



Unidad 01: **Almacenamiento de la información**

Bases de datos **CFGS DAM**

Índice de contenido

1.INTRODUCCIÓN.....	3
2.SISTEMAS DE INFORMACIÓN.....	4
2.1.INTRODUCCIÓN: La empresa como sistema.....	4
2.2.SISTEMA DE INFORMACIÓN MECANIZADO.....	5
3.ARCHIVOS EN SISTEMAS TRADICIONALES.....	5
4.PROBLEMAS DE LOS SISTEMAS TRADICIONALES.....	6
4.1.REDUNDANCIA.....	6
4.2.DEPENDENCIA DE LOS PROGRAMAS RESPECTO DE LOS DATOS.....	8
4.3.ANOMALÍAS POR ACCESOS CONCURRENTES.....	8
4.4.PROBLEMAS DE SEGURIDAD.....	8
4.5.COSTOSA RECUPERACIÓN DE FICHEROS.....	9
5.CONCEPTO DE BASE DE DATOS.....	9
5.1.CLASIFICACIÓN DE LAS BASES DE DATOS SEGÚN SU MODELO DE ADMINISTRACIÓN DE DATOS.....	10
6.SISTEMAS GESTORES DE BASES DE DATOS.....	14
6.1.CONCEPTO DE SISTEMA GESTOR DE BASE DE DATOS.....	14
6.2.FUNCIONES DE UN SGBD.....	15
6.3.EL LENGUAJE SQL.....	16
6.4.TIPOS DE SGBD.....	18
7.BASES DE DATOS CENTRALIZADAS Y DISTRIBUIDAS.....	19
7.1.DISTRIBUCIÓN DE LOS DATOS.....	20

1. INTRODUCCIÓN

Todo el mundo sabe, que hoy en día, en cualquier organización social o empresarial el tratamiento de datos resulta fundamental.

“El otro día fui a la piscina municipal de mi barrio. Hacía dos años que no usaba sus instalaciones pues estuve trabajando en otra ciudad. Me preguntaron por mi DNI. Cuando la chica que me atendió tecleó mi DNI en el ordenador automáticamente aparecieron mis datos, nombre, dirección, teléfono de contacto, etc.”

En las sociedades desarrolladas actuales, cada vez hay una mayor demanda de datos disparada por el uso y acceso a Internet. Por otra parte, resulta evidente que para el uso y gestión de datos, empleamos dispositivos informáticos y electrónicos como Smartphone, tabletas u ordenadores, accediendo con ellos a grandes cantidades de información.

El propio nombre Informática hace referencia al hecho de ser una ciencia que trabaja con información. Desde los albores de la creación de los ordenadores, la información se ha considerado como uno de los pilares de la computación digital. Por ello la gestión de bases de datos es una de las aplicaciones más antiguas de la informática.

Antes de la informática, el ser humano gestionaba datos controlando almacenes de alimentos, gestionando controles de inventario y otros muchos sistemas de datos con rudimentos como libros de contabilidad y otros documentos que almacenaban en archivos, cajones, carpetas y fichas en las que se almacenaban los datos. Antes de la aparición del ordenador, el tiempo requerido para manipular estos datos era enorme, sin embargo el proceso de aprendizaje era relativamente sencillo ya que se usaban elementos que el usuario reconocía perfectamente. Por esa razón, la informática adaptó sus herramientas para que los elementos que el usuario maneja en el ordenador se parecieran a los que utilizaba manualmente. Así en informática se sigue hablado de ficheros, formularios, carpetas o directorios.



En los primeros años de la informática la información se almacenaba en conjuntos de ficheros (secuenciales, directos, indexados, etc.). Los sistemas operativos, a través de sus rutinas básicas de inserción, modificación, borrado y recuperación de registros, manejaban la información contenida en los registros de dichos ficheros. Pero estos sistemas de información tradicionales, presentaban varios problemas, de los cuales hablaremos en breve.

2. SISTEMAS DE INFORMACIÓN

2.1. **INTRODUCCIÓN: La empresa como sistema.**

Según la RAE, la definición de **sistema** es “Conjunto de cosas que ordenadamente relacionadas entre sí contribuyen a un determinado objeto”.

El cliente fundamental del profesional de la informática es la empresa. La empresa se puede entender como un sistema formado por diversos objetos: el capital, los recursos humanos, los inmuebles, los servicios que presta, etc.

El sistema completo que forma la empresa, por otra parte, se suele dividir en los siguientes subsistemas:

- **Subsistema productivo.** También llamado subsistema real o físico. Representa la parte de la empresa encargada de gestionar la producción de la misma.
- **Subsistema financiero.** Encargado de la gestión de los bienes económicos de la empresa.
- **Subsistema directivo.** Encargado de la gestión organizativa de la empresa.

Cada subsistema se asocia a un departamento concreto de la empresa.

Los sistemas que, a su vez, aglutinan los elementos que intervienen para gestionar la información que manejan los subsistemas empresariales es lo que se llaman **Sistemas de Información (SI)** o IS (Information Server).

Realmente un SI sólo incluye la información que nos interesa de la empresa y los elementos necesarios para gestionar esa información.

Un sistema de información genérico está formado por los siguientes elementos:

- **Recursos físicos.** Carpetas, documentos, equipamiento, discos,...
- **Recursos humanos.** Personal que maneja la información
- **Protocolo.** Normas que debe cumplir la información para que sea manejada (formato de la información, modelo para los documentos,...)

Las empresas necesitan implantar sistemas de información ya que la competencia obliga a gestionar de la forma más eficiente sus datos para una mayor calidad en la organización de las actividades de los subsistemas empresariales.

A los Sistemas utilizados tradicionalmente para organizar y gestionar

estas informaciones de forma manual se les conoce como SISTEMAS DE INFORMACIÓN NO MECANIZADOS, ya que su creación y utilización son anteriores a la aparición de la tecnología computacional.

2.2. SISTEMA DE INFORMACIÓN MECANIZADO.

Los avances tecnológicos dieron lugar a la aparición de los Equipos Informáticos, y estos permitían almacenar y procesar la información de una forma mucho más rápida y eficiente. Así aparecieron los SISTEMAS DE INFORMACIÓN MECANIZADOS o SISTEMAS INFORMÁTICOS.

De esta forma podemos definir a los Sistemas de Información actuales o SISTEMAS INFORMÁTICOS como: “Conjunto de elementos tecnológicos relacionados que cooperan para la consecución de un determinado fin empresarial, profesional o personal”.

Pero para que un Sistema Informático sea útil, es decir, pueda recibir y dar información, no puede estar aislado del mundo exterior, sino que debe interaccionar, es decir, comunicarse con él, de tal forma que fruto de esa interacción, se produce el funcionamiento esperado del sistema, así debemos hablar del ENTORNO DE UN SISTEMA: “Es todo aquello que rodea y afecta al sistema, es decir, lo que interacciona con él y le influye”.

Ahora los elementos que conforman un sistema de información mecanizado son los siguientes:

- **Datos.** Se trata de la información relevante que almacena y gestiona el sistema de información.
- **Hardware.** Equipamiento físico que se utiliza para gestionar los datos. cada uno de los dispositivos electrónicos que permiten el funcionamiento del sistema de información.
- **Software.** Aplicaciones informáticas que se encargan de la gestión de la base de datos y de Recursos humanos.
- **Administrador.** Personal que maneja el sistema de información.
- **Usuarios.**

En un sistema de información mecanizado se pueden distinguir dos perspectivas diferentes pero evidentemente interrelacionadas: la perspectiva estática, relacionada con la información de la organización propiamente dicha, y la perspectiva dinámica, relacionada con las funciones que la organización realiza para el tratamiento de dicha información.

3. ARCHIVOS EN SISTEMAS TRADICIONALES

A continuación vemos dos ejemplos de tipos de archivos que se

empleaban para la organización de la información en los sistemas tradicionales, antes de la aparición de los sistemas gestores de bases de datos allá por los finales de los años 60 y principios de los 70 y de los que hablaremos más adelante.

- **ARCHIVO SECUENCIAL.**

El Archivo Secuencial es la forma más simple de almacenar y recuperar registros de un archivo. En un archivo secuencial, se almacenan los registros uno tras otro. El primer registro almacenado se coloca al principio del archivo. El segundo se almacena inmediatamente después, el tercero después del segundo y así sucesivamente. Este orden nunca cambia en la organización secuencial.

Para la lectura de un registro en un archivo secuencial, el sistema siempre comienza desde el principio del archivo, lee un registro a la vez hasta llegar al registro solicitado. Si el registro que se desea consultar es el número 100, el sistema leerá desde el inicio y leerá hacia adelante hasta llegar al deseado.

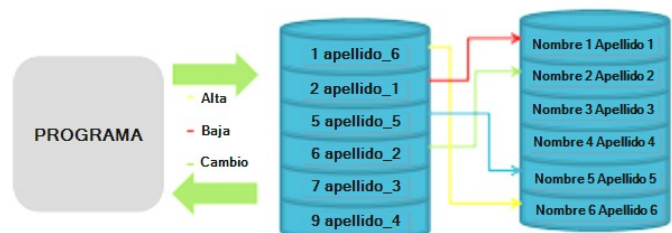
Ejemplo de Archivo Secuencial.



Las operaciones a realizar sobre algún registro de un fichero secuencial resultan bastante lentas.

- **ARCHIVO SECUENCIAL-INDEXADO.**

La estructura más simple de este tipo de archivos tiene como índice un archivo secuencial simple, cada registro del archivo índice tiene dos campos, un campo clave igual al del archivo principal y un puntero al archivo principal. Para encontrar un campo específico se busca en el índice hasta encontrar el valor mayor de la clave que es igual o precede al valor deseado, la búsqueda continua en el archivo principal a partir de la posición que indique el puntero.

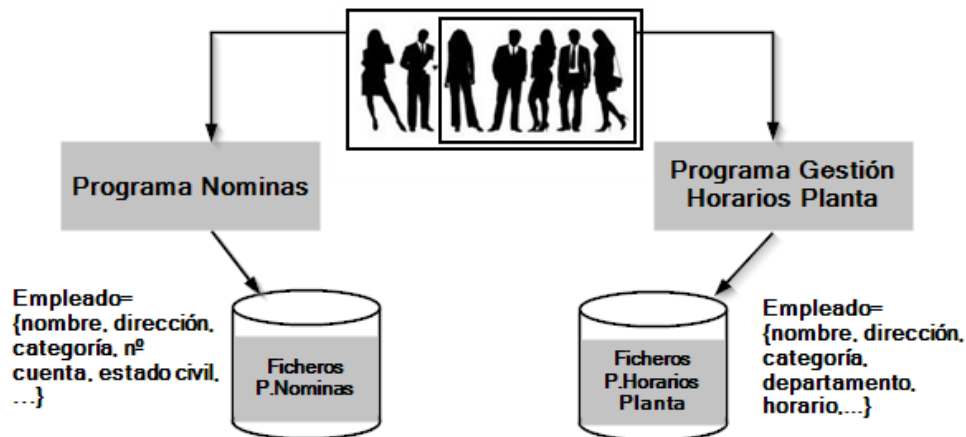


4. PROBLEMAS DE LOS SISTEMAS TRADICIONALES

4.1. REDUNDANCIA.

Una de las características de los sistemas tradicionales es que para cada sección de una organización se desarrollaban aplicaciones independientes con todos los ficheros de datos propios que sean necesarios.

Supongamos una empresa en la que por un lado está la sección de personal en la que se gestiona a los empleados de toda la empresa para el control de nóminas y demás cuestiones sobre los empleados. Por otra parte en la sección de producción de la empresa también se quiere gestionar al personal, pero en este caso sólo a los empleados que trabajan en la planta de producción y en este caso lo que se pretende es gestionar toda la cuestión sobre turnos, horarios, vacaciones, bajas laborales, etc.



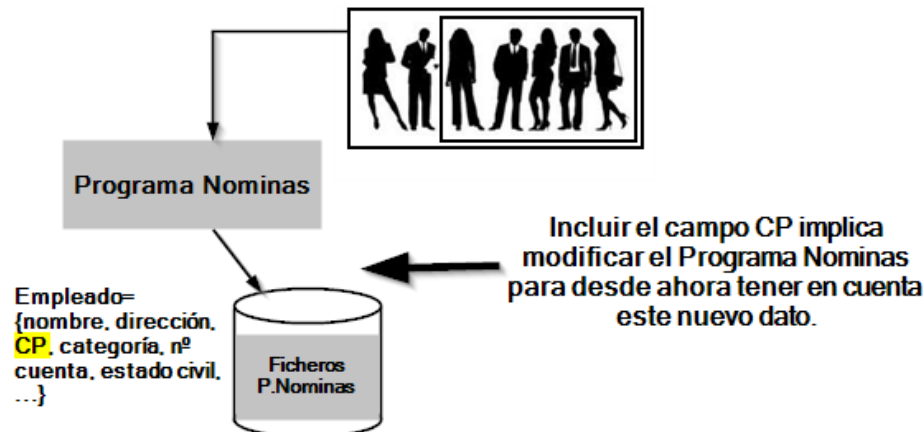
Como puedes ver, aunque cada sección gestiona información propia, siempre existen datos comunes a varias aplicaciones que, al estar almacenados en ficheros diferentes, suponen una redundancia de información.

La redundancia de datos provoca los siguientes problemas:

- **Inconsistencia de la información:** Si se producen muchas actualizaciones de la información, es difícil mantener todos los ficheros con información común en un estado coherente; de esta manera es posible tener información contradictoria sobre un mismo objeto.
- Obviamente, se produce una **mayor ocupación del espacio de almacenamiento** al repetirse datos.
- Actualizar un dato como por ejemplo la dirección de un empleado se hace más costoso.

4.2. **DEPENDENCIA DE LOS PROGRAMAS RESPECTO DE LOS DATOS.**

En los sistemas tradicionales, la descripción de los ficheros utilizados por una aplicación (formato de los registros, organización y modo de acceso, etc.) se adaptaba a la propia aplicación, de manera que cualquier modificación en un fichero (Por ejemplo por la necesidad de incluir un nuevo campo en un registro como un código postal) implica la modificación del programa que accede a dicho fichero.



Esto da lugar a un alto coste de mantenimiento del software y poca flexibilidad del sistema ante variaciones futuras en los requerimientos de los usuarios.

4.3. **ANOMALÍAS POR ACCESOS CONCURRENTES.**

Para ofrecer mayores prestaciones, muchos sistemas permiten que distintos usuarios accedan a los datos simultáneamente. Este acceso concurrente puede provocar datos incorrectos. Para evitar esto, el sistema debe proporcionar alguna forma de supervisión de este tipo de accesos.

Veamos un típico ejemplo: Imagina una cuenta corriente con 1000 euros. Si dos personas retiran fondos (por ejemplo, 100 y 200 euros respectivamente) de la cuenta casi al mismo tiempo, el resultado de la ejecuciones concurrentes puede dejar la cuenta en un estado incorrecto. En concreto, el resultado de esta operación podría ser 800 o 900 euros, en lugar de 700 euros.

4.4. **PROBLEMAS DE SEGURIDAD.**

En un SI, no todos los usuarios pueden tener acceso a todos los datos ni emplearlos de la misma forma. Para esta cuestión los sistemas tradicionales no contemplaban mecanismos para el control de autorizaciones o, en caso de haberlos, eran muy inflexibles, estando dirigidos a cada aplicación.

4.5. ***COSTOSA RECUPERACIÓN DE FICHEROS***

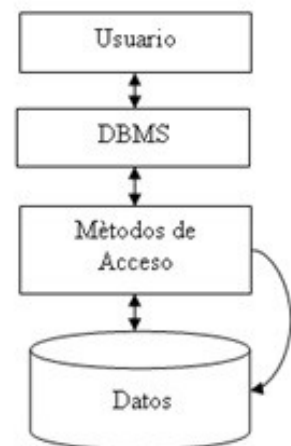
La recuperación de la información (de los ficheros) ante cualquier fallo como roturas de discos, fallos eléctricos, etc. resultaba muy costosa si las copias de seguridad eran antiguas.

5. **CONCEPTO DE BASE DE DATOS.**

Como ya se ha insinuado, las técnicas de bases de datos surgieron para tratar de resolver los problemas anteriores.

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido; una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, ofreciéndose un amplio rango de soluciones al problema del almacenamiento de datos.

Existen programas denominados **sistemas gestores de bases de datos**, abreviado **SGBD** (del inglés database management system o DBMS), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Estos sistemas proporcionan métodos para mantener la integridad de los datos, para administrar el acceso de usuarios a los datos y para recuperar la información si el sistema se corrompe. Permiten presentar la información de la base de datos en variados formatos. La mayoría incluyen un generador de informes. También pueden incluir un módulo gráfico que permita presentar la información con gráficos y tablas. De estos SGBD hablaremos detalladamente en los siguientes puntos.



Las aplicaciones más usuales de bases de datos son para la gestión de empresas e instituciones públicas aunque también son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

Aunque las bases de datos pueden contener muchos tipos de datos, algunos de ellos se encuentran protegidos por las leyes de varios países. Por ejemplo en España, los datos personales se encuentran protegidos por la **Ley Orgánica de Protección de Datos de Carácter Personal (LOPD)**, en México por la Ley Federal de Transparencia y Acceso a la Información Pública Gubernamental y en Argentina la Ley de Protección de Datos Personales.

5.1. CLASIFICACIÓN DE LAS BASES DE DATOS SEGÚN SU MODELO DE ADMINISTRACIÓN DE DATOS.

Para la construcción de un sistema de información se busca crear, por medio de una abstracción del mundo real de un conjunto estructurado de datos y un conjunto de operaciones definidas sobre ellos que permitan satisfacer las necesidades de información de una organización de la forma más eficiente.

Para realizar este proceso de abstracción que permite pasar del mundo real al mundo de los datos, es necesario apoyarse en un conjunto de conceptos y reglas, es decir en un modelo de datos.

Un **modelo de datos** es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas, son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.

Un modelo de datos es una herramienta intelectual (conjunto de conceptos y reglas) que permiten representar los aspectos estáticos y dinámicos de la parcela del mundo real que se desea estudiar:

- **Aspectos estáticos:** objetos de información (entidades en el modelo relacional), propiedades de los objetos (atributos en el modelo relacional), relaciones entre objetos y restricciones sobre los objetos o sus relaciones. A la descripción de las propiedades estáticas se la denomina **esquema**.
- **Aspectos dinámicos:** acciones sobre los objetos o sus relaciones (inserción de nueva información, borrado, modificación de un valor), relaciones entre operaciones (transacciones) y restricciones sobre la evolución de los objetos y sus relaciones.

A lo largo de los años los modelos de datos han ido sufriendo una larga evolución distinguiéndose tres generaciones:

- Modelos de datos primitivos.
- Modelos de datos clásicos.
- Modelos de datos semánticos.

MODELOS DE DATOS PRIMITIVOS

Los modelos primitivos coinciden con los modelos de ficheros. En estos

modelos, los objetos se representan como registros organizados en ficheros y las relaciones entre objetos a través de referencias explícitas (campo del registro cuyo valor referencia a otro objeto). Los lenguajes de manipulación son totalmente dependientes de la organización física de estos ficheros y las operaciones básicas para manipular los objetos son las operaciones primitivas sobre ficheros: lectura y escritura.

MODELOS DE DATOS CLÁSICOS

Por su parte, cuando hablamos de modelos clásicos, son aquellos en los que se empezaron a basar los sistemas de gestión de bases de datos (SGBD), pudiendo clasificarlos en tres grandes familias:

Modelos de datos clásicos:

- o Sistemas jerárquicos (Modelo jerárquico)
- o Sistemas en red (Modelo en red)
- o Sistemas relacionales (Modelo relacional)

Además de los anteriores, también podemos hablar de otros modelos más actuales:

- o Sistema orientado a objetos (Modelo orientado a objetos)
- o Sistemas de bases de datos multidimensionales.
- o Sistemas de bases de datos documentales.
- o Sistemas de bases de datos deductivas.

Los modelos jerárquico y en red surgieron como extensiones de los modelos de ficheros. En estos modelos, los objetos se siguen representando como registros organizados en ficheros, pero sin embargo proporcionan estructuras de datos más complejas (árbol, set) que permiten expresar directamente relaciones entre objetos (registros). En estos modelos, los lenguajes de manipulación constan de operadores que permiten manejar estas estructuras de datos y que son independientes de la representación física de dichas estructuras.

Sobre bases jerárquicas:

https://es.wikipedia.org/wiki/Base_de_datos_jer%C3%A1rquica

Sobre bases en red:

https://es.wikipedia.org/wiki/Base_de_datos_de_red

Por su parte, el **modelo relacional**, para el modelado y la gestión de bases de datos, es un modelo de datos basado en la lógica de predicados y en la teoría de conjuntos. Tras ser postuladas sus bases

en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos.

Su idea fundamental es el uso de **relaciones**. Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados **tuplas**. Una manera más fácil de imaginar este modelado, es pensar en cada relación como si fuese una tabla que está compuesta por registros (cada fila de la tabla sería un registro o "tupla") y columnas (también llamadas "campos").

Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente.

Ver: https://es.wikipedia.org/wiki/Modelo_relacional

Entre los modelos de datos más recientes, podemos hablar de las **bases de datos orientadas a objetos**. Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento).

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

- Encapsulación: Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- Herencia: Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
- Polimorfismo: Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros). La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado.

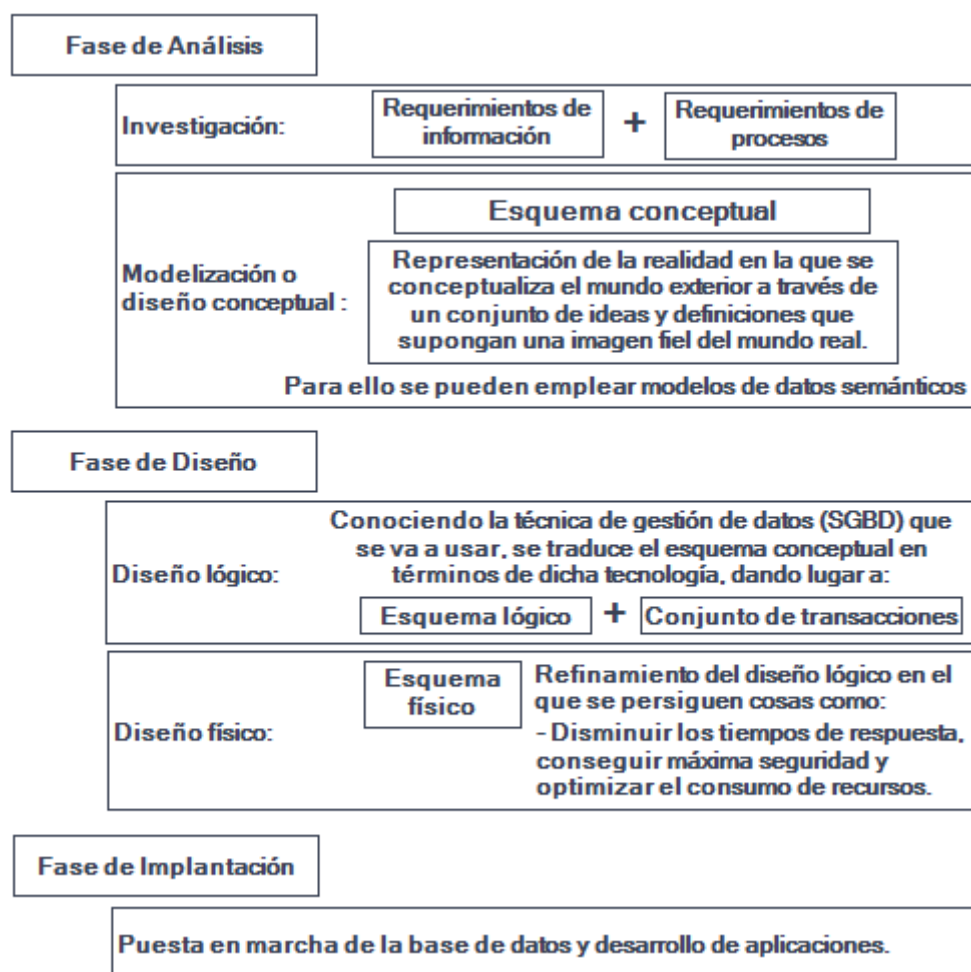
Para finalizar este apartado definiremos brevemente otros modelos usados para soluciones específicas y normalmente basados en el modelo relacional.

- **Bases de datos multidimensionales:** Se utilizan principalmente para crear aplicaciones **OLAP** (On-Line Analytical Processing). Es una solución utilizada en el campo de la llamada Inteligencia de negocios (o Business Intelligence) cuyo objetivo es agilizar la consulta de grandes cantidades de datos. Para ello utiliza estructuras multidimensionales (o Cubos OLAP) que contienen datos resumidos de grandes Bases de datos o Sistemas Transaccionales (OLTP).
- **Bases de datos documentales:** Sirven para almacenar grandes volúmenes de información de antecedentes históricos. Permiten la indexación a texto completo y en líneas generales realizar búsquedas muy potentes. Tesauro es un sistema de índices optimizado para este tipo de bases de datos.
- **Bases de datos deductivas:** También llamadas bases de datos lógicas, a raíz de que se basa en lógica matemática. Este tipo de base de datos surge debido a las limitaciones de la Base de Datos Relacional de responder a consultas recursivas y de deducir relaciones indirectas de los datos almacenados en la base de datos.

MODELOS DE DATOS SEMÁNTICOS: Modelo Entidad-Relación.

Los modelos de datos semánticos surgen con la intención de aumentar la capacidad expresiva de los modelos clásicos a la hora de modelizar la realidad en el proceso de definición de una solución de base de datos.

Hablamos de herramientas para la modelización (diseño conceptual) de bases de datos y no de modelos, como los clásicos vistos anteriormente y es que en el proceso de desarrollo de un SI, en el que se pretende obtener, por medio de una abstracción del mundo real, un conjunto de datos y un conjunto de operaciones sobre ellos que permitan satisfacer las necesidades de la organización, se dan las siguientes fases.



Cómo puedes observar, es en la fase de creación del esquema conceptual, en la que se pueden emplear modelos de datos semánticos para la definición del mismo.

Así pues, una herramienta para el modelado de datos propuesta por P. Chen en 1976, es el conocido como **modelo entidad-relación**, cuyo estudio abordaremos en siguientes unidades.

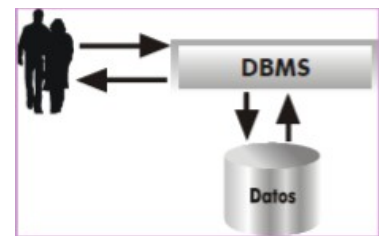
6. SISTEMAS GESTORES DE BASES DE DATOS.

6.1. CONCEPTO DE SISTEMA GESTOR DE BASE DE DATOS

Un sistema gestor de bases de datos o **SGBD** (aunque se suele utilizar más a menudo las siglas **DBMS** procedentes del inglés, Data Base Management System) es el software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos.

En estos Sistemas se proporciona un conjunto coordinado de programas, procedimientos y lenguajes que permiten a los distintos usuarios realizar sus tareas habituales con los datos, garantizando además la seguridad de los mismos.

El éxito del SGBD reside en mantener la seguridad e integridad de los datos. Lógicamente tiene que proporcionar herramientas a los distintos usuarios. Entre las herramientas que proporciona están:



- Herramientas para la creación y especificación de los datos. Así como la estructura de la base de datos.
- Herramientas para administrar y crear la estructura física requerida en las unidades de almacenamiento.
- Herramientas para la manipulación de los datos de las bases de datos, para añadir, modificar, suprimir o consultar datos.
- Herramientas de recuperación en caso de desastre.
- Herramientas para la creación de copias de seguridad.
- Herramientas para la creación de aplicaciones que utilicen esquemas externos de los datos.
- Herramientas de instalación de la base de datos.
- Herramientas para la exportación e importación de datos.

Un ejemplo de software Gestor de Base de Datos es Oracle 11g, que incorpora un conjunto de herramientas software que son capaces de estructurar en múltiples discos duros los ficheros de una base de datos, permitiendo el acceso a sus datos tanto a partir de herramientas gráficas como a partir de potentes lenguajes de programación (PL-SQL, php, c++. . .). Otros ejemplos son PostgreSQL, sistema de gestión de base de datos relacional orientada a objetos y libre bajo licencia BSD. MySQL o Microsoft SQL Server, entre otros.

6.2. ***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 los datos en tablas, registros y campos, las relaciones entre los datos, usuarios, permisos, etc. Este diccionario de datos debe ser también accesible de la misma forma sencilla 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 completas herramientas, 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, y permiten analizar hipotéticos mal funcionamientos.
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. De esta manera, se puede replicar y distribuir bases de datos. Además, todos los SGBD incorporan herramientas estándar de conectividad. El protocolo

ODBC4 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.

6.3. EL LENGUAJE SQL

La principal herramienta de un gestor de base de datos es **la interfaz de programación con el usuario**. Esta interfaz consiste en un lenguaje muy sencillo mediante el cual el usuario realiza preguntas al sistema gestor, contestando este a las demandas del usuario. Este lenguaje comúnmente se denomina **SQL, Structured Query Language**. Todas las bases de datos que soporten SQL deben tener la misma sintaxis a la hora de aplicar el lenguaje. Se divide en 4 sub-lenguajes:

Lenguaje DML: o lenguaje de manipulación de datos (Data Manipulation Language). Este lenguaje permite con 4 sentencias sencillas; 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 (desde tablas hasta usuarios). Sus cláusulas son del tipo DROP (Eliminar objetos) y CREATE (Crear objetos). En capítulos 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 paso de la ejecución, sucede algo inesperado, se pueden deshacer todos los pasos dados (ROLLBACK).

Sobre SQL <https://es.wikipedia.org/wiki/SQL>

Ejemplo de sentencia SQL:

```
SQL> SELECT APELLIDO FROM EMPLE WHERE OFICIO = 'ANALISTA';
```

6.4. TIPOS DE SGBD

Se pueden clasificar los SGBD de muchas formas, por ejemplo, clasificándolos según traten bases de datos relacionales, bases de datos orientadas a objetos, etc. Puesto que en la actualidad, la mayoría de los SGBD integran múltiples filosofías y tipos de funcionamiento, vamos a clasificar los gestores de bases de datos según su capacidad y potencia. Así tenemos:

Los Gestores de Bases de Datos ofimáticas, que son aquellos que manipulan bases de datos pequeñas (ofimáticas) orientadas a almacenar datos domésticos o de pequeñas empresas. Incluso estos gestores permiten construir pequeñas aplicaciones para ayudar a un usuario inexperto a manipular los datos de una base de datos de forma sencilla e intuitiva.

Un ejemplo de un SGBD ofimática es Microsoft Access, que posee tanto una interfaz de usuario muy sencilla como un potente lenguaje de programación (VBA=Visual Basic for Applications) para ofrecer a usuarios avanzados otras posibilidades de gestión mucho 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 de grandes dimensiones (generalmente un Servidor Unix, o Windows 20XX Server con altas prestaciones). 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. Un ejemplo típico de servidor de base de datos Corporativas es el antes comentado Oracle, actualmente, junto con DB2, es el servidor de base de datos más potente del mercado (también el más caro). Precisamente, ese coste tan alto es el que ha desencadenado que se haya recurrido a una solución intermedia entre gestores de base de datos ofimáticas y corporativas.

Entre estas soluciones intermedias se encuentra **MySQL**, un gestor de base 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. Aunque implementa SQL, no tiene un lenguaje de programación propio como SQL Server u Oracle (aunque está en desarrollo), pero a cambio 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 para acceso y publicación de los datos.

Sobre ANSI-SPARC [https://es.wikipedia.org/wiki/Arquitectura ANSI-SPARC](https://es.wikipedia.org/wiki/Arquitectura_ANSI-SPARC)

Sobre MySQL <https://es.wikipedia.org/wiki/MySQL>

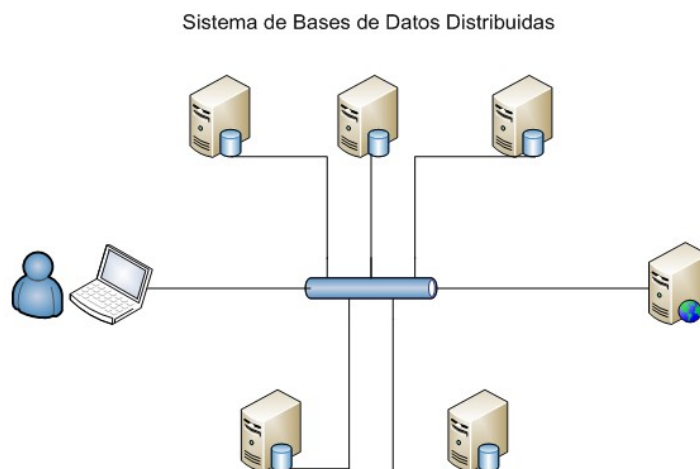
7. BASES DE DATOS CENTRALIZADAS Y DISTRIBUIDAS.

Otro aspecto que debemos conocer, atendiendo al título de este apartado, es la diferenciación entre las bases de datos centralizadas y las bases de datos distribuidas.

Una **base de datos centralizada** es aquella que almacena los datos que maneja íntegramente en un mismo lugar físico. En definitiva, guarda los datos en una única unidad de proceso que puede ser accedida desde terminales.

Una **base de datos distribuida** (BDD) es un conjunto de múltiples bases de datos lógicamente relacionadas las cuales se encuentran repartidas en diferentes espacios lógicos y geográficos (Por ejemplo, un servidor corriendo dos o más máquinas virtuales) e interconectados por una red de comunicaciones.

Un sistema de Bases de Datos Distribuida (SBDD) es un sistema en el cual múltiples sitios de bases de datos están ligados por un sistema de comunicaciones de tal forma que, un usuario en cualquier sitio puede acceder los datos en cualquier parte de la red exactamente como si estos fueran accedidos de forma local.



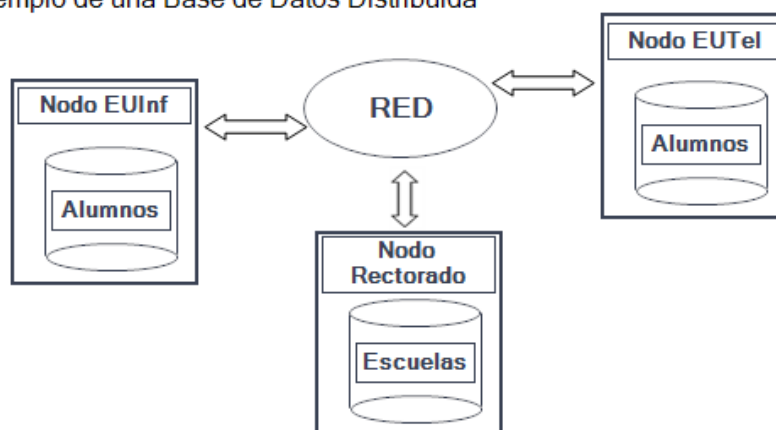
Un sistema distribuido de bases de datos se almacena en varias computadoras. Los principales factores que distinguen un SBDD de un sistema centralizado son los siguientes:

- Hay múltiples computadores, llamados sitios o nodos.
- Estos nodos deben de estar comunicados por medio de algún tipo de red para transmitir datos y órdenes entre los sitios.

Hay varios factores que han hecho que las bases de datos evolucionen a bases de datos distribuidas. En el mundo de los negocios se ha dado una globalización y a la vez las operaciones de las empresas son cada

vez más descentralizadas geográficamente. También el poder de las computadoras personales aumentó y el costo de las grandes computadoras centrales (Mainframes) ya no tenía sentido. Además la necesidad de compartir datos ha hecho que crezca el mercado de las bases de datos distribuidas.

Ejemplo de una Base de Datos Distribuida



- NODOS DE LAS ESCUELAS

DNI	Escuela	Nombre	Nota ingreso	Beca
-----	---------	--------	--------------	------

- NODO DEL RECTORADO

Escuela	Situación	Nº Alumnos
---------	-----------	------------

Dar de alta un nuevo alumno en una escuela implica una transacción local en el nodo de la escuela y una transacción global en el nodo del rectorado.

Sobre Bases de datos distribuidas:

https://es.wikipedia.org/wiki/Base_de_datos_distribuida

7.1. DISTRIBUCIÓN DE LOS DATOS

Una de las decisiones más importantes que el diseñador de bases de datos distribuidas debe tomar es el posicionamiento de los datos en el sistema y el esquema bajo el cual lo desea hacer. Para esto existen varias alternativas: replicada, fragmentada, e híbrida.

Distribución replicada: El esquema de BDD de replicación consiste en que cada nodo debe tener su copia completa de la base de datos. Es fácil ver que este esquema tiene un alto costo en el almacenamiento de la información. Debido a que la actualización de

los datos debe ser realizada en todas las copias, también tiene un alto costo de escritura, pero todo esto vale la pena si tenemos un sistema en el que se va a escribir pocas veces y leer muchas, y dónde la disponibilidad y fiabilidad de los datos sea de máxima importancia. El sistema seguirá funcionando aún en caso de caída de uno de los nodos y otra ventaja es el aumento del paralelismo: Varios nodos pueden realizar consultas en paralelo sobre la misma tabla. Cuantas más réplicas existan de la tabla, mayor será la posibilidad de que el dato buscado se encuentre en el nodo desde el que se realiza la consulta, minimizando con ello el tráfico de datos entre nodos.

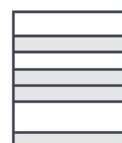
Fragmentación: Este modelo consiste en que solo hay una copia de cada elemento, pero la información está distribuida a través de los nodos. En cada nodo se aloja uno o más fragmentos disjuntos de la base de datos. Como los fragmentos no se replican esto disminuye el costo de almacenamiento, pero también sacrifica la disponibilidad y fiabilidad de los datos. Algo que se debe tomar en cuenta cuando se desea implementar este modelo es la granularidad de la fragmentación. Existen tres tipos de fragmentación la horizontal, la vertical y la mixta:

- **Fragmentación Horizontal:** Una tabla T se divide en subconjuntos, T_1, T_2, \dots, T_n . Los fragmentos se definen a través de una operación de selección y su reconstrucción se realizará con una operación de unión de los fragmentos componentes. Cada fragmento se sitúa en un nodo.

- **Fragmentación Vertical:** Una tabla T se divide en subconjuntos, T_1, T_2, \dots, T_n . Los fragmentos se definen a través de una operación de proyección. Cada fragmento debe incluir la clave primaria de la tabla.

Operadores Algebra relacional

Selección



Proyección



Ejemplo de fragmentación horizontal:

Tabla inicial de alumnos

DNI	Escuela	Nombre	Nota ingreso	Beca
38433678	EUInf	Ana Bayo	8.0	No
34755899	EUInf	Juan Peris	5.6	Sí
39278833	EUTel	Oscar Vidal	6.3	No
57099350	EUInf	Maite Pla	7.1	No
43234344	EUTel	Antonio León	5.8	Sí
42334544	EUInf	Eva García	7.5	Sí
65650291	EUTel	Maite Sanz	5.6	Sí

Tabla de alumnos fragmentada:

Fragmento de la EUInf

DNI	Escuela	Nombre	Nota ingreso	Beca
38433678	EUInf	Ana Bayo	8.0	No
34755899	EUInf	Juan Peris	5.6	Sí
57099350	EUInf	Maite Pla	7.1	No
42334544	EUInf	Eva García	7.5	Sí

Fragmento de la EUTel

DNI	Escuela	Nombre	Nota ingreso	Beca
39278833	EUTel	Oscar Vidal	6.3	No
43234344	EUTel	Antonio León	5.8	Sí
65650291	EUTel	Maite Sanz	5.6	Sí

Ejemplo de fragmentación Vertical:

Datos Rectorado:

Escuela	Situación	Alumnos
EUInf	Campus Este	2600
EUTop	Campus Este	1960
EUTel	Campus Sur	2100
ETFis	Campus Norte	2150

Departamento Infraestructura

Escuela	Situación
EUInf	Campus Este
EUTop	Campus Este
EUTel	Campus Sur
ETFis	Campus Norte

Departamento Ord. Académica

Escuela	Alumnos
EUInf	2600
EUTop	1960
EUTel	2100
ETFis	2150

- **Fragmentación Mixta:** Como el mismo nombre indica es una combinación de los dos anteriores modos de fragmentación de datos.

Por ejemplo, a partir de una tabla fragmentada horizontalmente, se fragmenta verticalmente.

Tabla inicial de alumnos

DNI	Escuela	Nombre	Nota ingreso	Beca
38433678	EUInf	Ana Bayo	8.0	No
34755899	EUInf	Juan Peris	5.6	Sí
39278833	EUTel	Oscar Vidal	6.3	No
57099350	EUInf	Maite Pla	7.1	No
43234344	EUTel	Antonio León	5.8	Sí
42334544	EUInf	Eva García	7.5	Sí
65650291	EUTel	Maite Sanz	5.6	Sí

Tabla de alumnos fragmentada:

Fragmento de la EUInf

DNI	Escuela	Nombre	Nota ingreso	Beca
38433678	EUInf	Ana Bayo	8.0	No
34755899	EUInf	Juan Peris	5.6	Sí
57099350	EUInf	Maite Pla	7.1	No
42334544	EUInf	Eva García	7.5	Sí

DNI	Escuela	Nombre	Nota ingreso
38433678	EUInf	Ana Bayo	8.0
34755899	EUInf	Juan Peris	5.6
57099350	EUInf	Maite Pla	7.1
42334544	EUInf	Eva García	7.5