

2. Bases de datos

Como hemos visto anteriormente, los ficheros permiten organizar y almacenar conjuntos de datos del mismo tipo con una determinada estructura. Estos ficheros pueden ser utilizados por una aplicación o ser resultado. Pero podemos encontrarnos con información duplicada, incoherencia de datos, fallos de seguridad, etc ...

2.1. Conceptos

Para solucionar estos problemas aparece el concepto de **Base de Datos**, que permitirá reunir toda la información relacionada en un único sistema de almacenamiento, de manera que cualquier aplicación pueda utilizarla de manera independiente, mejorando así el tratamiento de la información y por tanto participando en la evolución del desarrollo de aplicaciones.

Base de datos: es una colección de datos relacionados lógicamente entre sí, con una definición y descripción comunes y que están estructurados de una determinada manera. Este conjunto estructurado de datos representa entidades y sus interrelaciones, almacenados con la mínima redundancia y posibilitando el acceso a ellos de manera eficiente por parte de varias aplicaciones y usuarios.

Una base de datos consta de los siguientes elementos:

Entidades □ son los objetos del mundo real de los que se almacena información en la base de datos. Por ejemplo, en una base de datos de una clínica veterinaria podríamos tener como entidades: *mascotas*, *doctores*, etc,...

Atributos □ son los datos que se almacenan de la entidad. Cualquier propiedad o característica de la entidad puede ser un atributo. Por ejemplo, en la entidad *mascotas*, podríamos tener los atributos: *especie*, *raza*, *color*, *nombre*, etc,...

Registros □ donde se almacena la información de cada entidad. Es un conjunto de atributos que contienen los datos de una entidad. Siguiendo con nuestro ejemplo, un registro podría ser: *perro*, *galgo*, *negro*, *Jack*, ...

Campos □ donde se almacenan los atributos de cada registro, en nuestro ejemplo, un campo podría ser el valor "*galgo*"

Mascotas			
Especie	Raza	Color	Nombre
Perro	Galgo	Negro	Jack

Éstas son las ventajas fundamentales del uso de las bases de datos:

- **Acceso múltiple:** varios usuarios o aplicaciones pueden acceder a la base de datos, sin que existan problemas en el acceso o los datos.
- **Utilización múltiple:** cada uno de los usuarios o aplicaciones puede disponer de una visión particular de la estructura de la base de datos, de tal manera que cada uno de ellos accederá sólo a la parte que le corresponde.
- **Flexibilidad:** la forma de acceder a la información puede ser establecida de diferentes maneras, ofreciendo tiempos de respuesta muy reducidos.
- **Confidencialidad y seguridad:** el control del acceso a los datos puede ser establecido para que unos usuarios o aplicaciones puedan acceder a unos datos y a otros no, impidiendo a los usuarios no autorizados la utilización de la base de datos.
- **Protección contra fallos:** en caso de errores en la información, existen mecanismos bien definidos que permiten la recuperación de los datos de forma fiable.
- **Independencia física:** un cambio de soporte físico de los datos (por ejemplo: el tipo de disco), no afectaría a la base de datos o a las aplicaciones que acceden a ellos.

- **Independencia lógica:** los cambios realizados en la base de datos no afectan a las aplicaciones que la usan.
- **Redundancia:** los datos se almacenan, por lo general, una única vez. Aunque si es necesario, podríamos repetir información de manera controlada.
- **Interfaz de alto nivel:** mediante la utilización de lenguajes de alto nivel puede utilizarse la base de datos de manera sencilla y cómoda.
- **Consulta directa:** existe una herramienta para poder acceder a los datos interactivamente.

2.2. Usos

¿Quién utiliza las bases de datos?

Existen cuatro tipos de personas que pueden hacer uso de una base de datos:

- **el administrador** □ encargado del control técnico de todo el sistema, es la persona encargada de la creación o implementación física de la base de datos. Es quien toma las decisiones relacionadas con el funcionamiento físico del almacenamiento de información. Establecerá la política de seguridad y el acceso para garantizar el menor número de problemas
- **los diseñadores de la base de datos** □ Son las personas encargadas de diseñar cómo será la base de datos. Llevarán a cabo la identificación de los datos, las relaciones entre ellos, sus restricciones, etc. Para ello han de conocer a fondo los datos y procesos a representar en la base de datos. Si estamos hablando de una empresa, será necesario que conozcan las reglas de negocio en la que esta se mueve. Para obtener un buen resultado, el diseñador de la base de datos debe implicar en el proceso a todos los usuarios de la base de datos.
- **los programadores de aplicaciones** □ Una vez diseñada y construida la base de datos, los programadores se encargarán de implementar los programas de aplicación que servirán a los usuarios finales. Estos programas de aplicación ofrecerán la posibilidad de realizar consultas de datos, inserción, actualización o eliminación de los mismos. Para desarrollar estos programas se utilizan lenguajes de tercera o cuarta generación.
- **los usuarios finales** □ Son los clientes finales de la base de datos, utilizan aplicaciones desarrolladas para interactuar con la base de datos.

¿Para qué se utilizan las bases de datos?

Las bases de datos se utilizan en infinidad de campos:

- **Banca:** información de clientes, cuentas, transacciones, préstamos, ...
- **Líneas aéreas:** información de clientes, horarios, vuelos, destinos, ...
- **Universidades:** información de estudiantes, carreras, horarios, materias, ...
- **Transacciones de tarjeta de crédito:** para comprar con tarjetas de crédito y la generación de los extractos mensuales.
- **Telecomunicaciones:** para guardar registros de llamadas realizadas, generar facturas mensuales, mantener el saldo de las tarjetas telefónicas de prepago y almacenar información sobre las redes.
- **Medicina:** información hospitalaria, ...
- **Justicia y Seguridad:** delinquentes, casos, sentencias, investigaciones, ...
- **Organismos públicos:** datos ciudadanos, certificados, ...
- **Sistemas de posicionamiento geográfico.**
- **Hostelería y turismo:** reservas de hotel, vuelos, ...
- **Ocio digital:** juegos online, apuestas, ...
- **Cultura:** gestión de bibliotecas, museos virtuales, ...

2.3. ARQUITECTURA DEL SGBD (ANSI/SPARC)

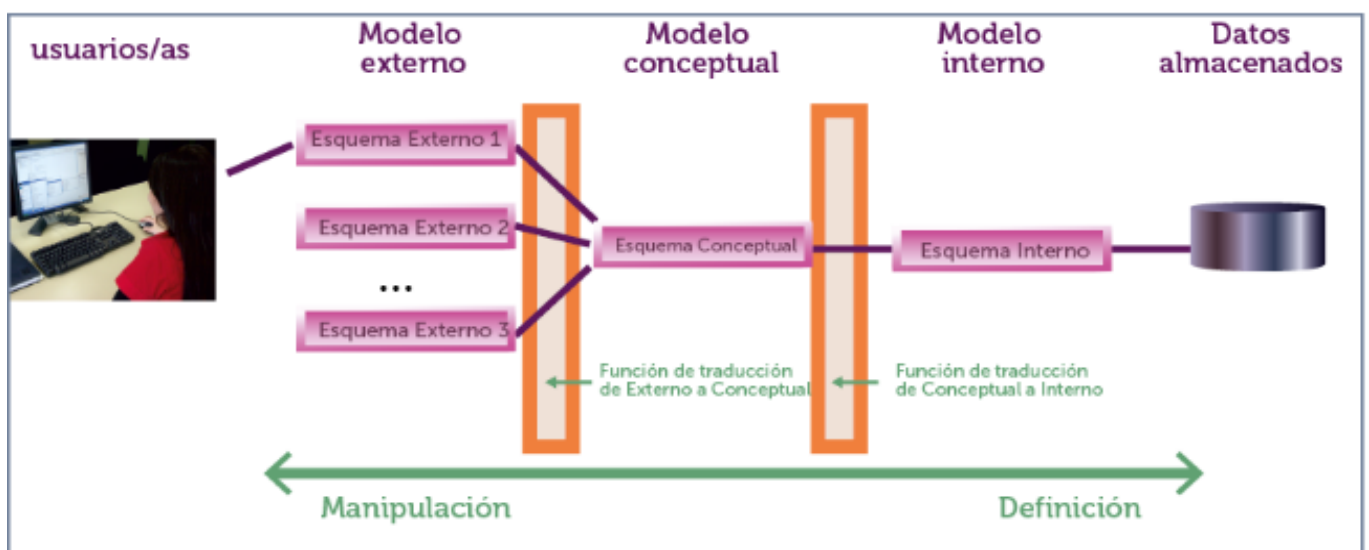
En 1975 el Organismo de Estandarización de Los Estados Unidos (ANSI) publica un informe clave para la posterior evolución de los S.G.B.D. Insiste en la necesidad de conseguir total independencia entre datos y aplicaciones; propone una arquitectura a 3 niveles y define el modelo conceptual para conseguir tales objetivos:

- **Nivel Interno o Físico:** se describe la estructura física de la base de datos mediante un esquema interno.
 - Este esquema se especifica mediante un modelo físico y describe todos los detalles para el almacenamiento de la base de datos, así como los métodos de acceso.
 - El Nivel Físico se encarga de engranar con el software más interno de cada máquina (Sistema Operativo y Sistema de Gestión de ficheros).
- **Nivel Conceptual:** se describe la estructura de toda la base de datos para una comunidad de usuarios (todos los de una empresa u organización), mediante un esquema conceptual
 - Este esquema oculta los detalles de las estructuras de almacenamiento y se concentra en describir entidades, atributos, relaciones, operaciones de los usuarios y restricciones.
 - En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar el esquema.
- **Nivel Externo o Lógico:** el nivel Lógico o externo de descripción, contiene las vistas externas de la B.D. que están asociadas cada una a un Esquema Externo y permite ver a cada tipo de usuario de la B.D. sólo aquella parte del esquema que es de su interés.

La BBDD debe presentar al usuario una visión de los datos que este sea capaz de manejar e interpretar. No necesita la misma información de la BBDD un usuario experto que desea conocer la ubicación de los archivos físicos, que un programador que desarrolla una funcionalidad concreta que un usuario que consulta parte de la información (no ve lo mismo en Raices un jefe de estudios / director que ve todos los alumnos de un centro o un inspector que ve alumnos de distintos centros, un profesor que sólo puede ver sus alumnos, un alumno que se ve así mismo....)

En una base de datos determinada habrá un único esquema interno, un único esquema conceptual, pero varios esquemas en el nivel externo.

De una B.D. se pueden derivar tantas vistas como haga falta. El propósito principal de esta arquitectura a 3 niveles es conseguir que el Esquema Conceptual sea una descripción estable de la organización e independiente de las "vistas" y de la forma de almacenamiento de los datos. De este modo se conseguiría separar los programas de aplicación de la base de datos física



- La **independencia lógica**: es la capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación. Se puede modificar el esquema conceptual para ampliar la base de datos o para reducirla. Si, por ejemplo, se reduce la base de datos eliminando una entidad, los esquemas externos que no se refieran a ella no deberán verse afectados.
- La **independencia física**: es la capacidad de modificar el esquema interno sin tener que alterar el esquema conceptual (o los externos). Por ejemplo, puede ser necesario reorganizar ciertos ficheros físicos con el fin de mejorar el rendimiento de las operaciones de consulta o de actualización de datos. Dado que la independencia física se refiere sólo a la separación entre las aplicaciones y las estructuras físicas de almacenamiento, es más fácil de conseguir que la independencia lógica.

La independencia de datos se consigue porque al modificarse el esquema en algún nivel, el esquema del nivel inmediato superior permanece sin cambios, sólo se modifica la correspondencia entre los dos niveles. No es preciso modificar los programas de aplicación que hacen referencia al esquema del nivel superior.

La mayoría de los SGBD no distinguen del todo los tres niveles. Algunos incluyen detalles del nivel físico en el esquema conceptual. En casi todos los SGBD que manejan vistas de usuario, los esquemas externos se especifican con el mismo modelo de datos que describe la información a nivel conceptual, aunque en algunos se pueden utilizar diferentes modelos de datos en los niveles conceptual y externo.

3. Sistemas gestores de bases de datos

El sistema gestor de bases de datos, o SGBD actúa como interfaz entre los datos y las aplicaciones que utilizan esos datos, y proporciona mecanismos para definirlos, almacenarlos de manera óptima, mantenerlos y manipularlos. Además ofrece potentes herramientas para la administración orientada a la gestión de usuarios y el almacenamiento de los datos, gestión de instancias, control de transacciones, recuperación y copias de seguridad.

Las principales funciones de un SGBD son:

- **Definición y control centralizado**: definición de los datos y las relaciones entre las estructuras que los almacenan.
- **Manipulación de los datos**: mecanismos para consultar y modificar datos almacenados o añadir otros nuevos.
- **Seguridad e integridad**: gestión de accesos autorizados y control de las restricciones para garantizar la integridad y consistencia de los datos
- **Tareas de administración**: copias de seguridad, recuperación ante fallos, roles, control de privilegios, etc
- **Control de recurrencia**: garantiza el acceso simultáneo a los datos y la veracidad de éstos

3.1. Tipos de SGBD Comerciales vs Libres

Los sistemas gestores de bases de datos pueden ser libres y no libres, e independientemente de eso, ser gratuitos o no. Por ejemplo, Microsoft SQL Server es un SGBD privativo pero cuenta con alguna versión gratuita como la Compact Edition Básica.

Por regla general, se suele usar el término libre como referencia a gratuito, y comercial para dar a entender un producto de pago. esta forma de diferenciarlas no es correcta, ya que muchos de los programas de código libre cuentan con versiones de pago, o proporcionan un servicio de soporte por el que hay que pagar. No por esto deja de ser código libre, y con los SGBD libres sucede exactamente igual.

Dentro de los SGBD es muy habitual que estos programas sean gratuitos hasta un determinado uso y que haya que pagar a partir de un momento.

La decisión final debe estar basada en las características del sistema de información de la organización y sus propios requisitos.

3.2. SGBD Relacionales Comerciales

SGBD	Descripción
Oracle	Reconocido como uno de los mejores a nivel mundial. Es multiplataforma, confiable y seguro. Es Cliente/Servidor. Basado en el modelo de datos Relacional. De gran potencia, aunque con un precio elevado hace que sólo se vea en empresas muy grandes y multinacionales. Ofrece una versión gratuita para educación y no sé si comercial hasta un determinado uso.
MySQL	Sistema muy extendido que se ofrece bajo dos tipos de licencia, comercial o libre. Para aquellas empresas que deseen incorporarlo en productos privativos, deben comprar una licencia específica. Es Relacional, Multihilo, Multiusuario y Multiplataforma. Su gran velocidad lo hace ideal para consulta de bases de datos y plataformas web.
DB2	Multiplataforma, el motor de base de datos relacional integra XML de manera nativa, lo que IBM ha llamado pureXML, que permite almacenar documentos completos para realizar operaciones y búsquedas de manera jerárquica dentro de éste, e integrarlo con búsquedas relacionales.
INFORMIX	Otra opción de IBM para el mundo empresarial. Es un gestor de base de datos relacional basado en SQL. Multiplataforma. Consume menos recursos que Oracle, con utilidades muy avanzadas respecto a conectividad y funciones relacionadas con tecnologías de Internet/Intranet, XML, etc.
Microsoft SQL SERVER	Sistema Gestor de Base de Datos producido por Microsoft. Es relacional, sólo funciona bajo Microsoft Windows, utiliza arquitectura Cliente/Servidor. Constituye la alternativa a otros potentes SGBD como son Oracle, PostgreSQL o MySQL.
SYBASE	Un DBMS con bastantes años en el mercado, tiene 3 versiones para ajustarse a las necesidades reales de cada empresa. Es un sistema relacional, altamente escalable, de alto rendimiento, con soporte a grandes volúmenes de datos, transacciones y usuarios, y de bajo costo.

3.3. SGBD Relacionales Libres

SGBD	Descripción
MySQL	Es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Distribuido bajo dos tipos de licencias, comercial y libre. Multiplataforma, posee varios motores de almacenamiento, accesible a través de múltiples lenguajes de programación y muy ligado a aplicaciones web.
PostgreSQL	Sistema Relacional Orientado a Objetos. Considerado como la base de datos de código abierto más avanzada del mundo. Desarrollado por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyados por organizaciones comerciales. Es multiplataforma y accesible desde múltiples lenguajes de programación.
Firebird	Sistema Gestor de Base de Datos relacional, multiplataforma, con bajo consumo de recursos, excelente gestión de la concurrencia, alto rendimiento y potente soporte para diferentes lenguajes.
Apache Derby	Sistema Gestor escrito en Java, de reducido tamaño, con soporte multilenguaje, multiplataforma, altamente portable, puede funcionar embebido o en modo cliente/servidor.
SQLite	Sistema relacional, basado en una biblioteca escrita en C que interactúa directamente con los programas, reduce los tiempos de acceso siendo más rápido que <i>MySQL</i> o <i>PostgreSQL</i> , es multiplataforma y con soporte para varios lenguajes de programación.

4. Ciclo de vida de una base de datos

El ciclo de vida de las BBDD están totalmente integrado en la ingeniería del software y sus fases están relacionadas con ellas.

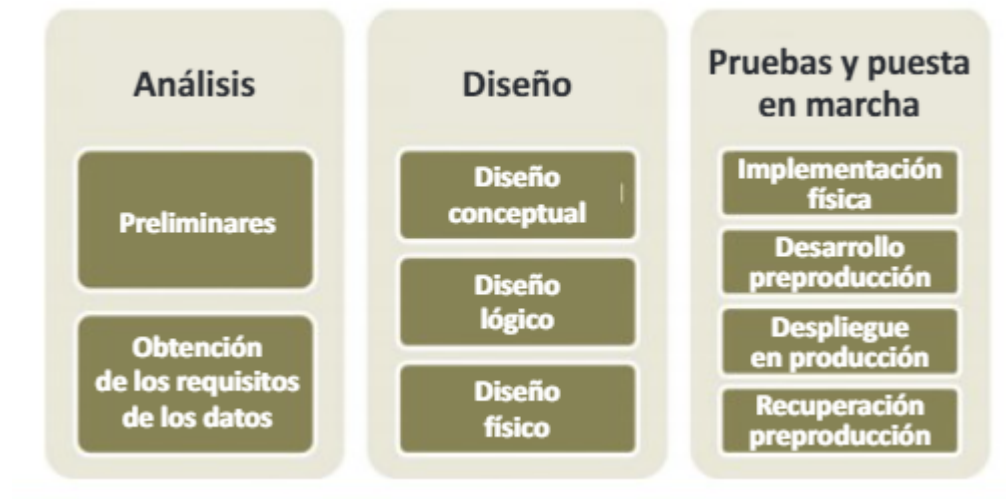
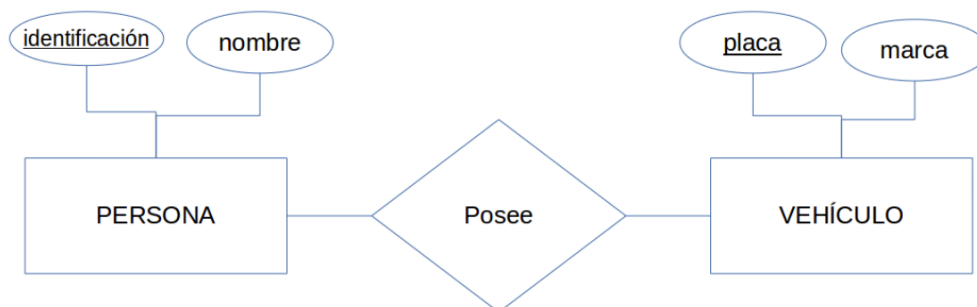


Imagen 1. Ciclo de vida de las BBDD

Podemos distinguir las siguientes fases:

- **Planificación:** qué vamos a hacer, con qué recursos contamos, qué técnicas se usan para la recopilación de los datos, formato, cómo se hará el diseño, ...
- **Definición:** se especifica el ámbito y los límites de aplicación de la base de datos, con qué sistemas actúa, se determinan los usuarios,...
- **Recogida y análisis de requisitos:** se recogen y analizan los requerimientos de los usuarios y de las áreas de aplicación, a través de entrevistas, observación del funcionamiento de la organización, ...
- **Diseño de la base de datos:** está dividido en 3 partes
 - o **Diseño conceptual:** es un esquema conceptual que parte de los requisitos del usuario y es independiente del SGBD



- o **Diseño lógico:** tras refinar el esquema conceptual, se obtiene el esquema lógico, en esta fase debemos pensar cómo normalizar nuestras tablas para evitar duplicar información y ahorrar espacio. También es independiente del SGBD

Persona	Posee	Vehículo
Identificación		Placa
Nombre		marca

- o **Diseño físico:** En esta última fase ya debemos ver qué tipos de datos utilizaremos, sus dominios (qué valores va a permitir), qué índices debemos crear para optimizar las consultas, entre otros. Aquí ya escribimos nuestro SQL para plasmar todo nuestro diseño en el SGBD elegido.

```
CREATE TABLE PERSONA (  
  Identificación INT,  
  Nombre varchar(150)  
)
```

- **Diseño de la aplicación:** diseño de los programas que interactuarán con la base de datos
- **Prototipado:** un prototipo que el usuario podrá utilizar para ver qué aspectos son o no adecuados. Este paso ayuda a detectar posibles errores en la toma de requisitos.
- **Implementación:** usando el lenguaje de definición de datos se crea el esquema de la base de datos, se implementan procesos, etc,...
- **Conversión y carga de datos:** puede que la base de datos deba crearse por la actualización de una aplicación, debemos reemplazar los datos antiguos por los nuevos incluidos en distintas estructuras
- **Prueba y mantenimiento:** se prueban y validan los requisitos del usuario, se diseñan test con datos reales y se toman medidas sobre la fiabilidad y calidad del software desarrollado.