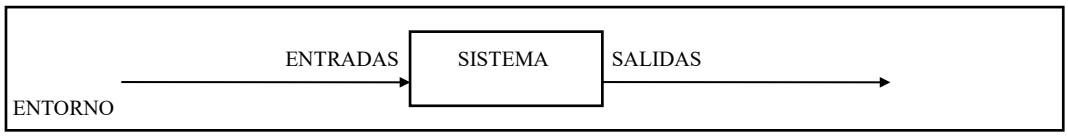
U.D.1. INTRODUCCIÓN A LAS BASES DE DATOS

1. INTRODUCCIÓN A LOS SISTEMAS DE INFORMACIÓN

¿Qué es un sistema?

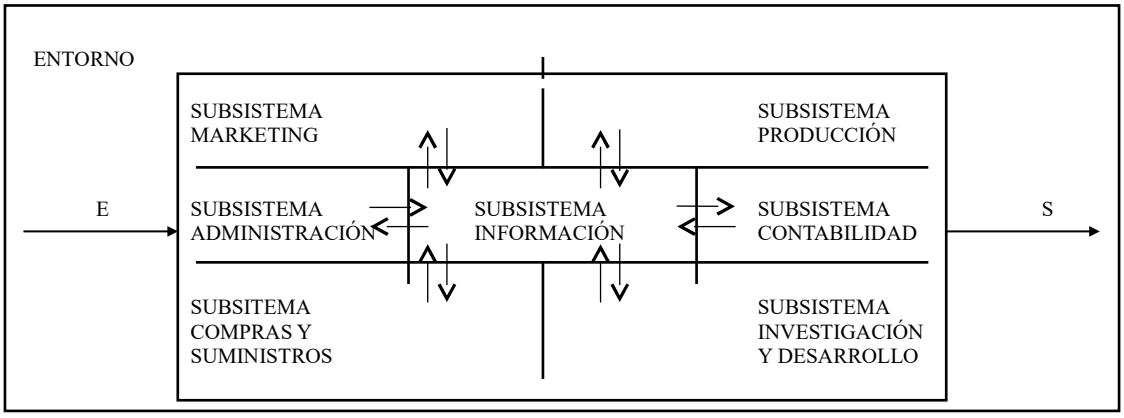
En el sentido más amplio, un **sistema** es un *conjunto de componentes que interaccionan entre sí para lograr un objetivo común*.

Un sistema suele estar situado en un *entorno* o ambiente con el que interactúa, recibe entradas y produce salidas, que establece los límites del sistema.



Una organización o empresa es un sistema en sí. Su entorno es el sistema productivo en que se inserta, del que recibe una serie de entradas, en forma de demandas, y al que entrega una serie de salidas, u oferta. Sus componentes -ventas, contabilidad, manufactura, personal, etc- trabajan juntos para crear utilidades que beneficien tanto a los empleados como a los accionistas. Cada uno de estos componentes es a su vez un sistema (***subsistema***).

Para que los subsistemas que forman la organización funcionen coordinadamente, es necesario otro subsistema más, el **sistema de información**, que tiene como misión asegurar que la información necesaria fluya, dentro del sistema, de unos subsistemas a otros y que inevitablemente existe en cualquier empresa u organización. Del buen funcionamiento de este subsistema depende en gran parte el éxito global del sistema.



Cuando parte o toda la gestión de un sistema de información se realiza con ordenadores se habla de **sistema informático**.

2. FICHEROS

Un ordenador almacena muchos tipos de información, desde datos administrativos, contables o bancarios hasta música, películas, páginas webs, etc. Toda esta información está almacenada en los dispositivos de almacenamiento del ordenador (discos duros, dvds, pen drives, etc). Para poder organizar la información en estos dispositivos, se utilizan los *ficheros* o *archivos*.

Los ficheros son estructuras de información que crean los sistemas operativos de los ordenadores para poder almacenar datos. Suelen tener un nombre y una extensión que determina el formato de la

información que contiene.

2.1. Tipos de ficheros y formatos

El formato y tipo de fichero determina la forma de interpretar la información que contiene, ya que, en definitiva, lo único que se almacena en un fichero es una ristra de bits (ceros y unos), de forma que es necesaria su interpretación para dar sentido a la información que almacena. Así, por ejemplo, para almacenar una imagen en un ordenador, se puede usar un fichero binario *bmp***,** que almacena un vector de datos con los colores que tiene cada píxel que forma la imagen. Además, la imagen posee una paleta de colores y unas dimensiones, información que también hay que almacenar en el fichero. Todos estos datos se ordenan según un formato, y el sistema operativo, o la utilidad que trate los gráficos, debe conocer este formato para poder extraer los píxeles y mostrarlos por pantalla en la forma y dimensiones correctas. Si se abre el gráfico con una utilidad como el bloc de notas, que sólo sabe interpretar texto, el resultado será ilegible e incomprensible.

Tradicionalmente, los ficheros se han clasificado de muchas formas, según su contenido (texto o binario), según su organización (secuencial, directa, indexada) o según su utilidad (maestros, históricos, movimientos).

El **contenido** de un fichero puede ser tratado como *texto* o como *datos* binarios, es decir, los bits almacenados en un fichero pueden ser traducidos por el sistema operativo a caracteres alfabéticos y números que entiende el ser humano, o pueden ser tratados como componentes de estructuras de datos más complejas, como ficheros que almacenan sonido, vídeo, imágenes, etc.

La **organización** de un fichero dicta la forma en que se han de acceder a los datos, así, los datos de un fichero con organización *secuencial*, están dispuestos siguiendo una secuencia ordenada, es decir, unos detrás de otros. Se caracterizan por tener que recorrer todos los datos anteriores para llegar a uno en concreto. Los ficheros de organización *directa*, permiten acceder a un dato en concreto sin necesidad de acceder a todos los anteriores. Finalmente, los de organización *indexada* acceden a los datos consultando un índice, es decir, una estructura de datos que permite acceder a la información rápidamente, simulando la forma en que el índice de un libro facilita el acceso a sus contenidos. Existen también variantes de las anteriores que mezclan las mejores características de cada una de ellas.

Por otro lado, la **utilidad** de un fichero indica qué uso se va a hacer de él, por ejemplo, puede contener datos fundamentales para una organización, como los datos de los clientes, que se almacenan en un fichero principal llamado *maestro.* Si hay variaciones (altas, modificaciones o bajas de clientes) en los ficheros maestros, se almacenan en los llamados ficheros de *movimientos* que posteriormente se enfrentan con los maestros para incorporar las modificaciones. Finalmente, cuando existen datos que ya no son necesarios para su proceso diario pasan a formar parte de los ficheros *históricos*.

Hoy en día, estas dos últimas clasificaciones han quedado en desuso. Por ejemplo, desde la aparición de las bases de datos modernas, ya no se clasifican según su utilidad u organización.

Actualmente un sistema operativo trata un fichero desde dos puntos de vista:

1. Según su contenido (texto o datos binarios)
2. Según su tipo (imágenes, ejecutables, clips de vídeos, etc)

2.2. Ficheros de texto

Los ficheros de texto suelen llamarse también ficheros planos o *ascii*. El vocablo ascii es un acrónimo de American Standard Code for Information Interchange. Es un estándar que asigna un

valor numérico a cada carácter, con lo que se pueden representar los documentos llamados de Texto Plano, es decir, los que son directamente legibles por seres humanos.

La asignación de valores numéricos a caracteres viene dada por la famosa tabla de códigos ascii, que es la más extendida, aunque existen otras. Se caracteriza por utilizar un byte para la representación de cada carácter. Con x bits se pueden generar 2^x combinaciones distintas de caracteres, y como 1 byte = 8 bits, existen 2^8 = 256 caracteres en la tabla de códigos ascii, numerados del 0 al 255.

Actividad 1: Conéctate a internet y busca una tabla de códigos ascii de 8 bits.

Algunos alfabetos como el katakana japonés utilizan más de 256 caracteres. En estos casos se requieren las tablas de caracteres *unicode*, que reservan dos bytes para cada carácter.

Actividad 2: Conéctate a http://www.unicode.org/charts y descárgate las tablas de códigos Latín (alfabeto latíno) y Katakana (alfabeto japonés). Observa las siguientes curiosidades:

- La tabla de códigos Latín es exactamente idéntica a la tabla de códigos ascii de 8 bits, solo que los bits del primer byte unicode están todos a 0.
- Las tablas de códigos Latín y Katakana tienen múltiples extensiones como Katakana Phonetic Extensions o Latin Extended Additional.

Los ficheros de texto, aunque no necesitan un formato para ser interpretado, suelen tener extensiones para conocer qué tipo de texto se halla dentro del fichero, por ejemplo:

- Ficheros de configuración: Son ficheros cuyo contenido es texto sobre configuraciones del sistema operativo o de alguna aplicación. Estos pueden tener extensión .ini, .inf, .conf
- Ficheros de código fuente: Su contenido es texto con programas informáticos. Ejemplos: .sql, .c, .java
- Ficheros de páginas web: Las páginas webs son ficheros de texto con hipertexto que interpreta el navegador: .html, .php, .css, xml
- Formatos enriquecidos: Son textos que contienen códigos de control para ofrecer una visión del texto más elegante: .rtf, .ps, .tex

¿Sabías que …? XML es un lenguaje estándar para el intercambio de datos entre aplicaciones informáticas. Se están desarrollando actualmente las llamadas bases de datos nativas XML, cuyo foco principal es el almacenamiento de documentos de texto con código en XML, y no las relaciones entre la información, como sucede con las bases de datos relacionales que se estudian en este módulo. Por ejemplo, DB2 incorpora dentro de su motor una nueva característica que potencia el XML: Xquery, esto es, un lenguaje innovador para hacer consultas directamente sobre documentos XML guardados directamente en la base de datos.

2.3. Ficheros binarios

Los ficheros binarios son todos los que no son de texto y requieren un formato para ser interpretado. A continuación se muestran algunos tipos de formatos de ficheros binarios:

- De imagen: .jpg, .gif, .tiff, .bmp, .wmf, .png, .pcx (entre otros muchos)
- De vídeo: .mpg, .mov, .avi, .qt
- Comprimidos o empaquetados: .zip, .Z, .gz, .tar, .lhz
- Ejecutables o compilados: .exe, .com, .cgi, .o, .a
- Procesadores de texto: .doc, .odt

Generalmente los ficheros que componen una base de datos son de tipo binario, puesto que la información que hay almacenada en ellos debe tener una estructura lógica y organizada para que las aplicaciones puedan acceder a ella de manera universal, esto es, siguiendo un estándar. Esta estructura lógica y organizada, generalmente es muy difícil de expresar mediante ficheros de texto, por tanto, la información de una base de datos se suele guardar en uno o varios ficheros:

- El software de gestión de base de datos de Oracle guarda la información en múltiples tipos de ficheros, llamados 'datafiles', 'tempfiles', 'logfiles', etc.
- Un tipo de tablas del gestor MySQL guarda su información en 3 ficheros de datos binarios, con extensión .frm, .myd y .myi

3. BASES DE DATOS

Una **base de datos** es una colección de información perteneciente a un mismo contexto, que está almacenada de forma organizada en ficheros.

Una base de datos está organizada mediante tablas, que almacenan información concerniente a algún objeto o suceso. Estas tablas se relacionan formando vínculos o relaciones entre ellas, que ayudan a mantener la información de los diversos objetos de forma ordenada y coherente (sin contradicciones). Cada una de estas tablas es una estructura que se parece a las hojas de cálculo, pues está dispuesta mediante filas y columnas. De este modo, cada fila almacena un registro con tantos campos como columnas tenga la tabla. Por ejemplo, se podría tener una tabla de Empleados, donde cada fila o registro es un empleado de la empresa y cada columna o campo representa un trozo discreto de información sobre cada empleado, por ejemplo el nombre o el número de teléfono.

CodEmp	CodDep	ExTelEmp	FecInEmp	FecNaEmp	NifEmp	NomEmp	NumHi	SalEmp
1	DIRGE	1111	1972-07-01	1961-08-07	21451451V	Saladino Mandamás, Augusto	1	7200000.00
2	IN&DI	2233	1991-06-14	1970-06-08	21231347K	Manrique Bacterio, Luisa	0	4500000.00
3	VENZS	2133	1984-06-08	1965-12-07	23823930D	Monforte Cid, Roldán	1	5200000.00
4	VENZS	3838	1990-08-09	1975-02-21	38293923L	Topaz Illán, Carlos	0	3200000.00
5	ADMZS	1239	1976-08-07	1958-03-08	38223923T	Alada Veraz, Juana	1	6200000.00
6	JEFZS	23838	1991-08-01	1969-06-03	26454122D	Gozque Altanero, Cándido	2	5000000.00

Tabla Empleados

3.1. Conceptos

- **Dato:** El dato es un trozo de información concreta sobre algún concepto o suceso. Por ejemplo, 3 es un número que representa un código de un empleado en la tabla anterior. Los datos se caracterizan por pertenecer a un tipo.
- **Tipo de dato:** El tipo de dato indica la naturaleza del campo. Así, se puede tener datos *numéricos*, que son aquellos con los que se pueden realizar cálculos aritméticos (sumas, restas, …) y los datos *alfanuméricos* que contienen caracteres alfabéticos y dígitos numéricos. Estos datos alfanuméricos y numéricos se pueden combinar para obtener tipos de datos más elaborados. Por ejemplo, el tipo de dato Fecha contiene tres datos numéricos que representan el día, el mes y el año (12/05/1990).
- **Campo:** Es una columna de una tabla. Cada campo pertenece a un tipo de datos. Por ejemplo, el campo *FecNaEmp* representa las fechas de nacimiento de los empleados que hay en la tabla anterior.
- **Registro:** Es una colección de datos referentes a un mismo concepto o suceso. Por ejemplo, los datos de un empleado en concreto serían su código de empleado, código del departamento al que pertenece, extensión telefónica, fecha de incorporación a la empresa, fecha de nacimiento, su NIF, su nombre, su numéro de hijos y su salario. Sería cada una de las filas o tuplas de la tabla.
- **Tabla:** Es un conjunto de registros bajo un mismo nombre que representa al conjunto de todos ellos. Por ejemplo, todos los empleados de una base de datos se almacenan en una tabla cuyo nombre es *Empledos*.
- **Campo clave:** Es un campo especial que identifica de forma única a cada registro. Así, el NIF es único para cada empleado, por tanto es un campo clave. También lo sería su código de empleado.
- **Consulta:** Es una instrucción para hacer peticiones a una base de datos. Puede ser una búsqueda simple de un registro específico o una solicitud para seleccionar todos los registros que satisfagan un conjunto de criterios. Aunque en castellano una consulta tiene un significado de extracción de información, en inglés, *query*, una consulta es una petición, por tanto, además de las consultas de búsqueda de información que devuelven campos de los registros solicitados, hay consultas (peticiones) de eliminación o inserción de registros y de actualización.
- **Índice:** Es una estructura que almacena campos de una tabla, organizándolos para hacer más fácil encontrar y ordenar los registros de esa tabla. Sería el equivalente al índice de un libro que permite el acceso a un capítulo determinado sin tener que ir buscando desde la primera página del libro.
- **Vista:** Es una nueva tabla virtual que se basa en la información de una o más tablas. Por ejemplo, podríamos generar una vista a partir de la tabla de empleados anterior que sólo nos mostrase el nif, nombre y salario de los empleados de un determinado departamento.
- **Informe:** Es un listado ordenado y formateado de la información de la base de datos.
- **Guiones o scripts:** Son un conjunto de instrucciones que ejectudas de forma ordenada realizan operaciones avanzadas de mantenimiento de los datos almacenados en la base de datos.
- **Procedimiento:** Son un tipo especial de script que está almacenado en la base de datos y que forma parte de su esquema.

3.2. Estructura de una base de datos

Una base de datos almacena los datos a través de un *esquema*. El esquema es la definición de la estructura donde se almacenan los datos, contiene todo lo necesario para organizar la información mediante tablas, registros (filas) y campos (columnas). También contiene otros objetos necesarios para el tratamiento de los datos (procedimientos, vistas, índices, etc). Al esquema también se le suele llamar *metainformación*, es decir, información sobre la información o *metadatos*.

Los gestores de bases de datos modernos (Oracle, MySQL, …) almacenan el esquema de la base de datos en tablas, de tal manera que el propio esquema de la base de datos se puede tratar como si fueran datos comunes de la base de datos.

3.3. Usos de las bases de datos

Las bases de datos están en cualquier tipo de sistema de informático. A continuación se exponen sólo algunos ejemplos de sus usos más frecuentes:

- Bases de datos Administrativas: Cualquier empresa necesita registrar y relacionar sus clientes, pedidos, facturas, productos, etc.
- Bases de datos Contables: También es necesario gestionar los pagos, balances de pérdidas y ganancias, patrimonio, declaraciones de hacienda, …
- Bases de datos para motores de búsqueda: Por ejemplo Google o Altavista, tienen una base de datos gigantesca donde almacenan información sobre los documentos de interenet. Posteriormente millones de usuarios buscan en la base de datos de esos motores.
- Científicas: Recolección de datos climáticos y medioambientales, químicos, geológicos, …
- Configuraciones: Almacenan datos de configuración de un sistema informático, como por ejemplo, el registro de windows.
- Bibliotecas: Almacenan información bibliográfica, por ejemplo, la biblioteca de un instituto.
- Censos: Guarda información demográfica de pueblos, ciudades y paises.
- Virus: Los antivirus guardan información sobre todos los potenciales software maliciosos.
- Otros muchos usos: Militares, videojuegos, deportes, etc.

3.4. Evolución y tipos de bases de datos

La clasificación de las bases de datos en tipos está ligada a su evolución histórica. Según ha ido avanzando la tecnología, las bases de datos han mejorado cambiando la forma de representar y extraer la información.

En la década de 1950 se inventan las cintas magnéticas que sólo podían ser leídas de forma secuencial y ordenadamente. Estas cintas almacenaban ficheros con registros que se procesaban se procesaban secuencialmente junto con ficheros de movimientos para generar ficheros actualizados. Estos sistemas se conocen como *aplicaciones basadas en sistemas de ficheros* y constituyen la generación cero de las bases de datos, pues ni siquiera entonces existía el concepto de bases de datos.

En la década de 1960 se generaliza el uso de discos magnéticos, cuya característica principal es que se podía acceder de forma directa a cualquier parte de los ficheros, sin tener que acceder a todos los datos anteriores. Con esta tecnología aparecen las bases de datos jerárquicas y en red, que aprovechan la capacidad de acceso directo a la información de los discos magnéticos para estructurar la información en forma de listas enlazadas y árboles de información.

Edgar Frank Codd, científico informático inglés de IBM, publica en 1970 en un artículo 'Un modelo relacional de datos para grandes bancos de datos compartidos', donde definió el modelo relacional, basado en la lógica de predicados y la teoría de conjuntos. Nacieron de esta forma las bases de datos relacionales, o segunda generación de bases de datos. Larry Ellison, fundador de Oracle, se inspiró en este artículo para desarrollar el famoso motor de de base de datos, que comenzó como un proyecto para la CIA americana. La potente base matemática de este modelo es el gran secreto de su éxito. Hoy en día, el modelo relacional de Codd, pese a tener muchas alternativas, sigue siendo el más utilizado a todos los niveles.

En la década de 1980 IBM lanza su motor de bases de datos DB2, para la plataforma MVS. Unos años después, IBM crea el SQL, un potente lenguaje de consultas para manipular información de bases de datos relacionales.

A mediados de 1990, IBM lanza una versión de DB2 que es capaz de dividir una base de datos enorme en varios servidores comunicados por líneas de gran velocidad, creándose de este modo las *bases de datos paralelas*. A esta versión se le llamó DB2 Parallel Edition, que ahora ha evolucionado hasta el DB2 Data Partition Feature, único SGBD de este tipo en sistemas distribuidos.

A finales de 1990 IBM y Oracle incorporan a sus bases de datos la capacidad de manipular objetos, creando así, las *bases de datos orientadas a objetos*. Estas bases de datos orientadas a objetos se basan en la existencia de objetos que se almacenan para su procesamiento mediante programas orientados a objetos. En lugar de la filosofía de almacenar relaciones y tablas, se almacenan colecciones de objetos que, además de información, tienen métodos (instrucciones sobre cómo procesar los datos).

La aparición de Internet y el comienzo de la era de la información, crean nuevos requerimientos para bases de datos. La cantidad de información comienza a crecer en proporciones desconocidas hasta el momento. De esta forma, se crean las *bases de datos distribuidas*, que consisten en multiplicar el número de ordenadores que controlan una base de datos (llamados nodos), intercambiándose información y actualizaciones a través de la red. Este increíble aumento de datos a almacenar, organizados muchas veces en datos estadísticos recopilados con el trascurso de los años, hizo necesaria la aparición de un software llamado Software de ayuda a la decisión. Este software avanzado trata de dar respuestas concretas examinando múltiples datos estadísticos que se han recopilado a lo largo del tiempo en bases de datos multidimensionales, formando lo que se denominan cubos de información.

También, a lo largo de la corta historia de la informática, han surgido otros tipos de bases de datos que se enumeran a continuación:

- *Bases de datos espaciales o geográficas*: Son bases de datos que almacenan mapas y símbolos que representan superficies geográficas. Google Earh es una aplicación que lanza consultas a bases de datos de este tipo.
- *Bases de datos documentales*: Permiten la indexación de texto para poder realizar búsquedas complejas en textos de gran longitud.
- *Bases de datos deductivas*: Es un sistema de bases de datos que almacenan hechos y que permite, a través de procedimientos de inferencia, extraer nuevos hechos. Se basan en la lógica, por ello también se suelen llamar bases de datos lógicas.

Bases de datos	**Datos almacenados**	**Ubicación**

Sistemas de ficheros Jerárquicas En red	Datos en ficheros Estructuras de datos (listas y árboles) Estructuras de datos (árboles y grafos)	Varios ficheros
Relacionales Orientados a objetos Geográficas Deductivas Documentales	Teoría de conjuntos y relaciones Objetos complejos con comportamiento Puntos, Líneas y Polígonos Hechos y Reglas Documentos	Una o varias BBDD
Distribuidas Multidimensionales	Múltiples Cubos	Varias BBDD en varios ordenadores

Resumen de los tipos de bases de datos

4. LOS SISTEMAS GESTORES DE BASES DE DATOS

4.1. Concepto de un Sistema Gestor de Bases de Datos

Se puede definir el SGBD como un *conjunto coordinado de programas, procedimientos, lenguajes, etc. que suministra, tanto a los usuarios no informáticos como a los analistas, programadores o al administrador, los medios necesarios para describir, recuperar y manipular los datos almacenados en la base, manteniendo su integridad, confidencialidad y seguridad*.

Ejemplos de SGBD comerciales serían MySQL, Oracle, Informix, …

4.2. Componentes

DATOS: Evidentemente el componente fundamental son los **datos** que constituyen la **base de datos**, que se puede definir como una *colección de datos interrelacionados entre sí, almacenados en un conjunto sin redundancias innecesarias, cuya finalidad es la de servir a una o más aplicaciones de la manera más eficiente*.

SOFTWARE: Este software es el encargado tanto de crear y organizar la base de datos como de atender a todas las solicitudes de acceso a la BD formuladas por las aplicaciones o los usuarios.

USUARIOS: Existen diferentes tipos de usuarios, que se distinguen por el modo en que interactúan con el sistema:

- Administrador. El administrador de la BD o ABD es el encargado de controlar y manejar la BD. Puede ser una sola persona o un equipo. Tiene las siguientes **funciones** específicas:
  - Describir el esquema de la BD, los campos, registros y las 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.
  - Especificar las reglas de integridad de los datos.
- Analistas o diseñadores. Realizan el diseño de la base de datos, debiendo identificar los datos, las relaciones entre los datos y las restricciones sobre los datos y sus relaciones.
- Programadores de aplicaciones. Son los encargados de escribir programas de aplicación que utilicen bases de datos.

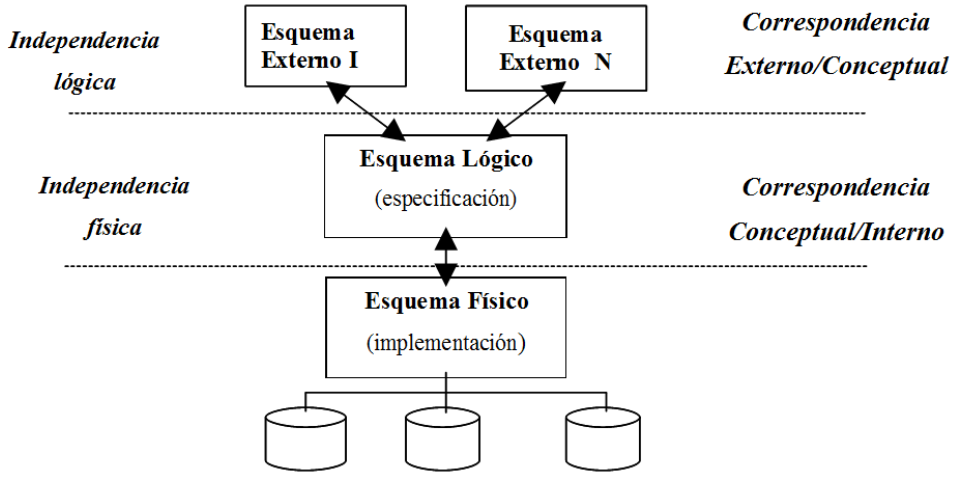
- Usuarios finales. Son usuarios poco experimentados que utilizan las aplicaciones escritas previamente.

4.3. Arquitectura de un SGBD

Uno de los objetivos fundamentales de los SGBD es el de la **independencia** de los datos.

El concepto de independencia de los datos implica la separación entre el almacenamiento y la organización lógica de los datos tal como éstos son contemplados por los distintos programas de aplicación que hacen uso de la base de datos.

Este objetivo de independencia ha tenido una fuerte repercusión en la arquitectura de los sistemas de bases de datos. Para responder a este objetivo de independencia el comité ANSI/SPARC propuso la arquitectura de los SGBD de tres niveles de abstracción, que finalmente se ha estandarizado.



Independencia física/lógica en una arquitectura a tres niveles.

- **Nivel físico (o interno)**. Este es el nivel más bajo de abstracción, en el que se define cómo se almacenan realmente los datos. A esta definición se le denomina **esquema físico**.
- **Nivel lógico**. Este es el siguiente nivel de abstracción, en el que se describe cuáles son los datos reales que están almacenados en la BD y qué relaciones existen entre los datos. Este nivel lo utilizan los administradores de la BD, quienes deciden qué información se guarda en la base de datos. A esta definición se le denomina **esquema lógico.**
- **Nivel externo (o vista).** Este es el nivel de abstracción más alto, en el cual se describe sólamente una parte de la base de datos. Muchos usuarios necesitarán solamente una parte de la base de datos. El sistema puede proporcionar muchas vistas diferentes de la misma base de datos.

Los tres niveles de abstracción anteriores dan lugar a tres esquemas diferentes de una base de datos: ***esquema externo o vista o subesquema***, que es la visión que tienen de la base de datos cada usuario en particular y en el que deberán encontrarse reflejados sólo aquellos datos e interrelaciones que necesite el correspondiente usuario, ***esquema lógico***, que es la visión global de los datos que tiene el ABD y que deberá incluir la descripción de todos los datos e interrelaciones entre éstos, las restricciones de integridad y de confidencialidad, y ***esquema interno***, muy dependiente de cada SGBD, que es la forma en que se organizan los datos en el almacenamiento físico y en el que se deberá especificar la estrategia de almacenamiento, caminos de acceso, índices, desbordamientos, técnicas de compresión, etc., también utilizado por el ABD.

En general los sistemas de bases de datos cuentan con un esquema interno, un esquema conceptual

y varios esquemas externos o vistas.

Por ejemplo, supongamos una vista donde sólo tenemos el nombre y la edad de los empleados. Una solicitud de toda la vista externa se deberá traducir en una solicitud de los datos correctos del esquema lógico, nombre y fecha de nacimiento. Posteriormente se traducirá al esquema físico donde se sabrá exactamente donde están los datos, si hay un índice para hacer más rápido el acceso, etc. Después, cuando se consigan los datos, el proceso de traducción irá a la inversa, es decir, de abajo a arriba en los niveles de la arquitectura.

El proceso de transformación solicitudes y resultados de un nivel a otro se denomina **correspondencia o transformación (*mapping*)**. Este proceso consumirá tiempo pero facilitará lo que se pretendía:

- **Independencia lógica** respecto a los datos: Es la capacidad de modificar el esquema lógico sin que ello afecte a los esquemas externos ni programas de aplicación. Por ejemplo, en el esquema lógico se puede incorporar un nuevo campo para los empleados que sea la fecha de entrada en la empresa. Esto no afectará para nada la vista externa de empleado y edad correspondiente.
- **Independencia física** respecto a los datos: se puede modificar el esquema físico sin que ello tenga que afectar al esquema lógico y mucho menos a los esquemas externos. Así por ejemplo, se puede decidir incluir un nuevo índice para acceder más rápido a los datos en un determinado orden.

4.4. Funciones de un SGBD

Los SGBD del mercado cumplen con casi todas las funciones que a continuación se enumeran:

1. Permiten a los usuarios almacenar datos, acceder a ellos y actualizarlos de forma sencilla y con un gran rendimiento, ocultando la complejidad y las características físicas de los dispositivos de almacenamiento.
2. Garantizan la integridad de los datos, respetando las reglas y restricciones que dicte el programador de la base de datos. Es decir, no permiten operaciones que dejen cierto conjunto de datos incompletos o incorrectos.
3. Integran, junto con el sistema operativo, un sistema de seguridad que garantiza el acceso a la información exclusivamente a aquellos usuarios que dispongan de autorización.
4. Proporcionan un **diccionario** de metadatos**,** que contiene el esquema de la base de datos, es decir, cómo están estructurados en tablas, registros y campos, las relaciones entre los datos, usuarios, permisos, etc. Este diccionario de datos debe ser accesible de la misma forma que es posible acceder al resto de datos.
5. Permiten el uso de transacciones, garantizan que todas las operaciones de la transacción se realicen correctamente, y en caso de alguna incidencia, deshacen los cambios sin ningún tipo de complicación adicional.
6. Ofrecen, mediante herramientas completas, estadísticas sobre el uso del gestor registrando operaciones efectuadas, consultas solicitadas, operaciones fallidas y cualquier tipo de incidencia. Es posible de este modo, monitorizar el uso de la base de datos.
7. Permiten la concurrencia, es decir, varios usuarios trabajando sobre un mismo conjunto de datos. Además, proporcionan mecanismos que permiten arbitrar operaciones conflictivas en el acceso o modificación de un dato al mismo tiempo por parte de varios usuarios.
8. Independizan los datos de la aplicación o usuario que esté utilizándolos, haciendo más fácil su migración a otras plataformas.
9. Ofrecen conectividad con el exterior. Se puede replicar y distribuir bases de datos. Todos los SGBD incorporan herramientas estándar de conectividad. El protocolo ODBC (Open

Database Connectivity, desarrollado por Microsoft) está muy extendido como forma de comunicación entre bases de datos y aplicaciones externas.

10. Incorporan herramientas para la salvaguarda y restauración de la información en caso de desastre. Algunos gestores tienen sofisticados mecanismos para poder establecer el estado de una base de datos en cualquier punto anterior en el tiempo. Además, deben ofrecer sencillas herramientas para la importación y exportación automática de la información.

Actividad 3: Busca en internet las leyes de Codd para el funcionamiento de sistemas gestores de bases de datos relacionales.

4.5. Modelos de BD

Una BD es una representación de la realidad (de la parte de la realidad que nos interesa en nuestro SI). Dicho de otro modo, una BD se puede considerar un modelo de la realidad. El componente fundamental utilizado para modelar en un SGBD relacional son las tablas (denominadas *relaciones* en el mundo teórico).

Sin embargo, en otros tipos de SGBD se utilizan otros componentes.

El conjunto de componentes o herramientas conceptuales que un SGBD proporciona para modelar recibe el nombre de ***modelo de BD****.*

Los cuatro modelos de BD más utilizados en los SI son el modelo relacional, el modelo jerárquico, el modelo en red y el modelo relacional con objetos.

Todo modelo de BD nos proporciona tres tipos de herramientas:

a) **Estructuras de datos** con las que se puede construir la BD: tablas, árboles, etc.

b) Diferentes tipos de **restricciones** (o reglas) de integridad que el SGBD tendrá que hacer cumplir a los datos: dominios, claves, etc.

c) Una serie de **operaciones** para trabajar con los datos. Un ejemplo de ello, en el modelo relacional, es la operación SELECT, que sirve para seleccionar (o leer) las filas que cumplen alguna condición. Un ejemplo de operación típica del modelo jerárquico y del modelo en red podría ser la que nos dice si un determinado registro tiene "hijos" o no.

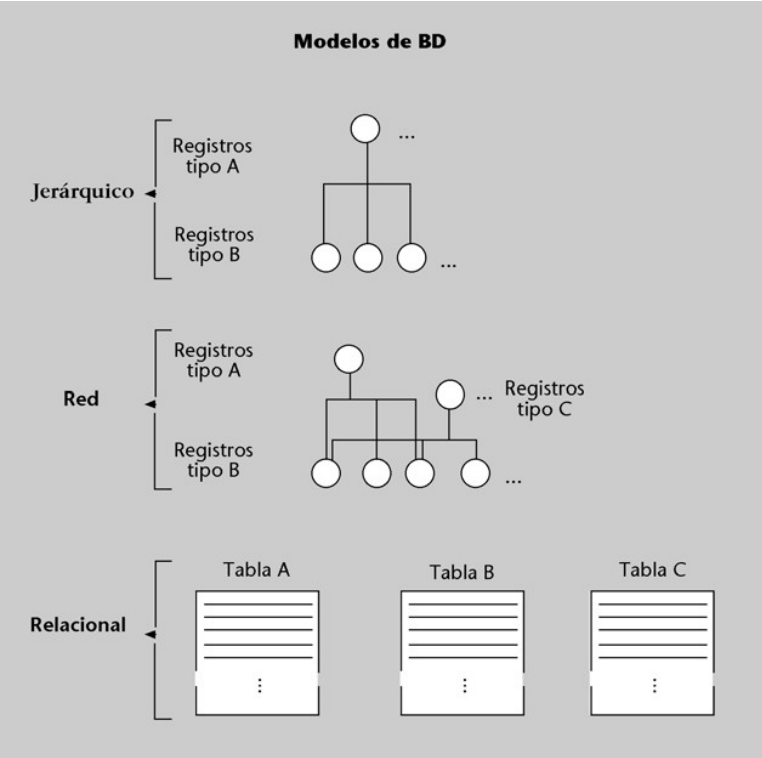
4.5.1. Evolución de los modelos de BD

De los cuatro modelos de BD que hemos citado, el que apareció primero, a principios de los años sesenta, fue el modelo **jerárquico**. Sus estructuras son registros interrelacionados en forma de árboles. El SGBD clásico de este modelo es el IMS/DL1 de IBM.

A principios de los setenta surgieron SGBD basados en un modelo en **red**. Como en el modelo jerárquico, hay registros e interrelaciones, pero un registro ya no está limitado a ser "hijo" de un solo registro tipo. El comité CODASYLDBTG propuso un estándar basado en este modelo, que fue adoptado por muchos constructores de SGBD. Sin embargo, encontró la oposición de IBM, la empresa entonces dominante. La propuesta de CODASYL-DBTG ya definía tres niveles de esquemas.

Durante los años ochenta apareció una gran cantidad de SGBD basados en el modelo **relacional** propuesto en 1969 por E.F. Codd, de IBM, y prácticamente todos utilizaban como lenguaje nativo el SQL El modelo relacional se basa en el concepto matemático de *relación*, que aquí podemos considerar de momento equivalente al término *tabla* (formada por filas y columnas). La mayor parte

de los SI que actualmente están en funcionamiento utilizan SGBD relacionales, pero algunos siguen utilizando los jerárquicos o en red (especialmente en SI antiguos muy grandes).



Estos últimos años se está extendiendo el modelo de BD relacional con objetos. Se trata de ampliar el modelo relacional, añadiéndole la posibilidad de que los tipos de datos sean tipos abstractos de datos, TAD. Esto acerca los sistemas relacionales al paradigma de la OO. Los primeros SGBD relacionales que dieron esta posibilidad fueron Oracle (versión 8), Informix (versión 9) e IBM/DB2/UDB (versión 5).

En esta asignatura hablamos sólo de BD con modelos de datos estructurados, que son los que normalmente se utilizan en los SI empresariales. Sin embargo, hay SGBD especializados en tipos de aplicaciones concretas que no siguen ninguno de estos modelos. Por ejemplo, los SGBD **documentales** o los de BD **geográficas**.

4.6. El lenguaje SQL

La principal herramienta de un gestor de base de datos es la interfaz de programación con el usuario. Este interfaz consiste en un lenguaje que permite cumplir al SGBD todas sus funciones.

El lenguaje SQL (Structured Query Language) es el más utilizado en las BD relacionales y está estandarizado por la ISO (International Organization for Standarization), por lo que todas las bases de datos que soportan SQL han de tener la misma sintaxis a la hora de utilizar el lenguaje. Se divide en cuatro sublenguajes que en conjunto permiten al SGBD cumplir con todas las funcionalidades requeridas por CODD:

- **Lenguaje DML:** o lenguaje de manipulación de datos (Data Manipulation language). Este lenguje permite con 4 sencillas sentencias seleccionar determinados datos (SELECT), insertar datos (INSERT), modificarlos (UPDATE) o incluso borrarlos (DELETE). En capítulos posteriores se desarrollará la sintaxis de cada una de estas sentencias.

- **Lenguaje DDL:** o lenguaje de definición de datos (Data Definition Language). Este lenguaje permite crear toda la estructura de una base de datos (tablas, índices, …) Sus cláusulas son del tipo CREATE (crear objetos) y DROP (eliminar objetos). En unidades posteriores se detallará la sintaxis de cada una de estas sentencias.
- **Lenguaje DCL:** o lenguaje de control de datos (Data Control Language). Incluye comandos (GRANT y REVOKE) que permiten al administrador gestionar el acceso a los datos contenidos en la base de datos.
- **Lenguaje TCL:** o lenguaje de control de transacciones. El propósito de este lenguaje es permitir ejecutar varios comandos de forma simultánea como si fuera un comando atómico o indivisible. Si es posible ejecutar todos los comandos, se aplica la transacción (COMMIT) y si en algún punto de la ejecución sucede algo inesperado, se pueden deshacer todos los pasos dados (ROLLBACK).

4.7. Tipos de SGBD

Se pueden clasificar los SGBD de muchas formas, por ejemplo, según las bases de datos que gestionan, clasificando los SGBD según traten bases de datos relacionales, bases de datos orientadas a objetos, etc. En la actualidad, la mayoría de los SGBD integran múltiples filosofías y tipos de funcionamiento. Por esta razón, clasificaremos los gestores de bases de datos según su capacidad y potencia del propio gestor:

- **Los Gestores de Bases de Datos Ofimáticas** son aquellos que manipulan bases de datos pequeñas orientadas a almacenar datos domésticos o de pequeñas empresas. Incluso permiten construir pequeñas aplicaciones para ayudar a un usuario inexperto a manipular los datos de una base de datos sencilla. Un ejemplo de un SGBD ofimático es Microsoft Access, que posee tanto una interfaz de usuario muy sencilla como un potente lenguaje de programación (Visual Basic for Aplications) para ofrecer a usuarios avanzados otras posibilidades de gestión más específicas.
- **Los Gestores de Bases de Datos Corporativas** son aquellas que tienen la capacidad de gestionar bases de datos enormes, de grandes o medianas empresas con una carga de datos y transacciones que requieren un servidor potente. Estos gestores son capaces de manipular grandes cantidades de datos de forma muy rápida y eficiente para poder resolver la demanda de muchos (cientos) de usuarios. Ejemplos típicos de servidor de bases de datos Corporativas serían Oracle y DB2, muy potentes pero caros.

Precisamente, ese coste tan alto de estos últimos gestores hace que se recurran a soluciones intermedias entre gestores de bases de datos Ofimáticas y Corporativas. Entre estas soluciones intermedias se encuentra MySQl, un gestor de bases de datos que, además de ser gratuito y sencillo, es capaz de manipular gran cantidad de datos cumpliendo prácticamente todos los estándares de la arquitectura ANSI SPARC. Además, se integra fácilmente en las típicas soluciones XAMPP que son paquetes que incluyen, además de MySQL, una versión del servidor web Apache y varios lenguajes de script (Php, Perl, …) que dotan a MySQL de potentes herramientas de acceso y publicación de los datos.