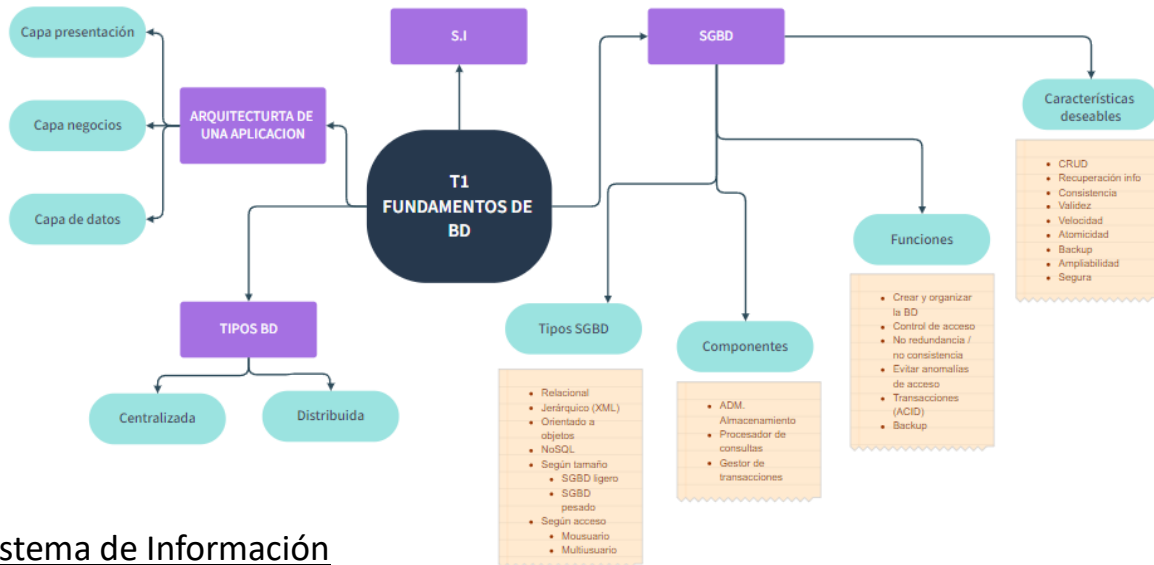


[Fundamentos de Base de Datos]



1. Sistema de Información

Es un conjunto de elementos destinados al tratamiento y administración de datos e información, organizados y listos para su posterior uso, generados para cubrir una necesidad (objetivo).

2. Sistema de gestor de base de datos

Es una herramienta software que permite la creación y gestión de una base de datos. El **SGBD** es la herramienta que se encarga de organizar esos ficheros manteniendo la información siempre accesible para el usuario de la forma más eficiente posible.

Base de datos: Una Base de Datos es un conjunto de ficheros interrelacionados entre sí, creados por un SGBD, que deben ser gestionados y accedidos a través de dicha herramienta.

Fichero: Es la unidad mínima de almacenamiento.

Características deseables:

- **CRUD:** Hace referencia a las 4 operaciones básicas que cualquier SGBD debería proporcionar: **Create** (crear), **Update** (modificar), **Read** (leer) y **Delete** (eliminar).
- **Recuperación de la información:** Cualquier fragmento de información tiene que poder ser recuperado.
- **Consistencia:** Los resultados de una lectura o búsqueda deben ser consistentes de forma que si se realiza la misma búsqueda dos veces se obtenga el mismo resultado.
- **Validez:** Debe proporcionar mecanismos que permitan validar la información que se registra en la base de datos.
- **Velocidad:** La información se tiene que poder crear, modificar, leer y eliminar rápidamente.
- **Transacciones atómicas:** Consiste en realizar una serie de acciones como si se tratara de una sola operación, y sin interferir de ninguna manera en la correcta ejecución de dichas acciones.
- **Backups:** Toda la información almacenada por el SGBD debe ser persistente, es decir, no debe perderse.

- **Capacidad de extender sus características:** Es la capacidad de extender las características del SGBD de manera que a medida que pasa el tiempo se pueda incorporar nuevas características a la base de datos.
- **Seguridad:** Debe proporcionar ciertos mecanismos de seguridad que protejan la información que se almacena.

Funciones:

- Crear y organizar la base de datos
- Control de acceso. Dan soporte a un control en el acceso a la información mediante el empleo de un sistema de cuentas de usuario.
- Evitar la redundancia e inconsistencia de los datos. Se refiere a la necesidad de asegurar que un dato no se encuentra duplicado.
- Evitar anomalías en el acceso concurrente. Habrá que dar una solución que pueda ocasionar la modificación simultánea de un mismo dato por más de un usuario.
- Garantizar la correcta ejecución de las transacciones. Existen un conjunto de normas conocidas como **ACID**:
 - **Atomicidad:** Se debe asegurar si la operación se ha llevado a cabo o no.
 - **Consistencia:** Todas las operaciones que empiezan a ejecutarse deben asegurarse de que pueden terminar sin romper ninguna de las reglas de integridad de la base de datos.
 - **Aislamiento:** Si se ejecutan dos transacciones sobre los mismos datos al mismo tiempo, deben ser independientes y no deben generar ningún error.
 - **Durabilidad:** Una vez se realice la operación, ésta debe persistir en el tiempo.
- Respaldo y recuperación

Componentes:

1. **Administrador de Almacenamiento:** Controla el acceso de la información de la base de datos en el disco.
2. **Procesador de consultas:** Se encarga de recibir las peticiones de consultas y debe encontrar la manera más óptima de realizarlas a través de lo que se conoce como **Plan de Ejecución**.
3. **Gestor de transacciones:** Asegura que se mantenga la integridad de la base de datos tras la ejecución de transacciones, sean o no fallidas.

Tipos:

- **Relacional**

Es actualmente el modelo de Base de Datos más extendido, basado en el concepto de tablas y las relaciones entre ellas como forma de relacionar la información entre sí. Destacan: Access, Oracle, Microsoft SQL Server y MySQL.

Los SGBD relacionales proporcionan una serie de funcionalidades, entre las que destacan:

1. **Tipos de datos:** Cada columna tiene un tipo de dato definido de manera que no permite almacenar valores de otro tipo de dicha columna.
2. **Restricciones:** Es posible definir restricciones que obliguen a cumplir una serie de requisitos a los valores que se almacenen.

3. **Integridad referencial:** Al momento de registrar un dato nuevo que deba estar relacionado con otro, se comprobará si el segundo existe antes de permitir el registro.

4. Consultas complejas

Son apropiadas si	No son apropiadas si
Se realizan consultas muy complejas	En contadas situaciones donde las necesidades de rendimiento son muy elevadas
Es necesario validar la información entre tablas	
Se permite que un dato pueda tener cualquier número de valores	
Es necesario construir nuevas consultas que no habían sido planificadas	

- **Jerárquico (XML)**

Es un lenguaje para almacenar información de forma jerárquica. Es un lenguaje que no proporciona ninguna forma de crear, buscar, modificar o validar información, sin embargo, se usa para almacenar, transferir y recuperar datos jerárquicos.

Son apropiadas si	No son apropiadas si
Los datos son jerárquicos	Los datos no son jerárquicos
Las herramientas XML proporcionan lo que necesitas	Es necesario realizar validaciones complicadas
	Es necesario relacionar de alguna manera la información
Se utiliza aplicaciones software que entienden el formato XML	La BD es muy grande como para reescribir el fichero completo cada que se realiza algún cambio

- **Orientado a objetos**

En este modelo todos los elementos de la Base de Datos se representan como objetos y siguen el paradigma de la Programación Orientada a Objetos.

Son apropiadas si	No son apropiadas si
Se está programando con algún lenguaje de programación orientado a objetos	Es necesario comunicarse con herramientas externas para almacenar cierta información
	Es necesario realizar consultas complejas
No es necesario realizar consultas muy complejas	La aplicación no se va a programar con un lenguaje orientado a objetos
	Es necesario realizar ciertas validaciones que este tipo de BD no proporciona

- **Objeto-Relacional**

Aparecen como una extensión al modelo relacional, por lo que los datos se almacenan en forma de tablas, y además, permiten la utilización de ciertas características propias de la Programación Orientada a Objetos.

Son apropiadas si	No son apropiadas si
La arquitectura de la aplicación está orientada a objetos	Para el desarrollo de la aplicación no se va a utilizar u lenguaje orientado a objetos
Es necesario realizar consultas complejas	
Es necesario realizar validaciones de datos	
En el equipo de trabajo se mantiene una separación entre programadores de la aplicación y de la BD	

- **NoSQL**

No emplean el lenguaje SQL como principal lenguaje de consultas. Tampoco soportan operaciones **JOIN** como en los sistemas relacionales y **no suelen garantizar ACID**. Su principal característica es la escalabilidad horizontal, es decir, el rendimiento de la Base de Datos se mantiene a medida que se van añadiendo nodos al sistema. Algunos son: MongoDB y Cassandra.

Son apropiadas si
Es necesario manejar grandes cantidades de datos
Es necesario escalar (añadiendo nodos al sistema)

- **Según accesos simultáneos soportados**

- **Monousuario**

Son aquellos SGBD que no permiten que más de un usuario pueda estar conectado con la Base de Datos al mismo tiempo.

- **Multiusuario**

Permiten conexiones simultáneas con una misma Base de Datos.

- **Según el tamaño**

- **SGBD ligero**

Se trata de pequeños SGBD que dan soporte a algunas de las principales funciones de estas herramientas, ideales en entornos donde no es necesario un gran rendimiento o bien no se dispone de gran potencia de cálculo. El más conocido es **SQLite**.

- **SGBD Ofimática**

Algo más potentes que los SGBD ligeros, con algunas funciones más, muy utilizados en Ofimática para pequeñas Bases de Datos.

3. Tipos de Bases de Datos

Bases de Datos centralizadas: Son el tipo más común. Normalmente, se dedica un único equipo para el almacenamiento y gestión de los datos utilizando algún tipo de SGBD.

- **Ventajas**

- Fácil implementación.
- Diseño sencillo

- **Desventajas**
 - En caso de fallo, al existir una sola ubicación física de la BD, éte afectará a todos los datos almacenados en ella.

Bases de Datos distribuidas: Es un conjunto de bases de datos relacionadas lógicamente entre sí que se encuentran localizadas en diferentes ubicaciones físicas. Pueden tratarse de dos máquinas conectadas mediante una LAN o bien a través de Internet.

- **Ventajas**
 - Es más barato crear una red de máquina de bajo rendimiento que disponer de una sola máquina muy potente
 - Un fallo en cualquiera de los nodos no afecta al resto
 - Ya que los datos se encuentran distribuidos, éstos pueden ubicarse donde más interesa.
- **Desventajas**
 - Es complejo de poner en marcha y mantener
 - Al disponer de varios nodos, se dispone de varios sistemas diferentes que habrá que proteger
 - Existe poca experiencia ante fallos poco comunes
 - Es posible que se necesite más mano de obra especializada

4. Arquitectura de una aplicación

1. **Capa de presentación:** Es la capa que ve el usuario. Presenta la aplicación al usuario, le permite interactuar con el mismo y le muestra la información que éste utiliza.
2. **Capa de negocio:** En esta capa se encuentra toda la lógica de la aplicación y será donde se realicen todos los procesos de tratamiento de la información obtenida en la capa de datos y cuyos resultados se muestran al usuario mediante la capa de presentación.
3. **Capa de datos:** En esta capa residen los datos. En ella se encuentran los SGBD.

5. Las bases de datos en la producción de software

El concepto de **ciclo de vida** se puede definir como el conjunto de etapas por las que atraviesa el sistema desde el análisis, diseño, desarrollo y explotación.

La **ingeniería de software** es el conjunto de mecanismos para que la producción de software sea más rentable, minimizando sus procesos y aumentando la eficiencia.

Ciclo de vida en las bases de datos:

Principalmente se distinguen las siguientes fases:

- **Planificación:** qué se ha de realizar, con qué recursos se cuenta y qué técnicas se usan para la recolección de datos, su formato y documentación y cómo se llevará el diseño y la implementación
- **Definición:** se especifica el ámbito y los límites de la aplicación de base de daos, así como los sistemas con los que interactúa.
- **Recogida y análisis de los requisitos:** se recogen y analizan los requerimientos de los usuarios y de las áreas de aplicación a través de entrevistas, análisis de documentos e instrumentos de recogida de información, etc.
- **Diseño de la base de datos:** está dividido en tres fases importantes:
 - **Diseño conceptual:** se produce un esquema conceptual independiente de nivel físico del sistema.

- **Diseño lógico:** tras refinar el esquema conceptual se obtiene el esquema lógico, en el que se ha eliminado toda la construcción que no se puede explotar con el modelo de datos que se haya escogido.
- **Diseño físico:** atendiendo al SGBD elegido, el esquema lógico se traduce en un esquema físico. En este, se tiene en cuenta las estructuras de almacenamiento y los métodos de acceso.
- **Diseño de la aplicación:** diseño de los programas de aplicación que interactuará con la base de datos y, en muchas ocasiones es paralela al diseño de de esta.
- **Prototipado:** un prototipo es un modelo de trabajo de las aplicaciones del sistema y es útil para que los usuarios puedan usar el sistema e identificar qué aspectos son o no adecuados.
- **Implementación:** haciendo uso de los comandos del lenguaje de definición de datos se crea el esquema de la base de datos, el sistema de almacenamiento en ficheros y el esquema de usuarios.
- **Conversión y carga de datos:** cuando se actualizan sistemas y se debe reemplazar los datos antiguos por nuevos incluidos en diferentes estructuras.
- **Prueba y mantenimiento:** se prueba y valida los requisitos de los usuarios, se diseñan test con datos reales y se toman medidas sobre la fiabilidad y calidad del software desarrollado.