



**GeeksHubs**  
academy \_

# BBDD SQL



# BBDD



# ¿Qué es una Base de datos?

- Colección **de datos interrelacionados y estructurados** almacenados en conjunto sin redundancias perjudiciales o innecesarias.
- Su finalidad es **servir a una aplicación o más**, de la mejor manera posible.
- Los datos se almacenan de modo que resulte **independiente de los programas** que los usan se emplean métodos bien determinados para incluir nuevos datos y para modificar o extraer los datos almacenados.



# DBMS (Database Managment System)

Un **conjunto coordinado de programas, procedimientos, lenguajes, etc.** Que suministra tanto a los usuarios no informáticos como a los analistas, programadores o al administrador los medios necesarios para **describir, recuperar y manipular los datos amacenados en la base, manteniendo su integridad, confidencialidad y seguridad.**



# DBMS (Database Managment System)

Un DBMS debe contener las **3 partes** fundamentales de base de datos que son:

1. **Definición**
2. **Manipulación**
3. **Utilización.**

A veces la utilización tiene una interface mas amigable a la manipulación, siendo esta última orientada a la programación.



# Lenguaje de definición de datos (DDL)

En la parte de definición de un DBMS debe existir un conjunto de instrucciones o comandos que permitan **crear las estructuras de las bases de datos**.

A este conjunto de instrucciones se le conoce como lenguaje de definición de datos. En esta se especifican los elementos de datos que integran la BD, su **estructura y las relaciones que existen entre ellos, las reglas de integridad semántica, los controles a efectuar antes de autorizar el acceso a la base de datos**, así como las características de tipo físico y las vistas lógicas de los usuarios.

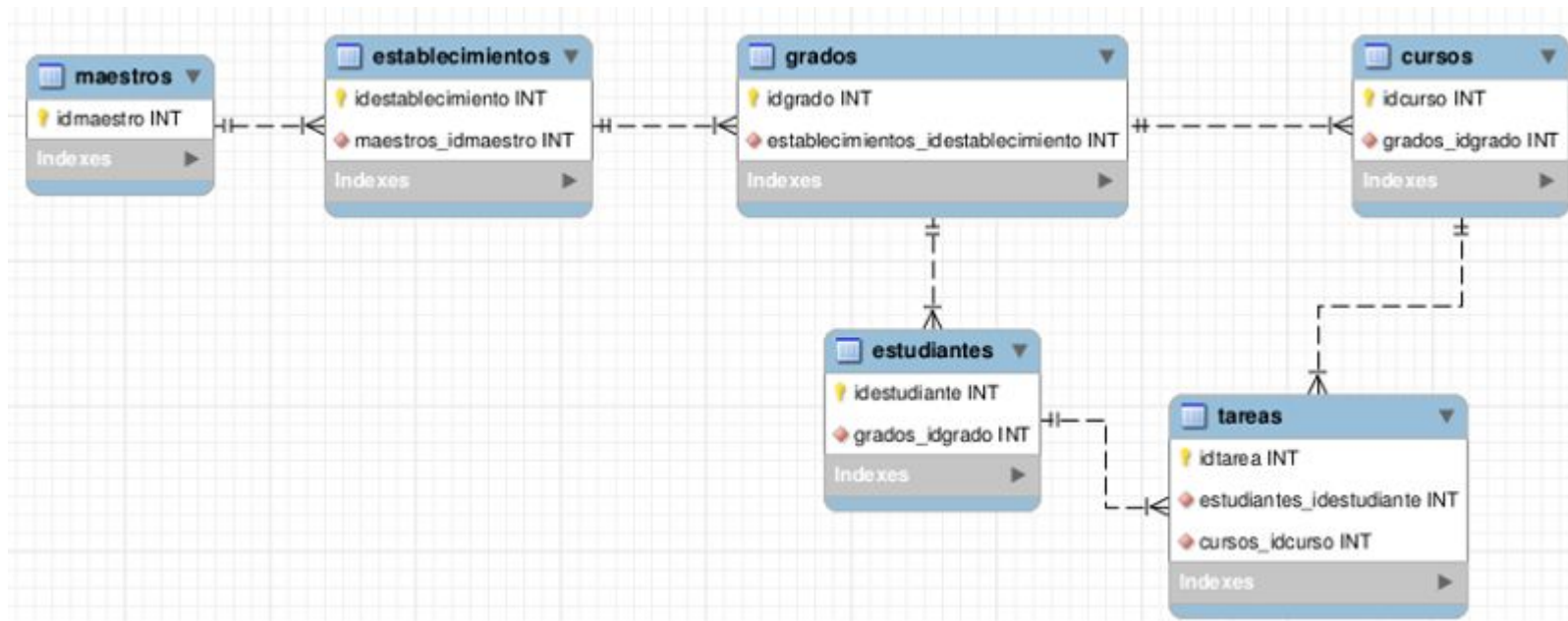


# Diagrama ER



# Diseño de la Base de Datos

Al igual que existe una metodología para crear programas que resuelvan una problemática, es necesario establecer un **procedimiento de diseño de las estructuras de las bases de datos**.





# Diseño de la Base de Datos

El diseño se divide en diferentes niveles: nivel conceptual, nivel lógico y nivel físico.



# Diseño de la Base de Datos

En el modelo conceptual se **visualiza los conjuntos de datos** que deben incluirse en el modelo de datos, **así como sus relaciones.**



En esta etapa se puede hacer uso del modelo **Entidad-Vínculo**, que es uno de los modelos con mayor semántica de diseño.



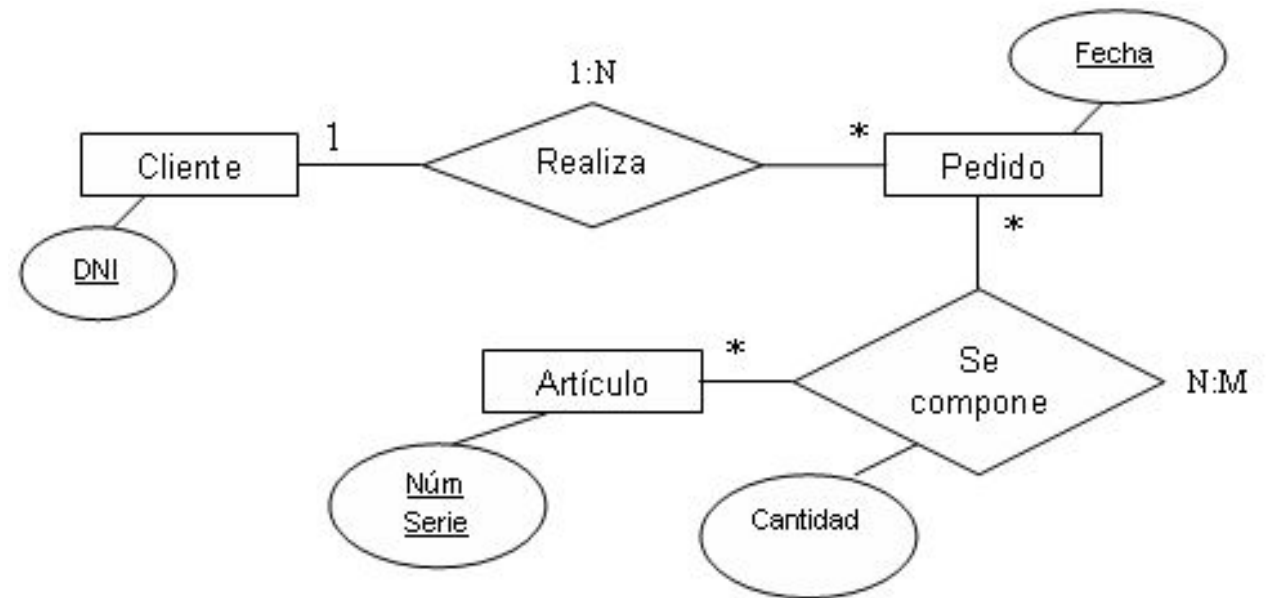
# Diseño de la Base de Datos

1. Después de **identificar el grupo de datos**, se debe generar una **estructura lógica** para la **manipulación** de la base de datos por medio del **DBMS**.
2. Para ello es necesario usar un **modelo** que puede ser el **relacional**, que además de establecer la estructura de campos y registros, permite establecer reglas de integridad de datos y reglas de normalización para mejorar la estructura.
3. Finalmente la parte de estructura física la maneja el DBMS con algunos datos que son necesarios, como por ejemplo la creación de índices.



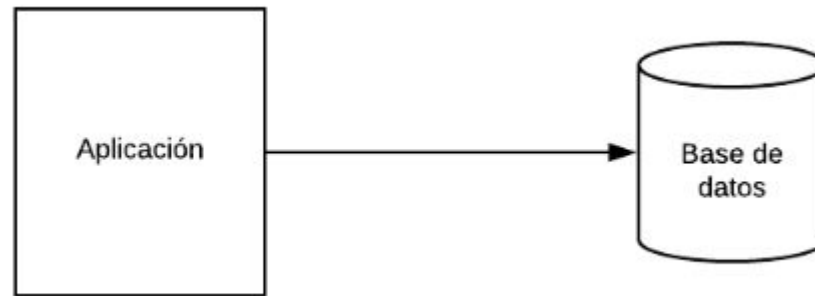
# Modelo Entidad Relación.

El modelo Entidad-Vínculo fue propuesto por **Peter P. Chen** en diferentes artículos en el año de 1976. Consiste en un modelo de representación del mundo real que consiste en Entidades y relaciones.



# Entidades

Se puede definir la entidad como aquel objeto (real o abstracto) acerca del cuál queremos almacenar información en la base de datos.



Es como una persona, lugar, cosa, concepto o suceso, real o abstracto, de interés para la empresa. La estructura genérica se denomina **entidad**.



# Entidades

Una entidad se representa por un rectángulo como lo muestra la siguiente figura:

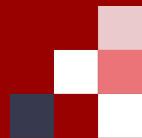


Además de las **entidades normales**, existe el concepto de **entidades débiles** que son entidades que no pueden existir sin una entidad normal (también llamada general).

Un ejemplo de entidad débil puede ser la cuenta de crédito de un cliente. Si el cliente no existe, la cuenta tampoco.



# Kata I



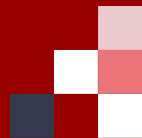
## KATA I: identifica las entidades

Una empresa que se dedica a la venta de productos de papelería desea llevar el control de su inventario a través de un sistema basado en bases de datos. El sistema debe registrar las ventas de mostrador e imprimir facturas si es necesario, con los datos de los clientes. Igualmente, se deben capturar las facturas de los proveedores para actualizar las existencias de los productos. Cada mes el sistema debe generar reportes sobre las existencias de los productos para realizar un inventario. El sistema debe ser capaz de manejar información global de ventas al mes, al año, así como la determinación de la existencia en la tienda de productos.





# Kata II



## KATA II: identifica las entidades

Una institución educativa desea llevar el control de sus alumnos a través de un sistema. La institución cuenta con 9 carreras diferentes, cada una de ellas divididas en un plan curricular de 9 semestres con 5 materias cada semestre. Dicho plan curricular asigna las materias a un semestre. El sistema debe llevar un control del historial académico de cada alumno. A cada catedrático se le asigna una o varias materias a impartir. En esta institución existen catedráticos que debe reportar 4 calificaciones parciales y un final durante el semestre. Hacer un diagrama E-R del modelo de datos

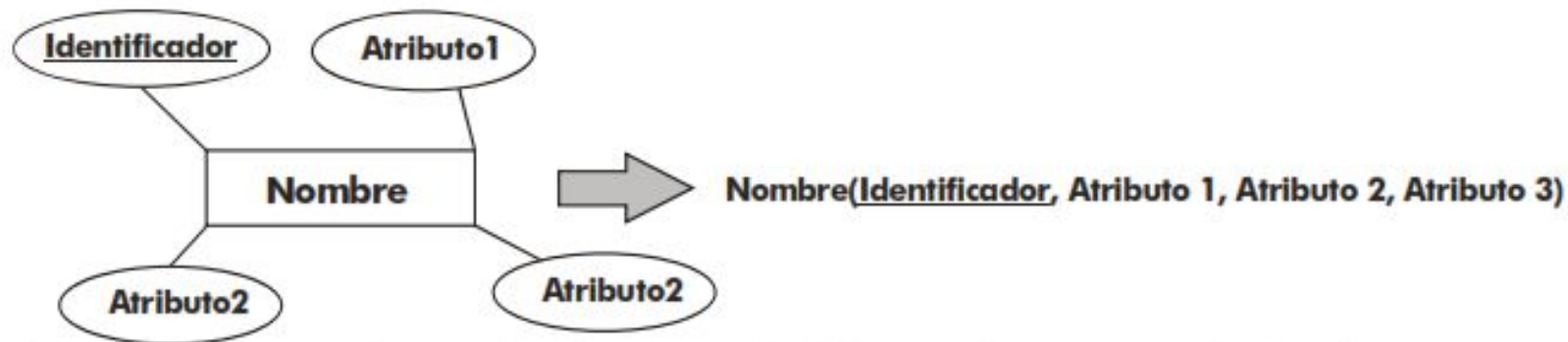


# Modelo Relacional



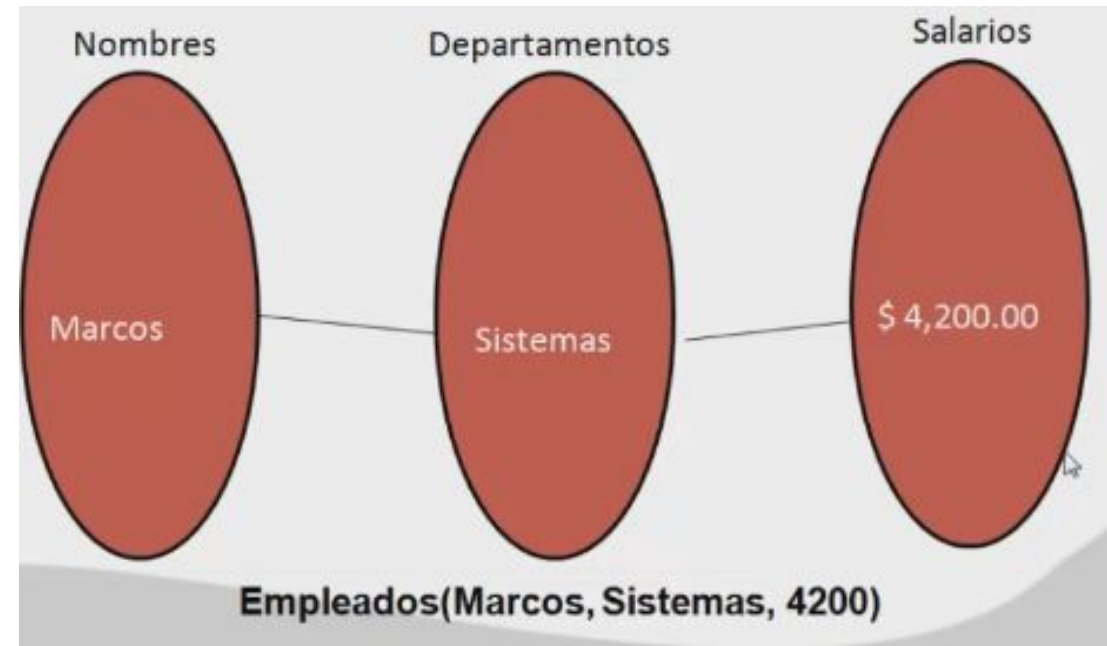
# Modelo Relacional

El modelo relacional fue propuesto por Codd en 1970 y es un modelo que establece un conjunto de datos que se obtiene de la **relación de los dominios de atributos** de los elementos que lo conforman. A este conjunto de datos se le conoce como **relaciones**.



# Modelo Relacional

Se le llama relacional porque en los dominios de cada atributo o campo, existe una relación única de valores de cada dominio a lo que se conoce como **tupla o registro**.



# Modelo Relacional

Los informáticos **ven a las relaciones como una tabla**, o sea un conjunto de **filas llamadas registros** que representan las **tuplas** y **columnas llamadas campos** que representan los **atributos**:

Nombre	Departamento	Salario
Pedro	Sistemas	3000
Marcos	Sistemas	4200
Raúl	Mercadotecnia	3000
Alberto	Contabilidad	2000



# Restricciones de dominio

El valor de cada atributo debe ser un valor atómico del dominio para ese atributo. Esto quiere decir que el valor de un atributo debe **ser un valor simple y debe estar dentro del conjunto de valores permitidos**.

Por ejemplo, la edad de una persona puede tomar un valor de 1 0 y no puede tomar un valor de 300 o un conjunto de valores como {11,12,13,14}.



# Restricciones de clave

Una relación se define como un conjunto de tuplas. Por definición, todos los elementos de una relación deben ser diferentes. Al conjunto de atributos se le llama superclave, pues entre todos se distingue a cada tupla de la relación.

**Una clave es un atributo o conjunto mínimo de atributos que identifican de manera única a las tuplas en una relación.** A esta clave se le llama superclave mínima. La restricción de clave establece que para cada relación se debe definir una clave.





# Restricciones de integridad

Un esquema de bases de datos relacional **es un conjunto de esquemas de relaciones y en conjunto de restricciones de integridad**. Las restricciones de integridad se dividen en integridad de entidades e integridad referencial. La restricción de entidad establece que ningún valor de clave primaria puede ser nulo.

La restricción de integridad referencial se especifica entre dos relaciones y sirve para mantener la consistencia entre tuplas de dos relaciones y establece que una tupla en una relación que haga referencia a otra relación deberá referirse a una tupla existente en esa relación.



# Transformación del esquema conceptual E-R al relacional

El paso de un esquema en el modelo E-R al relacional está basado en los tres principios siguientes:

1. Todo tipo de entidad se convierte en una relación.
2. Todo tipo de vínculo N:M se transforma en una relación.
3. Todo tipo de vínculo 1 :N se traduce en el fenómeno de propagación de clave si el vínculo no tiene atributos o se crea una nueva relación si los tiene.

El fenómeno de propagación de clave consiste en pasar la clave de la entidad débil a la entidad independiente, para que esta se comporte como una clave externa.



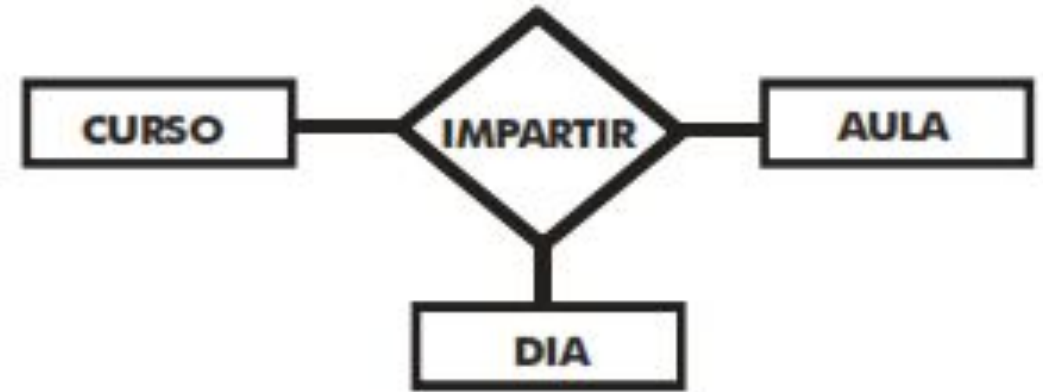
## Representación Gráfica.

La representación gráfica de las entidades se realiza con un rombo al que se le unen líneas que se dirigen a las entidades, las relaciones tienen nombre (se suele usar un verbo). Ejemplo trabajar:





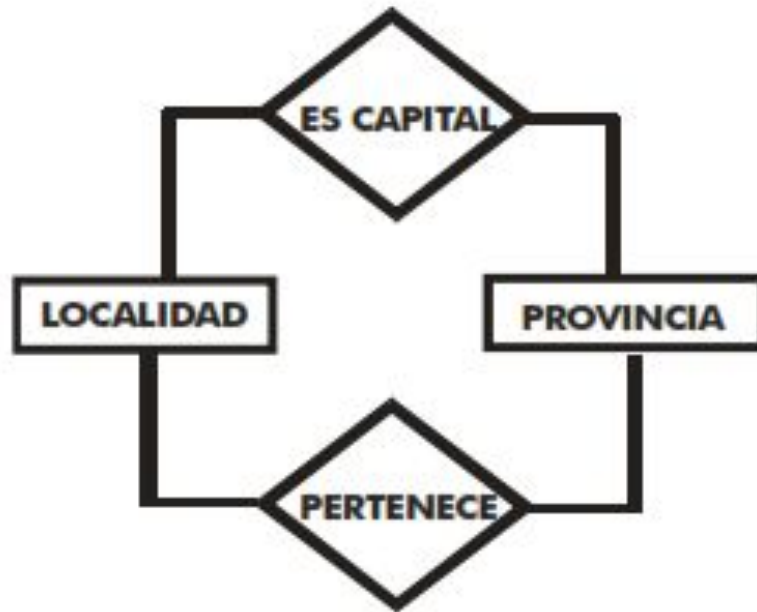
Relación binaria



Relación ternaria



Relación binaria



Relación doble



Relación reflexiva



# CARDINALIDAD

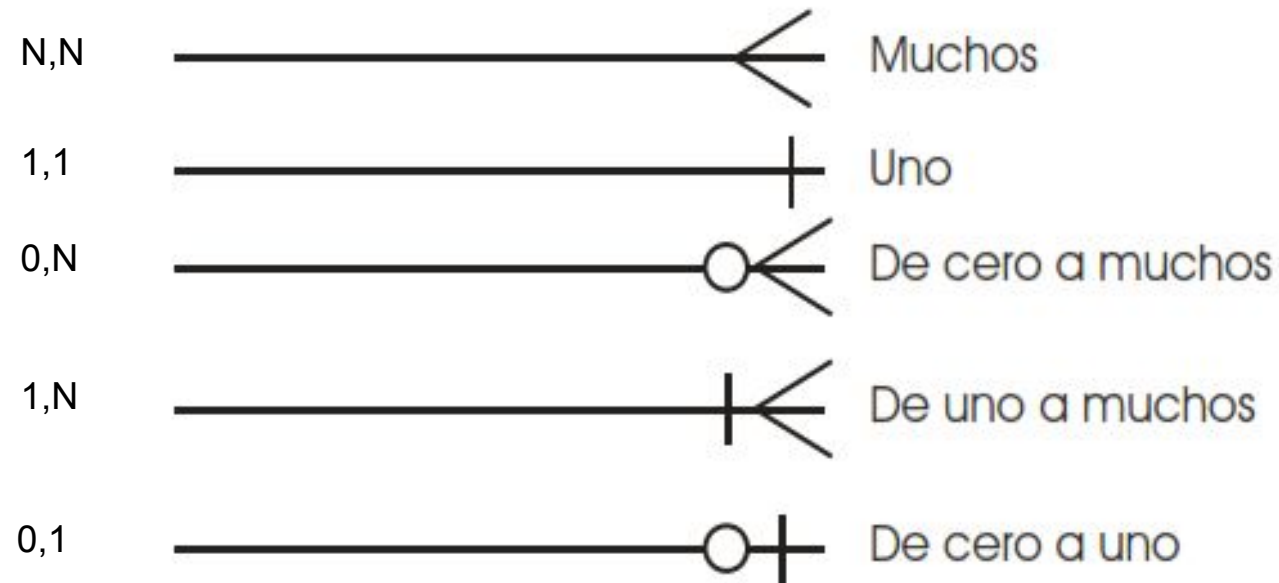
Indica el número de relaciones en las que una entidad puede aparecer. Se anota en términos de:

- **Cardinalidad mínima.** Indica el número mínimo de asociaciones en las que aparecerá cada ejemplar de la entidad (el valor que se anota es de cero o uno)
- **Cardinalidad máxima.** Indica el número máximo de relaciones en las que puede aparecer cada ejemplar de la entidad (puede ser uno o muchos)



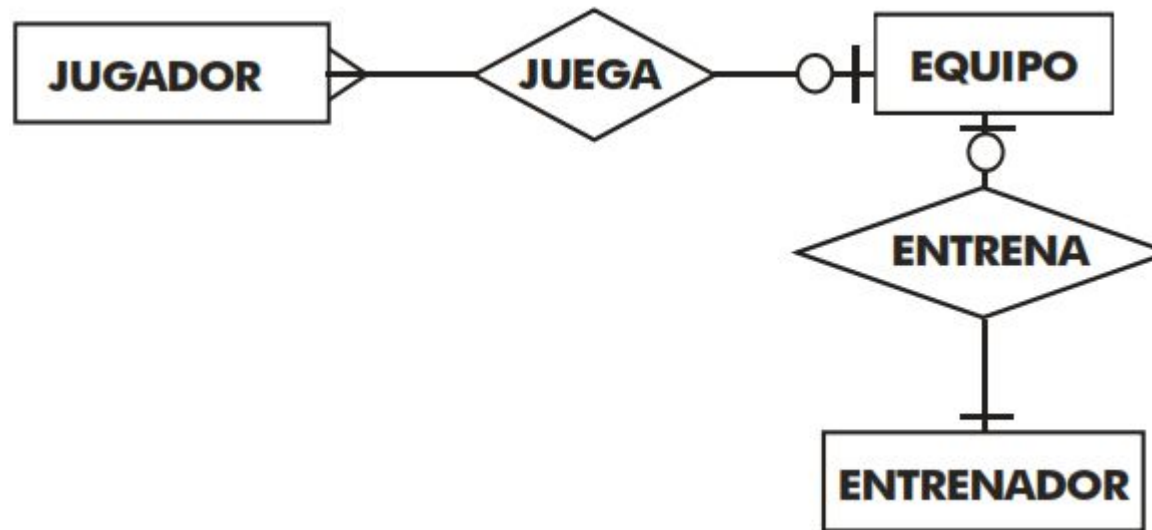
# CARDINALIDAD

En los esquemas entidad / relación la cardinalidad se puede indicar de muchas formas. Actualmente una de las más populares es esta:



## EJEMPLO

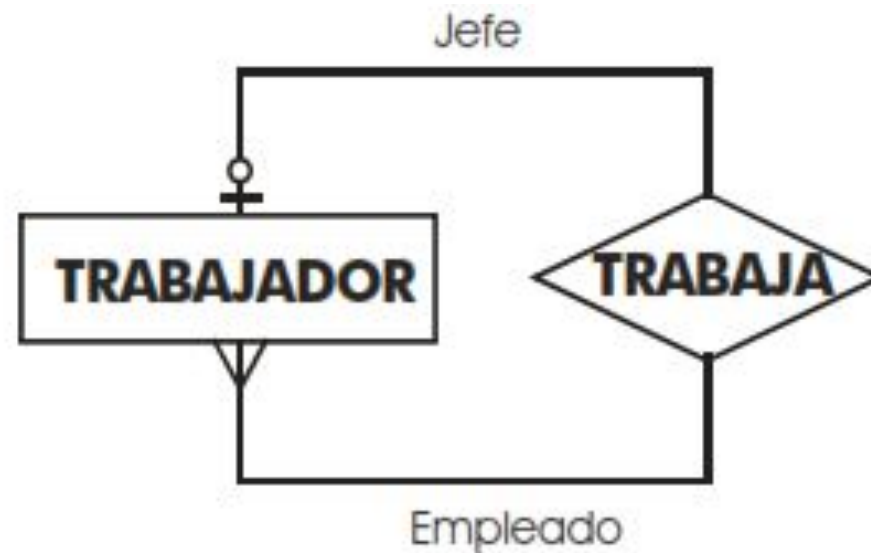
En el ejemplo, cada equipo cuenta con varios jugadores. un jugador juega como mucho en un equipo y podría no jugar en ninguno. Cada entrenador entrena a un equipo (podría no entrenar a ninguno), el cual tiene un solo entrenador.





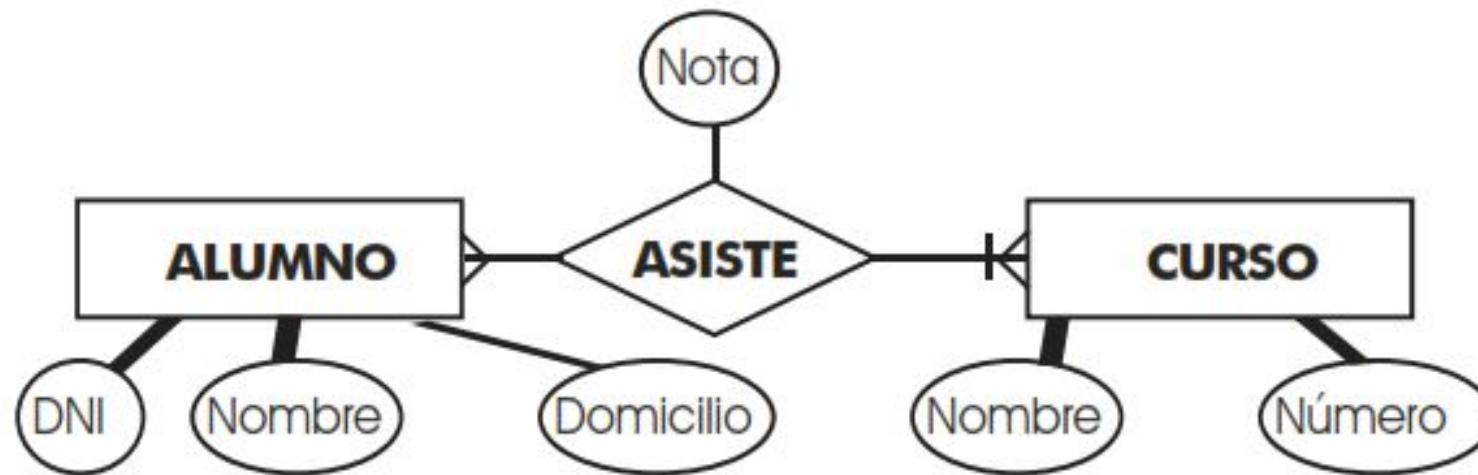
# ROLES

A veces en las líneas de la relación se indican **roles**. Los roles representan el papel que juega una entidad en una determinada relación. Ejemplo:



# ATRIBUTOS

Describen propiedades de las entidades y las relaciones. En este modelo se representan con un círculo, dentro del cual se coloca el nombre del atributo. Ejemplo:



# TIPOS DE ATRIBUTOS

compuesto



opcionales

Lo son si pueden tener valor nulo:



múltiples

Pueden tomar varios valores:



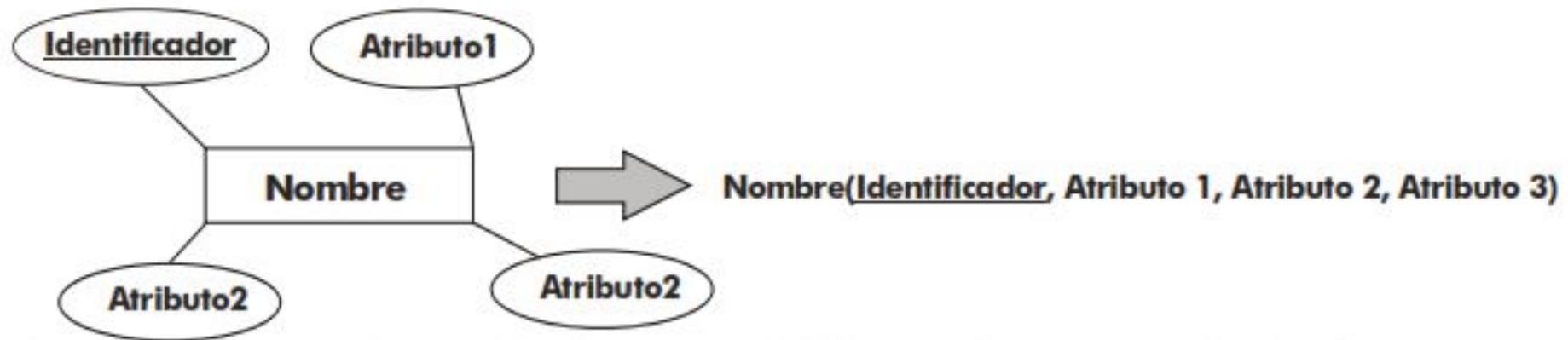
# IDENTIFICADOR - PK

Se trata de uno o más campos cuyos valores son únicos en cada ejemplar de una entidad. Se indican subrayando el nombre del identificador. Para que un atributo sea considerado un buen identificador tiene que cumplir:

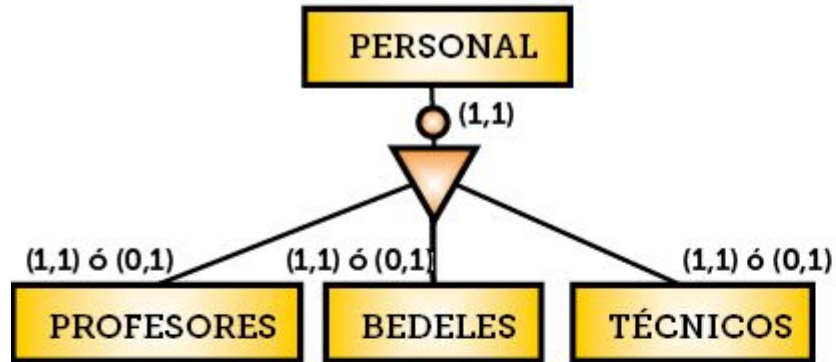
- 1- Deben distinguir a cada ejemplar teniendo en cuenta las entidades que utiliza el modelo.**  
No tiene que ser un identificador absoluto.
- 2- Todos los ejemplares de una entidad deben tener el mismo identificador.**
- 3- Cuando un atributo es importante aun cuando no tenga una entidad concreta asociada,**  
entonces se trata de una entidad y no de un atributo



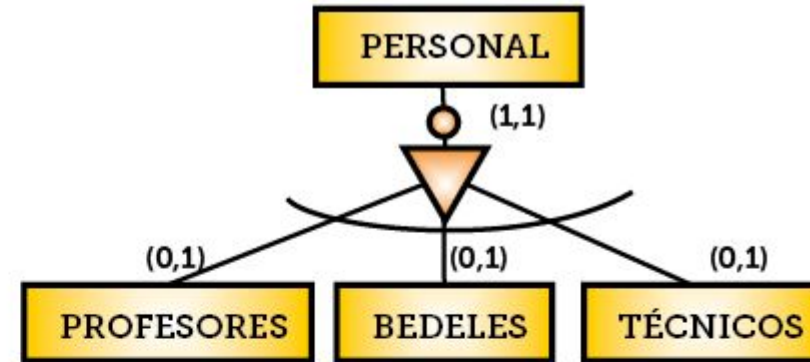
## IDENTIFICADOR



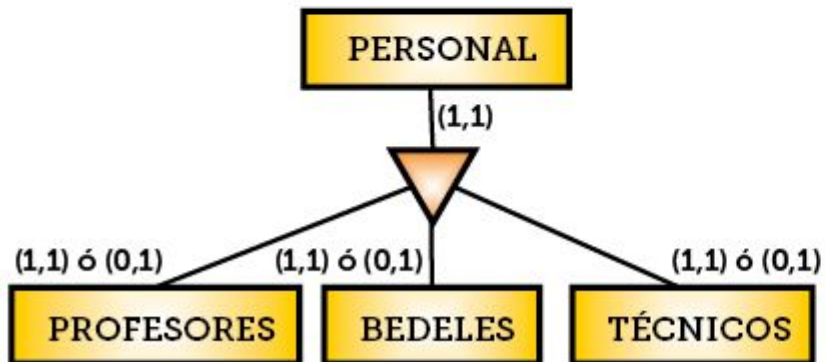
# Relaciones ISA o herencia



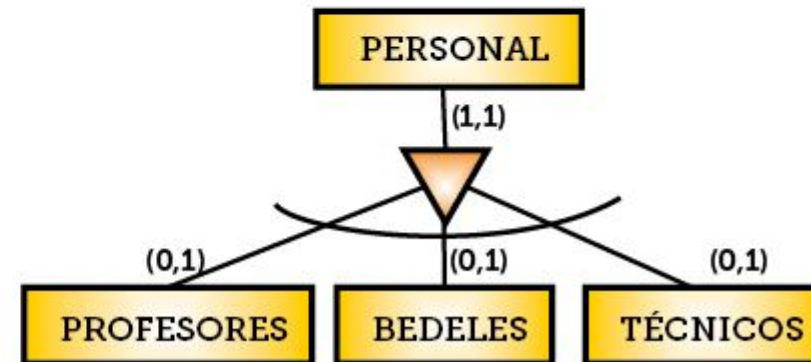
Relación ISA solapada parcial



Relación ISA exclusiva parcial



Relación ISA solapada total



Relación ISA exclusiva total



# Obligatoriedad.

Indica si los ejemplares obligatoriamente se relacionan con ejemplares de las subentidades. Es decir si hay personal que no es profesor ni bedel ni técnico o si fijo es alguna de esas tres profesiones. Hay dos posibilidades:

- **Relaciones de jerarquía parcial.** Indican que hay ejemplares de la superentidad que no se relacionan con ningún ejemplar de las subentidades (por ejemplo, hay personal que no es ni profesor, ni bedel ni técnico). Se indican con un círculo encima del triángulo de la relación ISA.
- **Relaciones de jerarquía total.** Indican que todos los ejemplares de la superentidad se relacionan con alguna subentidad (no hay personal que no sea ni profesor, ni bedel ni técnico).



# Número de relaciones.

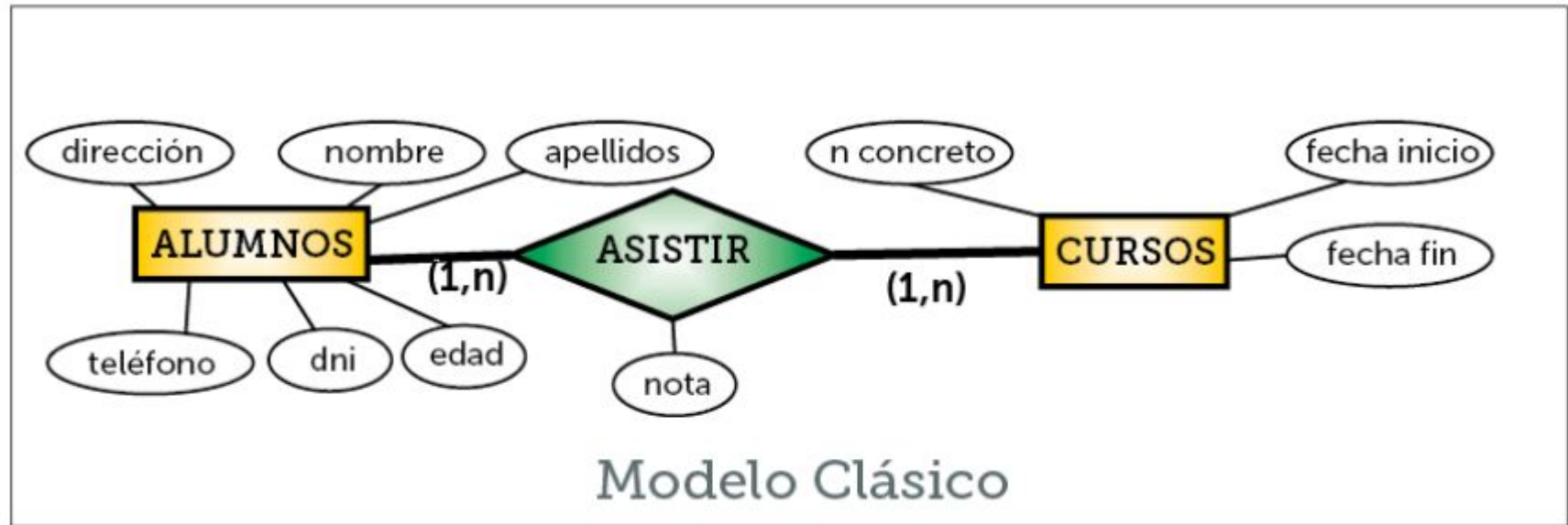
En este caso se mide con cuántas subentidades se relaciona la subentidad; es decir, si hay personal que pueda ser profesor y bedel a la vez o si sólo puede ser una cosa.

Posibilidades:

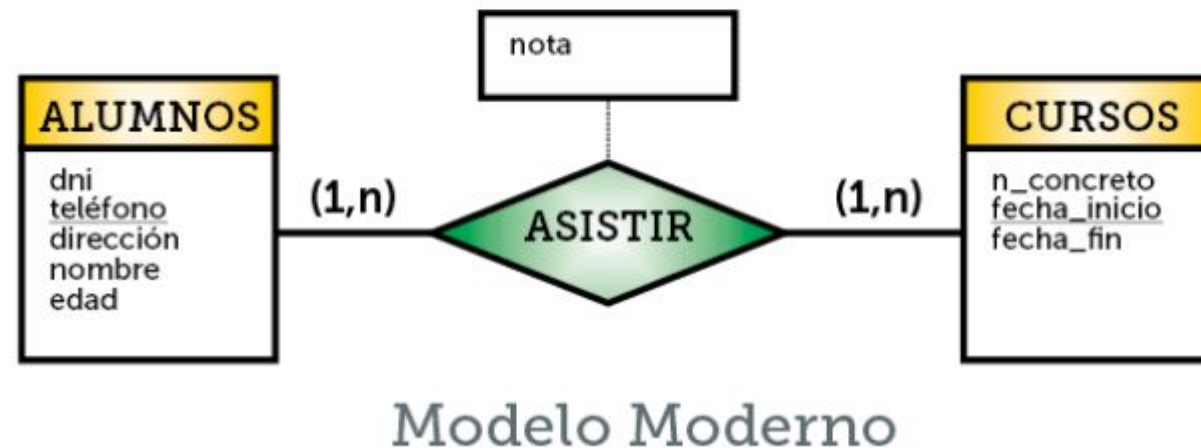
- **Relaciones de jerarquía solapada**. Indican que un ejemplar de la superentidad puede relacionarse con más de una subentidad (el personal puede ser profesor y bedel). Ocurren cuando no hay dibujado un arco de exclusividad.
- **Relaciones de jerarquía exclusiva**. Indican que un ejemplar de la superentidad sólo puede relacionarse con una subentidad (el personal no puede ser profesor y bedel). Ocurren cuando hay dibujado un arco de exclusividad.







## Otras formas de representación ER





# RETO 1: Diagrama ER



## Reto 1:

Crear un diseño entidad relación que permita gestionar los datos de una biblioteca de modo que:

- Las personas socias de la biblioteca disponen de un código de socio y además necesitan almacenar su dni, dirección, teléfono, nombre y apellidos
- La biblioteca almacena libros que presta a los socios y socias, de ellos se almacena su título, su editorial, el año en el que se escribió el libro, el nombre completo del autor (o autores), el año en que se editó y en qué editorial fue y el ISBN.
- Necesitamos poder indicar si un volumen en la biblioteca está deteriorado o no.
- Queremos controlar cada préstamo que se realiza almacenando la fecha en la que se realiza, la fecha tope para devolver (que son 15 días más que la fecha en la que se realiza el préstamo) y la fecha real en la que se devuelve el libro.



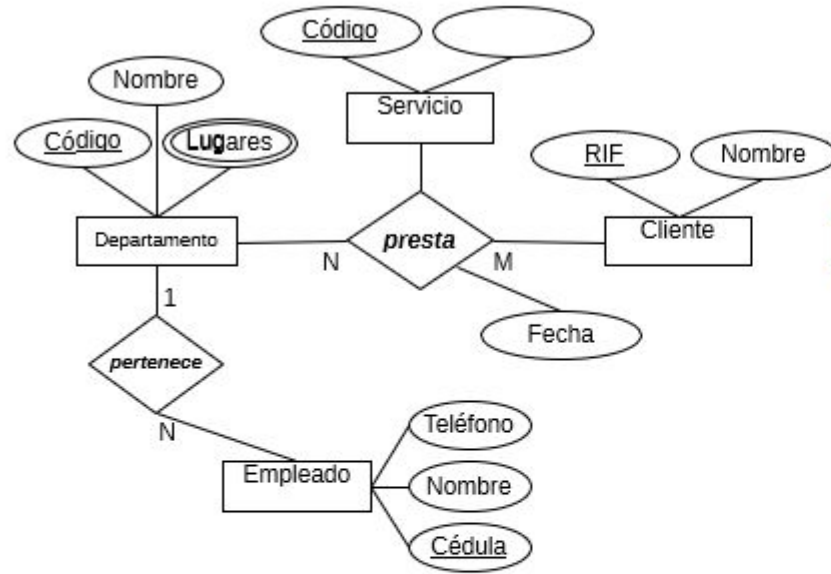


**GeeksHubs**  
academy \_

/ Desarrollamos { talento }

# Modelo Relacional





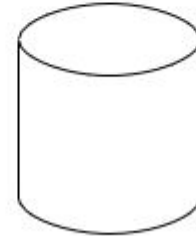
Departamento (Código, Nombre)

Empleado (Cédula, Nombre, Teléfono, CodDpto)

Cliente (RIF, Nombre)

Servicio (Código, Nombre)

Presta (CódDpto, CodServ, RIF, Fecha)



Base de Datos  
Relacional



# Terminología relacional

**Tupla.** Cada fila de la tabla (cada ejemplar que la tabla representa)

- **Atributo.** Cada columna de la tabla
- **Grado.** Número de atributos de la tabla
- **Cardinalidad.** Número de tuplas de una tabla
- **Dominio.** Conjunto válido de valores representables por un atributo.

**NOMBRE**

atributo 1	atributo 2	atributo 3	....	atributo n	
valor 1,1	valor 1,2	valor 1,3	....	valor 1,n	← tupla 1
valor 2,1	valor 2,2	valor 2,3	....	valor 2,n	← tupla 2
.....	.....	.....	....	.....	....
valor m,1	valor m,2	valor m,3	....	valor m,n	← tupla m



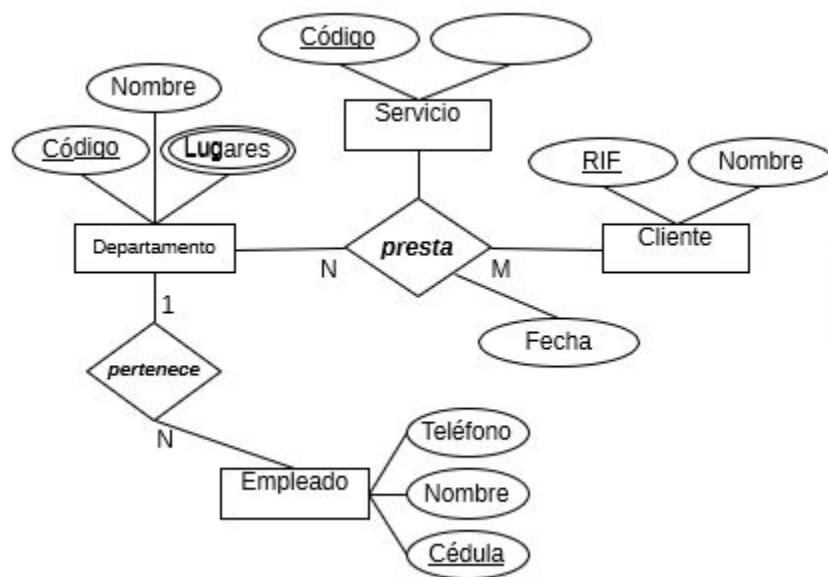
# Paso del esquema ER al Modelo Relacional

En principio las entidades fuertes del modelo Entidad Relación son transformados al modelo relacional siguiendo estas instrucciones:

- **Entidades.** Las entidades pasan a ser tablas.
- **Atributos.** Los atributos pasan a ser columnas.
- **Identificadores principales.** Pasan a ser claves primarias



# Paso del esquema ER al Modelo Relacional



Departamento (Código, Nombre)

Empleado (Cédula, Nombre, Teléfono, CodDpto)

Cliente (RIF, Nombre)

Servicio (Código, Nombre)

Presta (CódDpto, CodServ, RIF, Fecha)



# Paso del esquema ER al Modelo Relacional

La idea inicial es transformar a cada relación en una tabla en el modelo relacional. Pero hay que distinguir según el tipo de relación.

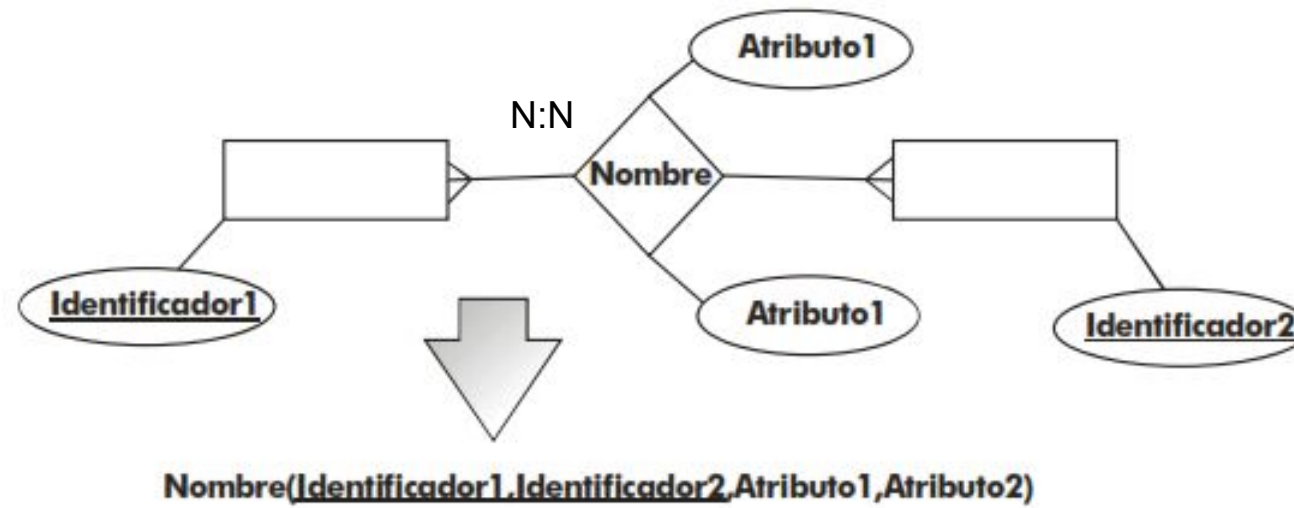
- relaciones varios a varios N:M
- relaciones de orden  $N$
- relaciones uno a varios y uno a uno 1:N y 1:1
- relaciones recursivas



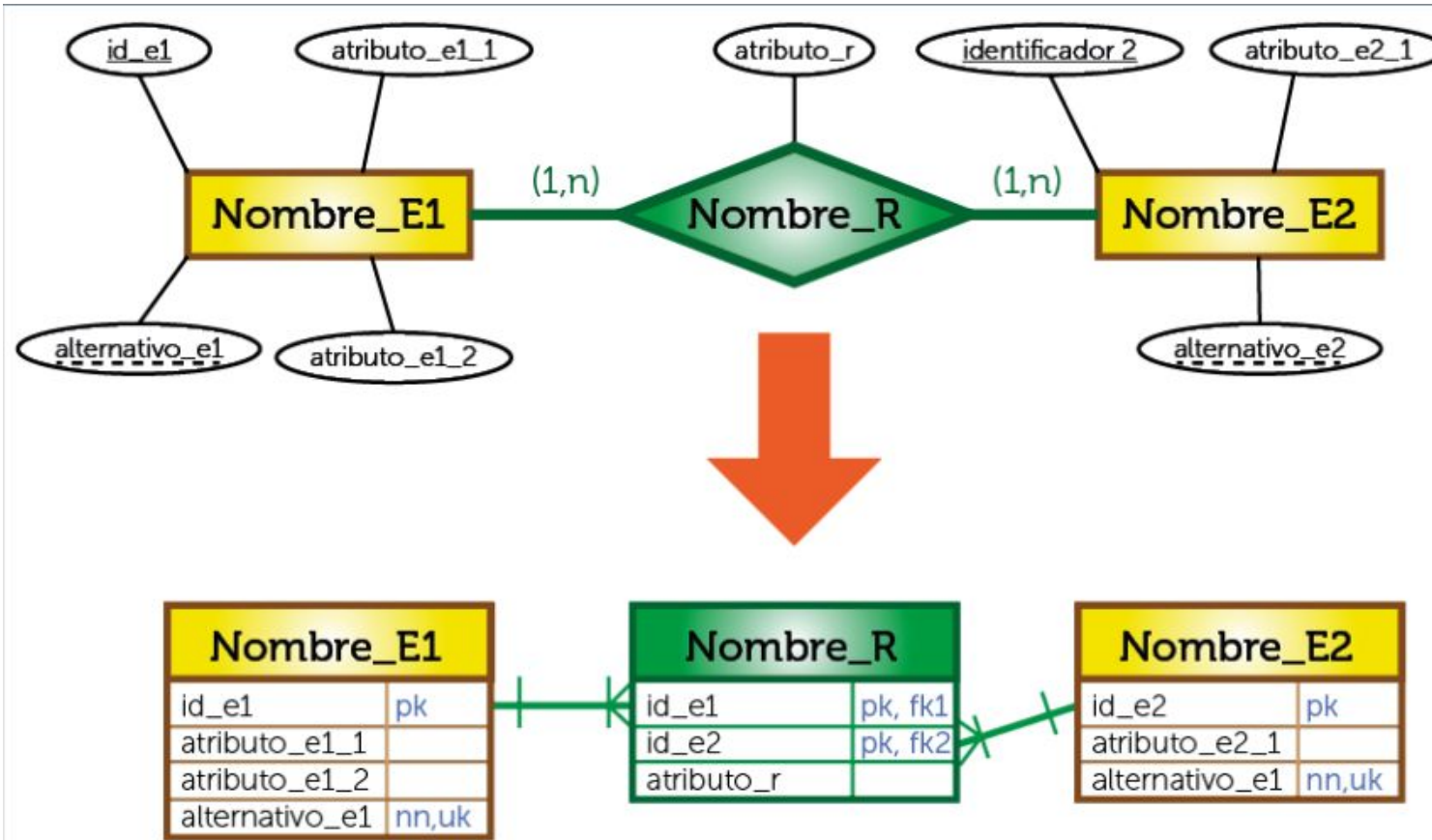
# Relaciones varios a varios

En las relaciones varios a varios, la relación se transforma en una tabla cuyos atributos son:

1. Los atributos de la relación y las claves de las entidades relacionadas (que pasarán a ser claves externas).
2. La clave primaria de la tabla la forman todas las claves externas.

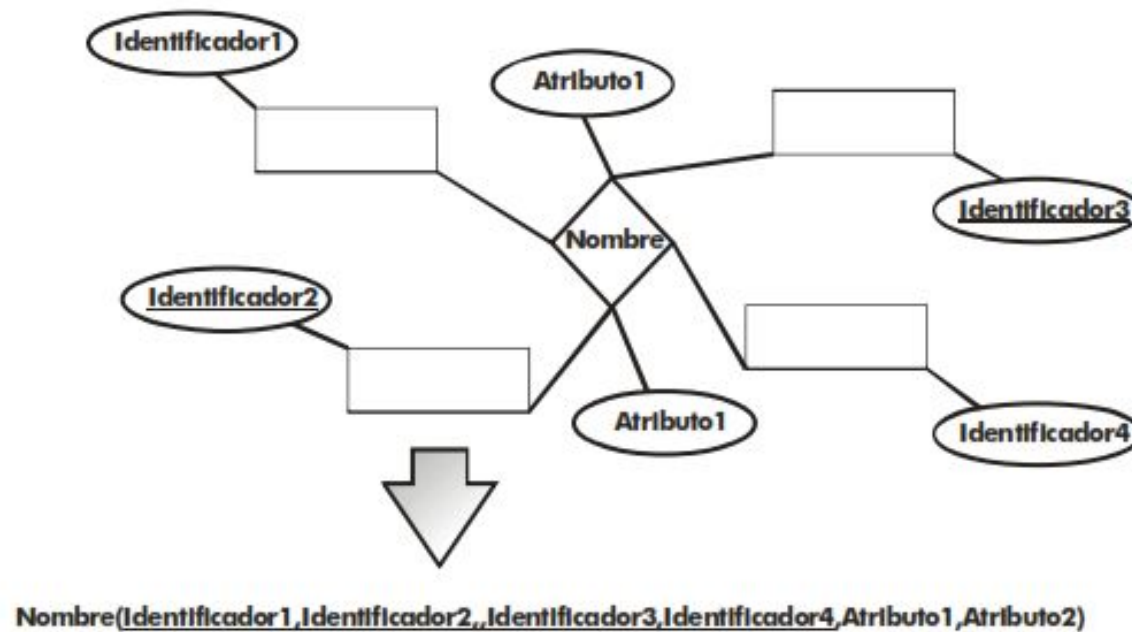


# Relaciones varios a varios N:N

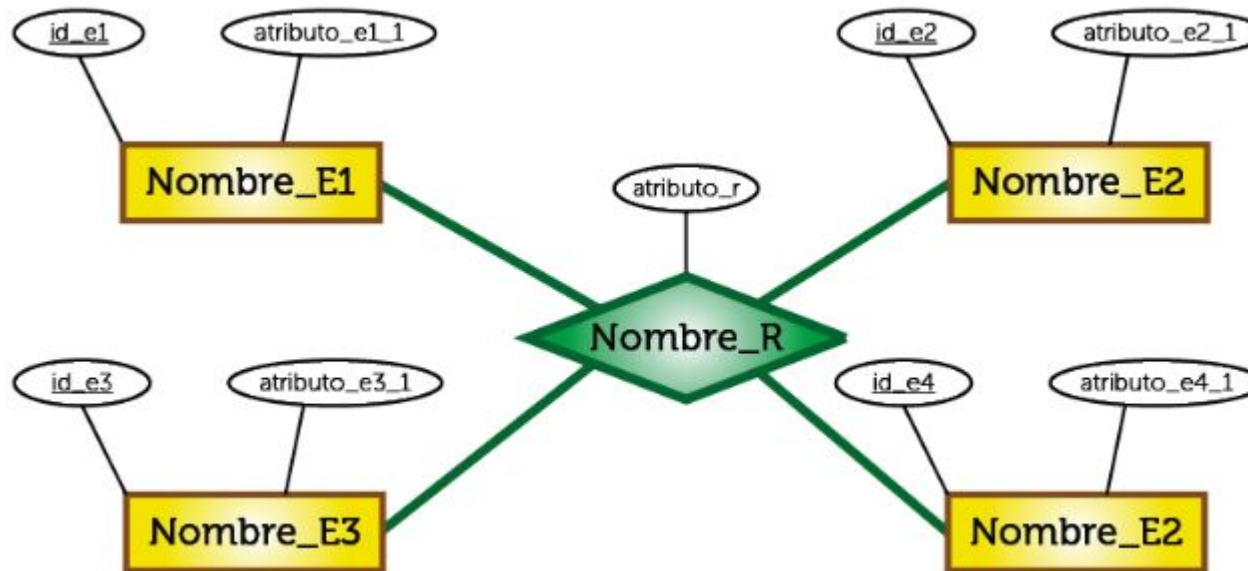


# Relaciones de orden N

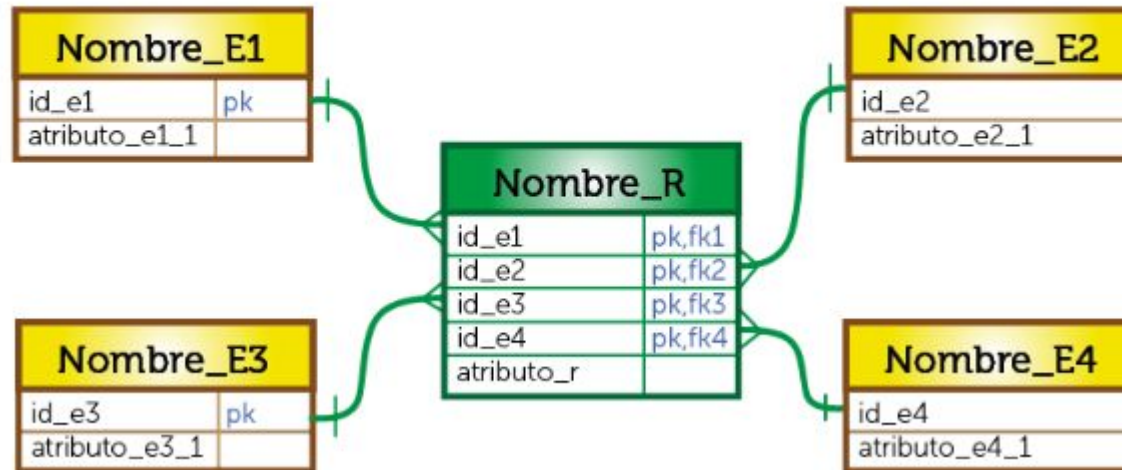
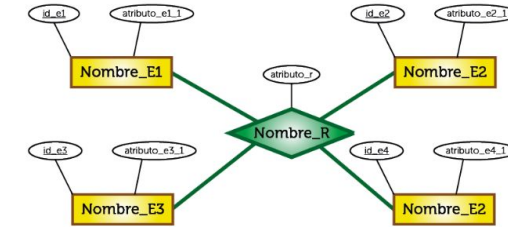
Las relaciones ternarias, cuaternarias y *n-arias* que *unen más de dos relaciones* se transforman en una tabla que contiene los atributos de la relación más los identificadores de las entidades relacionadas. La clave la forman todas las claves externas:



## Relaciones de orden N (Ternaria,...)

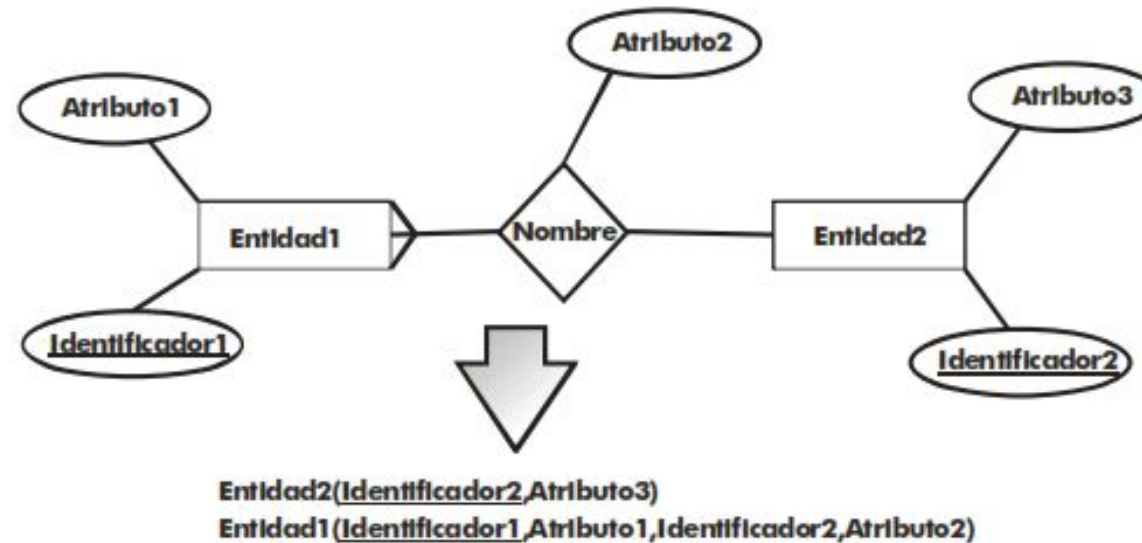


# Relaciones de orden N (Ternaria,...)

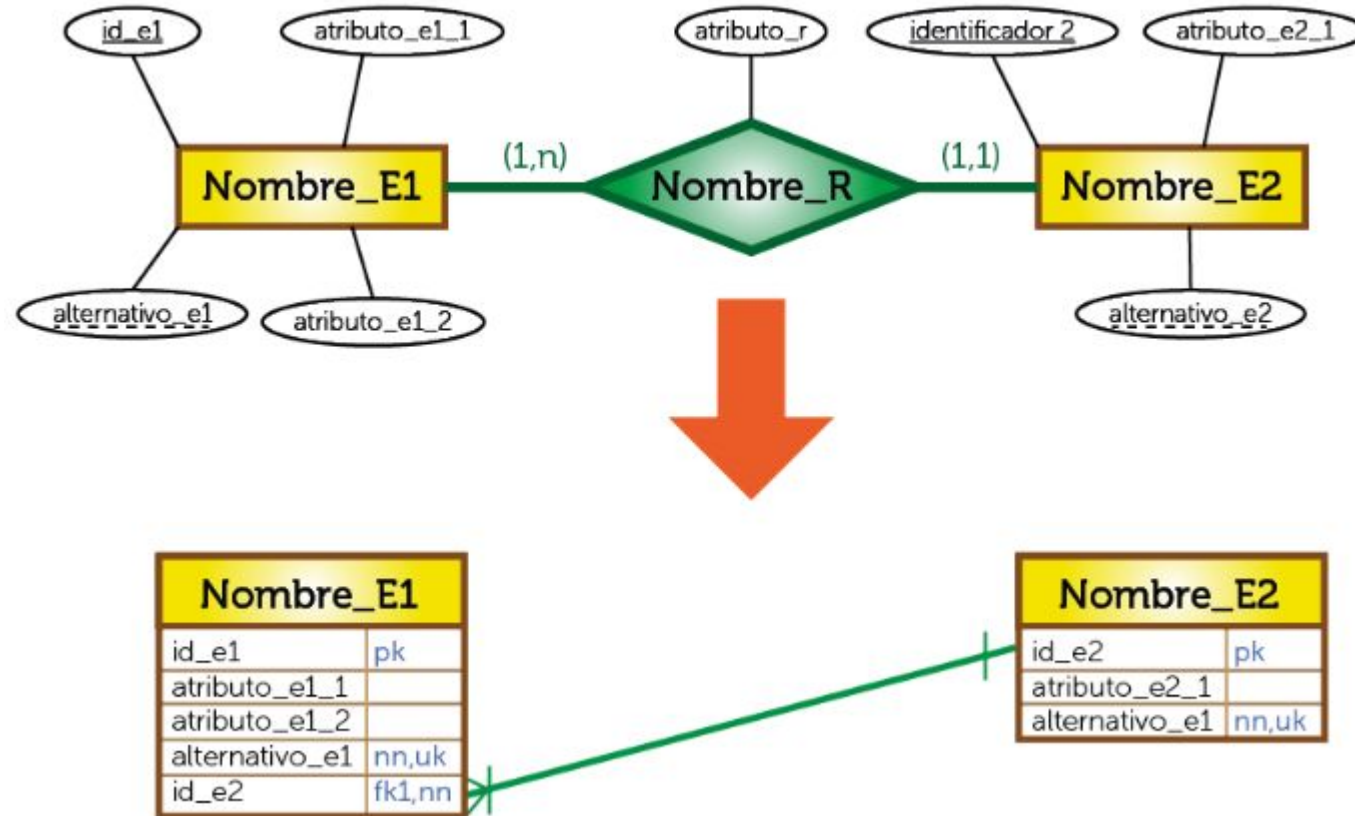


# Relaciones de 1:N y 1:1

Las relaciones binarios de tipo uno a varios no requieren ser transformadas en una tabla en el modelo relacional. En su lugar la tabla del lado varios (**tabla relacionada**) **incluye** como clave externa 1 el identificador de la entidad del lado uno (**tabla principal**):

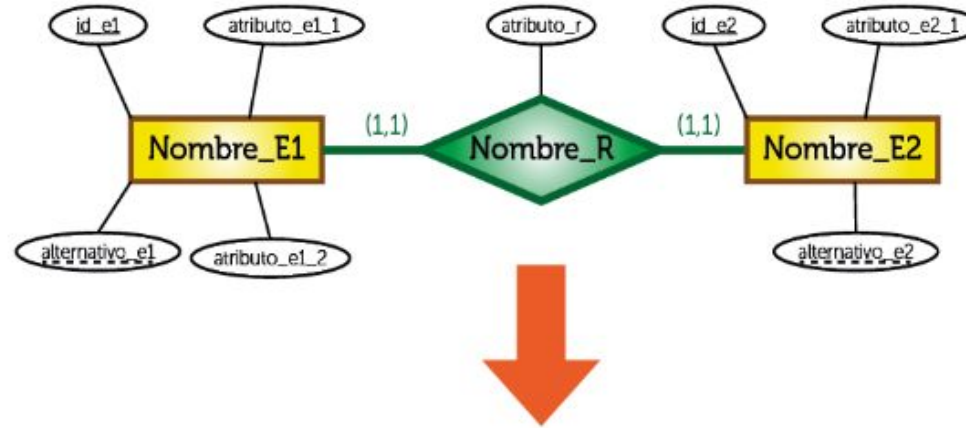


# Relaciones de 1:N





# Relaciones de 1:1



**Solución 1**

Nombre_E1	
id_e1	pk
atributo_e1_1	
atributo_e1_2	
alternativo_e1	nn,uk1
atributo_r	
id_e2	uk2,nn
atributo_e2_1	
alternativo_e2	uk3,nn

**Solución 2**

Nombre_E1	
id_e1	pk
atributo_e1_1	
atributo_e1_2	
alternativo_e1	nn,uk1
id_e2	fk,uk2,nn
atributo_r	

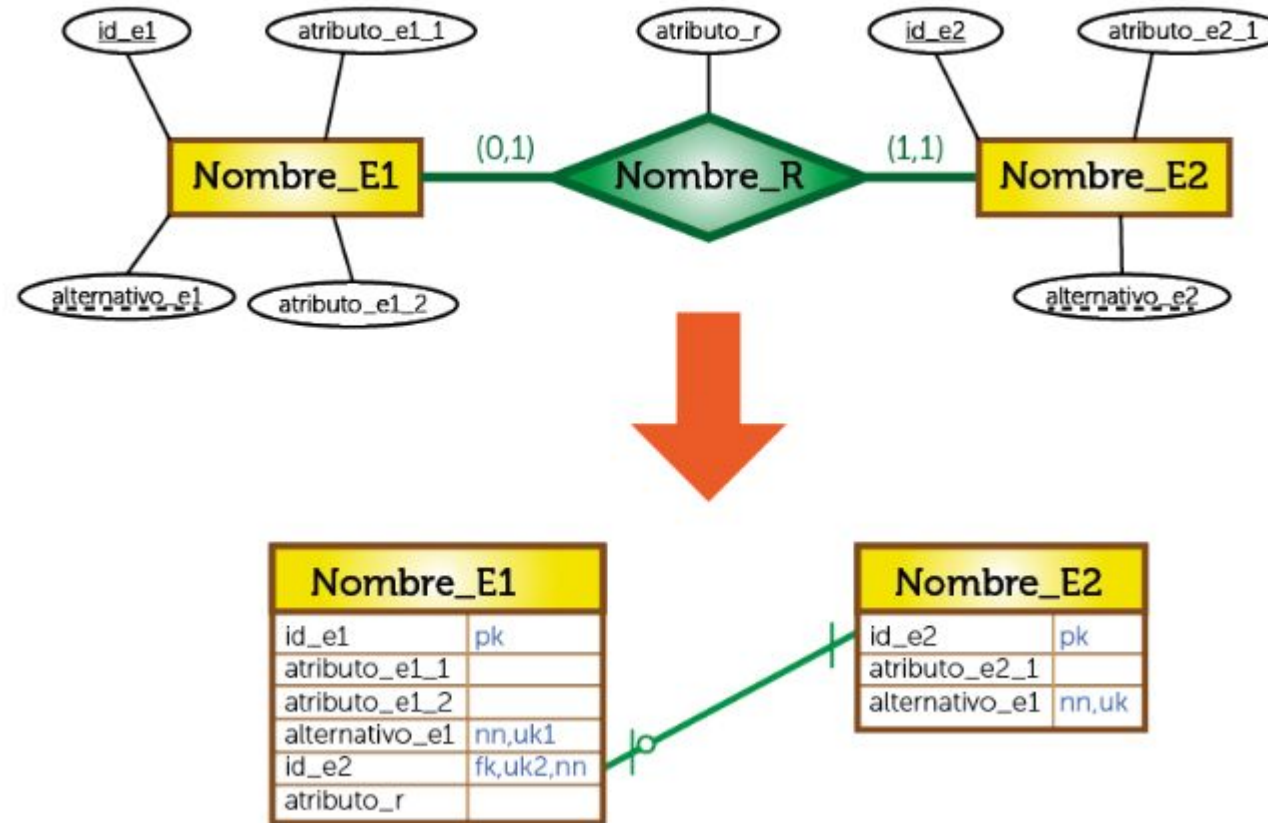
Nombre_E2	
id_e2	pk
atributo_e2_1	
alternativo_e1	nn,uk

**Solución 3**

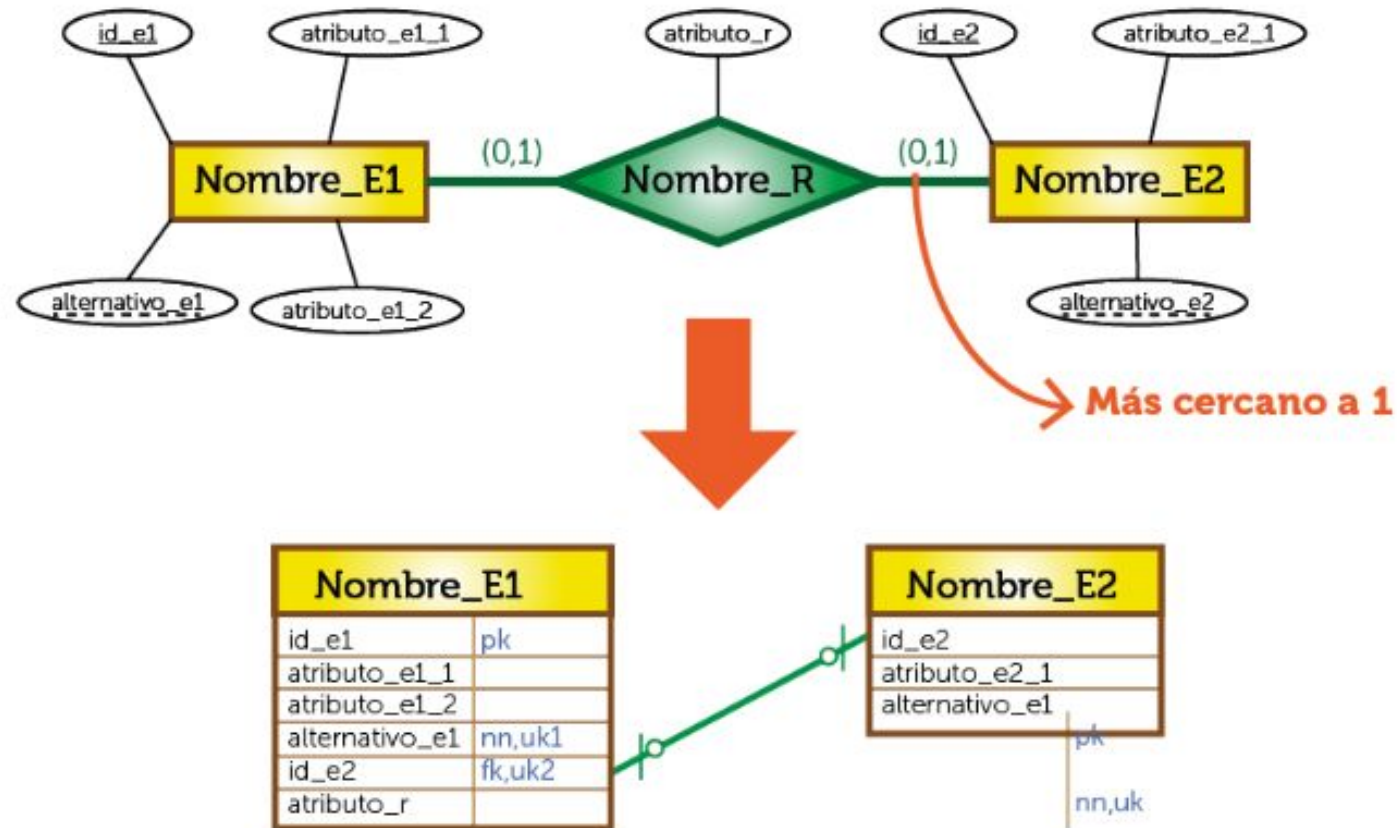
Nombre_E1	
id_e1	pk
atributo_e1_1	
atributo_e1_2	
alternativo_e1	nn,uk1

Nombre_E2	
id_e2	pk
atributo_e2_1	
alternativo_e1	nn,uk1
atributo_r	
id_e1	nn,fk,uk2

# Relaciones de 0:1

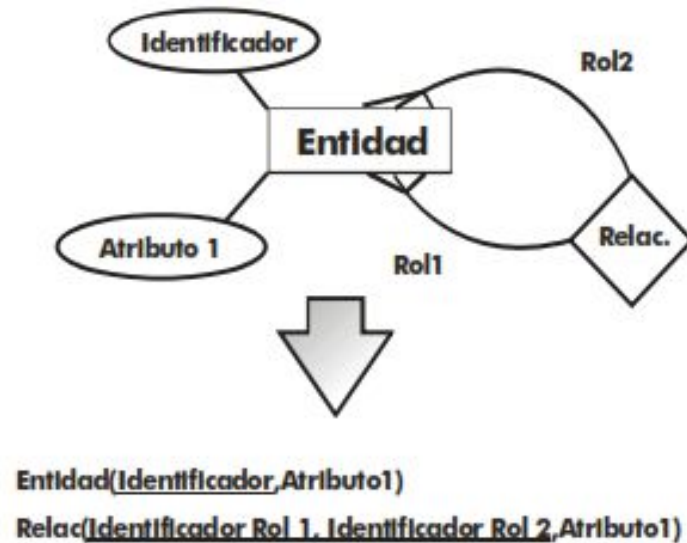
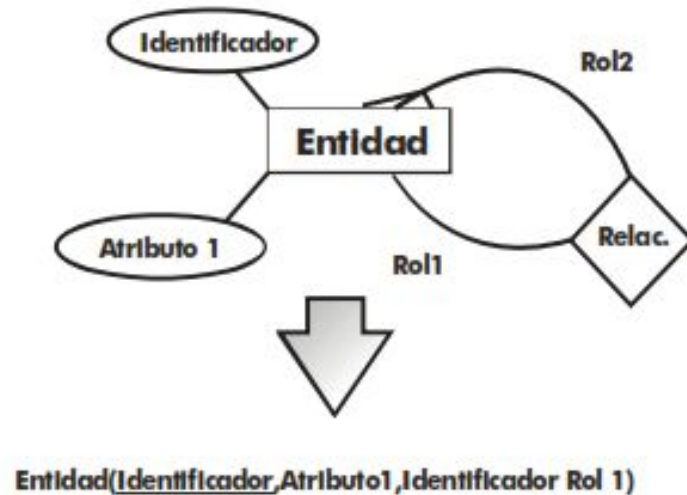


# Relaciones de 0:0

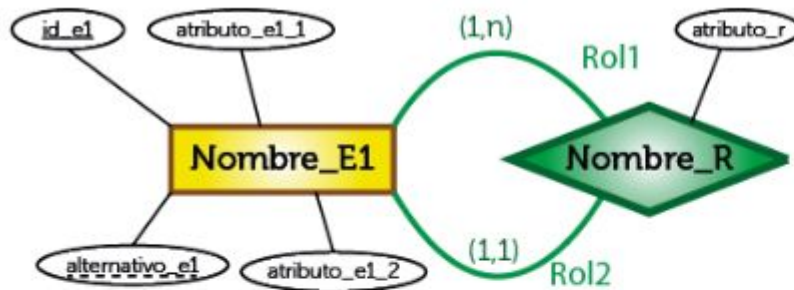


# Relaciones recursivas

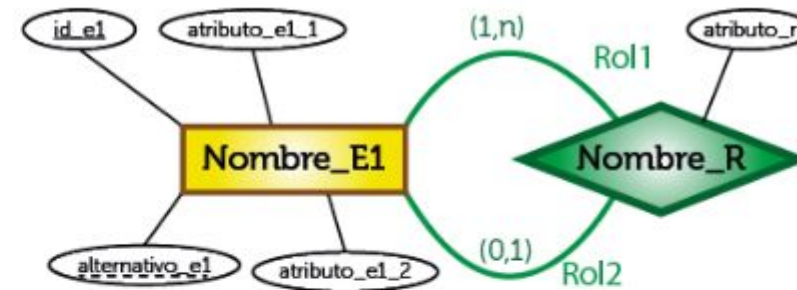
Las relaciones recursivas se tratan de la misma forma que las otras, sólo que un mismo atributo puede figurar dos veces en una tabla como resultado de la transformación:



# Relaciones recursivas 1:N



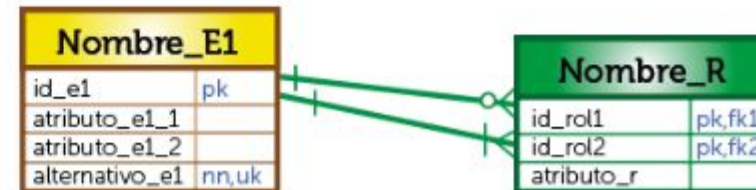
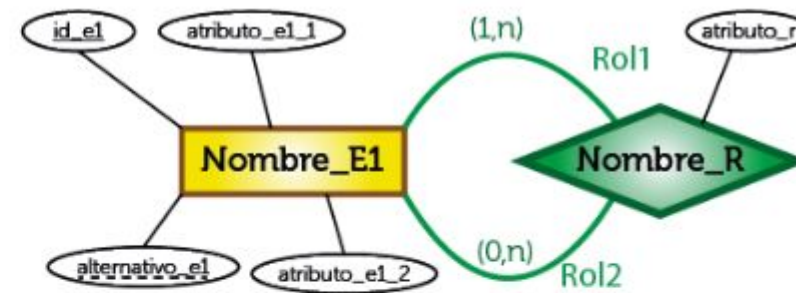
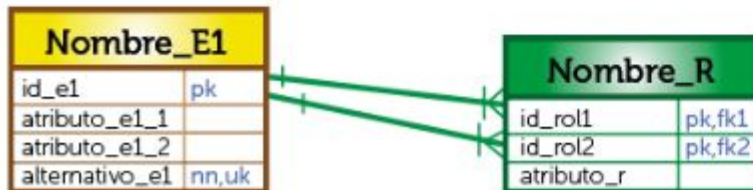
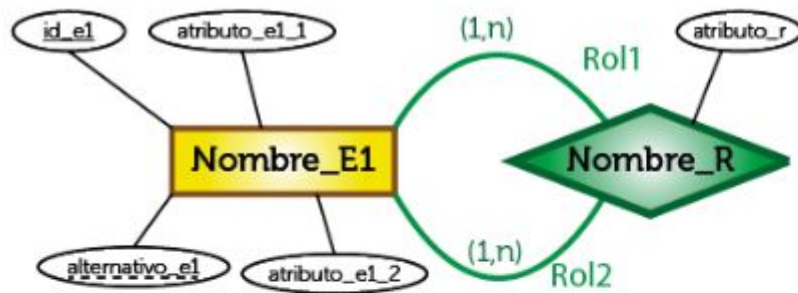
Nombre_E1	
id_e1	pk
atributo_e1_1	
atributo_e1_2	
alternativo_e1	nn,uk
atributo_r	
id_e1_rol2	nn,fk



Nombre_E1	
id_e1	pk
atributo_e1_1	
atributo_e1_2	
alternativo_e1	nn,uk
atributo_r	
id_e1_rol2	fk

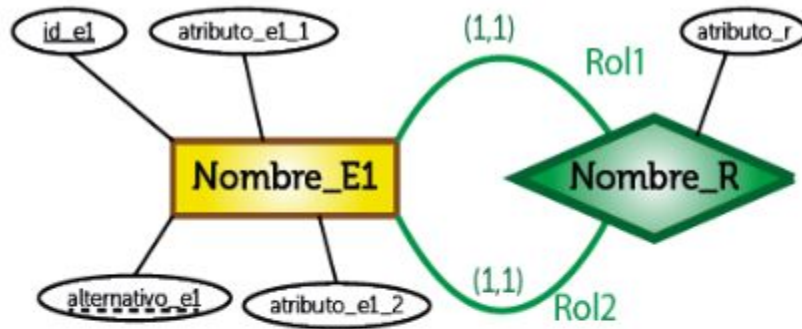


# Relaciones recursivas N:N

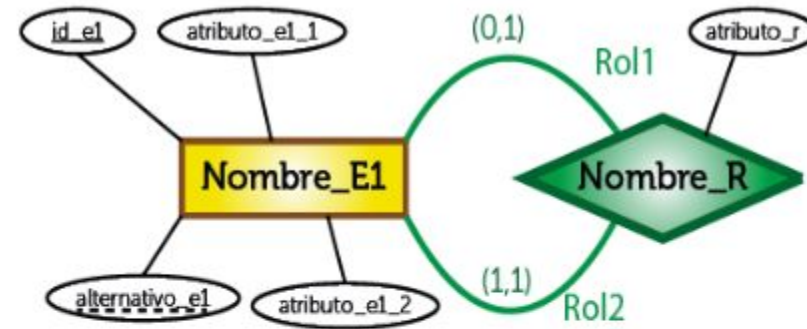




# Relaciones recursivas 1:1 o 0:1



Nombre_E1	
id_e1	pk
atributo_e1_1	
atributo_e1_2	
alternativo_e1	nn,uk
atributo_r	
id_e1_rol2	nn,fk,uk

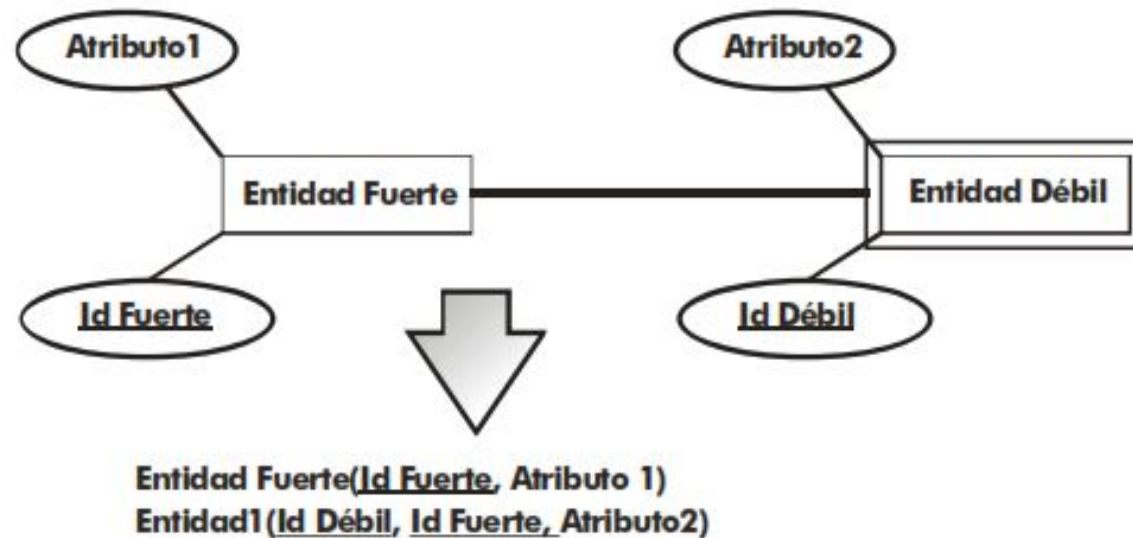


Nombre_E1	
id_e1	pk
atributo_e1_1	
atributo_e1_2	
alternativo_e1	nn,uk
atributo_r	
id_e1_rol2	nn,fk,uk



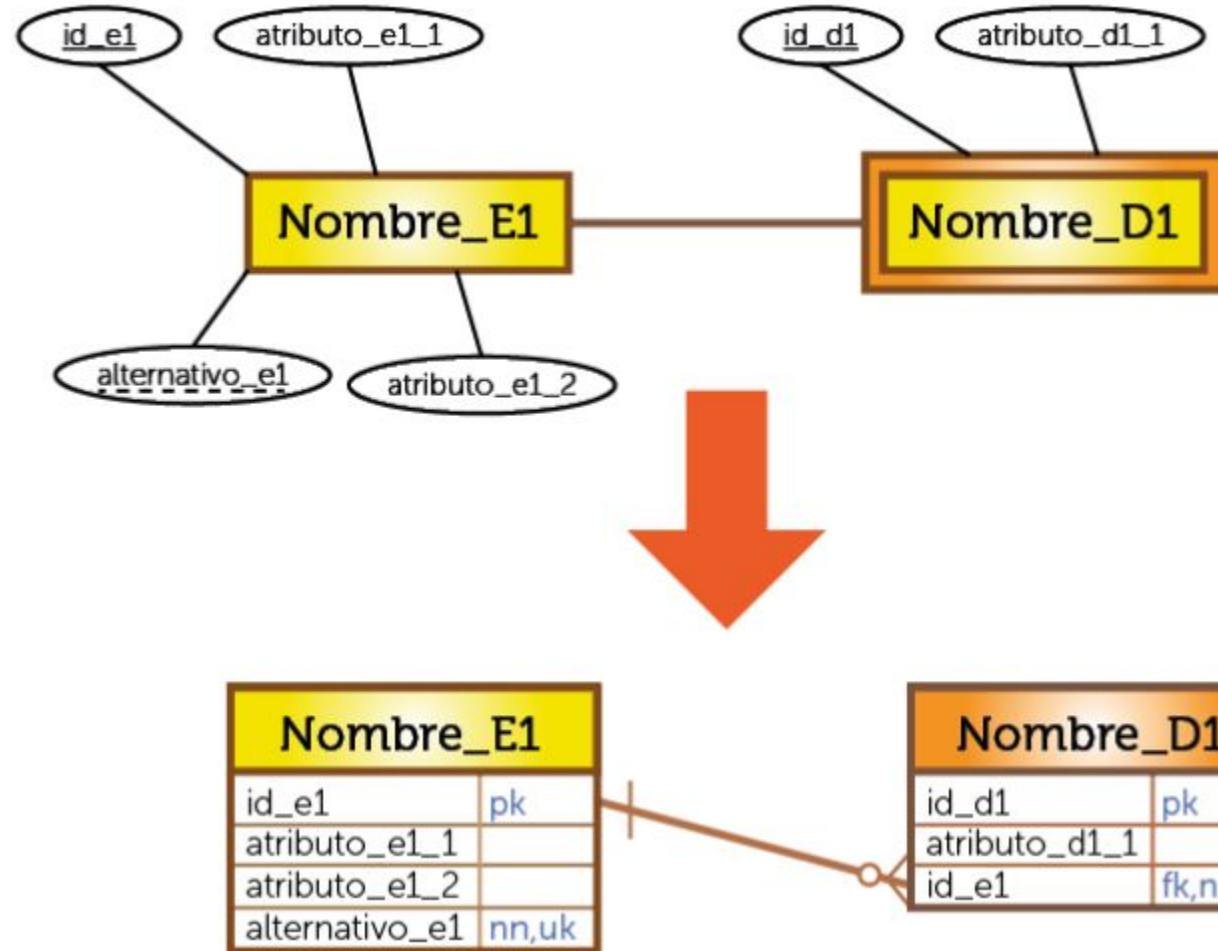
# Entidades débiles

Toda entidad débil incorpora una relación implícita con una entidad fuerte. Esta relación no necesita incorporarse como tabla en el modelo relacional. Sí se necesita incorporar la clave de la entidad fuerte como clave externa en la entidad débil. Es más, normalmente esa clave externa forma parte de la clave principal de la tabla que representa a la entidad débil. El proceso es:





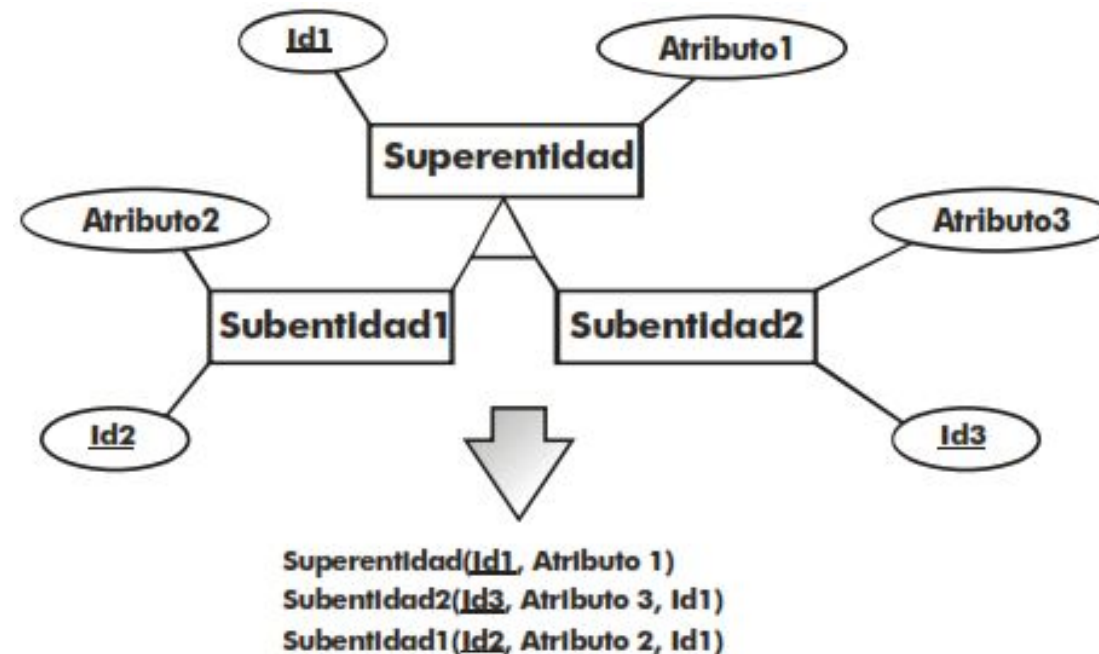
# Entidades débiles



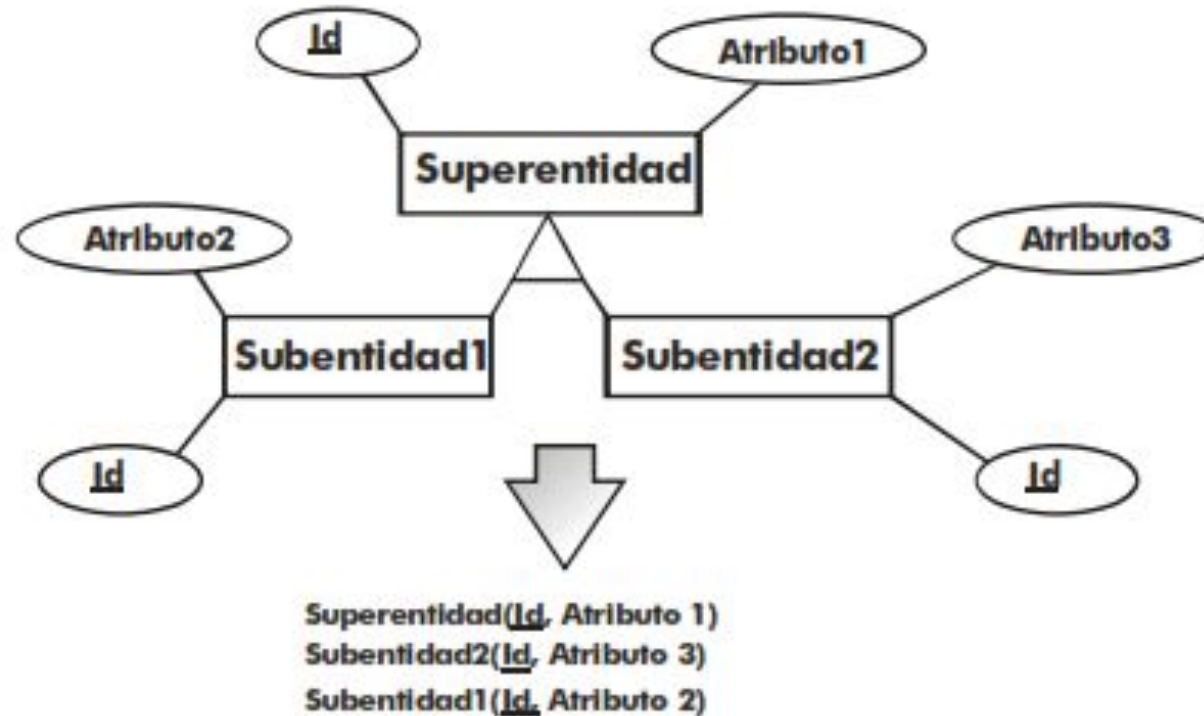
# Generalizaciones y especificaciones

Las generalizaciones y/o especificaciones se convierten al modelo relacional de esta forma:

- **1- Las subentidades pasan a ser tablas.**
- **2- Si la clave de la superentidad es distinta de las subentidades, entonces se coloca el identificador de la superentidad en cada subentidad como clave externa:**

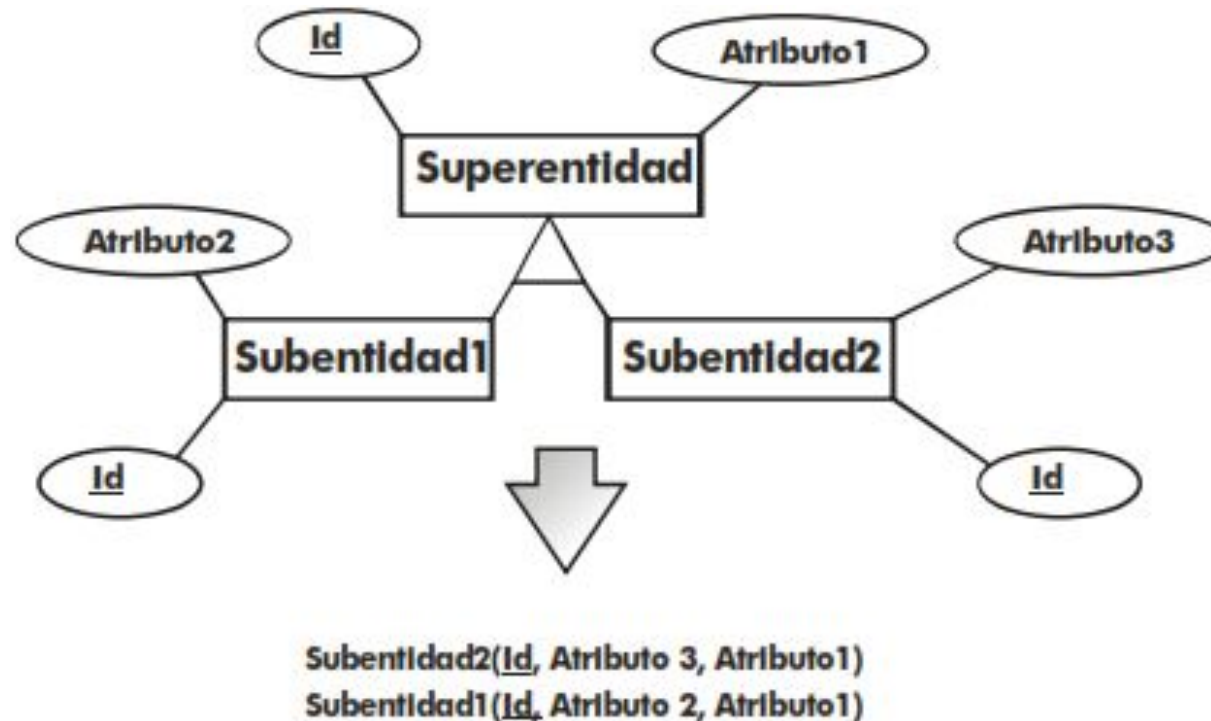


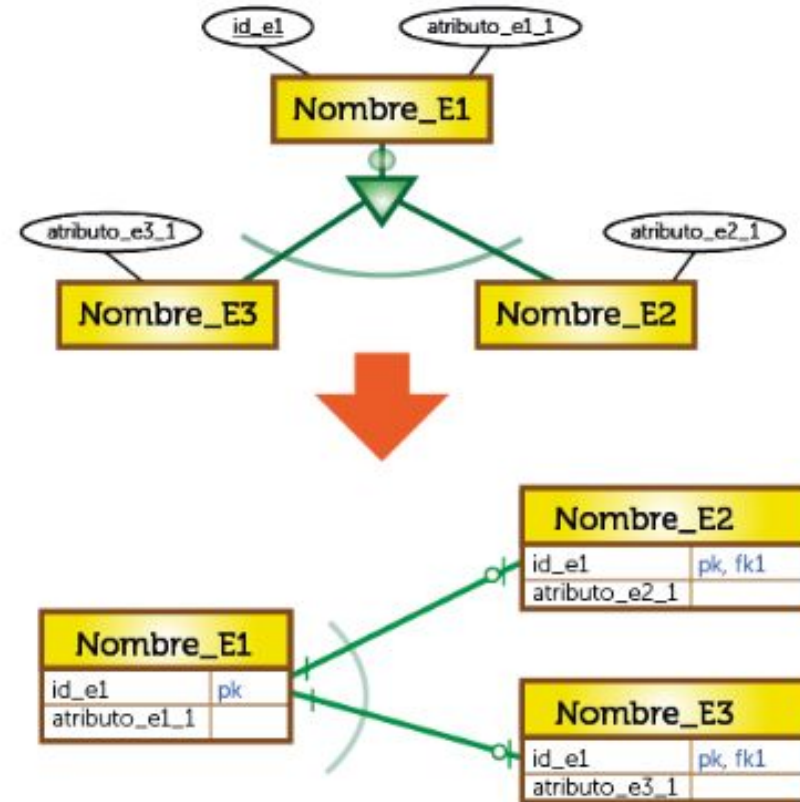
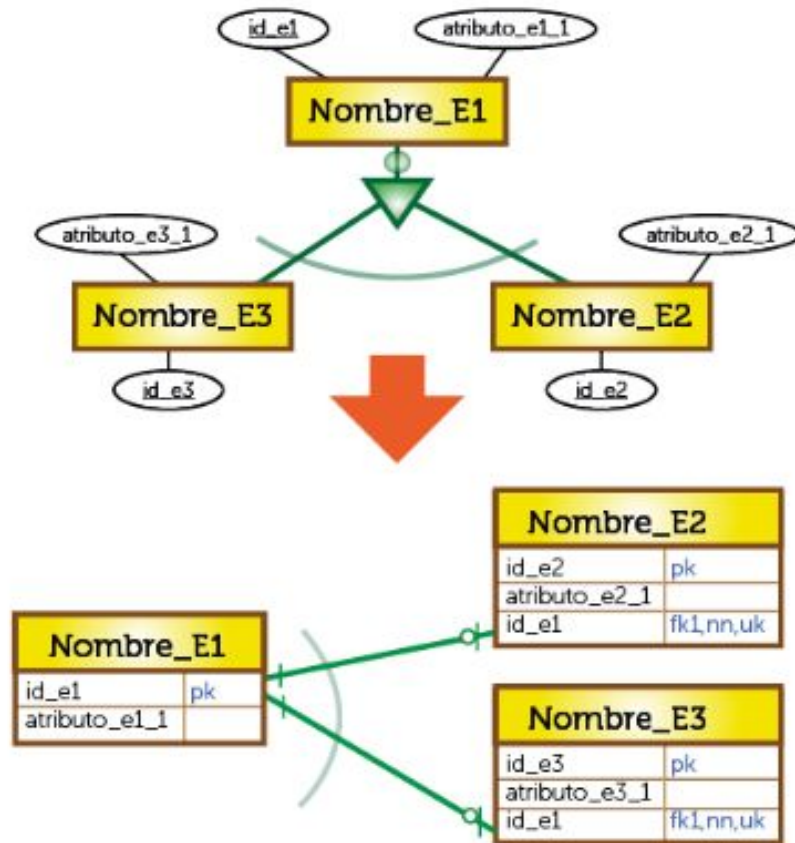
3- Si la clave es la misma, entonces todas las entidades tendrán la misma columna como identificador:



# Generalizaciones y especificaciones

4- La superentidad debe generar una tabla sólo en el caso de que haya posibilidad de que exista un ejemplar de dicha entidad que no sea ejemplar de las subentidades. De otro modo basta con generar las tablas de las subentidades e incluir los atributos de la entidad superior:





# Jose Marín

[www.geekshubsacademy.com](http://www.geekshubsacademy.com)

@geekshubs

