

# **Modelo Entidad / Relación**

Tema 2

# Objetivos

- Entender el proceso de desarrollo de una base de datos desde su concepción hasta su implementación física.
- Aprender técnicas de modelado conceptual según el modelo entidad/relación.
- Conocer las reglas de transformación del modelo lógico conceptual al modelo lógico relacional.
- Introducir los conceptos básicos de la teoría de la normalización para la optimización del diseño de base de datos.

## El modelo Entidad/Relación

- Es un modelo de datos de plena actualidad en el campo del diseño de base de datos.
- Refleja de una forma sencilla e intuitiva los datos de los sistemas que modela.
- Es un método de representación abstracta del mundo real.
- No es directamente aplicable a un SGBD (necesita una transformación).
- Se concibe como el diagrama inicial.

# 1. El proceso de diseño

Partimos de una descripción de un sistema de información.

- **Fase 1: Diseño conceptual**  
Se establecen las entidades, las relaciones que las vinculan y atributos.
- **Fase 2: Reglas de transformación**  
Se traduce el esquema inicial a un modelo lógico o relacional.
- **Fase 3: Normalización**  
Se refina el modelo.
- **Fase 4: SQL+ajustes**  
Se traduce el esquema a un sistema físico usando un SGBD.

## 2. Elementos del modelo Entidad/Relación

- **Entidades**
- **Atributos**
- **Relaciones**

## 2.1 Entidades

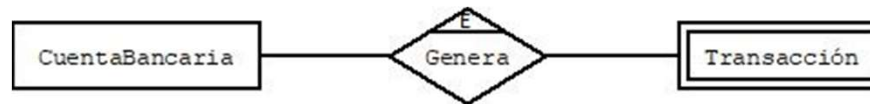
- **Entidad:** Objeto real o abstracto de interés acerca del cual se puede obtener una determinada información.
- **Ocurrencia de entidad:** es una realización concreta de una entidad.
- **Reglas que debe cumplir:**
  - *Tener existencia propia.*
  - *Cada ocurrencia de un tipo de entidad debe poder distinguirse de las demás.*
  - *Todas las ocurrencias de un tipo de entidad deben tener los mismos tipos de atributos.*
- **Clase de entidades:**
  - **Entidades regulares o fuertes:** *tienen existencia por sí mismas, no dependen de ninguna otra entidad para existir.*
  - **Entidades débiles:** *su existencia depende de otra entidad.*
- **Representación gráfica:**

Entidad\_fuerte

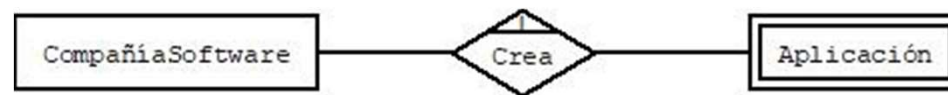
Entidad\_débil

## 2.1.1 Entidades débiles

- **Las entidades débiles dependen de una entidad fuerte mediante una relación.** Esta relación también es débil, puesto que desaparece si desaparece la entidad fuerte. La relación tiene una dependencia que puede ser:
  - **Dependencia de existencia:** expresa que, las ocurrencias de una entidad débil, no tienen ningún sentido en la base de datos sin la presencia de las ocurrencias de la entidad fuerte con la que están relacionadas. (Las transacciones que se dan en una cuenta bancaria).

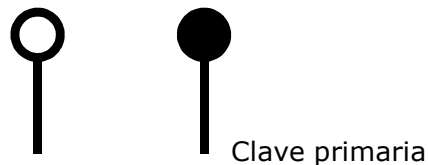


- **Dependencia de identificación:** además de la dependencia en existencia, la entidad débil necesita de la fuerte para poder crear una clave, de tal manera que pueda completar la identificación de sus ocurrencias. (Empresas de software que pueden tener aplicaciones con el mismo nombre).



## 2.2 Atributos

- **Atributo:** propiedad o característica asociada a una determinada entidad y común a todas las ocurrencias de esa entidad.
- **Dominio:** conjunto de valores permitidos para un atributo (cadena de caracteres, número entero...).
- **Atributos identificadores (atributos clave, clave primaria):** Conjunto de atributos que distinguen de manera única cada una de las ocurrencias de una entidad.
- **Atributos descriptores:** Se utilizan para describir la ocurrencia de una entidad.
- **Atributos de relación:** Atributo propio de la relación y que no puede ser cedido a las entidades que intervienen en la relación.
- **Representación gráfica:**





## 2.2 Atributos

- ***Tipos de atributos***

- **Atributos obligatorios:** debe tomar un valor obligatoriamente.
- **Atributos opcionales:** puede no tomar un valor porque sea desconocido en un momento determinado (valor nulo).
- **Atributos compuestos:** se puede descomponer en atributos más sencillos (hora\_de\_salida se puede descomponer en horas y minutos).
- **Atributos univaluados:** toma un único valor.
- **Atributos multivaluados:** puede tomar varios valores (teléfono puede tomar los valores de teléfono fijo y móvil).
- **Atributos derivados:** su valor se puede calcular a través de otros atributos. (Edad se puede calcular a partir de la fecha de nacimiento).

## 2.3 Relaciones

- **Relación:** asociación entre una o más entidades.
- **Propiedades de las relaciones:**
  - **Nombre** que la identifica.
  - **Grado:** número de tipos de entidad que asocia.
  - **Tipo de correspondencia:** número máximo de ocurrencias de cada tipo de entidad que pueden intervenir.
- Las relaciones se nombran con **verbos**.
- También pueden tener **atributos**.
- **Representación gráfica:**



## 2.4 Correspondencia y Cardinalidad

- **Tipo de correspondencia**

Número máximo de ocurrencias de cada tipo de entidad que pueden intervenir en una ocurrencia de la relación que se está tratando.

- **Tres clases:**

- ✓ Relaciones uno a uno (**1:1**)
    - ✓ Relaciones uno a varios (**1:N**)
    - ✓ Relaciones varios a varios (**N:M**)

- **Cardinalidad**

Representa la participación en la relación de cada una de las entidades afectadas, es decir, el número máximo y mínimo de ocurrencias de un tipo de entidad que pueden estar interrelacionadas con una ocurrencia de otro tipo de entidad.

- **Representación gráfica:**

(0,1), (1,1), (0,n) o (1,n) en cada entidad.

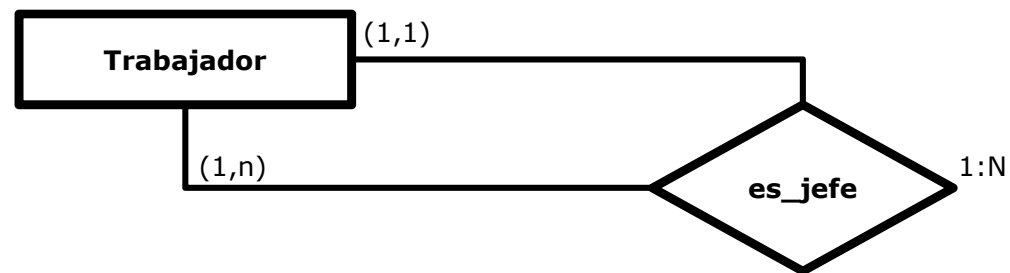
## 2.5 Casos especiales

- **Relaciones reflexivas:**

Aquellas que se dan entre ocurrencias de un mismo tipo de entidad.

***Ejemplo:***

Un trabajador tiene un jefe que a su vez es trabajador de la empresa.



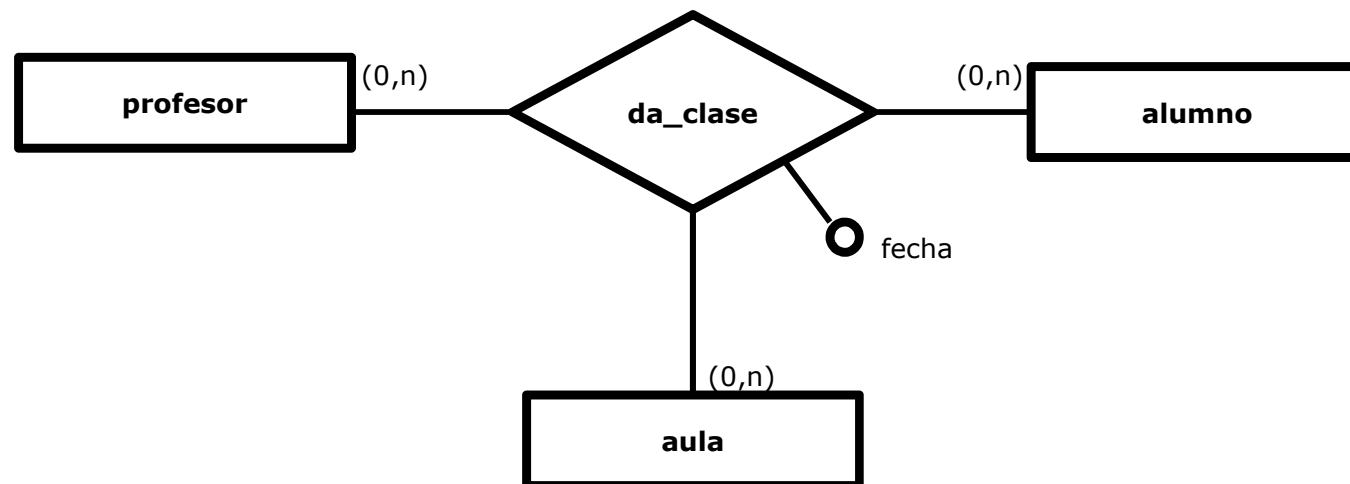
## 2.5 Casos especiales

- **Relaciones ternarias :**

Aquellas que se dan entre ocurrencias de tres entidades diferentes.

***Ejemplo:***

Disponemos de las entidades profesor, aula y alumno y queremos saber cuándo y en qué aula un profesor da clases a un alumno.



### 3. El modelo E/R extendido (o ampliado)

- **Generalización y especialización**
- **Agregación**

## 3.1 Generalización y especialización

### ■ **Generalización:**

- Una entidad  $E$  es una generalización de un grupo de entidades  $E_1, E_2, \dots, E_n$ , si cada ocurrencia de cada una de esas entidades es también una ocurrencia de  $E$ .
- Todas las propiedades de  $E$  son heredadas por las subentidades.
- Cada subentidad tendrá sus propios atributos independientes de la generalización.

### ■ **Especialización:**

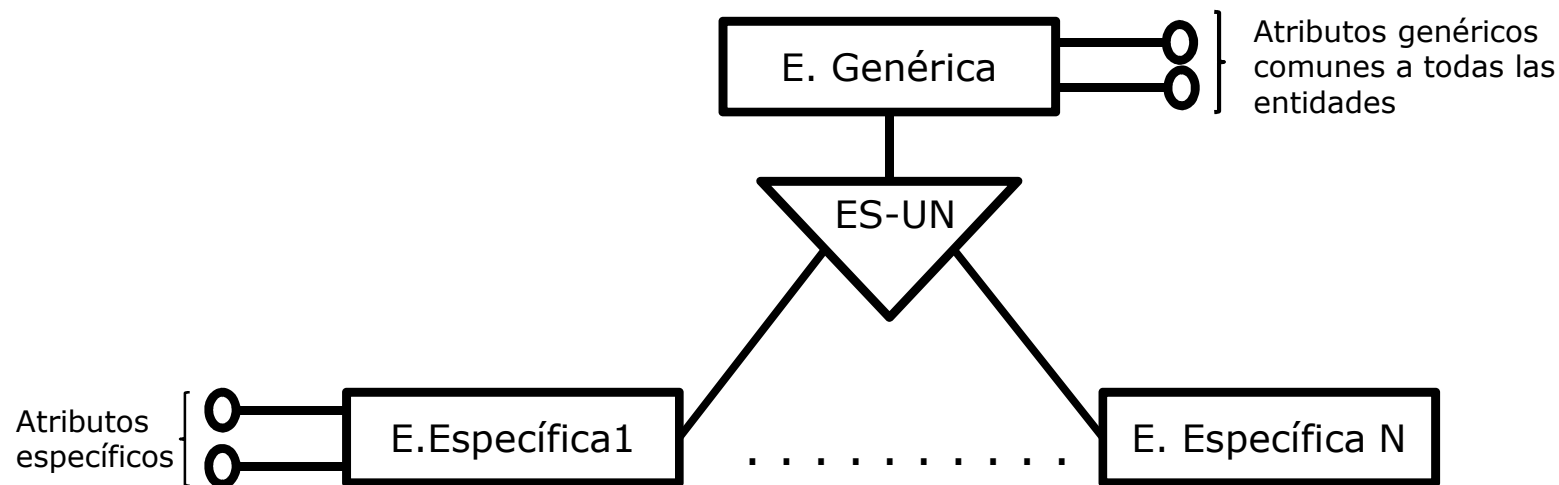
- El conjunto de entidades  $E_1, E_2, \dots, E_n$  está incluido en el conjunto de entidades  $E$ .

### ■ **Se emplean para:**

- Resaltar las características comunes de varios conjuntos de entidades.
- Especificar de manera más concreta las características de algunos de sus subconjuntos.

## 3.1 Generalización y especialización

- **Representación gráfica:**



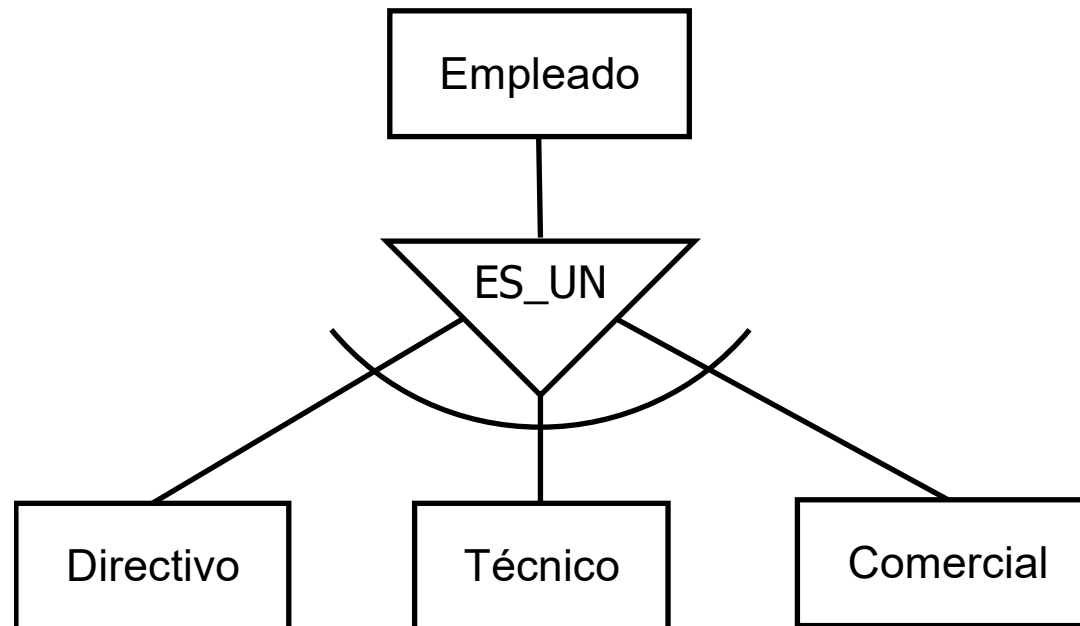
E es una generalización de  $E_i$ ,  $E_i$  es un subtipo de E, E es un supertipo de  $E_i$ ,  $E_i$  es una especialización de E.



### 3.1.1 Tipos de especialización

- **Especialización Exclusiva:** cada ocurrencia de la superclase solo puede materializarse en una de las especializaciones.

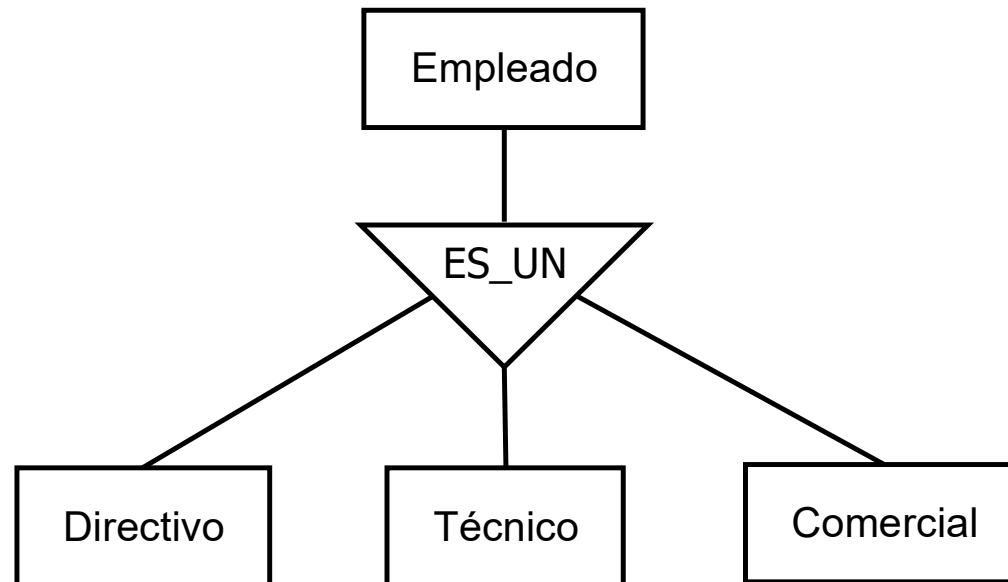
**Ejemplo:** un empleado puede ser de un solo tipo (directivo, técnico o comercial).



### 3.1.1 Tipos de especialización

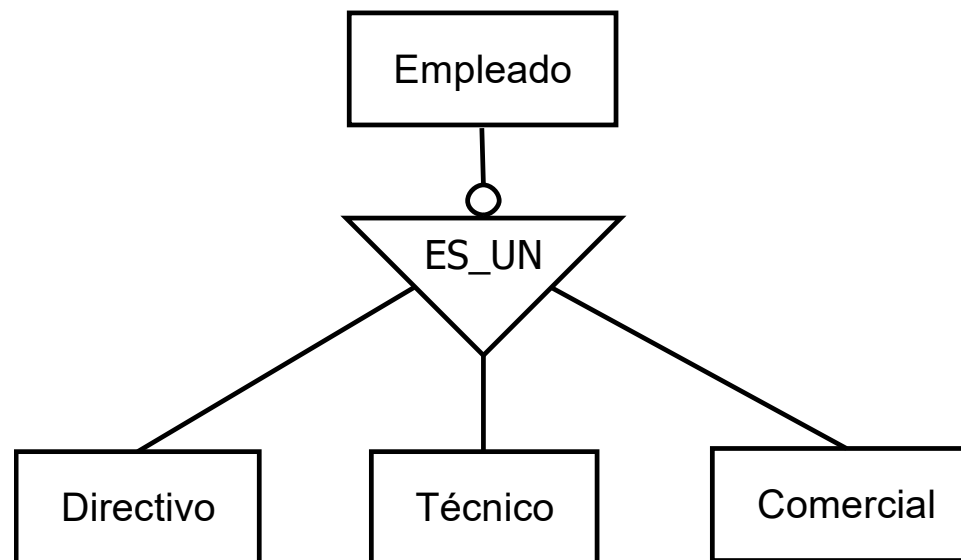
- **Especialización Inclusiva:** las ocurrencias de la superentidad pueden materializarse a la vez en varias ocurrencias de las subentidades.

**Ejemplo:** un empleado puede ser de varios tipos.



### 3.1.1 Tipos de especialización

- **Especialización Total:** la superentidad tiene que materializarse obligatoriamente en una de las especializaciones.



- **Especialización Parcial:** la superentidad no tiene por qué materializarse en una de las especializaciones (es opcional). Se representa igual que la especialización inclusiva.

## 3.2 Agregación

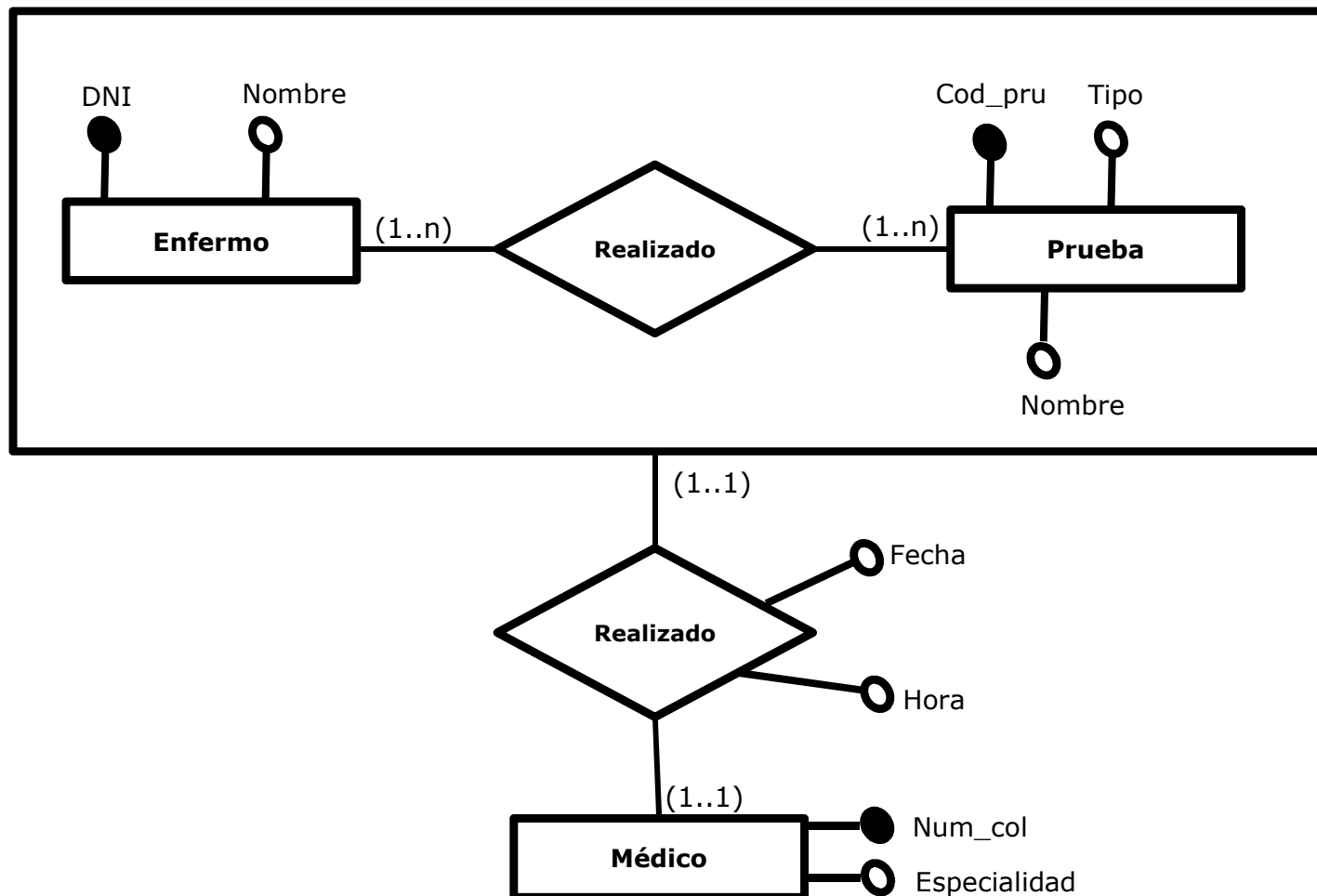
### **Agregación:**

Es una abstracción a través de la cual las relaciones se tratan como entidades de un nivel más alto.

### **Se utiliza para:**

- Expresar relaciones entre relaciones.
- Expresar relaciones entre relaciones y conjuntos de entidades.

## 3.2 Agregación



## 4. Reglas de transformación

- **Reglas básicas:**

- Toda **entidad** se transforma en una relación.
- Las relaciones **N:M** se transforman en una relación.
- Las relaciones **1:N** dan lugar a una propagación de clave o a una nueva relación.
- Las relaciones **1:1** se transforman siguiendo cualquier regla de las anteriores.

## 4.1 Transformación de entidades y atributos

Cada **entidad** → Una tabla del modelo relacional

**Clave primaria** → Identificador principal

Cada **atributo** → Un atributo o columna de la tabla

Los atributos que componen la **clave primaria** de cada tabla/relación se **subrayan**.

- **Entidades débiles**

La tabla contendrá sus atributos propios, más la clave primaria de la entidad de que depende.

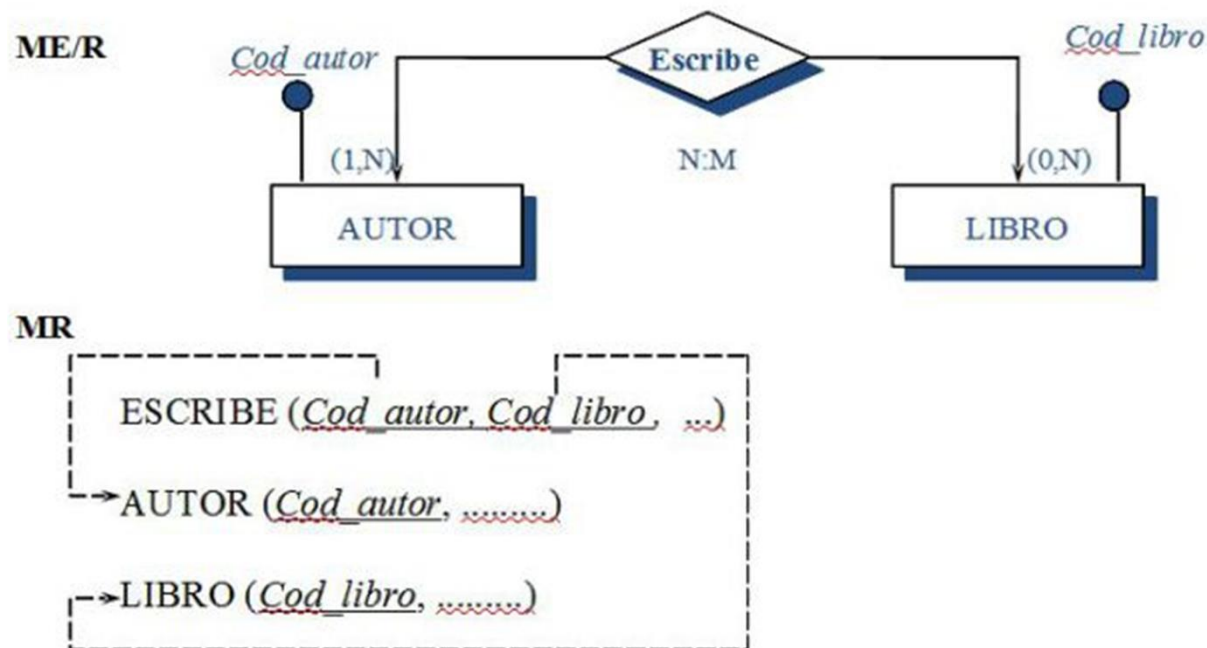
La **clave primaria** se formará según el tipo de dependencia:

- **Dependencia en existencia:** los atributos clave de la entidad débil.
- **Dependencia de identificación:** los atributos clave de la entidad débil y de la entidad fuerte.

## 4.2 Transformación de relaciones N:M

Cada entidad se transforma en una tabla y para la relación se genera una nueva tabla que tendrá como clave primaria la **concatenación de las claves primarias** de los tipos de entidad que asocia.

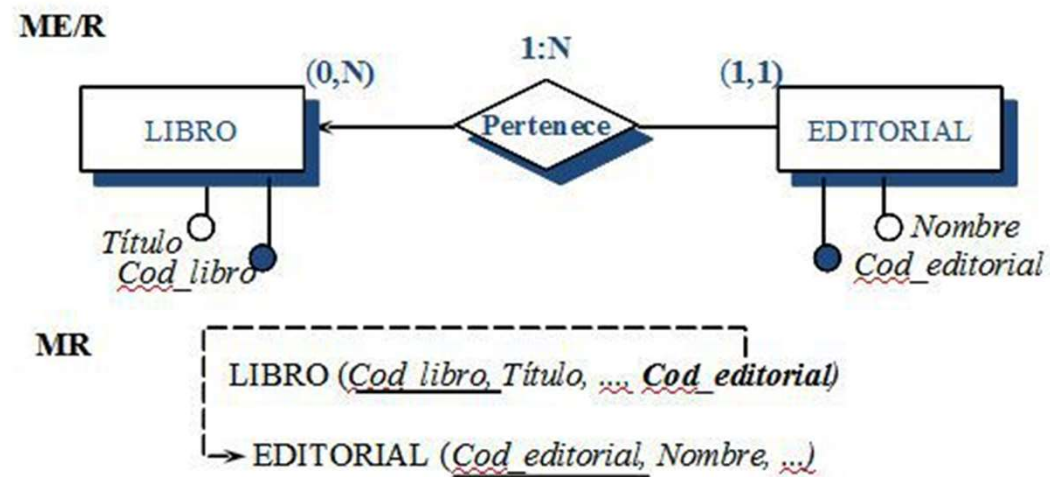
Si la relación tiene atributos, éstos se transforman en columnas de la tabla.





## 4.3 Transformación de relaciones 1:N

Propagar el identificador de la clase de cardinalidad máxima 1 a la que es N.



## 4.3 Transformación de relaciones 1:N

Transformamos en una relación (como en el caso N:M) cuando:

- La relación tiene atributos propios.
- La correspondencia 1 se debiera a una cardinalidad (0,1).

Hay que tener en cuenta la semántica.



## 4.4 Transformación de relaciones 1:1

Se puede usar cualquier regla anterior.

Se recomienda:

- **Si la relación es (0,1):(0,1):** crear una relación (evita tener muchos nulos).
- **Si la relación es (0,1):(1,1):** propagar la clave de la entidad (1,1) a la (0,1).
- **Si la relación es (1,1):(1,1):** la propagación es indiferente. Se atiende a los criterios de frecuencia de acceso a cada tabla.



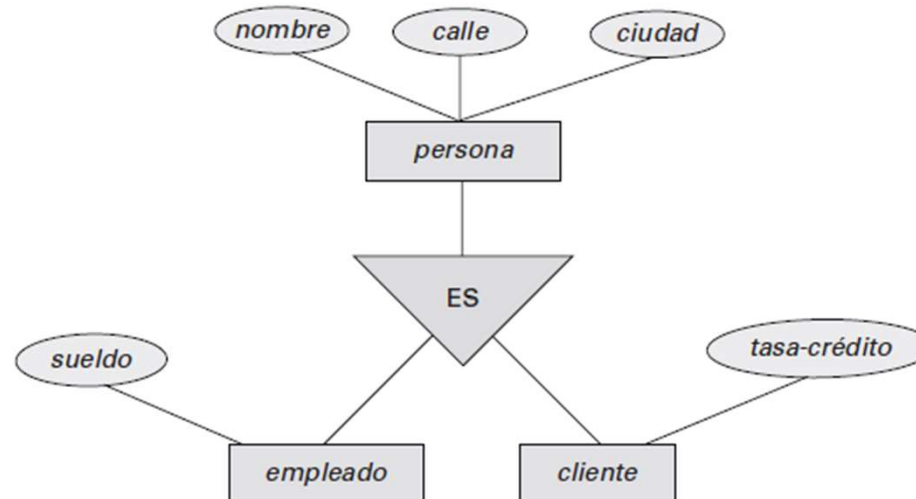
## 4.5 Transformación de relaciones ternarias

- **Cardinalidad M:N:P**  
Una tabla con las claves primarias de las entidades participantes junto con los atributos propios de la relación.
- **Cardinalidad M:N:1**  
Requiere análisis semántico.
- **Cardinalidad M:1:1**  
No existe una única interpretación posible.

## 4.6 Transformación de generalizaciones

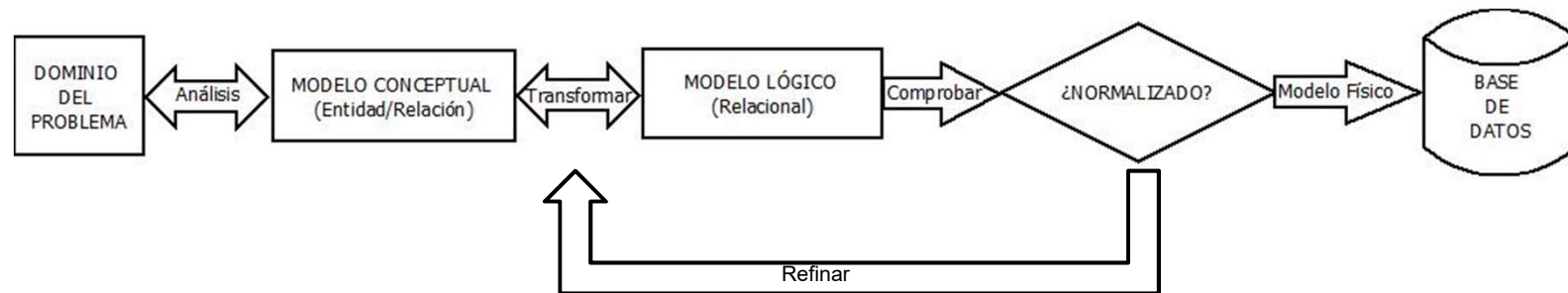
Se puede hacer de dos maneras:

- Crear **una tabla por cada conjunto de entidades** del diagrama (superentidad y subentidades), incorporando el campo clave de la superentidad a las tablas de las subentidades.
- Crear **una tabla por cada caso particular** incorporando todos los atributos de la superentidad (se pierde la idea de jerarquía).



## 5. Normalización

- **Normalización:** es el proceso de obligar a los atributos de un diseño a cumplir ciertas formas normales.



- **Objetivos de las formas normales:**
  - Evitar la redundancia de datos.
  - Evitar ciertas anomalías a la hora de operar con los datos.

## 5. Normalización

- **Dependencia funcional**

Se dice que un atributo Y depende funcionalmente de otro atributo X ( $X \rightarrow Y$ ) si cada valor de X tiene asociado en todo momento un único valor de Y.

- **Dependencia funcional completa**

Dada una combinación de atributos  $X(X_1, X_2, \dots)$  se dice que Y tiene dependencia funcional completa de X si depende funcionalmente de X, pero no depende de ningún subconjunto del mismo.

- **Dependencia funcional transitiva**

Dada la tabla T, con atributos (X,Y,Z), donde  $X \rightarrow Y$ ,  $Y \rightarrow Z$  e  $Y \rightarrow X$ , se dice que X depende transitivamente de Z  $\wedge$  que,  $X \rightarrow Z$ .

## 5. Normalización

### ■ Formas normales

- **FN1:** Se prohíbe que en una tabla haya atributos que puedan tomar más de un valor.
- **FN2:** Si está en FN1 y además, cada atributo que no forma parte de la clave tiene dependencia completa de la clave principal.
- **FN3:** Si está en FN2 y además, no hay ningún atributo no clave que depende de forma transitiva de la clave.
- **FNBC:** Exige que el modelo esté en FN3, y que además, todo implicante de la tabla sea una clave candidata.