

## **TEMA 2. DIAGRAMAS ENTIDAD / RELACIÓN Y DIAGRAMAS DE ESTRUCTURA DE DATOS**

**1. MODELO DE DATOS**

**2. DIAGRAMAS DE ESTRUCTURA DE DATOS**

**3. MODELO ENTIDAD / RELACIÓN EXTENDIDO**

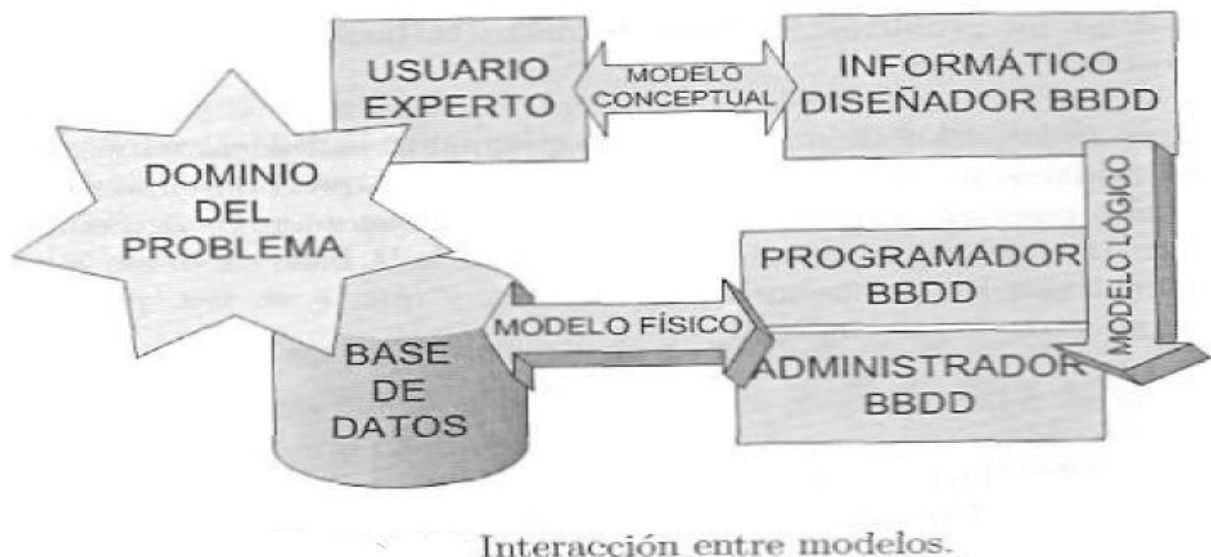
# 1. MODELO DE DATOS

Tal y como vimos en el tema anterior, para modelar un problema de base de datos se siguen los siguientes pasos:

**1. Diseño Conceptual.** Cuyo objetivo es obtener un **Modelo Conceptual** que nos de una representación de los recursos de información de la empresa, con independencia de los usuarios, así como del software o el hardware disponible. Para ello se utiliza un Modelo Entidad /Relación extendido.

**2. Diseño Lógico.** Cuyo objetivo es transformar el modelo conceptual anterior en un **Modelo Lógico** totalmente adaptado al tipo (en la actualidad modelo relacional, modelo orientado a objetos o modelo híbrido) de SGBD que se vaya a utilizar. En este tema nos centramos en el modelo relacional por lo tanto aprendemos a hacer Diagramas de Estructura de Datos o DED. Si el modelo es orientado a objetos o bien híbrido, tenemos que hacer Diagramas de Clases.

**3. Diseño Físico.** Su objetivo es conseguir un **Modelo Físico** de la Base de Datos adaptando el modelo lógico obtenido al SGBD propio de nuestra instalación.



## 2. DIAGRAMAS DE ESTRUCTURA DE DATOS (DED)

Podemos **definir un DED** como una descripción de la relación entre entidades (personas, lugares, eventos y objetos) de un sistema y el conjunto de información relacionado con cada entidad.

El **objetivo principal** de un DED es **comunicar los requerimientos de datos** (un modelo de datos lógico) a un **administrador** de la base de datos **para que pueda crear la base de datos** con sus correspondientes tablas.

### CONCEPTOS

#### ENTIDAD

- Una **entidad** es un concepto abstracto o un objeto real que tiene existencia por sí mismo y se puede identificar de manera clara y precisa.

#### OCURRENCIAS

- Dentro de una entidad encontraremos distintas **ocurrencias** que dan lugar a distintos objetos.

#### PROPIEDADES O ATRIBUTOS

- Cada uno de los objetos de una entidad tendrán características, **propiedades o atributos** comunes cuyas ocurrencias serán los distintos valores que puedan tomar.

#### CLAVE PRINCIPAL O PRIMARIA

- Dentro de las propiedades habrá una que identifique de forma unívoca al objeto (no se puede repetir), que además no podrá tener nunca un valor nulo. Esta propiedad se llamará **identificativo (identificador) , clave principal o primaria**.
- *Regla de integridad de la entidad (defin. Informal): Una clave primaria no debe contener valores nulos.*

#### CLAVE AJENA O FORANEA

- Entre las propiedades también podemos encontrar **claves ajenas o foraneas**, estas **se utilizan para establecer relaciones entre entidades**.
- Una clave ajena en una entidad será primaria en la entidad con la que se relaciona.

- Una clave ajena puede contener valores nulos y sus valores se pueden repetir.
- Regla de integridad referencial (defin. informal): No debe existir un valor de clave foránea sin concordancia (con su correspondiente clave primaria).

## RELACIÓN

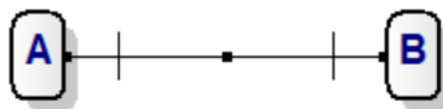
- Una **relación** es una asociación entre entidades.
- **Tipo de relación según el grado:** Las relaciones pueden ser en teoría de distinto tipo; **reflexiva** (de la entidad consigo misma), **binarias o n-arias** según se establezca entre dos o mas entidades. En la práctica (en DED) las relaciones siempre serán binarias o reflexivas.



- **Tipos de correspondencias:**

### De 1 a 1

En un tipo de correspondencia uno a uno, una fila de la entidad A se corresponde exactamente con una fila en la entidad B y viceversa.



Veamos cuáles son las opciones para pasar a tabla (o tablas) el modelo con este tipo de correspondencia.

#### OPCIÓN 1

Se queda todo en una sola tabla:

A (a, Atributos A, b, Atributos B)

#### OPCIÓN 2

Obtenemos dos tablas, pero la clave primaria de la tabla B pasa como clave foránea a la tabla A:

A (a, Atributos A, \*b)

B (b, Atributos B)

### OPCIÓN 3

Obtenemos dos tablas, pero la clave primaria de la tabla A pasa como clave foránea a la tabla B:

A (a, Atributos A)

B (b, Atributos B, \*a)

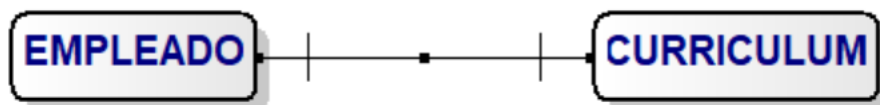
### OPCIÓN 4

Obtenemos dos tablas, pero la clave primaria de la tabla A pasa como clave foránea a la tabla B y la clave primaria de la tabla B pasa como clave foránea a la tabla A:

A (a, Atributos A, \*b)

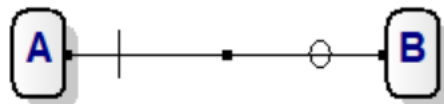
B (b, Atributos B, \*a)

**Ejemplo:**



### De 1 a 0

En un tipo de correspondencia uno a cero, una fila de la entidad A se corresponde con una fila en la entidad B o con ninguna.



Veamos cómo se pasa a tablas el modelo con este tipo de correspondencia.

A (a, Atributos A)

B (b, Atributos B, \*a)

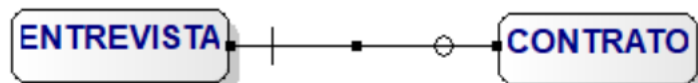
Según lo que modele la tabla B, la clave primaria en B puede ser:

b

a

a,b

**Ejemplo:**



### De 1 a N

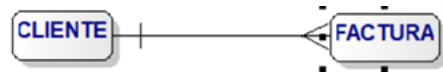
El tipo de correspondencia uno a muchos es uno de los más frecuentes. En este tipo de correspondencia, una fila de la entidad A se puede corresponder con muchas filas de la entidad B, pero una fila de la entidad B solo se corresponde con una fila de la entidad A.



Veamos cómo se pasa a tablas el modelo con este tipo de correspondencia.

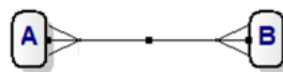
A (a, Atributos A)  
B (b, Atributos B, \*a)

**Ejemplo:**

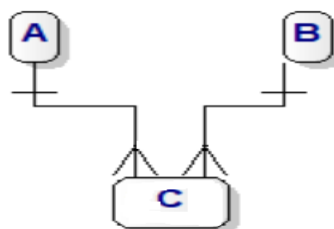


### De M a N

En un tipo de correspondencia muchos a muchos, una fila de la entidad A se puede corresponder con muchas filas de la entidad B y una fila de la entidad B se puede corresponder con muchas filas de la entidad A. En un Diagrama de Estructura de Datos, el tipo de correspondencia muchos a muchos entre la entidad A y la entidad B, se modela eliminando esa correspondencia y creando una nueva entidad y dos tipos de correspondencia uno a muchos.



Este modelo se convierte en:



Veamos cómo se pasa a tablas el modelo con este tipo de correspondencia.

Según lo que modele la tabla C, la clave primaria en C puede ser:

*OPCIÓN 1*

A (a, Atributos A)

B (b, Atributos B)

C (c, Atributos C, \*a, \*b)

*OPCIÓN 2*

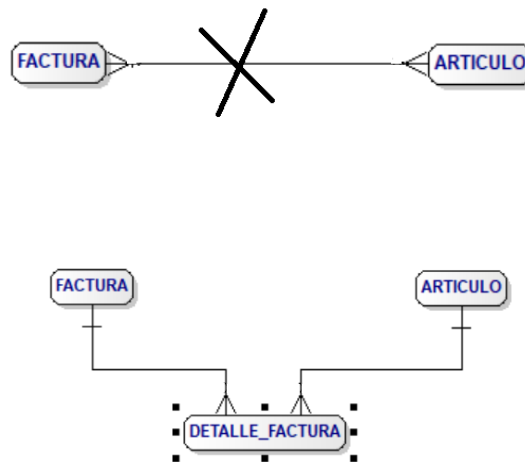
A (a, Atributos A)

B (b, Atributos B)

C(\*a, \*b, Atributos C)

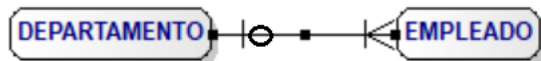
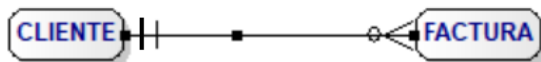
En este caso los valores de a se pueden repetir en la tabla C, los valores de b se pueden repetir en la tabla C, pero lo que no se puede repetir es la combinación de los dos valores.

**Ejemplo:**

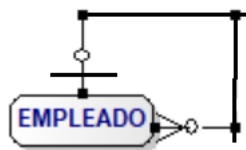


- Cardinalidad máxima y mínima: (0,1), (1,1), (1,N), (0,N)

Ejemplos:



## EJEMPLO DE TIPO DE RELACIÓN REFLEXIVA





### **EJEMPLO 1:**

Se desea hacer el modelo de datos para almacenar los departamentos y los empleados de una empresa. Queremos conocer para cada empleado cuál es su departamento, así como su código de trabajador, nombre, teléfono y DNI. De los departamentos necesitamos almacenar su código y nombre. También se sabe que un departamento puede tener muchos empleados, pero que un empleado pertenece a un único departamento.

*ENTIDADES*

*TIPO DE RELACIONES SEGÚN EL GRADO*

*TIPO DE CORRESPONDENCIA*

*CARDINALIDADES MÍNIMA Y MÁXIMA*

*ATRIBUTOS*

*CLAVE PRIMARIA*

*CLAVE FORÁNEA*

*OCURRENCIAS*

**Ejercicio:** Hacer utilizando una herramienta CASE los seis ejemplos explicados en clase. Para cada uno, generar la base de datos correspondiente. Abre cada base de datos e introduce algunas filas en cada tabla, de manera que comprendas el funcionamiento.

## EJEMPLO 2:

Se quiere diseñar la base de datos para gestionar los pedidos que se realizan a los proveedores de los materiales que vende una empresa. De los proveedores se conoce el código, nombre y dirección. En los pedidos se debe almacenar la fecha en que estos se realizan. De los materiales se conoce su descripción y código. Para cada material que se pide se debe indicar la cantidad solicitada.

Reglas de gestión:

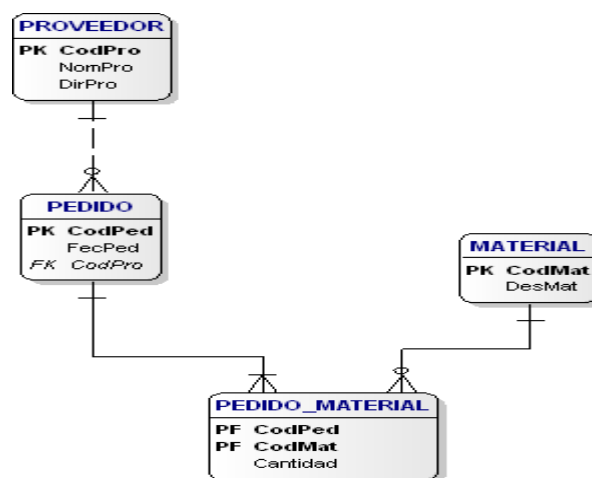
- A un proveedor se le pueden realizar muchos pedidos, pero cada pedido se le hace a un solo proveedor.
- En un solo pedido se pueden solicitar muchos materiales y un material se puede pedir muchas veces y a distintos proveedores.

SOLUCIÓN:

- **Identificar las entidades** dentro del sistema, teniendo previo conocimiento del funcionamiento del sistema. Estas entidades pueden ser: objetos reales, personas, actividades del sistema, objetos abstractos etc.

Por ejemplo, podríamos identificar las entidades Pedido, Materiales, Proveedor.

- **Establecer las relaciones entre las entidades**, describiendo la cardinalidad de los mismos, previo estudio de cada una de las asociaciones de una entidad con las demás identificadas.
- **Dibujar el modelo de datos.**



- Verificar el diagrama. (por ejemplo que no haya bucles)

### **EJEMPLO 3:**

Se quiere diseñar la base de datos de una empresa. La empresa consta de una serie de departamentos de los que se conoce su código, nombre y presupuesto anual. La empresa se dedica a realizar proyectos de diseño gráfico y quiere almacenar para cada proyecto un código, el nombre, descripción del proyecto, fecha de comienzo y la duración estimada del mismo. Interesa conocer que empleado trabaja o ha trabajado en cada proyecto y la fecha en la que ha comenzado a trabajar, así como la fecha en la que ha terminado (no tienen por qué coincidir con la fechas de comienzo y finalización del proyecto). Cada proyecto en el que trabaja un empleado se identifica por un número de referencia. Para cada empleado debemos almacenar su código, nombre y teléfono.

Reglas de gestión:

- Un departamento tiene 1 o más empleados.
- Un empleado pertenece a un solo departamento.
- Un empleado puede no tener asignado departamento durante algún tiempo.
- Un empleado puede trabajar en uno o más proyectos al mismo tiempo.
- En un mismo proyecto pueden trabajar uno o más empleados.

**También tendremos que asegurarnos de que todas las tablas estén normalizadas. Veremos este concepto en el tema 3.**