administrador.
3. Analistas y programadores: Realizan el sofá (programas y aplicaciones) necesario para que los usuarios puedan interactuar con la DB, sin tener conocimientos informáticos. Dirigiéndome a los programadores; como se suele decir, hemos de pensar que los usuarios van con guantes de boxeo al teclado. Con esto quiero decir, que si el programa pide que teclee el número 1, el personal tecleará el 2. Esto pasa y por ello debemos evitar y cerrar todos los caminos posibles a errores que puedan cometer estas personas. Lo mejor para esto, es ponernos en su lado, imaginarnos como fastidiar la aplicación. También podemos recurrir a pasar una beta (versión de prueba) de la aplicación y que una serie de betatesters (testean aplicaciones) nos comuniquen los posibles errores, de esta forma. Yo personalmente como programador sostengo la idea de que si tu has creado un programa y sabes como va el código interno del programa, inconscientemente funcionas con la aplicación de forma que no fallara, si se lo pasas a alguien que desconozca esa misma aplicación, obtendrás de forma más rápida los posibles errores que tu tardarías en ver o nunca llegarías a localizarlos.

2.- Usuarios

1. Habituales: Son usuarios que deben trabajar con la DB de forma periódica. A este tipo de usuarios en muchos casos, los programadores suelen darles pequeños cursillos de uso de la DB, para que los usuarios sepan desenvolverse mejor por el sistema, e incluso se les enseña a realizar pequeños trucos o caminos rápidos, aunque sea con comandos de difícil aprendizaje.
2. Esporádicos: Son los usuarios que debemos tomar como personas con guantes de boxeo en el ordenador. A este tipo de usuarios hay que presentarles las aplicaciones necesarias, sencillas y detalladas, de forma que no tenga cabida el error o confusión.