

Bases de Datos

Tema 01. Introducción a la Gestión de los datos

Tabla de contenidos

INTRODUCCIÓN	3
SISTEMA DE INFORMACIÓN	5
Componentes de un sistema de información electrónico.....	6
Archivos o Ficheros	7
Ficheros secuenciales	8
Ficheros de acceso directo o aleatorio.....	9
Ficheros secuenciales encadenados.....	11
Ficheros secuenciales indexados.....	12
Ficheros indexado-encadenados.....	15
BASES DE DATOS	18
EVOLUCIÓN DE LAS BD.....	6
Niveles de abstracción: independencia física y lógica	10
Ventajas de las BD	13
MODELO DE DATOS.....	14
CLASIFICACION DE LOS MODELOS DE DATOS	20
SISTEMAS GESTORES DE BASES DE DATOS	23
Objetivos de los SGBD:	24
Funciones de los SGBD:	24
Componentes de un SGBD:	26

INTRODUCCIÓN

En el mundo actual existe una cada vez mayor demanda de datos. Esta demanda siempre ha sido patente en empresas y sociedades, pero en estos años la demanda todavía se ha disparado más debido al acceso multitudinario a Internet.

En informática se conoce como **dato** a cualquier elemento informativo que tenga relevancia para el sistema.

Inicialmente los datos que se necesitaba almacenar y gestionar eran pocos, pero poco a poco han ido creciendo. En la actualidad las numerosas aplicaciones de Internet han producido enormes sistemas de información que incluso para poder gestionarlos se requieren decenas de máquinas haciendo la información accesible desde cualquier parte del planeta y de forma rápida.

Anteriormente el ser humano contaba con su memoria y con la escritura para poder almacenar los datos. Para poder gestionar esos datos, el ser humano ideó nuevas herramientas archivos, cajones, carpetas y fichas en las que se almacenaban los datos. La informática adaptó sus herramientas para que los elementos que el usuario maneja en el ordenador se parezcan a los que utilizaba manualmente. Así en informática se sigue hablado de ficheros, formularios, carpetas, directorios,....



SISTEMA DE INFORMACIÓN

Los sistemas que aglutinan los elementos que intervienen para gestionar la información que manejan los subsistemas empresariales es lo que se conoce como **Sistemas de Información**. Se suele utilizar las siglas SI o IS (de Information Server) para referirse a ello).

Realmente un sistema de información 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 estos sistemas de información debido a la competencia que las obliga a gestionar de la forma más eficiente sus datos para una mayor calidad en la organización de las actividades de los subsistemas empresariales.

Componentes de un sistema de información electrónico

En el caso de una **gestión electrónica de la información** (lo que actualmente se considera un **sistema de información electrónico**), los componentes son:

- **Datos.** Se trata de la información relevante que almacena y gestiona el sistema de información. Ejemplos de datos son: **Sánchez**, **12764569F**, **Calle Mayo 5**, **Azul**...
- **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 las herramientas que facilitan su uso.
- **Recursos humanos.** Personal que maneja el sistema de información.

Archivos o Ficheros

Los ficheros o archivos son la herramienta fundamental de trabajo en un ordenador todavía a día de hoy. Las computadoras siguen almacenando la información en ficheros, eso sí de estructura cada vez más compleja.

Los datos deben de ser almacenados en componentes de almacenamiento permanente, lo que se conoce como **memoria secundaria** (discos duros u otras unidades de disco). En esas memorias, los datos se estructuran en archivos (también llamados ficheros).

Un fichero es una secuencia de números binarios que organiza información relacionada a un mismo aspecto.

Cuando los ficheros almacenan datos, se dice que constan de **registros**. Cada registro contiene datos relativos a un mismo elemento u objeto. Por ejemplo en un fichero de personas, cada registro contiene datos de una persona. Si el archivo contiene datos de 1000 personas, constará de 1000 registros.

A continuación se explican los tipos más habituales de ficheros.

Ficheros secuenciales

En estos ficheros, los datos se organizan secuencialmente en el orden en el que fueron grabados. Para leer los últimos datos hay que leer los anteriores. Es decir leer el registro número nueve, implica leer previamente los ocho anteriores.

Ventajas

- Rápidos para obtener registros contiguos de una base de datos
- No hay huecos en el archivo al grabarse los datos seguidos, datos más compactos

Desventajas

- Consultas muy lentas al tener que leer todos los datos anteriores al dato que queremos leer
- Algoritmos de lectura y escritura más complejos
- El borrado provoca archivos que no son compactos
- La ordenación de los datos requiere volver a crearlo de nuevo

Ficheros de acceso directo o aleatorio

Se puede leer una posición concreta del fichero, con saber la posición (normalmente en bytes) del dato a leer. Cuando se almacenan registros, posicionarnos en el quinto registro se haría de golpe, lo único que necesitamos saber es el tamaño del registro, que en este tipo de ficheros debe de ser el mismo. Suponiendo que cada registro ocupa 100 bytes, el quinto registro comienza en la posición 400. Lo que se hace es colocar el llamado puntero de archivo en esa posición y después leer.

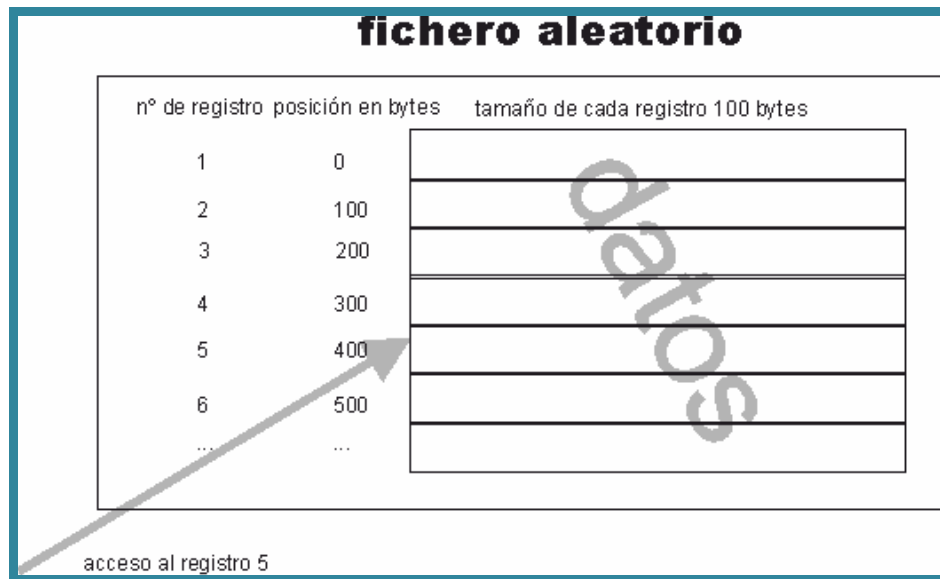


Ilustración 1, Ejemplo de fichero de acceso directo

Ventajas

- Acceso rápido al no tener que leer los datos anteriores
- La modificación de datos es más sencilla
- Permiten acceso secuencial
- Permiten leer y escribir a la vez

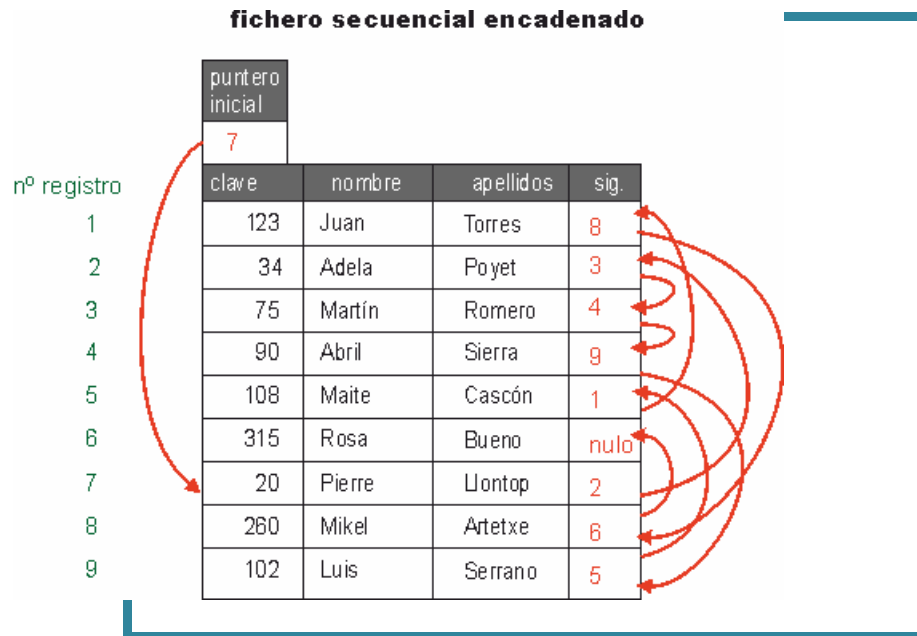
Desventajas

- Salvo en archivos relativos directos, no es apto por sí mismo para usar en bases de datos, ya que los datos se organizan en base a una clave
- No se pueden borrar datos (sí marcar para borrado, pero generarán huecos)
- Las consultas sobre multitud de registros son más lentas que en el caso anterior.

Ficheros secuenciales encadenados

Son ficheros secuenciales gestionados mediante punteros, (datos especiales que contienen la dirección de cada registro del fichero). Cada registro posee ese puntero que indica la dirección del siguiente registro y que se puede modificar en cualquier momento. El puntero permite recorrer los datos en un orden concreto.

Cuando aparece un nuevo registro, se añade al final del archivo, pero los punteros se reordenan para que se mantenga el orden.



Ventajas

- El fichero mantiene el orden en el que se añadieron los registros y un segundo orden en base al puntero.
- 🖨 La ordenación no requiere reorganizar todo el fichero, sino sólo modificar los punteros
- 🖨 Las mismas ventajas que el acceso secuencial

Desventajas

- Añadir registros requiere recalcular los punteros

Ficheros secuenciales indexados

Se utilizan dos ficheros para los datos, uno posee los registros almacenados de forma secuencial, pero que permite su acceso aleatorio. El otro posee una tabla con punteros a la posición ordenada de los registros. Ese segundo fichero es el índice, una tabla con la ordenación deseada para los registros y la posición que ocupan en

el archivo.

El archivo de índices posee sólo unas cuantas entradas en las que se indica la posición de ciertos valores claves en el archivo (cada 10, 15 ,20,... registros del archivo principal se añade una entrada en el de índices). El archivo principal tiene que estar siempre ordenado y así cuando se busca un registro, se busca su valor clave en la tabla de índices, la cual poseerá la posición del registro buscado. Desde esa posición se busca secuencialmente el registro hasta encontrarlo.

Existe un archivo llamado de desbordamiento u overflow en el que se colocan los nuevos registros que se van añadiendo (para no tener que ordenar el archivo principal cada vez que se añade un nuevo registro) este archivo está desordenado. Se utiliza sólo si se busca un registro y no se encuentra en el archivo principal. En ese caso se recorre todo el archivo de overflow hasta encontrarlo.

Para no tener demasiados archivos en overflow (lo que restaría velocidad), cada cierto tiempo se reorganiza el archivo principal. Ejemplo:

fichero secuencial indexado

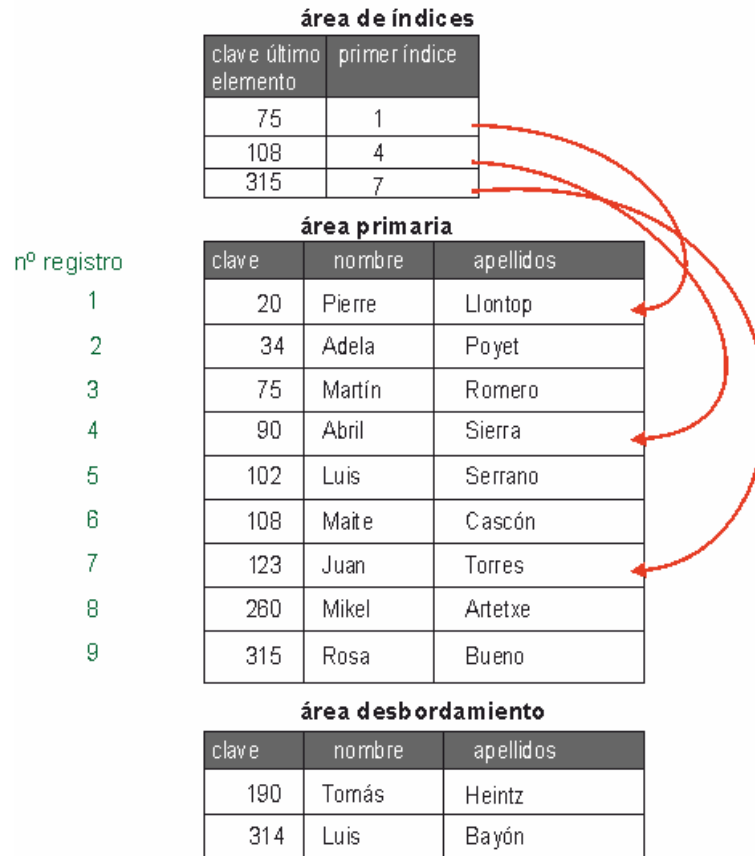


Ilustración 3, ejemplo de fichero secuencial indexado

Nota: Índices, son estructuras de datos que relacionan valores de un campo (normalmente el campo clave) de un registro con su dirección de memoria.

Ventajas

El archivo está siempre ordenado en base a una clave

- La búsqueda de datos es rapidísima
- Permite la lectura secuencial (que además será en el orden de la clave)

Desventajas

- Para un uso óptimo hay que reorganizar el archivo principal y esta operación es muy costosa ya que hay que reescribir de nuevo y de forma ordenada todo el archivo.
- La adición de registros requiere más tiempo que en los casos anteriores al tener que reordenar los índices

Ficheros indexado-encadenados

Utiliza punteros e índices, es una variante encadenada del caso anterior. Hay un fichero de índices equivalente al comentado en el caso anterior y otro fichero de tipo encadenado con punteros a los siguientes registros. Cuando se añaden registros se añaden en un tercer registro llamado de desbordamiento u **overflow**. En ese archivo los datos se almacenan secuencialmente, se accede a ellos si se busca un dato y no se encuentra en la tabla de índices.

fichero secuencial indexo-encadenado

área de índices

clave último elemento	primer índice
75	7
108	4
315	1

área primaria

puntero inicial				
7				
nº registro	clave	nombre	apellidos	sig.
1	123	Juan	Torres	8
2	34	Adela	Poyet	3
3	75	Martín	Romero	4
4	90	Abril	Sierra	9
5	108	Maite	Cascón	1
6	315	Rosa	Bueno	nulo
7	20	Pierre	Llontop	2
8	260	Mikel	Artetxe	6
9	102	Luis	Serrano	5

área desbordamiento

clave	nombre	apellidos
190	Tomás	Heintz
314	Luis	Bayón

Ilustración 4, Ejemplo de archivo secuencial indexado

Ventajas Posee las mismas ventajas que los archivos secuenciales indexados, además de una mayor rapidez al reorganizar el fichero (sólo se modifican los punteros)

Desventajas Requieren compactar los datos a menudo para reorganizar índices y quitar el fichero de desbordamiento

Hoy en día las clasificaciones de los archivos en función de su organización o utilidad ha quedado en desuso tras la aparición de las Bases de Datos porque éstas me van a garantizar:

Independencia física.

Se puede cambiar el soporte físico de la base de datos (modelo de discos, por ejemplo), sin que esto repercuta en la base de datos ni en los programas que la usan.

Independencia lógica.

Capacidad para que se puedan modificar los datos contenidos en la base, las relaciones existentes entre ellos o incluir nuevos datos, sin afectar a los programas que lo usan.

En los ficheros esto no era posible ya que por ejemplo la forma de organización del fichero o su estructura iba a depender del programa que generaba ese fichero.

Generalmente los ficheros que componen una base de datos son de tipo binario y la información se suele guardar en uno o varios ficheros. (pe en SQL Server se guardar en dos ficheros de extensiones .MDF (datos) y .LDF(registro de transacciones))

BASES DE DATOS

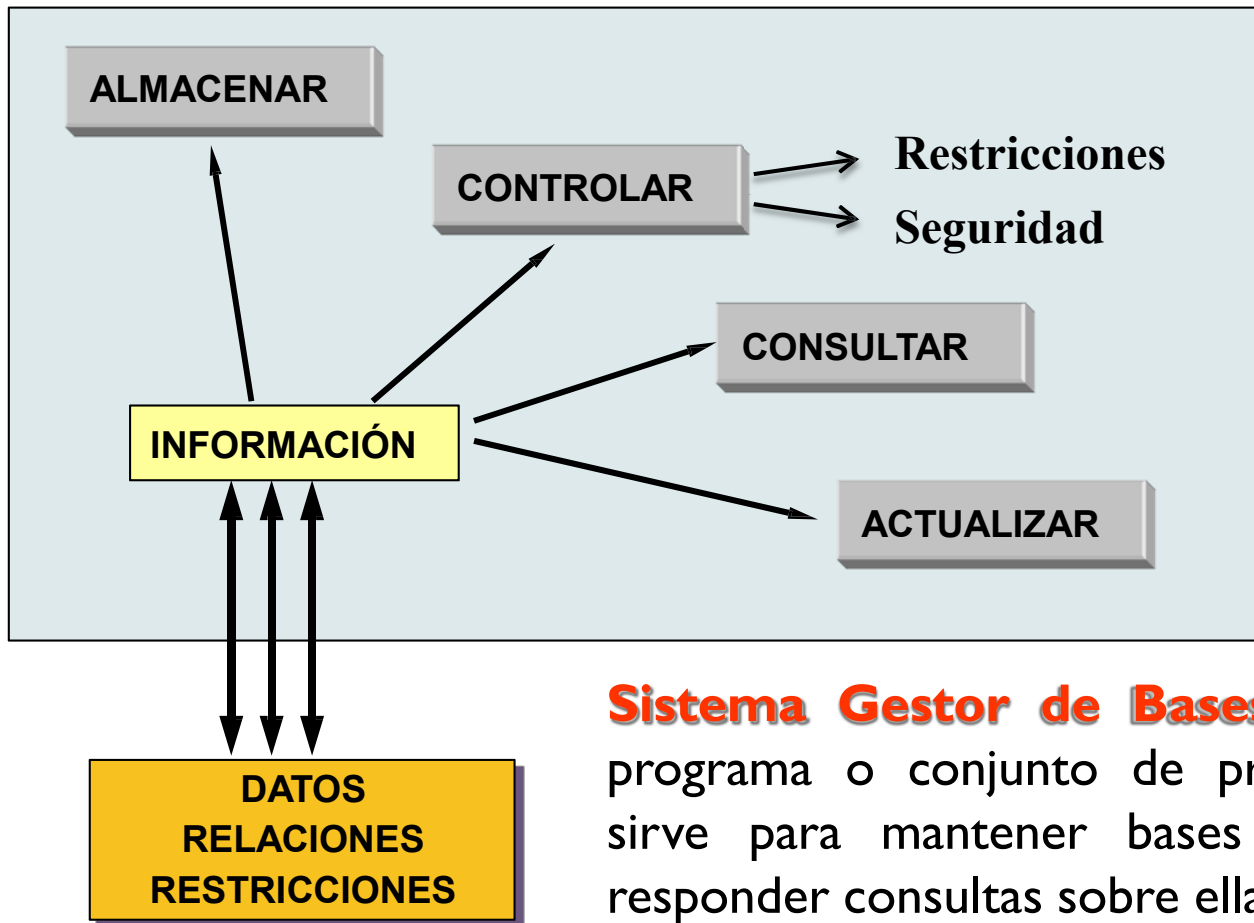
- ▶ **¿Qué es una base de datos?**
- ▶ **¿Utilizáis alguna?**
- ▶ **¿Consideráis que son importantes?**
- ▶ **¿Qué interfaces presentan?**



- ▶ Estructuras de datos gestionadas por un conjunto de programas que permiten almacenar grandes cantidades de información y manipularla de forma eficiente
- ▶ Dan soporte a:
 - ▶ Procesos transaccionales
(compras, préstamos, gestión académica, experimentos científicos...)
 - ▶ Planificación de trabajos
(Workflow, scheduler, ...)
 - ▶ Oferta de servicios
(búsquedas bibliográficas, videotecas,...)
- ▶ La inteligencia de negocio
- ▶ DW, OLAP, minería de datos



Base de Datos (def): colección organizada de datos, relativa a un problema concreto, que puede ser compartida por un conjunto de usuarios/aplicaciones. Sirven para:



Sistema Gestor de Bases de Datos: programa o conjunto de programas que sirve para mantener bases de datos y responder consultas sobre ellas.

EVOLUCIÓN DE LAS BD

- ▶ En los primeros años (década de los 70 y principios de los 80), las aplicaciones de BD se construían directamente sobre los sistemas de ficheros
- ▶ Esto tenía serias desventajas:
 - ▶ **Redundancia e inconsistencia de los datos**
 - ▶ Múltiples formatos de ficheros, duplicación de información en diferentes ficheros. Se desperdicia espacio y puede dar lugar a inconsistencias.
 - ▶ Ejemplo
 - Cuenta (nombre, direccion, tfno, n° cc, cantidad)
 - Ingreso (nombre, direccion, tfno, n° cc, cantidad, fecha)
 - ▶ **Datos aislados** — Múltiples formatos y ficheros. Más complicado acceder a los datos. La estructura física de los datos se encuentra codificada en los programas de aplicación.
 - ▶ **Problemas de integridad**
 - ▶ Restricciones de integridad como —sexo= M o F|| se encuentra en el código del

programa y no establecido explícitamente en la estructura del fichero

- ▶ Difícil y costoso el modificar o añadir nuevas restricciones (altura \geq 0, fecha \geq hoy, etc.)

-----▶ **Dificultad en el acceso a los datos**

- ▶ Necesidad de escribir un programa para realizar cada proceso

▶ **Atomicidad de las actualizaciones**

- ▶ Pérdida de consistencia por realización de actualizaciones parciales
- ▶ Ej.: Transferencia de fondos de una cuenta a otra (debe restarse una cantidad de una cuenta y sumarse esa misma cantidad a la otra, o no hacerse nada)

▶ **Acceso concurrente por múltiples usuarios**

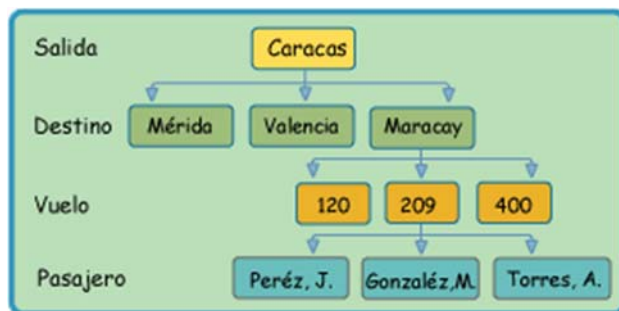
- ▶ Necesidad de acceso concurrente para incrementar el rendimiento
- ▶ Acceso concurrente bajo control para evitar inconsistencias
Ej.: Se ha de impedir que dos personas estén actualizando la misma cuenta corriente al tiempo

▶ **Problemas de seguridad**

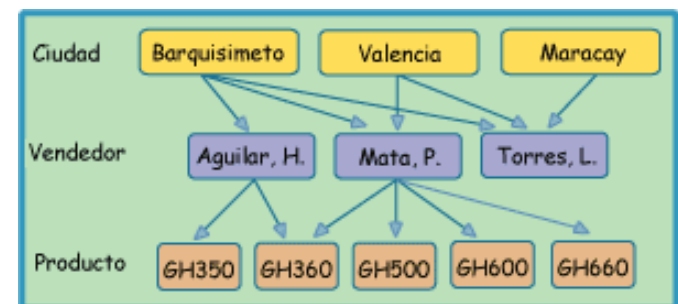
- ▶ Difícil permitir el acceso a los usuarios a parte de los datos y/o determinadas acciones (actualizar cuenta, leer datos personales, insertar clientes, etc.)

Por todo esto surgieron las Bases de Datos como modelo de organización de los datos y con ellas los sistemas de gestión de bases de datos (SGBD) como herramienta que permite la implementación y gestión de bases de datos.

- ▶ La gestión y almacenamiento de datos es el que impulsa el crecimiento del uso de las computadoras. De 1950 a 1960, se desarrollaron las cintas magnéticas para el almacenamiento de datos cuya lectura era secuencial. Apareció el lenguaje **COBOL**, primer lenguaje que tiene una parte donde se describen los datos y los ficheros que van a ser utilizados de manera independiente a las acciones que se van a realizar.
- ▶ En la década de los 70, aparecieron los discos magnéticos lo que permitió el acceso directo a los datos. Surgieron las **BD jerárquicas y en red**, en las que el tratamiento de información requería conocer detalles de implementación (uso de punteros) y la codificación de consultas se realizaba de forma procedimental
- ▶ **Codd definió el modelo relacional (1970)**
 - Modelo teórico bien fundamentado
 - Independencia lógica y física de los datos
 - Lenguaje de consulta declarativo, SQL



Jerárquica (IMS de IBM)



Red (IDMS/R de Cullinet)

- ▶ Pero hasta **1980** no aparecieron gestores relacionales comerciales (Oracle, IBM DB2, Ingres,...) con buen rendimiento y en los que el diseño y el mantenimiento de las BDs resulta más sencillo (independencia física y lógica).
- ▶ Asimismo comenzaron los:
 - ▶ Estudios de BD distribuidas y paralelas
 - ▶ Y de BD orientadas a objetos
- ▶ Desde **1990** se encuentran en el mercado:
 - ▶ BD objeto-relacionales
 - ▶ BD dimensionales (tecnología OLAP)
 - ▶ BD XML
 - ▶ BD geográficas
 - ▶ etc.

TABLA DE CLIENTES				
C.I.	Nombre	Identif/C.	Dirección	Teléfono
16325825	Rivas, Luis	RL708	23654 Santa Rosa	15325948
12035824	Torres, Yessy	TY011	2536 Calle Roma	12369581
10356528	Cruz, Carlos	CC125	2514 Av. Urdaneta	10256985

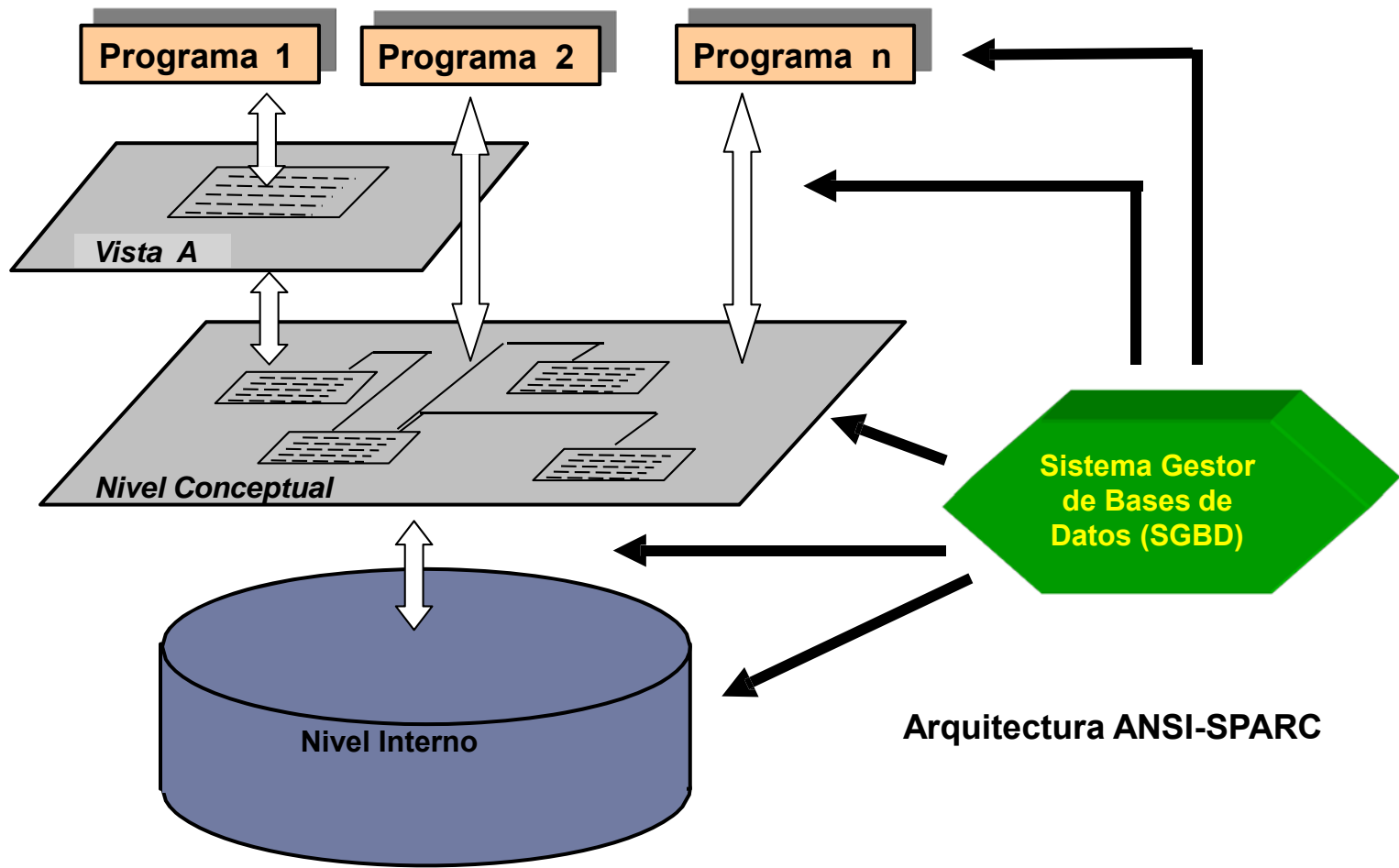
CAMPO CLAVE DE:

TABLA DE PEDIDOS					
Núm. Pedido	Identif/C.	Fecha	Monto	Embarque	Cargo envío
000454	RL708	11/02/2005	4.000.080,50	E401	10
000455	TY011	06/05/2005	1.032.200,00	E406	15
000456	CC125	07/05/2005	7.000.230,20	E900	10

Modelo relacional

Niveles de abstracción: independencia física y lógica

La finalidad de trabajar con técnicas de BD relacionales es disfrutar de una visión abstracta de los datos que facilite el desarrollo y uso de aplicaciones. Esto es, centrarse en determinar qué datos se han de almacenar y cómo se relacionan, olvidándose de los detalles internos de cómo el gestor los almacenan y gestiona.



- ▶ **ANSI-SPARC (1975)** propuso la arquitectura en tres niveles para conseguir la separación entre los programas de aplicación y los datos y poder obtener múltiples vistas de usuario
 - ▶ **Nivel interno:** Describe la estructura de almacenamiento físico de base de datos (Describe todos los detalles para el almacenamiento de la BD, ficheros, discos, directorios...). Los datos se almacenan en este nivel. Cada SGBD implementa su propio nivel interno. No estándar.
 - ▶ **Nivel conceptual o global:** Representación del conjunto de datos correspondiente al problema de información a gestionar. Recoge el esquema conceptual, esto es, la estructura de la base de datos en términos de elementos lógicos (entidades, atributos, relaciones y restricciones), ocultando los detalles físicos de almacenamiento.
 - ▶ **Nivel externo o de vistas:** Esquemas que recogen las distintas perspectivas de los usuarios y/o aplicaciones para cada proceso (compras, matrícula, expedientes, etc.). Permiten ocultar información por cuestiones de seguridad (Ley de protección de datos, datos sensibles para la seguridad de una empresa, etc.). Cada esquema externo describe la parte de la base de datos que interesa a un grupo de usuarios y oculta a ese grupo el resto de la BD. Este nivel es el que percibe el usuario final mediante el uso de las aplicaciones.

- ▶ La arquitectura de tres niveles es útil para explicar el concepto de *independencia de datos* :
 - ▶ **Independencia lógica:** capacidad de cambiar el nivel conceptual o global sin tener que cambiar las vistas ni los programas de aplicación
 - ▶ Añadir restricciones de dominio, incorporar un nuevo atributo, añadir nuevas relaciones,...
 - ▶ **Independencia física:** capacidad de cambiar el nivel interno sin tener que cambiar ni el nivel conceptual ni nivel externo.
 - ▶ Añadir índices, ampliar espacio de almacenamiento, realizar particiones, etc.

Ventajas de las BD

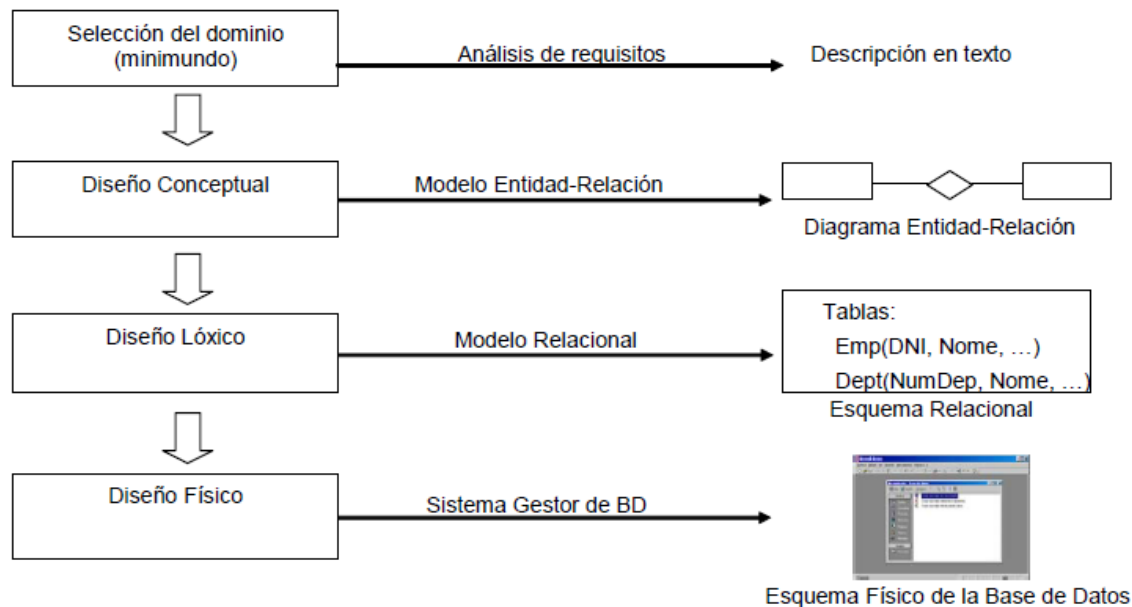
- Flexibilidad de adaptación a cada problema.
- Optimización en la gestión de la información.
- Garantiza la independencia física y lógica de los datos.
- Control de la integridad (coherencia de los datos) de los datos.
- Garantía sobre la consistencia (cumplir las restricciones) de la información.
- Facilidad de acceso concurrente.
- Protección ante fallos del sistema.
- Seguridad ante accesos restringidos.

MODELO DE DATOS

Una base de datos representa información del mundo real. Hay que extraer los datos relevantes que hay que almacenar.

El proceso de diseño

- Conseguir un buen diseño directamente no es fácil. Por ello, el problema se descompone en pasos



- La preocupación en cada paso es distinta:
 - Análisis de requisitos
 - ¿Qué información necesito almacenar?
 - ¿Qué consultas voy a querer realizar?
 - Diseño conceptual
 - ¿Cómo organizo la información?
 - Utilizo un lenguaje claro y no ambiguo
 - Diseño lógico
 - ¿Cómo represento el diseño conceptual en un ordenador?
 - El resultado debe ser utilizable en cualquier SGBD
 - Diseño físico
 - Implementación en un SGBD particular

EJEMPLO:

Supongamos que después del análisis de requisitos se obtiene la siguiente especificación:

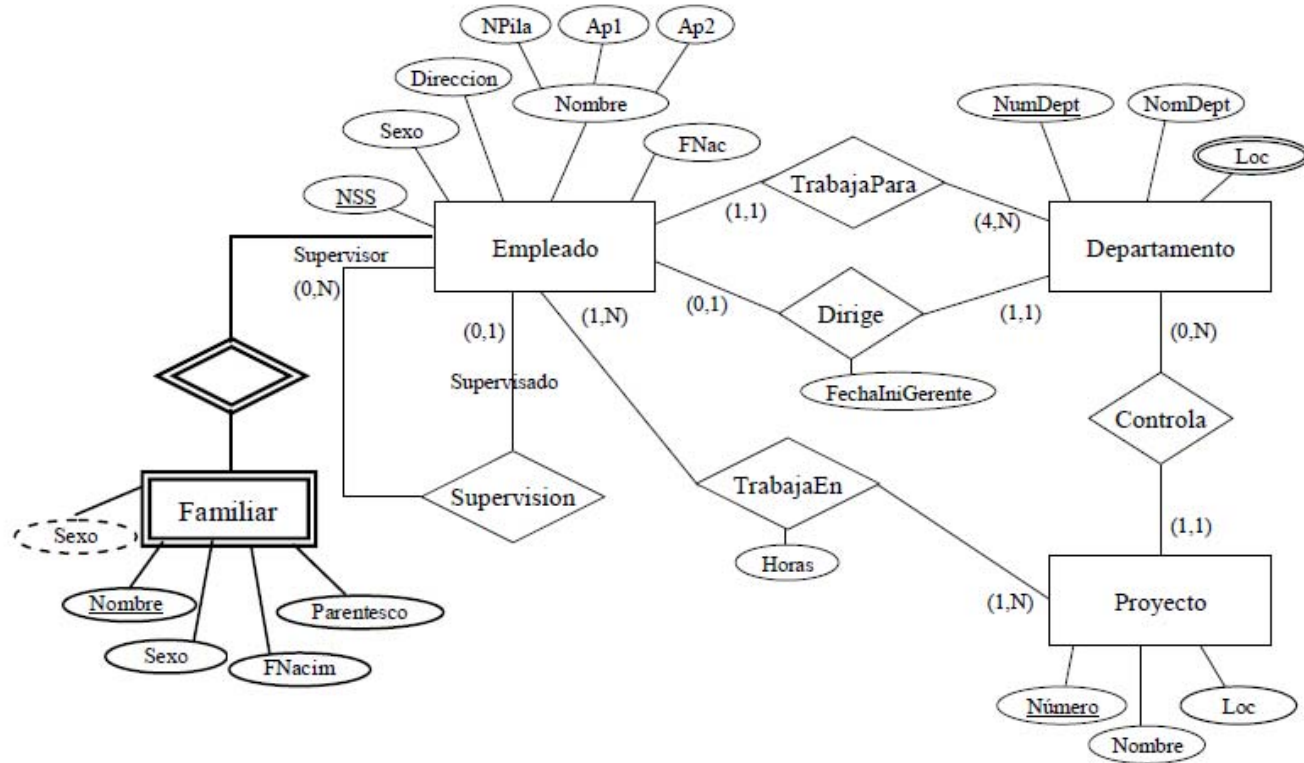
La empresa está organizada en departamentos. Cada departamento tiene un nombre único, un número único y siempre tiene un empleado que lo dirige. Nos interesa la fecha en que dicho empleado comenzó a dirigir el departamento. Un departamento puede estar distribuido en varios lugares.

Cada departamento controla un cierto número de proyectos, cada uno de los cuales tiene un nombre y número _únicos, y se efectúa en un solo lugar. Un departamento puede no estar involucrado en proyectos.

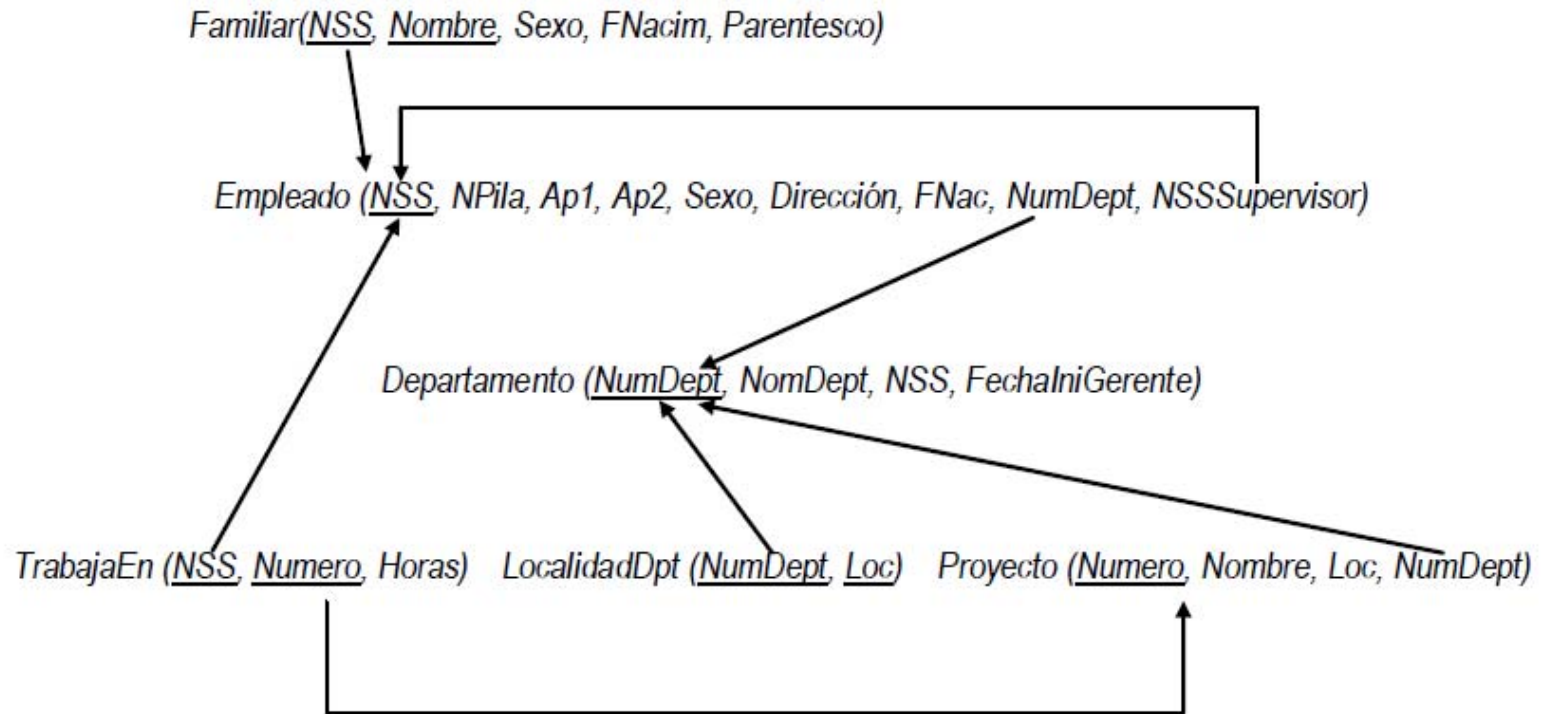
Almacenaremos el nombre, número de seguridad social, dirección, salario, sexo y fecha de nacimiento de cada empleado. Todo empleado está asignado a un departamento, pero puede trabajar en varios proyectos, que no necesariamente estarán controlados por el mismo departamento. Nos interesa el número de horas por semana que un empleado trabaja en cada proyecto, y también quién es supervisor directo de cada empleado. No todo empleado es supervisor.

Queremos mantenernos al tanto de los familiares de cada empleado para administrar sus seguros. De cada familiar almacenaremos el nombre, sexo, fecha de nacimiento y parentesco con el empleado.

Diseño conceptual



Diseño lógico (relacional)



Diseño físico.

Transformaremos el modelo relacional en modelo físico a través del sublenguaje DDL (Lenguaje de definición de datos) de SQL

```
CREATE TABLE Familiar  
(  
  NSS int NOT NULL PRIMARY KEY,  
  Nombre varchar(255),  
  Sexo nchar,  
  Fnacimiento date  
  Parentesco varchar(255)  
);
```

Un modelo de datos es una colección de herramientas conceptuales que se emplean para especificar datos, las relaciones entre ellos, la semántica asociada y las restricciones de integridad. Un modelo de datos está orientado a describir una Base de Datos. Típicamente un modelo de datos permite describir:

- Las estructuras de datos de la base: El tipo de los datos que hay en la base y la forma en que se relacionan.
- Las restricciones de integridad: Un conjunto de condiciones que deben cumplir los datos para reflejar correctamente la realidad deseada.

- Operaciones de manipulación de los datos: típicamente, operaciones de agregado, borrado, modificación y recuperación de los datos de la base.

CLASIFICACION DE LOS MODELOS DE DATOS

La arquitectura de tres niveles nos obliga a modelar nuestros datos en cada nivel. Nos vamos a centrar en el global y externo, ya que el interno es específico del software utilizado.

De acuerdo al nivel de abstracción que presentan pueden emplearse:

Modelos de Datos Conceptuales: Se usan fundamentalmente durante la etapa de análisis de un problema dado y están orientados a representar los elementos que intervienen en ese problema y sus relaciones. Con este modelo representamos los datos de forma parecida a como nosotros los captamos en el mundo real. El ejemplo más típico es el **Modelo Entidad-Relación (MER)**. Se representa la realidad a través de entidades que se distinguen de otras a través de sus características (atributos).

Modelos de Datos Lógicos: Son orientados a las operaciones más que a la descripción de una realidad. Usualmente están implementados en algún Gestor de Base de Datos. Se usan para especificar la estructura lógica global de las bases de datos y para proporcionar una descripción más estructurada y cercana a la implementación.

Los más comunes son:

- a) **Modelo Jerárquico**: utiliza árboles para la representación lógica de los datos. Un árbol está compuesto por una jerarquía de elementos llamados nodos. El nivel más alto de la jerarquía tiene un solo nodo que se llama raíz. Cada nodo representa un tipo de registro llamado segmento con sus correspondientes campos. Útiles en caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos.
- b) **Modelo Red**: utiliza estructura de datos en red donde las entidades se representan como nodos, y las relaciones como líneas que unen a los nodos. En una estructura de red cualquier componente puede vincularse con cualquier otro. Es posible describirla en términos de padres e hijos, pero, a

diferencia del modelo jerárquico, un nodo hijo puede tener varios padres. Difícil de administrar la información.

- c) **Modelo Relacional:** en el que cada ente del mundo real se almacena en un lugar diferente y posteriormente se establecen relaciones entre dichos entes.

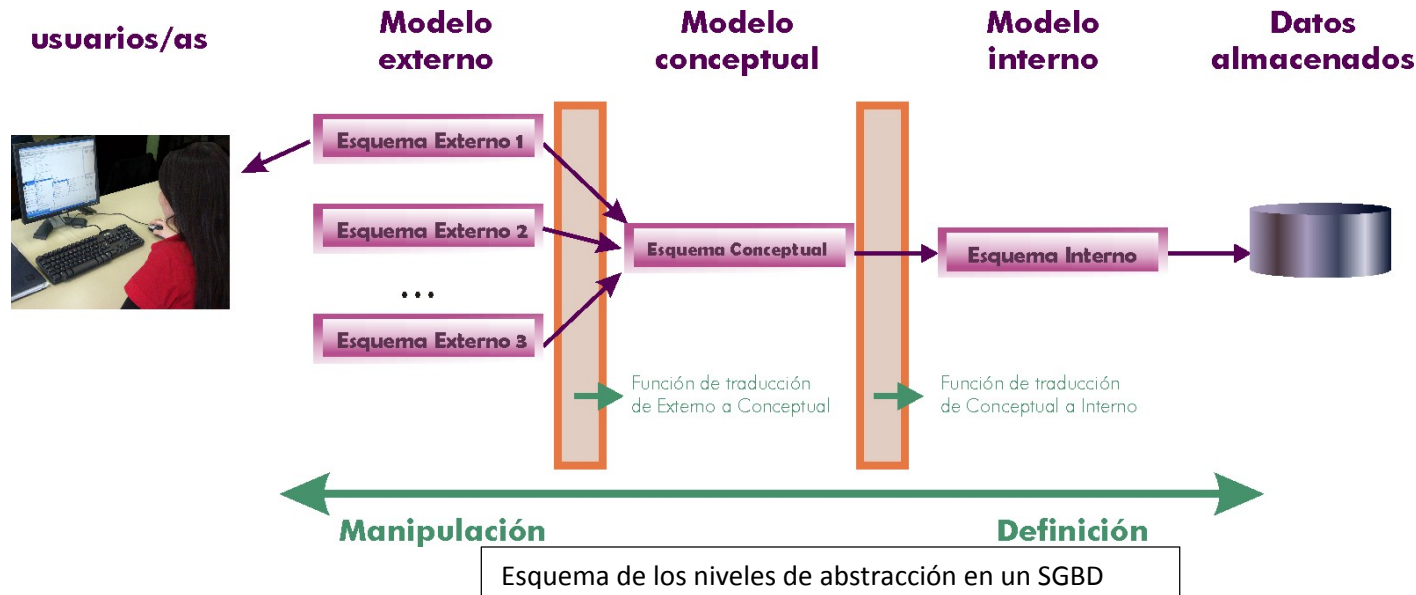
Las bases de datos relacionales se basan en el uso de tablas .Las tablas se representan gráficamente como una estructura rectangular formada por filas y columnas. Cada columna almacena información sobre una propiedad determinada de la tabla (se le llama también atributo) ej.: nombre, CI, apellidos, edad. Cada fila posee un registro de la relación representada por la tabla (a las filas se las llama también tuplas).

- d) **Modelo de datos orientado a objetos:** se usan principalmente en aplicaciones programadas con lenguajes orientados a objetos. Tratan de almacenar en la base de datos no solo los datos, sino la funcionalidad asociada (comportamiento)

SISTEMAS GESTORES DE BASES DE DATOS

El sistema gestor de bases de datos (SGBD) es una aplicación que permite a los usuarios definir, crear y mantener la base de datos y proporciona acceso controlado a la misma. Sirve de interfaz entre el usuario y las bases de datos.

Los usuarios acceden a la información usando aplicaciones (pe formulario web), estas se comunican con sistemas gestores que son los que acceden a los datos almacenados en la base de datos mediante la interacción del sistema operativo.



Objetivos de los SGBD:

- Asegurar los tres niveles de abstracción.
- Permitir la independencia física y lógica de los datos.
- Garantizar la consistencia de los datos.
- Ofrecer seguridad de acceso a los datos por parte de usuarios y grupos.
- Gestión de transacciones (conj de acciones que cambian el contenido de la base de datos) de forma que se garantice la ejecución de un conjunto de operaciones críticas como una sola excepción.
- Permitir la concurrencia de usuarios sobre los mismos datos mediante bloqueos que mantienen la integridad de los mismos.

Funciones de los SGBD:

Para conseguir dichos objetivos los SGBD incorporan las siguientes características y funciones:

- Un catálogo. Almacena las descripciones de los datos y es accesible por los usuarios. Es lo que se denomina diccionario de datos y contiene información que describe los datos de la base de datos (metadatos)
- Garantizar la integridad. Dispone de un mecanismo que garantice que todas las

actualizaciones correspondientes a una determinada transacción se realicen o que no se realice ninguna.

- Permitir actualizaciones. Asegurar que la BD se actualice correctamente cuando varios usuarios la estén actualizando concurrentemente.
- Recuperación de datos. Permitir recuperar las BD en caso de que ocurra algún imprevisto que afecte o destruya la BD.
- Integración: ser capaz de integrarse con algún software de comunicación de manera que no afecte cómo acceden los usuarios a las BD.
- Cumplir-restricciones. Debe proporcionar las herramientas necesarias que permitan garantizar que tanto los datos de la base de datos como los cambios que se realizan sobre estos datos sigan ciertas reglas.
- Herramientas de administración. Que permitan administrar la BD de forma efectiva, garantizar la integridad y disponibilidad de los datos, controlar el acceso al servidor...

Componentes de un SGBD:

Núcleo: Es el conjunto de programas que controlan y coordinan el funcionamiento del sistema gestor de base de datos.

Objetos: Tablas y vistas, consultas, dominios y tipos definidos de datos, Restricciones de tabla y dominio, funciones y procedimientos almacenados, disparadores o triggers.

Lenguajes de los SGBD: Todos los SGBD ofrecen lenguajes e interfaces apropiadas para cada tipo de usuarios: administradores, diseñadores, programadores y usuarios finales. A través del lenguaje el usuario realiza las preguntas al servidor, contestando este a sus demandas.

Los lenguajes del SGBD se clasifican en:

- Lenguaje de definición de datos (DDL): Se utiliza para especificar el esquema de la base de datos, las vistas de los usuarios y las estructuras de almacenamiento. Es el que define el esquema conceptual y el esquema interno. Lo utilizan los diseñadores y los administradores de la base de datos.
- Lenguaje de manipulación de datos (DML): Se utiliza para leer y actualizar los datos de la base de datos. Es el que emplean los usuarios para realizar consultas, inserciones, eliminaciones y modificaciones.
- Lenguaje de control de datos (DCL): Encargado del control y seguridad de los datos (privilegios, modos de acceso, etc.). Permite especificar los permisos sobre los objetos de las

bases de datos, así como crear y eliminar usuarios y cuentas.

Para todos estos lenguajes se usa principalmente el lenguaje SQL. Incluye instrucciones para los tres tipos de lenguajes comentados. SQL se ha convertido en el lenguaje estándar de los SGBD relacionales

Utilidades: Son aplicaciones que facilitan el trabajo a los usuarios y programadores. Tienen la característica común de tener un interfaz fácil de entender.(pe herramientas para seguridad, control de concurrencia, control de recuperación...)

Diccionario de Datos: es el lugar donde se deposita información acerca de todos los datos que forman la BD. Es una guía en la que se describe la BD y los objetos que la forman. En una BD relacional, el diccionario de datos proporciona información acerca de:

- La estructura lógica y física de la BD.
- Las definiciones de todos los objetos de la BD: tablas, vistas, índices, disparadores, procedimientos, funciones, etcétera.
- El espacio asignado y utilizado por los objetos.
- Los valores por defecto de las columnas de las tablas.
- Información acerca de las restricciones de integridad.
- Los privilegios y roles otorgados a los usuarios.
- Auditoría de información, como los accesos a los objetos

Usuarios: En los sistemas de gestión de bases de datos existen diferentes categorías

de usuarios, estas categorías se caracterizan porque tienen una serie de privilegios o permisos sobre los objetos que forman la base de datos.

Hay tres tipos de usuarios:

- *Programadores de aplicaciones*: Se encargan de diseñar y programar las aplicaciones necesarias para la utilización de la B.D., realizando las peticiones pertinentes al SGBD.
- *Usuario final*: Es la persona que se dedica a trabajar sobre los datos almacenados en la B.D. Hay usuarios finales avanzados que por medio del lenguaje de programación SQL pueden acceder a los datos.
- *Administrador de B.D.*: Es el usuario más importante de los tres, ya que es el que se encarga de diseñar y modificar la estructura de la B.D.

SISTEMAS GESTORES DE BASES DE DATOS V

Tipos de SGB

□ Según modelo lógico:

- Jerárquico
- En red
- Relacional
- Objeto-relacional
- Orientado a objetos

□ Según número de sitios:

- Centralizado
- Distribuido

□ Según tipo de datos:

- Relacionales
- XML
- Objeto-relacionales
- Orientados a objetos

□ Según número de usuarios:

- Monousuario
- Multiusuario

□ Según Ámbito de aplicación

- Propósito general
- Propósito específico

□ Según lenguajes soportados:

- SQL
- NoSQL

SISTEMAS LIBRES y COMERCIALES

Libres

Disponibles de forma abierta en la red (no necesariamente gratuitos) y modificables según licencia GPL

- MySQL
- PostgreSQL
- LibreOffice Base
- Redis
- MongoDB
- Apache couchDB

Comerciales

Requieren pago de licencias aunque suelen ofrecer versiones gratuitas limitadas

- ORACLE
- SQL server
- Microsoft Access

Pero, ¿Siempre es aconsejable usar bases de datos relacionales?

✦ Ficheros

- Aplicaciones sencillas cuyo esquema es poco probable que cambie
- Cuando no hay acceso multiusuario

✦ NoSQL (non-relational databases)

- Grandes volúmenes de datos sin requisitos ACID (transacciones)
- Sistemas distribuidos y escalables. Clusters de servidores.
 - Ej: Google (BigTable), Facebook (Cassandra), etc.