



Universidad de Granada

[decsai.ugr.es](http://decsai.ugr.es)

# **Fundamentos de Bases de Datos**

Grado en Ingeniería Informática

## **Tema 3: Modelos de datos**



**DECSAI**

**Departamento de Ciencias de la  
Computación e Inteligencia Artificial**

1. **Modelos de datos**
2. **Modelo Jerárquico**
3. **Modelo en Red**
4. **Modelo Relacional**



- 1. Modelos de datos**
2. Modelo Jerárquico
3. Modelo en Red
4. Modelo Relacional



### Proceso de transformación

#### a) Mundo real

- Delimitación **objetivos**
- **Selección** de **datos**
- Hipótesis **semánticas**
- **Organización** de los **datos** a almacenar

#### b) Esquema Inicial

- **Datos operativos**
- **Atributos**
- **Conexiones**
- **Restricciones**

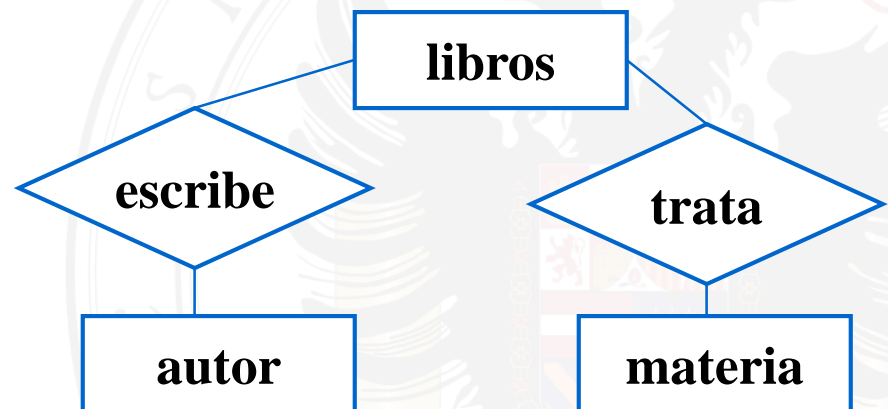
### Problema tipo

#### a) Biblioteca



#### b) Resultado

- Libros: título, isbn, editorial....
- Autor: nombre, nacionalidad,....
- Materia: código, descripción....



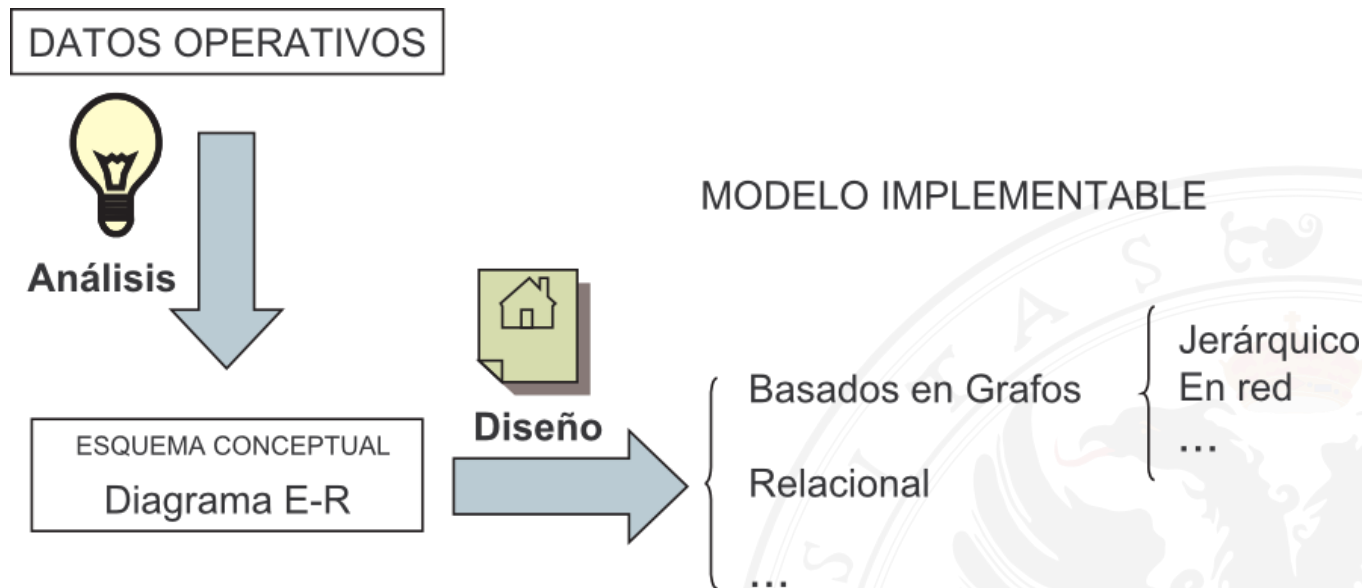
### Definición formal (de modelo de datos)

- Mecanismo formal para representar y manipular información de manera general y sistemática.
- Debe constar de:
  - 1.- Notación para describir datos.
  - 2.- Notación para describir operaciones.
  - 3.- Notación para describir reglas de integridad.

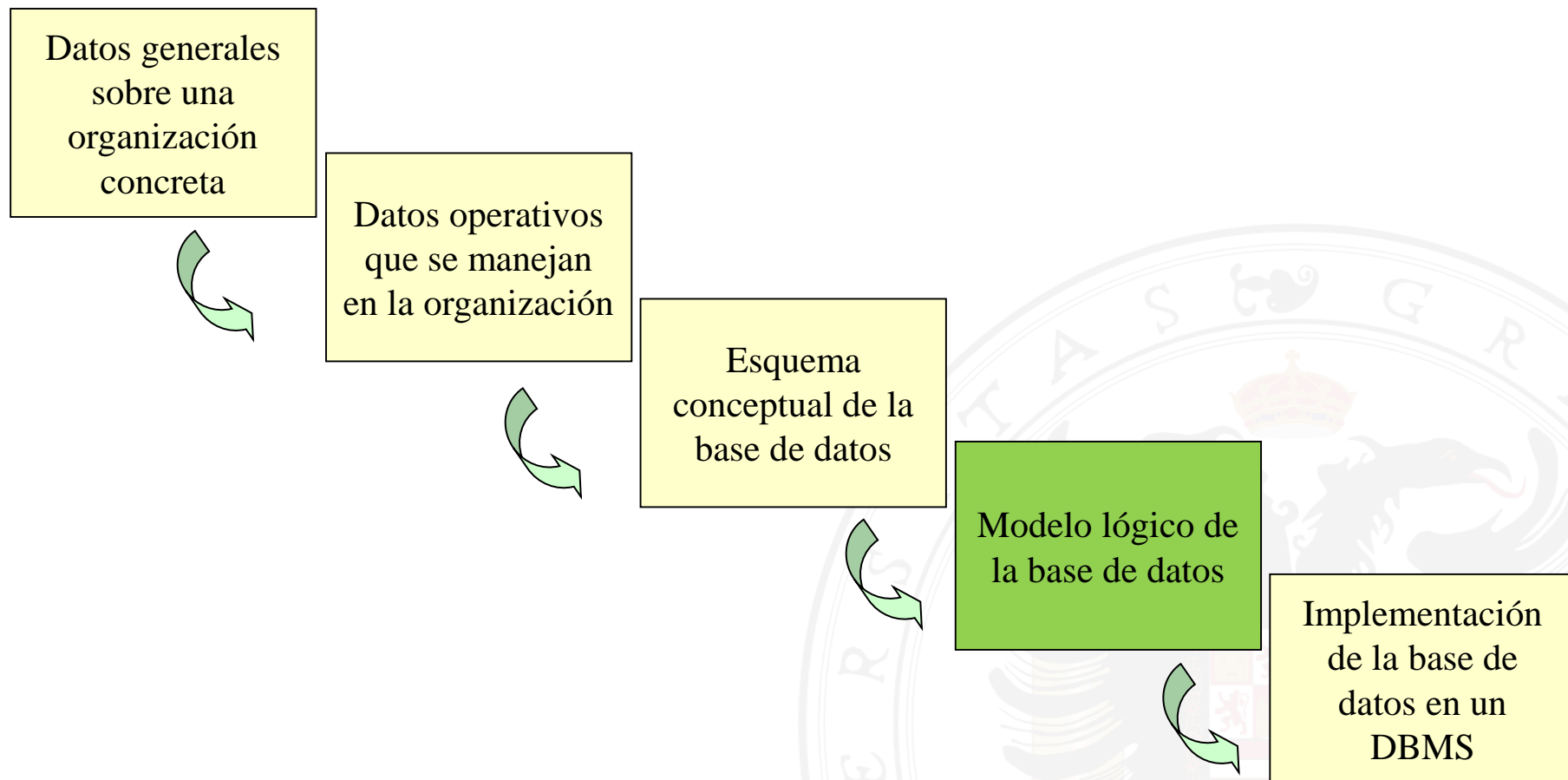
### Historia

- 1<sup>er</sup> Modelo es el relacional (Codd).
- Se recuperan los modelos basados en grafos (1974).
- El modelo E/R (Chen, 1975), otros modelos semánticos.
- Modelos orientados a objetos (1983, 1986,...).
- Modelos lógicos (1986...).

### Proceso de análisis y diseño de una BD:



- Una vez hemos llevado a cabo el proceso de análisis de datos y **obtenido el esquema conceptual** o lógico de nuestra BD, es necesario implantarla en un sistema a través de un proceso de DISEÑO, que nos permitirá **trasladar** la estructura actual **a un modelo de datos implementable**.



- **Modelado lógico:** Trasladamos a un **esquema lógico** en función de una estructura implementable.

<b>Título</b>	<b>ISBN</b>	<b>Editorial</b>	<b>...</b>
<b>Introducción a las BD</b>	<b>1234-1234</b>	<b>Thomson</b>	<b>...</b>
<b>Cálculo para todos</b>	<b>4321-4321</b>	<b>Delta</b>	<b>...</b>
<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>

- Tabla LIBROS

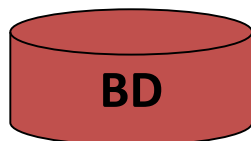
Datos generales sobre una organización concreta

Datos operativos que se manejan en la organización

Esquema conceptual de la base de datos

Modelo lógico de la base de datos

Implementación de la base de datos en un DBMS





Datos generales sobre una organización concreta

Datos operativos que se manejan en la organización

Esquema conceptual de la base de datos

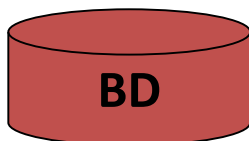
Modelo lógico de la base de datos

Implementación de la base de datos en un DBMS

Etapas de la creación de una BD

- Lo implementamos en un sistema comercial:

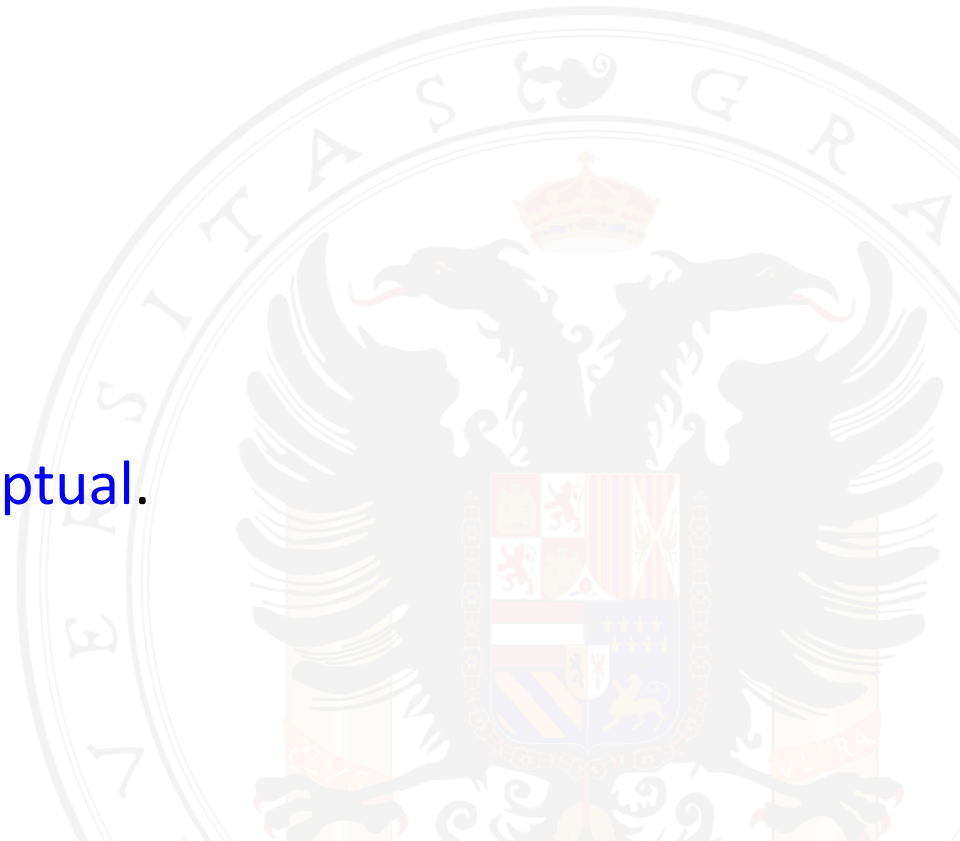
```
CREATE TABLE LIBROS (
    titulo char(45) NOT NULL,
    ISBN char(10) PRIMARY KEY,
    editorial char(30) REFERENCES ...
);
```



- Necesidad de modelos de datos:
  - Cada **esquema** se describe utilizando un lenguaje de definición de datos.
  - Este lenguaje es de muy **bajo nivel**, está muy ligado al SGBD.
  - Hacen falta otros **mecanismos de más alto nivel** que permitan **describir los datos** de una forma no ambigua y **entendible** por los **usuarios** implicados **en cada paso** del **proceso** de **implantación**.

- **Objetivo:**
  - Describir modelos que representen los datos y que los describan de una forma entendible y manipulable.
  - En relación con la Arquitectura ANSI/SPARC:
    - Nivel Externo:
      - » Modelo de datos externo.
    - Nivel Conceptual:
      - » Modelo de datos conceptual.
    - Nivel Interno:
      - » Modelo de datos interno.

- Clasificación:
  - Basados en registros.
  - Basados en objetos.
  - Físicos.
  
- Utilización:
  - Los dos primeros:
    - Nivel externo y conceptual.
  - Físicos:
    - Nivel interno.



- Modelos de datos **basados en registros**:
  - Modelo de datos **jerárquico**.
  - Modelo de datos **en red**.
  - Modelo de datos **relacional** (Codd, 1969).

1. Modelos de datos
2. **Modelo Jerárquico**
3. Modelo en Red
4. Modelo Relacional



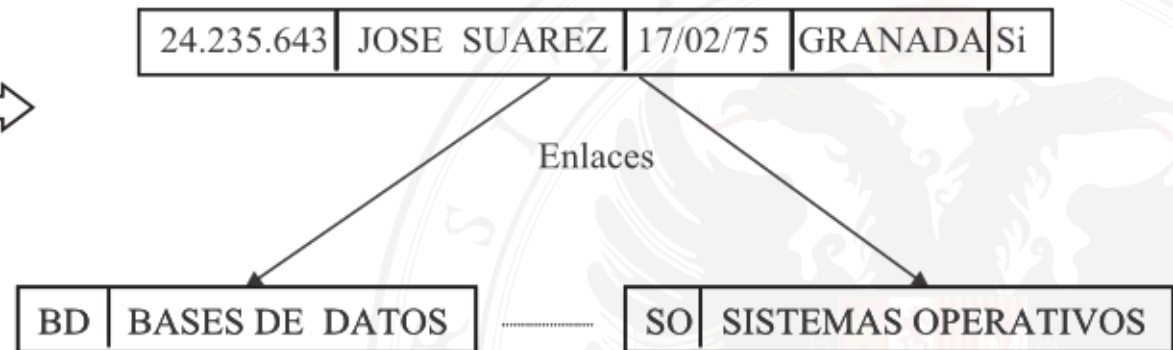
- Fue el **primero** en **implementarse** físicamente:
  - Nivel **externo**: **aplicaciones Cobol**.
  - No había interactividad:
    - **Carecía** de un **lenguaje de consulta**.
- **Estructura de datos** básica:
  - **Árbol**
    - Registro **maestro**.
    - Registros **secundarios**.
- La **BD** es una **colección** de **instancias de árboles**.

### Modelo jerárquico

#### Esquemas



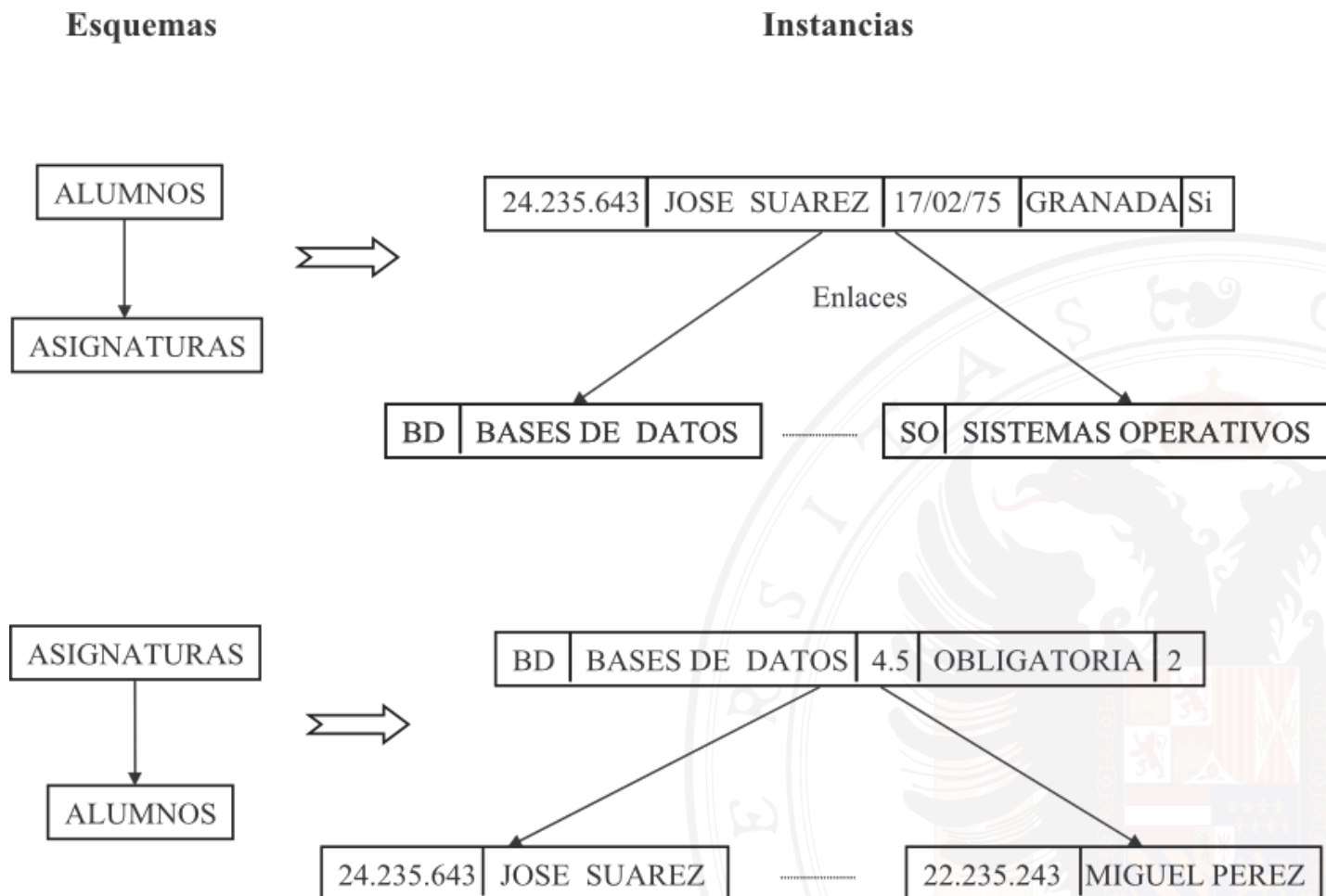
#### Instancias





- Esta estructura plasma de forma muy directa:
  - Relaciones muchos a uno.
  - Relaciones uno a uno.
- Relaciones muchos a muchos:
  - Hay que duplicar toda la información sobre las entidades involucradas.

### Modelo jerárquico



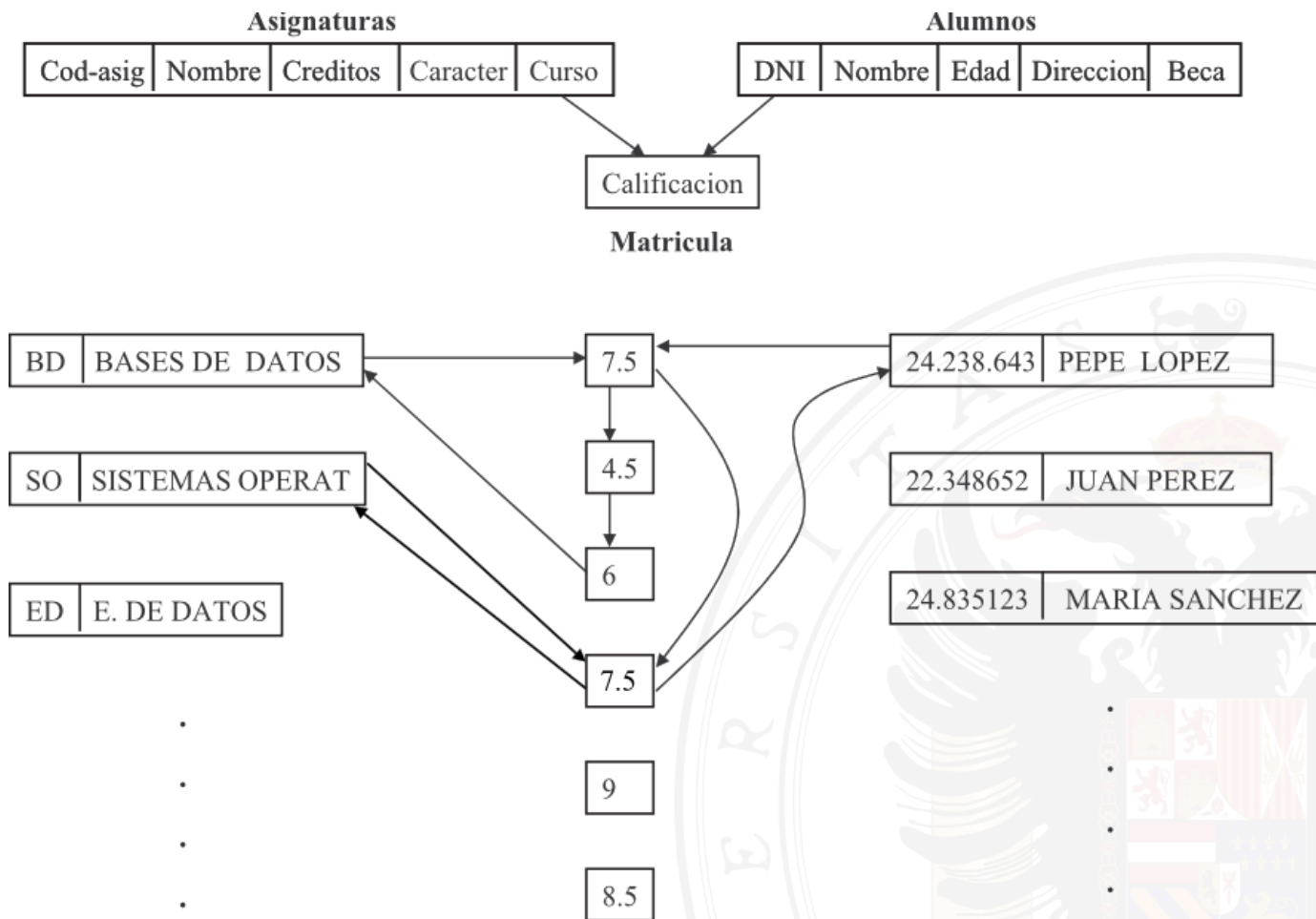
- Inconvenientes:
  - Almacenar árboles en ficheros es complejo:
    - Varios tipos de registros.
    - Punteros que hay que mantener.
  - DML difícil de:
    - Implementar.
    - Usar.
  - Dependencia existencial obligatoria de los registros de tipo secundario con respecto a los de tipo raíz:
    - No se podrá insertar un registro de tipo secundario mientras no exista uno de tipo raíz con el que *enlazar*.
  - Redundancia necesaria para plasmar relaciones muchos a muchos.
    - La integridad de los datos es costosa de mantener.

1. Modelos de datos
2. Modelo Jerárquico
3. **Modelo en Red**
4. Modelo Relacional



- Estructura de datos:
  - **Grafos** cuya topología depende de las conexiones existentes entre las entidades:
    - **Nodos**: registros.
    - **Arcos**: enlaces entre registros (punteros).
    - **Relaciones entre conjuntos de entidades**:
      - » **Conectores**: registros especiales (atributos propios de la relación).
      - » **Cada ocurrencia** de un **conector** representa una asociación distinta.
  - **Cualquier registro** puede relacionarse **con cualquier registro**.
- Base de datos:
  - **Colección de instancias de grafos**.
- La estructura es **muy genérica**:
  - Permite plasmar **todo tipo de relaciones**.
  - Implementa directamente las relaciones **muchos a muchos**.

### Modelo en red



## Modelo en Red

- Ventajas:
  - Estructura algo más homogénea.
  - Permite insertar nuevas entidades en un conjunto de forma independiente.
- Problemas:
  - La existencia de enlaces entre los registros hace que las operaciones del DDL y el DML sigan siendo complejas de implementar y utilizar.

1. Modelos de datos
2. Modelo Jerárquico
3. Modelo en Red
4. **Modelo Relacional**





- El modelo de datos relacional organiza y **representa los datos en forma de tablas o relaciones.**
- **Base de datos relacional: colección de tablas cada una de las cuales tiene un nombre único.**

### Modelo Relacional

Atributo

Id_trabajador	Nombre	Tarifa_hr	Tipo_de_oficio	Id_supv
1235	M. López	12,50	Electricista	1311
1412	J.L. Calvo	13,75	Fontanero	1520
2920	N. Marín	10,00	Carpintero	Nulo
3231	O. Pons	17,40	Albañil	Nulo
1540	M.A. Vila	11,75	Fontanero	Nulo
1311	J.C. Cubero	15,50	Electricista	Nulo
3001	D. Sánchez	8,20	Albañil	3231

Dominio

Tupla

Figura 1: Tabla Trabajadores

- Algunos **conceptos**:
  - **Esquema** de una base de datos relacional:
    - Colección de **esquemas de relaciones** junto con las restricciones de integridad.
  - **Instancia** o estado de una base de datos:
    - Colección de **instancias de relaciones** que **verifican** las restricciones de integridad.
  - **Base de datos** relacional:
    - Instancia de una **base de datos** junto con su **esquema**.

- Por ejemplo, podemos considerar la siguiente base de datos:

Trabajadores(id\_trabajador,nombre,trf\_hr,tipo\_de\_oficio,id\_supv)  
Edificios(id\_edificio, dir\_edificio, tipo, nivel\_calidad, categoria)  
Asignaciones(id\_trabajador, id\_edificio, fecha\_inicio, num\_dias)  
Oficios(tipo\_de\_oficio, prima, horas\_por\_sem)

Donde se representa cada tabla mediante su esquema,  
expresado de la forma: Tabla(Atributo1,Atributo2,....)

- Claves
  - **Superclave**: Cualquier conjunto de atributos que identifica unívocamente a cada **tupla** de una relación.
  - **Clave Candidata**: superclave minimal.
    - Ejemplo:
      - Relación Trabajadores
        - » {id\_trabajador, nombre} – superclave
        - » No es minimal: no es clave de la relación
  - De entre las **candidatas** (si hubiera más de una), hay que **elegir una como principal** que se denomina **Clave primaria**.
    - Criterio de selección: Tamaño, significado, capacidad para recordarla, fusión con otras tablas,...

- Completamos la **notación** para describir una relación, **subrayando los atributos** que forman su **clave primaria**, si existen otras claves candidatas también se subrayan etiquetando el subrayado con CC:

Trabajador(id\_trabajador, nombre, trf\_hr, tipo\_de\_oficio, id\_supv)

Edificios(id\_edificio, dir\_edificio, tipo, nivel\_calidad, categoria)

Asignaciones(id\_trabajador, id\_edificio, fecha\_inicio, num\_dias)

Oficios(tipo de oficio, prima, horas\_por\_sem)

### Comparativa

- Con respecto a la representación

- Relacional

- Un sólo elemento para la representación (esencialidad).
    - Conexiones lógicas.
    - Representación relaciones n:m simétrica.
    - Identidad por valor.

- Con respecto a la consulta

- Relacional

- Consultas simétricas en jerarquías.
    - Obtención de la consulta como resultado global.



- Lenguajes declarativos.

- Basado en grafos

- Dos elementos para la representación.
    - Conexiones en el modelo físico subyacente.
    - Representación conexiones n:m: imposible en modelos jerárquicos, difícil en modelos en red.
    - Identidad por posición.

- Basado en grafos

- Consultas no simétricas en jerarquías.
    - Mecanismo de navegación por punteros.



- Lenguajes procedimentales.