

Tema 01 – Almacenamiento de la Información

Módulo de Base de Datos – 1º DAW 24/25

Objetivos

- ▶ Conocer características principales y tipos de ficheros.
- ▶ Entender el origen de las bases de datos como alternativa a los sistemas de ficheros.
- ▶ Describir los componentes y funciones principales de un Sistema Gestor de Bases de Datos (SGBD).
- ▶ Conocer las diferencias entre SGBD libres y comerciales.
- ▶ Introducir los conceptos relacionados con bases de datos distribuidas.

Almacenamiento de la Información

- ▶ Todas las aplicaciones informáticas trabajan en última instancia con datos o información que deben ser almacenados en un medio físico, como discos duros, memorias flash o DVD
- ▶ Jerarquía medios físicos en tres niveles de almacenamiento:
 - ▶ Primario
 - ▶ Secundario
 - ▶ Intermedio

Almacenamiento Primario

- ▶ Se refiere a aquellos medios sobre los que la CPU del ordenador puede acceder directamente y, por tanto, más rápidamente
- ▶ Son la memoria principal o memoria RAM y las memorias caché de primer y segundo nivel, más pequeñas pero más rápidas

Almacenamiento Secundario

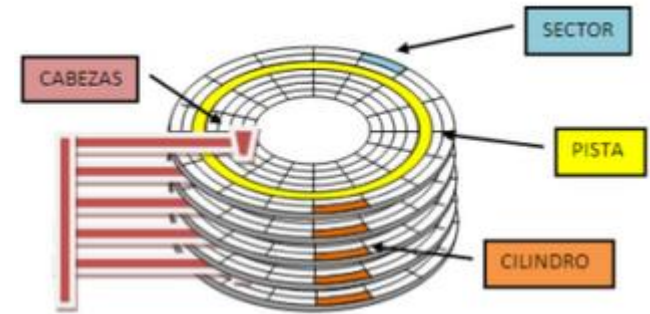
- ▶ Se refiere a dispositivos más lentos, pero de mayor capacidad
- ▶ Se distinguen:
 - ▶ Discos ópticos
 - ▶ Discos Magnéticos
 - ▶ Cintas
 - ▶ Memorias Flash
 - ▶ Discos SSD
- ▶ Para acceder a los datos la CPU debe copiarlos previamente en el almacenamiento primario

Almacenamiento Intermedio

- ▶ Cuando se necesita transferir varios bloques de disco a memoria principal y se conocen todas las direcciones de bloque es posible reservar varias áreas de almacenamiento intermedio o buffers dentro de la memoria principal para agilizar la transferencia.
- ▶ De este modo, mientras la CPU procesa datos de un buffer puede leer o escribir en otro.
- ▶ El uso de este tipo de almacenamiento es muy común en sistemas de bases de datos.

Estructura y Funcionamiento Disco Duro

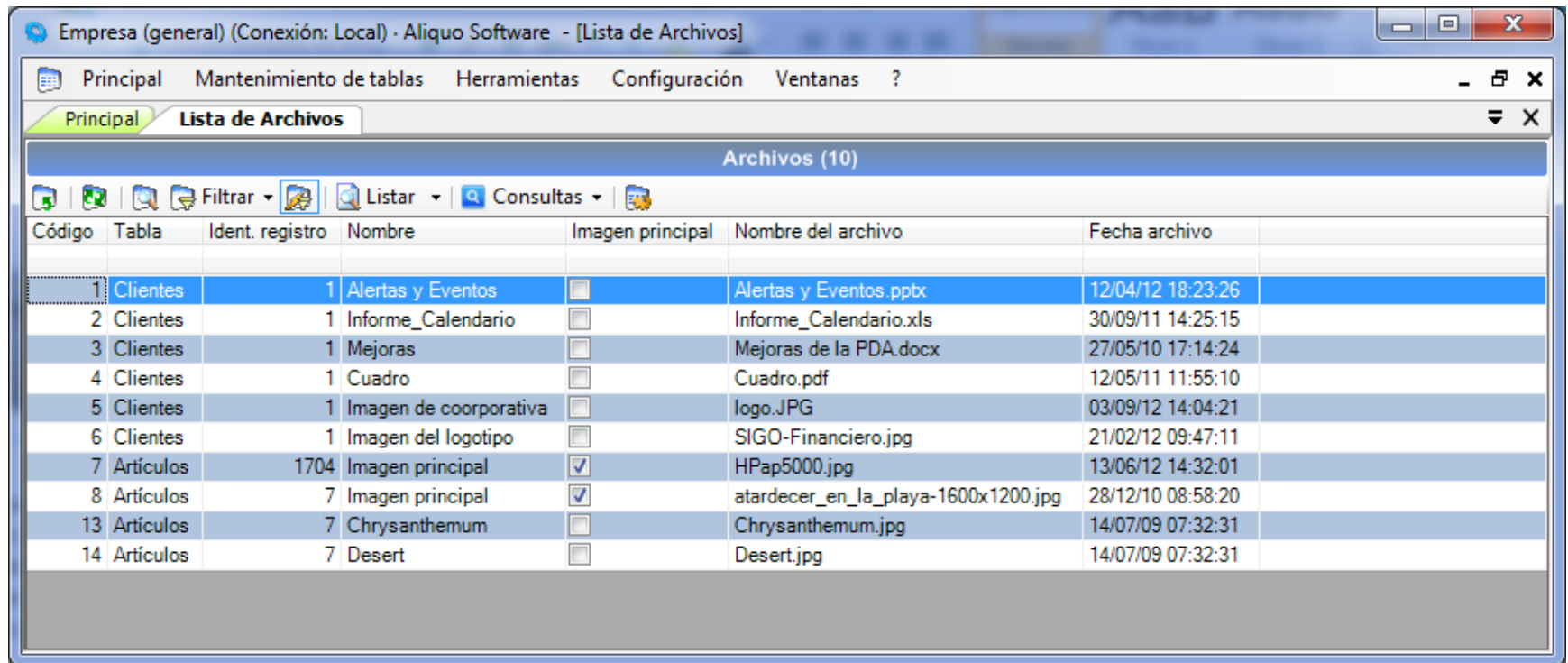
- ▶ En un disco duro normalmente se agrupan varios platos, cada uno de los cuales se divide en **pistas** o círculos concéntricos
- ▶ La agrupación de pistas de todos los discos se denomina **cilindro**
- ▶ Es importante que los datos a los que se suelen acceder simultáneamente estén en el mismo cilindro ya que se leen con mayor rapidez.
- ▶ Cada pista, al contener gran cantidad de información, se subdivide en bloques o **sectores** de un tamaño fijo. Los bloques, también llamados páginas, suelen tener un tamaño entre 512 y 4.096 bytes
- ▶ La transferencia de información entre memoria y disco tiene lugar en unidades de bloque
 - ▶ Cuando produce una orden de lectura se copia uno o varios bloques en el llamado buffer de la memoria
 - ▶ una escritura se copia el bloque correspondiente al bloque del disco



Almacenamiento Intermedio

- ▶ Cuando se necesita transferir varios bloques de disco a memoria principal y se conocen todas las direcciones de bloque es posible reservar varias áreas de almacenamiento intermedio o buffers dentro de la memoria principal para agilizar la transferencia
- ▶ De este modo, mientras la CPU procesa datos de un buffer puede leer o escribir en otro.
- ▶ El uso de este tipo de almacenamiento es muy común en sistemas de bases de datos

Sistemas de Archivos



Empresa (general) (Conexión: Local) · Aliquo Software - [Lista de Archivos]

Principal Mantenimiento de tablas Herramientas Configuración Ventanas ?

Principal **Lista de Archivos**

Archivos (10)

Filtrar Listar Consultas

Código	Tabla	Ident. registro	Nombre	Imagen principal	Nombre del archivo	Fecha archivo
1	Cientes	1	Alertas y Eventos	<input type="checkbox"/>	Alertas y Eventos.pptx	12/04/12 18:23:26
2	Cientes	1	Informe_Calendario	<input type="checkbox"/>	Informe_Calendario.xls	30/09/11 14:25:15
3	Cientes	1	Mejoras	<input type="checkbox"/>	Mejoras de la PDA.docx	27/05/10 17:14:24
4	Cientes	1	Cuadro	<input type="checkbox"/>	Cuadro.pdf	12/05/11 11:55:10
5	Cientes	1	Imagen de corporativa	<input type="checkbox"/>	logo.JPG	03/09/12 14:04:21
6	Cientes	1	Imagen del logotipo	<input type="checkbox"/>	SIGO-Financiero.jpg	21/02/12 09:47:11
7	Artículos	1704	Imagen principal	<input checked="" type="checkbox"/>	HPap5000.jpg	13/06/12 14:32:01
8	Artículos	7	Imagen principal	<input checked="" type="checkbox"/>	atardecer_en_la_playa-1600x1200.jpg	28/12/10 08:58:20
13	Artículos	7	Chrysanthemum	<input type="checkbox"/>	Chrysanthemum.jpg	14/07/09 07:32:31
14	Artículos	7	Desert	<input type="checkbox"/>	Desert.jpg	14/07/09 07:32:31

Sistema Archivos - Registros

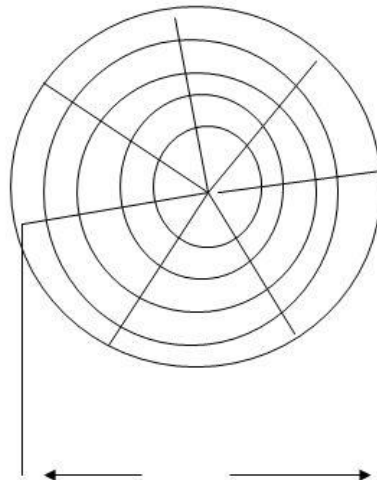
- ▶ La información se almacena en forma de registros, que son colecciones de valores o elementos de información relacionados, cada uno de los cuales corresponde a un **campo** del registro
- ▶ Cada **campo** tiene un tipo de dato que especifica el **tipo** de **valores** que puede tomar: Varchar, Char, INT, Float, TEXT,
- ▶ Cada campo tiene un **tamaño** determinado en bytes que puede ser fijo o variable según los requisitos de la aplicación

Sistema Archivos - Archivos

- ▶ Podemos definir un fichero informático como un conjunto de registros, grabados sobre un soporte que pueda ser leído por el ordenador
- ▶ Estos registros suelen tener longitud fija
- ▶ Cada campo tendrá un valor para cada registro, a su vez los campos podrán tener un tamaño variable.
- ▶ Los archivos de registros de longitud fija son más fáciles de manipular por parte de los programas, sin embargo desperdician espacio en disco
- ▶ 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

Sistema Archivos

REGISTRO FÍSICO Y LÓGICO



Bloque o registro físico:
2.603 bytes

Registro lógico = 507 bytes por registro

DNI cliente	Nombre	Apellidos	Domicilio	Localidad
31789634d	José	Luque Mera	San Francisco nº3	Cádiz

Factor de bloqueo: nº de registros lógicos
que caben en un registro físico

Tamaño registro físico

Tamaño registro
lógico

$$\frac{2.603}{507} = 5,13 \text{ (5 registros lógicos por bloque)}$$

¿Cuánto espacio de almacenamiento necesitaríamos para grabar
17.300 clientes?

Archivos

- ▶ Normalmente los registros de un archivo se asignan a uno o varios bloques en el sistema de almacenamiento para su manipulación por parte de los programas.
- ▶ El método de asignación elegido y el sistema de almacenamiento de los registros determinarán la eficiencia de este proceso.
- ▶ Los ficheros son importantes porque son la unidad básica de información utilizada por cualquier programa, incluidos los sistemas gestores de bases de datos.
- ▶ Todos los datos son, en última instancia gestionados mediante ficheros mediante cuatro operaciones básicas: consulta o lectura, inserción, modificación y borrado. Cualquier operación más compleja (como búsqueda, ordenación, etc.) es combinación de dos o más operaciones básicas.

Organización Primaria de Archivos

- ▶ Se aplica a la forma en que se colocan los datos contenidos en los registros de cada fichero sobre el soporte informático
- ▶ Existen dos formas básicas de organización:
 - ▶ Secuencial.
 - ▶ En la organización secuencial los registros se van grabando unos a continuación de los otros, en el orden que se van dando de alta
 - ▶ Relativa.
 - ▶ La organización relativa los registros se graban en las posiciones que les corresponda según el valor que guarden en el campo denominado clave, que permite identificar el registro dentro del fichero

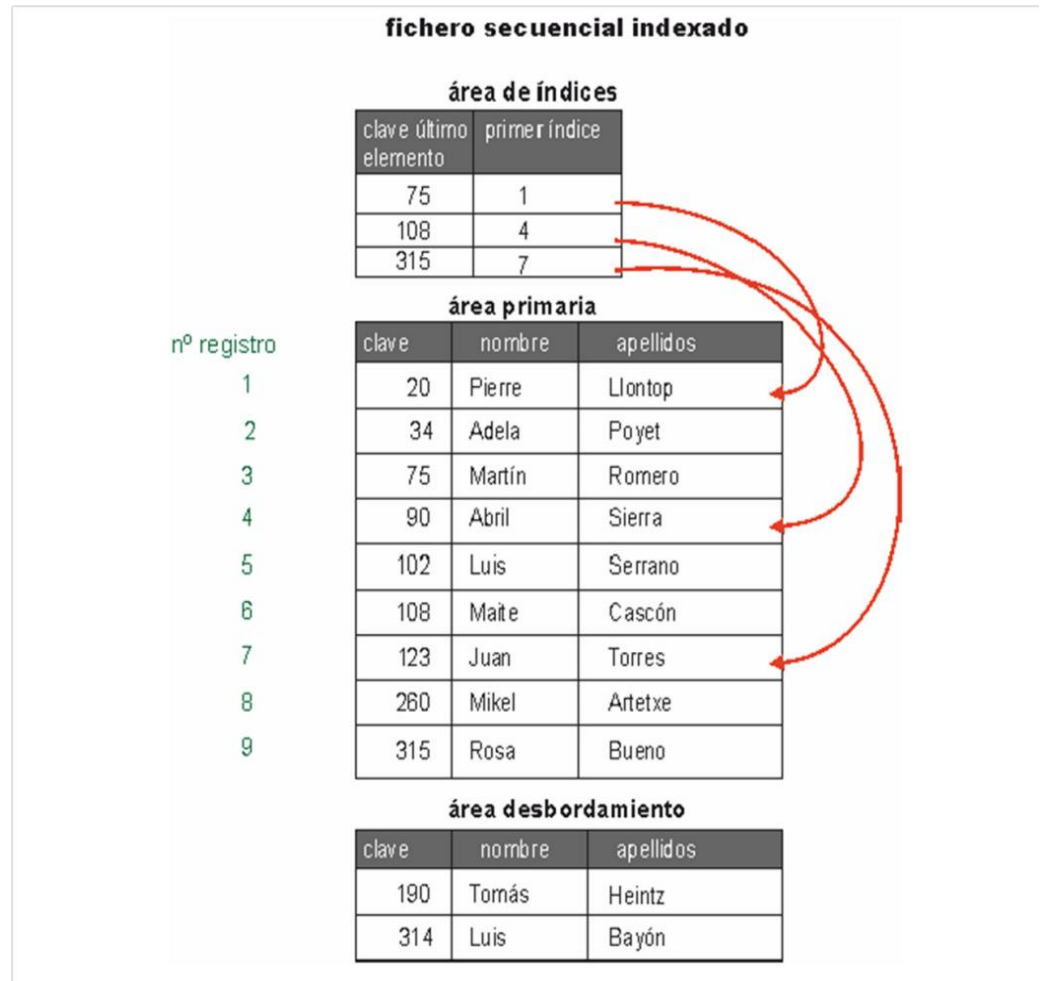
Organización Secuencial

- ▶ Los registros se colocan secuencialmente uno a continuación del otro
- ▶ Los registros nuevos se añaden al final del fichero
- ▶ Operaciones:
 - ▶ **Inserción:**
 - ▶ Más eficiente siempre en el último bloque disponible
 - ▶ **Búsqueda:**
 - ▶ Complicada de tipo lineal bloque a bloque hasta llegar al encontrado
 - ▶ **Borrado:**
 - ▶ Ineficiente, deja huecos. Hay técnicas de reorganización de ficheros
 - ▶ **Acceso:**
 - ▶ Para el caso de archivos de registros de longitud fija el acceso a un registro por su posición dentro del archivo es muy sencillo ya que de este modo el i -ésimo registro se encontrará en el bloque resultado de obtener la parte entera de i/fbl
 - ▶ Dentro de ese bloque, será el registro número $i \bmod fbl$



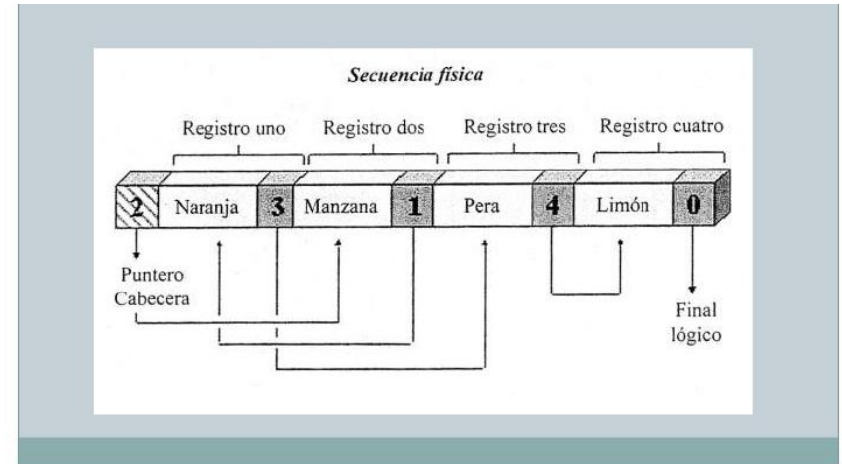
Organización Secuencial Indexada

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



Organización Secuencial Encadenada

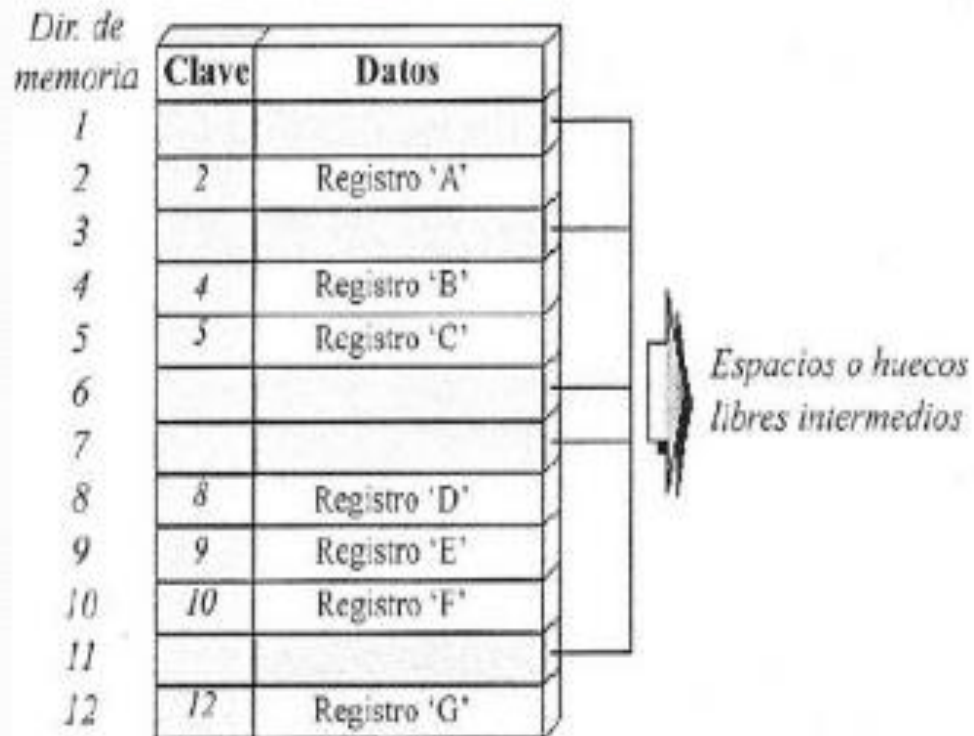
- Permite tener los registros ordenados según un orden lógico diferente del orden físico en el que están grabados gracias a la utilización de unos campos adicionales llamados punteros



Organización Relativa

- ▶ En este tipo de archivos los registros se graban en orden según el valor de uno de sus campos llamado campo de ordenación.
- ▶ Normalmente se usa un campo especial denominado campo clave, cuyos valores son distintos para cada registro.
- ▶ En estos archivos la **lectura** es muy eficiente cuando se hace en orden según el campo de ordenación ya que el siguiente registro se encontrará a continuación del actual en el mismo bloque, o en el siguiente, si es el último.
- ▶ Las **búsquedas** son muy rápidas, siempre que la condición de búsqueda incluya el campo de ordenación ya que en tal caso puede usarse la técnica de búsqueda binaria.
- ▶ Este sistema **no ofrece ventajas** cuando se trata de acceder a registros de manera aleatoria o de manera ordenada según un campo distinto al de ordenación, en cuyo caso deberemos usar un archivo adicional para ir almacenando los registros ordenados
- ▶ La **inserción** también es costosa, ya que debe mantenerse el orden, lo que puede implicar desplazamiento de registros para insertar uno nuevo en el orden apropiado
- ▶ la **eliminación** es menos costosa si se usan registros marcados y se reorganiza el archivo periódicamente
- ▶ 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.

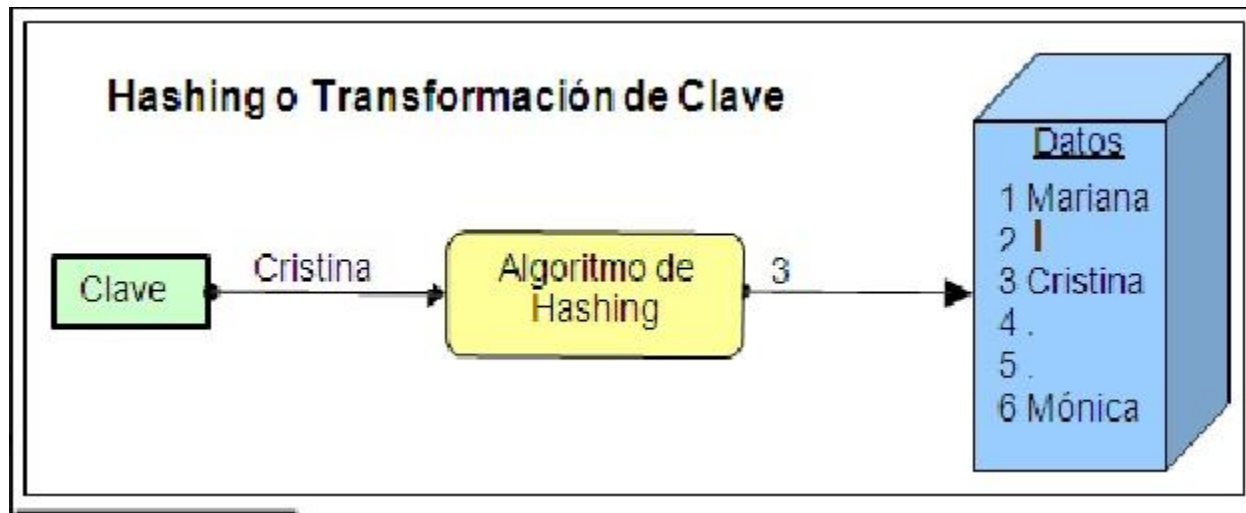
Organización Relativa



Dispersión o hashing

- ▶ En este sistema se elige un campo llamado campo de **dispersión**. Al valor de ese campo se le aplica una función llamada función de aleatorización o de **dispersión** que, tomando como entrada dicho valor, devuelve un número que será la dirección del bloque de disco en que se almacenará el registro.
- ▶ Las técnicas de dispersión o hashing se utilizan para acelerar el acceso a los registros cuando se busca un único registro según el llamado campo de dispersión. Normalmente este campo suele ser el campo clave.
- ▶ Existen numerosas funciones de dispersión y su eficacia radica en que distribuyan los registros lo más posible minimizando el número de colisiones.

Algoritmo de hashing



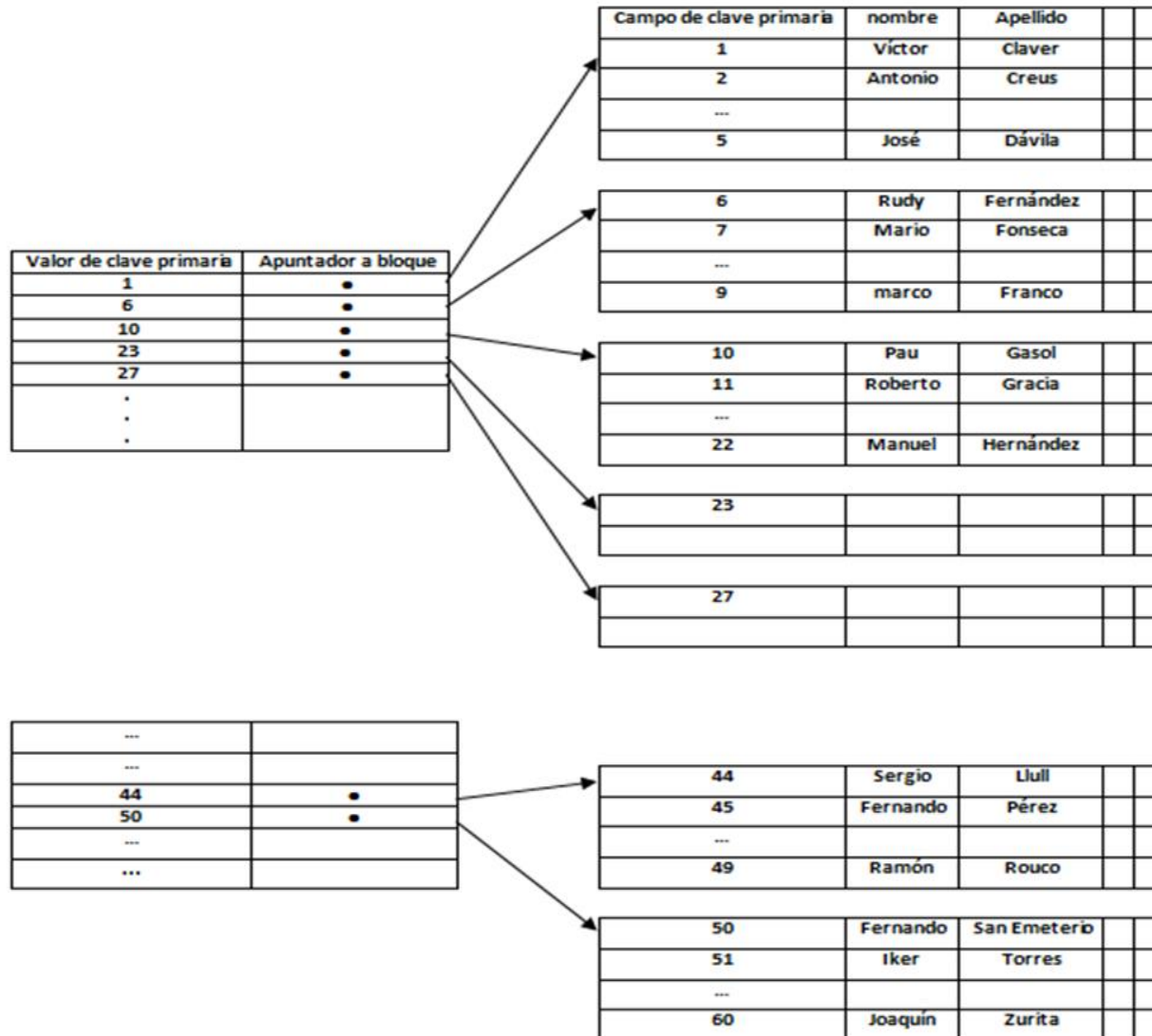
1.2.4 Métodos de Acceso

- ▶ Procedimiento seguido para acceder a uno o más registros del fichero
- ▶ Mejorar la eficiencia se basan en el uso Ficheros Índices. Similar a los índices analíticos de cualquier libro de texto
- ▶ Características Fichero Índice:
 - ▶ Contiene sólo dos campos:
 - ▶ Campo de Indización: depende del tipo de índice
 - ▶ Dirección: Dirección del bloque donde se encuentra dicho registro
 - ▶ Fichero Índice está ordenado por el campo de Indización. Permitiendo búsquedas binarias.
- ▶ Tipos de Índices:
 - ▶ Primarios: campo de indización es el campo clave no permite valores repetidos
 - ▶ Agrupamiento: campo de indización no es el campo clave y existen valores repetidos
 - ▶ Secundario: campo de indización no es el campo clave y no permite valores repetidos.

1.2.4.1 Índices Primarios

- ▶ Relaciona el valor de la clave con la dirección que ocupa dicho registro en el fichero.
- ▶ Cada entrada del registro índice:
 - ▶ Valor de la Clave
 - ▶ Un apuntador al bloque que lo contiene
- ▶ Cada entrada apunta a un bloque de registros
- ▶ Tipos:
 - ▶ NO DENSOS. Cuando en el índice no están todos los valores del campo clave. Cada entrada se asocia con varios registros.
 - ▶ DENSOS. Cuando en el índice están todos los valores del campo clave. Cada entrada se asocia con un solo registro.

Indices Primarios



1.2.4.2 Índices de Agrupamiento

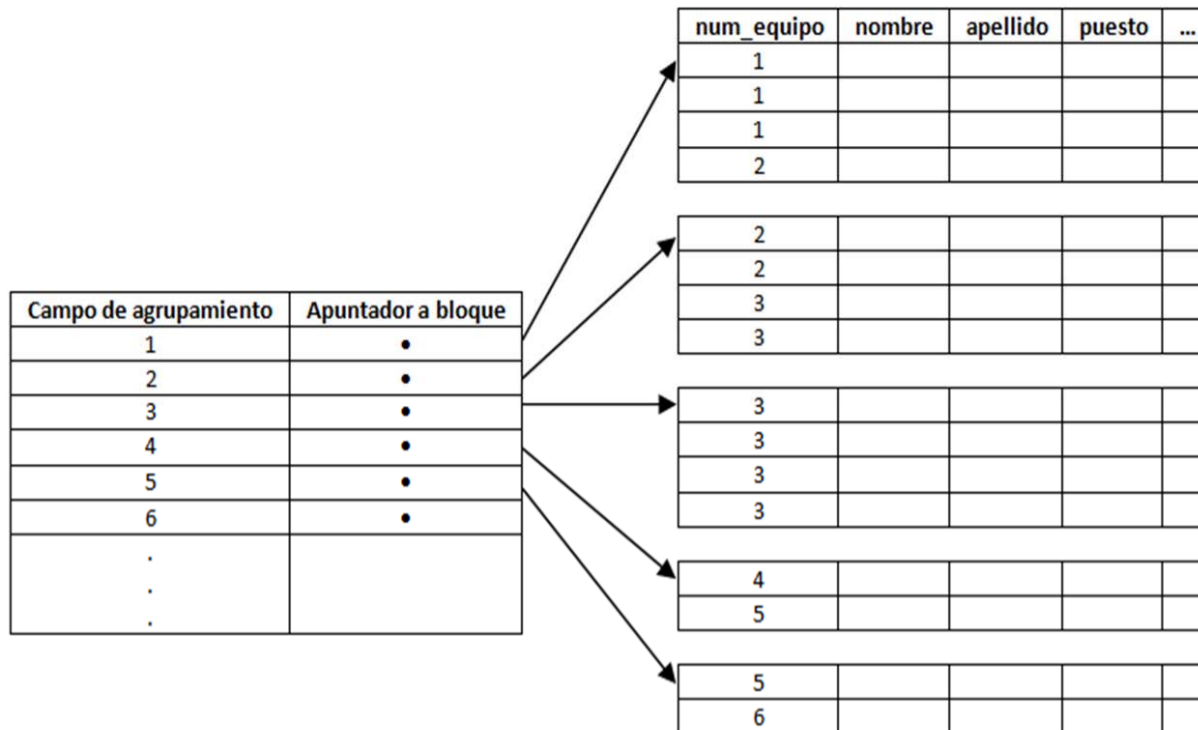
- ▶ **Campo de Agrupamiento.**

- ▶ Los registros del fichero están ordenados a partir de un campo no clave donde los valores pueden repetirse a ese campo se le llama campo de agrupamiento.
- ▶ Imaginemos que queremos agrupar los empleados por la categoría a la que pertenecen.

- ▶ **Fichero índice:**

- ▶ Campo de agrupamiento
- ▶ Apuntador al bloque de agrupamiento
- ▶ Hay una entrada distinta por cada valor de agrupamiento y contiene un apuntador al primer bloque con ese valor.

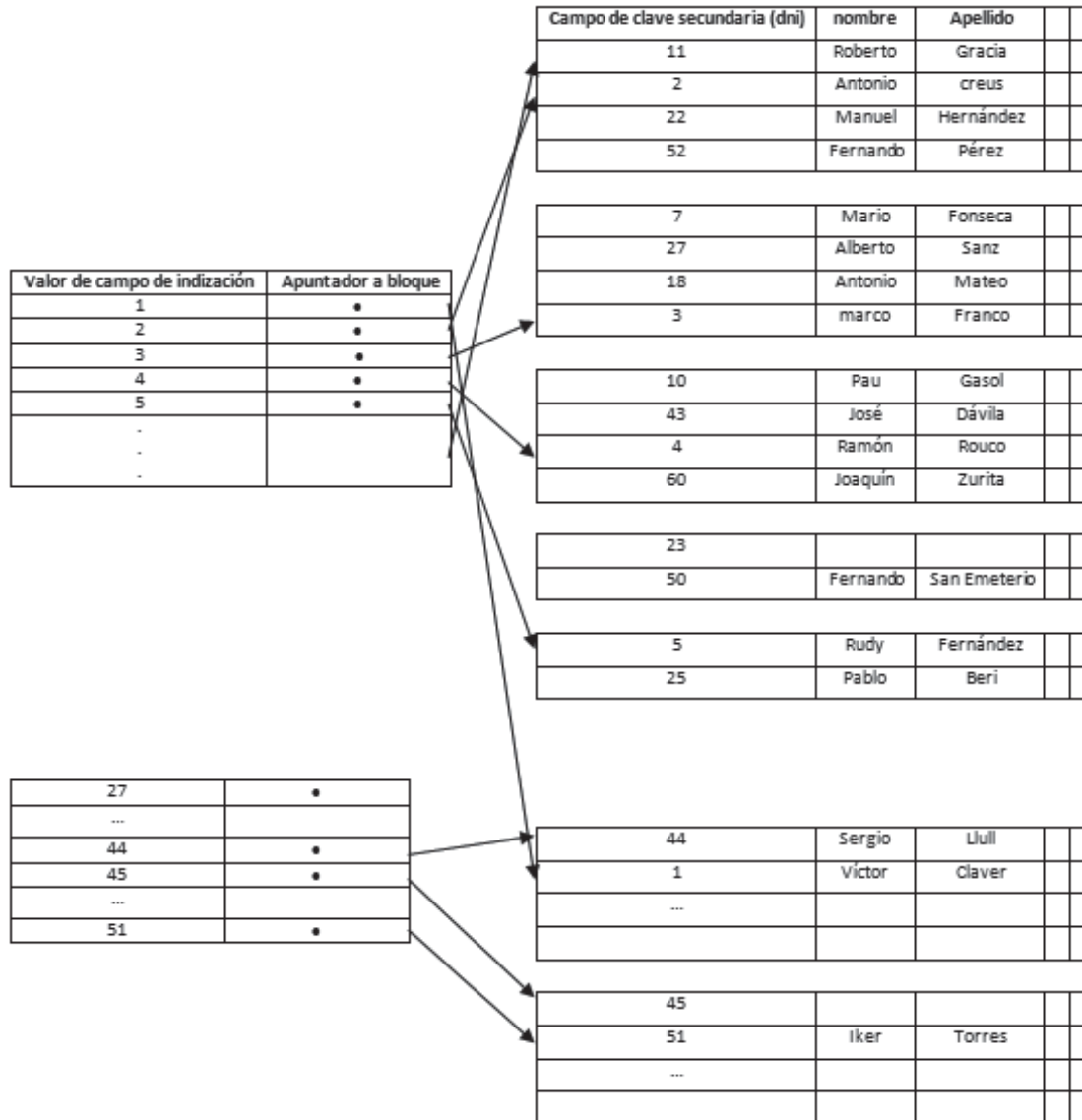
Indices de Agrupamiento



1.2.4.3 Índices Secundarios

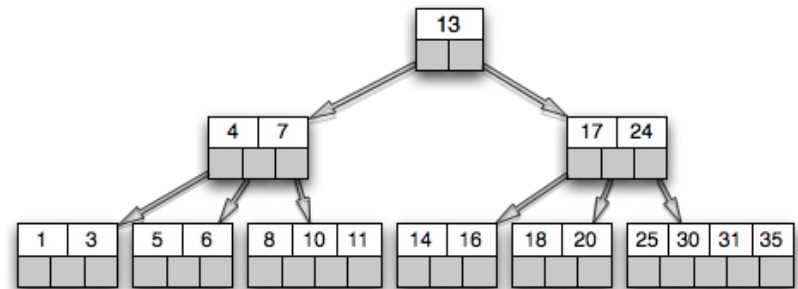
- ▶ El campo de indización es distinto al campo clave aunque sus valores no se pueden repetir.
- ▶ El campo de dirección es un apuntador al bloque
- ▶ Puede haber varios índices secundarios sobre un mismo archivo
- ▶ El archivo se encuentra ordenado por el campo clave y no por el campo de indización
- ▶ Ejemplo clave secundaria DNI de Clientes

Indices Secundarios



1.2.4.4 Otros tipos índices

- ▶ **Multinivel:**
 - ▶ Se crean nuevos archivos índices sobre el índice inicial, creando así un nuevo nivel. Todo ello se puede extender a más niveles creando índices multinivel, mejorando así la rapidez de acceso.
- ▶ **Árboles B y B+**
 - ▶ Se utilizan estructuras tipo árbol para los archivos índices. Mejorando la rapidez de acceso a los datos para grandes archivos
- ▶ **Índices Hash:**
 - ▶ El índice está formado por el valor de dispersión y dirección al bloque
- ▶ **Índices Lógicos:**
 - ▶ Se suelen usar cuando los registros físicos de datos, cambian con frecuencia de posición
 - ▶ Cada entrada está formada por un valor de indización secundario y el campo clave.



1.3 Sistemas Base de Datos

- ▶ **Inicialmente se usaban Sistemas de Ficheros:**
 - ▶ Programas manejaban la información almacenada en ficheros
 - ▶ Cada equipo y programa trabajaba con sus propios datos, no se compartían
- ▶ **Funcionó sólo al principio, pero surgen problemas debido a:**
 - ▶ Incremento de información
 - ▶ Incremento de usuarios
 - ▶ Falta de integridad
 - ▶ Duplicación de datos
- ▶ **Como solución se aporta el Sistema de Base de Datos:**
 - ▶ Organiza la información de forma ordenada y manejable
 - ▶ Centralización de la gestión
 - ▶ Organización de los datos en forma de Base de Datos

Problemas Sistemas Ficheros

- ▶ **Separación y aislamiento de datos**
 - ▶ Incrementa la dificultad de acceso
- ▶ **Duplicación de los datos**
 - ▶ Provoca redundancia e inconsistencia
- ▶ **Dependencia de datos**
 - ▶ Falta de independencia lógica-física.
- ▶ **Formatos de ficheros incompatibles**
 - ▶ Cada lenguaje crea sus propios ficheros
- ▶ **Consultas ligadas a la programación**
 - ▶ Consultas e informes realizadas mediante programación
- ▶ **Control de concurrencia inexistente**
 - ▶ Acceso de varios clientes al mismo fichero genera inconsistencia
- ▶ **Autorización**
 - ▶ Falta de control en el acceso
- ▶ **Catálogo**
 - ▶ Dificultad de disponer de un esquema general de la base de datos

Bases de Datos

► Definición

- *Una base de datos es un conjunto de datos almacenados entre los que existen relaciones lógicas y ha sido diseñada para satisfacer los requerimientos de información de una empresa u organización*
- La base de datos es un conjunto de datos organizados en estructuras que se definen una sola vez y que se utilizan al mismo tiempo por muchos equipos y usuarios
- los datos en una base de datos están centralizados y organizados
- Mínima redundancia y se fácil gestión
- Almacena también una descripción de los datos METADATOS que son los Catálogos o diccionario de datos.

1.3.1 Arquitectura Sistemas Base de Datos

- ▶ En 1975, el comité ANSI-SPARC (American National Standard Institute - Standards Planning and Requirements Committee) propuso un estándar para la creación de sistemas de bases de datos.
- ▶ Basado en una arquitectura de tres niveles:
 - ▶ Nivel Interno
 - ▶ Nivel Global
 - ▶ Nivel Externo
- ▶ Nivel Interno:
 - ▶ Define la estructura física de la Base de Datos. Métodos acceso, directorios, discos, ficheros.
- ▶ Nivel Global:
 - ▶ Describe la estructura de la Base de Datos mediante Esquema Conceptual: Tablas, Entidades, Atributos, Relaciones y Restricciones
- ▶ Nivel Externo:
 - ▶ Describen los esquemas externos o vistas usuario.

Independencia de datos

- ▶ La arquitectura a tres niveles permite el concepto de **Independencia de Datos**
- ▶ Capacidad de realizar modificaciones en uno de los niveles, sin tener que modificar el nivel inferior o superior.
- ▶ Tipos de Independencia de Datos:
 - ▶ Independencia Lógica:
 - ▶ Capacidad de modificar el esquema conceptual sin modificar los esquemas externos, vistas o programas de aplicación.
 - ▶ Ej. Modificación esquema de la Base de Datos
 - ▶ Independencia Física
 - ▶ Capacidad de modificar el esquema interno sin modificar el esquema conceptual
 - ▶ Ej: Reorganización de los ficheros para mejorar acceso

ARQUITECTURA DE BASES DE DATOS

Nivel externo o vista de usuario

Independencia lógica

Nivel lógico o global

Independencia física

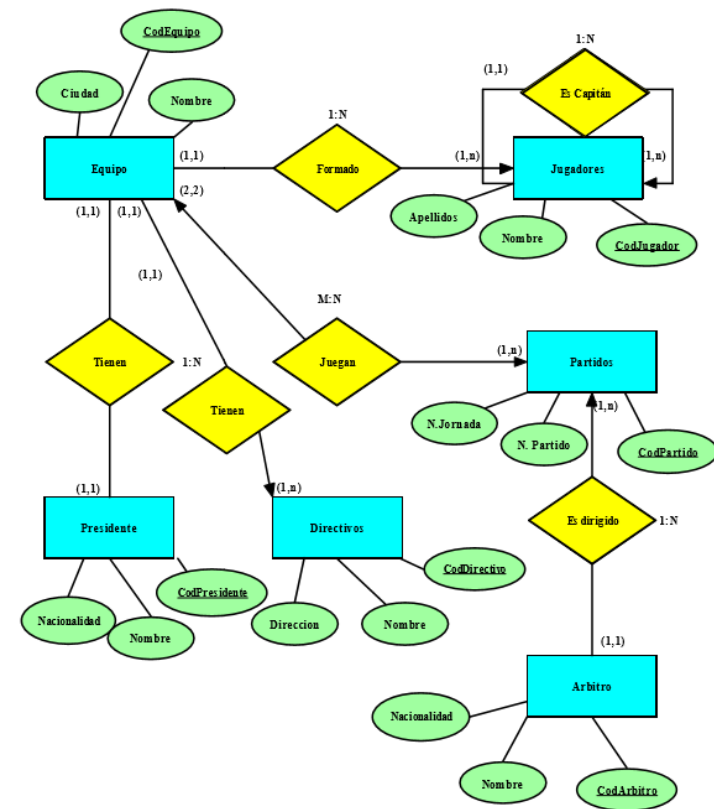
Nivel interno o físico

1.3.2 Modelo 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*
- ▶ **Tipos de Modelos:**
 - ▶ Modelos conceptuales.
 - ▶ MER (Modelo Entidad Relación)
 - ▶ Modelos lógicos tradicionales.
 - ▶ Modelo Relacional
 - ▶ Modelo Red
 - ▶ Modelo Jerárquico
 - ▶ Modelos lógicos avanzados
 - ▶ Orientados a Objetos
 - ▶ Modelos Datos Declarativos

Modelos Conceptuales (MER)

- ▶ Se usan para describir los datos de forma a nivel global mediante una abstracción de la vida real.
- ▶ Existen diferentes modelos de este tipo, el más utilizado es el Modelo Entidad-Relación (MER):
 - ▶ Sencillez
 - ▶ Eficiencia
- ▶ Objetos MER:
 - ▶ Entidades
 - ▶ Atributos
 - ▶ Relaciones
 - ▶ Restricciones

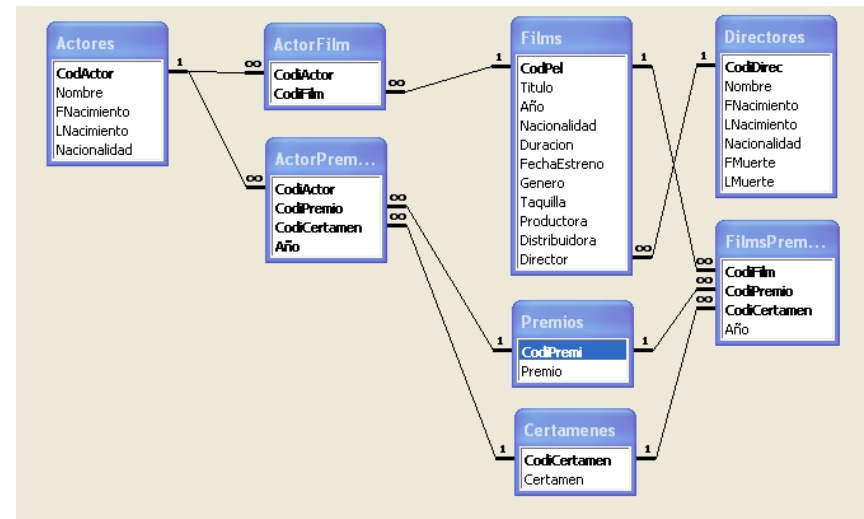


Modelos Lógicos Tradiciones

- ▶ Se usan describir la estructura lógica global de la Base de Datos con una descripción más estructurada y cercana a la implementación (SQL)
- ▶ Los tres modelos de datos más ampliamente aceptados:
 - ▶ Modelo Relacional
 - ▶ Modelo Red
 - ▶ Modelo Jerárquico

Modelo Relacional

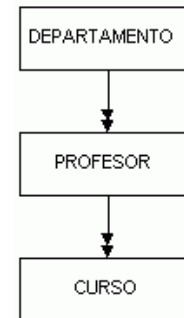
- ▶ Representa los datos y las relaciones mediante una colección de tablas (*tuplas y atributos*)
- ▶ Permite representar también el nivel externo o vistas.
- ▶ Modelo más usado en la actualidad desde 1970 por Edgar Frank Codd.



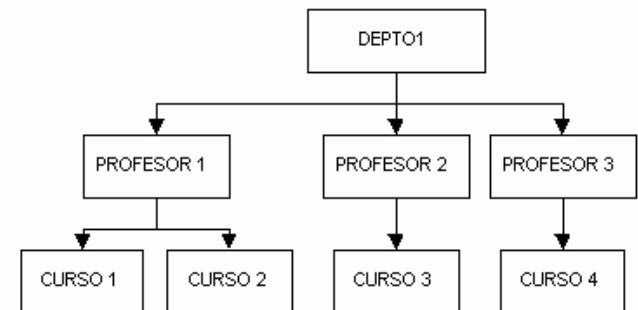
Modelo Jerárquico

- ▶ Utiliza una estructura árbol (al revés) para la representación de los datos.
- ▶ Útiles en caso de aplicaciones que manejan gran volumen de información y datos compartidos

Estructura lógica

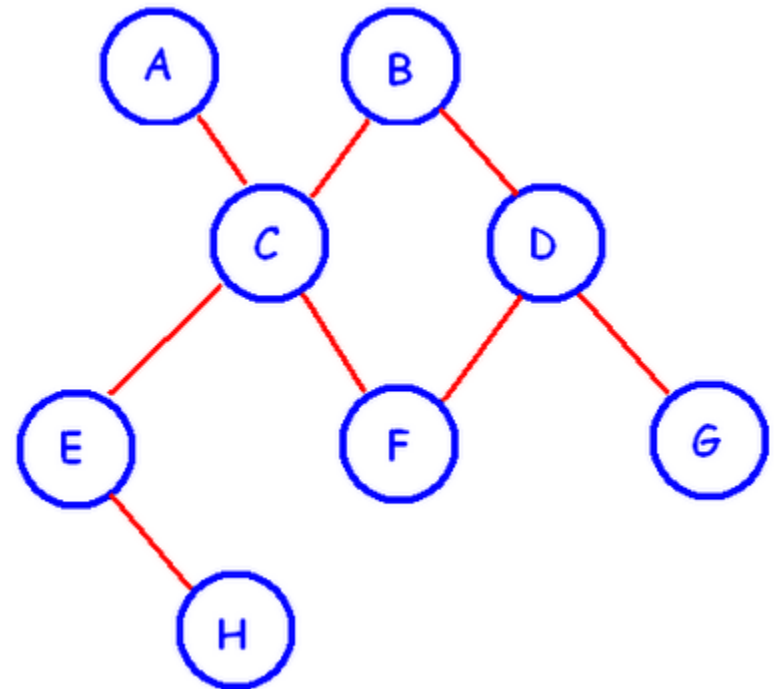


Ejemplo de base de datos



Modelo Red

- ▶ Representa los datos mediante una estructura tipo Red. Donde un nodo puede tener varios nodos hijos y también varios nodos padres.
- ▶ Ofrecía una mejora respecto a la redundancia de datos pero gran dificultad de programación

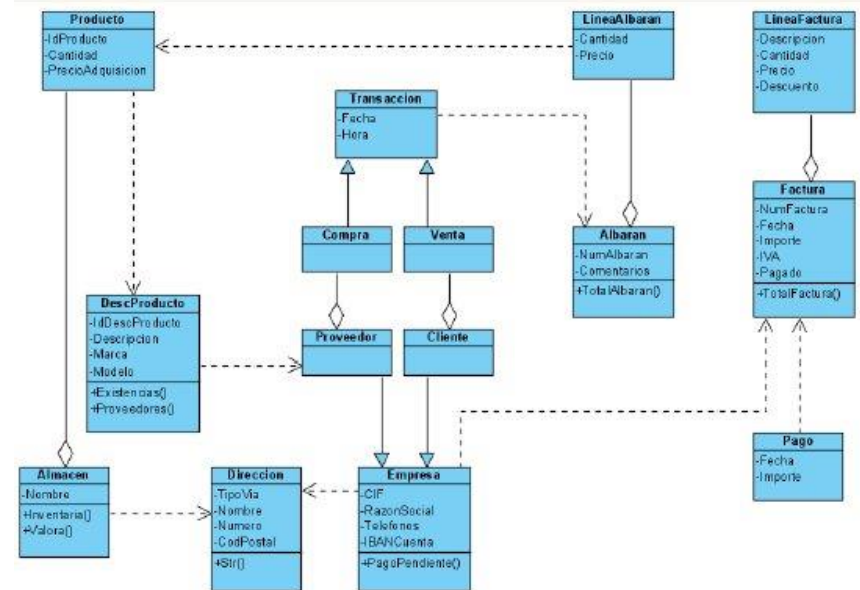


Modelos Lógicos Avanzados

- ▶ Son modelos de datos relativamente recientes y cada vez más utilizados, sobre todo en aplicaciones específicas que manejan nuevos y más complejos tipos de datos.
- ▶ Tipos:
 - ▶ Modelos de Datos Orientados a Objetos
 - ▶ Modelos Datos Declarativos

Modelo Datos Orientada a Objetos

- ▶ Aplicaciones programadas bajo paradigma orientada objetos
- ▶ Almacena Base Datos:
 - ▶ Datos (Estado)
 - ▶ Funcionalidad Asociada a los datos (comportamiento)
- ▶ Almacena la Base de Datos mediante objetos relacionados entre sí, siendo un objeto entidades con estado y comportamiento.
- ▶ Conceptos relacionados:
 - ▶ Encapsulación: propiedad que permite ocultar información al resto de objetos.
 - ▶ Herencia: los objetos y tablas heredan atributos y métodos de otros objetos de nivel superior.
 - ▶ Polimorfismo: métodos que pueden aplicarse a distintos tipos objetos
- ▶ Este tipo de modelado de datos permite la independencia entre programas y operaciones.



Modelo de Datos Declarativos

- ▶ Existen dos tipos:
 - ▶ Deductivos
 - ▶ Funcionales
- ▶ Suelen usarse para bases de conocimiento, que no son más que bases de datos con mecanismos de consulta en los que el trabajo de extracción de información a partir de los datos recae en realidad sobre el sistema informático.
- ▶ Estos mecanismos de consulta exigen que la información esté distribuida de manera que haga eficiente las búsquedas de los datos, ya que normalmente las consultas de este tipo requieren acceder una y otra vez a los datos en busca de patrones que se adecúen a las características de los datos que ha solicitado el usuario

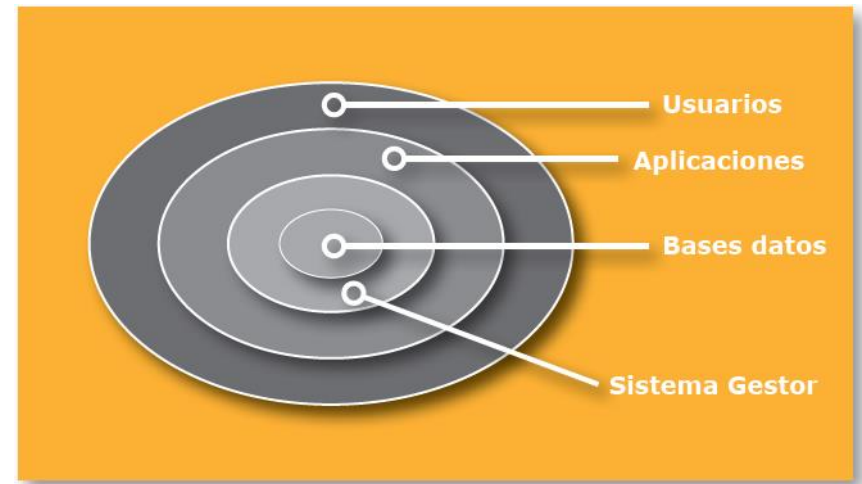
1.4 SISTEMAS GESTORES BASES DE DATOS

► Definición:

- SGBD)es una aplicación que permite a los usuarios definir, crear y mantener la base de datos y proporciona acceso controlado a la misma. Es una herramienta que sirve de interfaz entre el usuario y las bases de datos

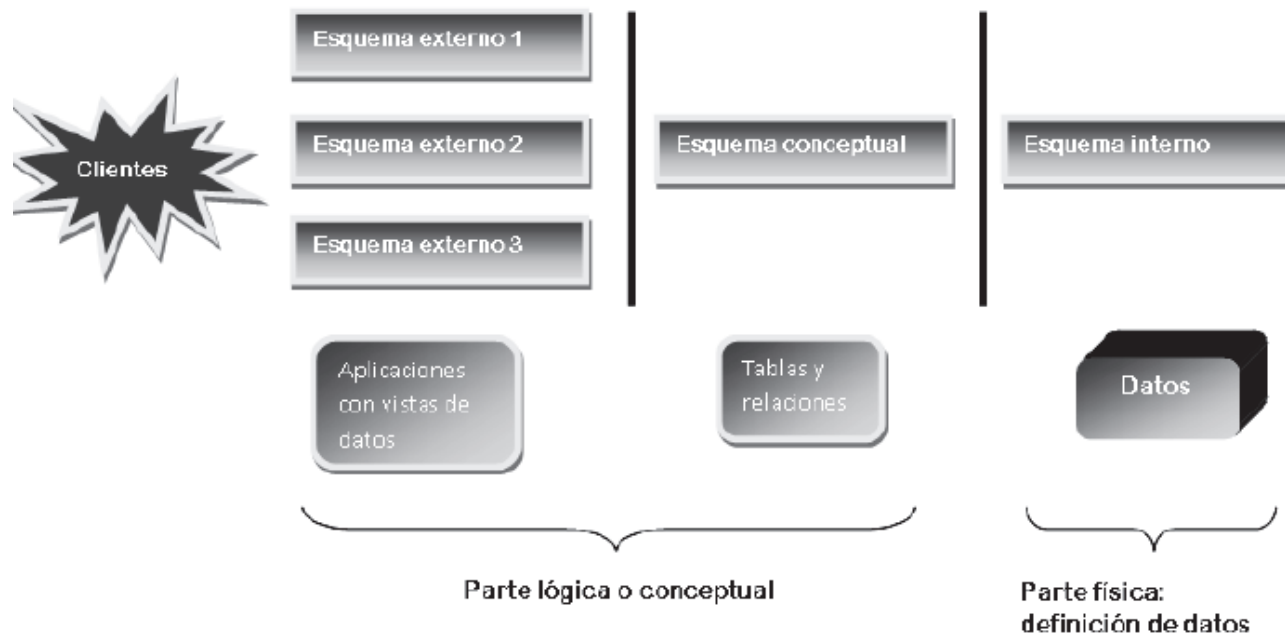
► Esquema Funcionamiento:

- usuarios acceden a la información usando aplicaciones (por ejemplo, un formulario web) que a su vez se comunican con sistemas gestores que son los que en última instancia acceden a los datos almacenados en las bases de datos mediante la interacción con el sistema operativo



Arquitectura

- 1975, el comité ANSI-SPARC (American National Standard Institute – Standards Planning and Requirements Committee) estableció los tres niveles de abstracción (externo, lógico y físico) como requisito en los sistemas gestores tal como queda reflejado en la siguiente figura.
- Se observa como en el nivel externo se encuentran los usuarios finales que tienen acceso a distintos esquemas los cuales a su vez se derivan del esquema o representación lógica que a su vez se traduce a un esquema físico o forma de almacenamiento de los datos.



1.4.1 Objetivos

- ▶ Los objetivos de un SGBD se pueden resumir:
 - ▶ Asegurar los tres niveles de abstracción: físico, lógico y externo.
 - ▶ Permitir la independencia física y lógica de los datos.
 - ▶ Garantizar la consistencia de los datos. Ya que puede haber datos duplicados o derivados que deben mantener sus valores de forma coherente.
 - ▶ Ofrecer seguridad de acceso a los datos por parte de usuarios y grupos.
 - ▶ Gestión de transacciones de forma que se garantice la ejecución de un conjunto de operaciones críticas como una sola operación.
 - ▶ Permitir la concurrencia de usuarios sobre los mismos datos mediante bloqueos que mantienen la integridad de los mismos.

1.4.2 Funciones SGBD

- ▶ Para la consecución de los objetivos comentados en la sección anterior la mayoría de SGBD comerciales y libres incorporan las siguientes características y funciones:
 - ▶ Catálogo
 - ▶ Garantizar la integridad
 - ▶ Permitir actualizaciones
 - ▶ Recuperación de los datos
 - ▶ Integración
 - ▶ Cumplir restricciones
 - ▶ Herramientas de administración

1.4.2.1 Un Catálogo

- ▶ Donde se almacenen las descripciones de los datos y sea accesible por los usuarios. Este **catálogo es lo que** se denomina *diccionario de datos y contiene información que describe los datos de la base de datos (metadatos)*.
- ▶ Normalmente, un diccionario de datos describe entre otras cosas:
 - ▶ Nombre, tipo y tamaño de los datos.
 - ▶ Relaciones entre los datos.
 - ▶ Restricciones de integridad sobre los datos.
 - ▶ Usuarios autorizados a acceder a los objetos de base de datos.
 - ▶ Estadísticas de utilización, tales como la frecuencia de las transacciones y el número de accesos realizados a los objetos de la base de datos.

1.4.2.2 Integridad de los datos

- ▶ Disponer de un mecanismo que garantice que todas las actualizaciones correspondientes a una determinada transacción se realicen, o que no se realice ninguna.
- ▶ Una transacción es un conjunto de acciones que cambian el contenido de la base de datos.
- ▶ Una transacción en el sistema informático de la empresa inmobiliaria sería dar de alta a un jugador o eliminar un inmueble.
- ▶ Una transacción un poco más complicada sería eliminar un jugador y reasignar sus inmuebles a otro jugador. En este caso hay que realizar varios cambios sobre la base de datos. Si la transacción falla durante su realización, por ejemplo, porque falla el hardware, la base de datos quedará en un estado inconsistente.
- ▶ Algunos de los cambios se habrán hecho y otros no, por lo tanto, los cambios realizados deberán ser deshechos para devolver la base de datos a un estado consistente.

1.4.2.3 Permitir actualizaciones

- ▶ Asegurar que la base de datos se actualice correctamente cuando varios usuarios la están actualizando concurrentemente.
- ▶ Uno de los principales objetivos de los SGBD es el de permitir que varios usuarios tengan acceso concurrente a los datos que comparten.
- ▶ El acceso concurrente es relativamente fácil de gestionar si todos los usuarios se dedican a leer datos, ya que no pueden interferir unos con otros. Sin embargo, cuando dos o más usuarios están accediendo a la base de datos y al menos uno de ellos está actualizando datos, pueden interferir produciendo inconsistencias en la base de datos.
- ▶ El SGBD debe garantizar que no se produzcan, es decir que los datos estén en todo momento en un estado consistente.

1.4.2.4 Recuperación Datos

- ▶ Permitir recuperar las bases de datos en caso de que ocurra algún suceso imprevisto que afecte o destruya la base de datos.
- ▶ Como se ha comentado antes, cuando el sistema falla en medio de una transacción, la base de datos se debe devolver a un estado consistente.
- ▶ Esta falta puede ser a causa de un fallo en algún dispositivo hardware o un error del software, que hagan que el SGBD aborte, o puede ser a causa de que el usuario detecte un error durante la transacción y la aborte antes de que finalice.
- ▶ También puede ser que simplemente se pierdan los datos por cualquier motivo.
- ▶ En todos estos casos, el SGBD debe proporcionar mecanismos capaces de recuperar y llevar la base de datos consistente lo más cercano posible en el tiempo al momento del fallo.

1.4.2.5 Integración

- ▶ Ser capaz de integrarse con algún software de comunicación. Muchos usuarios acceden a la base de datos desde terminales.
- ▶ En ocasiones estos terminales se encuentran conectados directamente a la máquina sobre la que funciona el SGBD.
- ▶ En otras ocasiones los terminales están en lugares remotos, por lo que la comunicación con la máquina que alberga al SGBD se debe hacer a través de una red.
- ▶ En cualquiera de los dos casos, el SGBD recibe peticiones en forma de mensajes y responde de modo similar.
- ▶ Todas estas transmisiones de mensajes las maneja el gestor de comunicaciones de datos.
- ▶ Aunque este gestor no forma parte del SGBD, es necesario que el SGBD se pueda integrar con él para que el sistema sea comercialmente viable.

1.4.2.6 Cumplir Restricciones

- ▶ Proporcionar los medios necesarios para garantizar que tanto los datos de la base de datos, como los cambios que se realizan sobre estos datos, sigan ciertas reglas.
- ▶ Se puede considerar como otro modo de proteger la base de datos, pero además de tener que ver con la seguridad, tiene otras implicaciones.
- ▶ La integridad se ocupa de la calidad de los datos. Normalmente se expresa mediante restricciones, que son una serie de reglas que la base de datos no puede violar.
- ▶ Por ejemplo, en una base de datos de una liga de baloncesto se puede establecer la restricción de que cada equipo no puede tener asignados más de veinte jugadores. En este caso sería deseable que el SGBD controlara que no se sobrepase este límite cada vez que se asigne un jugador a un equipo.

1.4.2.7 Herramientas Administración

- ▶ Proporcionar herramientas que permitan administrar la base de datos de modo efectivo, lo que implica un diseño óptimo de las mismas, garantizar la disponibilidad e integridad de los datos, controlar el acceso al servidor y a los datos, monitorizar el funcionamiento del servidor y optimizar su funcionamiento.
- ▶ Muchas de ellas van integradas en el sistema gestor, otras son creadas por terceros o por el propio administrador según sus requerimientos.

1.4.3 Componentes SGBD

- ▶ Son los elementos que deben proporcionar los servicios comentados en la sección anterior. No se pueden generalizar ya que varían mucho según la tecnología. Sin embargo, normalmente todo SGBD incluye los siguientes:
 - ▶ Lenguajes de Base de Datos
 - ▶ Diccionario de Datos
 - ▶ Objetos
 - ▶ Herramientas
 - ▶ Otras

1.4.3.1 Lenguajes SGBD

- ▶ Son lenguajes para la manipulación de datos, tanto desde el punto de vista de su acceso y modificación como del control y seguridad de los mismos.
- ▶ Normalmente se distinguen tres tipos según su funcionalidad:
 - ▶ Lenguaje de Definición de datos. (LDD o DDL)
 - ▶ Lenguaje de Manipulación de datos. (LMD o DML)
 - ▶ Lenguaje de Control de datos. (LCD o DCL)
- ▶ *Para todos estos lenguajes se usa principalmente el lenguaje SQL (Structured Query Language). Incluye instrucciones para los tres tipos de lenguajes comentados y por su sencillez y potencia se ha convertido en el lenguaje estándar de los SGBD relacionales.*

Lenguaje Definición Datos (DDL)

Lenguaje de definición de datos (DDL, *Data Definition Language*)

- ▶ Sencillo lenguaje artificial para definir y describir los objetos de la base de datos, su estructura, relaciones y restricciones.
- ▶ Para ello se han de definir los tres niveles de la base de datos (relacionadas con sus tres esquemas).
 - ▶ **Estructura interna**
 - ▶ **Estructura conceptual**
 - ▶ **Estructura externa**
- ▶ **Algunas operaciones posibles:**
 - ▶ Se definen las estructuras de datos
 - ▶ Se definen las relaciones entre los datos
 - ▶ Se definen las reglas que han de cumplir **los** datos

```
CREATE TABLE My_table(  
    my_field1    INT,  
    my_field2    VARCHAR(50),  
    my_field3    DATE        NOT NULL,  
    PRIMARY KEY (my_field1, my_field2)  
);
```

Lenguaje Manipulación Datos (DML)

Lenguaje de manipulación de datos (DML, *Data Manipulation Language*)

- ▶ Es el lenguaje encargado de la manipulación del contenido de las bases de datos.
- ▶ Permite la inserción, actualización, eliminación y consulta de datos en las tablas de las bases de datos.
- ▶ Este tipo de sentencias se realizan mediante un lenguaje de alto nivel llamado anfitrión que incluye de forma embebidas las sentencias del lenguaje LMD.

```
SELECT *  
  FROM Book  
 WHERE price > 100.00  
 ORDER BY title;
```

```
INSERT INTO My_table  
  (field1, field2, field3)  
VALUES  
  ('test', 'N', NULL);
```

Lenguaje Control Datos (DCL)

Lenguaje de control de datos (**DCL**, *Data Control Language*)

- ▶ Encargado del control y seguridad de los datos (privilegios y modos de acceso, etc.).
- ▶ Este lenguaje permite especificar los permisos sobre los objetos de las bases de datos (tablas, vistas, procedimientos, etc.), así como la creación y eliminación de usuarios y cuentas.

```
GRANT SELECT, UPDATE
ON My_table
TO some_user, another_user;

REVOKE SELECT, UPDATE
ON My_table
FROM some_user, another_user;
```

1.4.3.2 Diccionario de Datos

- ▶ Es el lugar donde se deposita información acerca de todos los datos que forman la BD. Es una guía donde se describe la BD y todos los objetos que la forman.
- ▶ El diccionario contiene las características lógicas de los sitios donde se almacena los datos del sistema, incluyendo el nombre, descripción, alias, contenido y organización. Identifica los procesos donde se emplean los datos y los sitios donde se necesita el acceso inmediato a la información.
- ▶ En una BD relacional proporciona la siguiente información:
 - ▶ Estructura lógica y física de la BD
 - ▶ Las definiciones de todos los objetos de la BD: tablas, vistas, índices, disparadores.
 - ▶ Espacio asignado y utilizado por los objetos
 - ▶ Restricciones de integridad
 - ▶ Los privilegios y roles de los usuarios
 - ▶ Auditoria de la información sobre el acceso a los objetos.

1.4.3.3 Objetos

- ▶ **Tablas base y vistas (tablas derivadas).**
- ▶ **Consultas.**
- ▶ **Dominios y tipos definidos de datos.**
- ▶ **Restricciones de tabla y dominio y aserciones.**
- ▶ **Funciones y procedimientos almacenados.**
- ▶ **Disparadores o *triggers*.**

1.4.3.4 Herramientas

- ▶ Seguridad: de modo que los usuarios no autorizados no puedan acceder a la base de datos.
- ▶ Integridad: que mantiene la integridad y la consistencia de los datos.
- ▶ El control de concurrencia: que permite el acceso compartido a la base de datos.
- ▶ El control de recuperación: que restablece la base de datos después de que se produzca un fallo del hardware o del software.
- ▶ Gestión del diccionario de datos (o catálogo): accesible por el usuario que contiene la descripción de los datos de la base de datos.
- ▶ Programación de aplicaciones.
- ▶ Importación/exportación de datos (migraciones).
- ▶ Distribución de datos.
- ▶ Replicación (arquitectura maestro-esclavo).
- ▶ Sincronización (de equipos replicados).

1.4.3.5 Varias

- ▶ **Optimizador de Consultas.** Para determinar la estrategia óptima para la ejecución de las consultas.
- ▶ **Gestión de Transacciones.** Este módulo realiza el procesamiento de las transacciones.
- ▶ **Planificador (*scheduler*).** Para programar y automatizar la realización de ciertas operaciones y procesos.
- ▶ **Copias de seguridad.** Para garantizar que la base de datos se puede devolver a un estado consistente en caso de que se produzca algún fallo o error grave.

1.4.4 Usuarios SGBD

- ▶ Generalmente distinguimos cuatro grupos de usuarios de sistemas gestores de bases de datos:
 - ▶ los usuarios administradores
 - ▶ los diseñadores de la base de datos
 - ▶ los programadores
 - ▶ usuarios de aplicaciones que interactúan con las bases de datos.

1.4.4.1 Administradores

- ▶ Trabajan en el nivel de abstracción físico relacionado con el almacenamiento.
- ▶ Distinguimos los administradores del propio sistema gestor encargados:
 - ▶ La instalación y configuración del sistema
 - ▶ Control de acceso a los recursos
 - ▶ Seguridad
 - ▶ Monitorización y optimización del sistema gestor.
- ▶ Por su parte los administradores de bases de datos se encargan:
 - ▶ Diseño físico de la misma
 - ▶ Implementación y mantenimiento de la base de datos

1.4.4.2 Diseñadores

- ▶ 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.
- ▶ El diseño lógico de la base de datos es independiente del SGBD concreto que se vaya a utilizar, es independiente de los programas de aplicación, de los lenguajes de programación y de cualquier otra consideración física.

1.4.4.3 Programadores

- ▶ Tanto de aplicaciones que, mediante API de lenguajes de programación, interactúan con las bases de datos como de objetos de la base de datos como rutinas almacenadas o disparadores.
- ▶ Estas aplicaciones servirán a los usuarios finales para, de una forma amigable, poder consultar datos, insertarlos, actualizarlos y eliminarlos.

1.4.4.4 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.

1.4.5 MODELO ANSI/X3/SPARC

- ▶ El organismo ANSI ha marcado la referencia para la construcción de SGBD.
- ▶ ANSI se indica que hay tres distintos tipos: *externo*, *conceptual* e *interno*
 - ▶ Los esquemas externos reflejan la información preparada (filtrada en forma de vistas) para el usuario final.
 - ▶ El esquema conceptual refleja los datos y relaciones de la base de datos (el nivel lógico)
 - ▶ El esquema interno determina la organización física de los datos (sistemas, de ficheros, estructura de directorios, espacios de almacenamiento, índices, etc.)
- ▶ El modelo ANSI es una propuesta teórica sobre las características deseables en un sistema gestor de bases de datos

1.4.6 Tipos Sistemas Gestores Bases de Datos

- ▶ **Modelo lógico en el que se basan:**
 - ▶ Jerárquico.
 - ▶ En red.
 - ▶ Relacional.
 - ▶ Objeto-relacional.
 - ▶ Orientado a objetos.
- ▶ **Número de usuarios:**
 - ▶ Mono usuarios: sólo permite un usuario
 - ▶ Multiusuario: permite la conexión con varios usuarios
- ▶ **Según la distribución componentes:**
 - ▶ Distribuidas: en varios equipos homogéneos y heterogéneos.
 - ▶ Centralizadas: en un solo servidor o equipo
- ▶ **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.
- ▶ **Lenguajes soportados**
 - ▶ SQL estándar.
 - ▶ NoSQL o nuevo lenguaje de consulta: menos estructurado y orientado a bases documentales o de tipo clave-valor. Es muy útil para manejar consultas de grandes cantidades de datos distribuidos en clusters de servidores.

1.4.7 SGBD Comerciales y Libres

- ▶ La elección de un SGBD es una decisión muy importante a la hora de desarrollar proyectos. A veces, el sistema más avanzado, “el mejor” según los entendidos, puede no serlo para el tipo de proyecto que estemos desarrollando.
- ▶ Comerciales:
 - ▶ Son los que se distribuyen bajo licencia comercial, con derecho de autor y copyright
- ▶ Libres:
 - ▶ SGBD de código abierto o libres, también llamados Open Source.
 - ▶ Son sistemas que se desarrollan y distribuyen libremente pero que exigen un personal más cualificado para su uso

SGBD COMERCIAL	SGBD LIBRES
ORACLE	MYSQL (Oracle)
DB2 (IBM)	PostgreSQL
INFORMIX (IBM)	Firebird
SQL SERVER (Microsoft)	Apache Derby
SYBASE	SQLite

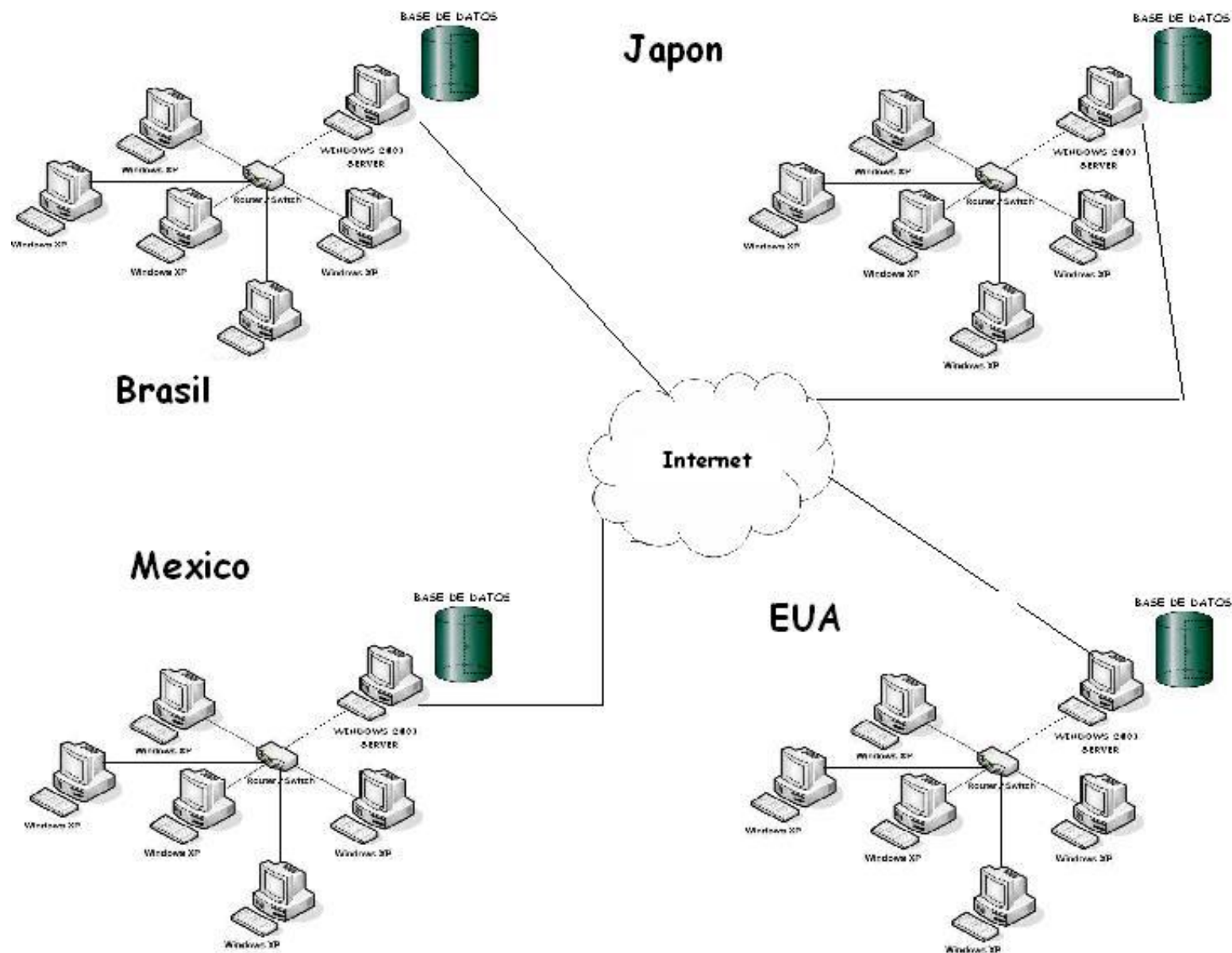
1.5 BBDD Centralizadas y Distribuidas

▶ BBDD Centralizada:

- ▶ Todos los componentes (software, datos y soportes físicos) residen en un único lugar físico.
- ▶ Los clientes (aplicaciones, funciones, programas cliente, usuarios) acceden al sistema a través de distintas interfaces que se conectan al servidor.

▶ BBDD Distribuidas:

- ▶ Los componentes se distribuyen en distintos computadores comunicados a través de una red de cómputo.
- ▶ Dichos sistemas se conocen con el nombre de Sistemas Gestores de Bases de Datos Distribuidas o DDMGS
- ▶ colección de datos que pertenece lógicamente al mismo sistema pero que se encuentra físicamente almacenada en distintas máquinas conectadas por una red



Razones surgimiento

▶ **Mejora de rendimiento**

- ▶ Consultas y transacciones más rápidas.
- ▶ Los sitios no se ven congestionados por muchas transacciones, ya que éstas se dispersan entre varios sistemas

▶ **Fiabilidad.**

- ▶ Si falla algún sitio es poco probable que afecte a la mayoría de transacciones.

▶ **Disponibilidad.**

- ▶ Mismas razones expuestas anteriormente. Esta característica se ve mejorada especialmente por la replicación de datos en varios sistemas.

▶ **Tipos de aplicaciones.**

- ▶ Muchas aplicaciones tienen un marcado carácter distribuido al estar sus componentes dispersos en distintos sitios.

Dificultades

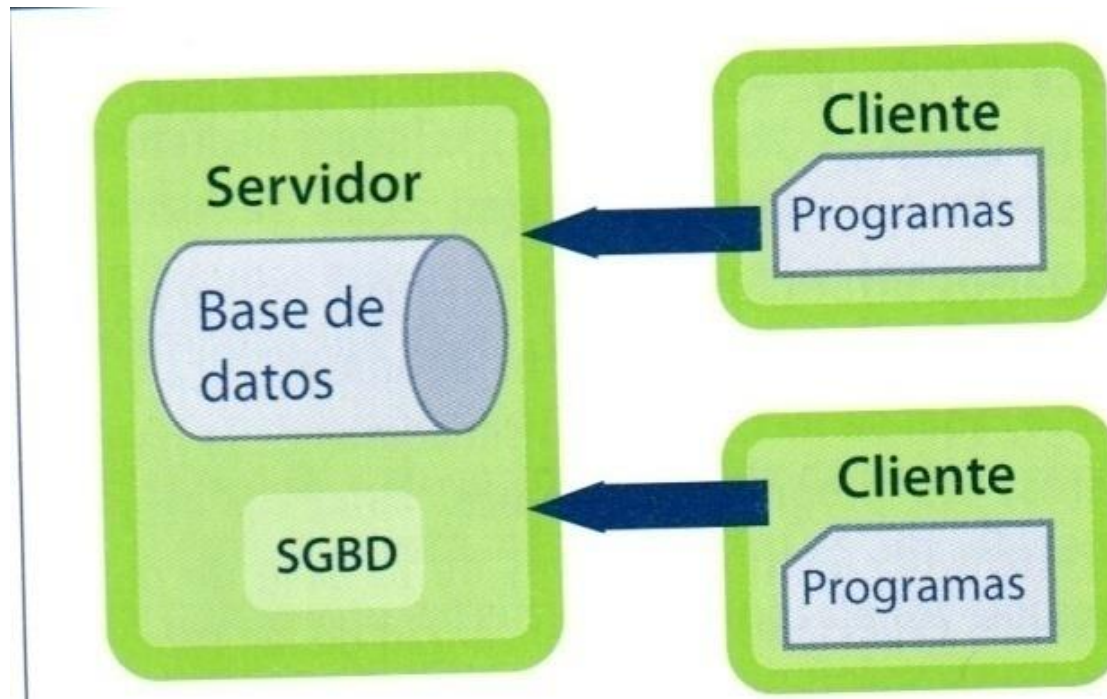
- ▶ **Distribuir implica complejidad:**
 - ▶ Diseño, implementación y gestión de datos
- ▶ **Nuevos componentes SGBDD:**
 - ▶ Acceso remotos e intercambiar información
 - ▶ Catálogo con información de cómo están distribuidos y replicados los datos
 - ▶ Optimizador de consultas y transacciones sobre datos replicados
 - ▶ Mantener la integridad en datos replicados
 - ▶ Mantener la consistencia de copias en datos replicados
 - ▶ Garantizar la recuperación de datos en una caída del sistema
- ▶ **Añadir la dificultad de diseño añadida:**
 - ▶ ¿Qué datos distribuir y qué datos replicar?

Ventajas

- ▶ El acceso y procesamiento de los datos es más rápido ya que varios nodos comparten carga de trabajo.
- ▶ Desde una ubicación puede accederse a información alojada en diferentes lugares.
- ▶ Los costes son inferiores a los de las bases centralizadas.
- ▶ Existe cierta tolerancia a fallos. Mediante la replicación, si un nodo deja de funcionar el sistema completo no deja de funcionar.
- ▶ El enfoque distribuido de las bases de datos se adapta más naturalmente a la estructura de las organizaciones. Permiten la incorporación de nodos de forma flexible y fácil.
- ▶ Aunque los nodos están interconectados, tienen independencia local.

1.5.1 Arquitectura DDBMS

- ▶ **Basada en la Arquitectura Cliente/Servidor**
 - ▶ La arquitectura cliente-servidor es un modelo de SGBD distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes
 - ▶ Componentes:
 - ▶ Servidor: Parte principal del SGBD, gestiona la BD. Se relaciona con la máquina y con el sistema operativo.
 - ▶ Cliente: Parte que utilizan los usuarios, las aplicaciones, los lenguajes y en ocasiones, las herramientas de gestión y administración.



Posibilidades de Organización

- ▶ **Autonomía.** Determina el nivel de independencia de los datos en un SGBD
 - ▶ Integración Fuerte: un equipo hace de coordinador
 - ▶ Semiautónomos: los SGBD son independientes pero participan en el conjunto compartiendo parte de sus datos.
 - ▶ Sistema aislado: el SGBD no tiene constancia de la existencia de otros SGBD
- ▶ **Distribución:**
 - ▶ Sin distribución de datos
 - ▶ Clientes Servidor: los datos residen en los servidores mientras que los clientes tienen un interfaz de acceso a los mismos
 - ▶ Servidores Colaborativos: cada máquina tiene toda la funcionalidad de un SGBD.
- ▶ **Heterogeneidad:**
 - ▶ Se refiere a la heterogeneidad de los componentes en los distintos niveles: hardware, comunicaciones y sistema operativo.
 - ▶ Cualquier combinación de estos parámetros determina un modelo arquitectónico distinto.

1.5.2 Técnicas Fragmentación, Replicación y Distribución

- ▶ Diseñar bases de datos distribuidas implica decidir sobre cómo fragmentarlas, cómo replicarlas y cómo distribuirlas, además del diseño previo de la base de datos en sí
- ▶ **Fragmentación:**
 - ▶ Fragmentación horizontal que divide las tuplas según una o varias condiciones sobre distintos atributos, ejemplo dividir las tuplas según el valor de la región.
 - ▶ Fragmentación vertical que divide la información entre los campos más accedidos y ligeros y los de más peso. Los primeros se almacenarían en servidores más potentes, mientras que los segundos en servidores con mayor capacidad de almacenamiento.
 - ▶ Fragmentación mixta.
 - ▶ Desventajas:
 - ▶ Costos y eleva el nivel de gestión
 - ▶ Ventajas:
 - ▶ Acerca los datos al cliente
 - ▶ Mejora el tráfico de la red
 - ▶ Mejora la disponibilidad de los datos

Replicación

▶ Replicación:

- ▶ Fundamental para mejorar la disponibilidad de los datos.
- ▶ Tipos:
 - ▶ Replicación Completa: BBDD replicada en cada uno de los servidores. Mayor disponibilidad pero reduce la eficiencia.
 - ▶ No Replicación. Cada fragmento de la BBDD está repartido entre los servidores, denominado reparto no redundante.
 - ▶ Catálogo de replicación. Opción intermedia donde parte de la BBDD está replicada y otra no. Se requiere de un esquema de replicación:
 - Objetos replicados
 - Donde se encuentra cada réplica
 - Cómo se propagan las actualizaciones
- ▶ Ventajas:
 - ▶ Distribuye la carga
 - ▶ Acelera las consultas
 - ▶ Mejora la disponibilidad
 - ▶ Óptimo para sistemas con muchas lecturas
- ▶ Desventajas:
 - ▶ Dificulta las actualizaciones