

BD

Bases de Datos

UD 1

Sistemas de almacenamiento de información

ÍNDICE

1. Sistemas lógicos de almacenamiento
2. Bases de datos
3. Modelos de bases de datos
 - 3.1 Modelo jerárquico o en árbol
 - 3.2 Modelo en red
 - 3.3 Modelo relacional
 - 3.4 Modelo orientado a objetos
 - 3.5 Modelo no relacional (NoSQL)
4. Sistemas Gestores de Bases de Datos (SGBD)
 - 4.1 Arquitectura
 - 4.2 Elementos
 - 4.3 Características
 - 4.4 ¿Qué SGBD elijo?
 - 4.5 Componentes
 - 4.5.1 Lenguajes de los SGBD
 - 4.5.2 Diccionario de Datos
 - 4.5.3 Seguridad
 - 4.5.4 Usuarios

1. Sistemas lógicos de almacenamiento

En cualquier actividad económica es necesario tomar decisiones. Para tomar decisiones acertadas se requiere manejar una buena información que se obtendrá a partir de los **datos**. Cuando los datos se organizan y se tratan de obtiene **información**. Para manejar los datos con eficacia utilizaremos una **base de datos**, que nos ayudará a **almacenar y procesar esos datos, extraer la información** necesaria y **tomar decisiones**.

Por ejemplo,

- *En una consulta médica se guardaban en carpetas las historias clínicas de los pacientes*
- *En una empresa se guardaban en carpetas las facturas y albaranes tanto de clientes como de proveedores*

Para almacenar la información de modo permanente se utilizan **dispositivos de almacenamiento** masivo denominados memoria secundaria, ya que los datos guardados en la memoria principal desaparecen al desconectar el ordenador. La información contenida en los dispositivos de almacenamiento se estructura en unidades denominadas **ficheros o archivos (en inglés “file”)**.

Para entender mejor el funcionamiento del sistema de ficheros empezaremos por conocer la terminología básica con el fichero de clientes de una empresa:

FICHERO DE CLIENTES					
Campos:	DNI	NOMBRE	APELLIDOS	DIRECCION	TELEFONO
Registro 1:	73564765M	Javier	Barquín Arce	C/ Alta, 234	918342156
Registro 2:	56558765W	Luis	Gómez de Miguel	Avda. de Castilla, 2A	956235567
Registro 3:	13874521M	María Belén	Márquez Ruiz	C/ <u>Floranes</u> , 2	568732212
Registro 4:	75675317R	Carmen	Rodríguez Mata	Paseo Pereda, 123	942665544

Los **dispositivos de almacenamiento** masivo según la forma de acceder a la información se clasifican en:

- **Dispositivos secuenciales:** la información se guarda en posiciones consecutivas, de forma que para acceder a un dato hay que recorrer los datos anteriores. Por ejemplo, la cinta magnética.
- **Dispositivos direccionables:** permiten el acceso directo a los datos. En estos dispositivos el espacio destinado a almacenamiento está dividido en segmentos direccionables de forma individual. Por ejemplo, el disco duro.

Los ficheros que almacenan datos necesitan disponer de **acceso a un registro**, es decir, disponer de un procedimiento para seleccionarlo. Este acceso está condicionado por el tipo de soporte en el que se encuentre almacenada la información. Los tipos de acceso son:

- **Secuencial:** los registros se leen uno detrás de otro desde el principio del fichero hasta localizar el registro buscado o hasta el final del fichero.
- **Directo:** permite seleccionar a un registro sin tener que leer los anteriores, accediendo directamente a él mediante su clave.
- **Indexado:** para seleccionar un registro consultamos previamente de forma secuencial en una tabla que contiene la clave más alta y la dirección de comienzo de cada bloque de registros. Una vez localizado se utiliza el acceso directo a ese bloque de registros y, dentro del bloque, la lectura secuencial hasta localizarlo.
- **Dinámico:** permite tanto el acceso directo y como el indexado.

Tipos de ficheros y formatos

Dependiendo de la información almacenada en un fichero, será de un tipo u otro según el contenido que almacena. Por ejemplo, tendremos ficheros de imágenes, de vídeos, de sonidos, de textos, de datos, etc.

Dependiendo del formato del fichero, se suele añadir una extensión a los nombres de los ficheros. Tenemos extensiones de archivos como *txt* para ficheros en texto plano, *docx* para documentos de texto Word, *avi* para vídeo o *mp3* para audio.

2. Bases de Datos

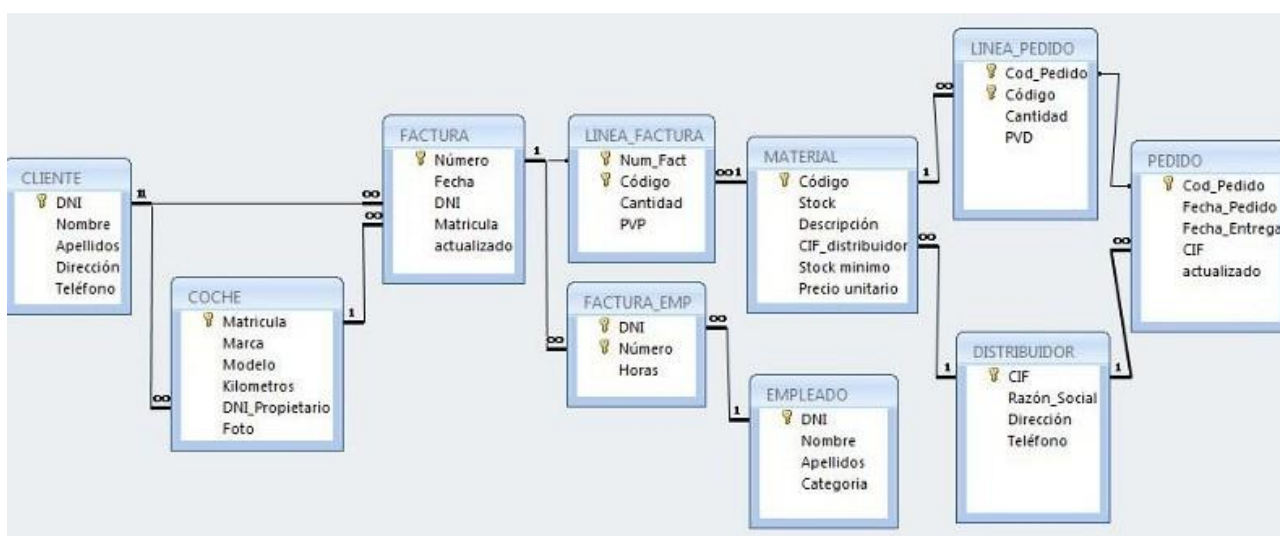
Una **base de datos** es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Las ventajas que aportan las bases de datos sobre los sistemas de ficheros son:

- **Control sobre la redundancia de datos.** todos estos ficheros están integrados, por lo que no se almacenan varias copias de los mismos datos.
- **Consistencia de datos.** Cada dato solo está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez, y está disponible para todos los usuarios inmediatamente.
- **Más información sobre la misma cantidad de datos.** Al estar todos los datos integrados, se puede extraer información adicional sobre los mismos.
- **Compartición de datos.** la base de datos pertenece a la empresa y puede ser compartida por todos los usuarios que estén autorizados. Las nuevas aplicaciones que se vayan creando pueden utilizar la base de datos existente.
- Simplifica el esfuerzo de **programación y mantenimiento de los programas.**

3. Modelos de Bases de Datos

Un **modelo de datos** es un conjunto de herramientas conceptuales que permiten describir los datos, sus relaciones y las reglas de integridad que deben cumplir.



Representación gráfica del esquema de una base de datos en la ventana relaciones de Microsoft Access

Este esquema se especifica durante el diseño, y no es de esperar que se modifique a menudo. Sin embargo, los datos que se almacenan en la base de datos pueden cambiar con mucha frecuencia: se insertan datos, se actualizan, se borran, etc.

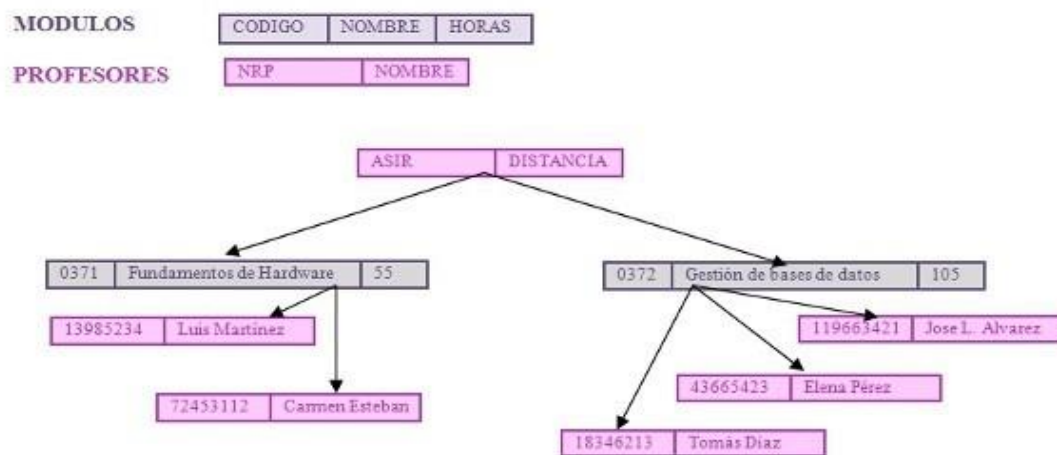
Para realizar un esquema más cercano al diseño de la base de datos, se utilizan diferentes modelos de datos según sea el SGBD donde vayamos a implementarlo.

En su evolución las bases de datos se han basado en 3 modelos de datos.

3.1 Modelo jerárquico o en árbol

Dentro de la jerarquía el nivel superior se percibe como el "padre" de los registros situados debajo de él, de forma que:

- Cada padre puede tener muchos hijos
- Cada hijo sólo tiene un padre

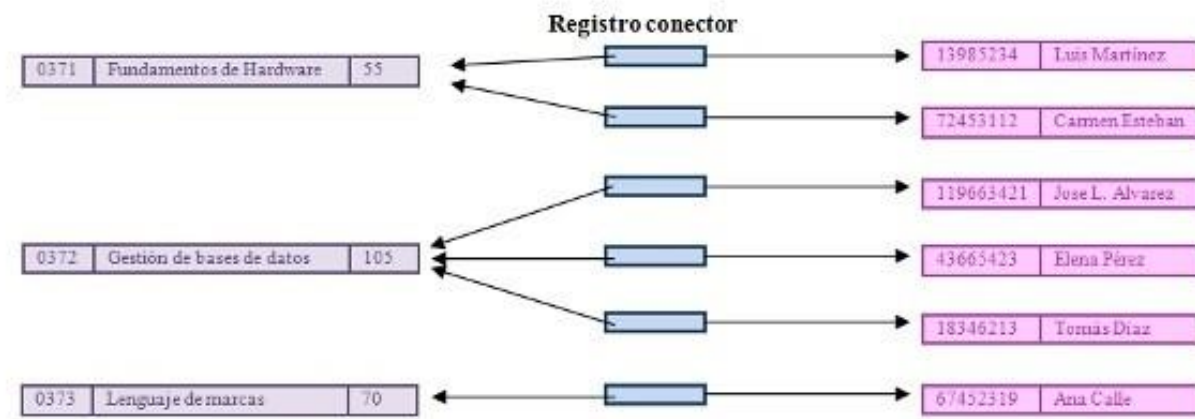


Algunos de sus inconvenientes son:

- **La ejecución es compleja**, obliga a tener conocimientos en cuanto a las características del almacenamiento físico.
- **Difícil de administrar**, si se hace algún cambio en la ubicación de los registros habrá que modificar todas las aplicaciones que accedan a la base de datos
- **No tiene independencia estructural**, puesto que la navegación por los registros se hace siguiendo una ruta de acceso (padre-hijo, de izquierda a derecha, etc.) y si se hacen cambios en la estructura los programas no funcionarán.
- **Complejidad a la hora de programar aplicaciones**, es necesario que tanto los programadores como los usuarios sepan cómo están distribuidos los datos para acceder a ellos.

3.2 Modelo en red

Es un modelo propio de los sistemas comerciales de los años 70. Fue **creado para representar relaciones más complejas y eficientes que las del modelo jerárquico** y así imponer un estándar de bases de datos que ayudara a los diseñadores y a los programadores. Se basan en el uso de punteros llamados “registros conectores” que permiten conectar dos o más registros entre sí.



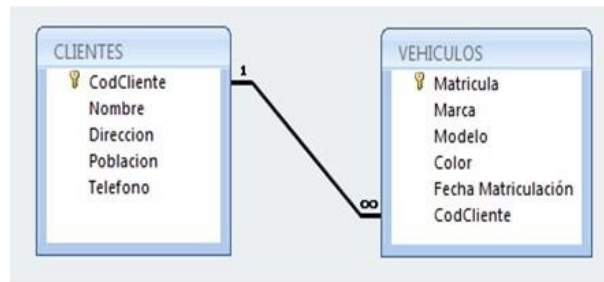
Algunos de sus inconvenientes son:

- **Complejidad del sistema:** el acceso a los datos se hace leyendo un registro cada vez, por tanto, los programadores y los usuarios finales deben conocer las estructuras internas, por tanto, no es fácil de utilizar. Además tiene un consumo de memoria alto a consecuencia del uso de los registros conectores.
- **Falta de independencia estructural:** si se hacen cambios en la estructura de la base de datos, es necesario cambiar las aplicaciones. Aunque logra la independencia de los datos, no produce independencia estructural.

3.3 Modelo relacional

Fue desarrollado por Codd para IBM en los años 70, pero inicialmente los ordenadores carecían de prestaciones para poder ejecutarlo. Actualmente es el modelo más utilizado para modelar problemas reales y administrar datos dinámicamente.

Su principal ventaja es que permite al usuario y al diseñador operar en un entorno que se percibe como un **conjunto de tablas**, mientras que los detalles físicos más complejos los maneja el sistema. Este es el modelo que se estudiará a lo largo de todo el curso.



CLIENTES				
CodCliente	Nombre	Dirección	Población	Teléfono
1	Francisco Álvarez	C/ La Mata 9.	Alcántara	925767788
2	Raquel Marcos	C/ La Amapola 7.	Toledo	925998811
3	Carlos Revuelta	C/ La Arboleda 12.	Madrid	919090771
4	José María Gabello	C/ La plaza 72	Madrid	914409071
5	Jorge Peña	C/ Fresno, 23	Guadalajara	949788896
6	Dolores Manzano	C/ Autonomía, 8	Madrid	916767560

VEHICULOS					
Matricula	Marca	Modelo	Color	Fecha Matriculación	CodCliente
4534 FNG	Ford	Focus	Negro	14/04/2007	2
1203 CLL	Cotroën	C4	Magenta	23/08/2005	5
3367 GHB	Suzuki	Vitara	Cobalto	16/05/2009	2
1004 JLG	Kia	Rio	Rojo	02/07/2009	6
6709 BFG	Peugeot	206	Gris plata	12/10/2006	3

3.4 Modelo Orientado a Objetos

Los modelos de bases de datos intentan representar cada vez con más fidelidad los problemas del mundo real que cada vez son más complejos. Uno de los modelos que se han desarrollado recientemente (años 90) es el modelo de bases de datos orientada a objetos. Se denomina así porque **su estructura básica es un objeto**, que recoge tanto datos como sus relaciones.

Algunos de los inconvenientes son:

- En la actualidad, **no existen estándares de este modelo, por lo que no se está utilizando prácticamente**
- El método de acceso se parece al jerárquico y en red.
- El modelado y ejecución es difícil debido a que tienen mucho contenido semántico y no hay estándares.
- La complejidad y elevados requerimientos del sistema hace que las transacciones sean lentas.

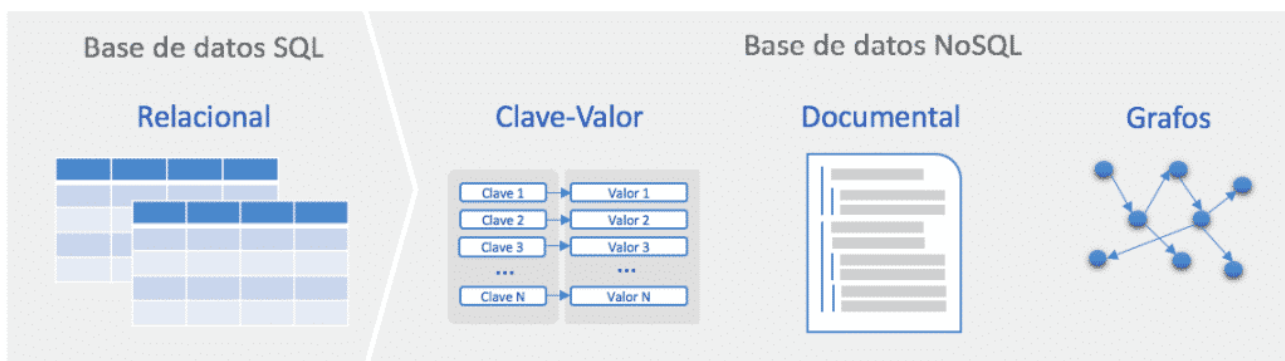
3.5 Modelo No Relacional (NoSQL)

La evolución de los sistemas de bases de datos ha ido siempre marcada por la búsqueda de nuevas herramientas para reflejar el mundo real lo mejor posible.

Cada modelo ha ido sustituyendo al anterior intentando eliminar sus defectos: el modelo de red sustituyó al modelo jerárquico, el modelo relacional reemplazó al modelo de red.

A medida que las aplicaciones se van volviendo más complejas aparece el modelo de datos orientado a objetos y el modelo relacional ampliado que incluye muchas características del modelo orientado a objetos. Ambos intentan recoger la mayor información semántica posible, pero desde orientaciones distintas: el modelo orientado a objetos se enfoca más a aplicaciones de ingeniería y científicas muy especializadas y el modelo relacional ampliado se dirige más a aplicaciones de negocios.

Sin embargo, el mercado de las bases de datos ha cambiado sustancialmente con el creciente desarrollo del uso de internet. Ahora los esfuerzos se dirigen a la creación y desarrollo de bases de datos que se comuniquen fácilmente por internet. Algunas de ellas son denominadas **NoSQL** (Not Only SQL) y fueron creadas para su uso en redes sociales, criptomonedas, o tecnologías como Blockchain.



4. Sistemas Gestores de Bases de Datos (SGBD)

Un **Sistema Gestor de Base de Datos (SGBD)** es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos. Es decir, esencialmente es un programa que agrega funcionalidad para gestionar la base de datos, la cual podrá seguir cualquier modelo o sistema de almacenamiento de los que se han visto a lo largo de la unidad.

4.1 Arquitectura de un SGBD

Los SGBD se basan en la **arquitectura ANSI/X3/SPARC**, la cual establece **tres niveles de abstracción** que todo SGBD debe cumplir. Los niveles establecen distintos grados de abstracción sobre el mismo conjunto de datos. Estos 3 niveles de abstracción, desde el más cercano a la máquina hasta el más cercano al usuario final, son:

Nivel Físico o Interno

Acceden los Administradores de Sistemas o Diseñadores

Es el nivel más bajo de abstracción y en el que existe realmente la base de datos. En él se describen cómo se almacenan los datos y la estructura de los mismos.

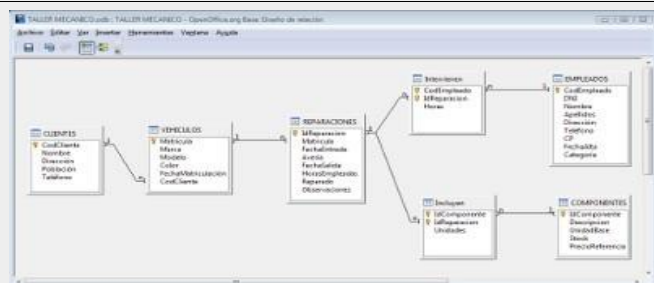
Se encuentra en el Servidor de Bases de Datos



Nivel Lógico o Conceptual

Acceden los Administradores de las Bases de Datos

En él se describe cuáles son los datos que se almacenan y qué relaciones se establecen entre ellos, representando de forma completa la base de datos con estructuras simples.



Se encuentra en el Servidor de Bases de Datos

Nivel de Usuario o Externo

Acceden los Usuarios de las Bases de Datos

Está formado por el conjunto de las vistas de usuario, siendo una vista de usuario la fracción de la base de datos que ve un usuario en concreto. En este nivel donde trabajan los programas de aplicación de usuario.

Se encuentra en los programas o aplicaciones de los Equipos Cliente (pueden ser remotos)

4.2 Elementos básicos de un SGBD

Los elementos principales de los SGBD son las **tablas** que contienen campos organizados en registros. Una **tabla** es un conjunto de registros de la misma estructura referentes a una misma entidad.

Por ejemplo, tenemos una tabla denominada CLIENTES en una base de datos donde se almacenan los siguientes datos de cada cliente: NIF, NOMBRE, FECHA DE NACIMIENTO, DIRECCIÓN, TELÉFONO, EMAIL.

Cada campo almacena un dato y para ello debemos indicarle al SGBD de qué tipo es. El **tipo de dato** indica la naturaleza del campo. Números, Textos o Fechas son algunos tipos de datos disponibles en los SGBD.

Por lo tanto, se puede tener datos **numéricos**, con los que se pueden realizar cálculos aritméticos (sumas, restas, multiplicaciones...), **cadenas de caracteres** que contienen caracteres alfabéticos y dígitos numéricos, o **fechas** que contienen tres datos numéricos, representando el día, el mes y el año de esa fecha.

Para poder identificar un registro en una tabla necesitamos un campo que no se repita. El **campo clave** es un campo especial que **identifica de forma única** a cada registro.

En nuestro ejemplo anterior, el NIF es único para cada persona, y por tanto será el campo clave de la tabla CLIENTES.

Cuando ordenamos, buscamos o filtramos registros de una tabla necesitamos procesar una gran cantidad de datos. El **índice** es una estructura que almacena uno o varios campos de una tabla organizándolos para hacer más fácil encontrar y ordenar los registros de esa tabla.

El índice tiene un funcionamiento similar al índice de un libro. Cuando deseamos encontrar un registro con la información de un campo indexado, el índice conoce su posición en la base de datos. Por lo tanto, para buscar un elemento que esté indexado, solo hay que buscar primero en el índice dicho elemento y una vez encontrado, devolver el registro que se encuentre en la posición marcada por el índice.

Para facilitar el procesamiento de la información, los SGBD disponen también de la posibilidad de almacenar:

- **Consultas:** instrucciones que nos permiten realizar peticiones a una base de datos sobre registros almacenados en una o varias tablas.
- **Informes:** listados ordenados y clasificados de campos y registros seleccionados para que sean fáciles de entender.
- **Vistas:** transformaciones, completas o parciales, de una o varias tablas para obtener una nueva tabla como combinación de sus registros.

Cuando se procesan datos de un SGBD, generalmente necesitamos realizar operaciones sobre registros. Para ello los SGBD disponen de:

- **Guiones o scripts:** son un conjunto de instrucciones, que, ejecutadas de forma ordenada, realizan operaciones avanzadas de mantenimiento de los datos almacenados en una base de datos.
- **Procedimientos y funciones:** son tipos especiales de scripts que están almacenados en la base de datos y que forman parte de su esquema.
- **Disparadores o Triggers:** son tipos especiales de scripts que están almacenados en la base de datos y que se ejecutan cuando se produce una acción concreta con un registro en una tabla.

4.3 Características de los SGBD

Entre las distintas categorías posibles destacan las que citamos a continuación:

Modelo lógico empleado: Define la estructura de datos que se va a implementar. Según este criterio tenemos las siguientes clases:

- Modelo Jerárquico / en Red
- Modelo Relacional / Orientado a Objetos.
- Modelo de BD e Internet (NoSQL).

Número de usuarios concurrentes: Según la cantidad de usuarios que pueden estar conectados al tiempo, los clasificamos en:

- Monousuario.
- Multiusuario.

Precio de la licencia:

- Gratuitos.
- Comerciales.

Número de plataformas soportadas: Hace referencia a la cantidad de sistemas operativos en los que se puede instalar.

- Monoplataforma.
- Multiplataforma.

Acceso al código fuente:

- Libres: Permiten el acceso al código fuente.
- Privativos: No permiten el acceso al código fuente.

Estructura geográfica de la empresa:

- Centralizados.
- Distribuidos.

Ámbito de la aplicación:

- Propósito general.
- Propósito específico.

En base a esto, tendremos que, por ejemplo, Postgres es un SGBD de código libre, multiplataforma, multiusuario, gratuito y que se basa en un modelo relacional, aunque también admite un modelo orientado a objetos.

4.4 ¿Qué SGBD elijo?

A la hora de escoger un SGBD, el primer aspecto a tener en cuenta es centrar las **necesidades reales** y determinar qué es lo que mejor se ciñe a las particularidades del entorno de funcionamiento. Por ejemplo: si deseo gestionar una aplicación de biblioteca personal, puede ser suficiente usar OpenOffice Base o MS Access como SGBD, e incluso una simple hoja de cálculo puede cubrir suficientemente la necesidad descubierta. Por tanto, no hay que escoger un SGBD porque sea el más novedoso o el que más nos guste, sino porque es el que mejor se adapta a nuestras necesidades.

Uno de los factores a considerar es el coste, pero a pesar de ser un factor importante no debería ser tan prioritario, ya que el día a día nos demuestra que el rendimiento, las horas de trabajo para su mantenimiento o simplemente que no se estropee, es mucho más determinante para la empresa.

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 Basic.

Por regla general, se suele usar el término libre como referencia a gratuito (porque en inglés las dos palabras son “free”), y comercial para dar a entender un producto de pago. Por lo que hemos visto hasta ahora, 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.

Como se ha dicho al principio, la decisión final debe estar basada en las características del sistema de información de la organización y sus propios requisitos.

4.5 Componentes

Los SGBD son paquetes de software complejos que deben proporcionar una serie de servicios que permiten almacenar y explotar los datos de forma eficiente.

A continuación, se describen los componentes principales.

4.5.1 Lenguajes de los SGBD

Todos los SGBD ofrecen lenguajes apropiados a cada tipo de usuarios: administradores, diseñadores, programadores de aplicaciones y usuarios finales. Los lenguajes que intervienen en un SGBD se clasifican en:

- **DCL (Data Control Language):** es un lenguaje proporcionado por el SGBD que incluye una serie de comandos que permiten al administrador controlar el acceso a los datos contenidos en la Base de Datos.
- **DDL (Data Definition Language):** se utiliza para definir el esquema conceptual y el esquema interno de la base de datos: los objetos de la base de datos, las estructuras de almacenamiento y las vistas de los distintos usuarios. Lo emplean los diseñadores de la base de datos y los administradores.
- **DML (Data Modification Language):** se utiliza para consultar y actualizar los datos de la base de datos. Lo emplean los usuarios para consultar, insertar, modificar o borrar datos en una base de datos. A menudo estas sentencias están embebidas en un lenguaje de alto nivel llamado lenguaje anfitrión

4.5.2 El diccionario de datos

El **diccionario de datos** es una guía donde se describe la base de datos con todos los objetos que la forman. Se dice también que contiene "metadatos" porque es información sobre los datos. También se le suele denominar **esquema de datos**.

Contiene las características lógicas como: nombre, descripción, alias, contenido y organización, donde se almacenan los datos del sistema. Identifica los procesos donde se emplean los datos y los sitios donde se necesita acceder a la información.

4.5.3 Seguridad e integridad de los datos

El componente de **Seguridad e integridad de los datos** consiste en una serie de mecanismos que proporciona el SGBD para garantizar un acceso correcto, seguro y eficiente a los datos. Se hace mediante un componente software que se encarga de:

- Garantizar que el acceso a los datos se permita solo a los usuarios autorizados.
- Disponer de herramientas para planificar y realizar copias de seguridad y restauración.
- Realizar los procedimientos necesarios para recuperar los datos tras un fallo o pérdida temporal.
- Ofrecer mecanismos para implantar restricciones de integridad que los datos deberán cumplir.
- Controlar el acceso concurrente de varios usuarios a los datos sin que se pierda la consistencia

4.5.4 Usuarios del SGBD

Existen distintos tipos de usuarios que acceden al sistema. En general, los podemos agrupar en:

- **Administradores de Sistema o Diseñadores:** Los diseñadores planifican y desarrollan las bases de datos. Definen el esquema lógico y físico de la base de datos, optimizando el almacenamiento y generando la documentación de análisis necesaria para los programadores.
- **Administradores de BD (DBA – DataBase Administrator):** Una vez creada la BD, los administradores gestionan la seguridad (usuarios y permisos), y la integridad de los datos asegurando que las transacciones sean correctas y no se pierdan datos. También se ocupan de crear las copias de seguridad. Tienen el máximo nivel de acceso y utilizan fundamentalmente DDL. Su objetivo es que la base de datos está siempre disponible y con un rendimiento óptimo.
- **Usuarios programadores:** Realizan aplicaciones sobre los objetos de la base de datos para facilitar su trabajo a los usuarios finales. Emplean DDL, DML y cualquier lenguaje anfitrión.
- **Usuarios finales:** Podrán acceder a la información sobre la que le hayan sido concedidos privilegios.