

# U.T. 2: Sistemas Gestores de Bases de Datos.

**I.E.S. ANTONIO SEQUEROS**

**ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN RED**

**GESTIÓN DE BASES DE DATOS**

**CURSO 2022/2023**

# Índice

1.INTRODUCCIÓN.....	1
2. BASES DE DATOS. COMPONENTES.....	2
2.1. DATOS.....	2
2.2. HARDWARE.....	2
2.3. SOFTWARE.....	3
2.4. USUARIOS.....	3
2.5. ADMINISTRADORES.....	5
3. OBJETIVOS DE UN SGBD.....	5
4. FUNCIONES DE UN SGBD.....	8
4.1. FUNCION DE DESCRIPCIÓN O DEFINICIÓN.....	8
4.2. FUNCIÓN DE MANIPULACIÓN.....	8
4.3. FUNCIÓN DE CONTROL.....	10
5. LENGUAJES DE LOS SGBD.....	11
5.1. LENGUAJE DE DEFINICIÓN DE DATOS (LDD).....	12
5.2. LENGUAJES PARA MANIPULACIÓN DE DATOS (LMD).....	13
6. INTERACCIÓN DEL USUARIO CON EL SGBD.....	16
7. FUNCIONAMIENTO DEL SGBD: INTERACCIÓN CON EL SISTEMA OPERAT...17	
8. LA ARQUITECTURA ANSI/X3/SPARC.....	20
9. LA ARQUITECTURA Y LA INDEPENDENCIA FÍSICO/LÓGICA.....	23
10. ESTRUCTURA GENERAL DE UN SGBD.....	25
11. MODELOS DE DATOS.....	26
11.1. MODELO JERÁRQUICO.....	26
11.2. MODELO EN RED.....	27
11.3. MODELO RELACIONAL.....	28
11.4. MODELO ORIENTADOS A OBJETOS.....	28
11.5. SGBD DISTRIBUIDOS.....	29

## Bibliografía

DE MIGUEL y PIATTINI (1999). *Fundamentos y Modelos de Bases de Datos*.  
Ed. Ra-Ma

DE MIGUEL y PIATTINI (1993). *Concepción y Diseño de Bases de Datos*. Ed. Ra-Ma

KORTH y SILBERSCHATZ (1998) *Fundamentos de Bases de Datos*. Ed. McGrawHill

# 1. INTRODUCCIÓN

Antes de centrarnos en los sistemas gestores de bases de datos, y como recordatorio de lo visto en la unidad de trabajo anterior, vamos a ver con un ejemplo la forma tradicional de gestionar y almacenar unos datos, y las novedades que incorpora una base de datos.

Supongamos una pequeña empresa que se dedica a realizar ventas por correo. Esta necesitará gestionar un mantenimiento de clientes, un control de almacén y un sistema de facturas. La forma tradicional de resolver este problema consiste en utilizar una serie de aplicaciones (programas) en los que cada aplicación dispone de su propio conjunto de ficheros, que contiene los datos necesarios, y que están organizados de acuerdo a la forma que tiene la aplicación de tratarlos. Estas organizaciones de los ficheros serían las ya conocidas: secuencial, directa o indexada.

Si se decide cambiar la estructura de alguno de estos ficheros, será necesario también cambiar la propia aplicación. A la inversa, si se tiene que cambiar la aplicación, casi con toda seguridad habrá que cambiar el número de ficheros, su organización, tipo de campos, etc.

Además, y dado que las aplicaciones son para la misma empresa, es seguro que existirá un amplio grado de redundancia (repetición) de los datos entre los distintos ficheros de las aplicaciones, por ejemplo, datos de clientes repetidos en varios ficheros.

A causa de este tipo de trabajo basada en ficheros, se denomina Sistema orientado al proceso.

Si se reemplazan todos estos ficheros por una única colección de datos apropiada, generalmente de gran tamaño, que sea accesible por todas las aplicaciones, que sea consistente y que presente una redundancia mínima, entonces la colección de datos se transforma en una base de datos (BD). En el caso de la empresa de venta por correo la BD estará constituida por una única colección de datos usada por las aplicaciones de clientes, almacén y facturas.

A esta forma de trabajo basada en la utilización de BD se le denomina Sistema orientado a los datos porque los datos están estructurados y almacenados de forma independiente a las aplicaciones que los pueden manejar.

## 2. BASES DE DATOS. COMPONENTES

### **Definición de Base de Datos:**

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

Una definición concisa de lo que es una BD sería la siguiente: *Una base de datos es una colección interrelacionada de datos, almacenados en un conjunto sin redundancias innecesarias, cuya finalidad es la de servir a una o más aplicaciones de la manera más eficiente.* Una base de datos consta de los siguientes componentes: datos, software, hardware, usuarios y administrador.

### **2.1. DATOS**

Evidentemente, el componente fundamental de una BD son los DATOS, que como decíamos en la definición, están interrelacionados entre sí formando un conjunto con un mínimo de redundancias.

Los datos se pueden definir como *“antecedentes necesarios para llegar al conocimiento exacto de una cosa o para deducir las consecuencias de un hecho”*. Los datos por sí mismos no aportan conocimiento. Hay que procesarlos y transformarlos. La información es el resultado de esta transformación.

Estos datos estarán almacenados sobre unos determinados soportes de almacenamiento secundario.

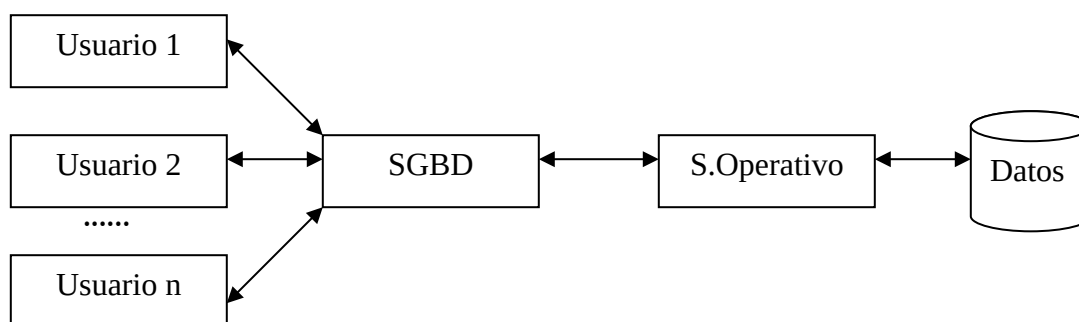
### **2.2. HARDWARE**

Los datos deben almacenarse en algún sitio, al igual que tienen que poder introducirse a través de algún dispositivo, y mostrarse, etc. en todas estas operaciones necesitamos elementos físicos (hardware) con los que interactuar y trabajar.

## 2.3. SOFTWARE

Para poder servir a varias aplicaciones o usuarios, los datos deben estar estructurados y almacenados de forma totalmente independiente a las aplicaciones que quieran utilizarlos: ¿Cómo se consigue esta independencia?. Pues se consigue utilizando un software o conjunto de programas que actúe como interfaz entre los datos y las aplicaciones. A este software se le denomina Sistema de Gestión de Base de Datos (SGBD) (también conocido con el acrónimo inglés DBMS)

El SGBD es el encargado tanto de crear y organizar la base de datos como de atender todas las solicitudes de acceso a la BD expresadas por las aplicaciones o los usuarios.



## 2.4. USUARIOS

Los distintos tipos de usuarios de una base de datos pueden clasificarse en usuarios informáticos y usuarios finales:

- ❖ **Usuarios informáticos:** Tienen a su cargo las tareas de creación y mantenimiento de la base de datos, así como la realización de los procedimientos y programas que necesitan los usuarios finales. Entre ellos se pueden distinguir:
  - **Diseñadores:** Tienen la responsabilidad de identificar los datos que han de estar contenidos en la base de datos, de acuerdo con las necesidades que les manifiesten los usuarios, así como de determinar las estructuras más apropiadas para conseguir satisfacer estas necesidades. Según la fase en que intervengan los diseñadores, es posible distinguir:
    - **Diseñadores lógicos:** Deben ser capaces de determinar qué tipos de datos han de estar contenidos en la base y conseguir plasmar el punto de vista del conjunto de usuarios en las estructuras lógicas más adecuadas para éstos. Para ello, deben mantener constantes entrevistas con los usuarios (incluida la dirección) a fin de que la base de datos represente lo más fielmente posible el mundo real que se trata de recoger. Es decir, los diseñadores lógicos deben perseguir un objetivo de eficacia de la base de datos.

- **Diseñadores físicos:** Su objetivo es transformar las estructuras lógicas en estructuras físicas que proporcionen la mayor eficiencia de cara a la máquina, minimizando el tiempo de respuesta y el consumo de recursos, es decir, optimizar la ratio coste / beneficio. Muchas veces esta función es asumida por el administrador.
- **Administradores:** Al ser un elemento fundamental en la gestión de la base de datos se verá con detenimiento en el siguiente subapartado 2.5.
- **Analistas y Programadores:** Tienen a su cargo el análisis y la programación de las tareas que no pueden ser llevadas a cabo por los usuarios finales, para lo cual han de desarrollar distintos procedimientos y programas que ponen a disposición de los usuarios finales a fin de facilitarles su trabajo.
- ❖ **Usuarios finales:** Son aquellos que tienen que acceder a los datos porque los necesitan para llevar a cabo su actividad. A diferencia de los usuarios informáticos, su interés suele estar centrado en el contenido de la base de datos, es decir, en los datos. Existen también distintas clases de usuarios finales:
  - **Habituales:** Suelen hacer consultas y/o actualizaciones en la base de datos como parte habitual de su trabajo. Utilizan en general menús previamente preparados por analistas y/o programadores, de forma que se facilite su interrelación con el ordenador. En otros casos pueden usar lenguajes sencillos para el acceso a la base de datos, así como paquetes de programas.
  - **Operadores de Entrada:** Su labor consiste en actualizar la base de datos para lo que se les preparan menús de actualización, aunque con la diferencia respecto a otros usuarios habituales de que tienen exigencias muy estrictas respecto a los tiempos de respuesta, además de no estar interesados en la información contenida en la base de datos.
  - **Esporádicos:** Es un tipo de usuarios muy parecido al anterior en la medida en que necesitan el ordenador a fin de que les preste una ayuda en su trabajo, pero en cambio no lo utilizan habitualmente porque el tipo de actividad que realizan no lo exige. Son, por tanto, usuarios a los que se les tiene que suministrar herramientas sencillas y, en general, potentes, ya que en bastantes casos así lo exige la clase de tareas que han de llevar a cabo; tareas que, en muchas ocasiones, no son fácilmente formalizables, por lo que no pueden ser atendidas por medio de menús.

En resumen, se puede decir que la finalidad del SGBD es establecer las adecuadas interfaces entre los diferentes tipos de usuarios y la base de datos.

## 2.5. ADMINISTRADORES

El uso compartido de recursos, propio de una base de datos, obliga a la existencia de un administrador, cuya misión es la vigilancia y gestión de los datos. El principal recurso en una base de datos son los datos, y el administrador debe velar para que éstos no se destruyan ni se contaminen, perdiendo su confidencialidad, disponibilidad e integridad. Por ellos, el administrador debe impedir consultas o actualizaciones no autorizadas y proteger la base de datos contra fallos del equipo lógico o físico. Será el responsable, por tanto, de establecer el sistema de autorizaciones de acceso y deberá coordinar y controlar su uso.

También el administrador tendrá a su cargo la gestión de otros recursos distintos de los datos, como pueden ser el SGBD y otras herramientas relacionadas con el mismo. Deberá ocuparse del buen funcionamiento de todo el sistema, sin que se produzcan paradas y de modo que se proporcionen los tiempos adecuados de respuesta.

En muchas organizaciones es la misma persona o grupo de personas las que tienen a su cargo las funciones de diseño (ver subapartado anterior) y de administración. Luego, si fundiéramos las funciones de diseño y de administración en la figura del Administrador, podríamos decir que es el encargado de controlar y manejar la Base de Datos, teniendo las siguientes funciones específicas:

- Describir el esquema de la BD, los campos, registros y relaciones entre ellos.

- Definir la estructura física de almacenamiento y el método de acceso.

- Modificar el esquema y la organización física.

- Conceder autorizaciones para el acceso a los datos a los usuarios o las aplicaciones.

- Especificar las reglas de integridad de los datos.

## 3. OBJETIVOS DE UN SGBD

Una vez que tenemos una visión general de lo que es un SGBD, vamos a ver los objetivos que un sistema debe cumplir para considerarse un SGBD:

- ❖ **Abstracción de la Información:** El primer objetivo de un SGBD es proporcionar a los usuarios una visión abstracta de la información, es decir, el sistema ahorra al usuario la necesidad de conocer los detalles de cómo se almacenan los datos así como de su mantenimiento. Para ocultar estos detalles, se definen varios niveles de abstracción tal y como se verá más adelante.

- ❖ **Independencia:** La independencia de los datos se puede definir como la capacidad para modificar un esquema de definición sin afectar a los programas de aplicación. Existen dos niveles:
  - **Independencia física:** Cuando es posible modificar el esquema interno sin que afecte a las aplicaciones. Se realizan para mejorar el rendimiento.
  - **Independencia lógica:** Cuando es posible modificar el esquema conceptual sin obligar a escribir de nuevo las aplicaciones. Se realizan cuando cambia la estructura lógica de la BD (tamaño registros, características de los campos, etc.).

La independencia lógica es más difícil de lograr que la física, ya que las aplicaciones dependen en gran medida de la estructura lógica de los datos.
- ❖ **Redundancia Mínima:** La utilización de BD supone evitar la repetición o redundancia de los datos en múltiples ficheros. En principio, puede parecer que lo más conveniente es una redundancia nula, pero, en la práctica, ésta comprobado que el mantener duplicados ciertos datos puede resultar beneficioso a efectos de búsquedas más rápidas, por lo que se prefiere que exista una redundancia mínima.
- ❖ **Consistencia:** Es consecuencia del punto anterior. Si existen datos duplicados en varios ficheros, y se realiza una actualización de esos datos, será necesario que el SGBD garantice la adecuada actualización del dato en todos los ficheros, para evitar la inconsistencia del dato.
- ❖ **Seguridad:** Otro objetivo a conseguir por un SGBD es la protección de los datos frente al acceso accidental o intencionado por parte de individuos no autorizados. Para establecer esta seguridad se suele recurrir a establecer claves de seguridad (passwords) necesarias para poder acceder a la BD.
- ❖ **Integridad:** Se refiere a las medidas necesarias para conservar la corrección de los datos en la BD, es decir, mantener su validez y estructura física. Existen varias circunstancias que pueden hacer que se estropeen los datos:
  - **Fallos del Hardware.**
  - **Actualizaciones incompletas.**
  - **Inserción de datos no válidos:** El Administrador de la base de datos define unas *reglas de integridad* (por ejemplo que el saldo de una cuenta no sea inferior a una cantidad) y el SGBD se debe de encargar de hacerlas cumplir al insertar un dato.

Una **regla de integridad** está constituida por tres componentes:

  - ✓ La restricción propiamente dicha
  - ✓ La respuesta a la violación, que especifica las acciones a tomar, como rechazar la operación, informar al usuario, etc.
  - ✓ La condición de disparo, que especifica cuando debe desencadenarse la acción especificada: antes, después o durante cierto evento.



- ❖ **Respaldo y Recuperación:** El SGBD debe proporcionar mecanismos eficientes para conservar **Copias de Seguridad** de cada fichero en prevención de posibles fallos.

El proceso de copiar un fichero de forma periódica se llama *respaldo (BACK-UP)*. Estas copias de seguridad deben de realizarse regularmente y guardarse en un lugar seguro. Se realizan sobre cintas o discos.

El proceso contrario, recuperar la información original a partir de las copias de seguridad se llama *recuperación*. Para poder realizar estas recuperaciones con la máxima rapidez y eficacia se recurre a un fichero especial llamado Bitácora o Diario (archivo log). En este fichero (de movimientos) se registran todos los datos que se vayan modificando debido a las operaciones con la BD. Se guarda tanto el valor que tenía el dato antes de ser modificado como el valor después de la modificación.

- ❖ **Control de la Concurrencia:** Lo más habitual es que la BD trabaje en un entorno de multiprogramación y multiusuario. Esto presenta el problema de que dos o más procesos (transacciones) distintos accedan al mismo dato de forma simultánea.

El SGBD debe de controlar este problema para evitar la inconsistencia de los datos. La solución consiste en utilizar una técnica denominada "*candado*":

- Un proceso sólo puede acceder a un dato si tiene un candado para ese dato. Existen dos tipos de candados:
  - Compartido: si un proceso obtiene un candado compartido en un dato, entonces puede leer ese dato pero no escribir en él.
  - Exclusivo: si un proceso obtiene un candado exclusivo en un dato, entonces puede leer y escribir ese dato.
- Las transacciones que deseen acceder a los datos deben solicitar un candado y si no lo obtienen pasarán a una cola de espera.

El módulo encargado de los accesos concurrentes a los datos, es decir, de proporcionar los candados, se llama Scheduler. También se encarga de definir la granularidad, es decir, el tamaño de la unidad a la que se concede el acceso concurrente. Esta puede variar considerablemente. En un Sistema Operativo el tamaño mínimo suele ser el fichero, pero en un SGBD se puede establecer el acceso a nivel de registro, o incluso de campo.

Algunas implementaciones de SGBD actuales (por ejemplo Interbase de Borland-Inprise-Corel) utilizan otro mecanismo para el control de la concurrencia consistente en mantener temporalmente varias versiones de los mismos datos cuando éstos sufren procesos de actualización.

- ❖ **Tiempo de Respuesta:** Las bases de datos están diseñadas para ser utilizadas por los usuarios finales, por lo que deben de asegurar un tiempo de respuesta idóneo a las peticiones de los usuarios. Cuando se trata de procesos por lotes, el tiempo de respuesta no tiene importancia, pero cuando se trata de procesos on-line críticos (reserva de plazas, terminal bancaria) el SGBD debe de proporcionar un tiempo de respuesta suficientemente corto.

## **4. FUNCIONES DE UN SGBD**

Las funciones esenciales de un SGBD son las de descripción, manipulación y control (o utilización).

### **4.1. FUNCION DE DESCRIPCIÓN O DEFINICIÓN**

La función de definición (también llamada de descripción) debe permitir al diseñador de la base especificar los elementos de datos que la integran, su estructura y las relaciones que existen entre ellos, las reglas de integridad semántica, etc., así como las características de tipo físico y las vistas lógicas de los usuarios.

Esta función, realizada por el lenguaje de descripción o definición de datos (LDD, aunque también se puede encontrar como DDL- Data Description Language-) propio de cada SGBD, debe suministrar los medios para definir las tres estructuras de datos (externa, conceptual e interna), especificando las características de los datos a cada uno de estos niveles.

A nivel interno, se ha de indicar el espacio (volúmenes, cilindros y pistas) reservado para la base, la longitud de los campos o elementos de datos, su modo de representación (binario, decimal, alfanumérico, punto fijo o flotante, etc.). Además, se deben poder definir caminos de acceso, como punteros, índices, etc.

Para las estructuras externa y conceptual, la función de descripción ha de proporcionar los instrumentos para la definición de los objetos (entidades, tablas, registros, etc.), así como su identificación, atributos de los mismos, interrelaciones entre ellos, autorizaciones de acceso, restricciones de integridad, etc. Las descripciones de las estructuras externas de los usuarios han de estar referidas a la estructura conceptual. El SGBD, además de suministrar facilidades de descripción, se ocupará de la función de correspondencia o transformación (mapping) de la estructura conceptual a las estructuras externas, y entre aquella y la estructura física.

### **4.2. FUNCIÓN DE MANIPULACIÓN.**

Una vez descrita la base de datos, es preciso cargar los datos en las estructuras previamente creadas, con lo que la base de datos estará ya dispuesta para su utilización. Los usuarios tendrán necesidad de recuperar la información (consultar la base de datos), o bien de actualizarla porque se hayan producido cambios en los datos.

La **consulta** a la base de datos puede ser de dos tipos:

- ✓ **Totalidad de los datos:** en la que se recuperan todos los datos de la base de datos o todos los de un determinado tipo; por ejemplo, para la confección de la nómina será preciso recuperar todos los registros de los empleados de la empresa.
- ✓ **Consulta selectiva:** en la que se tendrán que localizar los registros que cumplan una determinada condición (**criterio de selección**); por ejemplo, obtener los empleados que sean informáticos y sepan inglés.

En ambos casos será preciso especificar la estructura lógica externa que se desea recuperar; así, nombre, datos bancarios y salario en el ejemplo de la nómina; nombre, departamento y categoría profesional en el ejemplo de la consulta selectiva. El SGBD deberá, con estos datos, acceder a la estructura física de la base de datos donde se encuentran almacenados los datos, localizar aquellos registros indicados y ponerlos a disposición del usuario.

La **actualización** o puesta al día de una base de datos supondrá tres tipos de operaciones distintas:

- ✓ **Insertión:** cuando aparezcan nuevos elementos; por ejemplo, en un fichero de personal es preciso dar de alta a los nuevos empleados.
- ✓ **Borrado:** porque hayan desaparecido algunos elementos; por ejemplo en el fichero de personal es preciso dar de baja a los empleados que ya no están en la empresa.
- ✓ **Modificación de los datos:** de aquellos registros en los cuales se hayan producido cambios; por ejemplo, cuando se ha alterado la categoría profesional de un empleado.

La función de manipulación permite a los usuarios de la base, informáticos o no, buscar, añadir, suprimir o modificar los datos de la misma, siempre de acuerdo con las especificaciones y normas de seguridad dictadas por el administrador.

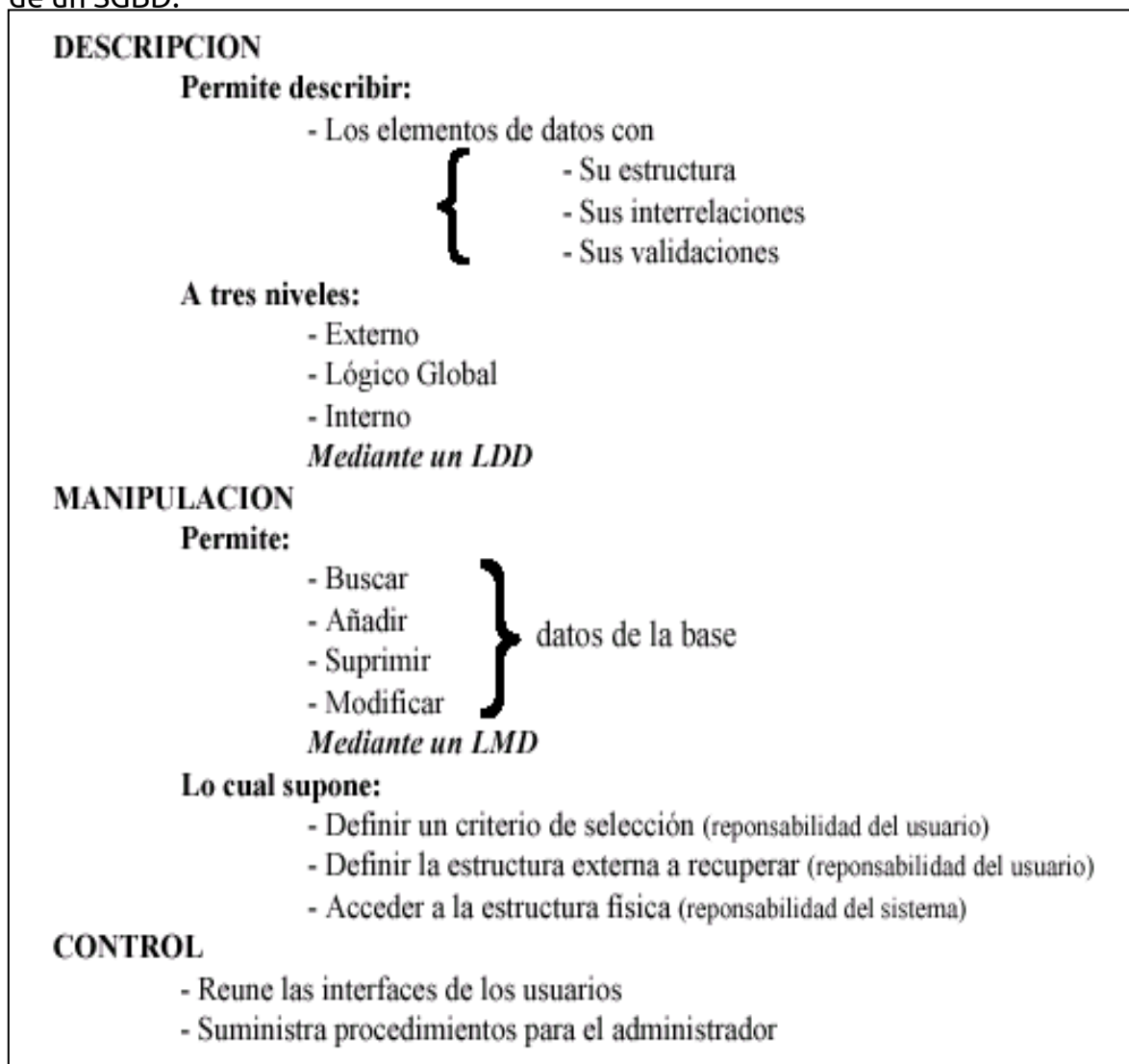
La función de manipulación se llevará a cabo por medio de un lenguaje de manipulación de datos (LMD, aunque también se puede encontrar como DML-Data Manipulation Language-) que facilita los instrumentos necesarios para la realización de estas tareas. Muchas veces se trata de un conjunto de comandos (**lenguaje huésped**) que se escriben en un lenguaje de programación (**lenguaje anfitrión**); mientras que otras veces se trata de un **lenguaje autocontenido** que no precisa apoyarse en ningún otro lenguaje, ya que dispone en sí mismo del conjunto de instrucciones necesarias para llevar a cabo tanto la recuperación como la actualización de los datos. La mayoría de los SGBD actuales tienen ambos tipos de lenguajes, huéspedes y autocontenidos; estos últimos, orientados a los usuarios no informáticos, suelen usarse de forma interactiva.

### 4.3. FUNCIÓN DE CONTROL

Esta función reúne todas las interfaces que necesitan los diferentes usuarios para comunicarse con la base y proporciona un conjunto de procedimientos para el administrador.

Las exigencias respecto a la forma de utilizar la base de datos son muy diferentes, según los tipos de procesos y según los usuarios, siendo preciso que la función de control (o utilización) responda a todas ellas. En especial, esta función debe integrar una serie de instrumentos que faciliten las tareas del administrador. En la mayoría de los SGBD existen funciones de servicio, como cambiar la capacidad de los ficheros, obtener estadísticas de utilización, cargar archivos, etc. y principalmente las relacionadas con la seguridad física (copias de seguridad, re arranque en caso de caída del sistema, etc.) y de protección frente a accesos no autorizados.

En la figura siguiente se muestra un cuadro con las funciones esenciales de un SGBD.



## 5. LENGUAJES DE LOS SGBD

Las distintas funciones que ha de cumplir un SGBD hacen necesario disponer de diferentes tipos de lenguajes y procedimientos que permitan la comunicación con la base de datos; unos están orientados hacia la función (definición o manipulación) y otros dirigidos a diferentes tipos de usuarios o de aplicaciones. Si lo vemos esquemáticamente, tenemos los siguientes tipos de lenguajes:

### ❖ **Por tipo de función:**

- Definición
- Manipulación

### ❖ **Por tipos de usuarios y de aplicaciones**

- Informáticos
- Finales
  - Aplicaciones formalizables (como la gestión de personal).
  - Aplicaciones no formalizables (como los procesos de toma de decisiones).

Cuando se trata de procesos formalizables y muy repetitivos, el programador se encarga en general de escribir los correspondientes programas, que muchas veces son sometidos a un tratamiento por lotes, con periodicidad fija (emisión diaria de recibos, obtención mensual de la nómina, etc.) o a un tratamiento interactivo (consultas).

Por el contrario, si el proceso es difícilmente formalizable o no es lo suficientemente repetitivo, escribir un programa puede no resultar rentable, y suele ser más conveniente que el mismo usuario final resuelva directamente la consulta mediante los instrumentos que el SGBD pone a su alcance.

También el SGBD debe proporcionar al administrador de la base de datos un conjunto de procedimientos que le faciliten su tarea.

Los lenguajes de los SGBD habrán de tener, por tanto, un enfoque distinto según el tipo de usuario, ya que las exigencias de un programador o del administrador no tienen nada que ver con lo que puede necesitar un usuario no informático.

En general, los usuarios informáticos (diseñador, administrador, analistas, programadores, etc.) requerirán medios potentes y flexibles mediante los cuales consigan definir, administrar, extraer o manipular los datos de la base. En muchos casos desearán apoyarse en el lenguaje de programación que están habituados a manejar (*lenguaje anfitrión*), para lo cual deberá permitir hacer llamadas desde un programa de aplicación al SGBD. El conjunto de sentencias de manipulación del SGBD que pueden ser llamadas desde un lenguaje de programación, permitiendo así el acceso a la base de datos, se suele denominar *sublenguaje de datos* y también *lenguaje huésped* o *lenguaje embebido*.

El usuario final requerirá medios simples para comunicarse con la base, lo que se puede conseguir mediante un lenguaje de manipulación *autocontenido* que tenga una sintaxis sencilla, pero lo suficientemente potente como para soportar demandas de información muy variadas, o por medio de tratamientos parametrizados que suelen presentarse al usuario en forma de menús.

### **5.1. LENGUAJE DE DEFINICIÓN DE DATOS (LDD)**

El administrador de la base de datos ha de disponer de instrumentos que le permitan describir los datos con facilidad y precisión, especificando sus distintas estructuras; es lo que se denomina lenguaje de definición de datos. Los lenguajes de definición de datos son autocontenidos y no necesitan apoyarse en ningún lenguaje de programación.

El SGBD tendrá que facilitar medios para describir la estructura conceptual (lógica global), para hacer las especificaciones relativas a la estructura interna y para declarar las estructuras externas que sean requeridas para el desarrollo de las diferentes aplicaciones.

#### **Lenguajes de definición de la estructura lógica global**

Desde el punto de vista conceptual, será necesario que el administrador disponga de un instrumento de descripción que le permita asignar nombres a los campos, a los agregados de datos, a los registros, etc., estableciendo sus longitudes y características, así como las relaciones entre estos elementos; especificar los identificadores, e indicar las restricciones semánticas que se han de aplicar a los diferentes objetos descritos.

#### **Lenguajes para la definición de la estructura interna**

En un SGBD en el cual fuesen totalmente independientes las estructuras físicas y conceptual (lógica global), y que consiguiese automáticamente la optimización en el almacenamiento y recuperación de los datos, el SGBD podría encargarse de, a partir de la estructura conceptual, definir la estructura interna adecuada sin intervención del usuario (administrador), para lo cual habría que suministrar al SGBD las informaciones precisas, como volúmenes, crecimiento previsto, tipos de registros más accedidos con indicaciones sobre número medio de accesos, relación entre actualizaciones y consultas, etc. Sin embargo, en la práctica, y en el estado actual de la técnica, se puede mejorar considerablemente el rendimiento de la base si el administrador especifica características respecto a la estructura física que tiendan a conseguir una máxima eficiencia en el almacenamiento y en la recuperación de los datos de la base. Para ello, tendrá que disponer de un lenguaje de definición de la estructura interna o, simplemente, deberá dar valores a ciertos parámetros.

## **Lenguaje de definición de estructuras-esquemas- externas**

El SGBD debe poner a disposición de los usuarios medios que les permitan recuperar o actualizar los datos contenidos en la base de acuerdo con la visión lógica o estructura externa (vista) que precise cada aplicación.

Al definir una estructura externa es preciso darle un nombre e indicar qué datos y qué interrelaciones de la estructura conceptual (lógica global) se encontrarán en la misma. Cuando se desee utilizar un esquema externo ya definido se podrá hacer referencia al mismo invocando su nombre desde el lenguaje de manipulación.

## **5.2. LENGUAJES PARA MANIPULACIÓN DE DATOS (LMD)**

Para cumplir los objetivos asignados a la función de manipulación se ha de contar con lenguajes que ofrezcan a los usuarios la posibilidad de referirse a determinados conjuntos de datos, que cumplan ciertas condiciones (*criterio de selección*), como que un atributo tenga un determinado valor, o que un conjunto de atributos y valores satisfagan cierta expresión lógica. Además del criterio de selección, es preciso indicar la estructura externa que se desea actualizar o recuperar.

Una vez especificados el criterio de selección y los datos a actualizar o recuperar, el SGBD debe ocuparse de acceder al correspondiente soporte físico de donde se extraerán los datos definidos para su transferencia a un dispositivo de salida, o en donde se insertarán, modificarán o borrarán los datos si se trata de una actualización.

En la figura (siguiente página) se presenta la estructura de un programa escrito en un lenguaje anfitrión que interactúa con una base de datos mediante un lenguaje huésped. En la parte declarativa se hace una llamada a la vista de usuario; además, se definirán, dentro del programa de aplicación, una áreas de entrada / salida y un área al que se pasarán los mensajes enviados por el SGBD para informar sobre distintos aspectos relativos a la ejecución de una sentencia (si se ha producido algún error durante la ejecución, si ésta ha terminado normalmente, etc.). En la parte de proceso se hace una llamada a la base de datos (mediante las sentencias de recuperación o actualización del LMD), la cual, si se trata de una sentencia de recuperación, copia los datos desde la base de datos al área de entrada del programa de aplicación; si la sentencia es de inserción, los datos pasan del área de salida del programa a la base de datos; cuando la sentencia es de modificación, los datos se transfieren al área de entrada/salida del programa donde se realizan los cambios y desde allí se devuelven a la base de datos; en las operaciones de borrado no existe transferencia de datos, sino que éstos son eliminados de la base de datos. En todos los casos habrá que comprobar el contenido del área de mensajes, a fin de saber si la correspondiente sentencia ha finalizado o no con éxito.

## Un LMD huésped dentro de otro lenguaje anfitrión

### ..... SENTENCIAS DECLARATIVAS

.....  
declaración de áreas de E/S para las transferencias de datos desde/hacia la BD  
declaración de áreas para la comunicación de mensajes

.....  
LLAMADA A LA VISTA DE USUARIO (interacción con la BD)

### ..... SENTENCIAS DE PROCESO

.....  
LLAMADA A LA SENTENCIA DEL LMD (interacción con la BD)  
comprobación del contenido del área de mensajes

Pero al igual que el programador precisa de un lenguaje de manipulación que se embeba en un lenguaje de programación, el usuario no informático deberá disponer también de un instrumento análogo, pero mucho más sencillo, para comunicarse con la base y extraer de ella (o introducir) las informaciones que precise. Para atender a estas necesidades, los SGBD suelen contar con lenguajes autocontenidos que ofrecen facilidades a los usuarios con pocos conocimientos de programación, para, desde un terminal y en modo interactivo, acceder a la base y manipular los datos almacenados en ella sin necesidad de apoyarse en un lenguaje de programación.

Los lenguajes autocontenidos suelen incluir no sólo los medios de manipulación, sino también facilidades para describir los datos que se desean recuperar o actualizar.

A veces, el mismo lenguaje de manipulación puede actuar como huésped y como autocontenido, así ocurre, por ejemplo con el SQL, que puede ser llamado desde un lenguaje anfitrión (por ejemplo C), o bien puede interactuar directamente con la base de datos sin necesidad de estar apoyado en ningún lenguaje de programación. Esta característica aumenta la productividad de los sistemas que dispongan de esta clase de lenguajes duales.

Los LMD admiten, además, otro tipo de clasificaciones que es interesante comentar. Un LMD puede ser *procedimental* o *no procedimental*. Un lenguaje es más procedimental cuanto con más detalle sea preciso especificar el procedimiento necesario para acceder a la base de datos a fin de recuperar o actualizar los datos. Los lenguajes orientados al usuario deben ser poco procedimentales, mientras que los lenguajes que utilizan los programadores suelen ser bastante más procedimentales. En un lenguaje poco procedimental basta con decir *qué* se quiere, sin explicar *cómo* obtenerlo; mientras que si el



lenguaje es más procedimental no es suficiente con que se le indique el *qué*, sino que es necesario, además, precisar el algoritmo.

Aunque algunos LMD se utilizan en diferido (tratamiento por lotes), en la actualidad, la mayoría de los LMD permiten su uso en modo conversacional desde un terminal.

Existen lenguajes de manipulación de datos llamados *navegacionales* que recuperan o actualizan los datos registro a registro, y es el programador quien debe indicar el camino que se ha de recorrer, a través de la estructura definida, hasta llegar al registro buscado; en este caso, cada sentencia o llamada al lenguaje de manipulación de datos produce la recuperación o actualización de un registro único (tal es el caso de los lenguajes Codasyl o Jerárquicos). En cambio otros lenguajes actúan sobre conjuntos de registros (lenguajes de *especificación*), de forma que una única sentencia puede dar lugar a la recuperación actualización del conjunto de registros que cumpla el criterio de selección especificado (como ocurre en los lenguajes relacionales, por ejemplo en el SQL).

En general, los lenguajes de tipo huésped son procedimentales, se explotan en diferido y actúan registro a registro, mientras que los autocontenidos suelen ser no procedimentales, se usan en conversacional y recuperan conjuntos de registros. Por ejemplo, el DL/I (IMS de IBM) es de tipo huésped, muy procedimental, diferido y navegacional; el lenguaje SQL, es poco procedimental, de especificación (actúa sobre conjunto de registros) y puede usarse en modo autocontenido o como huésped desde un lenguaje anfitrión, ya que goza de la propiedad dual. Un ejemplo de sentencia SQL es:

```
SELECT nombre,apellido  
FROM persona  
WHERE fecha_nac="11/09/01"
```

En la actualidad, la mayoría de los SGBD disponen además de lenguajes de cuarta generación (4GL), en general muy poco procedimentales, que permiten la consulta y actualización de la base de datos, para desarrollar aplicaciones (informes, gráficos) de una forma rápida, intuitiva y fácil. Un ejemplo serían las herramientas SQL Forms, SQL Reports, SQL/PL, etc. de Oracle.

La figura (siguiente página) presenta la clasificación de los lenguajes de datos que acabamos de ver:

## Caracterización de los lenguajes de datos

{ Huésped  
Autocontenido

{ Muy Procedimental  
Poco procedimental

{ Diferido (por lotes)  
Conversacional (interactivo)

{ Registro a registro (navegacional)  
Conjunto de registros (especificación)

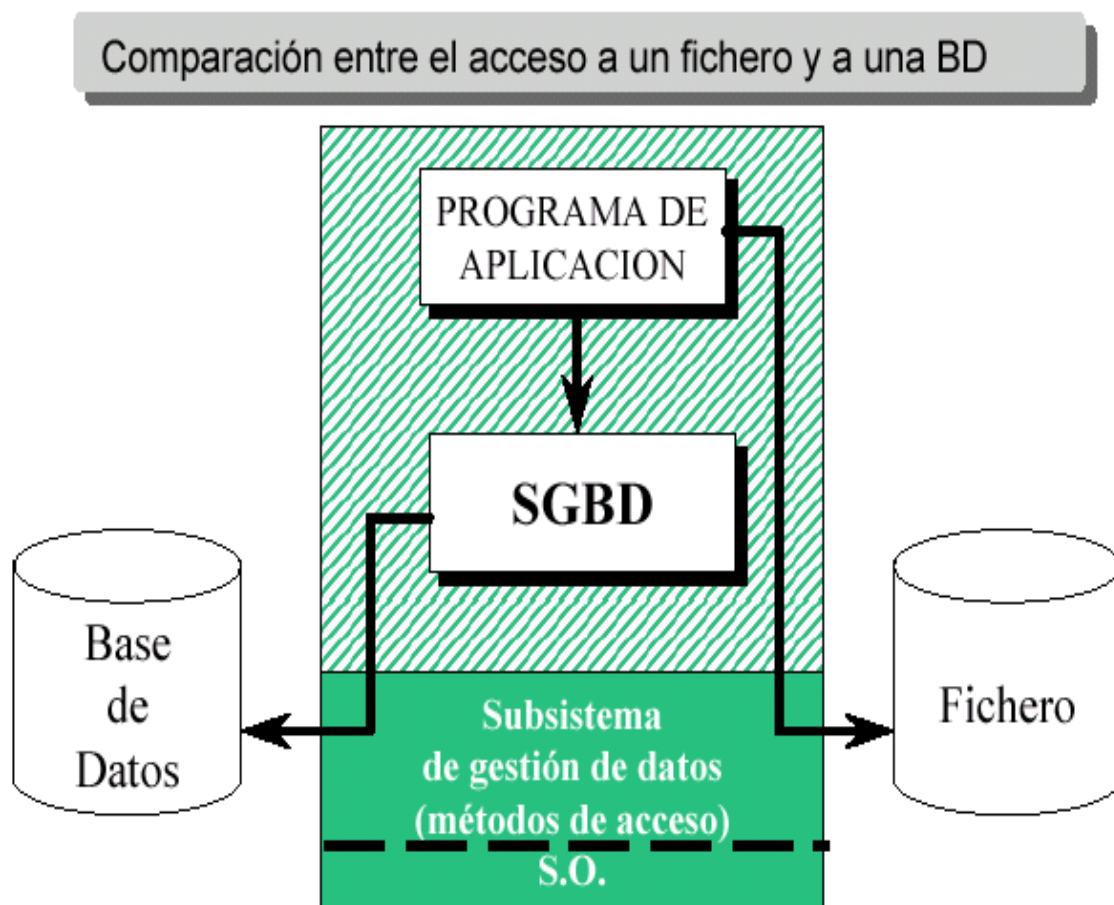
## 6. INTERACCIÓN DEL USUARIO CON EL SGBD

Funciones		DESCRIPCION			MANIPULACION
Usuarios		INTERNA	LOGICA GLOBAL	EXTERNA	
	ADMINISTRADOR	LENGUAJE DE DESCRIPCION DE DATOS INTERNOS	LENGUAJE DE DESCRIPCION DE DATOS GLOBALES	LENGUAJE DE DESCRIPCION DE VISTAS EXTERNAS	PROCEDIMIENTOS: - CREACION - OPTIMIZACION - REORGANIZACION - RECUPERACION - COPIAS SEGURIDAD .....
NO INFORMÁTICOS	INFORMÁTICO			LLAMADA A UNA VISTA DEFINIDA POR EL ADMINISTRADOR	- LMD (EMBEBIDO) - GENERADOR DE INFORMES - AYUDAS - L4G .....
	FORMALIZABLE			VISTA EXTERNA DESCRITA EN EL PROPIO PROCEDIMIENTO	- USO DE APLICACIONES - L4G - LENGUAJES PARAMETRICOS PAQUETES
	NO FORMALIZABLE			FACILIDADES INCLUIDAS EN LOS LENGUAJES O EN LOS PAQUETES	- LENGUAJES AUTOCONTENIDOS - PAQUETES

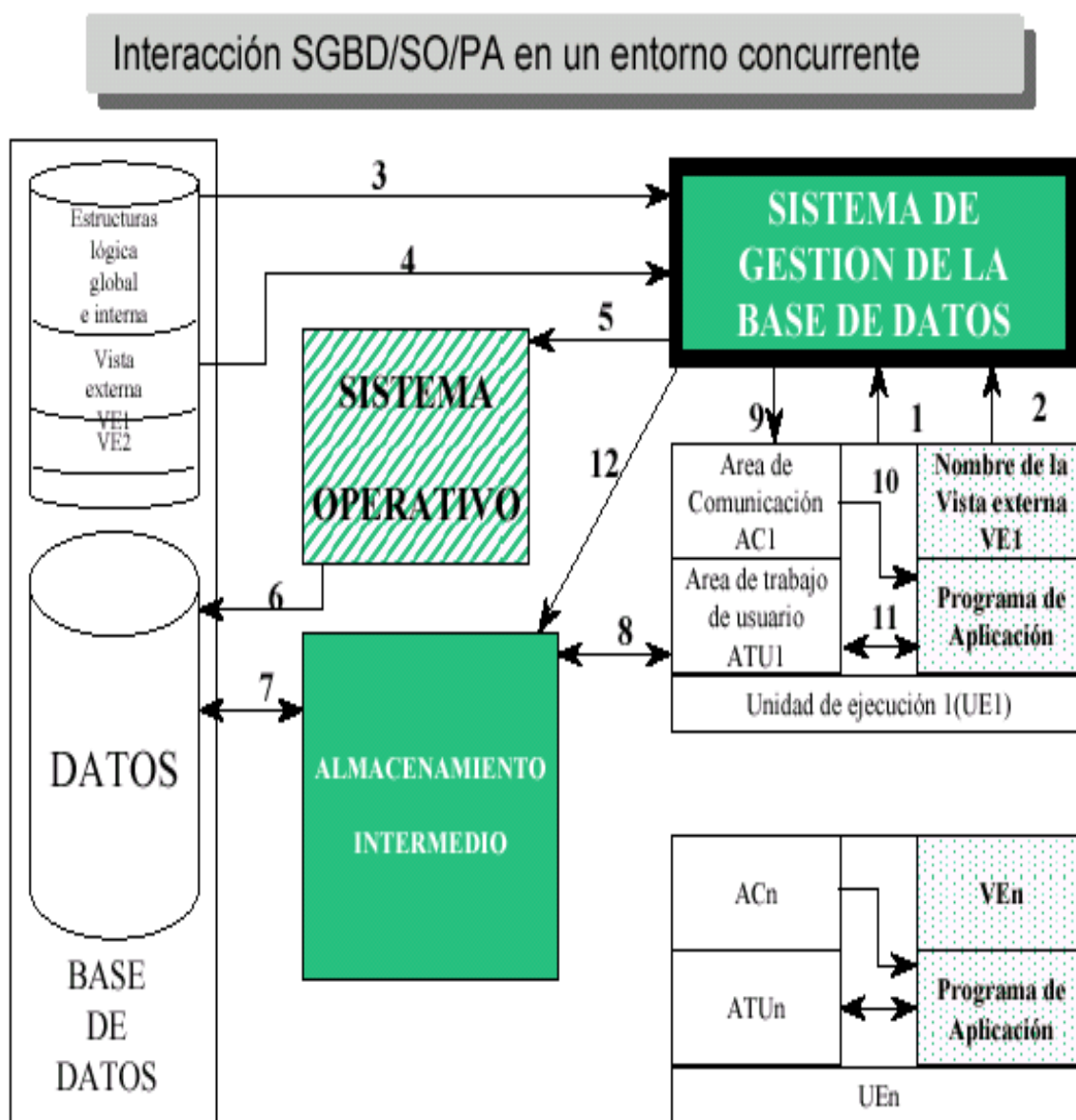
## 7. FUNCIONAMIENTO DEL SGBD: INTERACCIÓN CON EL SISTEMA OPERATIVO.

Si se contempla desde la perspectiva del sistema informático, el SGBD constituye un subsistema de éste y, más en particular, del software. El funcionamiento del SGBD está muy interrelacionado con el de otros componentes, especialmente con el Sistema Operativo.

Si comparamos la forma en la que los programas de aplicación interaccionan con los datos en un sistema de ficheros y en una base de datos, encontramos bastantes diferencias. En el primer caso los datos se organizan en ficheros y las aplicaciones acceden a los ficheros con datos a través del sistema operativo mientras que en las bases de datos, los datos se organizan en conjuntos estructurados, sin redundancias perjudiciales, que pretenden ser una representación del mundo real y que son utilizados por todas las aplicaciones. Las aplicaciones se desarrollan mediante las facilidades del SGBD, y éste, a su vez, se apoya en los métodos de acceso del S.O. La comparación de formas de acceso se ve más claramente en la figura siguiente:



La interacción en un entorno concurrente entre el SGBD, el sistema operativo y los programas de aplicación, se muestra de forma general en la figura siguiente. Por cada programa de aplicación (PA) que se está ejecutando existe una unidad de ejecución (UE) en la que se encuentra un área de trabajo de usuario (ATU) con sus áreas de entrada / salida (E/S) y un área de comunicación con el SGBD destinada a recibir los mensajes y la información de control procedente de éste. Desde el programa de aplicación se hace referencia a la vista externa (VE) utilizada por el correspondiente programa. En la biblioteca del sistema se encuentran almacenados, además de los datos, la estructura conceptual (lógica global) y la estructura interna que los describe, así como las vistas externas que serán llamadas por los programas de aplicación de los usuarios.

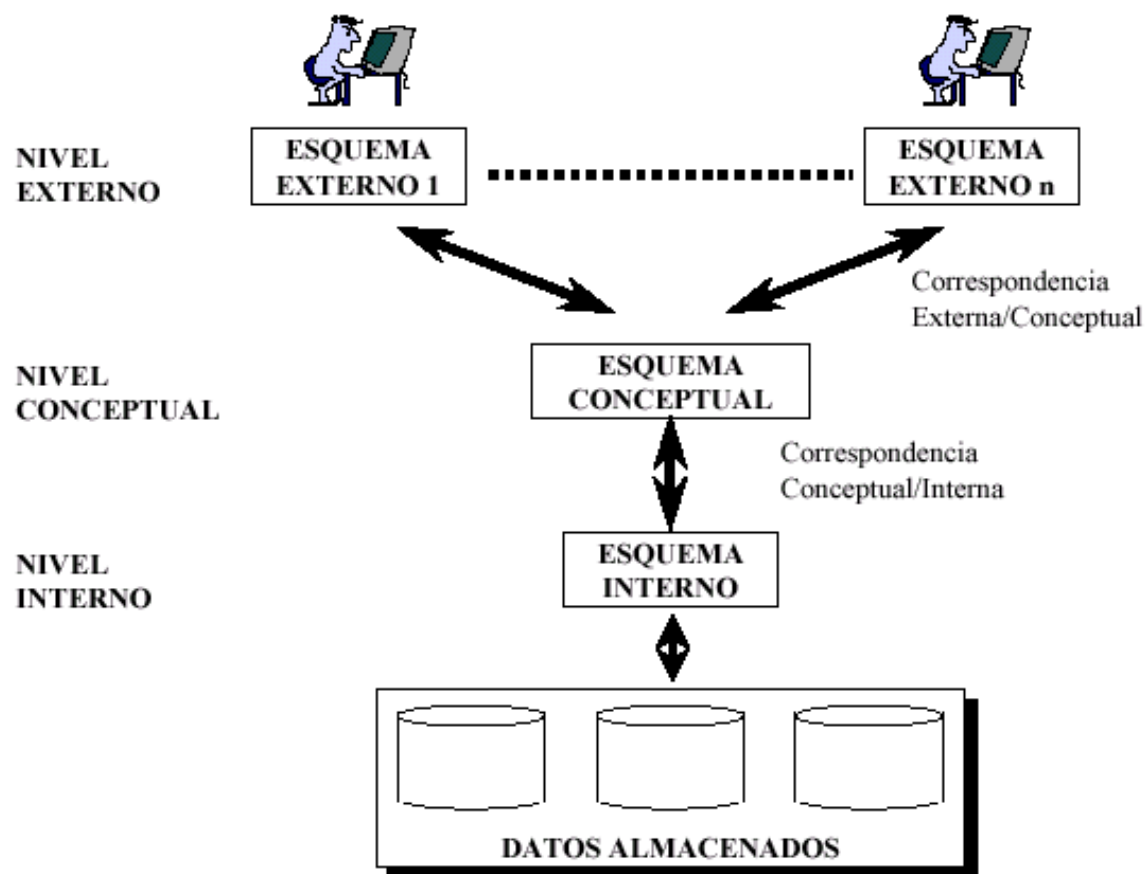


El flujo de datos e instrucciones entre estos elementos es el siguiente:

- a) Se produce una llamada desde una UE al SGBD (1); en la llamada se ha de hacer referencia a la vista externa implicada (2).
- b) El SGBD analiza la llamada y completa los argumentos con la información de la vista externa a la que se ha hecho referencia en la llamada, así como con la correspondiente a la estructura lógica global y la estructura interna con ella relacionadas. Esta información se encuentra previamente almacenada en los ficheros del sistema (Diccionario de Datos), desde donde pasa al SGBD (3 y 4).
- c) Una vez comprobado el derecho del programa de aplicación a utilizar esta vista, y después de verificar su corrección, el SGBD traduce la llamada convirtiéndola en órdenes a los métodos de acceso del sistema operativo (5). Si el sistema había utilizado recientemente los mismos datos, es posible que éstos aún se hallen en el área de almacenamiento intermedio (12).
- d) El SO accede al soporte secundario (disco) donde se encuentran almacenados los datos (6).
- e) Los datos a recuperar pasan del soporte donde se encuentra almacenada la base de datos al área de almacenamiento intermedio (buffer); si se tratase de una inserción o modificación pasarían en sentido contrario (7).
- f) Los datos son transferidos desde el buffer al área de trabajo del usuario de la UE que hizo la llamada (8), o en sentido contrario si se trata de una inserción o modificación, realizándose las correspondientes transformaciones entre las representaciones de los datos.
- g) El SGBD, una vez terminada la operación de manipulación, pasa al área de comunicación los indicadores de estado (9); en los que se señala si la operación ha acabado satisfactoriamente o no, al tiempo que se dan otras informaciones sobre la operación realizada.
- h) El PA revisa el estado de los indicadores que se encuentran en el área de comunicación de la unidad de ejecución desde la que se efectuó la llamada y toma las decisiones oportunas (10).
- i) En el caso de que la operación haya terminado satisfactoriamente, los datos que se encuentran en el área de E/S de la correspondiente unidad de ejecución ya pueden ser utilizados por el PA (11).

## 8. LA ARQUITECTURA ANSI/X3/SPARC

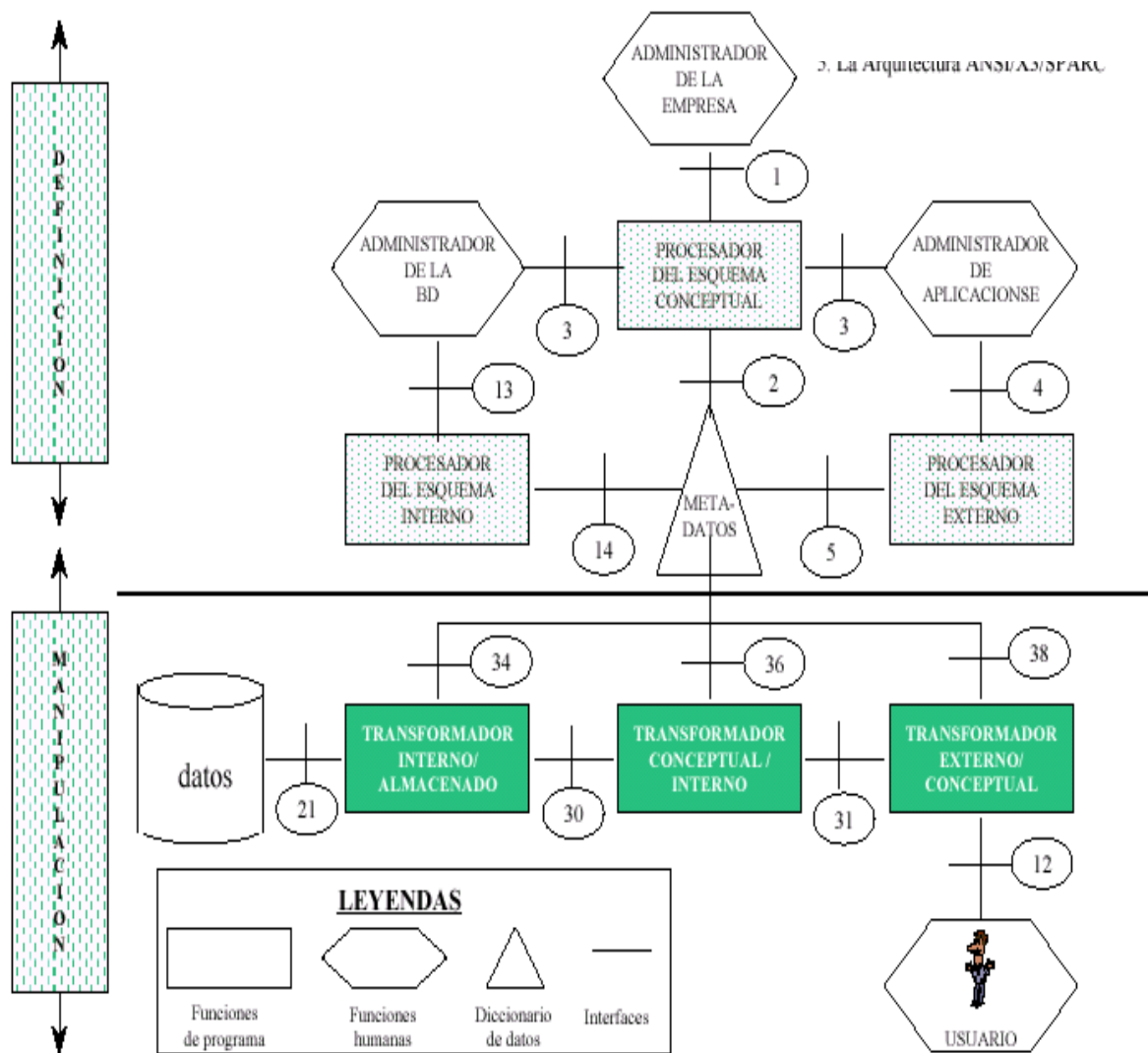
La arquitectura a tres niveles del grupo ANSI en sus estudios de la normativa ANSI/X3/SPARC (Comité de Planificación y Requerimientos del Instituto Nacional Americano de estándares que , a su vez, depende del organismo ISO, surgida en el año 1977) ha marcado una clara línea de investigación en el campo de las bases de datos. Aun cuando en propuestas de normalización anteriores ya se había indicado la conveniencia de separar los tres niveles de estructuras (externo, conceptual e interno), ninguno de estos trabajos profundizó tanto en el tema.



Del esquema conceptual (descripción global de los datos) se deriva una colección de esquemas externos que son la visión que tienen de la base de datos las distintas aplicaciones; el esquema interno es la descripción de los datos de cara a la máquina. La transformación de unos esquemas en otros (funciones de correspondencia) los lleva a cabo el SGBD.

La arquitectura ANSI/X3/SPARC está parcialmente basada en el concepto de máquinas anidadas (tipo cebolla). El flujo de datos pasa a través de distintas capas, que están separadas por interfaces y cuyas funciones se describen con detalle en la documentación. Las múltiples interfaces, cuyo número se ha considerado excesivo, tienden a aislar los diversos componentes del sistema con vistas a conseguir el objetivo de independencia.

En la arquitectura propuesta se distinguen dos partes: la superior para la definición de la base de datos, y la inferior para su manipulación. En esta arquitectura se describen distintas funciones: humanas, representadas en la figura por hexágonos; funciones de programa, que se presentan por medio de rectángulos; interfaces, que se representan mediante líneas y para cuya instrumentación el informe no dicta ninguna norma, pudiendo ser, por tanto, interfaces físicas, lógicas, microprogramadas, etc., y **metadatos** o **diccionario de datos** que desempeñan un papel fundamental en esta arquitectura y que se representa por medio de un triángulo.



**Definición de la base de datos.** La definición se realiza a través de una serie de funciones de programa e interfaces, dando lugar a un conjunto de datos llamados *metadatos* (*datos acerca de los datos*) que se almacenan en el *diccionario*. Este **diccionario de datos** es el eje principal de la arquitectura alrededor del cual giran los demás elementos, en el diccionario de datos se almacenarán:

- ✓ Las descripciones interna, conceptual y externa de la BD así como las reglas de correspondencia necesarias para el paso de un esquema a otro.
- ✓ Las descripciones de los campos, registros y referencias cruzadas entre los registros de varios ficheros.
- ✓ Los códigos de autorización y seguridad de los datos.
- ✓ Los esquemas externos que son empleados por cada aplicación, quiénes son sus usuarios y qué autorizaciones tienen.

Una base de datos se define especificando primeramente el esquema conceptual a través de la interfaz 1, que podría ser un lenguaje de definición del esquema conceptual – o bien una herramienta CASE integrada. Este esquema conceptual es compilado por el procesador del esquema conceptual y se almacena por medio de la interfaz 2 en el diccionario de datos.

El procesador del esquema conceptual es capaz de mostrar la información acerca del esquema conceptual utilizando la interfaz 3, que podría consistir, por ejemplo, en un conjunto de *menús*. Utilizando esta información pueden definirse los esquemas externo e interno a través de las interfaces 4 y 13, que serían controlados por los procesadores correspondientes, y almacenados en la base de datos a través de las interfaces 5 y 14. Estos tres tipos de esquemas, claramente diferenciados, llevan a ANSI/SPARC a proponer la existencia de tres tipos de administradores: el administrador de la empresa, el administrador de la base de datos y el administrador de aplicaciones; triple funcionalidad que puede ser asumida por una única persona o grupo de personas.

**Manipulación de la base de datos.** El usuario puede entonces manipular (insertar, borrar, modificar y recuperar) los datos utilizando la interfaz 12, que podría ser un lenguaje de manipulación, por ejemplo SQL. Una petición de datos por parte del usuario es ejecutada por los transformadores externo/conceptual, conceptual/interno y interno/almacenado, que utilizan los metadatos por medio de las interfaces 38,36 y 34, respectivamente. La solicitud del usuario en la interfaz 12 la convierten los transformadores en peticiones a las interfaces 31,30 y 21, que devuelven el resultado al usuario. Estas últimas interfaces constituyen la función de vinculación entre los distintos niveles (conceptual, interno y almacenado).

Todas las interfaces las ha de suministrar el SGBD, pero ANSI no especifica la forma de implementación.

La arquitectura a tres niveles de ANSI/SPARC responde positivamente a las exigencias de independencia, flexibilidad y capacidad de evolución. La transformación del esquema conceptual por cambios en el sistema de información de la empresa, siempre que no sean demasiado drásticos, será admitida por el SGBD que proporcionará nuevas funciones de correspondencia



(*mapping*) de forma que los programas de usuario no adviertan siquiera que dichos cambios se han producido. De igual modo, el añadir vistas externas que permitan abordar nuevas funciones de la empresa o el introducir cambios en el esquema interno, sólo afectará a los procesadores del SGBD que, mediante las correspondientes interfaces, tendrán que modificar las funciones de correspondencia adaptándolas a las nuevas circunstancias.

## 9. LA ARQUITECTURA Y LA INDEPENDENCIA FÍSICO/LÓGICA

Uno de los principales objetivos de las bases de datos es conseguir la independencia entre las estructuras lógica y física de los datos, que tiene como consecuencia la independencia entre datos y aplicaciones, de modo que los cambios en la estructura de los datos (siempre necesarios, ya que las empresas, como organismos vivos, son por su misma esencia cambiantes) tengan una repercusión mínima en los programas de aplicación, y viceversa.

El concepto de independencia de los datos, elemento clave en las bases de datos, implica la separación entre el almacenamiento y la organización lógica de los datos tal como éstos se contemplan por los distintos programas de aplicación que hacen uso de la base, con lo que se consigue una doble finalidad:

- ✓ Unos mismos datos se presentarán de formas distintas según las necesidades de los usuarios.
- ✓ El almacenamiento de los datos, su estructura lógica y los programas de aplicación serán independientes unos de otros, de modo que un cambio en uno de ellos no obliga a modificar los demás.

En relación con la **independencia** es posible considerar tres aspectos:

- ✓ Su influencia en la **arquitectura** del sistema (niveles de abstracción que como hemos visto ayudan a aumentar la independencia).
- ✓ Características físicas o lógicas que están implicadas en la dependencia (**grado de independencia**), donde se indica cuáles son las características que pueden alterarse en cada uno de los niveles de abstracción sin que ello influya en los otros niveles.
- ✓ Fase o etapa del proceso (compilación, montaje, carga, ejecución) en la que se efectúa la correspondencia o transformación (*mapping*) entre los distintos niveles; el sistema de gestión de la base de datos es el que se ocupa de llevar a cabo la *vinculación* (*binding*) entre niveles, durante la cual se efectúa la resolución de los símbolos.

La rápida evolución que se está produciendo en los ordenadores y las profundas transformaciones que implica la actual concepción de la informática ponen en un primer plano de actualidad el objetivo de independencia ante la

necesidad de garantizar las costosas inversiones que las organizaciones han realizado en el desarrollo e implantación de sus sistemas de información. Es preciso aislar las aplicaciones existentes de las técnicas de almacenamiento y acceso, pero al mismo tiempo es imprescindible dotar al sistema de la flexibilidad suficiente para que pueda beneficiarse de los avances que suponen las nuevas tecnologías.

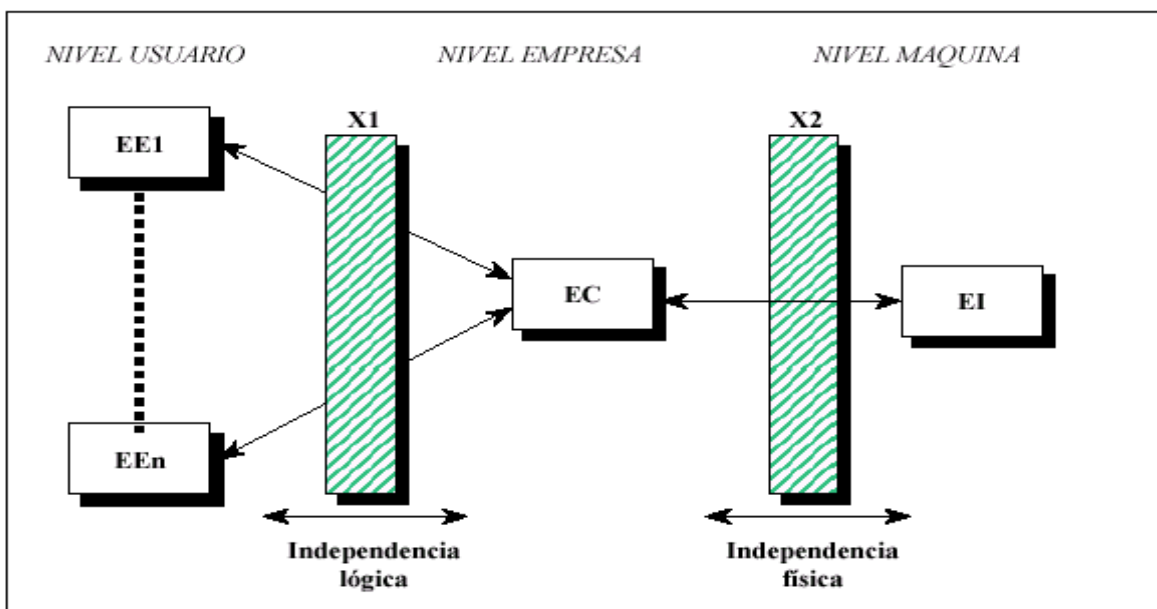
Dentro de la independencia física/lógica, debemos distinguir dos tipos distintos, que se corresponden con las dos principales funciones de los SGBD: la independencia en la descripción y la independencia en la manipulación.

La *independencia en la descripción* permite separar la definición de los datos a nivel físico y a nivel lógico, mientras que la *independencia en la manipulación* se refiere a la de los programas de aplicación respecto a los caminos de acceso y al soporte físico donde se almacenan los datos. Así como en la independencia en la descripción es fundamental la arquitectura del SGBD (sus niveles), en la independencia en la manipulación influye también el modelo de datos, ya que ciertos modelos, como el Jerárquico y el Red, incluyen caminos de acceso en el nivel lógico, por lo que los usuarios han de conocerlos y apoyarse en ellos cuando escriben sus programas de aplicación.

Hay que insistir en que las arquitecturas basadas en distintos niveles de abstracción están muy relacionadas con la independencia físico/lógica.

En una arquitectura a tres niveles, los usuarios están aislados de los datos almacenados físicamente en la máquina por las pantallas X1 y X2, que representan dos funciones de correspondencia (*mapping*). La primera -por medio de la cual se representa la independencia lógica- realiza la transformación de la estructura conceptual (EC) a los esquemas externos (EE), y la segunda -independencia física- del esquema interno (EI) al conceptual.

#### Independencia físico/lógica en una arquitectura a tres niveles



## 10. ESTRUCTURA GENERAL DE UN SGBD

Un SGBD se divide en módulos que se encargan de cada una de las tareas del sistema. El Sistema Operativo proporciona únicamente los servicios más elementales y el SGBD parte de ese fundamento.

Los componentes funcionales son:

- ✓ El **gestor de archivos**: encargado de asignar espacio en el disco para las estructuras de datos. Puede ser más o menos grande, según su grado de relación con el SO.
- ✓ El **gestor de la BD**: que constituye la interfaz entre los datos de bajo nivel almacenados en la BD y los programas de aplicación.
- ✓ El **procesador de consultas**: que traduce las proposiciones en lenguaje de consulta a instrucciones de bajo nivel que puede entender el gestor de la BD. También se encarga de optimizar las consultas.
- ✓ El **precompilador de DML**: que convierte las proposiciones en DML incrustadas en un programa, a llamadas normales a funciones en el lenguaje huésped. Debe interactuar con el procesador de consultas para generar el código apropiado.
- ✓ El **compilador de DDL**: que convierte las proposiciones escritas en DDL en un conjunto de tablas que contienen *metadatos*. Estas tablas se almacenan en el Diccionario de Datos.

Además se requieren otras estructuras dentro del sistema:

- ✓ **Archivos de datos**: almacenan físicamente la Base de Datos.
- ✓ **Diccionario de Datos**: conjunto de archivos que contiene la información relativa a la estructura de la BD. Es consultado constantemente por el gestor de la BD.
- ✓ **Archivos de índices**: Permiten el acceso rápido a los datos. Al igual que el diccionario están incluidos físicamente en los archivos de datos.

## 11. MODELOS DE DATOS

Modelar consiste en definir un mundo abstracto y teórico tal que las conclusiones que se puedan sacar de él coinciden con las manifestaciones aparentes del mundo real.

Llamamos *modelo* al instrumento que se aplica a una parcela del mundo real (universo del discurso UD) para obtener una estructura de datos a la que denominamos *esquema*. El modelo será un conjunto de conceptos, reglas y convenciones que nos permiten describir los datos del UD.

A los valores que toman los distintos objetos de un esquema en un momento determinado los denominaremos *instancia u ocurrencia* del esquema.

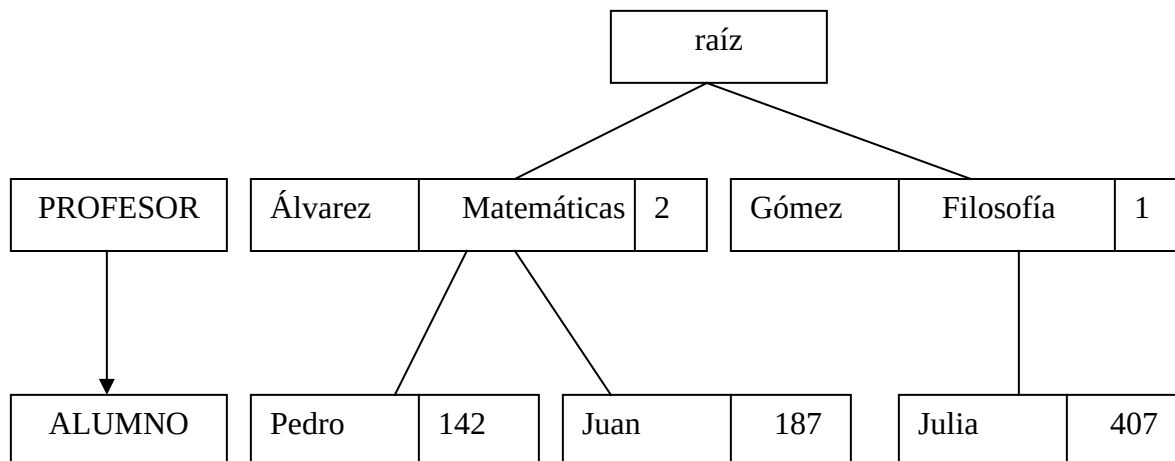
Los modelos son la base para los lenguajes de datos. Por ejemplo, el lenguaje SQL es el resultado de aplicar una determinada sintaxis al modelo relacional.

Los modelos que han contado hasta hace poco con mayor grado de aceptación son tres: Modelo jerárquico, modelo en red y modelo relacional. No obstante, la complejidad creciente de las aplicaciones informáticas actuales está haciendo que el modelo de BD orientadas a objetos gane cada vez más adeptos y, por el contrario, los modelos jerárquico y en red se están dejando de utilizar.

Cada uno de estos modelos nos proporciona un enfoque o tipo diferente de SGBD, implementados en SGBD comerciales, aunque en la actualidad el modelo relacional es el más usado.

### 11.1. MODELO JERÁRQUICO

Un SGBD de tipo jerárquico utiliza **árboles** para la representación lógica de los datos. La figura siguiente muestra un diagrama de estructura de árbol con dos tipos de registros: profesor y alumno, donde un profesor puede tener varios alumnos y una posible ocurrencia de la BD:



Un SGBD jerárquico posee las siguientes **características**:

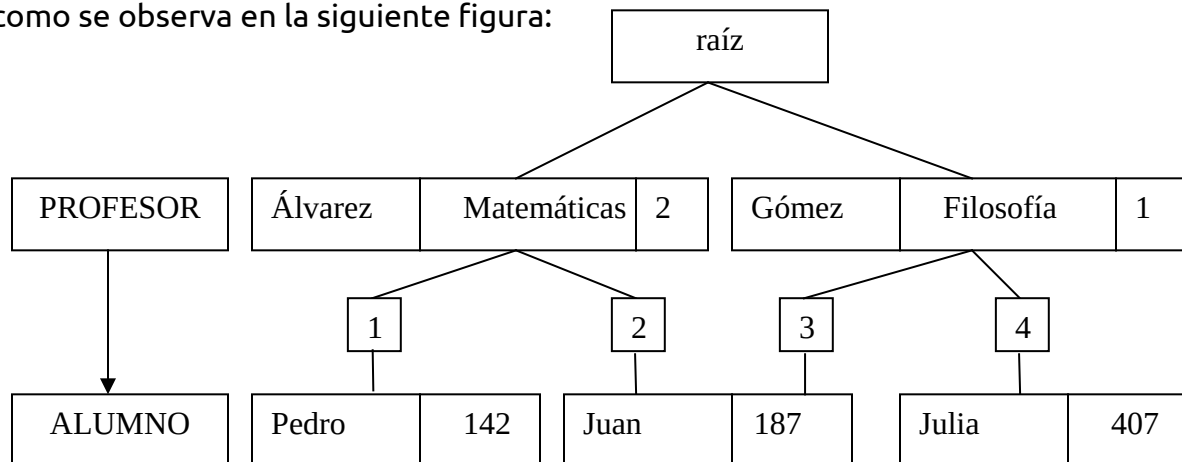
- ✓ Los registros (llamados segmentos) están dispuestos en forma de árbol y no pueden existir ciclos.
- ✓ Los registros sólo pueden estar enlazados mediante relaciones uno a uno o uno a muchos.
- ✓ Cuando se elimina un registro padre se deben borrar todos sus registros hijos.

El SGBD jerárquico más conocido en el mercado es el IMS que utiliza el lenguaje de consulta DL/I.

## 11.2. MODELO EN RED

Los SGBD en red se basan en la utilización de la estructura no lineal red (o Plex), en la que cada registro hijo puede tener más de un nodo padre.

El modelo red más extendido es el modelo Codasyl que presenta la restricción de que no permite directamente la relación muchos a muchos. Si se presenta, se utiliza un registro intermediario, llamado conector o liga-D, que contiene los campos clave principales de los registros que se desean relacionar, como se observa en la siguiente figura:



Un SGBD Codasyl tiene las siguientes **características**:

- ✓ La estructura principal del esquema conceptual es el conjunto DBTG que consiste en dos tipos de registros, con una relación entre ellos. El registro padre se denomina propietario del conjunto (en el ejemplo "profesor"), mientras que el registro hijo se llama miembro (en el ejemplo "alumno"). Existe un solo propietario y uno o más miembros.
- ✓ Los dos tipos de registros deben de ser diferentes.
- ✓ Un registro miembro se puede asociar con más de un propietario.

Algunos SGBD Codasyl existentes en el mercado son: DMS 1100, IDMS e IDS.

## 11.3. MODELO RELACIONAL

Este modelo es posterior a los dos modelos anteriores, y fue desarrollado por Codd en 1970.

Un SGBD relacional utiliza tablas bidimensionales (relaciones) para la representación lógica de los datos y las relaciones entre ellos.

Se llamará registro, entidad o tupla a cada fila de la tabla y campo o atributo a cada columna de la tabla. Una clave será un atributo o conjunto de atributos que identifique de forma única a una tupla.

Las tablas deben cumplir una serie de requisitos:

- ✓ La tabla sólo puede tener un tipo de registro.
- ✓ No existen registros duplicados.
- ✓ Los registros dentro de una relación no tienen una secuencia determinada.
- ✓ Se pueden crear nuevas tablas relacionando campos procedentes de dos o más tablas ya existentes.

Algunos de los SGBD relacionales existentes son: Oracle, Informix, Sybase, MySQL, PostgreSQL, Ms SQL Server, Ms Access.

## 11.4. MODELO ORIENTADOS A OBJETOS

El modelo orientado a objetos se basa en encapsular código y datos en una única entidad, llamada *objeto*. El interfaz entre un objeto y el resto del sistema se define mediante un conjunto de *mensajes*.

Un **objeto** tiene asociado:

- ✓ Un conjunto de variables que contiene los datos del objeto
- ✓ Un conjunto de mensajes a los que el objeto responde.
- ✓ Un método, que es un trozo de código para implementar cada mensaje.

Los objetos similares en una BD se agrupan formando clases, organizadas en forma jerárquica.

Un SGBD orientado a objetos permite:

- ✓ Herencia. Heredan características de su clase y de sus antecesores.
- ✓ Polimorfismo. Distintos métodos con el mismo nombre en el mismo objeto.
- ✓ Objetos complejos. Objetos que contienen otros objetos, por ejemplo, relaciones que pueden almacenarse en otras relaciones.
- ✓ Datos de comportamiento. Distintos objetos necesitan responder de diferente manera a la misma orden. Por ejemplo, la eliminación de ciertas tuplas puede requerir la eliminación de otras tuplas asociadas.
- ✓ Metaconocimiento. A menudo son más importantes reglas generales sobre la relación, más que las tuplas específicas. Por ejemplo, la regla "Todas las cuentas con saldo mayor de 100000 pagan el 5% de interés". Esta regla no se puede representar fácilmente en un SGBD tradicional, mientras que en uno orientado a objetos si se puede.

## 11.5. SGBD DISTRIBUIDOS

La forma tradicional de tratar los datos consiste en que todo el SGBD está concentrado en un solo ordenador. Esto no significa que no pueda servirse a varios usuarios conectados de manera remota, pero esta operación se efectúa compartiendo los recursos del servidor de la misma manera que se hace con cualquier otro tipo de aplicación. De esta forma, el servidor asume la carga de todo el trabajo.

En los sistemas de **tipo distribuido** el SGBD se divide en partes y cada parte puede estar instalada en una máquina diferente. Esta división admite diferentes variantes como, por ejemplo, tener los datos en una máquina y el software del SGBD en otra, tener una copia exacta del SGBD en cada servidor de manera que el usuario se conecte al que menos tráfico tenga en cada momento, o dividir la propia base de datos entre diferentes máquinas en función del uso que vaya a hacerse desde cada una de ellas, entre otras.

Las posibilidades son muchas, pero vamos a centrarnos en dos: la arquitectura cliente/servidor y las bases de datos distribuidas.

### ✓ **Arquitectura Cliente/Servidor**

Consiste en dividir dicho SGBD en dos partes:

- **Servidor.** Se trata de la parte principal del SGBD, la que gestiona la base de datos. Se relaciona con la máquina y con el sistema operativo.
- **Cliente.** Es la parte que utilizan los usuarios, las aplicaciones, los lenguajes y, en ocasiones, las herramientas de gestión y administración.

La división de las funciones en servidor y cliente puede hacerse incluso en la misma máquina, separando los programas que hacen las funciones de servidor de los programas cliente, que son los que utilizará el usuario para de esta forma trabajar con aplicaciones no sobrecargadas de opciones que no va a usar.

La máxima ventaja de esta distribución se obtiene cuando el servidor y el cliente se encuentran en máquinas diferentes: a los terminales remotos (clientes) les corresponde la parte del SGBD que pueden utilizar los usuarios, y desde allí se mandan las peticiones al servidor, que las procesa, hace las consultas o los cambios pertinentes y devuelve al cliente el resultado. De esta manera, instalando la parte servidor del SGBD sólo en una máquina y la parte cliente sólo en los terminales que usan los usuarios, todos los ordenadores quedan descargados de tareas que no les corresponden, y así se aprovechan todos sus recursos. El tráfico entre el servidor y los clientes se limitará a las peticiones de éstos, y a la respuesta correspondiente.

En la actualidad, se intenta evitar instalar parte del SGBD en los ordenadores de los usuarios finales y se prefiere crear un nivel intermedio: los servidores de aplicaciones (son ordenadores que sólo se emplean como servidores de las aplicaciones que los clientes usan y a los cuales acceden desde sus terminales).

### ✓ **Bases de Datos Distribuidas**

En el subapartado anterior se ha mencionado la posibilidad de distribuir entre varias máquinas el SGBD. También es posible hacer lo propio con los datos.

Una base de datos distribuida es aquella donde los datos están repartidos entre diferentes máquinas. Cada máquina dispone de un SGBD (o una parte de él) para dar el servicio correspondiente a los datos que contiene.

Lo más habitual es que algunos de los datos estén repetidos en los diferentes servidores, y entonces se habla de base de datos replicadas. En ocasiones, toda la base de datos se encuentra repetida en todos los servidores. La ventaja de esta opción es que se puede dar un mejor servicio a los clientes; las desventajas son, por un lado, que se necesita una continua actualización de los datos entre los diferentes servidores y, por otro, su coste económico, tanto en hardware como en comunicaciones. Según la utilización que se vaya a hacer de la base de datos, existen diversas posibilidades para distribuir la información.

- ✓ **Toda la base de datos replicada.** Consiste en que cada servidor tiene una copia completa de la base de datos.
- ✓ **Toda la información distribuida sin replicación.** Se basa en que la información se encuentra fragmentada y cada fragmento se guarda en un servidor.
- ✓ **Combinar distribución y replicación.** Se trata de un servidor con toda la base de datos completa y otros con sólo las partes que más se emplean. También es posible tener replicada en varios servidores la parte más utilizada y el resto en uno solo. Por último, otra opción es dejar en la máquina del cliente una copia de la parte que más usa.

Un objetivo básico de un SGBD distribuido es que el usuario lo perciba como si de una BD centralizada se tratase, es decir, el usuario no tiene que saber dónde se encuentran almacenados los datos físicamente.



## EJERCICIOS DE REPASO

1. Describe las funciones del administrador de la base de datos.
2. ¿Qué tipos de diseñadores se pueden distinguir en un sistema de bases de datos y cuáles son sus responsabilidades?
3. ¿Qué diferencias existen entre las funciones de manipulación y descripción?
4. ¿Qué tipos de usuarios interaccionan con una BD?
5. Indica qué es un lenguaje huésped, un lenguaje anfitrión y un lenguaje autocontenido.
6. ¿Cuáles deben ser las características más importantes de un lenguaje orientado a usuarios finales? ¿Y a programadores?
7. Expón las diferencias entre un lenguaje navegacional y uno de especificación.
8. ¿Para qué sirve el área de comunicaciones en un programa que accede a una base de datos?
9. Di si son ciertas y por qué las siguientes sentencias:
  - a. Los lenguajes huésped son más propios de los usuarios finales que los autocontenidos.
  - b. Un lenguaje anfitrión ha de estar siempre embebido en un lenguaje de programación.
  - c. El LDD puede ser tanto de tipo huésped como autocontenido.
  - d. El LMD puede ser tanto de tipo huésped como autocontenido.
  - e. Tanto el acceso a un fichero como a una base de datos se apoya en los métodos de acceso del sistema operativo.
10. ¿Qué debe permitir especificar el lenguaje de definición de la estructura conceptual (lógica global)?
11. ¿Qué elementos se almacenan en el Diccionario de Datos?

12. ¿Qué ventajas y/o que inconvenientes puede tener separar la descripción lógica y física de los datos?
13. Los términos "Abstracción de la información", "Independencia", "Redundancia Mínima", "Consistencia", "Seguridad", "Integridad", "Respaldo y Recuperación", "Control de la Concurrencia" y "Tiempo de Respuesta" ¿Qué relación existe entre ellos?
14. La modificación de los datos de un registro de la base de datos ¿qué función de la base de datos lo realiza y que tipo de lenguaje tiene asociado?
15. ¿Por qué un lenguaje de definición de datos es normalmente autocontenido?
16. Compara la forma en que los programas de aplicación acceden a los datos en un sistema de ficheros y en una base de datos.
17. ¿En qué nivel del SGBD trabajarías si fueras un usuario sin apenas conocimientos de informática?
18. ¿Qué busca la independencia físico/lógica?
19. Define con tus propias palabras modelo de datos.
20. Di que entiendes por esquema.
21. Distingue entre esquema y ocurrencia del esquema. Pon un ejemplo.
22. ¿Qué diferencia existe entre el modelo jerárquico y el modelo en red o Codasyl?
23. ¿Quién desarrollo y en que año el modelo relacional?

24. Una empresa te pide consejo para implantar una base de datos de direcciones y teléfonos de posibles clientes, con la idea de realizar campañas de publicidad. La empresa en cuestión tiene la central en Madrid, así como delegaciones en Sevilla (para la zona centro-sur), Valencia (para la zona este), Santander (para todo el norte de España) y Santa Cruz de Tenerife (para las Islas Canarias). Cada delegación trabajará con la información de su zona. Los datos de los posibles clientes se actualizan anualmente. Aconseja a la empresa sobre la mejor distribución posible para la base de datos, basándote en un razonamiento.
25. ¿En qué consiste la arquitectura cliente/servidor? ¿Qué beneficios ofrece respecto a la forma tradicional de compartir aplicaciones, en la que todo el software estaba en el servidor?
26. ¿Qué diferencia existe entre una base de datos sólo distribuida y una base de datos replicada?
27. Una empresa tiene dos tiendas en dos ciudades muy próximas entre sí. ¿Cómo distribuirías su base de datos? ¿Qué harías, por ejemplo, con el fichero de productos? ¿Y con el de clientes?
28. ¿Qué ventajas e inconvenientes tienen los SGBD Distribuidos?
29. Los conceptos de Herencia y Polimorfismo ¿a qué modelo de datos pertenecen?
30. Indica y describe brevemente los componentes que componen la estructura de un SGBD.