

# INTRODUCCIÓN Y PERSPECTIVA GENERAL DE LOS SISTEMAS DE INFORMACIÓN

---

Tema 1

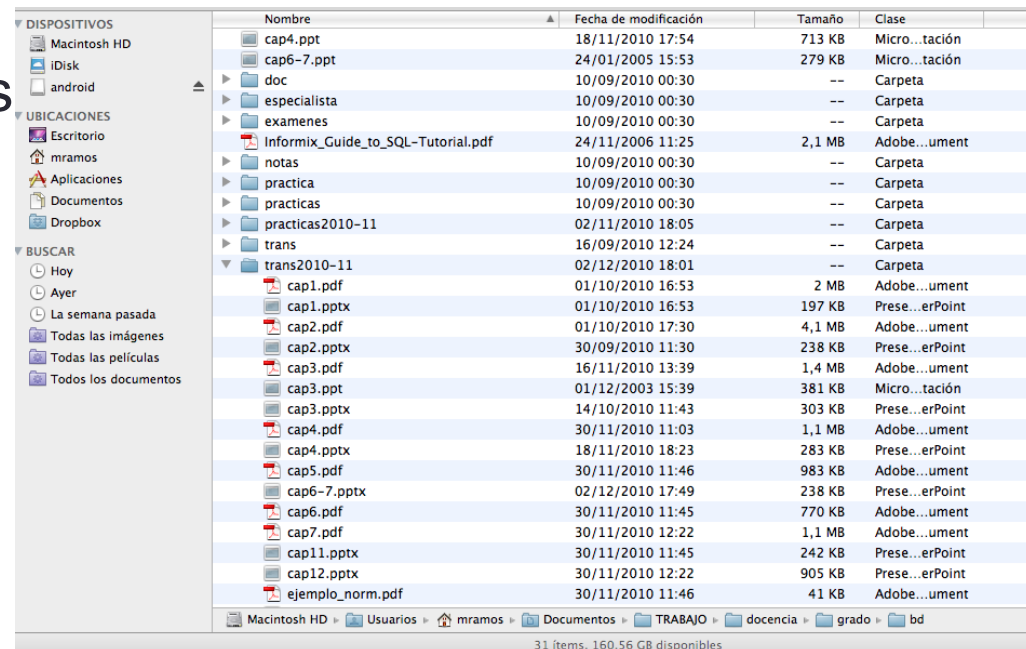
Manuel Ramos Cabrer

# ÍNDICE

- Introducción
- Definición de sistema de información y base de datos
- Sistemas Gestores de Bases de Datos (SGBD)
- Conceptos básicos
- Diseño de bases de datos
- El modelo Entidad-Asociación (E-A)
- El modelo lógico relacional
- El lenguaje SQL

# Sistemas de archivos (o ficheros)

- Surgieron a raíz de la necesidad de almacenamiento de la información para su correspondiente reutilización (**persistencia**)
- Cada aplicación define y utiliza sus propios ficheros,
  - con sus propios datos
  - con sus propios formatos

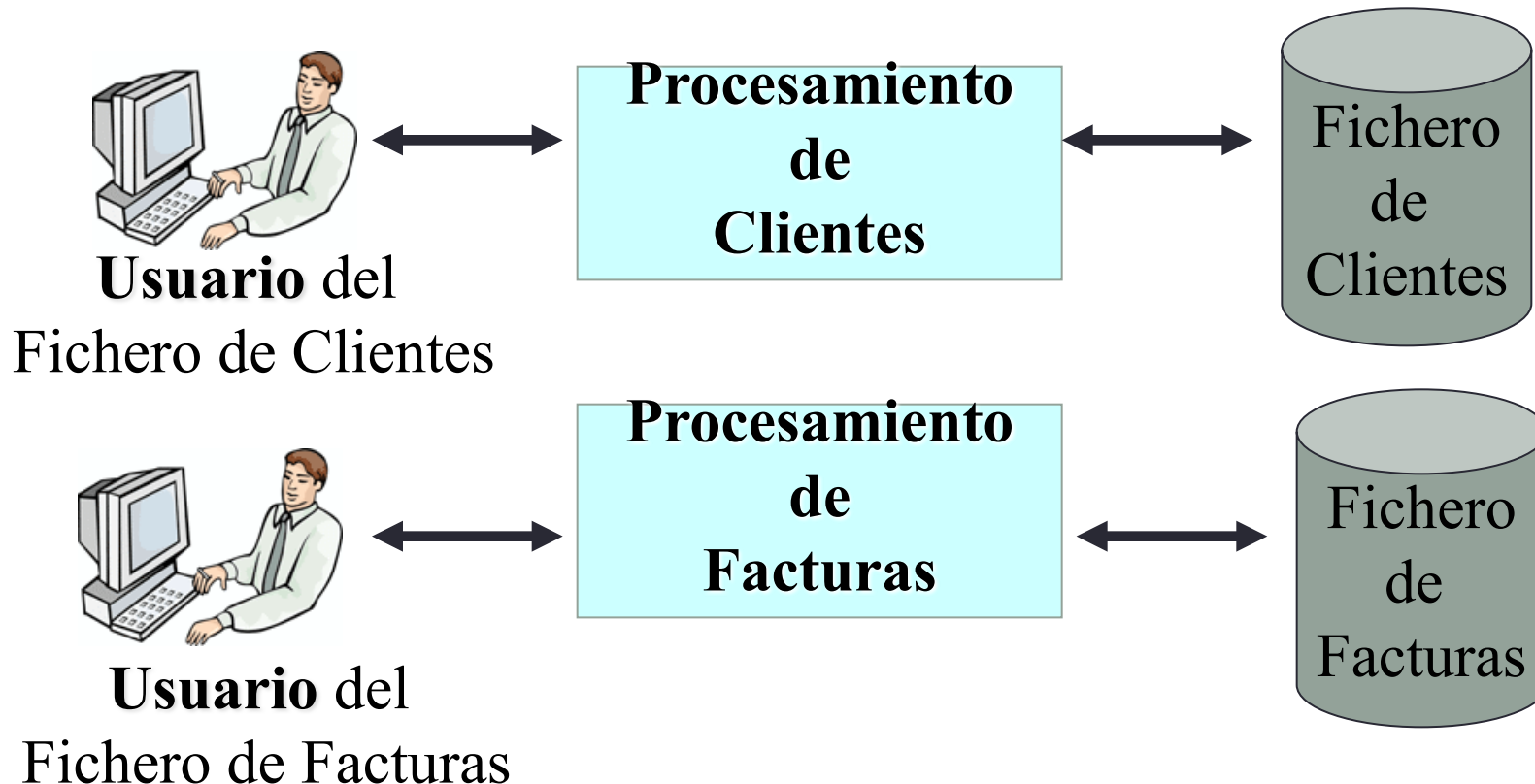


Nombre	Fecha de modificación	Tamaño	Clase
cap4.ppt	18/11/2010 17:54	713 KB	Micro...tación
cap6-7.ppt	24/01/2005 15:53	279 KB	Micro...tación
doc	10/09/2010 00:30	--	Carpeta
especialista	10/09/2010 00:30	--	Carpeta
exámenes	10/09/2010 00:30	--	Carpeta
Informix_Guide_to_SQL-Tutorial.pdf	24/11/2006 11:25	2,1 MB	Adobe...ument
notas	10/09/2010 00:30	--	Carpeta
practica	10/09/2010 00:30	--	Carpeta
practicass	10/09/2010 00:30	--	Carpeta
practicass2010-11	02/11/2010 18:05	--	Carpeta
trans	16/09/2010 12:24	--	Carpeta
trans2010-11	02/12/2010 18:01	--	Carpeta
cap1.pdf	01/10/2010 16:53	2 MB	Adobe...ument
cap1.pptx	01/10/2010 16:53	197 KB	Prese...erPoint
cap2.pdf	01/10/2010 17:30	4,1 MB	Adobe...ument
cap2.pptx	30/09/2010 11:30	238 KB	Prese...erPoint
cap3.pdf	16/11/2010 13:39	1,4 MB	Adobe...ument
cap3.ppt	01/12/2003 15:39	381 KB	Micro...tación
cap3.pptx	14/10/2010 11:43	303 KB	Prese...erPoint
cap4.pdf	30/11/2010 11:03	1,1 MB	Adobe...ument
cap4.pptx	18/11/2010 18:23	283 KB	Prese...erPoint
cap5.pdf	30/11/2010 11:46	983 KB	Adobe...ument
cap6-7.pptx	02/12/2010 17:49	238 KB	Prese...erPoint
cap6.pdf	30/11/2010 11:45	770 KB	Adobe...ument
cap7.pdf	30/11/2010 12:22	1,1 MB	Adobe...ument
cap11.pptx	30/11/2010 11:45	242 KB	Prese...erPoint
cap12.pptx	30/11/2010 12:22	905 KB	Prese...erPoint
ejemplo_norm.pdf	30/11/2010 11:46	41 KB	Adobe...ument

Macintosh HD > Usuarios > mramos > Documentos > TRABAJO > docencia > grado > bd

31 items, 160,56 GB disponibles

# Ejemplo de ficheros convencionales



# Ejemplo de ficheros convencionales

- El primer fichero contiene los datos básicos de los clientes, mientras que el segundo almacena las ventas realizadas.

## Clientes

Num	Nombre	Dirección	Telefono	E-mail
1225	Juan García	America, 4	4182569	jgarc@gmail.com
1226	Fernando Martínez	Florida 27	9157878	fmar@gmail.com
....	.....	.....	.....	.....

## Facturas

Num	Nombre	Dom.	Producto	Precio
1225	Joaquín García	Castrelos, 5	Azulejos	1250
1226	Fernando Martínez	Romil, 2	Pintura	900
.....	.....	.....	.....	.....

## Problemas:

- Duplicidad de esfuerzos
- Redundancia
- Posibles inconsistencias

# Sistemas de ficheros: desventajas

- Información duplicada en varios ficheros (**redundancia** de datos):
  - Alto costo de propagación de cambio
  - Coste de almacenamiento
  - Posibles **inconsistencias**
- Dificultad en acceder a los datos
  - Necesidad de escribir un nuevo programa para cada nueva tarea
- Aislamiento de datos (múltiples ficheros y formatos)
  - No se pueden combinar fácilmente
  - Posibles pérdidas de información
- Problemas de integridad
  - Las restricciones de integridad (p.e.  $\text{precio} > 0$ ) son parte del código del programa
  - Dificultad en añadir nuevas restricciones o cambiar las existentes
- Dificultad para gestionar grandes cantidades de información
- Falta de atomicidad en las actualizaciones
- Problemas a la hora de **compartir** datos (seguridad e integridad)

# Concepto de sistema de información

*Es una colección de datos operativos almacenados y utilizados por los programadores de aplicaciones y por usuarios finales de muy diversa índole*

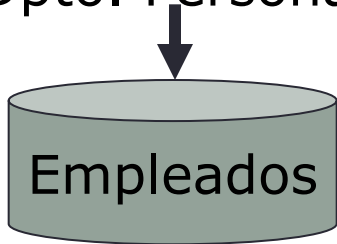
Ideas básicas tras el concepto de SI:

Solucionar o minimizar los problemas anteriores  
Aprovechar los mismos datos para tantas aplicaciones como sea posible.

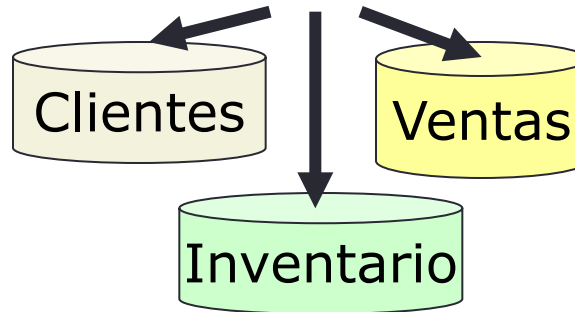
... Antes



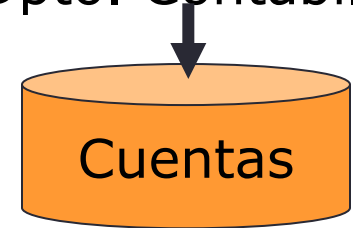
Dpto. Personal



Dpto. Ventas



Dpto. Contabilidad



... Ahora





# Aplicaciones típicas

- Banca: Clientes, cuentas, prestamos, operaciones, operaciones en cajeros, tarjetas de crédito,...
- Líneas aéreas: Reservas, vuelos, clientes, operaciones,...
- Universidades: alumnos, personal, matrículas, expedientes, contabilidad,...
- Operadores de telecomunicaciones: clientes, tarifas, llamadas, tarjetas, prepagos, terminales,....
- Servicios por internet: catálogos, ventas, blogs, buscadores, redes sociales,...

Los sistemas de información cada vez están más presentes en nuestras vidas

# Tipos de sistemas de información

- Estructurados:
  - Se organizan en torno a una estructura formal y rígida
  - Ejemplo: Bases de datos
- Semiestructurados:
  - Disponen de estructura, pero ésta no es rígida
  - Ejemplo: XML
- No estructurados
  - Textos
  - Imágenes
  - Audios
  - Vídeos
  - ...

# Concepto de base de datos

- Terminó acuñado en 1963
- Conjunto de información
  - relacionada
    - sobre un tema.
    - con un objetivo determinado.
  - organizada de manera lógica
    - para facilitar consulta.
    - evitando redundancias e inconsistencias.
- Ejemplos: alumnos, cursos, asignaturas, notas, expedientes,...

*“Base de Datos es un conjunto de datos relacionados entre sí y que tienen un significado implícito”.*

Ramez Elmasri y Shamkant B. Navathe

# Propiedades de los datos

- Es un conjunto exhaustivo (en su modelización del mundo real)
- Estructurados
- Organizados independientemente de las aplicaciones y del soporte de almacenamiento que los contiene.
- Compartibles por usuarios y/o aplicaciones concurrentes que tienen necesidades de información diferentes y no predecibles en el tiempo.
- Presentan la menor redundancia posible.
- Están bajo un control centralizado.

# SGBD: Definición

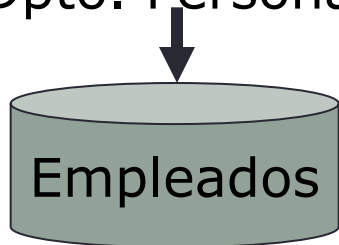
- Un **Sistema de Gestión de Bases de Datos (SGBD)** es un software que permite **manipular** las bases de datos

- Construir
- Utilizar
- Mantener
- Reorganizar

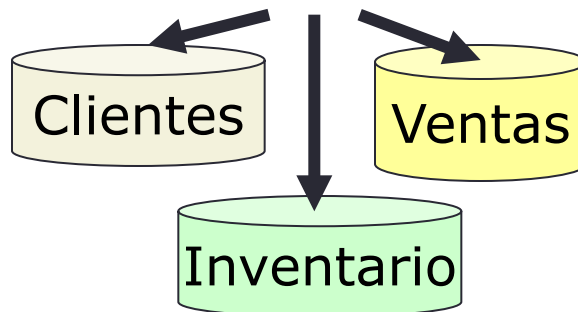
... Antes



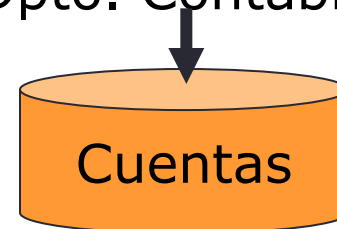
Dpto. Personal



Dpto. Ventas



Dpto. Contabilidad



... Ahora

Personal



Ventas



Contabilidad

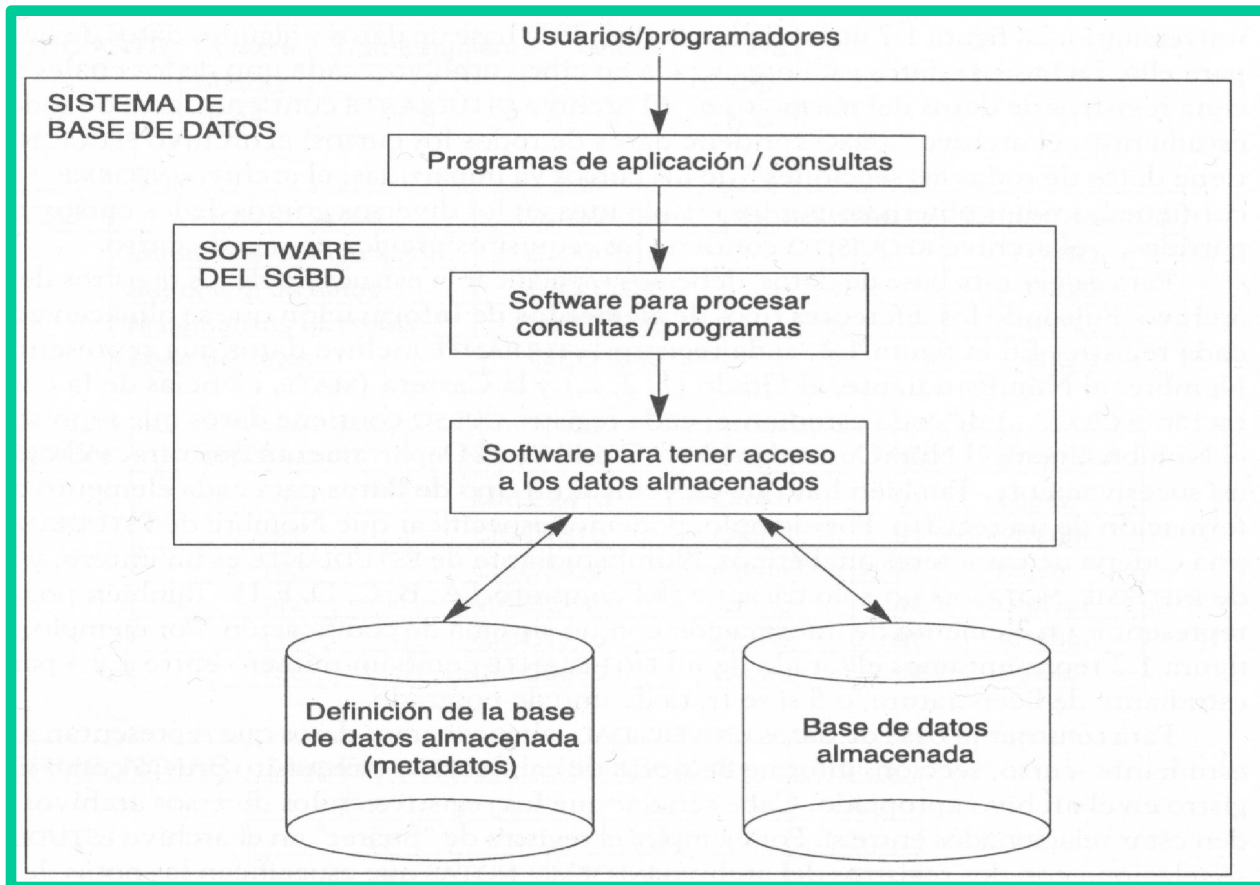


SGBD

**BASE DE DATOS**

**Empleados  
Clientes  
Ventas  
Inventario  
Cuentas**

# Sistema de Bases de Datos



En el esquema se observa que los **usuarios** interactúan con una **aplicación** (por ej. un sistema integrado de gestión o un paquete contable) que utiliza un **SGBD** para procesar las consultas, el cual accede a los **metadatos** y a la **base de datos** correspondiente.

# Sistema Gestor de Bases de Datos

- Un **Sistema de Bases de Datos** consiste en:
  - Una **colección de datos** interrelacionados (BD)
  - Un **conjunto de programas** para acceder a los datos
  - Un SGBD
- **Objetivo Principal de un SGBD:**
  - Proporcionar una forma **práctica y eficiente** de **almacenar y recuperar** la información de una Base de Datos
    - Proporciona una interfaz entre los datos almacenados a bajo nivel en la base de datos y los programas de aplicación y consultas enviadas al sistema.
- Definición de estructuras de almacenamiento, diseño de mecanismos de manipulación de los datos, control de la fiabilidad (ante caídas del sistema o intentos de acceso no autorizado), control de inconsistencias, ...

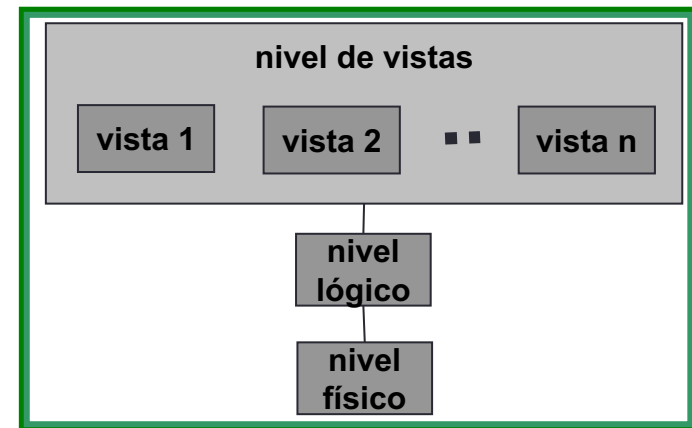
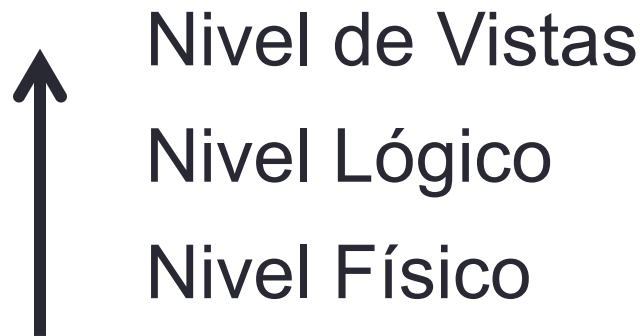


# Objetivos de un SGBD

- Acceso concurrente
- Independencia física y lógica
- Redundancia controlada de los datos
- Alta relacionabilidad de los datos
- Integridad de los datos
- Consistencia de los datos (soporte de transacciones)
- Seguridad de los datos
  - acceso
  - respaldo y recuperación
- Capacidad de auditoría
- Actualización fácil y coherente
- Alto rendimiento
- Reducción de la dificultad de programación

# Niveles de abstracción

- Otro de los objetivos fundamentales de un SGBD es proporcionar a todos los usuarios una **Visión Abstracta** de los datos almacenados:
  - Esconder detalles acerca de cómo se almacenan y mantienen los datos
- **Niveles de Abstracción**



# SGBD: Niveles de abstracción

- **NIVEL FÍSICO:**

- nivel más bajo de abstracción
- describe **cómo** se almacenan realmente los datos
- se describen en detalle las estructuras complejas de bajo nivel.

- **NIVEL CONCEPTUAL:**

- nivel intermedio y alto de abstracción
- describe **qué** datos son realmente almacenados en la BD y sus **relaciones**
- se describe la BD completa con estructuras sencillas correspondientes a otras del nivel físico, generalmente más complejas
- Es el más importante para nosotros: en este nivel se definen la base de datos

- **NIVEL de VISTAS:**

- nivel más alto de abstracción
- describe sólo **parte de la BD** completa (la que interese a cada usuario)
- suele proporcionarse muchas vistas distintas de la misma BD (tantas como tipos de usuarios)

# Nivel de Vistas

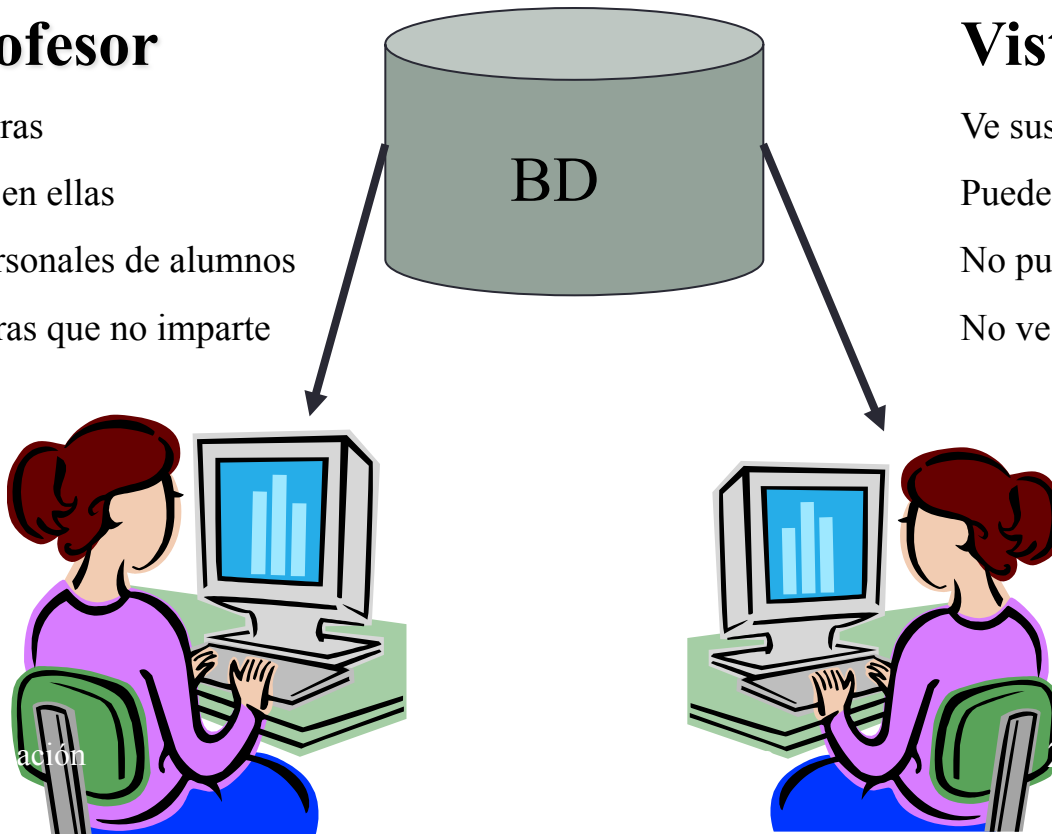
Cada vista puede proporcionar diferentes representaciones y niveles de acceso a los mismos datos

## Vista Profesor

- Ve sus asignaturas
- Puede calificar en ellas
- No ve datos personales de alumnos
- No ve asignaturas que no imparte

## Vista Alumno

- Ve sus calificaciones
- Puede modificar sus datos personales
- No puede modificar calificaciones
- No ve expedientes de otros



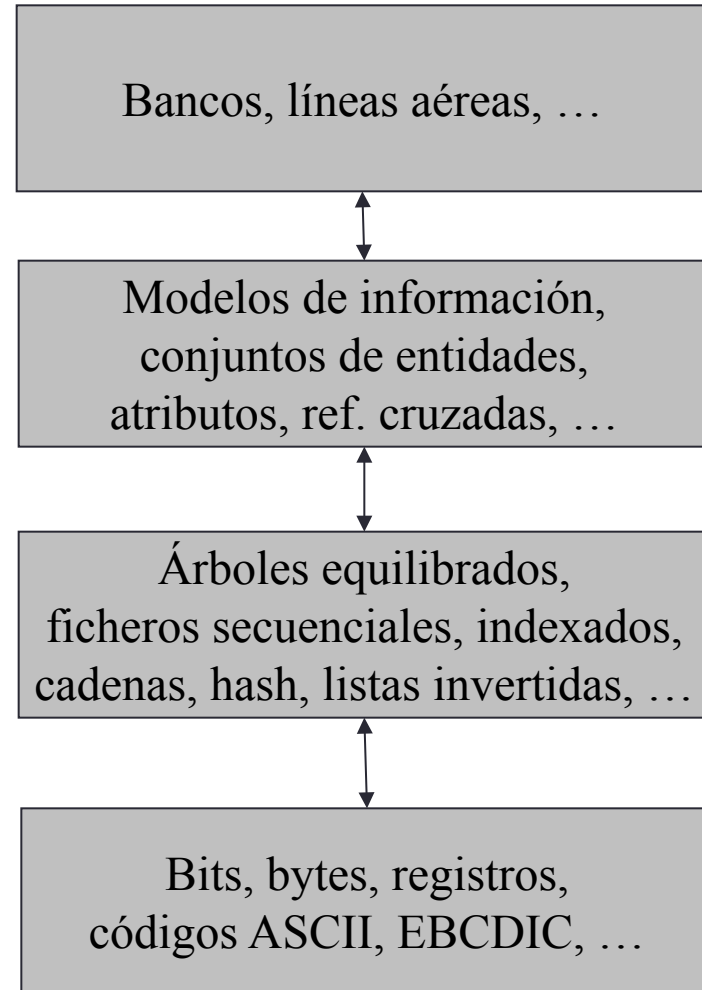
# Visión global

**El mundo real**

**Las bases de datos**

**Las estructuras de datos**

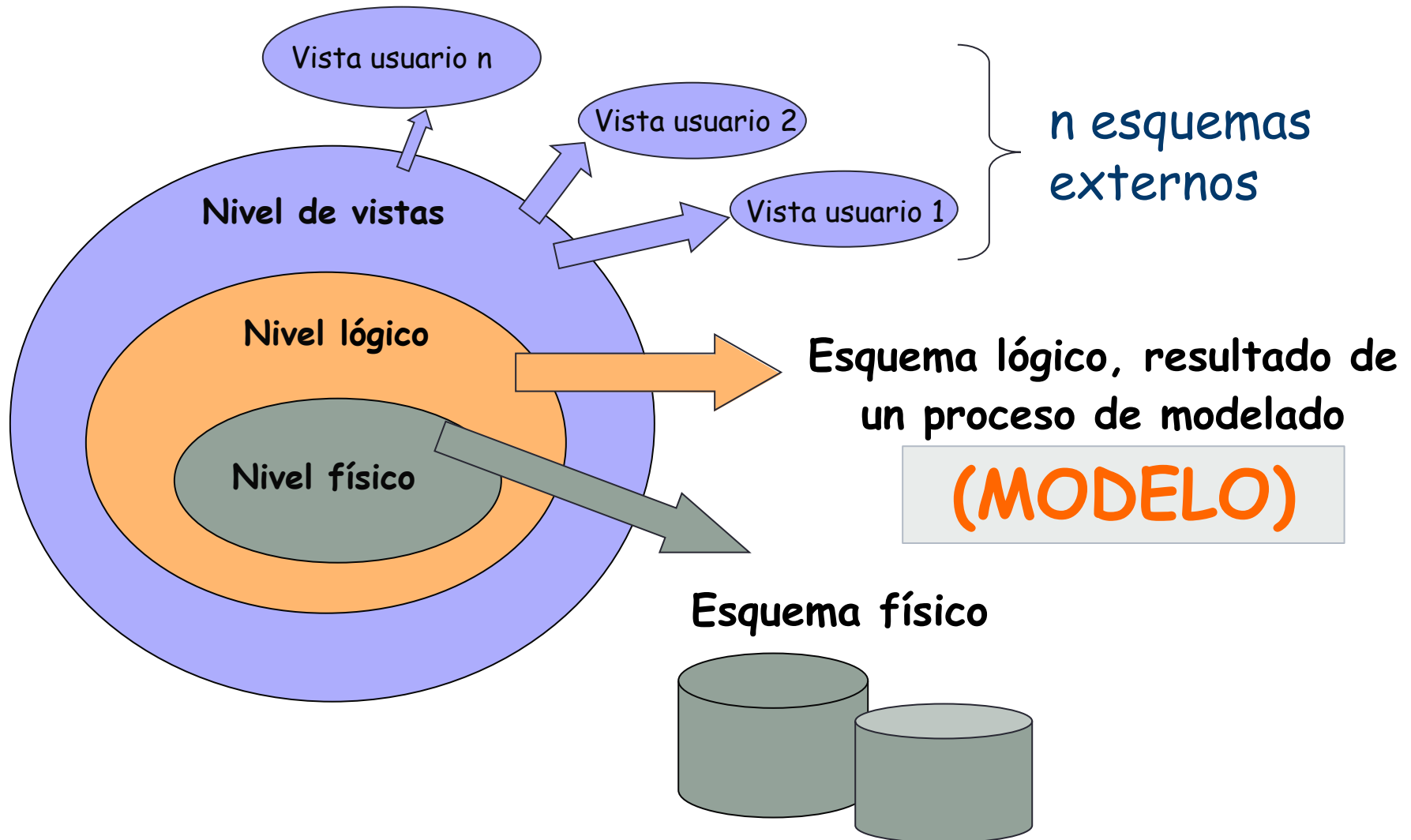
**El almacenamiento físico**



# Instancias, esquemas e independencia

- Similar a tipos de datos y variables en lenguajes de programación
- **Esquema** – la estructura lógica de la base de datos
  - P.e., la base de datos consiste en información sobre un conjunto de clientes y cuentas y la relación entre ellos)
  - Esta estructura permanece “estática” durante un gran período de tiempo, aunque puede sufrir modificaciones ocasionales
  - Análogo al tipo de datos de una variable en un programa
  - **Esquema físico**: el diseño de la base de datos a nivel físico
  - **Esquema lógico**: el diseño de la base de datos a nivel lógico
- **Instancia** – el contenido actual de la base de datos en un instante concreto
  - Varía de forma habitual
  - Análogo al valor de una variable
- **Independencia física de los datos** – la habilidad para modificar el esquema físico sin cambiar el esquema lógico
  - Las aplicaciones dependen del esquema lógico
  - En general, los interfaces entre los diferentes niveles y componentes deberían estar bien definidos para que los cambios en algunas partes no afecten de forma importante a otras.

# Niveles de abstracción



# Lenguajes soportados por un SGBD

- Lenguaje de definición de datos (DDL): creación y modificación de estructura de bases de datos.
  - Creación y modificación de esquemas
- Lenguaje de manipulación de datos (DML): acceso a los datos almacenados en la base de datos
  - Añadir, modificar, borrar, consultar datos de la instancia actual de la base de datos.



# Lenguajes de definición de datos (DDL)

- Especificación de notación para definir el esquema de las bases de datos
  - P.e.  

```
create table profesor (  
    ID                char(5),  
    nombre            varchar(20),  
    nombre_dpto varchar(20),  
    salario           numeric(8,2))
```
- El compilador de DDL genera un conjunto de tablas almacenadas en un *diccionario de datos*
- Los diccionarios de datos contienen metadatos (es decir, datos sobre los datos)
  - esquema de la base de datos
  - Restricciones de integridad
  - Permisos

## Lenguaje de manipulación de datos (DML)

- Lenguaje para acceder y manipular a los datos organizados por el modelo de datos apropiado
  - El DML también se conoce como lenguaje de consulta
- Dos clases de lenguajes
  - Procedimental – el usuario especifica que datos quiere consultar y como obtener esos datos
  - No procedimental – el usuario especifica que datos quiere consultar sin especificar como obtener esos datos
- SQL es el lenguaje de consulta más ampliamente usado

# SQL

- SQL: lenguaje no procedimental ampliamente utilizado
  - P.e. encontrar el nombre del profesor con identificador 22222

```
select  nombre
from    profesor
where profesor.ID = '22222'
```

- P.e. encontrar los identificadores y edificios de los profesores del departamento de Física

```
select profesor.ID, departamento.edificio
from    profesor, departamento
where profesor.nombre_dpto = departamento.nombre_dpto and
        departamento.nombre_dpto = 'Física'
```

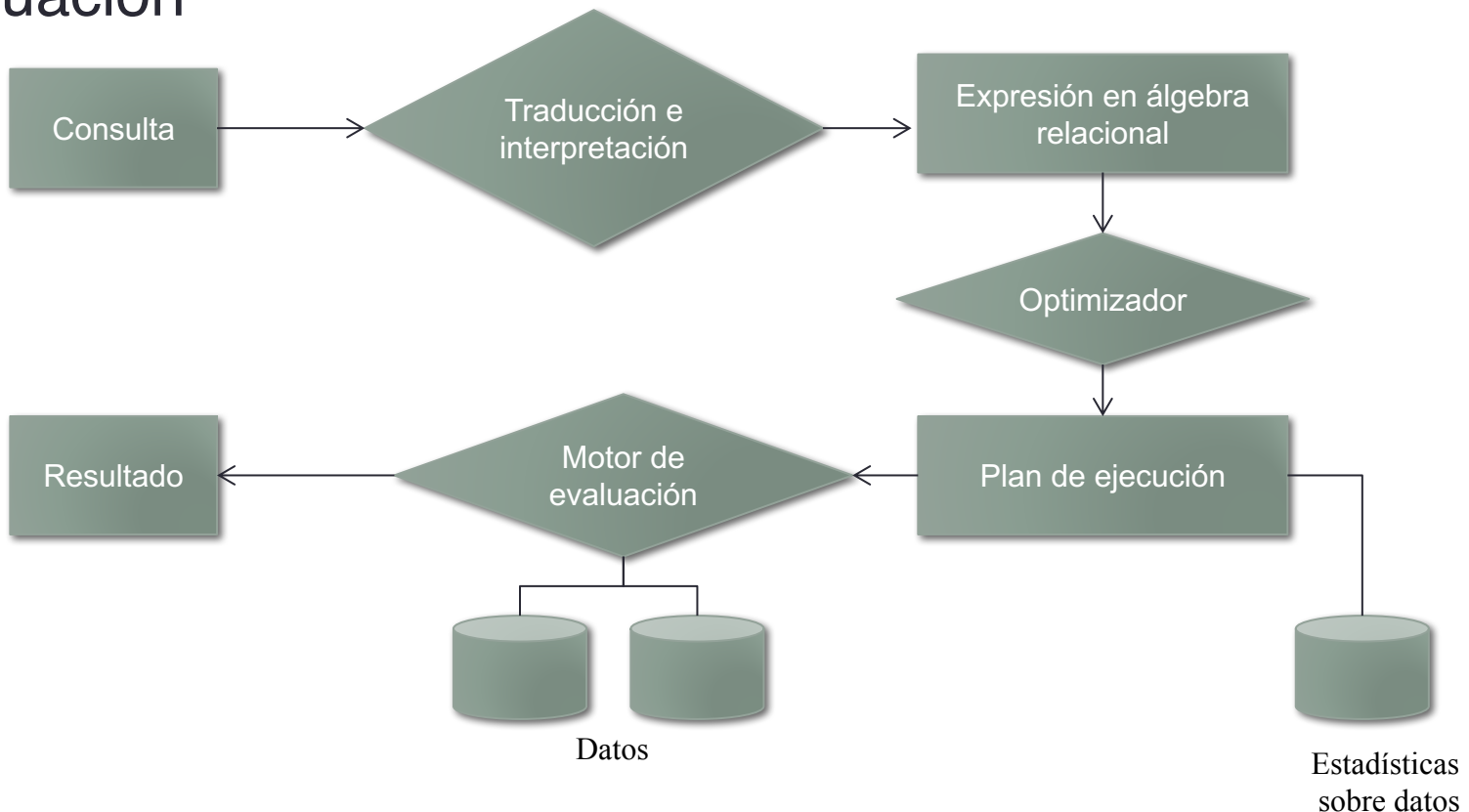
- Los programas de aplicación normalmente acceden a bases de datos mediante
  - Extensiones de lenguaje que permiten embeber SQL
  - Interfaces de programación de aplicaciones (p.e. ODBC/JDBC) que permiten enviar consultas SQL a una base de datos

# Gestión de almacenamiento

- El gestor de almacenamiento es un módulo que proporciona el interfaz entre los datos de bajo nivel almacenados en la base de datos y los programas de aplicación y consultas enviadas al sistema.
- El gestor de almacenamiento se encarga de:
  - Interactuar con el gestor de ficheros
  - Conseguir un almacenamiento, recuperación y actualización eficientes de los datos
- Aspectos clave:
  - Acceso a los dispositivos de almacenamiento físico
  - Organización de ficheros
  - Indexado y asociación (*hashing*)

# Procesado de consultas

1. Traducción e interpretación
2. Optimización
3. Evaluación



# Gestión de transacciones

- Una *transacción* es un conjunto de operaciones que realizan una única función lógica en una aplicación de bases de datos
- El componente de **gestión de transacciones** asegura que la base de datos se mantiene en un estado consistente (correcto) aun cuando se produzcan fallos en el sistema (p.e. fallos de alimentación o del sistema operativo) o fallos de transacción.
- El **gestor de control de concurrencia** controla la interacción entre transacciones concurrentes, para asegurar la consistencia de la base de datos.

# Usuarios de bases de datos

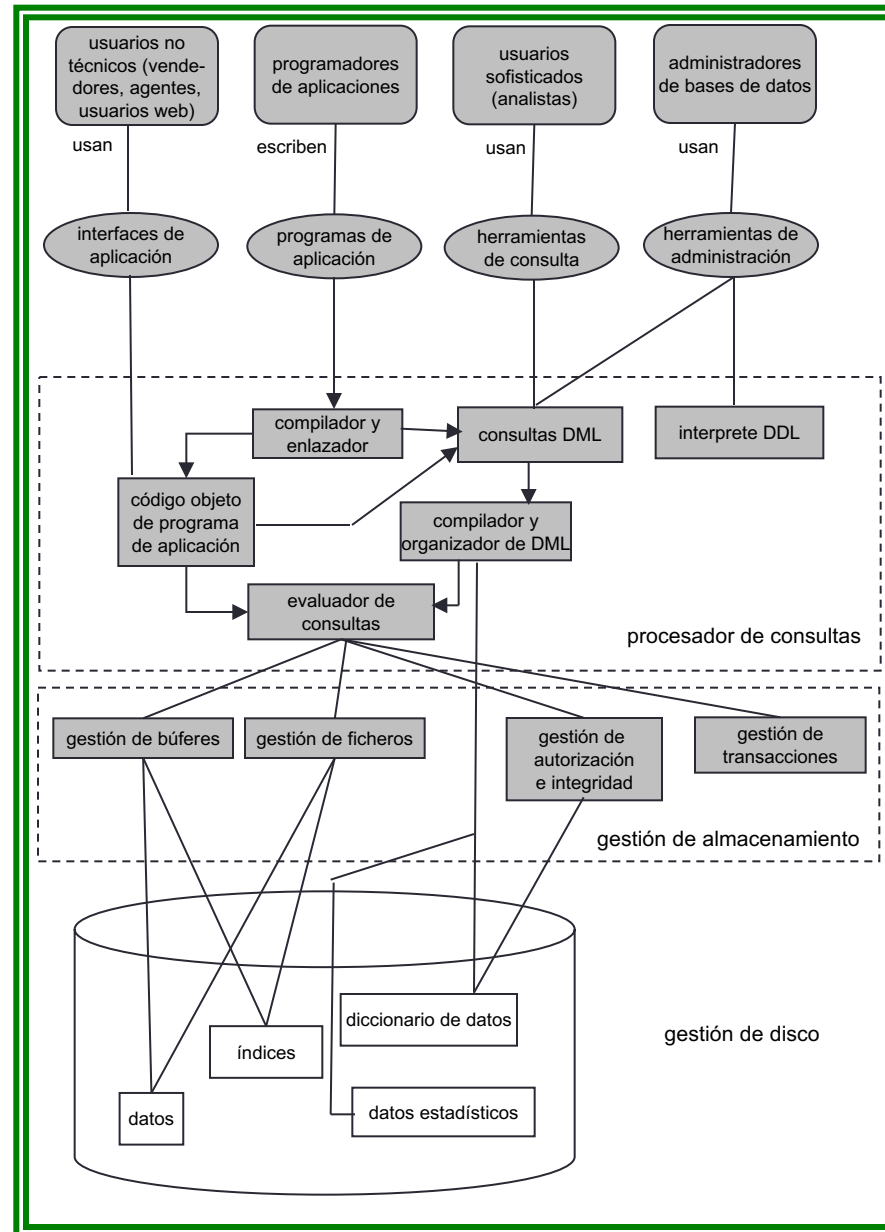
- Los usuarios se diferencian por la forma en que esperan interactuar con el sistema
  - Programadores de aplicaciones – realizan aplicaciones para los usuarios finales que interactúan con el sistema a través de llamadas DML
  - Usuarios sofisticados – hacen peticiones en un lenguaje de consulta de bases de datos (DML directo)
  - Usuarios especializados – programan y manipulan los esquemas de bases de datos
  - Usuarios no técnicos – llaman a uno de los programas de aplicación que se han escrito previamente
    - P.e. gente accediendo a la base de datos a través del web, cajeros de banco, personal de oficina

# Administrador de bases de datos

- Coordina todas las actividades del sistema de bases de datos; el administrador de bases de datos tiene un buen conocimiento de los recursos y necesidades de información de la organización.
- Las funciones del administrador de bases de datos incluyen:
  - Definición de esquemas
  - Definición de métodos de acceso y estructuras de almacenamiento
  - Modificación de esquemas y organización física
  - Garantizar derechos de acceso a la base de datos
  - Especificar restricciones de integridad
  - Actuar como enlace con los usuarios
  - Monitorizar las prestaciones y responder a cambios en los requisitos



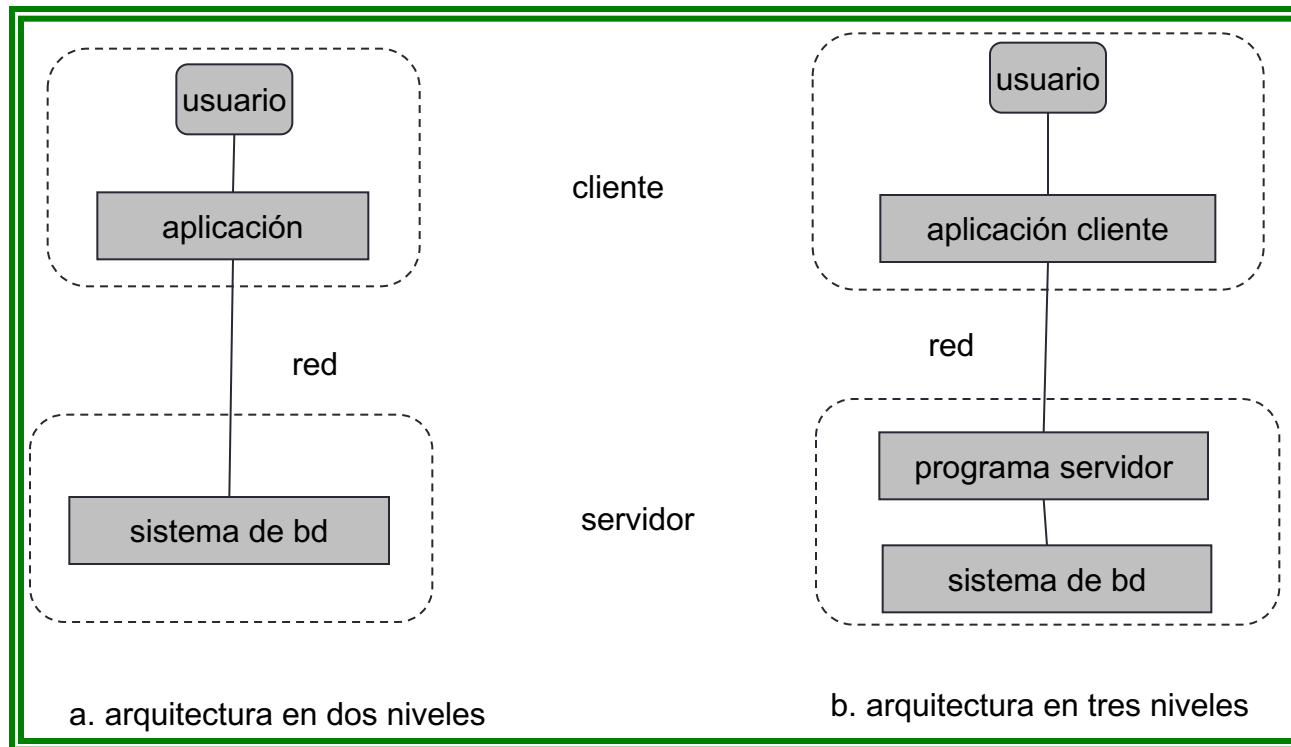
# SGBD



# SGBD. Clasificación

- Según modelo de datos
  - **relacional**
  - jerárquico
  - en red
  - objeto-relacional
  - orientado a objetos.
- Según número de usuarios
  - monousuarios
  - multiusuario
- Según arquitectura del sistema que lo soporta
  - Centralizado
  - Cliente-servidor
  - Paralelo
  - Distribuido

# Arquitecturas de aplicación



- **Arquitectura en dos niveles:** P.e. los programas clientes utilizan ODBC/JDBC para comunicarse con la base de datos
- **Arquitectura en tres niveles:** P.e. las aplicaciones basadas en web, o las aplicaciones que utilizan “middleware”

# Historia de los sistemas de Bases de Datos

- 1950s y principios de los 1960s:
  - Procesamiento de datos con almacenamiento en cinta magnética
    - Las cintas sólo permiten acceso secuencial
  - Entrada mediante tarjetas perforadas
- Finales de los 1960s y 1970s:
  - Discos duro con acceso directo a datos
  - Utilización generalizada de los modelos jerárquico y de red
  - Ted Codd define el modelo de datos relacional:
    - IBM comienza a desarrollar el prototipo del sistema R
    - La Universidad de Berkeley comienza el prototipo de Ingres
  - Procesamiento de transacciones de altas prestaciones (para la época)

# Historia de los sistemas de Bases de Datos

- 1980s:
  - Los prototipos relacionales se transforman en sistemas comerciales
    - SQL se convierte en el estándar de la industria
  - Sistemas de bases de datos paralelos y distribuidos
  - Sistemas de bases de datos orientados a objetos
- 1990s:
  - Aplicaciones complejas para el soporte a la toma de decisiones y minería de datos
  - Grandes *Data* warehouses de varios terabytes
  - Aparición del comercio Web
- Principios de 2000s:
  - Estándares XML y Xquery
  - Administración automatizada de bases de datos
- Finales de 2000s:
  - Sistemas de almacenamiento gigantescos
    - Google BigTable, Yahoo PNuts, Amazon, ..

# Tendencias de los Sistemas de Información

- ⇒ Sistemas para el soporte para toma de decisiones (Data Warehouse) y/o con capacidades deductivas y bases de datos temporales
- ⇒ Manejar información multimedia: Imágenes, audio, videos
- ⇒ Bases de datos orientadas a objetos y objeto-relacionales
- ⇒ Manejar información georeferenciada: Sistemas de Información Geográficos (SIG)
- ⇒ Manejo de información documental: Motores de búsqueda, sistemas para el manejo de documentos.
- ⇒ Bases de datos nativas para XML
- ⇒ Tratamiento semántico de la información

# ÍNDICE

- Introducción
- Definición de base de datos
- Sistemas Gestores de Bases de Datos (SGBD)
- Conceptos básicos
- Diseño de bases de datos
- El modelo Entidad-Asociación (E-A)
- El modelo lógico relacional
- El lenguaje SQL

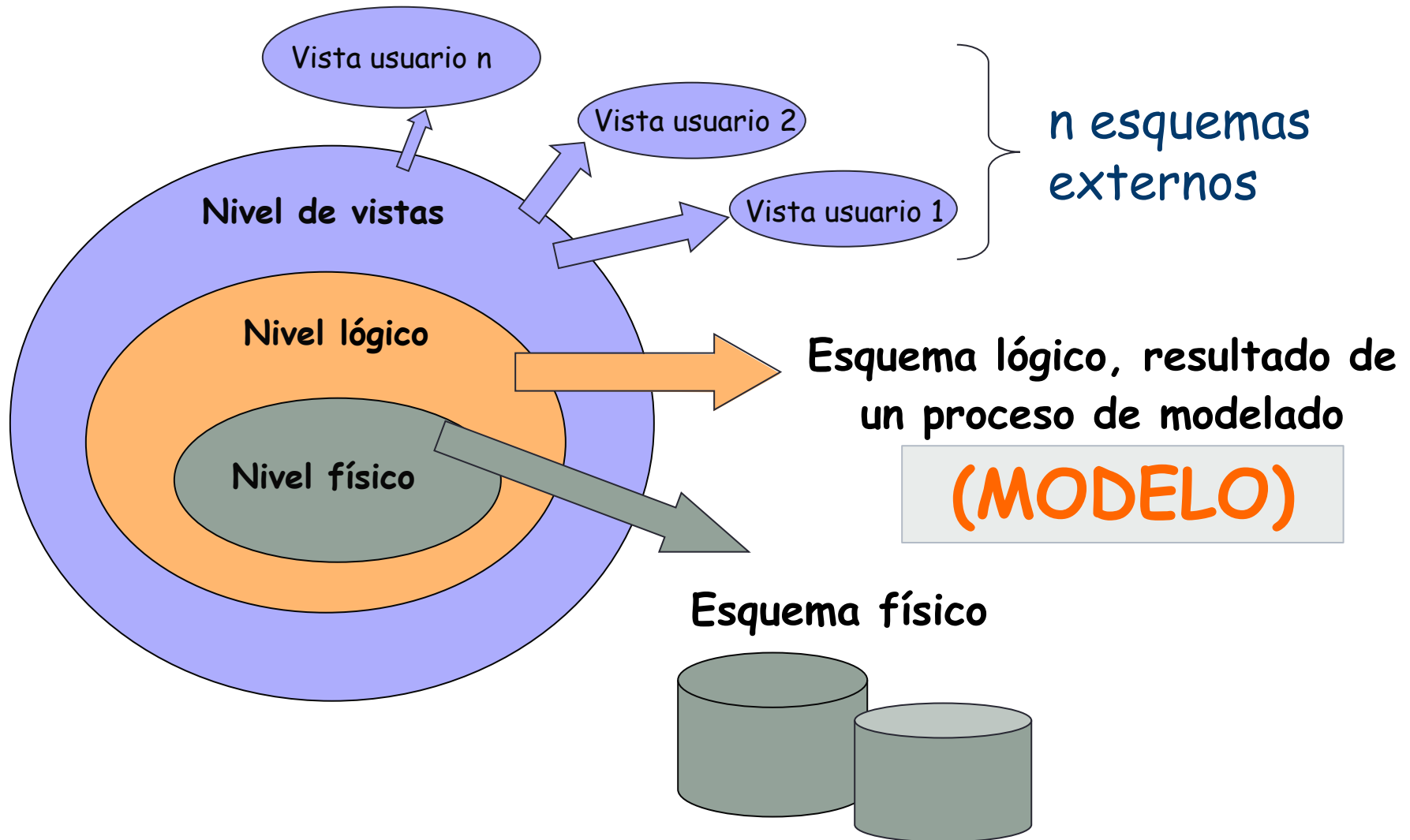
# Metodología de diseño de bases de datos

Una metodología de diseño debería tener las siguientes características:

- **Claridad y comprensibilidad**, ya que en el diseño participan distintas clases de usuarios y la metodología debe ser comprensible para todos ellos.
- **Capacidad de soportar la evolución de los sistemas**, es decir debe soportar los cambios de los sistemas sin tener que cambiar todo el diseño de base de datos.
- **Facilitar la portabilidad**, es decir cambiar de un sistema informático a otro sin mayores inconvenientes.
- **Versatilidad respecto a los tipos de aplicaciones**, ya que no debe estar orientada a un tipo de aplicaciones en concreto sino que puede utilizarse en aplicaciones diversas.
- **Flexibilidad** (independencia respecto de la dimensión de los proyectos), ya que debe servir tanto para proyectos grandes como pequeños.
- **Adopción de estándares**, aplicar en lo posible estándares internacionales para elaborar productos con altos niveles de calidad.



# Niveles de abstracción



# Modelos de datos

- Una colección de herramientas para describir:
  - datos
  - relaciones entre datos
  - semántica de los datos
  - restricciones de los datos
- **Modelo relacional**
- Otros modelos:
  - Modelos basados en objetos:
    - Orientado a objetos
    - Objeto-relacional
  - Modelos de datos semiestructurados
  - Modelos antiguos: modelo de red y modelo jerárquico
- Modelo Entidad-Asociación o Entidad-Relación (para diseño)

## Diseño de bases de datos

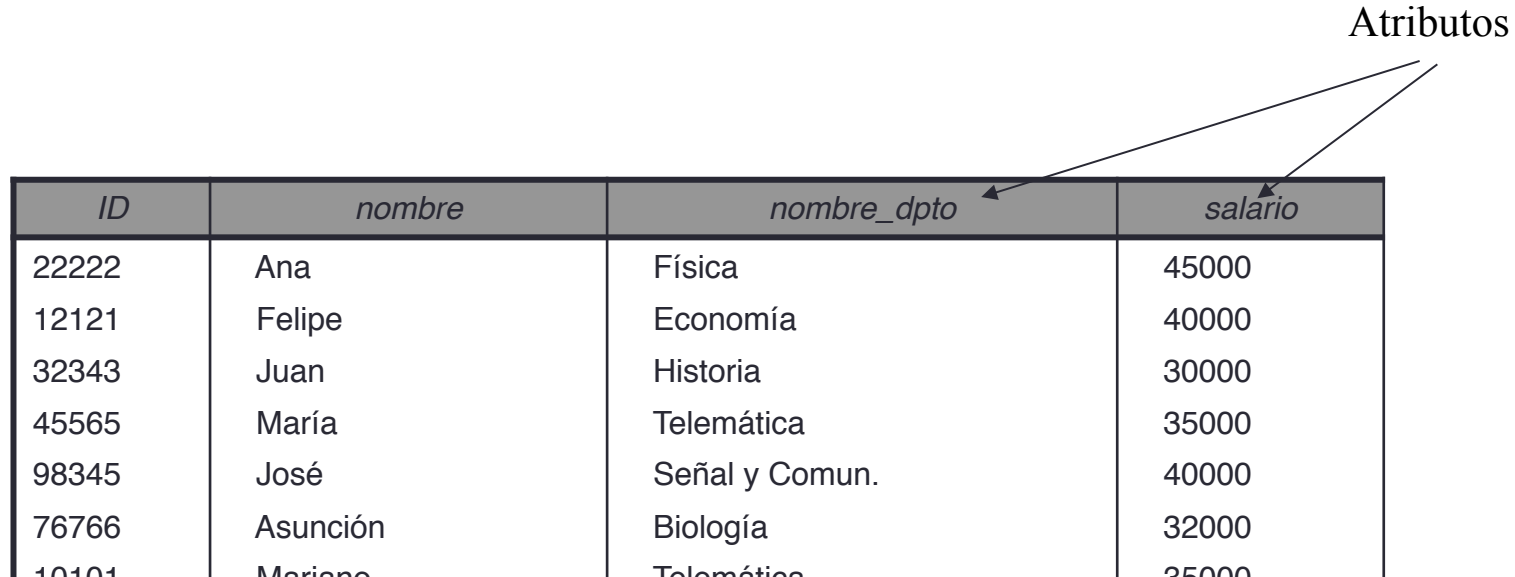
- Proceso para diseñar la estructura general de la base de datos
  - Diseño Lógico: Decisión del esquema lógico de la base de datos. Es decir, encontrar un conjunto “adecuado” de esquemas de relación
    - Decisiones de negocio: ¿qué atributos debería almacenar en la base de datos?
    - Decisiones informáticas: ¿qué esquemas de relación utilizar y cómo distribuir entre ellos los atributos?
  - Diseño físico: Escribir el programa.

# Modelo de datos relacional

- Fue introducido por Codd en 1970. Todo el modelo tiene un fuerte apartado matemático subyacente.
- Una Base de Datos en el modelo relacional es un conjunto de relaciones.
- Relación:
  - información relativos a un conjunto de elementos homogéneos.
  - dividida en filas y columnas.
    - Las filas las consideraremos registros o tuplas.
    - Las columnas las consideraremos campos o atributos.
    - Siempre debe haber una clave para identificar de forma única a los registros.

# Modelo Relacional

- Ejemplo de datos tabulares en el modelo relacional



<i>ID</i>	<i>nombre</i>	<i>nombre_dpto</i>	<i>salario</i>
22222	Ana	Física	45000
12121	Felipe	Economía	40000
32343	Juan	Historia	30000
45565	María	Telemática	35000
98345	José	Señal y Comun.	40000
76766	Asunción	Biología	32000
10101	Mariano	Telemática	35000
58583	Loreto	Historia	50000
83821	Pedro	Telemática	47000
15151	Luis	Música	33000
33456	Paula	Física	40000
76543	Lucía	Economía	42000

(a) Relación *profesores*

# Un ejemplo de base de datos relacional

<i>ID</i>	<i>nombre</i>	<i>salario</i>
22222	Ana	45000
12121	Felipe	40000
32343	Juan	30000
45565	María	35000
98345	José	40000
76766	Asunción	32000
10101	Mariano	35000

(a) Relación *profesores*

<i>nombre_dpto</i>	<i>edificio</i>	<i>presup.</i>
Física	Fac. Ciencias	90000
Economía	Fac. Económicas	120000
Historia	Fac. Humanidades	40000
Telemática	Esc. Teleco	230000
Señal y Comun.	Esc. Teleco	180000
Biología	Fac. Ciencias	95000
Música	Fac. Humanidades	12000

(b) Relación *departamentos*

<i>ID-profesor</i>	<i>Nombre_dpto</i>
22222	Física
12121	Economía
32343	Historia
45565	Telemática
98345	Señal y Comun.
76766	Biología
10101	Telemática

(c) Relación *pertenece*

## Diseño de bases de datos

Este diseño ¿tiene algún problema?

<i>ID</i>	<i>nombre</i>	<i>salario</i>	<i>nombre_dpto</i>	<i>edificio</i>	<i>presupuesto</i>
22222	Ana	45000	Física	Fac. Ciencias	90000
12121	Felipe	40000	Economía	Fac. Económicas	120000
32343	Juan	30000	Historia	Fac. Humanidades	40000
45565	María	35000	Telemática	Esc. Teleco	230000
98345	José	40000	Señal y Comun.	Esc. Teleco	180000
76766	Asunción	32000	Biología	Fac. Ciencias	95000
10101	Mariano	35000	Telemática	Esc. Teleco	230000
58583	Loreto	50000	Historia	Fac. Humanidades	40000
83821	Pedro	47000	Telemática	Esc. Teleco	230000
15151	Luis	33000	Música	Fac. Humanidades	12000
33456	Paula	40000	Física	Fac. Ciencias	90000
76543	Lucía	42000	Economía	Fac. Económicas	120000

(a) Tabla *profesores*

# Herramientas para un buen de diseño

- Modelo E-A del mundo real
  - Entidades (objetos)
    - P.e. profesores, asignaturas, departamentos
  - Asociaciones entre entidades
    - P.e. El profesor Pedro imparte la asignatura SSOO.
    - La asociación *impartir* asocia profesores con asignaturas
  - Ampliamente utilizado para diseño conceptual de bases de datos
- Teoría de la normalización:
  - Formaliza qué diseños presentan problemas y los identifica.



# Pasos de diseño

- Especificación de requisitos
- Diseño conceptual
- Diseño lógico
- Normalización
- Diseño físico

# Especificación de requisitos

- Especificación de requisitos de usuario
  - Describe en detalle los datos que hay que almacenar en la base de datos y cómo se deben utilizar
- Especificación de requisitos del sistema
  - Describe las características que debe incluir el sistema: prestaciones, seguridad, etc.

# Especificación de requisitos de usuario (ejemplo)

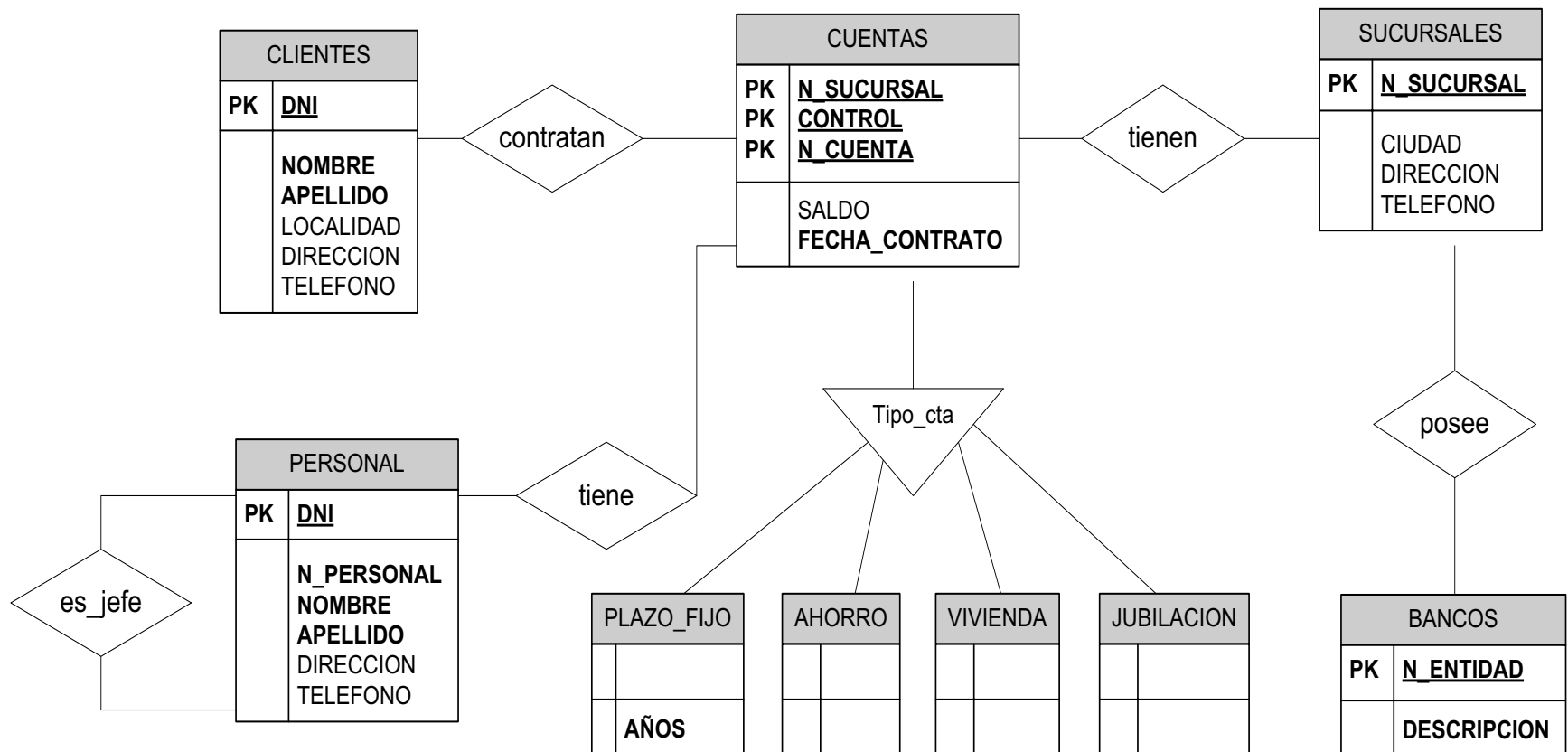
Se trata de gestionar los datos de un banco que maneja estos datos:

- Posee clientes que contratan cuentas en el banco. De los clientes tenemos su DNI, Nombre, Apellidos, Localidad, Dirección y Teléfono.
- Todas las cuentas tienen un número de 20 dígitos de los cuatro primeros se refieren al banco, los cuatro siguientes a la entidad, los dos siguientes son de control y los 10 siguientes el número en sí de la cuenta. Toda cuenta tiene un saldo actual y se anota la fecha de contratación
- Las cuentas pueden ser de ahorro, de plazo fijo (indicando el tiempo mínimo de supervivencia en años (1,2,3,...), cuentas vivienda y cuentas de pensiones (jubilación).
- Del personal se anotan los siguientes datos, DNI, Nº personal, Nombre, Apellidos, Dirección, Teléfono y una Cuenta bancaria (que tiene que estar en el banco)
- El personal puede tener o ser jefe. Un jefe puede tener un jefe superior.

# Diseño conceptual

- El propósito de ésta fase es obtener una buena representación de los recursos de información, con independencia de usuarios o aplicaciones (SGBD), y sin realizar consideraciones sobre la eficiencia.
- Se utiliza el modelo Entidad-Asociación (E-A)
  - Resultado: **Diagrama E-A**

# Diagrama E-A. Ejemplo



# Diseño lógico

- Consiste en transformar el esquema conceptual obtenido en la etapa anterior, adaptándolo a una representación de relaciones, apoyándose para ello en el modelo relacional.

# Diseño lógico

## PROFESOR

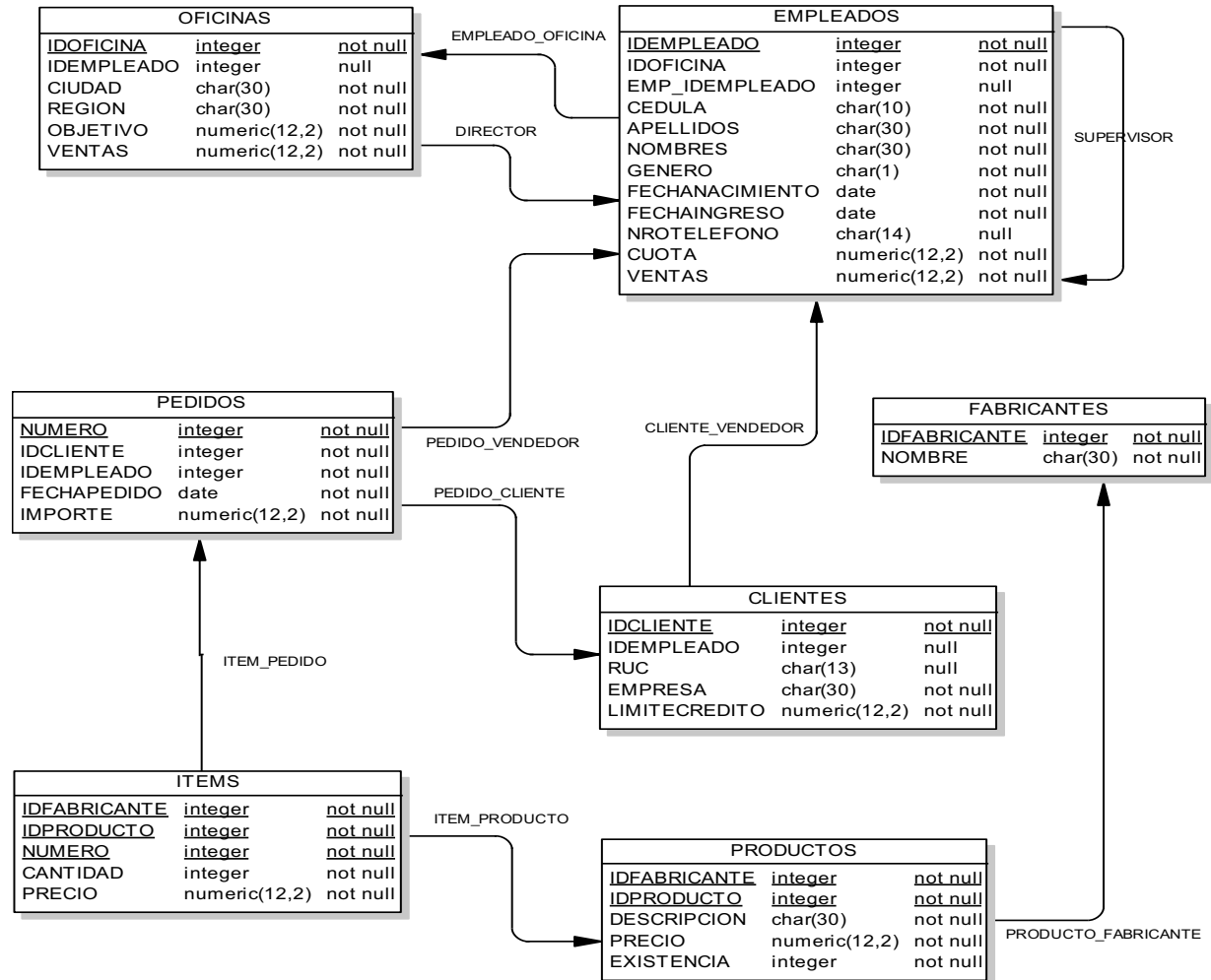
<i>Cod prof</i>	<i>Nombre</i>	<i>DNI</i>	<i>Dirección</i>	<i>Teléfono</i>	<i>Materia</i>
00001	Juan	12223433	Rios Rosas, 23	670123123	Ing. Software
00002	Coral	54656754	Alarcos, 8	567983456	Bases de datos
00003	Belén	53567523	Getafe, 4	6º9267854	Orientación objetos
00004	Goyo	97856757	Pez, 102	679345763	Sistemas operativos
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
03568	Roberto	34534522	Fundación, 10	639456239	Redes



CLAVE PRIMARIA

# Diseño lógico

El resultado  
es el  
esquema  
relacional





# Normalización

- Una vez creadas las relaciones, éstas deben pasar por el proceso de normalización:
  - Método formal para identificar anomalías de diseño en las relaciones que forman nuestro diseño de bases de datos
  - Se definen una serie de **Formas Normales**:
    - Conjuntos de condiciones que deben cumplir las relaciones para no tener anomalías
    - Cuantas más cumpla una relación, mejor será su diseño.

# Diseño físico

- Es el proceso de conseguir una implementación, lo más eficiente posible, del esquema lógico.
- Permite al diseñador tomar decisiones sobre cómo hay que implementar la base de datos, por tanto el diseño físico está adaptado a un SGBD concreto.
- Se introduce directamente en el SGBD
- Lenguaje más utilizado: SQL

# Diseño físico

→ TEMA (Cód\_tema, ...)  
CONSTA (Cód\_tema, Cod\_tema\_sup...)

```
CREATE TABLE Consta (  
    Cod_Tema Codigos,  
    Cod_Tema Sup Codigos,  
    PRIMARY KEY (Cod_tema)  
    FOREIGN KEY (Cod_Tema) REFERENCES Tema  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    FOREIGN KEY (Cod_Tema_SUP) REFERENCES Tema  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

# FIN DEL TEMA 1

---