

# UD 3. MODELO RELACIONAL

## Parte 1. Paso a tablas desde el modelo conceptual

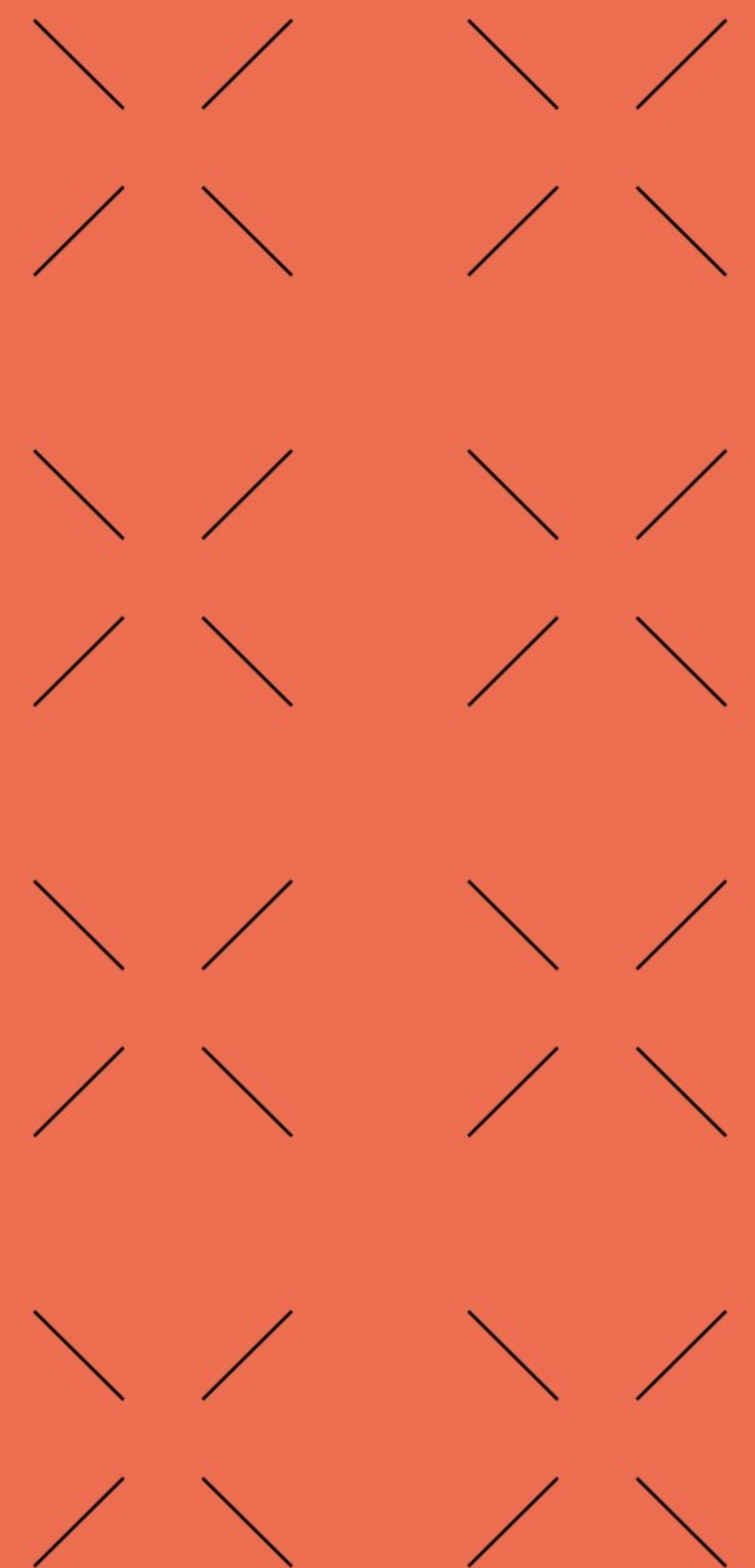
**Bases de Datos (DAW/DAM)**

**CFGS Desarrollo de Aplicaciones Web (DAW)**

**CFGS Desarrollo de Aplicaciones Multiplataforma (DAM)**

**Abelardo Martínez y Pau Miñana**

**Curso 2023-2024**



# Créditos



- Apuntes realizados por Abelardo Martínez y Pau Miñana.
- Basados y modificados de Sergio Badal ([www.sergiobadal.com](http://www.sergiobadal.com)) y Raquel Torres.
- Las imágenes e iconos empleados están protegidos por la licencia [LGPL](#) y se han obtenido de:
  - [https://commons.wikimedia.org/wiki/Crystal\\_Clear](https://commons.wikimedia.org/wiki/Crystal_Clear)
  - <https://www.openclipart.org>

# Contenidos

¿Qué veremos en esta parte?



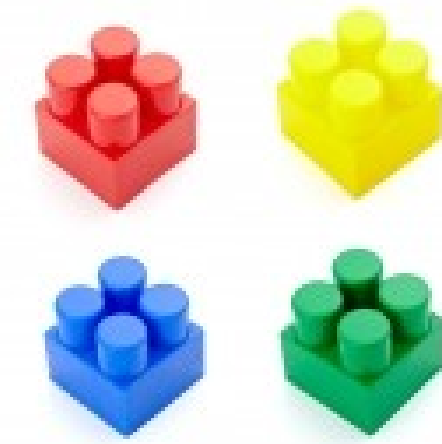
- Modelado lógico
- Elementos
- Restricciones
- Transformaciones
- Boletín A: diagramas de complejidad básica (resueltos)
- Actividades propuestas (no evaluables)
  - Boletín B: diagramas de complejidad media

# Contenidos

1. MODELADO LÓGICO
2. ELEMENTOS
3. RESTRICCIONES
  1. Restricciones de clave
  2. Restricciones de valor
  3. Notaciones equivalentes
4. PÉRDIDAS SEMÁNTICAS
5. PASO DEL MODELO CONCEPTUAL AL MODELO LÓGICO
  1. Transformación de entidades
  2. Transformación de relaciones
  3. Transformación de especializaciones
6. BIBLIOGRAFÍA



# 1. MODELADO LÓGICO



# Modelado lógico

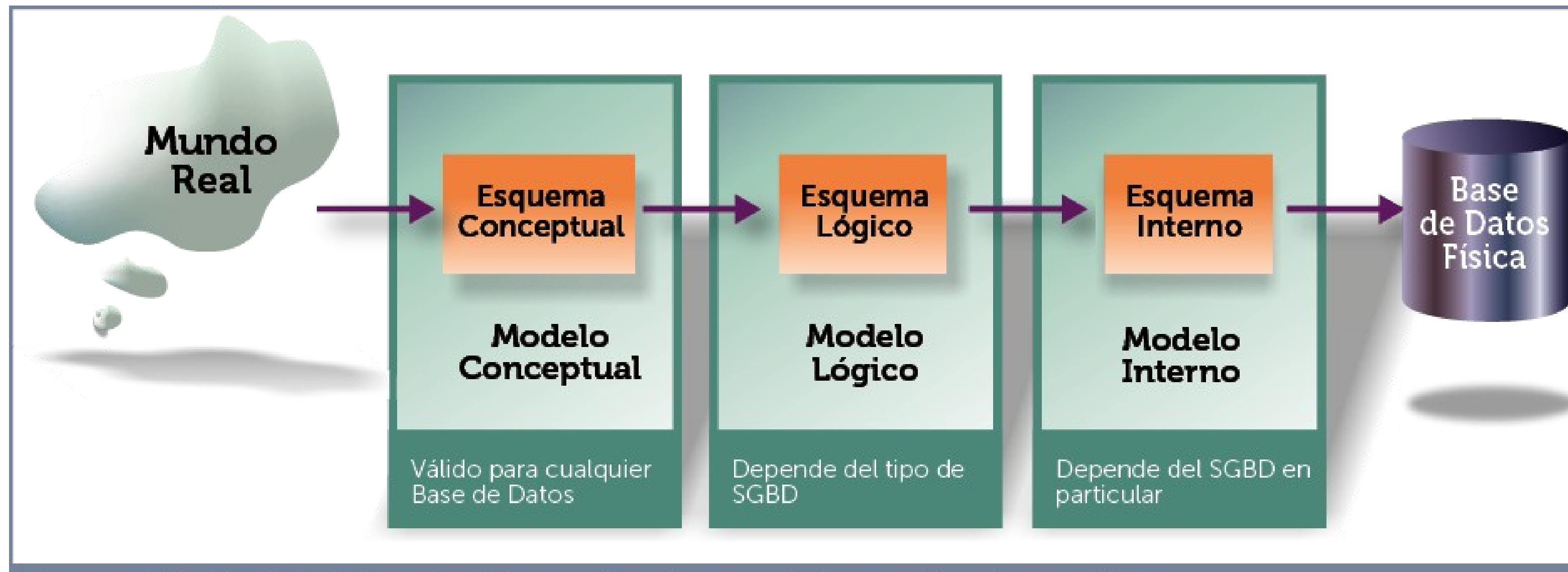
Una vez creado el **modelo conceptual de datos** (modelo entidad-relación), tal y como vimos en el tema anterior, procederemos a crear el **modelo lógico** de datos a partir del cual obtendremos el **modelo físico de datos** que será el que realmente utilizaremos para crear la base de datos.

El modelo conceptual de datos es independiente del Sistema Gestor de Base de Datos (SGBD), pero **el modelo lógico sí es dependiente del TIPO de SGBD** que vamos a utilizar, pudiendo ser este relacional, no-relacional o de otro tipo.

Del mismo modo, **el modelo físico de datos también será dependiente del SGBD** concreto con el que vayamos a trabajar.

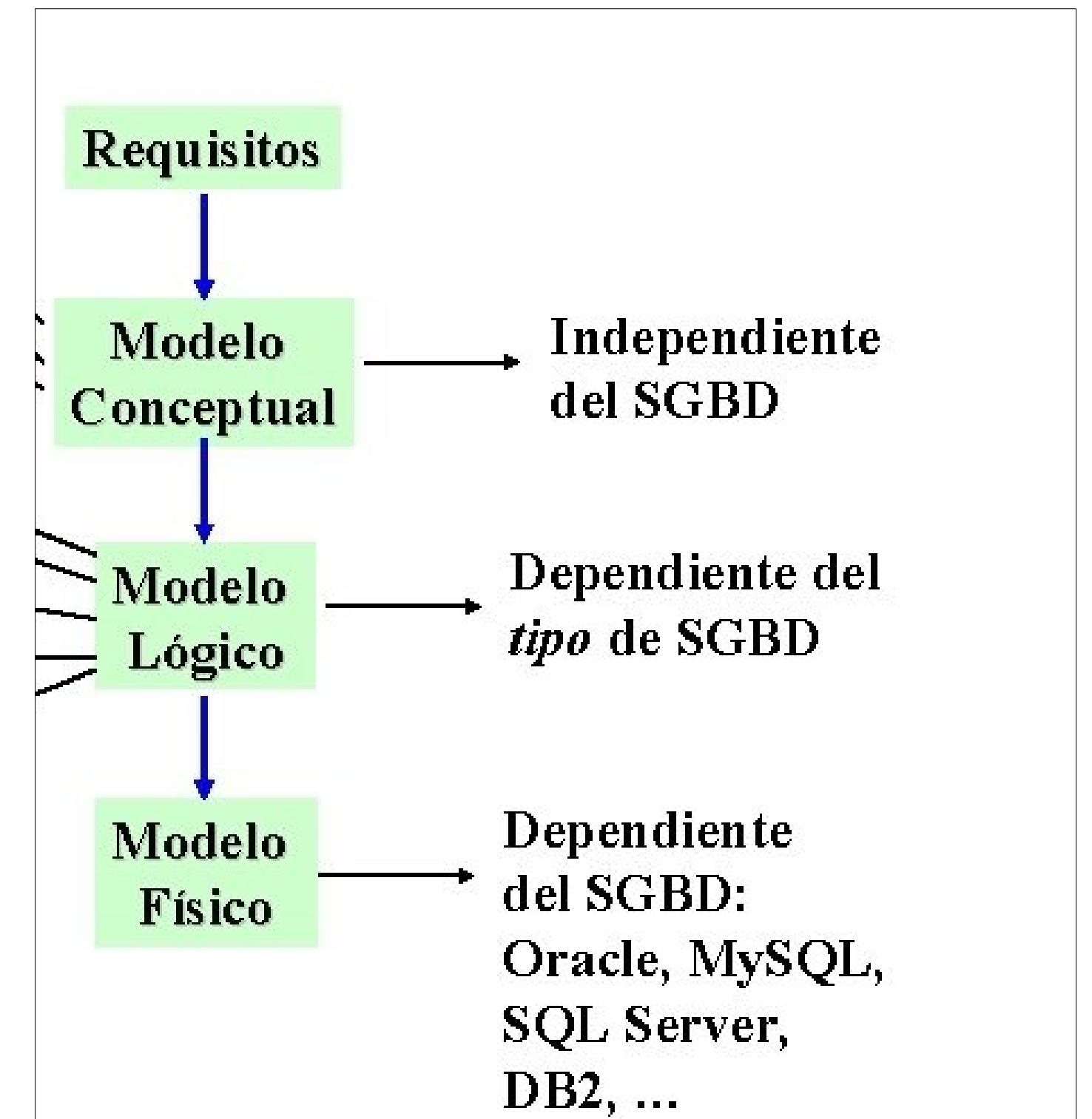
# Pasos a seguir

Fases del diseño de bases de datos:



Información complementaria:

[https://www.tuinstitutoonline.com/cursos/baseavanzado1\\_v1606/05paso\\_er\\_relacional.php](https://www.tuinstitutoonline.com/cursos/baseavanzado1_v1606/05paso_er_relacional.php)



# Fases del diseño de bases de datos

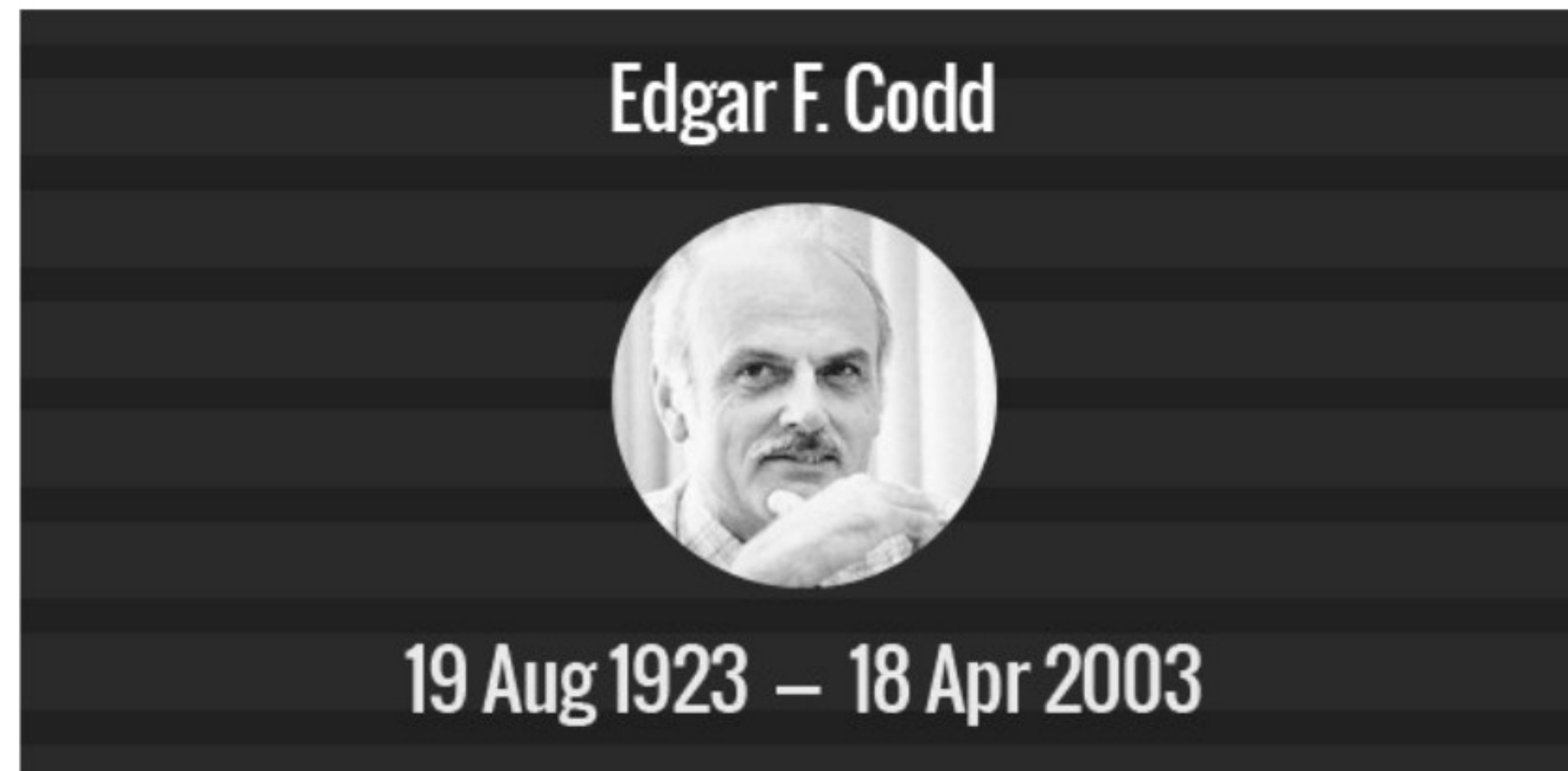
Nivel de abstracción	Fase del modelado	Dependencia con el SGBD	Técnica más usada
ALTA	REALIDAD		
	Conceptual	<ul style="list-style-type: none"><li>• Ninguna dependencia</li><li>• <i>Modelo 100% teórico</i></li></ul>	<b>MODELO E-R</b> DE PETER CHEN
	Lógico	<ul style="list-style-type: none"><li>• Dependencia con tipo de SGBD.</li><li>• Relacional, jerárquico, objetos, no-relacional, etc.</li></ul>	<b>MODELO RELACIONAL</b> DE EDGAR CODD
	Físico	Con el SGBD concreto. <ul style="list-style-type: none"><li>• Relacional: MariaDB, Oracle, MySQL, PostgreSQL, etc.</li><li>• No-relacional: MongoDB, Cassandra, etc.</li></ul>	Especificaciones de cada SGBD
BAJA	SISTEMA DE INFORMACIÓN		



# Modelo relacional

El modelo relacional de bases de datos es un **método para estructurar información utilizando relaciones, mediante estructuras en forma de cuadrícula**, que se componen de columnas y filas. Es el principio conceptual de las bases de datos relacionales.

Fue propuesto por **Edgar F. Codd** a finales de los 60 y publicado en un artículo en 1970 y desde entonces se ha convertido en el modelo de base de datos dominante para las aplicaciones comerciales, si se compara con otros modelos de bases de datos, como el jerárquico, de red y de objetos.



## **2. ELEMENTOS**

# Relación. Equivalencias

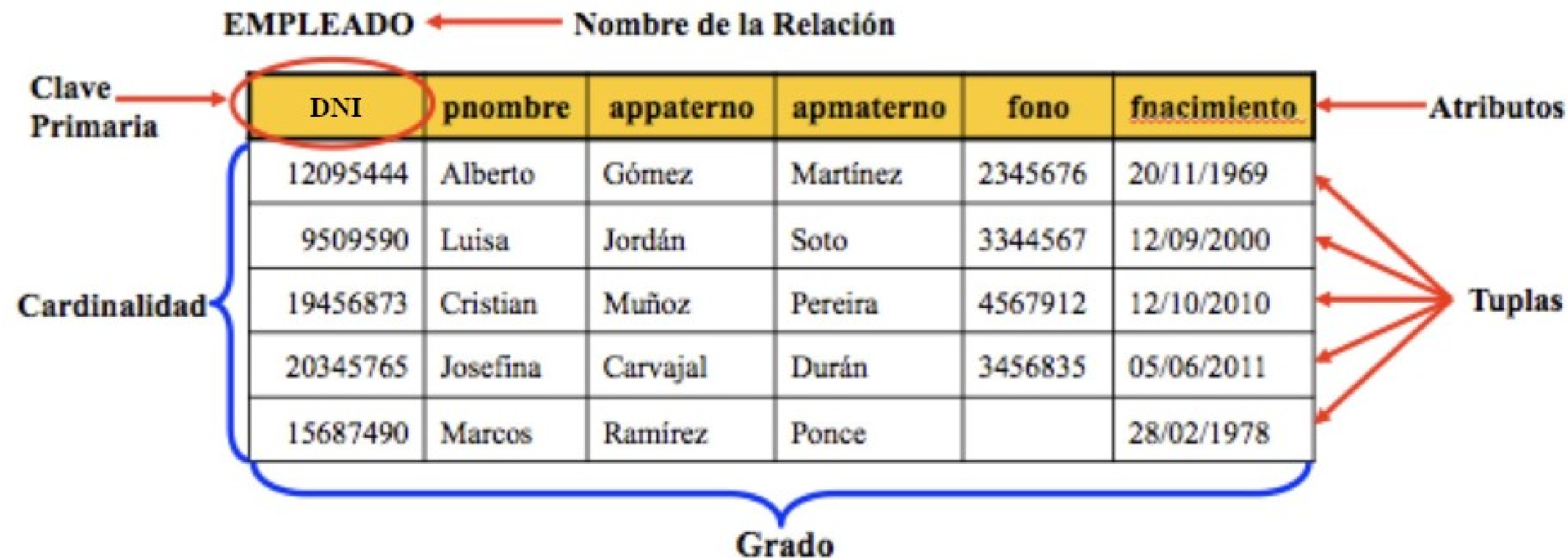
La relación es el elemento básico de este modelo. Las **entidades y relaciones del E-R se transforman en relaciones** del modelo relacional (llamadas también **TABLAS**), teniendo las siguientes equivalencias:

MODELO ENTIDAD RELACIÓN (P. Chen)	MODELO RELACIONAL (E. Codd)
Relación	Relación o Tabla
Entidad	Relación o Tabla
Atributo	Campo o columna o atributo
Ocurrencia	Tupla o fila

Terminología Relacional		Terminología de Tablas		Terminología de Archivo
Relación	=	Tabla	=	Archivo
Tupla	=	Fila	=	Registro
Atributo	=	Columna	=	Campo
Grado	=	Número de columnas	=	Número de campos
Cardinalidad	=	Número de filas	=	Número de registros

# Relación. Representación

Una **relación** o una **entidad** del **modelo E-R** se representa en forma de **tabla (relación)** del modelo relacional, donde las cabeceras de las columnas representan los diferentes atributos de la relación y cada una de las filas -que reciben el nombre de tuplas- representa la información de una ocurrencia de la relación o entidad.



# Relación. Esquema y campos

El **esquema** de una relación suele representarse con el nombre de la relación junto a sus campos entre paréntesis y separados por comas.

Representaciones de relaciones
Alumnos (DNI, Nombre_Completo, Fecha_Nacimiento, Ciclo, Curso)

Los **campos o atributos** pueden ser de distintos tipos, como se vio en el tema anterior, y cada uno de ellos cuenta con su propia forma de representarlos.

Representaciones de atributos	
Multivaluados	Se incluyen destacados en la relación, cada uno de ellos, entre llaves y con una n que puede estar al lado o ponerse como exponente <b>{campo_multi}<sup>n</sup></b>
Compuestos	Se representan descomponiéndolos en sus atributos componentes. Por ejemplo: <b>Nombre_Completo.Nombre, Nombre_Completo.Apellidos</b> o simplemente <b>Nombre, Apellidos.</b>
Derivados	Se incluyen en la relación como un atributo más, debiendo especificarse su fórmula de derivación como una restricción adicional.

### **3. RESTRICCIONES**

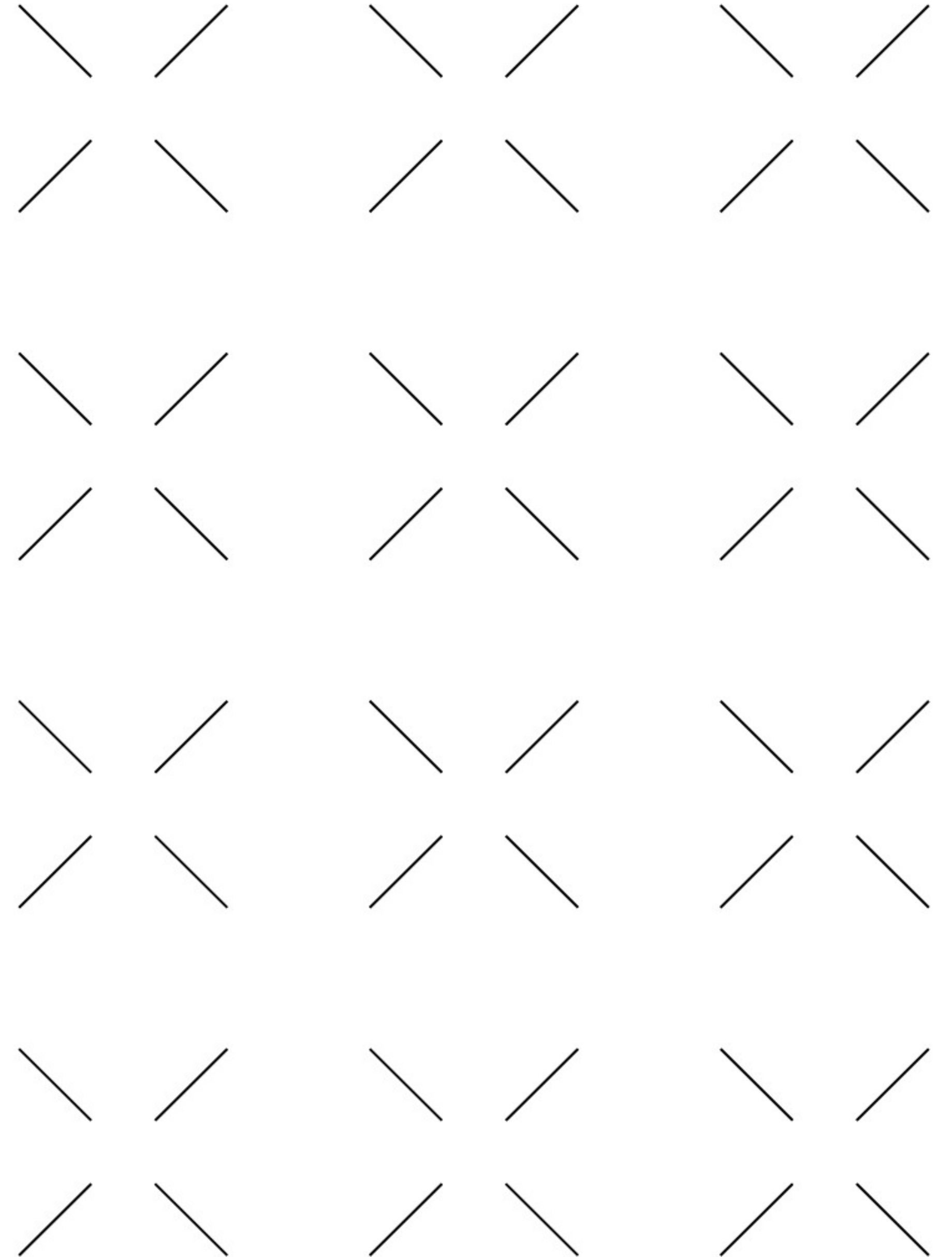
# Restricciones generales



Algunas de las restricciones generales del modelo lógico son:

- No podemos tener dos tuplas (filas) iguales. Las tuplas deben tener un valor diferente en al menos uno de sus atributos (campos).
- El orden de las tuplas (filas) es indiferente. El orden en el que se almacenen las filas de una tabla no es significativo.
- El orden de los atributos (campos) de la relación también es indiferente, no es significativo.
- En cada tupla (fila) cada atributo (campos) solamente puede tener un valor.
- Un atributo que forme parte de la clave primaria o principal no puede tomar valores nulos en una tupla (un valor nulo es ausencia de valor, es decir que no tiene un valor asignado).

## 3.1 Restricciones de clave





# Restricciones de claves

Una **clave** es un atributo (campo) o conjunto de atributos (campos) que **permite identificar de forma unívoca cada una de las tuplas de una relación** (antes hablábamos de las ocurrencias de una entidad).

## CLAVE PRIMARIA

Se representará subrayada o con CP o PK (PRIMARY KEY), para distinguirla del resto. Si es compuesta se subrayarán todos los atributos que la forman. Puede haber varias opciones a ser CP, las no escogidas son denominadas **claves alternativas**.

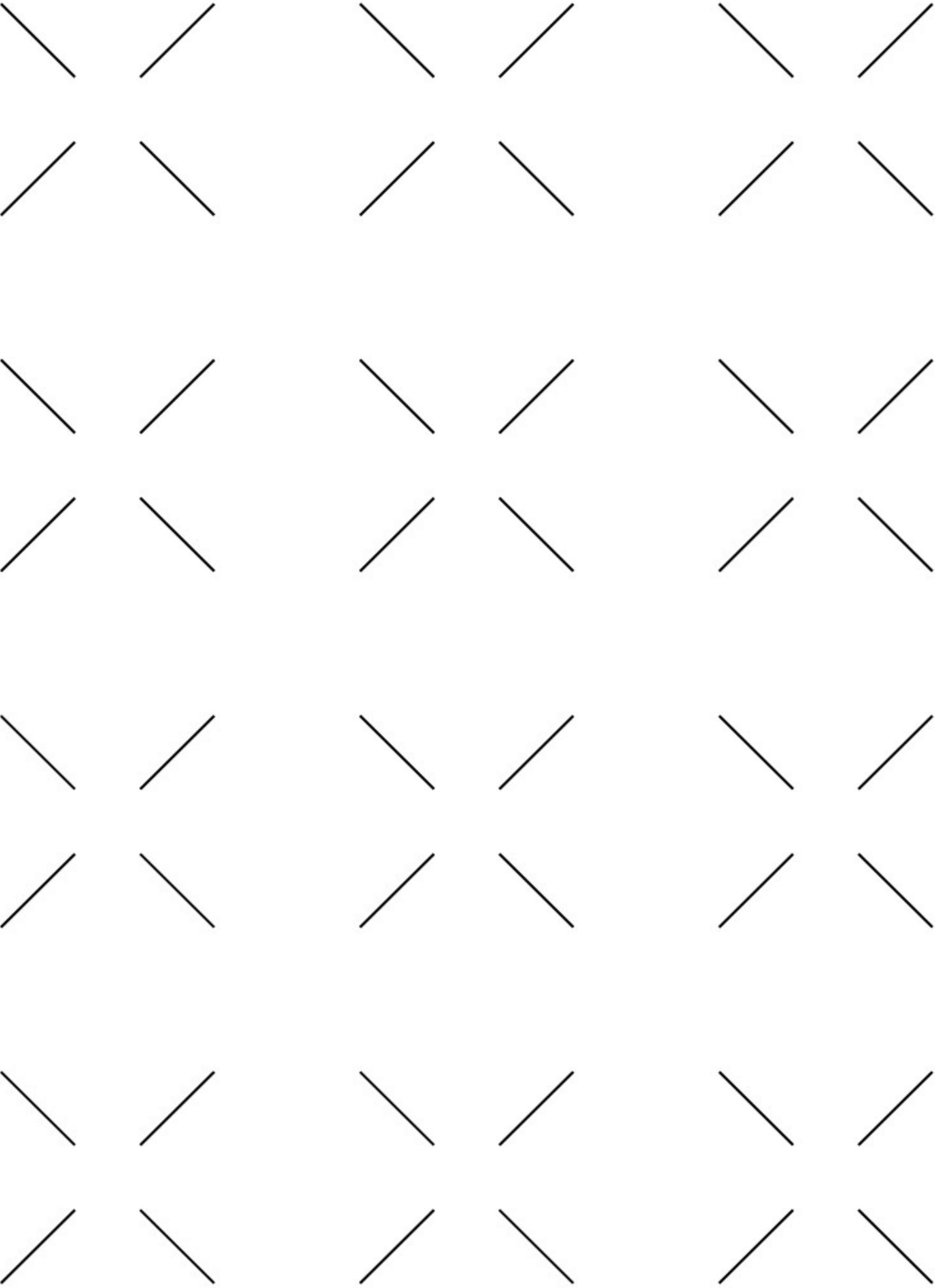
## CLAVE AJENA

Se representará con subrayado discontinuo o indicándolo con CAj o FK (FOREIGN KEY). En cualquier caso, hay que indicar SIEMPRE la tabla a la que se refiere (es opcional indicar el campo si se llama igual). No identifica esa entidad, identificaba la tabla a la que hace referencia.

Representaciones de la clave primaria	
Alumnos (DNI, Nombre, Fnacimiento) CP: DNI	Alumnos ( <u>DNI</u> , Nombre, FNacimiento)

Representaciones de la clave ajena	
Facturas (NFactura, Fecha, CIF) CP: NFactura CAj: CIF → Clientes {CIF}	Facturas ( <u>NFactura</u> , Fecha, <u>CIF</u> → Clientes)
Clientes (CIF, Nombre, FNacimiento) CP: CIF	Clientes ( <u>CIF</u> , Nombre, FNacimiento)

# 3.2 Restricciones de valor



# Restricciones de valor

## VALOR NO NULO

La restricción de valor NO NULO (NOT NULL) implica la obligatoriedad de que un atributo tenga un valor en cada tupla o fila. Se representará indicando un asterisco o mediante la cláusula VNN.

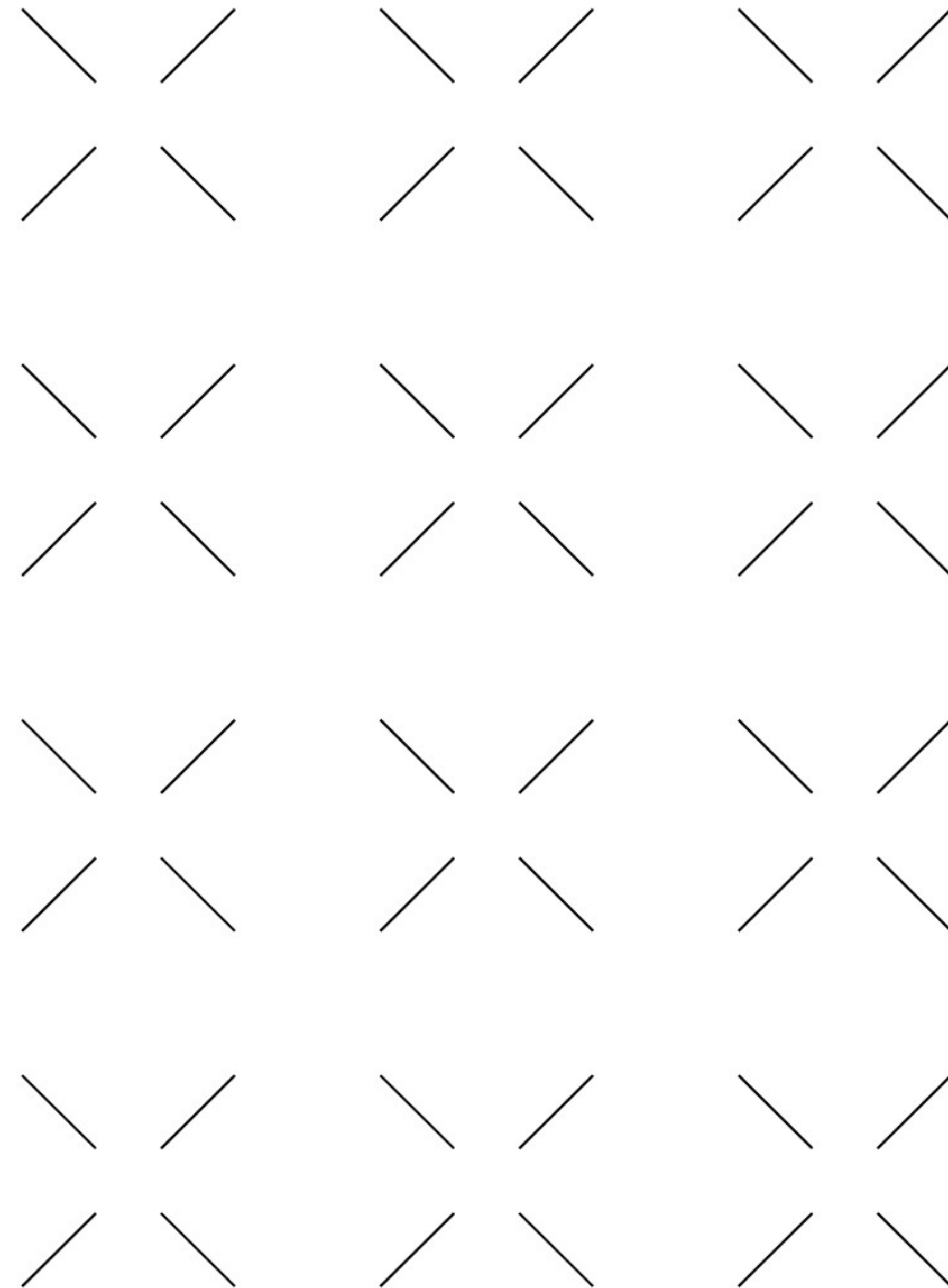
## UNICIDAD

Las restricciones de valor único (UNIQUE) nos permiten asegurar que en un atributo de la base de datos no se puedan introducir valores repetidos. Se representará indicando la cláusula ÚNICO o UK (UNIQUE KEY).

Representaciones de valor no nulo	
Alumnos (DNI, Nombre, Fnac) CP: DNI VNN: Nombre	Alumnos ( <u>DNI</u> , Nombre*, FNac)

Representaciones de valor único
Empleados ( <u>DNI</u> , Nombre, NumSegSocial) ÚNICO: NumSegSocial

### 3.3 Notaciones equivalentes



# Notaciones equivalentes

Según hemos visto, hay varias notaciones equivalentes que pueden usarse en el modelo relacional. Como ya deberías saber, escojas la notación que escojas, debes ser COHERENTE y aplicar siempre la misma en todo el modelado. Por ejemplo:

## Notación 1: Desglosada

**Cuenta\_Bancaria** (Número, Saldo, Fecha\_Creación, Número\_Contrato , Crédito, DNI)

CP: Número

CAj: DNI → Cliente {DNI}

VNN: Crédito

## Notación 2: Compacta

**Cuenta\_Bancaria** (Número, Saldo, Fecha\_Creación, Número\_Contrato , Crédito\*, DNI → Cliente {CIF})

## Notación 3: Combinada (combinación de las anteriores)

**Cuenta\_Bancaria** (Número, Saldo, Fecha\_Creación, Número\_Contrato , Crédito, DNI)

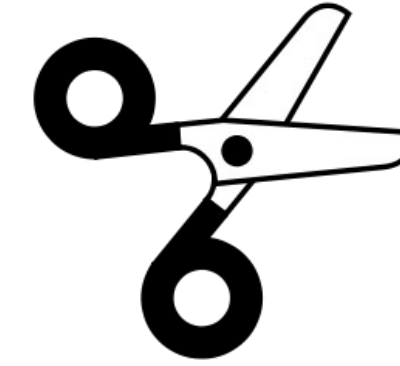
CP: Número

CAj: DNI → Cliente {DNI}

VNN: Crédito

## **4. PÉRDIDAS SEMÁNTICAS**

# Concepto de **pérdida semántica**



El modelo (conceptual) E-R de Peter Chen es mucho más potente que el modelo (lógico) relacional de Edgar Codd. Esto provoca que haya realidades que podemos representar a nivel conceptual (E-R) que no podemos representar a nivel lógico (relacional), como son las mínimas de cualquier participación (x,n) y cualquier participación mínima en las n-arias (ternarias, cuaternarias, etc.).

Esta carencia la resolveremos con una **restricción de integridad** que consiste en describir, de momento en **lenguaje natural**, qué aspectos del conceptual (E-R) estamos pasando por alto en el lógico (relacional) para que los tengamos en cuenta en el modelado físico (el que depende del SGBD elegido).

La **pérdida semántica** puede explicarse de muchas maneras:

- No podemos expresar solamente con tablas lo que nos indica gráficamente el diagrama ER
- No podemos expresar en la fase 2 lo que nos indican en la fase 1
- El modelo lógico no puede expresar todo lo que expresa el modelo conceptual
- El diagrama de Chen es más rico que el modelo de Codd



# Casos

Resumiendo, podemos decir que las **PÉRDIDAS SEMÁNTICAS** al pasar del modelo conceptual ER al modelo lógico relacional, se pueden dar por tres casos. Indicamos qué hacer en cada uno de ellos:

## (1) Relaciones ternarias con alguna participación (1,n) o (1,1)

- Hacemos la transformación ignorando esas mínimas.
- Indicamos en lenguaje natural qué deberá comprobarse en el modelado físico.

## (2) Relaciones binarias con alguna participación (1,n)

- Hacemos la transformación ignorando esas mínimas.
- Indicamos en lenguaje natural qué deberá comprobarse en el modelado físico.

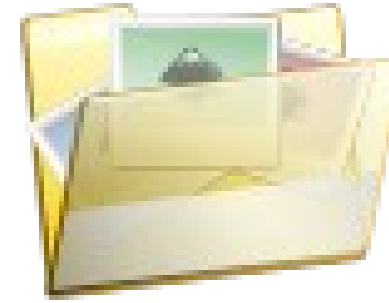
## (3) Especializaciones distintas a PS (parcial y solapada)

- Hacemos la transformación como si se tratara de una PS
- Indicamos en lenguaje natural qué deberá comprobarse en el modelado físico.



## **5. PASO DEL MODELO CONCEPTUAL AL MODELO LÓGICO**

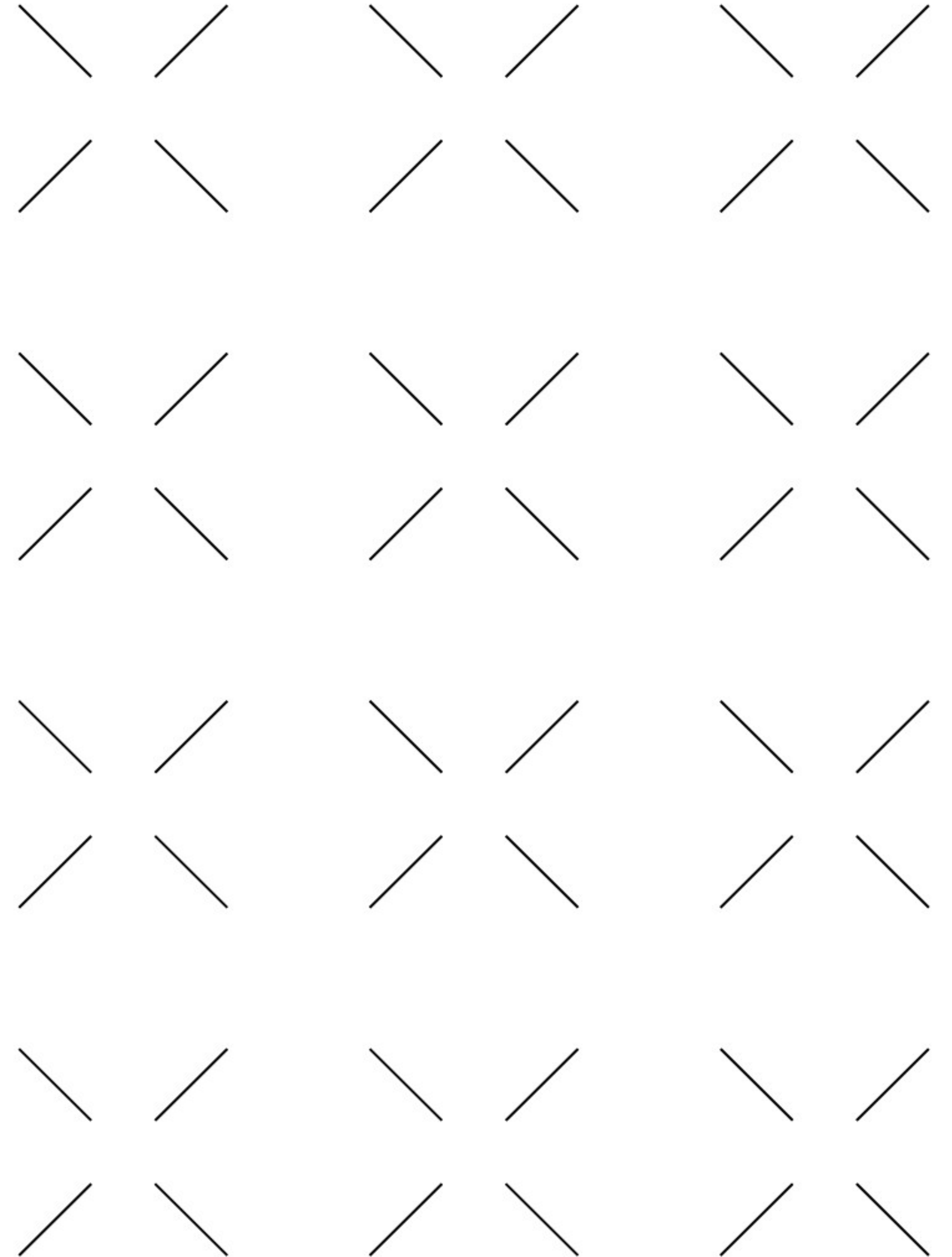
# Relaciones (tablas)



Las relaciones y las entidades del modelo conceptual (E-R) **se pasan a tablas** en el modelo lógico (relacional) aunque la correlación no es 1 a 1.

Por ejemplo, tres elementos del E-R (dos entidades y una relación) pueden convertirse en dos tablas del relacional, en una sola o en tres, como veremos a continuación.

## 5.1 Transformación de entidades



# Entidades



Las entidades se transformarán en una relación (tabla). Cada atributo de la entidad se transformará en un atributo o campo de la relación.

Cada dominio de los atributos se transformará en el dominio del atributo o campo correspondiente de la Relación.

La clave primaria o principal de la relación estará formada por la clave principal o identificador de la entidad que estamos transformando (simple o compuesto).

# Ejemplo

- **Clave primaria:**

- DNI

- **Compuestos:**

- FechaNacimiento.Dia, FechaNacimiento.Mes, FechaNacimiento.Año
- PrimerNombre, Apellido

- **Multivaluados:**

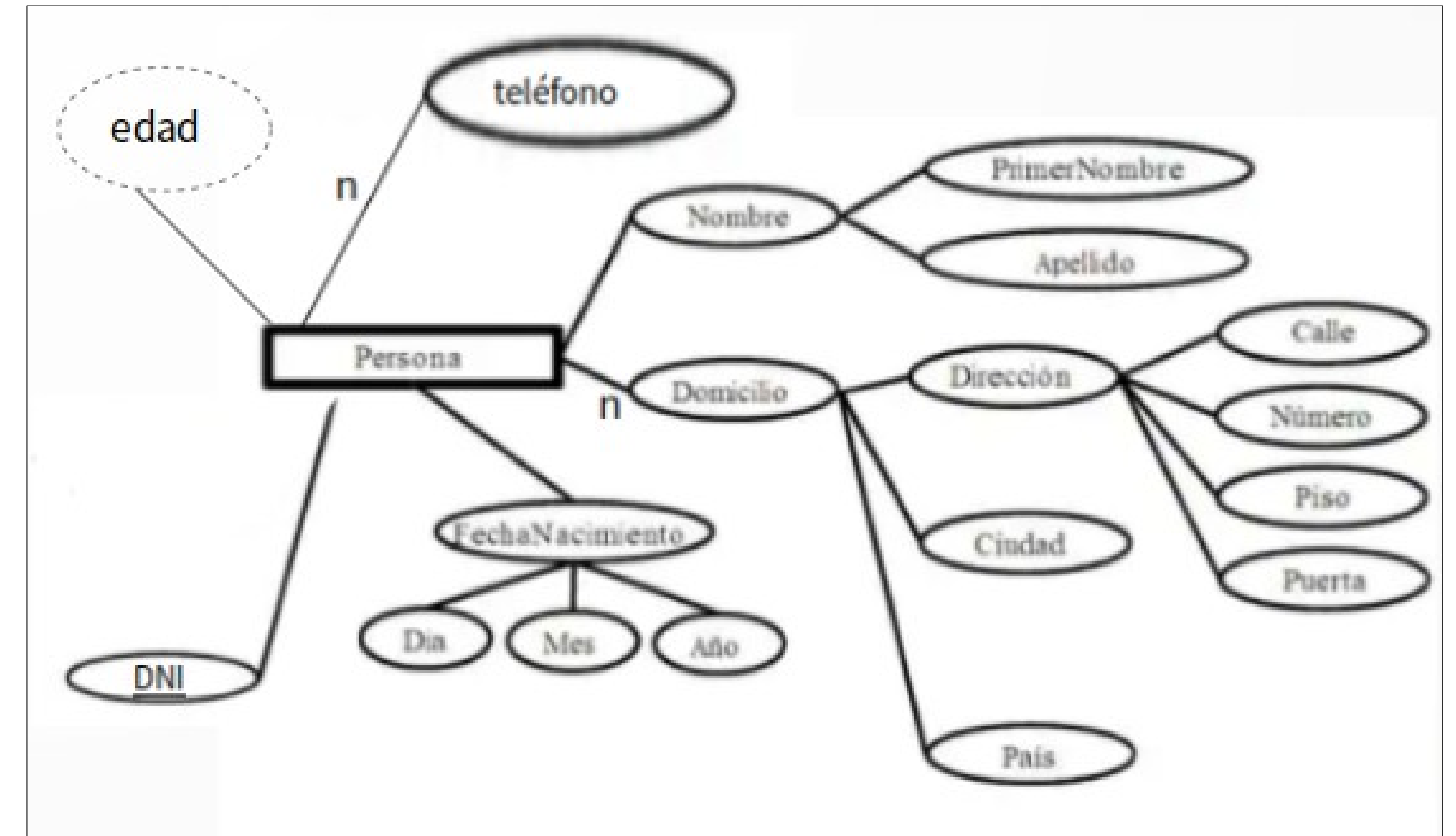
- {teléfono}<sup>n</sup>

- **Derivados:**

- edad = FechaActual.Año – FechaNacimiento.Año

- **Compuestos multivaluados:**

- {Domicilio.Dirección, Domicilio.Dirección.Calle, Domicilio.Dirección.Número, Domicilio.Dirección.Piso, Domicilio.Dirección.Puerta, Domicilio.Ciudad, Domicilio.País}<sup>n</sup>



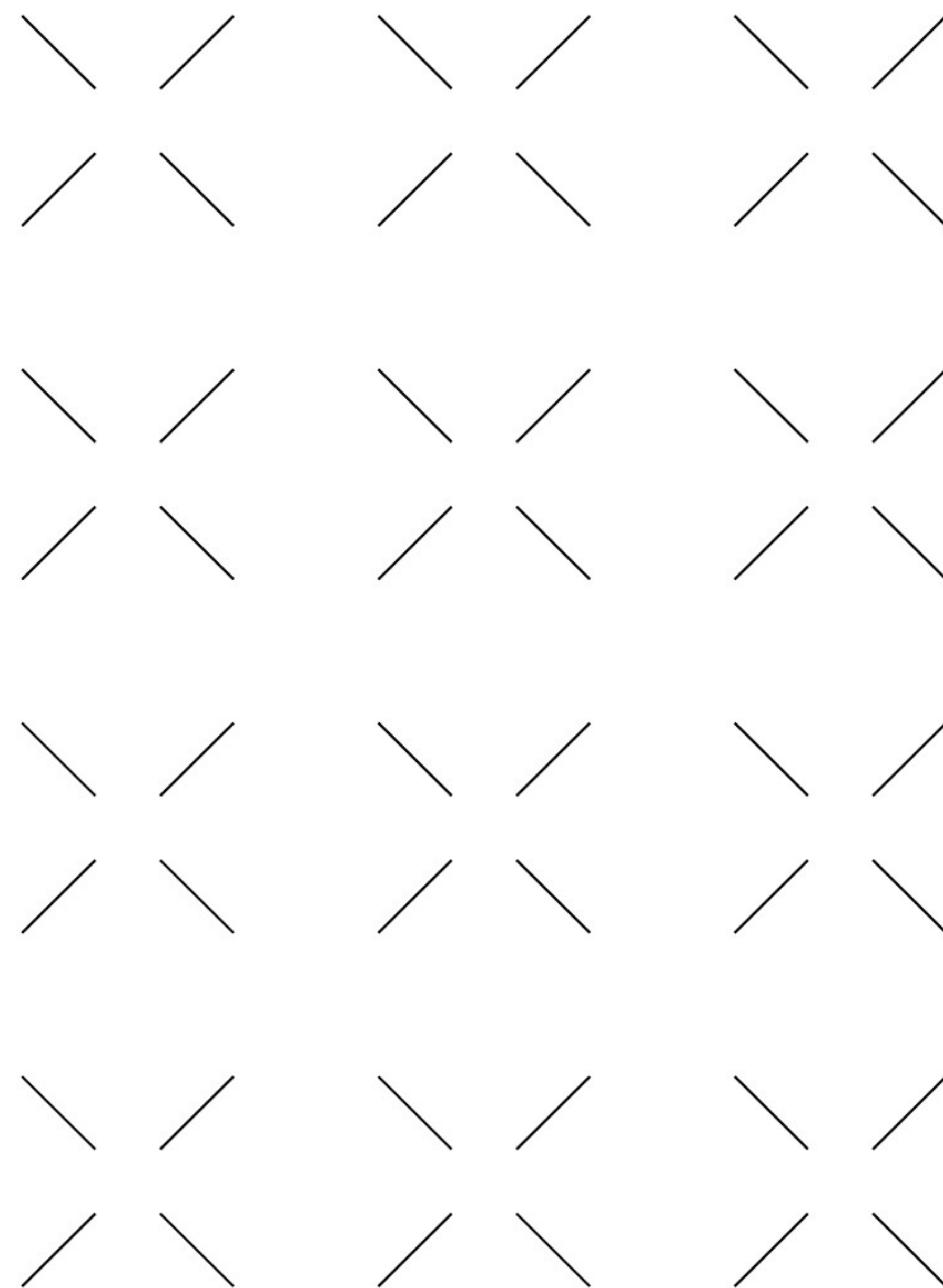
Quedando la transformación de la siguiente manera:

**Persona** (DNI, FechaNacimiento.Dia, FechaNacimiento.Mes, FechaNacimiento.Año, {Domicilio.Dirección, Domicilio.Dirección.Calle, Domicilio.Dirección.Número, Domicilio.Dirección.Piso, Domicilio.Dirección.Puerta, Domicilio.Ciudad, Domicilio.País}<sup>n</sup>, {teléfono}<sup>n</sup>, PrimerNombre, Apellido)

CP: DNI

**edad** = FechaActual.Año – FechaNacimiento.Año

# 5.2 Transformación de relaciones



# Relaciones binarias N:M

Por un lado tendremos la tabla Alumno, por otro la tabla Cursos y además aparecerá una nueva tabla fruto de la relación N:M “Está matriculado” que podemos llamar Matrículas. La transformación quedaría así (**los atributos NO CLAVES son inventados, pues no aparecen en el ejemplo**):

**Alumnos** (DNI, Nombre, Fecha\_Nacimiento)

CP: DNI

**Cursos** (Cod\_Curso, Nombre, Fecha\_Comienzo)

CP: Cod\_Curso

**Matrículas** (DNI, Cod\_Curso)

CP: {DNI, Cod\_Curso}

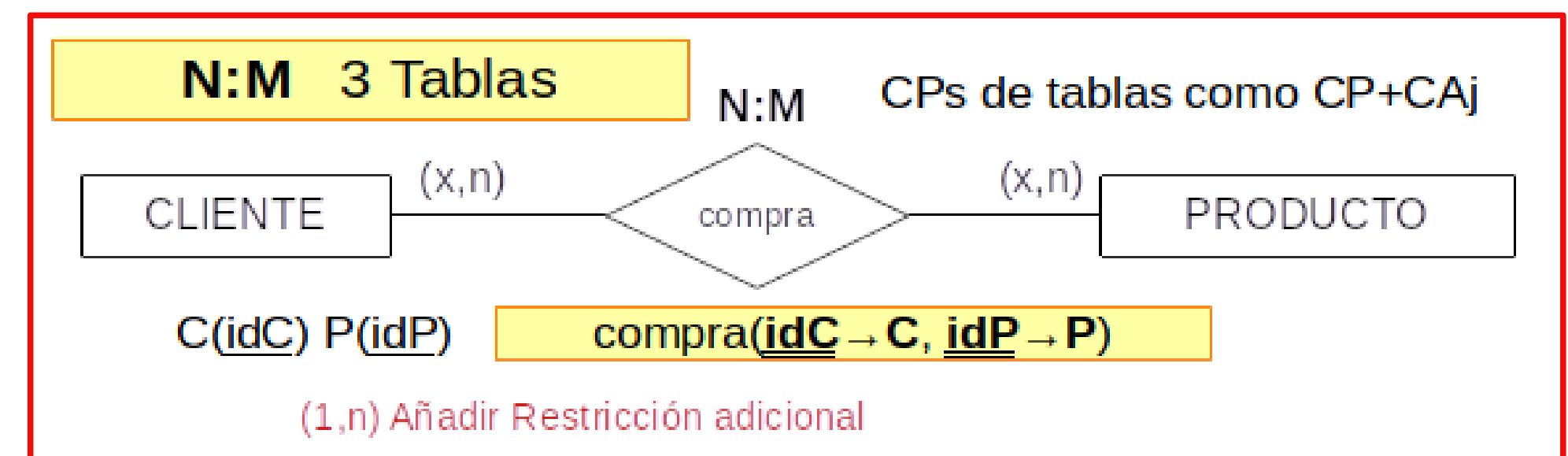
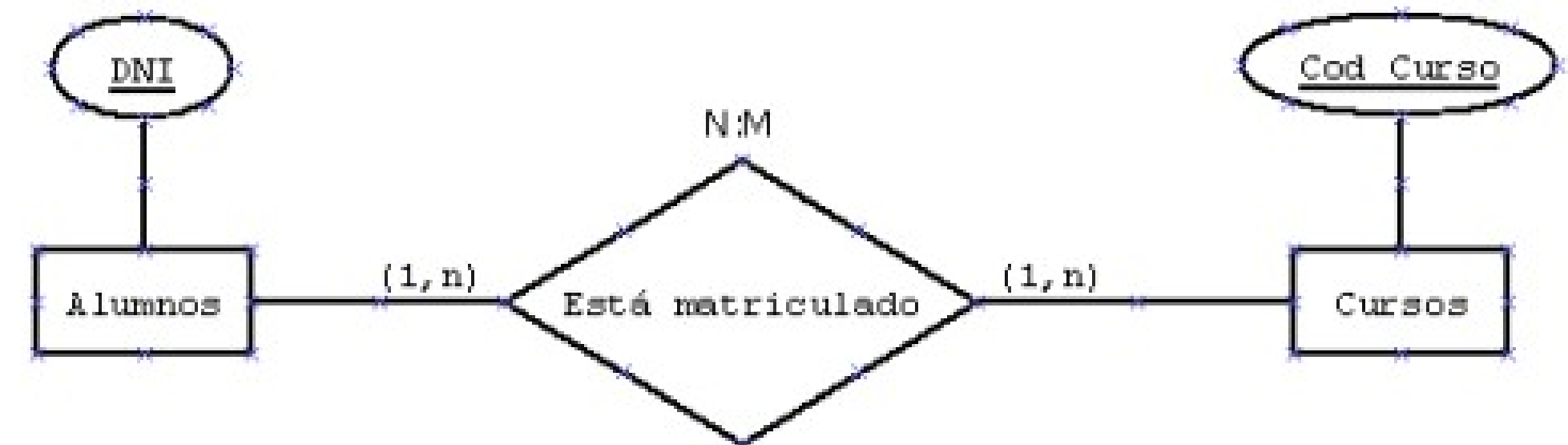
CAj: DNI → Alumnos {DNI}

CAj: Cod\_Curso → Cursos {Cod\_Curso}

**NOTA:** Cuando un atributo es clave primaria y ajena a la vez, también puede representarse con subrayado doble continuo.

## RESTRICCIÓN DE INTEGRIDAD (por pérdida semántica):

- Sería necesario comprobar que todo alumno tiene un curso asociado y viceversa.



# Relaciones binarias 1:N

La transformación de este fragmento (los atributos **NO CLAVES** son inventados, pues no aparecen en el ejemplo) dará lugar a dos tablas, la tabla Profesor y la tabla Alumnos, quedando de la siguiente forma:

**Profesor** (DNI\_Profesor, Nombre, Fecha\_Incorporación)

CP: DNI\_Profesor

**Alumno** (DNI\_Alumno, Correo, Teléfono, DNI\_Profesor\_Tutor)

CP: DNI\_Alumno

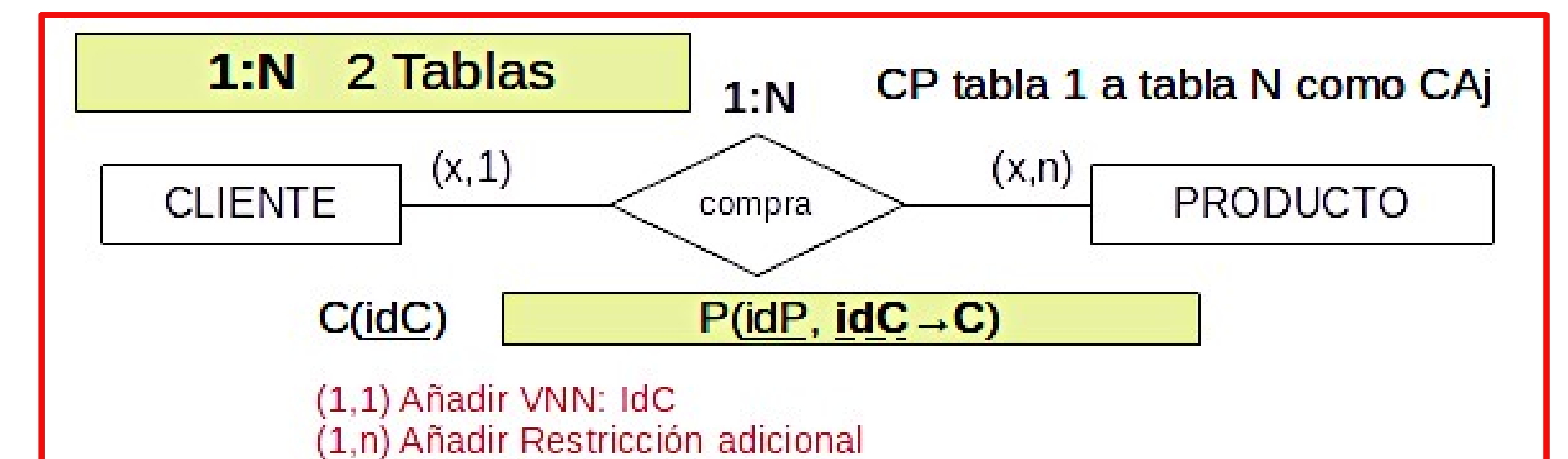
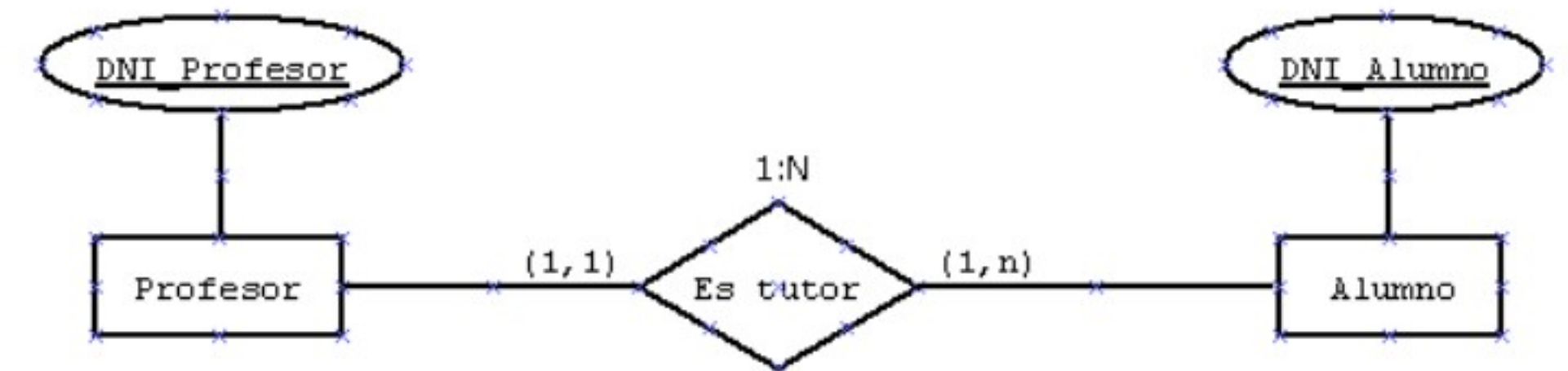
CAj: DNI\_Profesor\_Tutor → Profesor {DNI\_Profesor}

VNN: DNI\_Profesor\_Tutor

No hay que comprobar si un alumno tiene un profesor asociado, puesto que la **restricción de valor no nulo** ya garantiza que un alumno tiene un profesor tutor.

## RESTRICCIÓN DE INTEGRIDAD (por pérdida semántica):

- Sería necesario comprobar que todo profesor tiene un alumno asociado.

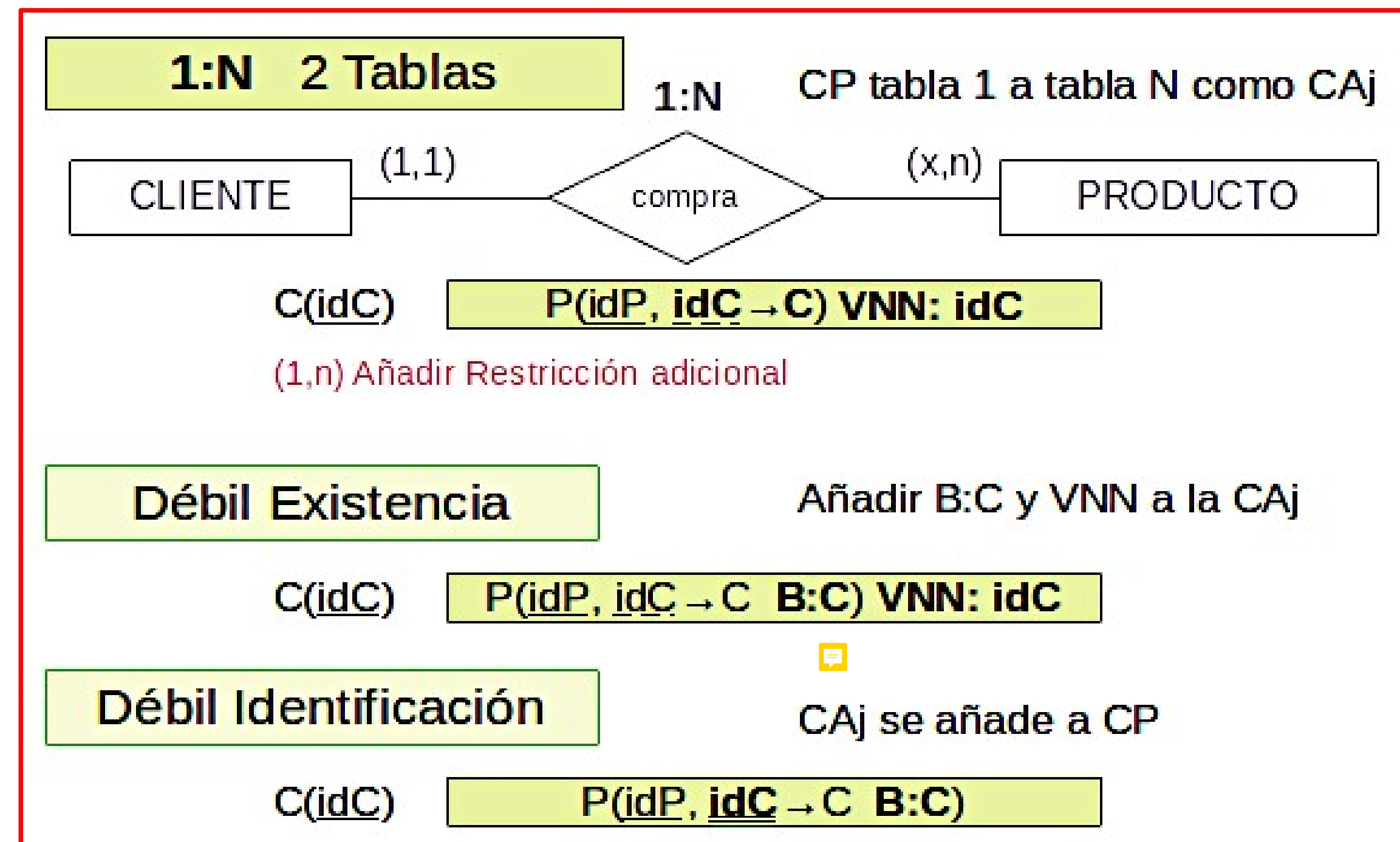




# Entidades débiles en 1:N

Recuerda que una entidad débil es aquella que necesita de una fuerte para poder existir.  
Se transforman como las 1:N con mínima (1,1), añadiendo a la CAj un borrado en cascada.

- **Existencia:** No olvidar poner el VNN en la CAj (por ser la participación mínima 1).
- **Identificación:** La CAj además se suma a la CP (ya no necesita VNN al ser parte de la CP).



# Relaciones binarias 1:1

Para asegurar que un coche oficial tenga un y solo un alto cargo asignado necesitamos una restricción de unicidad. Los atributos no claves son inventados pues no aparecen en el modelo E-R del que partimos.

**Alto\_Cargo** (DNI, Nombre, Puesto, Fecha\_Posesión)

CP: DNI

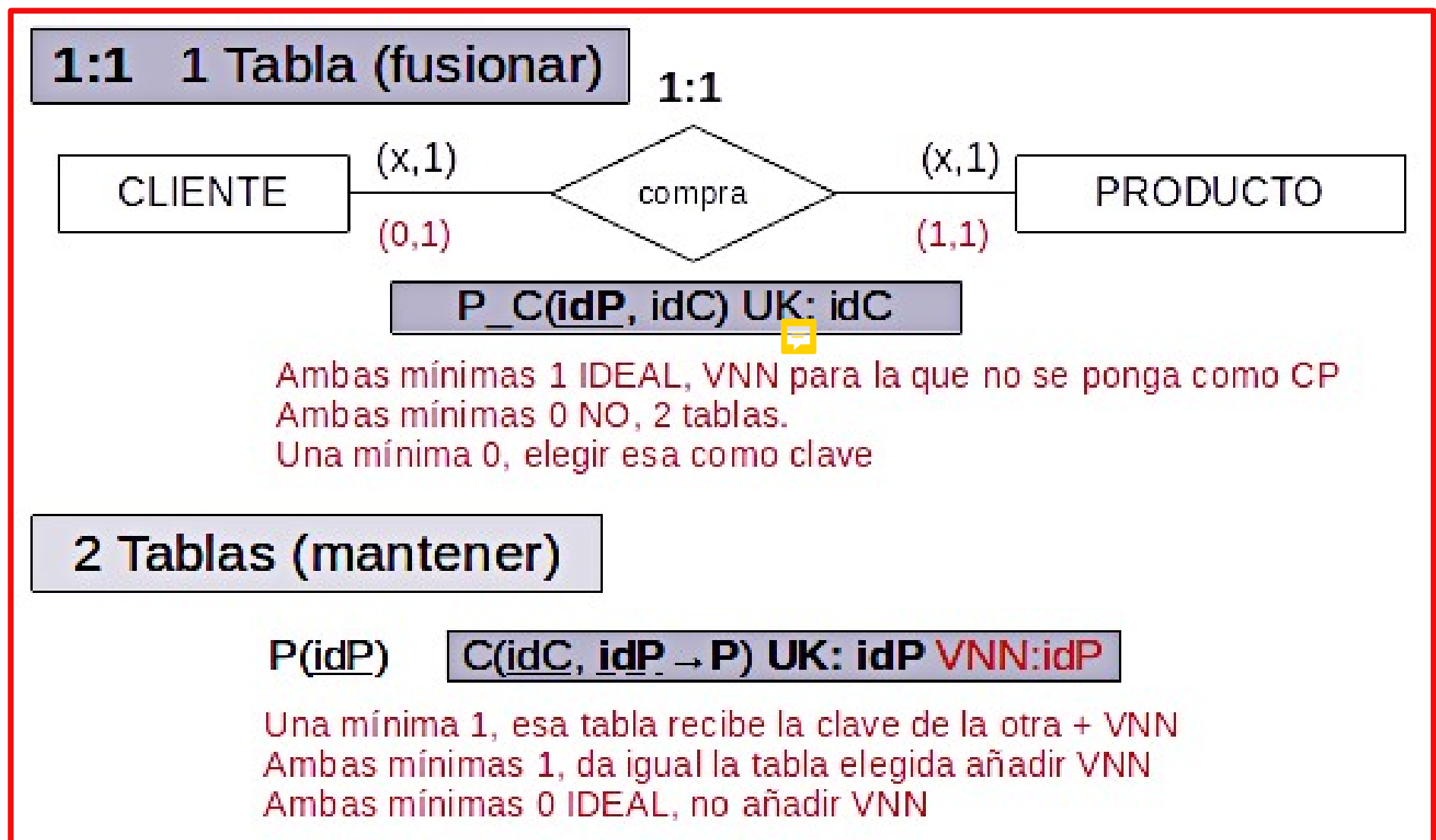
**Coche\_Oficial** (Matrícula, Marca, Modelo, Color, DNI)

CP: Matrícula

CAj: DNI → Alto\_Cargo {DNI}

ÚNICO: DNI

VNN: DNI



# Relaciones unarias (reflexivas) 1:N

La transformación será la misma que en las relaciones binarias, pero ahora el lugar donde se debe añadir el nuevo atributo será en la misma tabla generada a partir de la entidad (lógico, pues solo disponemos de una entidad). Por ello cambiaremos el nombre del atributo para que sepamos a qué corresponde. Por ejemplo, tenemos un club de socios.

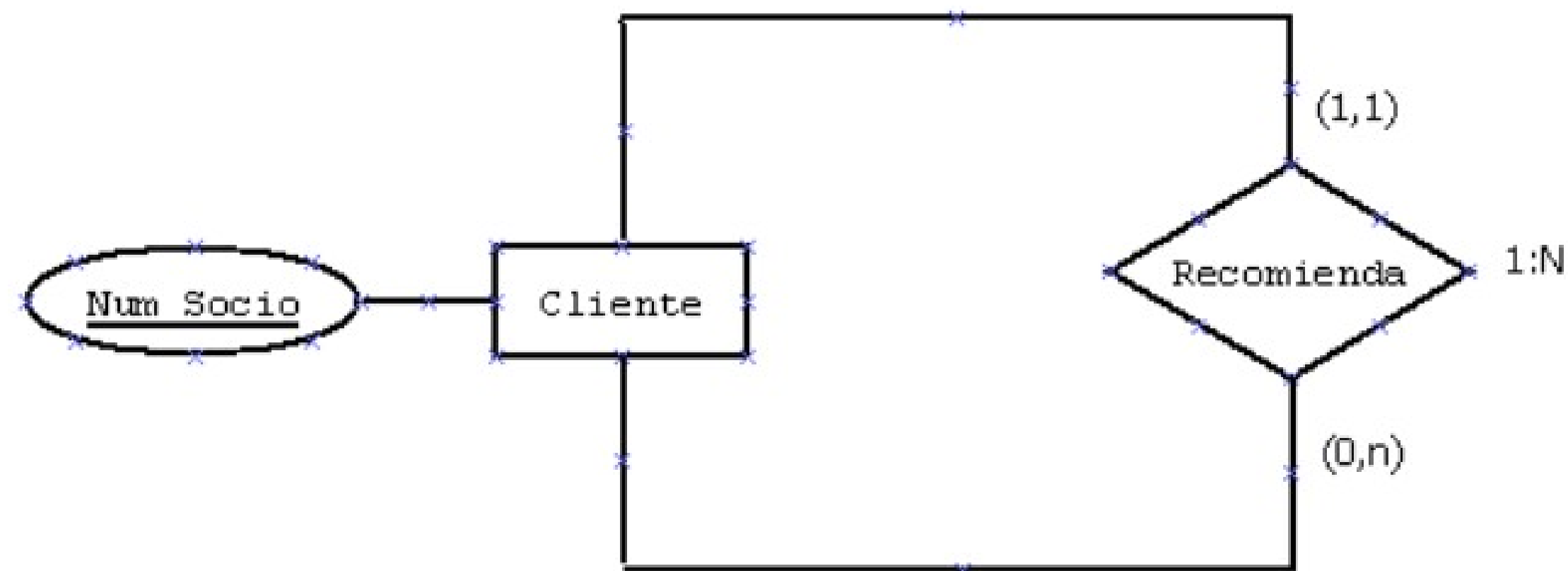
**Cliente** (Num\_Socio, Nombre, Recomendado\_Por\_Num\_Socio)

CP: Num\_Socio

CAj: Recomendado\_por\_Num\_Socio → Cliente (Num\_Socio)

VNN: Recomendado\_por\_Num\_Socio

Podemos ver que el nuevo campo al que hemos llamado “Recomendado\_Por\_Num\_Socio” representará al socio que realizó la recomendación de ese cliente. Si la participación fuera (0,1) en lugar de (1,1), no pondríamos la restricción de valor no nulo.



# Relaciones unarias (reflexivas) N:M

El proceso es similar al de las relaciones binarias N:M. En este caso, debido a que la relación reflexiva tiene cardinalidad máxima N:M, es necesario utilizar una relación independiente para representarla. Nótese como ahora hay (en la nueva relación) dos claves ajenas que se refieren a la misma relación. Si hubiera definida alguna restricción de existencia, también sería necesaria la inclusión de una restricción de integridad (en lenguaje natural). La presencia de restricciones de existencia en las relaciones reflexivas es, sin embargo, poco frecuente.

**Cliente** (Num\_Socio, Nombre)

CP: Num\_Socio

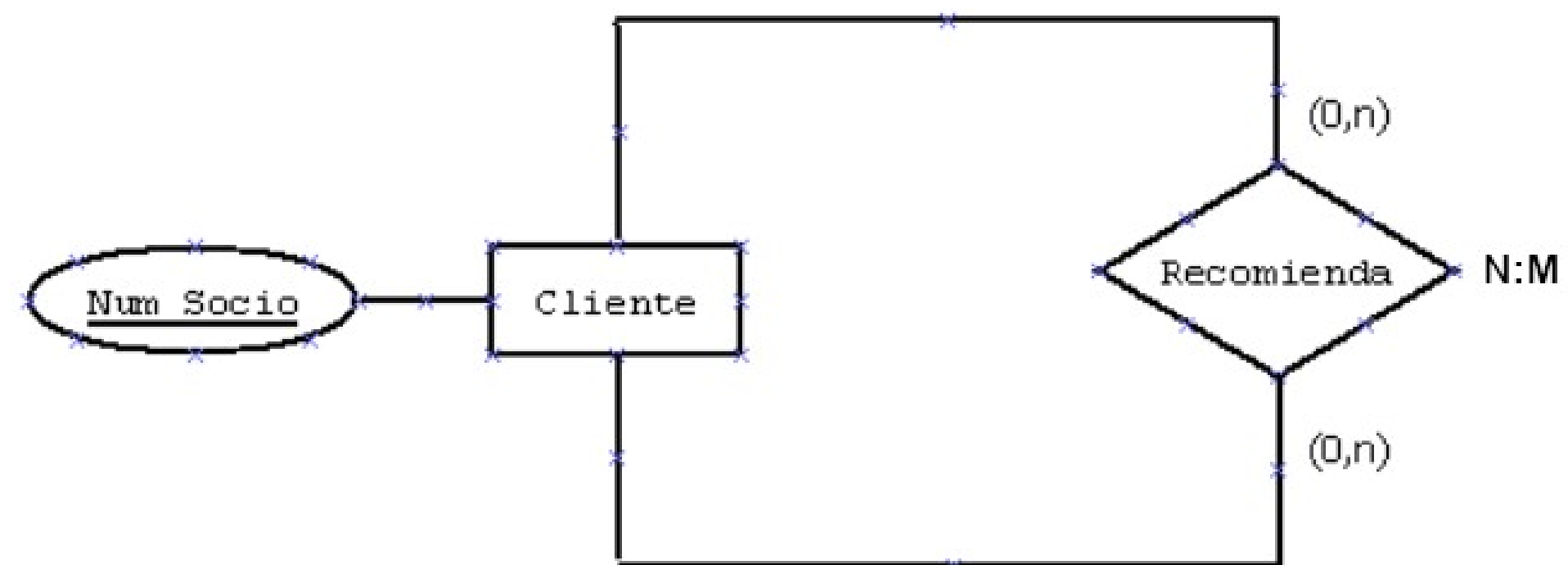
**Recomendado** (Num\_Socio, Recomendado\_Por\_Num\_Socio)

CP: {Num\_Socio, Recomendado\_Por\_Num\_Socio}

CAj: Num\_Socio → Cliente (Num\_Socio)

CAj: Recomendado\_por\_Num\_Socio → Cliente (Num\_Socio)

Nuevamente, el nuevo campo al que hemos llamado “Recomendado\_Por\_Num\_Socio” representará al socio que realizó la recomendación de ese cliente

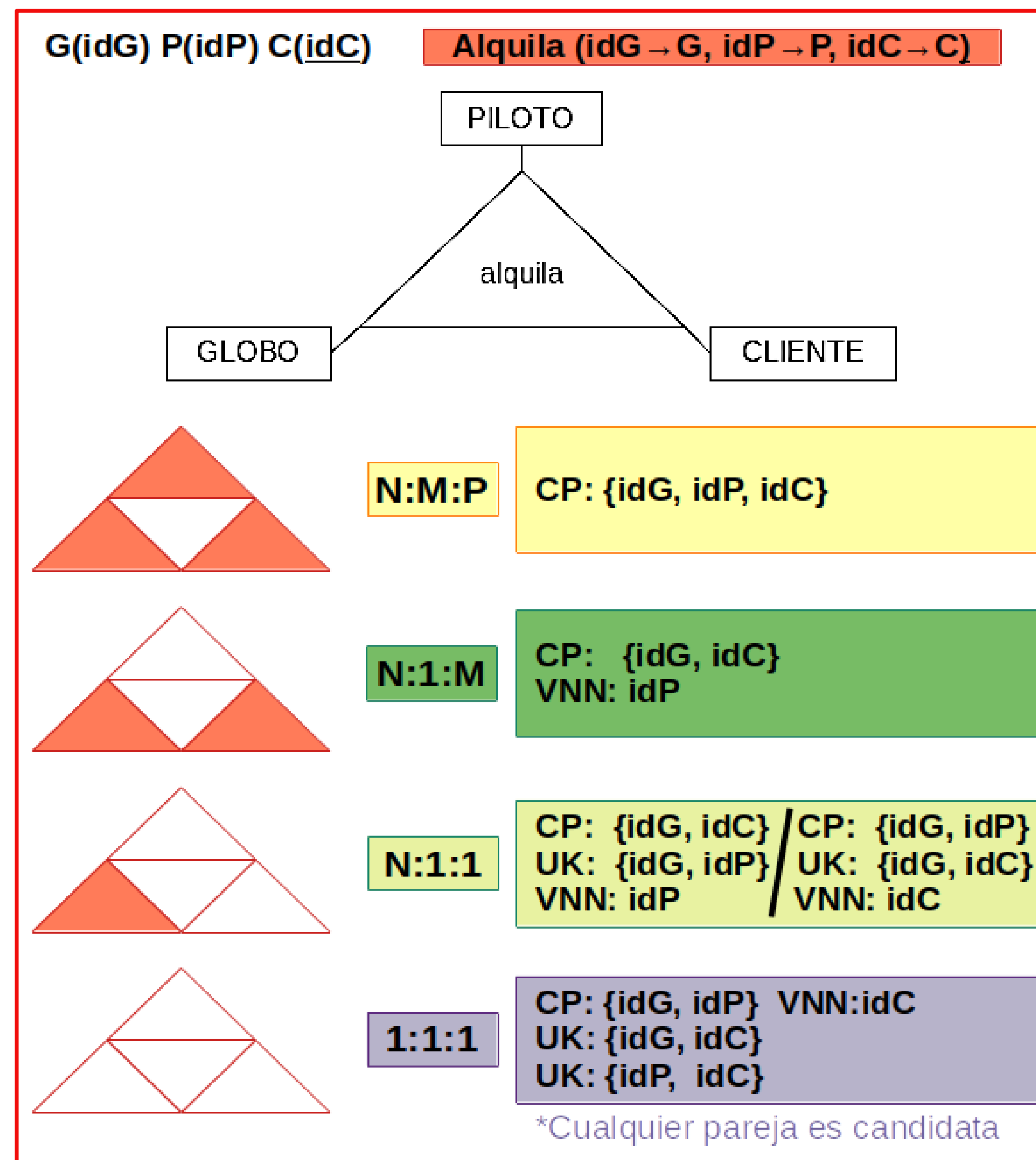


# Relaciones ternarias

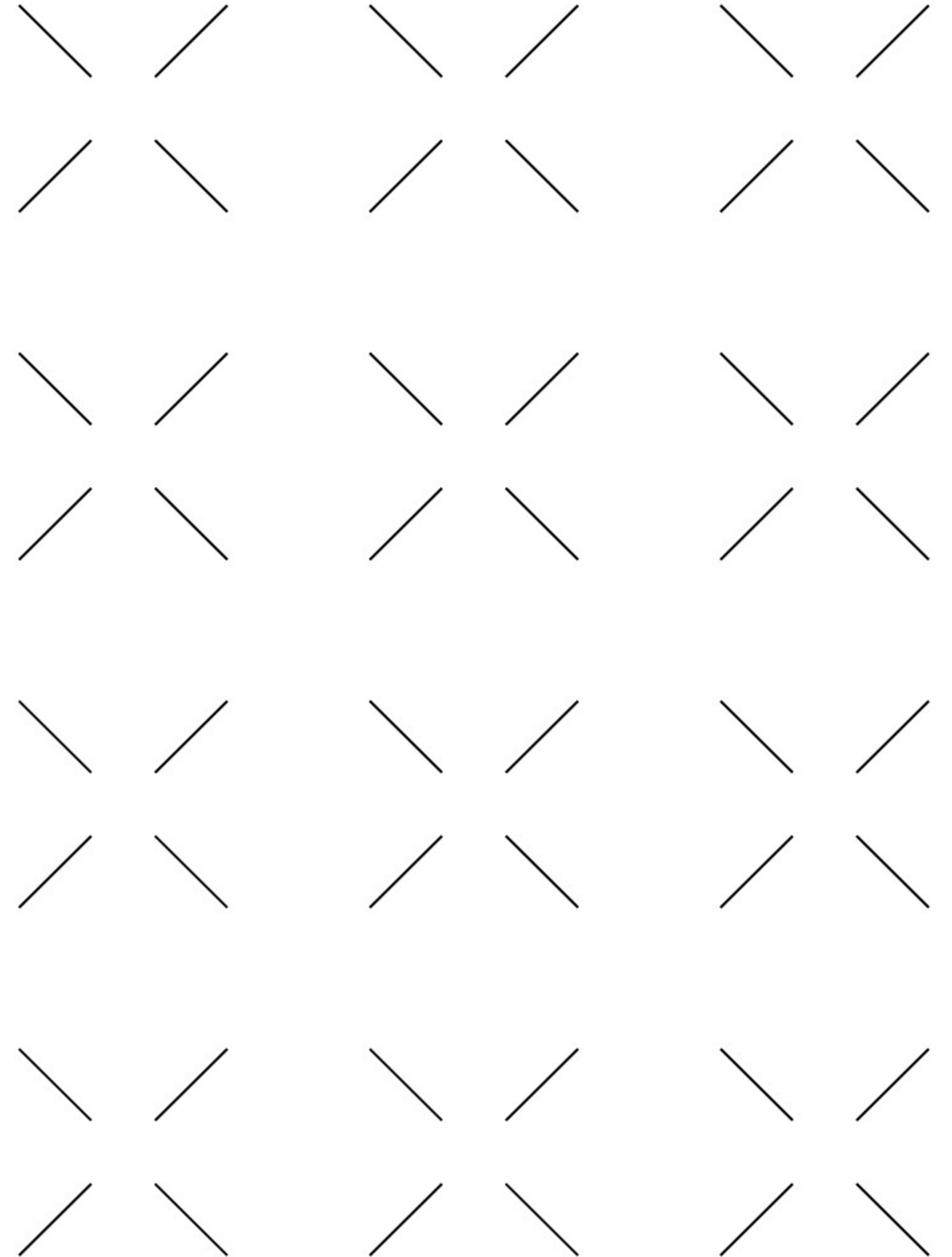
Se crea una nueva tabla con las claves principales de cada una de las respectivas entidades y los atributos de la relación E-R (si hay). Además cada una de ellas será respectivamente CAj referenciando a la tabla original.

Dependiendo de la cardinalidad se aplicará una CP u otra a la nueva tabla, además de ciertas restricciones.

Las participaciones solo reflejan como funciona una pareja de entidades respecto a la tercera. Tanto para expresar las mínimas de éstas como para reflejar cualquier limitación entre solo 2 de las 3 entidades se recurre a restricciones adicionales.



## 5.3 Transformación de especializaciones

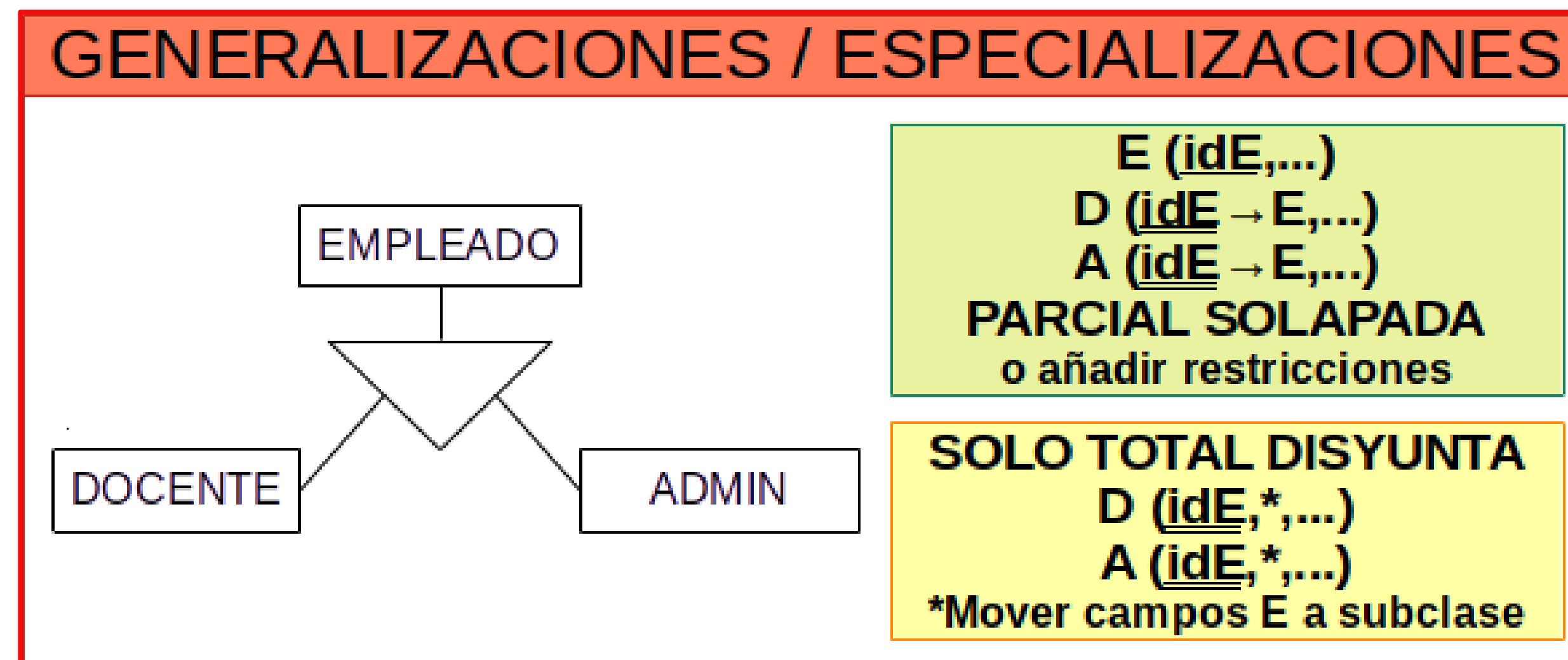




# Relaciones de especialización

Existen distintas formas de representarlas, aunque algunas significan más bien ignorar la especialización (subtipos) o la generalización (supertipo). Lo más común es escoger entre estas 2 opciones:

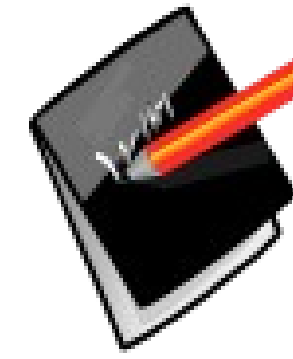
- 1) Crear una tabla para cada entidad. La CP del supertipo se usa también en los subtipos como CP+CAj. Es la opción más común; ideal para relaciones **parciales y solapadas**, en **totales y/o disyuntas** hay que añadir **restricciones adicionales** para reflejar esas condiciones.
- 2) Crear tablas solo para los subtipos incorporando en cada una todos los atributos del supertipo. **Solo válida para totales** y no es recomendable en **solapadas** (redundante). **Ideal para totales disyuntas que no tengan relaciones en el supertipo**. Si hay relaciones allí obliga a "separarlas" en una para cada subtipo.



## 6. BIBLIOGRAFÍA



# Recursos



- Bases de datos. Tuinstitutoonline. <https://www.tuinstitutoonline.com/>
- Iván López, M.<sup>a</sup> Jesús Castellano. John Ospino. Bases de Datos. Ed. Garceta, 2a edición, 2017. ISBN: 978-8415452959
- Matilde Celma, Juan Carlos Casamayor y Laura Mota. Bases de datos relacionales. Ed. Prentice-Hall, 2003
- Cabrera Sánchez, Gregorio. Análisis y diseño detallado de aplicaciones informáticas de gestión. Ed. McGraw-Hill, 1st edition, 1999. ISBN: 8448122313
- Carlos Manuel Martí Hernández. Bases de dades. Desenvolupament d'aplicacions multiplataforma i Desenvolupament d'aplicacions web. Creative Commons. Departament d'Ensenyament, Institut Obert de Catalunya. Dipòsit legal: B. 12715-2016. <https://ioc.xtec.cat/educacio/recursos>

