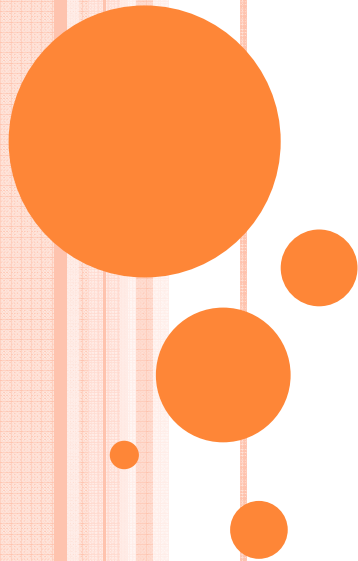


TEMA 1

Sistemas de almacenamiento de la información



1. INTRODUCCIÓN

- Hoy en día cualquier empresa u organización empresarial o no, necesita manejar gran cantidad de datos (información) y procesar esta información con rapidez para obtener nueva información.
- La mejor manera de procesar esta información es mediante un sistema informatizado (ordenadores) y es entonces cuando se nos plantea **cómo almacenar estos datos** y trabajar con ellos.
- La primera opción fue la utilización de **archivos**, las aplicaciones y los programas se desarrollaban independientemente y cada uno de ellos procesaba sus propios archivos.
- A finales de los años setenta surgen las **bases de datos** con la idea de que los datos se relacionen entre sí y puedan compartirse por varios usuarios con garantías de integridad (los datos se conserven correctos)



2. SISTEMAS GESTORES DE FICHEROS

- Anteriormente a la aparición de los **Sistemas Gestores de Bases de Datos** SGBD (década de los setenta) la información se almacenaba en **sistemas de gestión de archivos**.
- Un sistema gestor de ficheros consiste en un conjunto de programas que define y trabajaba con datos organizados y almacenados en archivos (ficheros). Los programas manejan estos archivos para obtener información.
- Si la estructura de un archivo cambia, todos los programas que utilizan este archivo tienen que modificarse, dado que la definición de los datos se encuentra codificada dentro de los programas de aplicación en vez de almacenarse de forma independiente.
- Esto supone un gran inconveniente a la hora de tratar grandes volúmenes de información.



2. SISTEMAS GESTORES DE FICHEROS

- Surge así la idea de separar los datos contenidos en los archivos de los programas que los manipulan, es decir, que se pueda modificar la estructura de los datos de los archivos sin que por ello se tengan que modificar los programas con los que trabajan.
- Se trata de estructurar y organizar los datos de forma que se pueda acceder a ellos con independencia de los programas que los gestionan.



2.1.REGISTROS

- La información se almacena en forma de **registros**, que a su vez están formados por elementos llamados **campos** de registro.
- Por ejemplo, un registro de alumno incluiría campos como el *nombre, fecha de nacimiento o teléfono*, cuyos valores para cada alumno forman cada registro.
- Cada campo tiene un tipo de dato que especifica el tipo de valores que puede tomar.
- Por ejemplo, tendríamos que para el nombre del alumno usaríamos un tipo carácter o *char*, para la fecha de nacimiento un tipo de fecha y para la edad un tipo numérico entero o *int*.



2.2.ARCHIVOS O FICHEROS

- Podemos definir un **fichero informático** o **archivo** como un conjunto de registros, grabados sobre un soporte que pueda ser leído por el ordenador.
- Los registros que contiene un archivo pueden tener longitud fija (todos los registros son iguales en tamaño), o variable (si existen campos de tamaño variable u opcionales)
- Los archivos de registros de longitud fija son más fáciles de manipular por parte de los programas, sin embargo desperdician espacio en disco.
- Por el contrario, para el caso de longitud variable los programas son más complejos ya que los registros no ocupan posiciones fijas, sino que dependen del valor de cada campo.



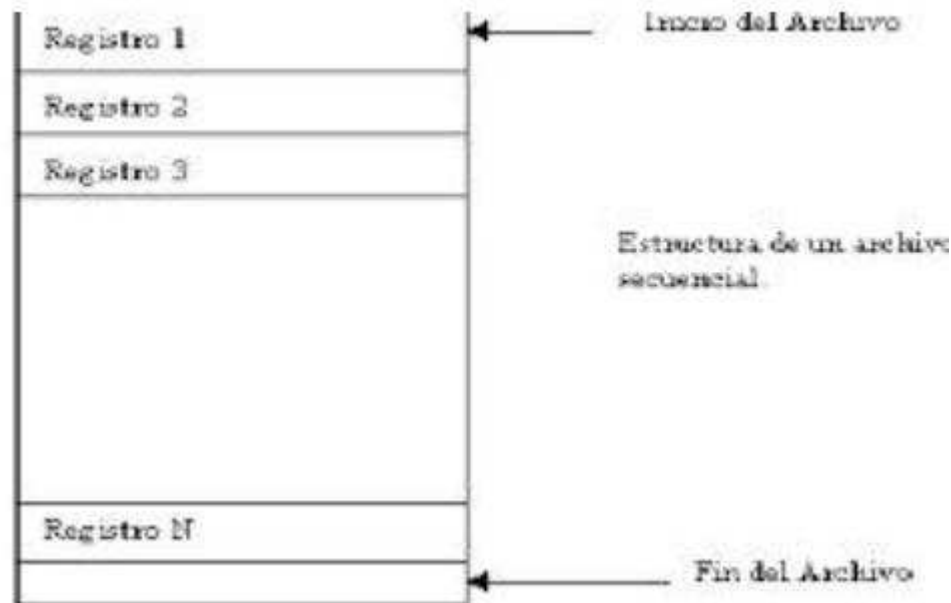
2.2.1 TIPOS DE FICHEROS

- El término **organización** de ficheros se aplica a la forma en que se colocan los datos contenidos en los registros de cada fichero sobre el soporte informático (disco, cinta, etc.) durante su grabación.
- Si clasificamos los ficheros en función de su organización ó método de acceso a los datos:
 - **Secuenciales:** los registros se van grabando unos a continuación de los otros, en el orden que se van dando de alta. La inserción es muy eficiente ya que siempre se hace en el último bloque disponible, sin embargo, la búsqueda de datos resulta más complicada ya que se requiere una búsqueda lineal, bloque por bloque hasta llegar al registro buscado.



2.2.1 TIPOS DE FICHEROS

- **Secuenciales:**



- En cuanto al **borrado de datos** es muy ineficiente ya que al eliminar registros y no ocupar el espacio liberado quedan huecos en los bloques que, salvo que se elimine el fichero, no serán reutilizados. Aunque hay técnicas para aprovechar estos espacios lo mejor es reorganizar el fichero periódicamente, de modo que se empaqueten los registros eliminando los borrados.

2.2.1 TIPOS DE FICHEROS

- **Secuenciales:** la **modificación** es más complicada ya que implica la búsqueda del registro y su reescritura. Esto es especialmente problemático en el caso de usar registros de longitud variable ya que puede ocurrir que el nuevo registro no quepa en el bloque, en cuyo caso se debe eliminar del bloque y añadirlo al final del fichero como si fuese una inserción.
- **Aleatorios:** los registros se graban en las posiciones que les corresponda según el valor que guarden en el campo denominado **clave**, que actúa como campo de ordenación, que permite identificar el registro dentro del fichero y cuyos valores son distintos para cada registro.

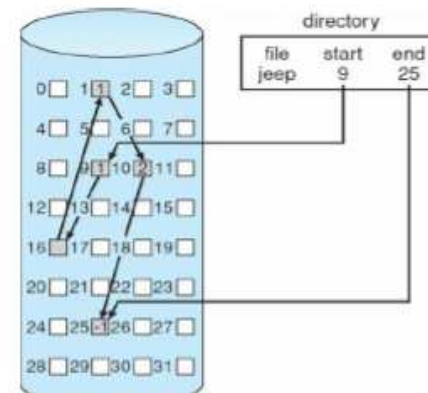


2.2.1 TIPOS DE FICHEROS

- **Aleatorios:** La **inserción** es costosa, ya que debe mantenerse el orden, lo que puede implicar desplazamiento de registros para insertar uno nuevo en el orden apropiado.

La **modificación** no ofrece problemas si no afecta al campo de ordenación, salvo que el registro sea de tamaño variable y no quepa en el bloque con los nuevos valores. Si se quiere modificar el campo de ordenación deberá reinsertarse en la posición correspondiente según el valor de dicho campo.

En cuanto a la **eliminación** es menos costosa si se usan registros marcados y se reorganiza el archivo periódicamente.

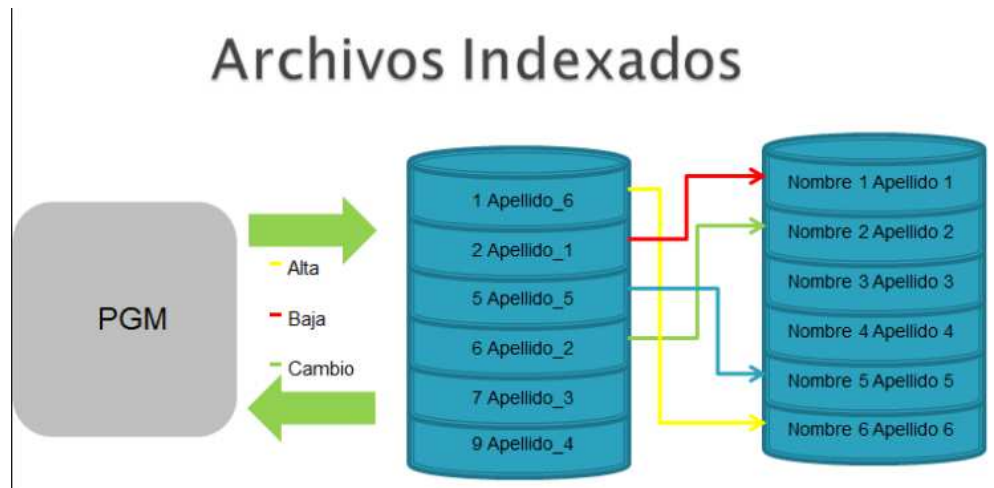


2.2.1 TIPOS DE FICHEROS

- **Indexados:** Los registros con los datos se graban en un fichero secuencialmente, pero se pueden recuperar con acceso directo gracias a la utilización de un fichero adicional, llamado **índice**, que contiene información de la posición que ocupa cada registro en el fichero de datos.

En el caso de los archivos de acceso aleatorio, la clave era siempre el número de registro.

Simulan la forma en que el índice de un libro facilita el acceso a sus contenidos.



2.3. INCONVENIENTES DE UN SISTEMA DE GESTIÓN DE ARCHIVOS

- **Redundancia e inconsistencia de los datos:** Dado que son distintos programas los que crean los archivos puede darse la situación de que trabajen con distintos formatos y los datos estén duplicados en varios sitios. Por ejemplo el teléfono de un alumno puede aparecer en más de un archivo, con formato distinto y a la hora de actualizarlo pueden crearse inconsistencias si no es modificado en todos los archivos en los que aparece.
- **Dependencia de los datos física-lógica:** La estructura física de los datos (definición de archivos y registros) se encuentra codificada en los programas de aplicación. Cualquier cambio en esta estructura implica al programador identificar, modificar y probar todos los programas que manipulan esos archivos.



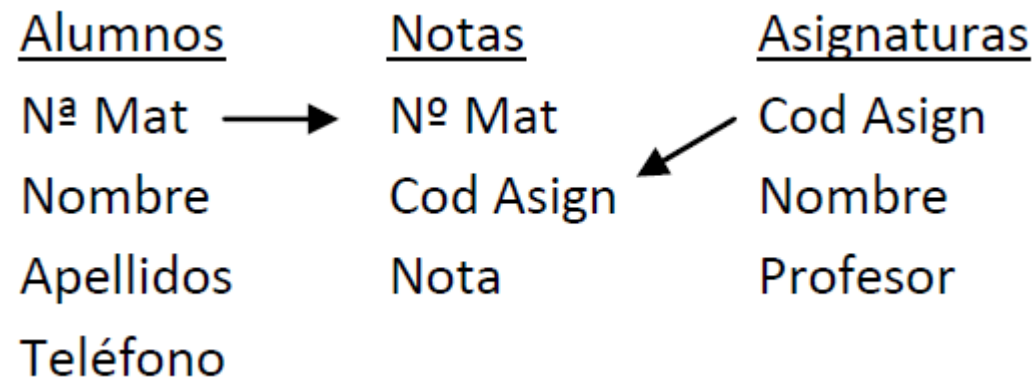
2.3. INCONVENIENTES DE UN SISTEMA DE GESTIÓN DE ARCHIVOS

- **Dificultad para el acceso concurrente:** las actualizaciones concurrentes por varios usuarios pueden llevar fácilmente a datos inconsistentes
- **Problemas de integridad de datos:** los valores almacenados en los archivos deben cumplir con restricciones de consistencia. (Por ejemplo, no puede haber una nota en el fichero de Notas de un alumno que no exista en el de Alumnos, no se debe poder borrar un alumno de Alumnos si tiene notas en Notas) Esto implica tener que añadir código a muchos programas.



3. BASES DE DATOS

- Debido a los inconvenientes anteriormente comentados desaparecen los ficheros independientes para dar paso a las **bases de datos** como un todo indivisible.
- Una base de datos se define como un conjunto de datos interrelacionados entre sí, cuyo contenido pueden compartir varios usuarios con integridad máxima (medidas de seguridad para conservar correctos los datos) y redundancia mínima (evitar repeticiones en los datos).



3. BASES DE DATOS

◦ **Integridad de los datos**

- No existan registros duplicados
- Datos dos tablas relacionadas:
 - No puede haber un registro de Notas sin su correspondiente registro relacionado de Alumnos
 - No puede borrarse un Alumno si existen notas para ese alumno
- Que se cumplan ciertas restricciones definidas con anterioridad (*por ejemplo que la nota esté entre 0 y 10*)



3. BASES DE DATOS

- **Redundancia mínima**

- Redundancia significa repetición, a veces resulta inevitable pero en una BD bien diseñada se minimiza
 - Alumnos: N°Mat Nombre Apell Dir
 - Notas: N° Mat Cod Asig Nota
 - Mejor que Notas: Nombre Apell Cod Asig Nota

- **Compartición de datos:** Las BD se construyen en general para ser manejadas por varios usuarios, a menudo al mismo tiempo. Es necesario:

- **Gestión de permisos:** cada usuario tendrá acceso a determinados datos y operaciones
- **Gestión de concurrencia:** cuando varios usuarios intentan realizar operaciones que afecten a los mismos datos ha de existir un mecanismo que impida la pérdida o inconsistencia de los datos, por ejemplo, un registro está bloqueado por un usuario hasta que este acabe de actualizarlo.

3.1 ARQUITECTURA DE LAS BASES DE DATOS

- Con una BD trabajan usuarios e informáticos. Ambos deben conocer la BD pero no del mismo modo.
- Los **usuarios** de una BD:
 - Deben tener una visión lo más abstracta posible de los datos almacenados en ella
 - No tienen necesidad de saber cómo están organizados y almacenados los datos
- Únicamente necesitan una visión de los datos que les permita interpretarlos y manejarlos.
- Sin embargo, un experto informático o el **administrador** de la base de datos deberá conocer la ubicación de los archivos físicos y un **programador** tener claro los campos que relacionan las tablas.



3.1 ARQUITECTURA DE LAS BASES DE DATOS

- En 1975, el comité ANSI-SPARC (American National Standard Institute – Standards Planning and Requirements Committee) propuso una arquitectura de **tres niveles de abstracción según la visión que se tenga de los datos**:
 - **Nivel interno o físico.**
 - Es la representación más cercana al almacenamiento físico de los datos
 - Describe la estructura física de la BD mediante el esquema interno
 - Se describen los archivos que contienen la información, su organización, ubicación, la forma de acceso a sus registros, los tipos de registro, su longitud, los campos que los componen, los índices, etc.




3.1 ARQUITECTURA DE LAS BASES DE DATOS

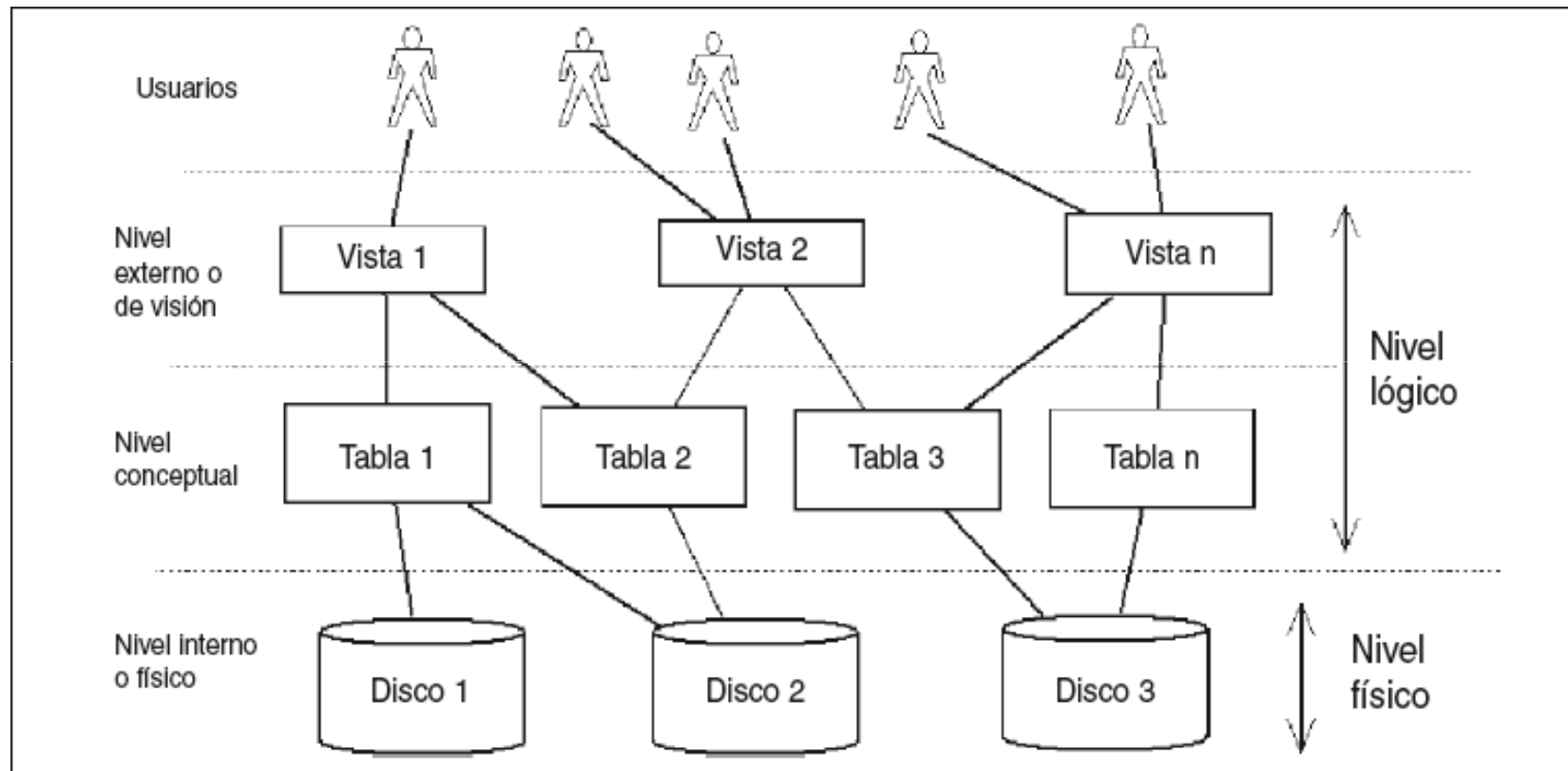
- **Nivel conceptual**

- Describe la estructura de toda la BD mediante el esquema conceptual o global
- A partir de los requerimientos de los usuarios se obtiene una visión global.
- Se describen las entidades, atributos, relaciones, operaciones de los usuarios y restricciones, ocultando los detalles de las estructuras físicas de almacenamiento.

- **Nivel externo o de visión.**

- Es el más cercano a los usuarios, es decir, es donde se describen varios esquemas externos o vistas de usuarios. Cada esquema describe la parte de la BD que interesa a un grupo de usuarios en este nivel se representa la visión individual de un usuario o de un grupo de usuarios.
- 

3.1 ARQUITECTURA DE LAS BASES DE DATOS



3.1 ARQUITECTURA DE LAS BASES DE DATOS

Caso práctico

- 1 Vamos a representar la arquitectura de una base de datos (relacional) para un centro de estudios. Definimos un esquema interno, un esquema conceptual y varios esquemas externos.

ESQUEMA EXTERNO: Visión parcial de las tablas de la base de datos según el usuario.

Vista para el programa de listado de notas de alumnos con los siguientes datos: Curso, Nombre, Nombre Asignatura y Nota.

Curso	Nombre	Nombre Asignatura	Nota
1	Ana	Programación en lenguajes estructurados	6
1	Ana	Sistemas informáticos multiusuario y en red	8
2	Rosa	Desa. de aplic. en entornos de 4ª Generación y H. Case	5
2	Juan	Desa. de aplic. en entornos de 4ª Generación y H. Case	7
1	Alicia	Programación en lenguajes estructurados	5
1	Alicia	Sistemas informáticos multiusuario y en red	4

Tabla 1.1. Esquema externo.

3.1 ARQUITECTURA DE LAS BASES DE DATOS

ESQUEMA CONCEPTUAL: Definición de todas las tablas, columnas y restricciones.

Tabla ALUMNOS. Columnas: Matrícula, Nombre, Curso, Dirección, Población. Clave: Matrícula.

Matrícula	Nombre	Curso	Dirección	Población
11111	Ana	1	C/ Pílon 10	Oropesa
11110	Rosa	2	C/ Las Viñas 26	Lagartera
11122	Juan	2	C/ Amapolas 24, 3F	Berrocalejo
23445	Alicia	1	C/ Lámina 34	Caleruela

Tabla 1.2. Tabla ALUMNOS.

Tabla ASIGNATURAS. Columnas: Código, Nombre Asignatura. Clave: Código.

Código	Nombre Asignatura
1	Desa. de aplic. en entornos de 4.ª Generación y H. Case
2	Programación en lenguajes estructurados
3	Sistemas informáticos multiusuario y en red

Tabla 1.3. Tabla ASIGNATURAS.

3.1 ARQUITECTURA DE LAS BASES DE DATOS

Tabla NOTAS. Columnas: Matrícula, Código Asignatura, Nota.

Matrícula	Código Asignatura	Nota
11111	2	6
11111	3	8
11110	1	5
11122	1	7
23445	2	5
23445	3	4

Tabla 1.4. Tabla NOTAS.

ESQUEMA INTERNO: Almacenamiento físico de los datos.

Archivo de índices para ALUMNOS: Clave alumno, Dirección de la fila.

Archivo de índices para ASIGNATURAS: Clave asignatura, Dirección de la fila.

Archivo de ALUMNOS: Matrícula, Nombre, Curso, Dirección, Población.

Archivo de ASIGNATURAS: Código, Nombre Asignatura.



3.1 ARQUITECTURA DE LAS BASES DE DATOS

Archivo de NOTAS: Matrícula, Código Asignatura, Nota.

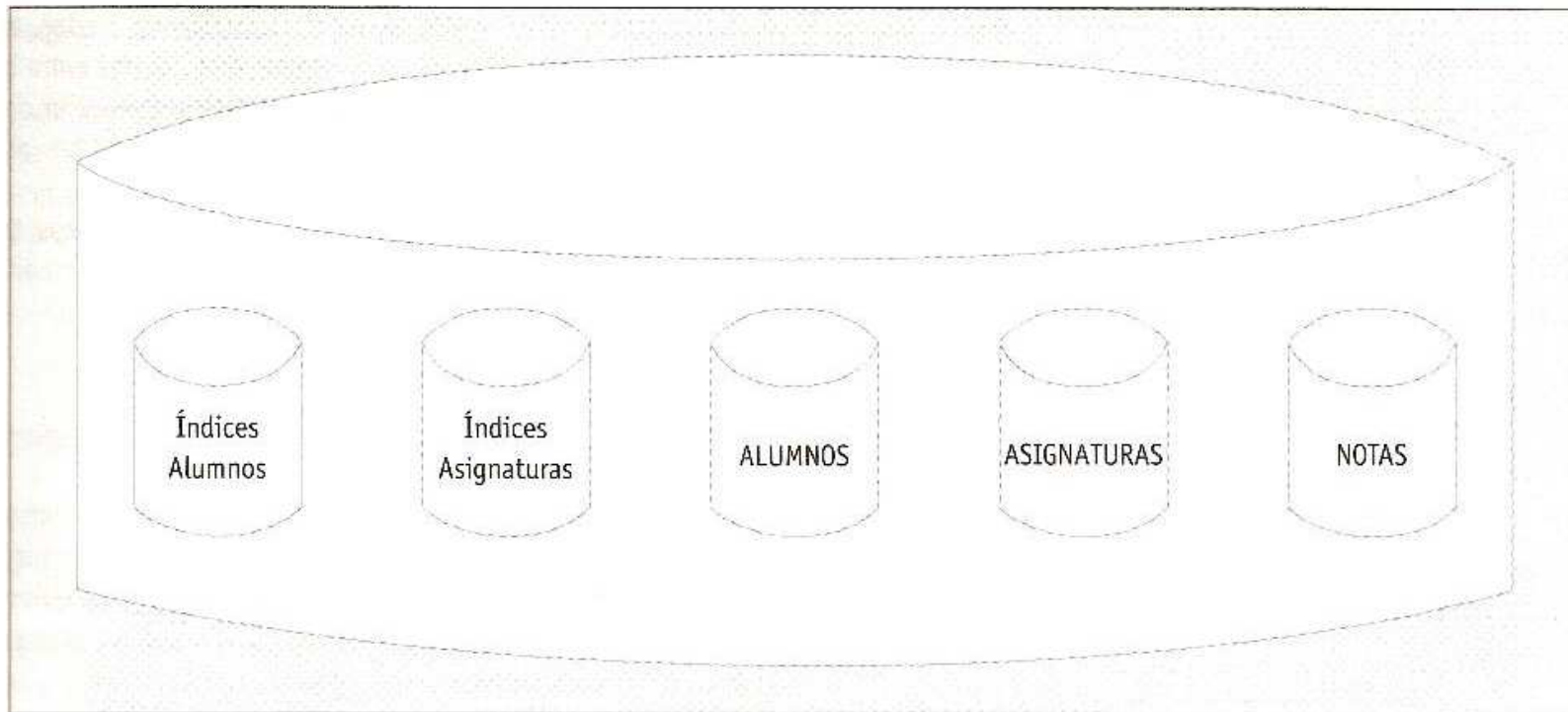
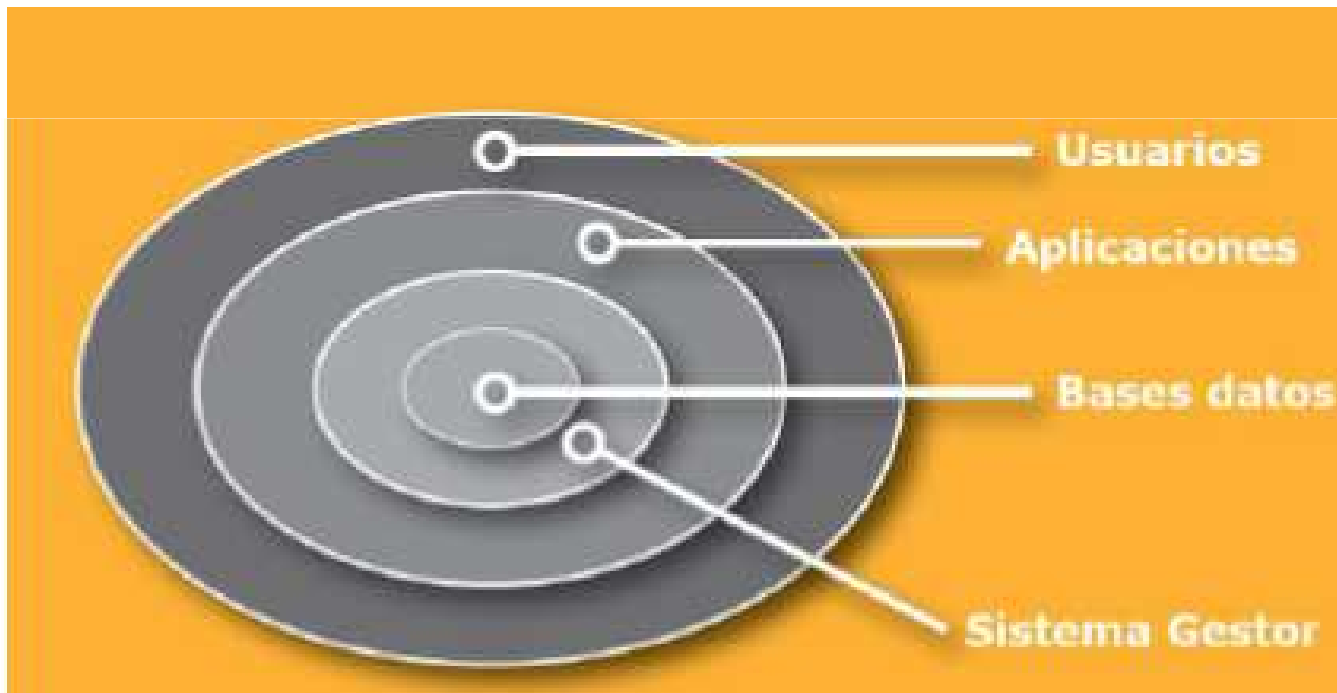


Figura 1.2. Esquema interno.



4. SISTEMA GESTOR DE BASE DE DATOS

- También llamado *Data Base Management System (DBMS)*
- Es un conjunto de programas que permiten administrar y gestionar la información de una base de datos.



4.1 USUARIOS DE LOS SGBD

- **Administradores:**

- Trabajan en el nivel de abstracción físico relacionado con el almacenamiento.
- Distinguimos los administradores del propio sistema gestor encargados de la instalación y configuración del sistema, del control de acceso a los recursos, de la seguridad y de la monitorización y optimización del sistema gestor.
- Por su parte, los administradores de bases de datos se encargan del diseño físico de la misma, implementación y mantenimiento de la base de datos.



4.1 USUARIOS DE LOS SGBD

- **Diseñadores de la base de datos**

- Realizan el diseño lógico de la base de datos, debiendo identificar los datos, las relaciones entre datos y las restricciones sobre los datos y sus relaciones.
- El diseñador de la base de datos debe tener un profundo conocimiento de los datos de la empresa y también debe conocer sus reglas de negocio. Las reglas de negocio describen las características principales de los datos tal y como los ve la empresa.
- Para obtener un buen resultado, el diseñador de la base de datos debe implicar en el desarrollo del modelo de datos a todos los usuarios de la base de datos, tan pronto como sea posible.



4.1 USUARIOS DE LOS SGBD

- **Programadores de aplicaciones**

- Estas aplicaciones servirán a los usuarios finales para, de una forma amigable, poder consultar datos, insertarlos, actualizarlos y eliminarlos.

- **Usuarios finales**

- Trabajan en el nivel externo mediante vistas o porciones de las bases de datos. Son clientes de las bases de datos que hacen uso de ellas sin conocer en absoluto su funcionamiento y organización interna. Son personas con pocos o nulos conocimientos de informática.



4.2. FUNCIONES DE LOS SGBD

- Permiten a los usuarios almacenar datos, acceder a ellos y actualizarlos de forma sencilla y con un gran rendimiento.
- Garantizan la integridad de los datos, respetando las reglas y restricciones que dicte el programador de la base de datos.
- Integran un sistema de seguridad que garantiza el acceso a la información exclusivamente a aquellos usuarios que dispongan de autorización.
- 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.
- Permiten la concurrencia, es decir, varios usuarios trabajando sobre un mismo conjunto de datos.



4.3.Lenguajes de los SGBD


- La principal herramienta de un gestor de bases de datos es la interfaz de programación con el usuario.
- Esta interfaz consiste en un lenguaje mediante el cual el usuario realiza preguntas al servidor, contestando éste a las demandas del usuario.
- Este lenguaje se denomina SQL (Structured Query Language)
- Se divide en 4 sublenguajes:
 - **Lenguaje DML o lenguaje de manipulación de datos:** permite seleccionar datos (SELECT), insertar datos (INSERT), modificarlos (UPDATE) y borrarlos (DELETE).
 - **Lenguaje DDL ó lenguaje de definición de datos:** permite crear toda la estructura de la base de datos (CREATE, DROP, etc).

4.3.Lenguajes de los SGBD

- **Lenguaje DCL o lenguaje de control de datos:** incluye comandos que permiten al administrador gestionar el acceso a los datos contenidos en la base de datos (GRANT, REVOKE).
- **Lenguaje TCL ó lenguaje de control de transacciones:** permite ejecutar varios comandos de forma simultánea como si fuera un comando atómico o indivisible.



4.4 TIPOS DE SGBD

- Existen numerosos SGBD en el mercado que podemos clasificar según los siguientes criterios:
 - **Modelo lógico en el que se basan**
 - Jerárquico.
 - En red.
 - Relacional.
 - Objeto-relacional.
 - Orientado a objetos.
 - **Número de usuarios**
 - Monousuario: solo permiten un usuario.
 - Multiusuario: permiten la conexión de varios usuarios.
 - **Número de sitios**
 - Centralizados: en un solo servidor o equipo.
 - Distribuidos: en varios equipos que pueden ser homogéneos y heterogéneos.
- 

4.4 TIPOS DE SGBD

- Existen numerosos SGBD en el mercado que podemos clasificar según los siguientes criterios:
 - **Software libre o propietarios**
 - SGBD comerciales: Oracle, SQL Server...
 - SGBD código libre con licenciamiento dual: MySQL (actualmente de Oracle, aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso)
 - SGBD libre: PostgreSQL SGBD orientado a objetos y libre



4.4 TIPOS DE SGBD

- Existen numerosos SGBD en el mercado que podemos clasificar según los siguientes criterios:
 - **Tipos de datos**
 - Sistemas relacionales estándar: manejan tipos básicos (int, char, etc.).
 - XML: para el caso de bases de datos que trabajan con documentos xml.
 - Objeto-relacionales: para bases relacionales que incorporan tipos complejos de datos.
 - De objetos: para bases de datos que soportan tipos de objeto con datos y métodos asociados.



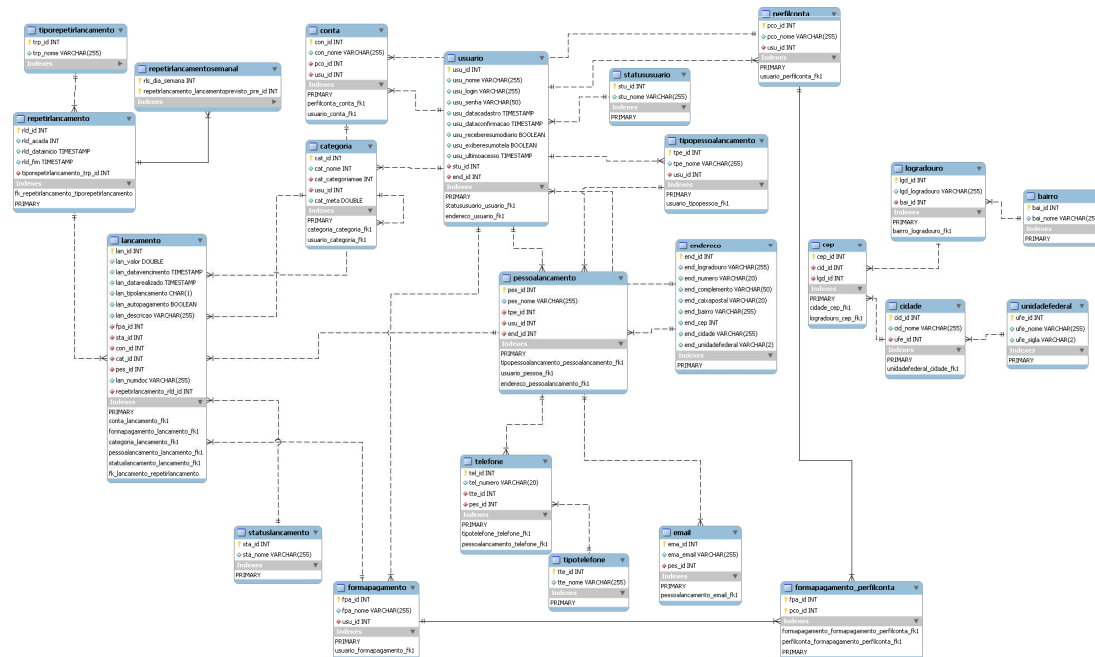
5. MODELOS DE DATOS

- Un **modelo de datos** es una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos y sus restricciones.
- Una vez realizado el diseño conceptual de la base de datos, el esquema resultante ha de traducirse a un modelo lógico de datos.
- Los modelos más difundidos son:
 - Modelo Relacional (basado en tablas como ORACLE)
 - Modelo Jerárquico (IMS de IBM, SYSTEM 2000 de Intel)
 - Modelo en red (DMS 1100 de UNIVAC, IDMS)



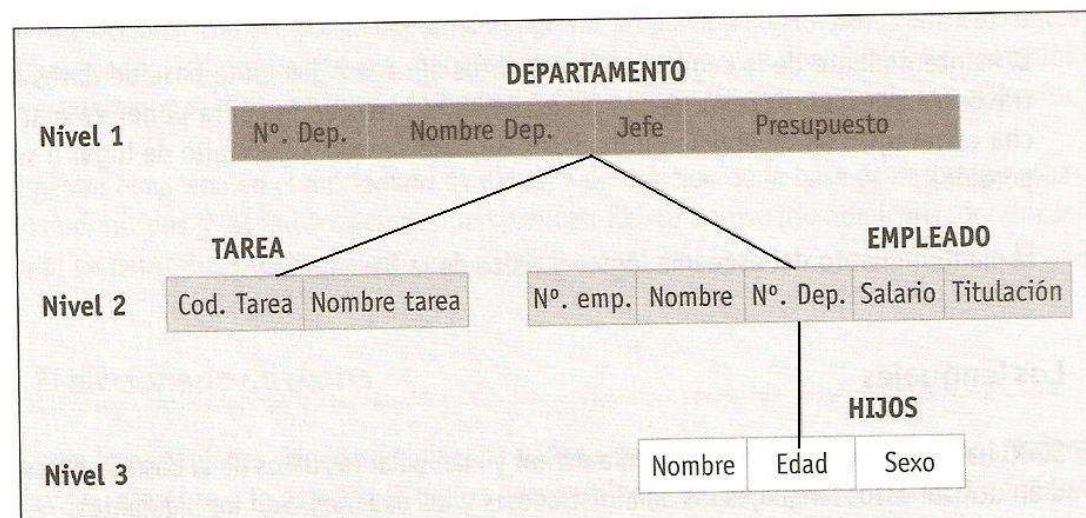
5.1 MODELO RELACIONAL

- En este modelo se representan los datos y las relaciones entre estos, a través de una colección de tablas
- Las filas (tuplas) equivalen a cada uno de los registros que contendrá la base de datos
- Las columnas corresponden a las características (atributos) de cada registro.



5.2 MODELO JERÁRQUICO

- Se sirve de árboles para la representación lógica de los datos
- Un árbol se compone de 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.
- Ejemplo:



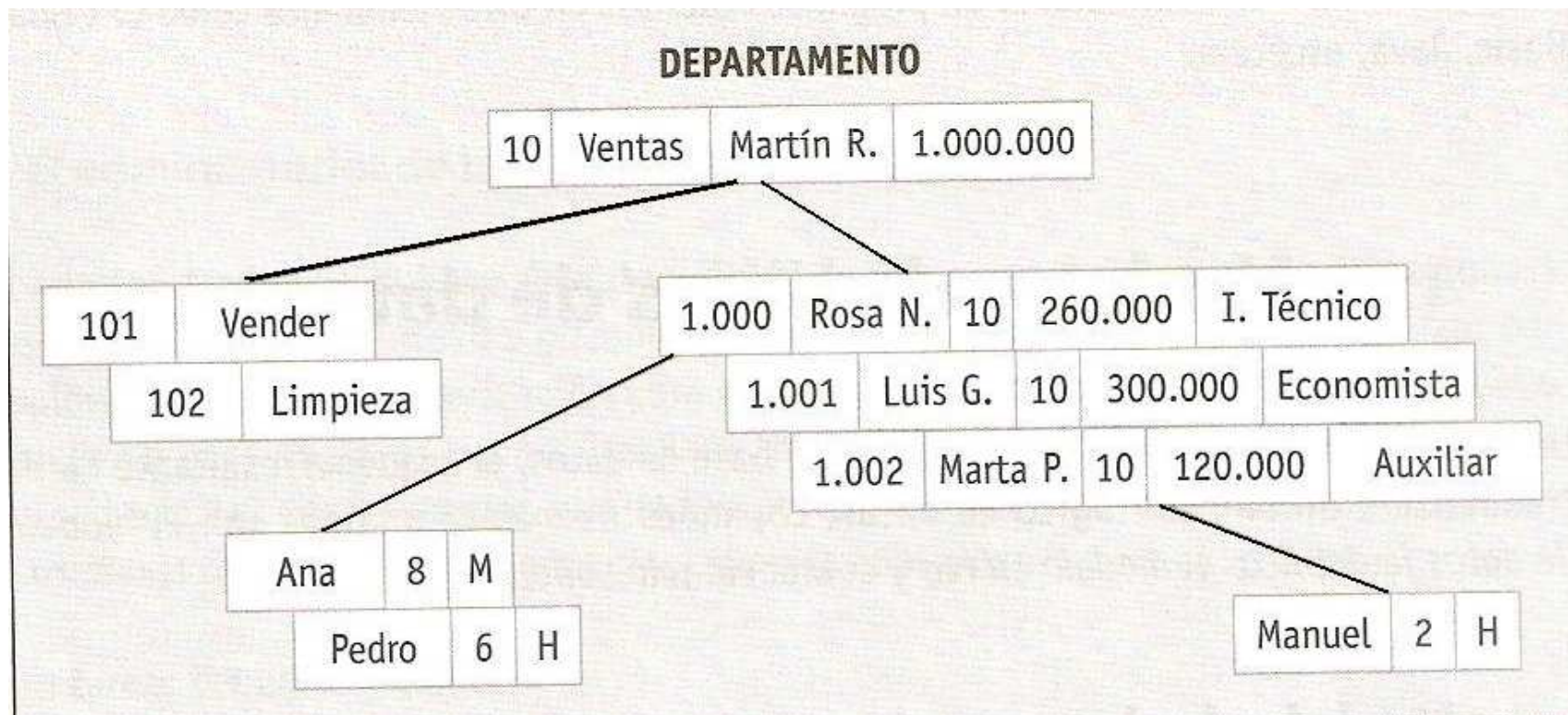
5.2 MODELO JERÁRQUICO

- Representa un esquema jerárquico con cuatro segmentos y tres niveles. El segmento raíz (primer nivel) almacena la información referente a los departamentos de una empresa; se llama DEPARTAMENTO.
- En el segundo nivel hay dos segmentos que dependen del segmento raíz. El primero llamado TAREA, contiene las tareas que se realizan en los departamentos. El segundo, denominado EMPLEADO, contiene los datos de los empleados.
- En el tercer nivel hay un segmento llamado HIJOS que depende del segmento EMPLEADO. Este contiene los datos de los hijos de los empleados.



5.2 MODELO JERÁRQUICO

- La representación de una ocurrencia concreta sería la siguiente:




5.3 MODELO EN RED

- Utiliza estructuras de datos en red, conocidas también como estructuras plex.
- Las entidades se representan como nodos y las relaciones como enlaces o punteros que unen los nodos.
- En una estructura en red cualquier componente puede vincularse con cualquier otro.
- Si la describimos en términos de padres e hijos, un hijo puede tener varios padres.



5.4 MODELOS DE DATOS ORIENTADOS A OBJETOS

- Estos modelos son utilizados, sobre todo, en aplicaciones programadas bajo el paradigma de la orientación a objetos.
- Una base de datos de este tipo está formada por objetos relacionados entre sí.
- Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos como son: 
 - **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 o tablas heredan atributos y métodos (comportamiento) de otros situados en un nivel superior según una jerarquía de clases.
 - **Polimorfismo:** hace referencia a métodos que pueden aplicarse a distintos tipos de objetos. 