

CFGSS

***Desarrollo de Aplicaciones
Multiplataforma
Administración de Sistemas
Informáticos en Red***

MÓDULOS

Bases de Datos

Gestión de Bases de Datos

UT3: DISEÑO LÓGICO. MODELO RELACIONAL

Índice

1. Historia del modelo relacional
2. Modelo lógico I
 - 2.1 Tablas, registros y atributos
 - 2.2 Claves y dominios del modelo lógico
 - 2.3 Representación de una tabla en el modelo relacional
3. Modelo lógico II
 - 3.1 Clave ajena e integridad referencial
 - 3.2 Orden de campos
 - 3.3 Transformaciones de relaciones uno a varios
 - 3.4 Clave primaria compuesta
 - 3.5 Transformación de atributos compuestos y derivados
 - 3.6 Transformaciones de relaciones varios a varios
 - 3.7 Cardinalidades en el modelo relacional

Índice

4. Modelo lógico III

4.1 Transformación de entidades débiles

4.2 Transformación de atributos multivaluados

4.3 Transformación de relaciones uno a uno

4.4 Transformación de relaciones reflexivas

~~4.5 Transformación de relaciones ternarias~~

4.6 Transformación de jerarquías

4.7 Normalización

4. Ampliación

5.1 Transformaciones de relaciones uno a varios CON NUEVA TABLA

5.2 Otros métodos de transformación de jerarquías

1. Historia del modelo relacional

- Edgar Frank Codd definió las bases del modelo relacional a finales de los 60.
- Lo que intentaba era evitar que los usuarios de las bases de datos, tuvieran que verse obligadas a aprender los entresijos internos del sistema.
- Pretendía que los usuarios trabajaran de forma sencilla e independiente del funcionamiento físico de la base de datos en sí.
- Fue un enfoque revolucionario.

2.1 Tablas, registros y campos

El elemento fundamental es lo que se conoce como **tabla** o **relación**.

Para evitar confusiones con las relaciones del modelo ER en adelante las llamaremos **tablas**.

Las **tablas** constan de:

- **Atributos.** Cada una de las **propiedades** de los datos que se almacenan en la tabla (nombre, dni,...). También se llaman columnas o **campos**.
 - ✓ Se corresponden con los **atributos de las entidades** del modelo ER.
- **Tuplas.** Cada **ocurrencia** de la tabla. También se llaman filas o **registros**.
 - ✓ Por ejemplo si se tiene una tabla que almacena personas, una tupla o registro representaría cada una de las personas almacenadas (Ana, Pedro, Carmen,...).

2.1 Tablas, registros y campos

Ejemplo: Tabla “Diplomáticos”

The diagram illustrates the components of a table. An orange box labeled "Campos o Atributos" (Fields or Attributes) is positioned above the table, with a bracket indicating it applies to the header row. Another orange box labeled "Registros o Tuplas" (Records or Tuples) is positioned to the left of the table, with a bracket indicating it applies to the data rows.

Nombre	Nacionalidad	Institución
Sofia	España	O.N.U.
John	EE. UU.	O.M.S.
Jacob	Francia	N.A.S.A.
Sebastian	Chila	UNASUR

2.1 Tablas, registros y campos

Las tablas deben cumplir una serie de **propiedades**:

- Cada tabla tiene **un nombre distinto**.
- Los nombres de **los atributos dentro una tabla son distintos**, aunque pueden repetirse en tablas diferentes.
- Cada atributo de la tabla **toma un solo valor** en cada registro.
- Cada registro es único. No hay registros duplicados. (Siempre se distinguirán al menos por la **clave primaria**).
- El orden de los atributos no importa.
- El orden de los registros no importa.

2.1 Tablas, registros y campos

Registro o tupla

- Es cada una de **las filas de la tabla**.
- Representa por tanto cada elemento individual de esa tabla.
- Tiene que cumplir que:
 - Cada registro se debe corresponder con un elemento del mundo real.
 - No puede haber dos registros iguales (con todos los valores de sus atributos iguales). Es decir, contará con algún campo de **clave primaria**.

2.2 Claves y dominios del modelo lógico

Clave candidata

- Conjunto de atributos que identifican **unívocamente** cada registro de una tabla.
- Son las columnas cuyos valores **no se repiten** en ningún otro registro de esa tabla.
- Toda tabla en el modelo relacional debe tener **al menos una clave candidata**.

2.2 Claves y dominios del modelo lógico

Clave primaria

- Clave candidata que se escoge como **identificador de los registros**. Se elige la que identifique mejor a cada registro en el contexto de la base de datos.
- Por ejemplo un campo *DNI* sería clave candidata en una tabla de clientes, pero si tiene un campo *código cliente*, éste sería mejor como clave primaria.

Clave alternativa

Cualquier clave candidata que no sea primaria.

2.2 Claves y dominios del modelo lógico

- Será necesario establecer el tipo de datos (dominio) que vamos usar para cada atributo.
- Nosotros trabajaremos con **tipos de datos básicos o dominios**:
 1. Texto.
 2. Número.
 3. Fecha
 4. Hora.
 5. Fecha/Hora
 6. Booleano: dos valores posibles (Si/No, ON/OFF, 0/1, etc.).

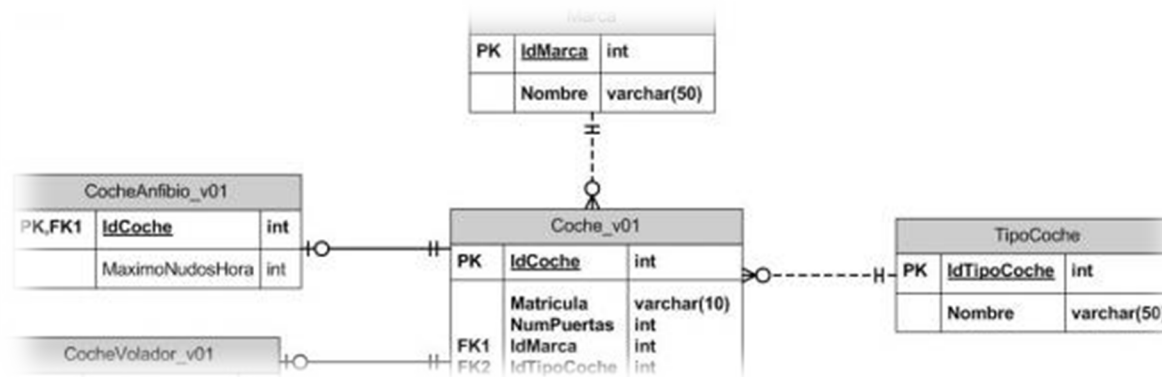
2.2 Claves y dominios del modelo lógico

En el modelo relacional existen ciertas **reglas semánticas o restricciones**:

- ✓ Clave principal (**primary key**). Marca uno o más atributos como identificadores de la tabla.
- ✓ Unicidad (**unique**). Impide que los valores de los atributos marcados de esa forma puedan repetirse. Esta restricción debe indicarse en todas las claves alternativas.
- ✓ Obligatoriedad (**not null**). Prohíbe que el atributo marcado de esta forma quede vacío, es decir, que pueda contener un valor nulo.

2.3 Representación de una tabla en el modelo relacional

- Hay una **confusión general** de términos respecto a la manera de representar el modelo relacional.
- Hay bibliografía que habla de estos diagramas del modelo relacional como modelo Entidad-Relación lógico (en oposición al conceptual).
- Nosotros le llamaremos **diagrama relacional** para evitar confusiones.



2.3 Representación de una tabla en el modelo relacional

En principio las entidades fuertes del modelo E/R son transformadas al modelo relacional siguiendo estas instrucciones:

- **Entidades.** Pasan a ser **tablas**.
- **Atributos.** Se convierten en columnas o campos de tabla.
- **Identificadores principales.** Pasan a ser **claves primarias**.
- **Identificadores candidatos.** Pasan a ser **claves candidatas**.

2.3 Representación de una tabla en el modelo relacional

- La tabla tendrá un **título** con su nombre.
- Dentro de la tabla se irán especificando los **atributos, tipos de datos y las reglas semánticas** de la siguiente manera:
 - nombreAtributo: tipoDeDatos (reglas semánticas)
- Donde la regla semántica puede ser:
 - **PK: Primary Key (Clave primaria)**
 - UNQ: Unique (Único o clave alternativa)
 - NN: Not Null (No puede ser nulo)
 - *FK: Foreign Key (Clave Ajena)*

CLIENTE
DNI: texto (PK)
NSS: num (UNQ, NN)
nombre: texto (NN)
apellido1: texto (NN)
apellido2: texto

3.1 Clave ajena e integridad referencial

Campo de clave ajena

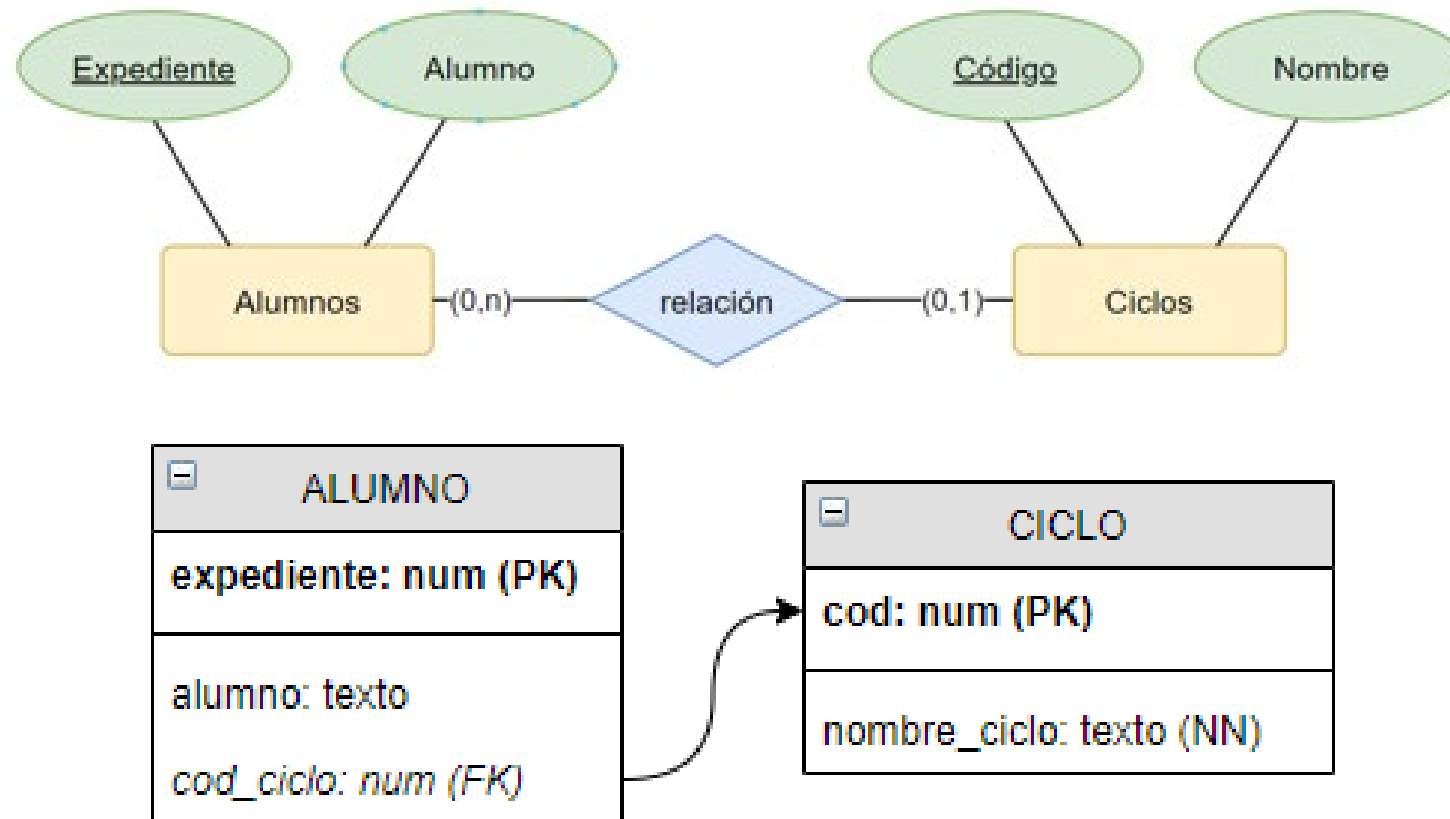
- Las claves ajenas son campos de una tabla cuyos valores están **relacionados** con atributos de otra tabla.
- Estos campos sirven para relacionar unas tablas con otras.
- Las claves ajenas estarán relacionadas siempre con el atributo de la **clave primaria de la tabla a la que apuntan**.
- Esta relación entre dos registros se determina estableciendo un **valor en la clave ajena de una tabla que coincide con un valor de la clave primaria** en la tabla a la que apunta.

3.1 Clave ajena e integridad referencial

Integridad referencial

- ✓ Es una regla de restricción de la **clave ajena**.
- ✓ Establece que debe contener **valores que coincidan con la clave principal** de la tabla que relacionan.
- ✓ Por tanto, las claves ajenas siempre **tienen que pertenecer al mismo dominio** (tipo de datos) que las claves primarias a las que apuntan.
- ✓ La clave ajena, por lo general, **será nula** (no apuntar a ninguna tabla) **cuando un registro no apunte a ningún registro** de otras tablas (cardinalidad mínima 0).

3.1 Clave ajena e integridad referencial



3.1 Clave ajena e integridad referencial

<u>Expediente</u>	Alumno	Código Ciclo (Clave ajena)	<u>Código</u> (Clave primaria)	Nombre ciclo
175	Andrés	2	1	ASIR
102	Sofía	3	2	DAM
305	Pablo		3	DAW
419	Julia	1		
533	Inés	1		

- El atributo **Código Ciclo** en la primera tabla, es una **clave ajena** cuyos valores coinciden con los valores de la clave primaria de la tabla de profesores.
- Sirve para **relacionar** el alumno con el profesor.

3.2 Orden de campos

Orden de los campos

- ✓ Aunque el orden de los campos es irrelevante, seguiremos un orden establecido que nos facilitará la lectura e interpretación de las tablas adecuadamente.
- ✓ Este orden será el siguiente:
 1. El campo de clave primaria o, en caso de clave primaria compuesta, los campos de clave primaria.
 2. Campos UNQ.
 3. Campos NN.
 4. Campos sin restricciones.
 5. Campos de clave ajena.

Ojo: En caso de campos de **clave primaria y ajena simultáneamente** irán en primera posición como las claves primarias ordinarias.

3.2 Orden de campos

Orden de los campos

Ojo

- ✓ En caso de campos de **clave primaria y ajena simultáneamente** irán en primera posición como las claves primarias ordinarias.
- ✓ En caso de **campos UNQ y NN** (claves alternativas) irán justo después de las PK.

PERSONA
DNI: num (PK)
NSS: num (UNQ, NN)
nombre: texto (NN)
apellido1: texto (NN)
apellido2: texto
fecha_nac: fecha
género: texto
cod_empleo: num (FK, NN)

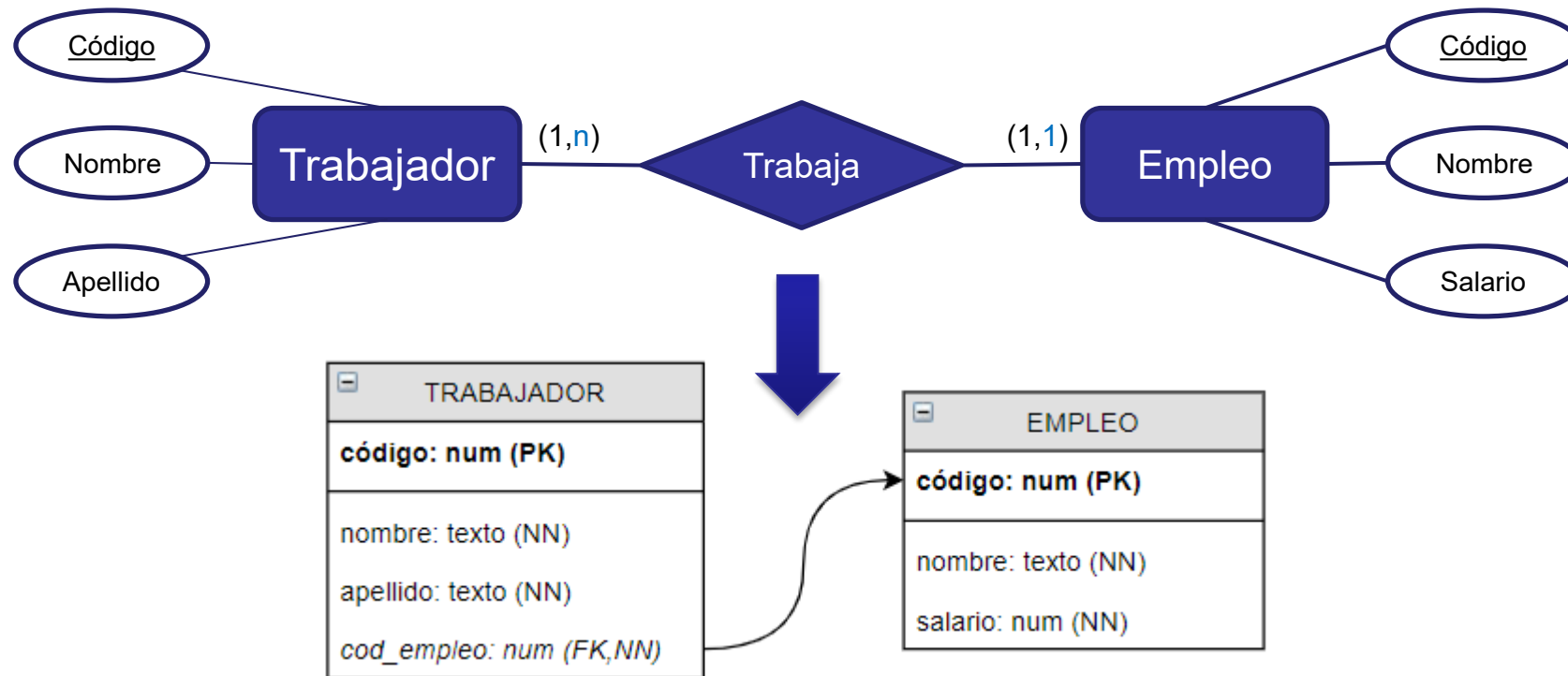
3.3 Transformaciones de relaciones uno a varios

Relaciones con sólo una entidad con cardinalidad N: UNO A VARIOS

- Son las relaciones de (1,N) y en este caso **no se crea ninguna tabla nueva**.
- Simplemente, a la tabla que tiene cardinalidad N, se le añadirá el atributo de clave principal (que actuará como clave ajena) de la entidad con cardinalidad 1.
- Las llamaremos relaciones **uno a varios**.
 - ✓ Es decir, **una ocurrencia** de la entidad A una entidad está relacionada con **varias ocurrencias** de la entidad B.
 - ✓ Sin embargo **una ocurrencia** de la entidad B solo está relacionada con **una entidad** de la entidad A.

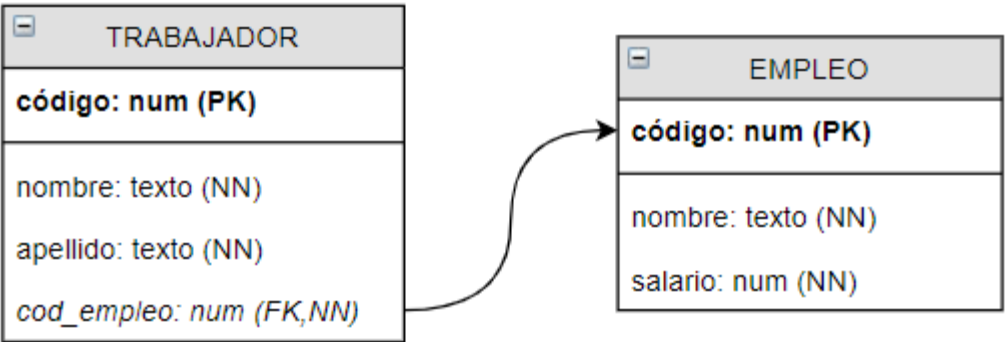
3.3 Transformaciones de relaciones uno a varios

Relaciones con sólo una entidad con cardinalidad N: UNO A VARIOS



3.3 Transformaciones de relaciones uno a varios

Relaciones con sólo una entidad con cardinalidad N: UNO A VARIOS



<u>Código</u>	Nombre	Ape1	Cód_Empleo FK
T001	Andrés	Dorado	E001
T002	Sofía	Benito	E001
T003	Pablo	Garcés	E002
T004	Julia	Delgado	E003
T005	Inés	Moreno	E003

<u>Código</u>	Nombre	Salario
E001	Barrendera	10
E002	Médico	5
E003	Ingeniera	30

Luis Dorado Garcés

3.3 Clave primaria compuesta

Clave primaria compuesta

Combinación de dos o más campos cuyos valores no puede repetirse y no pueden ser nulos.

GRUPO
ciclo: texto (PK)
curso: num (PK)
tutor: num (FK, NN)
aula: num (FK)

COCHE
marca: texto (PK)
modelo: texto (PK)
año: fecha (PK)
gama: texto (NN)
precio: num

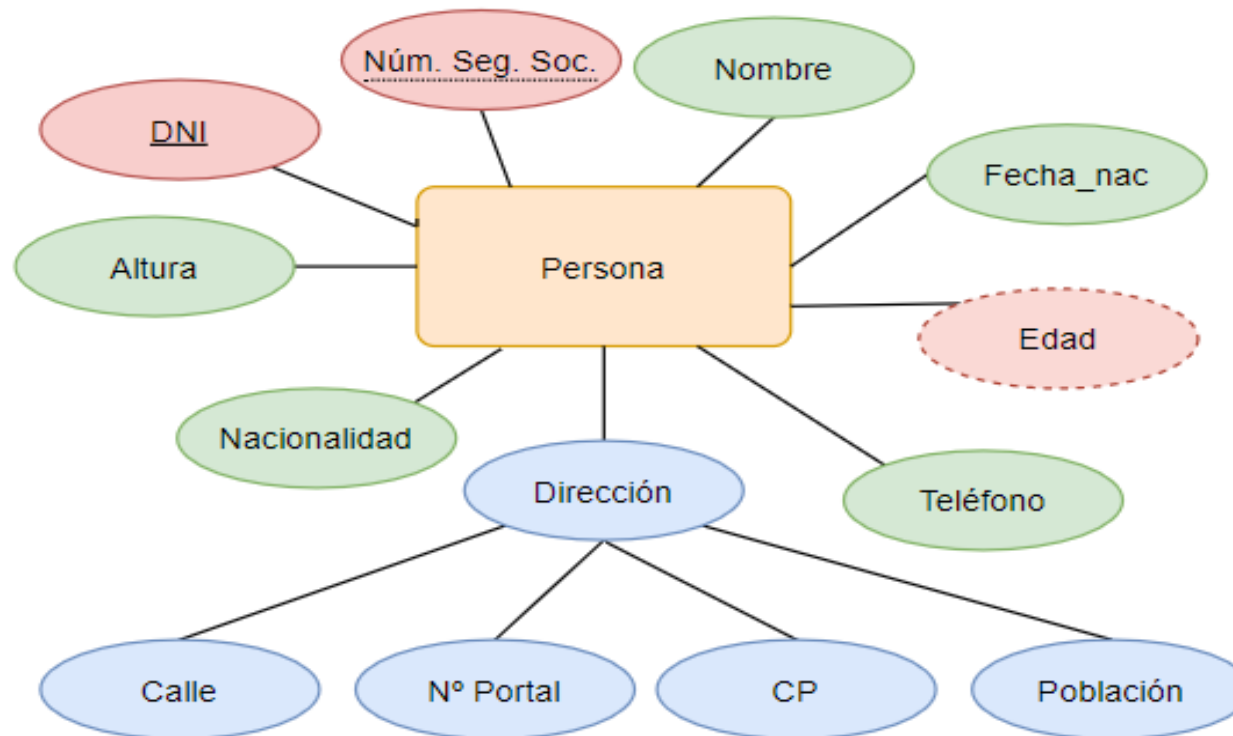
3.5 Transformación de atributos compuestos y derivados

Atributos compuestos y derivados

- **Atributos compuestos.** Cada sub-atributo se transforma en un campo.
- **Atributos derivados.** No se incluyen en la tabla ya que no se almacenan.

3.5 Transformación de atributos compuestos y derivados

Atributos compuestos y derivados



PERSONA
DNI: num (PK)
NSS: num (UNQ, NN)
nombre: texto (NN)
* fecha_nac: fecha *
Calle: texto
Nº Portal: num
CP: num
Población: texto
Teléfono: num
Altura: num
Nacionalidad: texto

3.6 Transformaciones de relaciones varios a varios

Relaciones con dos entidades tienen cardinalidad N: **VARIOS A VARIOS**

- Es decir, **una ocurrencia** de la entidad A una entidad está relacionada con **varias ocurrencias** de la entidad B.
- También **una ocurrencia** de la entidad B solo está relacionada con **varias entidad** de la unidad A.

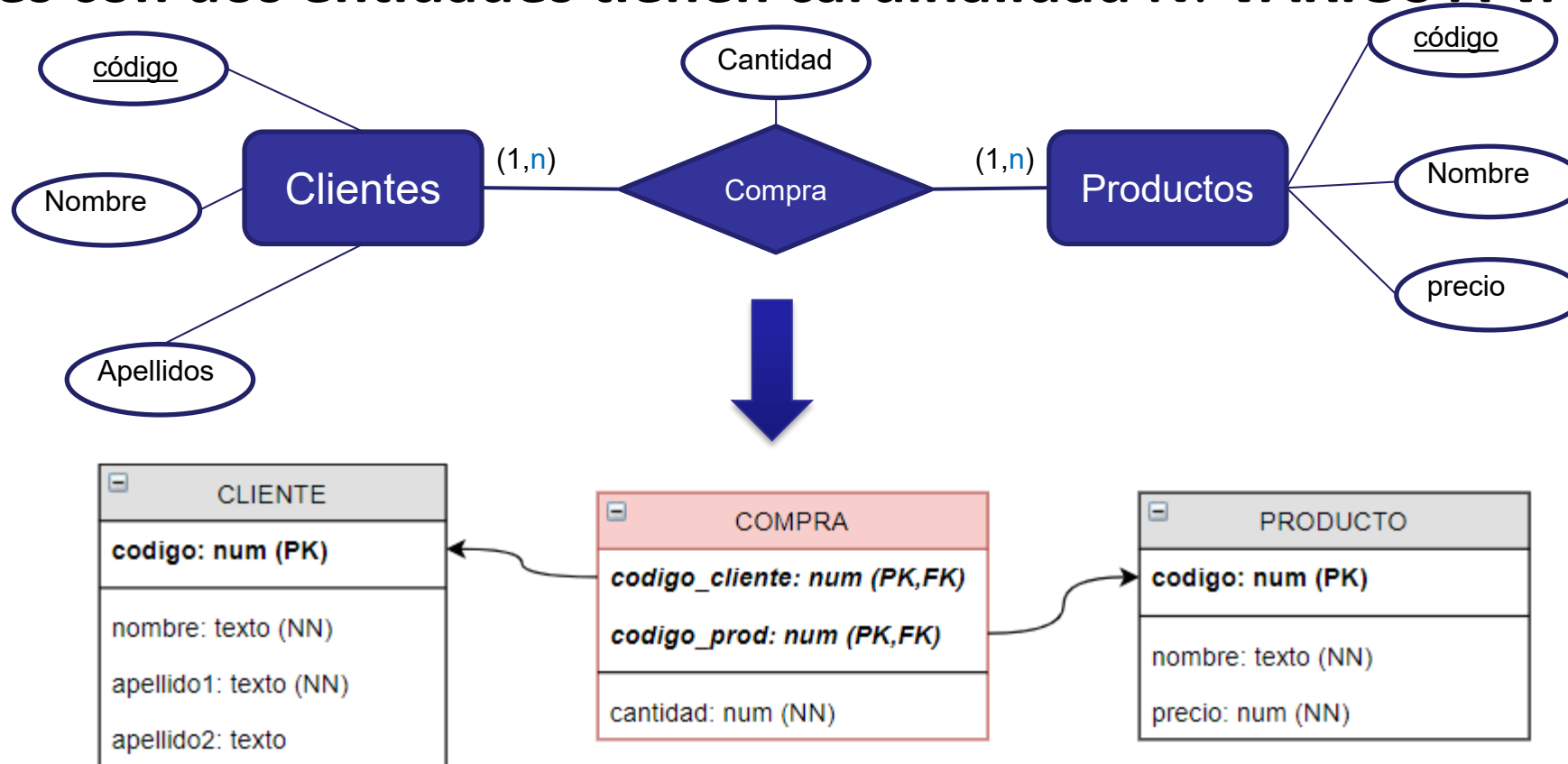
3.6 Transformaciones de relaciones varios a varios

Relaciones con dos entidades tienen cardinalidad N: VARIOS A VARIOS

- El proceso de transformación obliga a crear una **nueva tabla que contendrá las claves primarias** de las entidades que participan en la relación.
- Éstas, actuarán de **clave primaria** de la nueva tabla.
- Además todos estos atributos importados, serán **claves ajenas** respecto a las entidades de las que provienen.
- Si la relación tenía atributos, éstos formarán parte de la tabla.

3.6 Transformaciones de relaciones varios a varios

Relaciones con dos entidades tienen cardinalidad N: VARIOS A VARIOS



3.6 Transformaciones de relaciones varios a varios

Relaciones con dos entidades tienen cardinalidad N: varios a varios

En este caso **surgen 3 nuevas tablas**:

Cientes (código, nombre, apellidos,...)

CP {código} → *Clave Principal*

Productos (código, nombre, precio,...)

CP {código} → *Clave Principal*

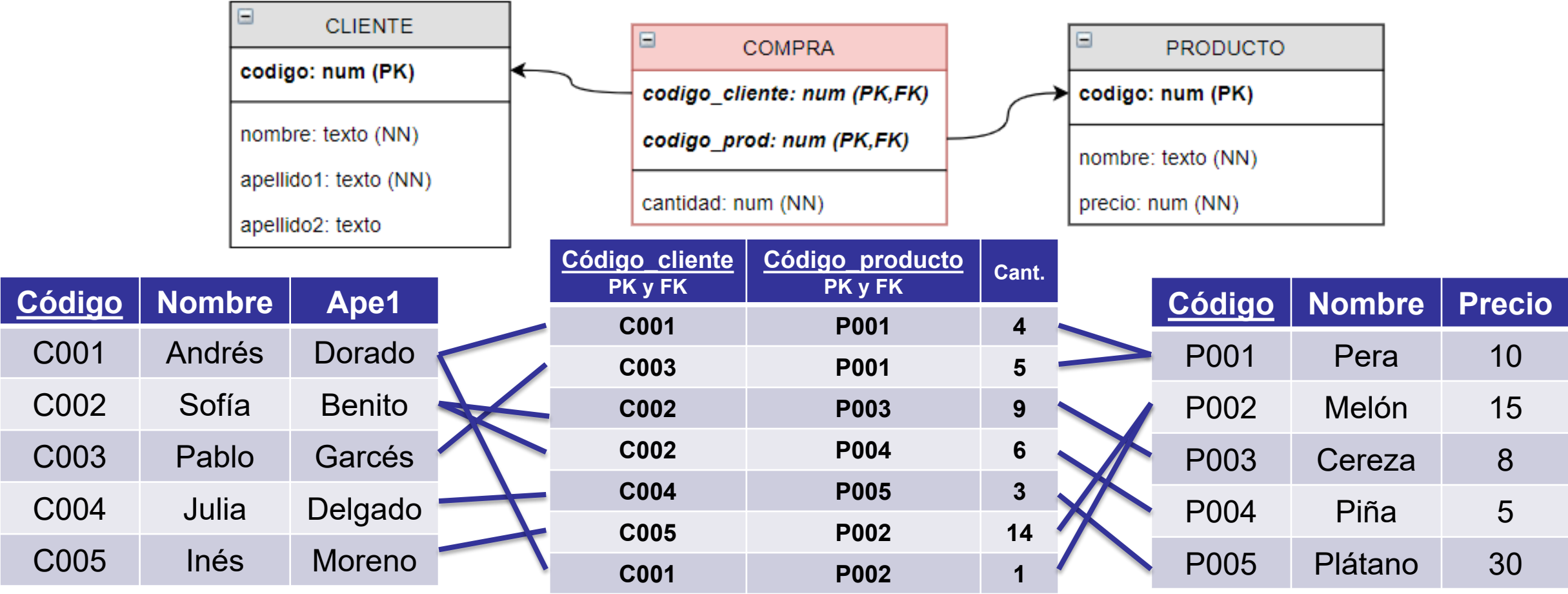
Compra(código_cliente, código_producto, cantidad)

CP {código_cliente, código_producto} → *Clave Principal*

CAj {código_cliente} referencia a Cliente → *Clave Ajena*

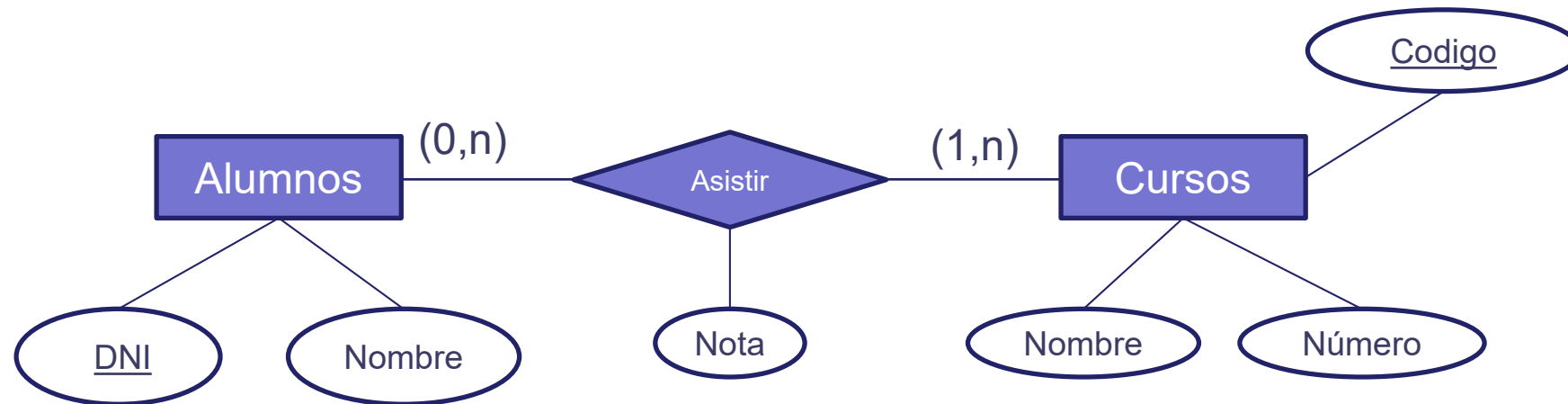
CAj {código_producto} referencia a Producto → *Clave Ajena*

3.6 Transformaciones de relaciones varios a varios



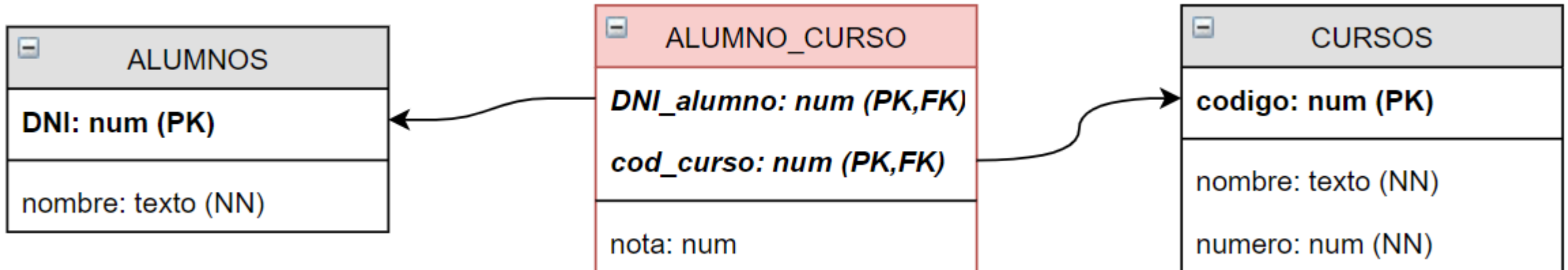
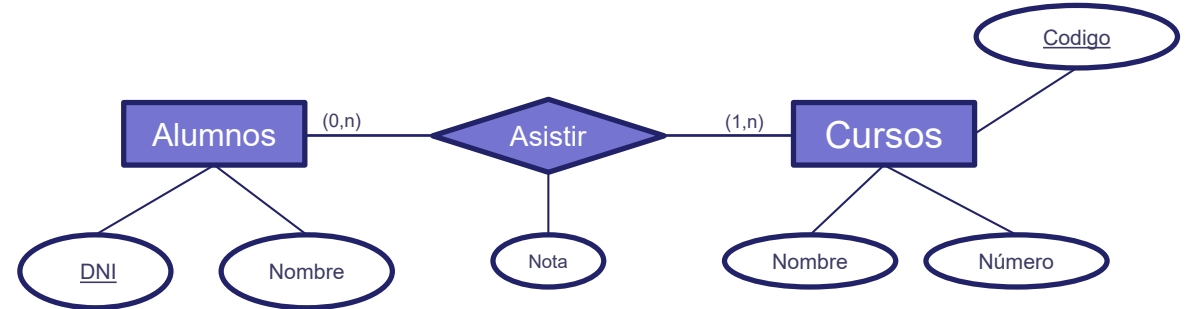
3.6 Transformaciones de relaciones varios a varios

Ejemplo



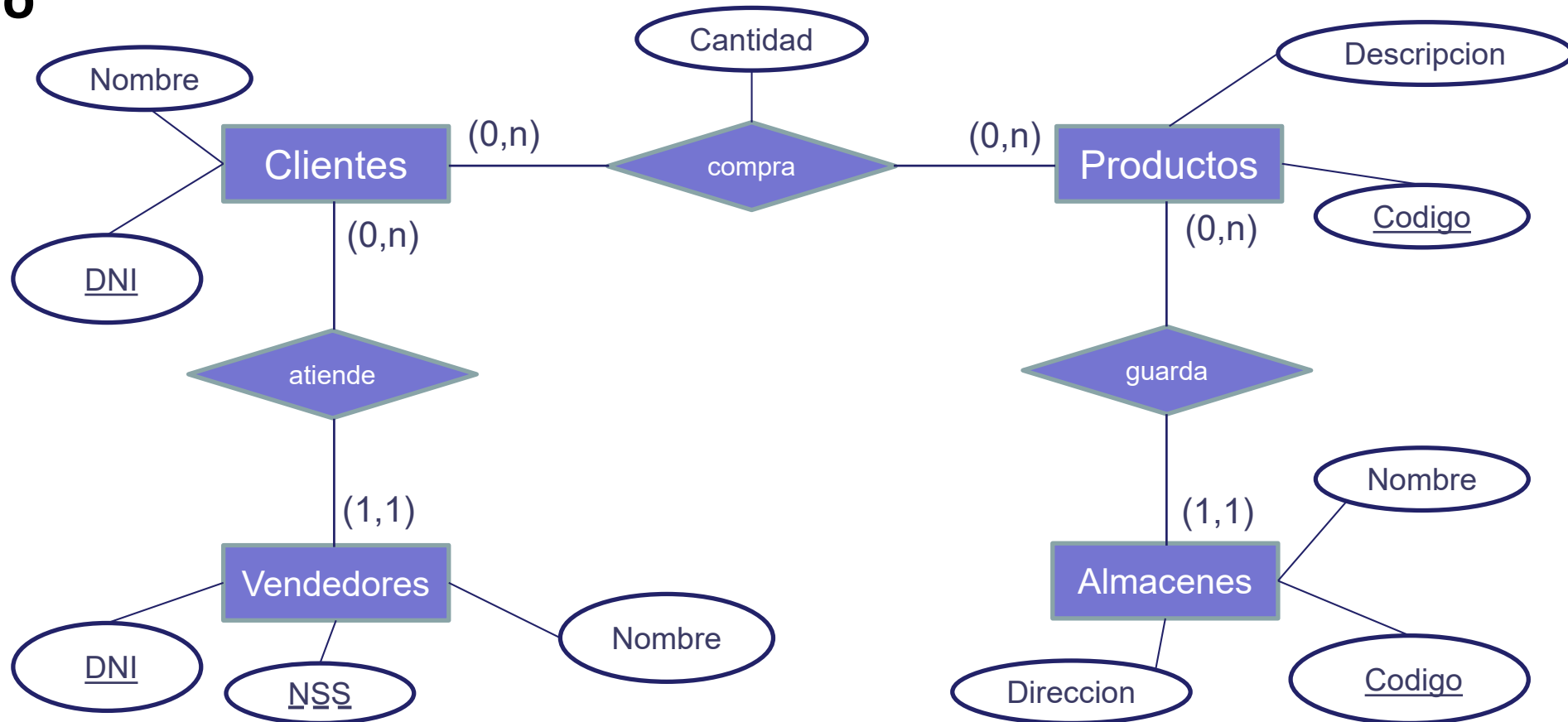
3.6 Transformaciones de relaciones varios a varios

Ejemplo



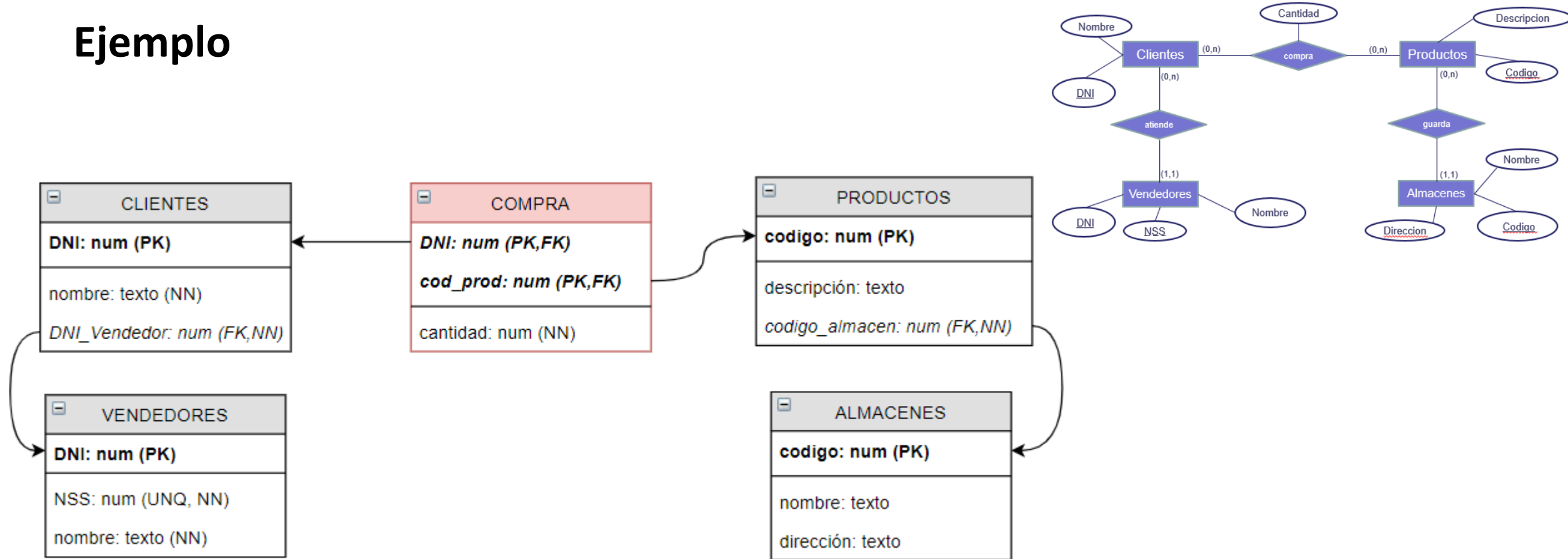
3.6 Transformaciones de relaciones varios a varios

Ejemplo



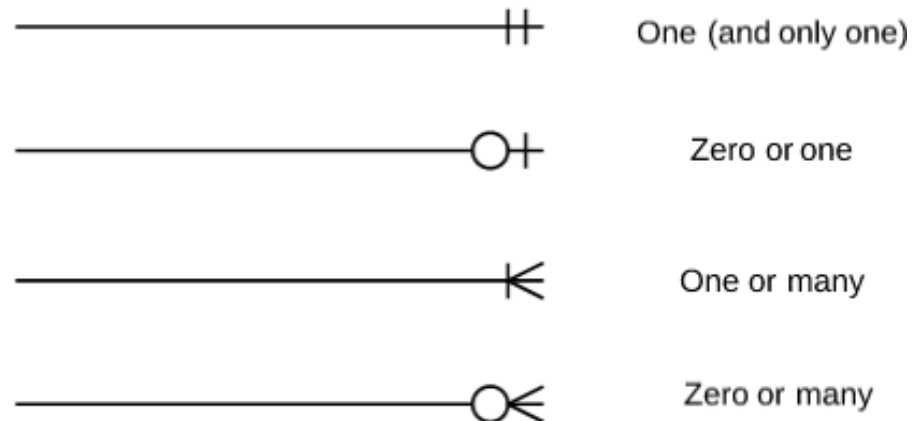
3.6 Transformaciones de relaciones varios a varios

Ejemplo



3.7 Cardinalidades en el modelo relacional

- Las relaciones ilustran una **asociación** entre dos tablas.
- Las relaciones **siempre** vincularán **una clave ajena de una tabla** (la que apunta) **con una clave primaria** (la que es apuntada).
- En el modelo de datos físicos, las relaciones están representadas por **líneas estilizadas**.



3.7 Cardinalidades en el modelo relacional

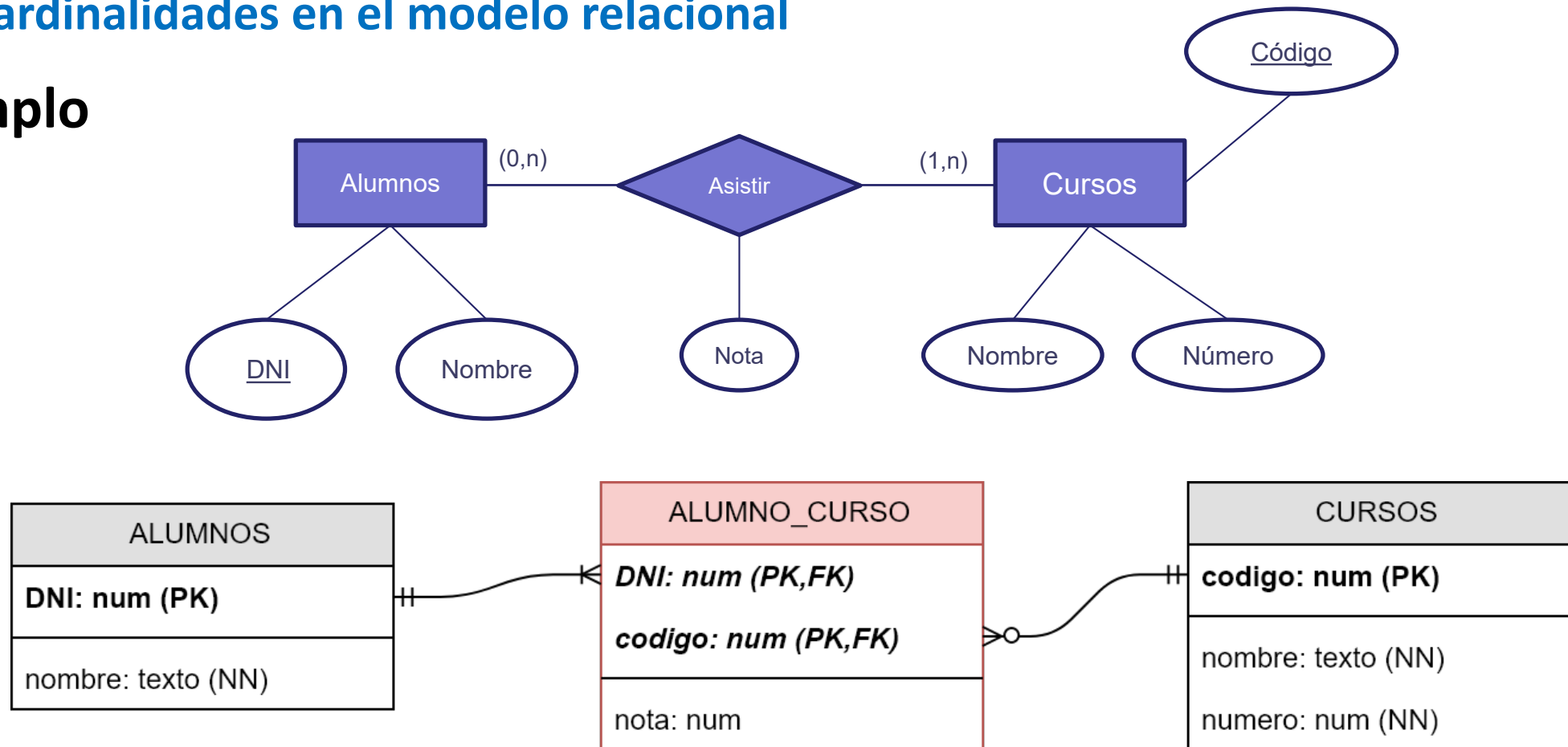
Recuerda: la **cardinalidad** se refiere al **máximo y mínimo** número de veces que una instancia en una entidad puede ser asociada con ocurrencias en la entidad relacionada.

El primer símbolo será la cardinalidad mínima (1 ó 0) y el segundo para la máxima (1 ó ∞)

_____	One (and only one)
_____○	Zero or one
_____<	One or many
_____○<	Zero or many

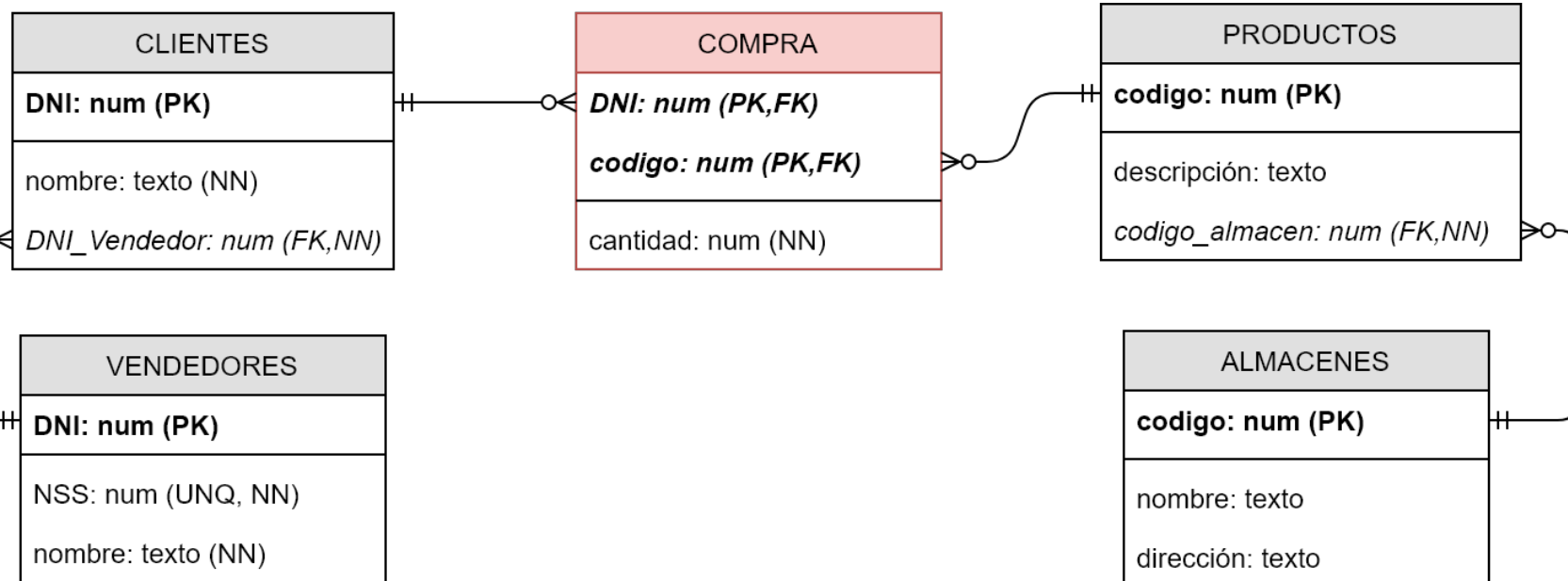
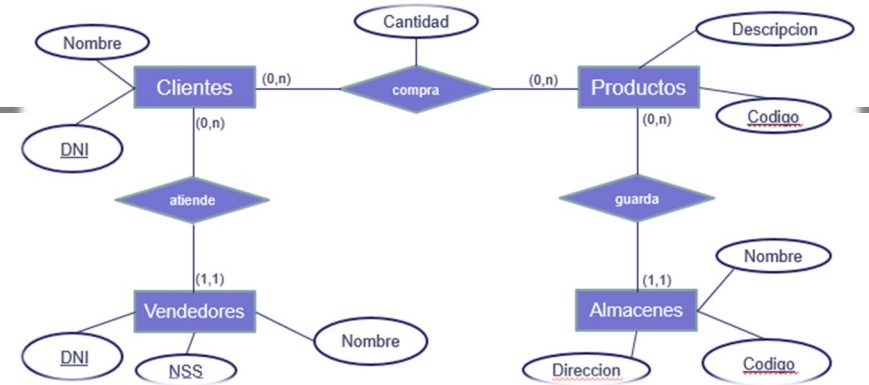
3.7 Cardinalidades en el modelo relacional

Ejemplo



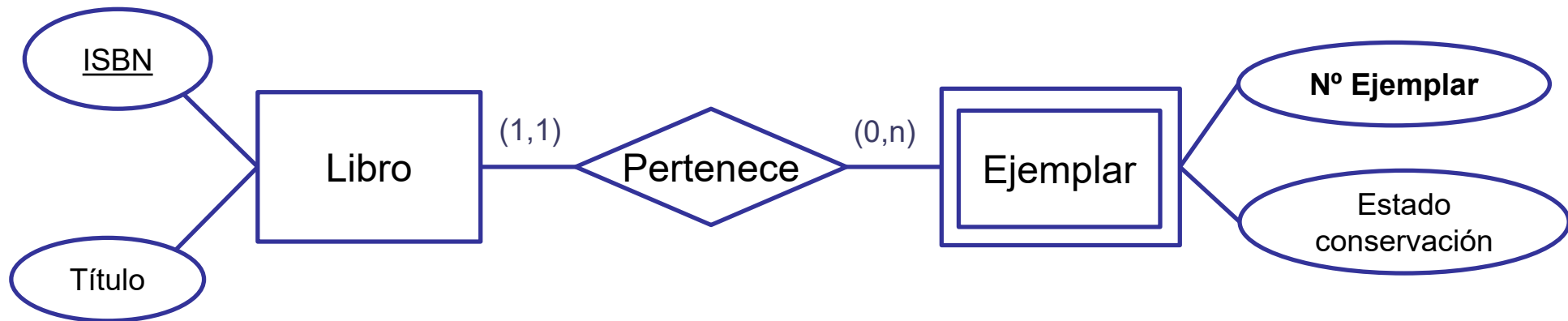
3.7 Cardinalidades en el modelo relacional

Ejemplo



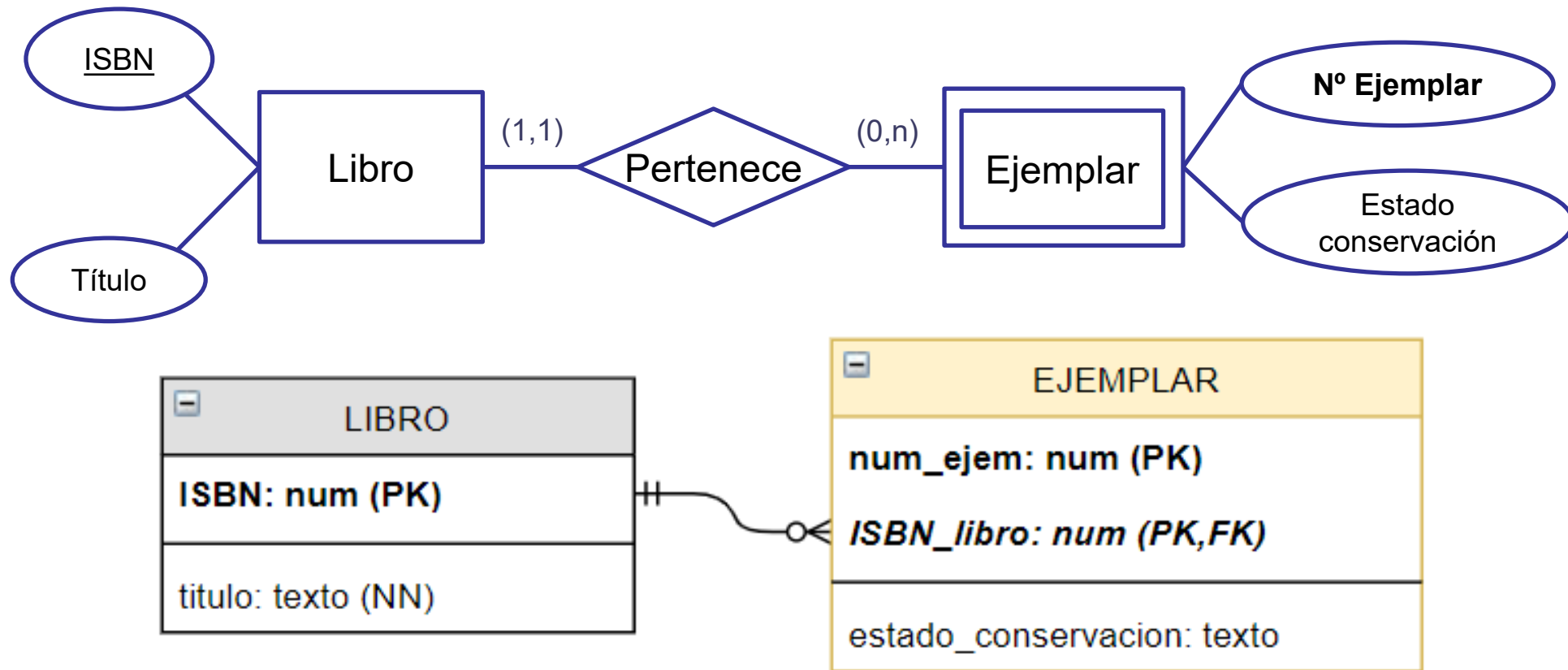
4.1 Transformación de entidades débiles

- ✓ Toda entidad débil implica una **relación implícita con una entidad fuerte**.
- ✓ Esta relación no necesita incorporarse como una nueva tabla en el modelo relacional. Basta con **añadir como atributo y clave ajena en la entidad débil**, el identificador de la fuerte.



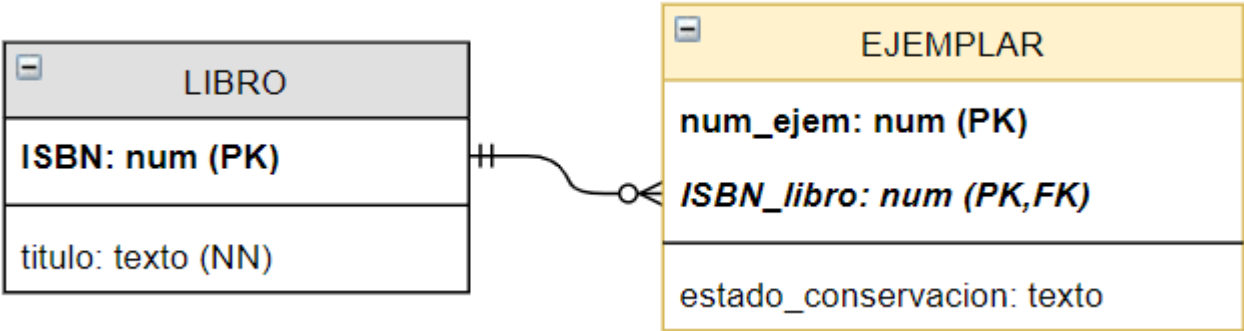
4.1 Transformación de entidades débiles

Toda entidad débil implica una relación implícita con una entidad fuerte.

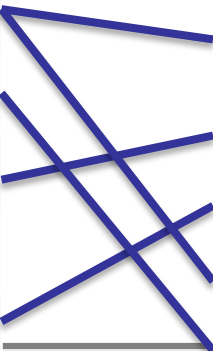


4.1 Transformación de entidades débiles

Toda entidad débil implica una relación con una entidad fuerte.



<u>ISBN</u>	Título
C001	El Capital
C002	Así habló Zaratrusta
C003	1984
C004	Vigilar y Castigar
C005	La Naranja Mecánica



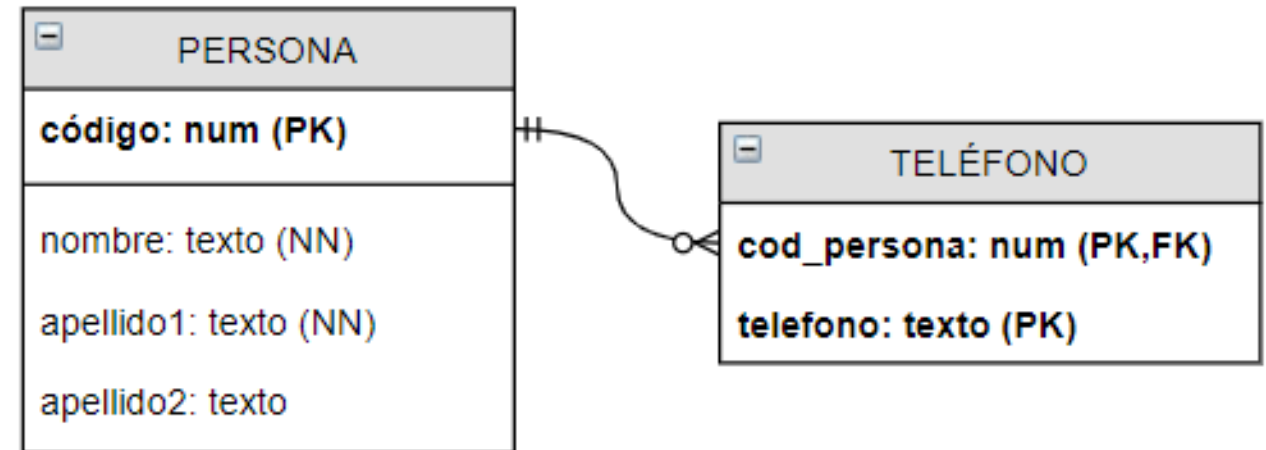
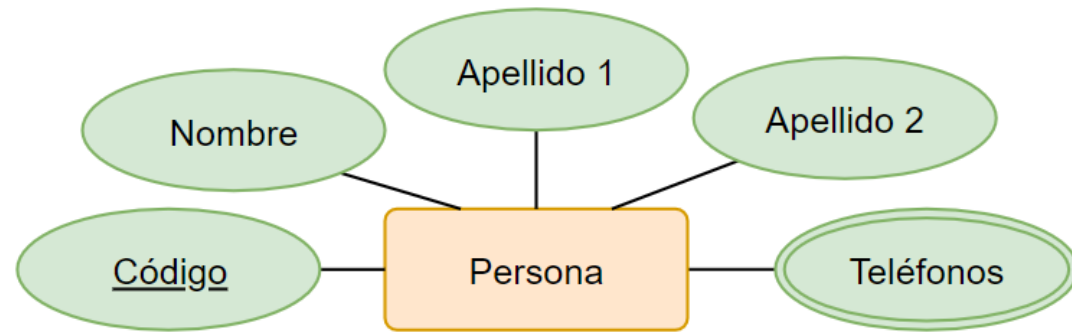
<u>Núm. ejemplar</u> PK	<u>ISBN libro</u> PK y FK	Estado conserv.
01	C001	Bueno
01	C003	Malo
01	C005	Lamentable
02	C001	
02	C002	Regulín

4.2 Transformación de atributos multivaluados

- Se realizará de una manera **similar a las entidades débiles**.
- Crearemos **una nueva tabla que tendrá una clave primaria compuesta** por:
 1. Un campo de clave ajena que apunte a la tabla a la que pertenece el atributo multivaluado.
 2. Un campo que almacene los valores del atributo.

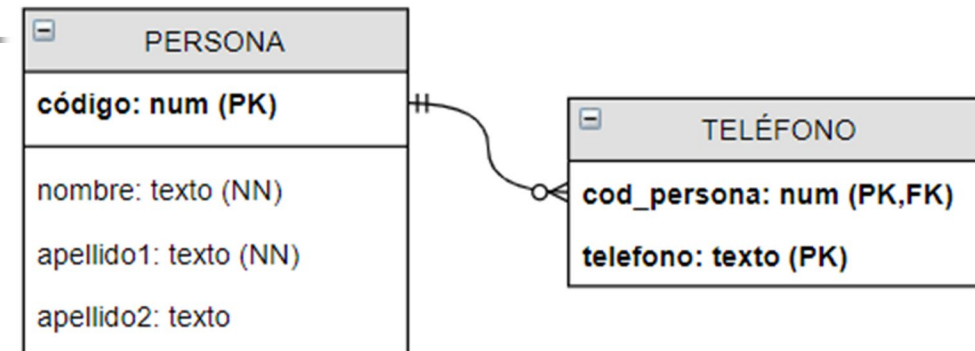
4.2 Transformación de atributos multivaluados

- Se realizará de una manera **similar a las entidades débiles**.



4.2 Transformación de atributos multivaluados

- Se realizará de una manera **similar a las entidades débiles**.



<u>Código</u>	Nombre	Ape1	<u>Cod_persona</u> PK y FK	Teléfono PK
C001	Andrés	Dorado	C001	941252525
C002	Sofía	Benito	C001	948658546
C003	Pablo	Garcés	C003	914567896
C004	Julia	Delgado	C004	914567896
C005	Inés	Moreno	C004	931254654

- ✓ Dos personas pueden compartir un número pero una persona no puede tener el mismo número dos veces.

4.3 Transformación de relaciones uno a uno

Relaciones en las que todas las entidades participan con cardinalidad 1 como máximo (1:1)

- ✓ Las llamaremos relaciones **uno a uno**.
- ✓ Es decir, **una ocurrencia** de la entidad A una entidad está relacionada con **una ocurrencia** de la entidad B.
- ✓ También **una ocurrencia** de la entidad B solo está relacionada con **una entidad** de la entidad A.

4.3 Transformación de relaciones uno a uno

Tendremos 3 casos posibles:

1. **Caso 1:** Si las dos entidades participan con participación (0,1), entonces se crea una nueva tabla para la relación.
2. **Caso 2:** Si solo una entidad participa con cardinalidad (0,1), entonces se pone la clave ajena en dicha entidad como UNQ y No Nulo .
3. **Caso 3:** Si ambas entidades tienen cardinalidad (1,1) elegimos la clave principal de una de ellas (cualquiera) y la introducimos como clave ajena en la otra tabla como NN (cardinalidad mínima) y UNQ (cardinalidad máxima).

4.3 Transformación de relaciones uno a uno

Tendremos 3 casos posibles:

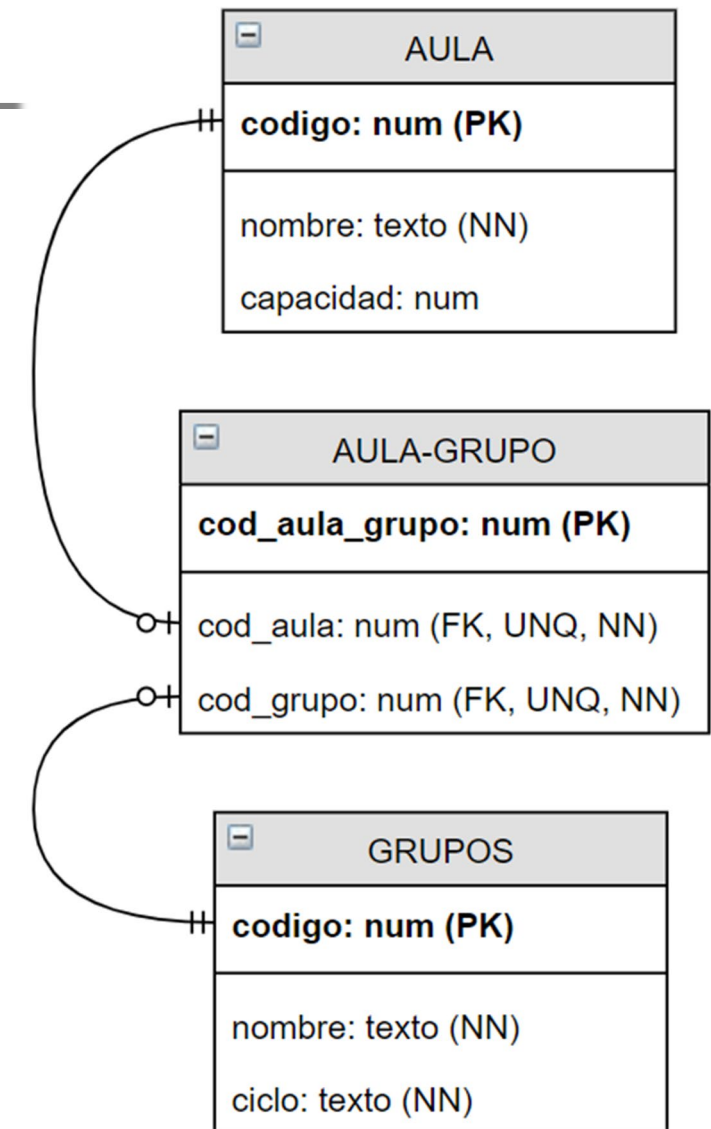
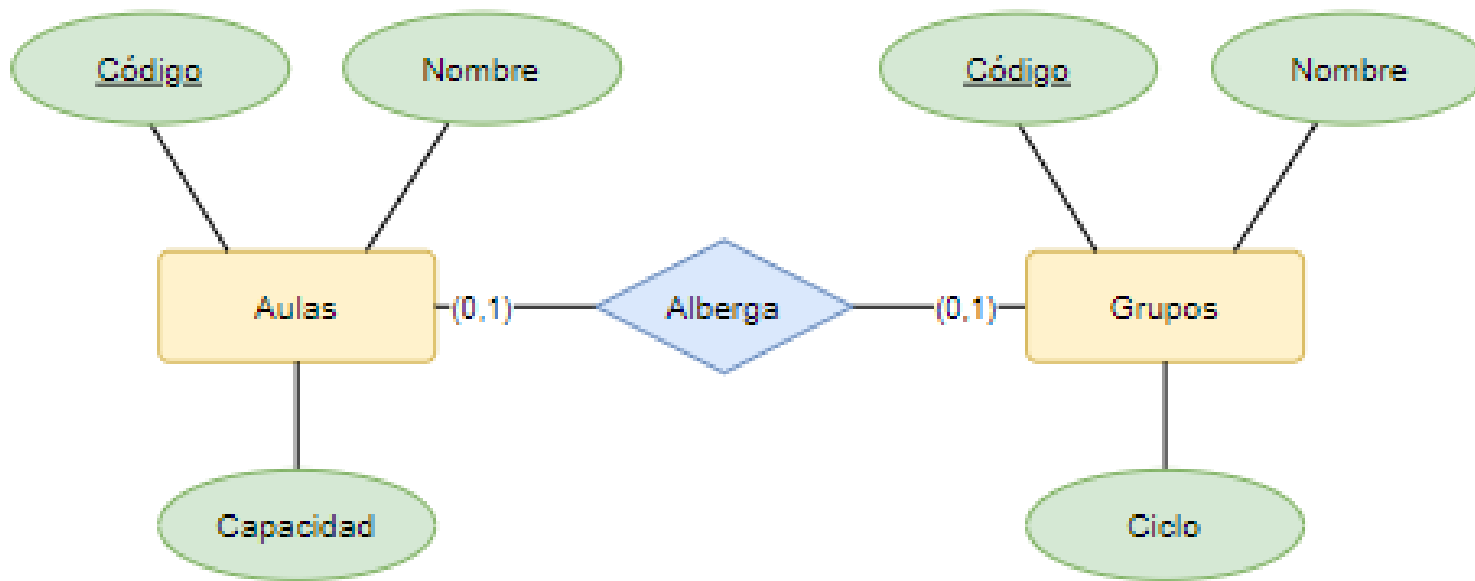
Caso 1

Si las dos entidades participan con participación **(0,1)**, entonces **se crea una nueva tabla** para la relación.

- ✓ En esta tabla de relación tendrá **dos claves ajenas serán únicas** para respetar la cardinalidad máxima y **no nulas** para la mínima.
- ✓ Apuntarán a cada una de las entidades unidas por la relación 1:1.
- ✓ Tendrán una **clave primaria independiente** de las claves ajenas.

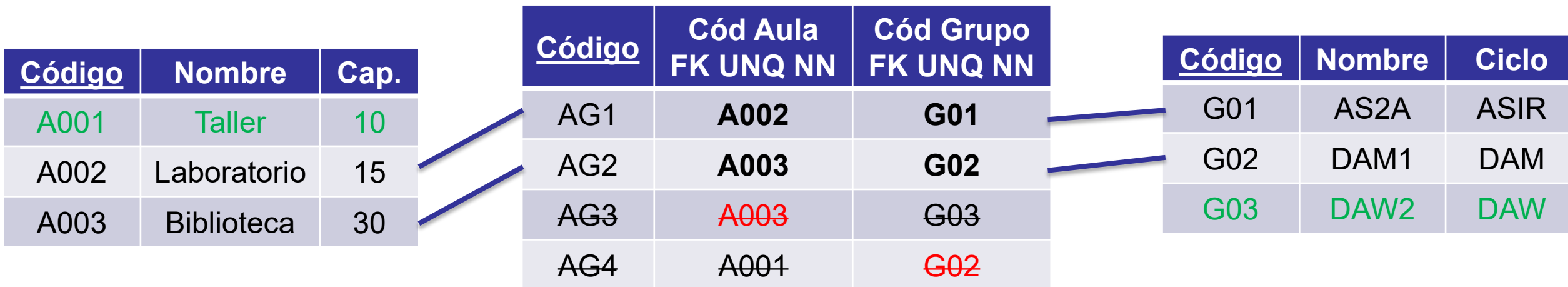
4.3 Transformación de relaciones uno a uno

Caso 1: Un grupo puede estar en ninguna (ON-LINE) o en 1 sola aula. Un aula alberga un grupo o ninguno (esta vacía).



4.3 Transformación de relaciones uno a uno

Caso 1: Un grupo puede estar en ninguna (ON-LINE) o en 1 sola aula.



- ✓ Como la cardinalidad mínima es 0 en ambas entidades, **puede haber grupos y aulas que no están relacionadas con ningún registro.**
- ✓ Como la relación es uno a uno, un aula no puede estar relacionada con más de un grupo y viceversa, es decir, en la nueva tabla no puede repetirse ninguna aula ni grupo.

4.3 Transformación de relaciones uno a uno

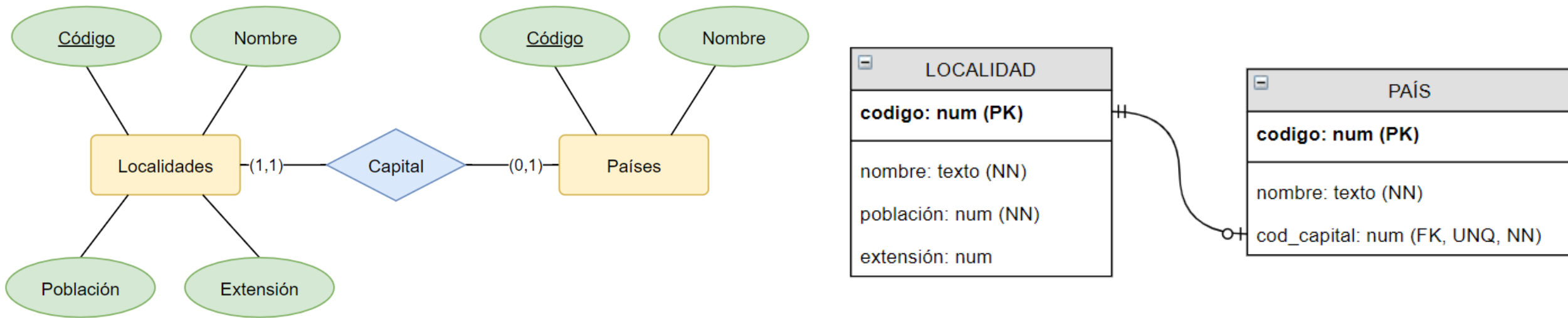
Tendremos 3 casos posibles:

Caso 2

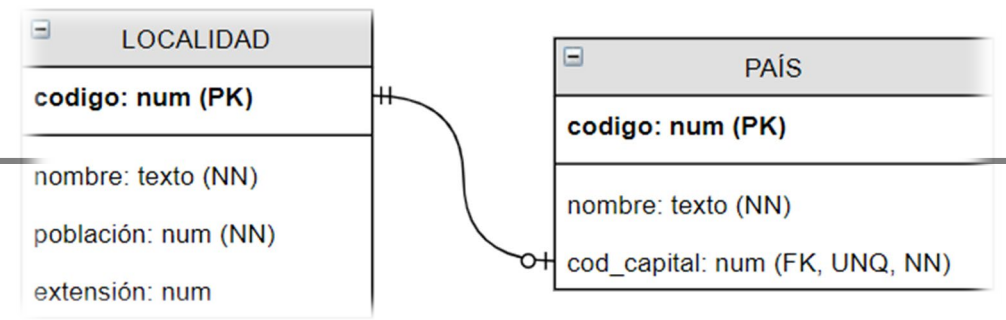
Si solo una entidad participa con cardinalidad (0,1), entonces se pone la clave ajena en dicha entidad, como única (para cardinalidad máxima) y No Nula (cardinalidad mínima).

4.3 Transformación de relaciones uno a uno

Caso 2: Una localidad puede ser capital de un solo país y un país debe tener una sola capital.



4.3 Transformación de relaciones uno a uno



Caso 2: Una localidad puede ser capital de un solo país y un país debe tener una sola capital.

<u>Código</u>	Nombre	Pob.
L001	Logroño	10
L002	Buenos Aires	15
L003	Bratislava	30

<u>Código</u>	Nombre	Cód Localidad FK UNQ NN
P01	Eslovaquia	L003
P02	Argentina	L002
P03	Chile	L002
P04	España	

- ✓ Como las localidades pueden no ser capital, nuestro modelo permite que una localidad no sea apuntada por ningún país.
- ✓ Todo país debe tener una capital (NN) y esta capital no puede compartirse con ningún otro país UNQ.

4.3 Transformación de relaciones uno a uno

Tendremos 3 casos posibles:

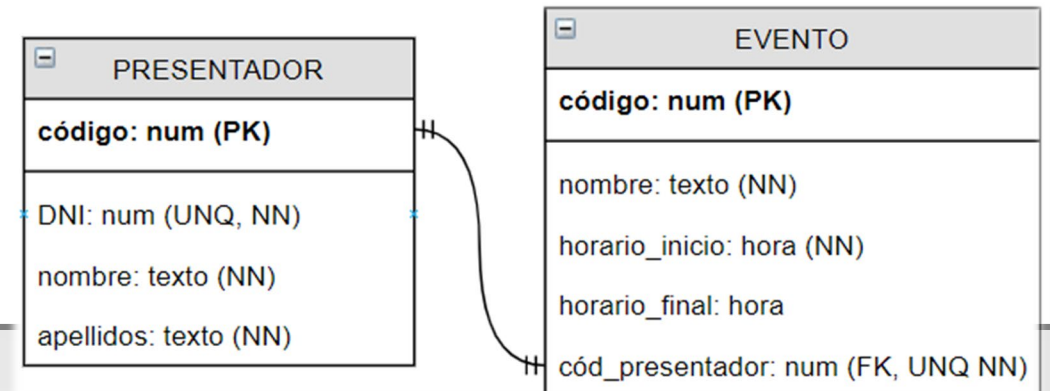
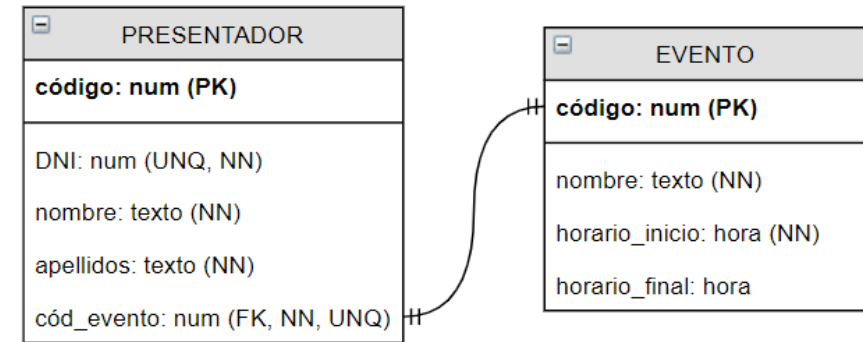
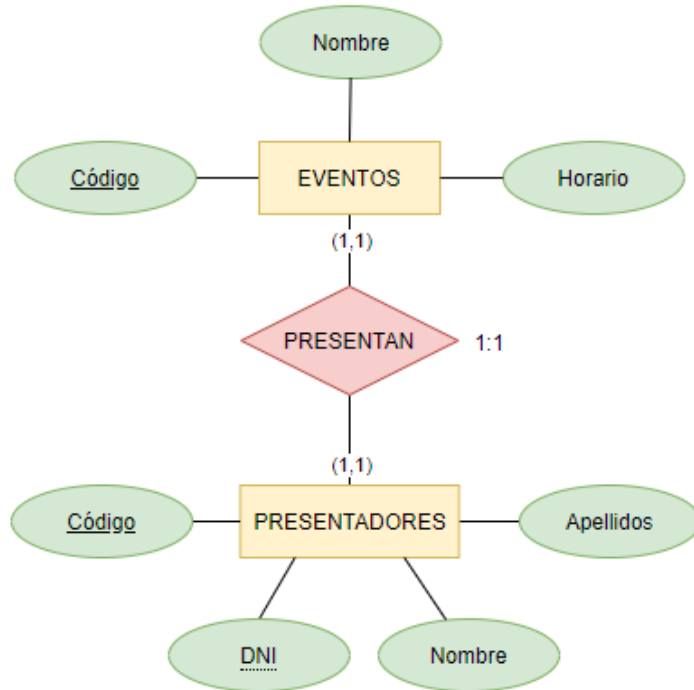
Caso 3

Si **ambas entidades tienen cardinalidad (1,1)** elegimos las tablas una de ellas (**cualquiera**) y le introducimos una **clave ajena única** que apunte a la otra tabla.

- ✓ Se elegirá una u otra forma en función de cómo se quiera organizar la información para facilitar las consultas.
- ✓ En esta etapa de vuestro aprendizaje **elegid la que más rabia os de.**

4.3 Transformación de relaciones uno a uno

Caso 3: Cada presentador presenta 1 y solo un 1 evento. Cada evento es presentado por 1 y solo 1 presentador.



4.4 Transformación de relaciones reflexivas

Las relaciones **reflexivas o unarias** podrán **generar o no una nueva tabla** en función de la cardinalidad de la relación, al igual que las relaciones binarias sencillas.

- **Reflexiva 1:1**

Se aplicarán las reglas según las relaciones 1:1. Tres métodos:

✓ $(0,1)$ y $(0,1)$ // $(0,1)$ y $(1,1)$ // $(1,1)$ y $(1,1)$.

- **Reflexiva 1:N**

Como en las binarias no genera tabla. En la entidad se introduce dos veces la clave, una como clave principal y otra como clave ajena.

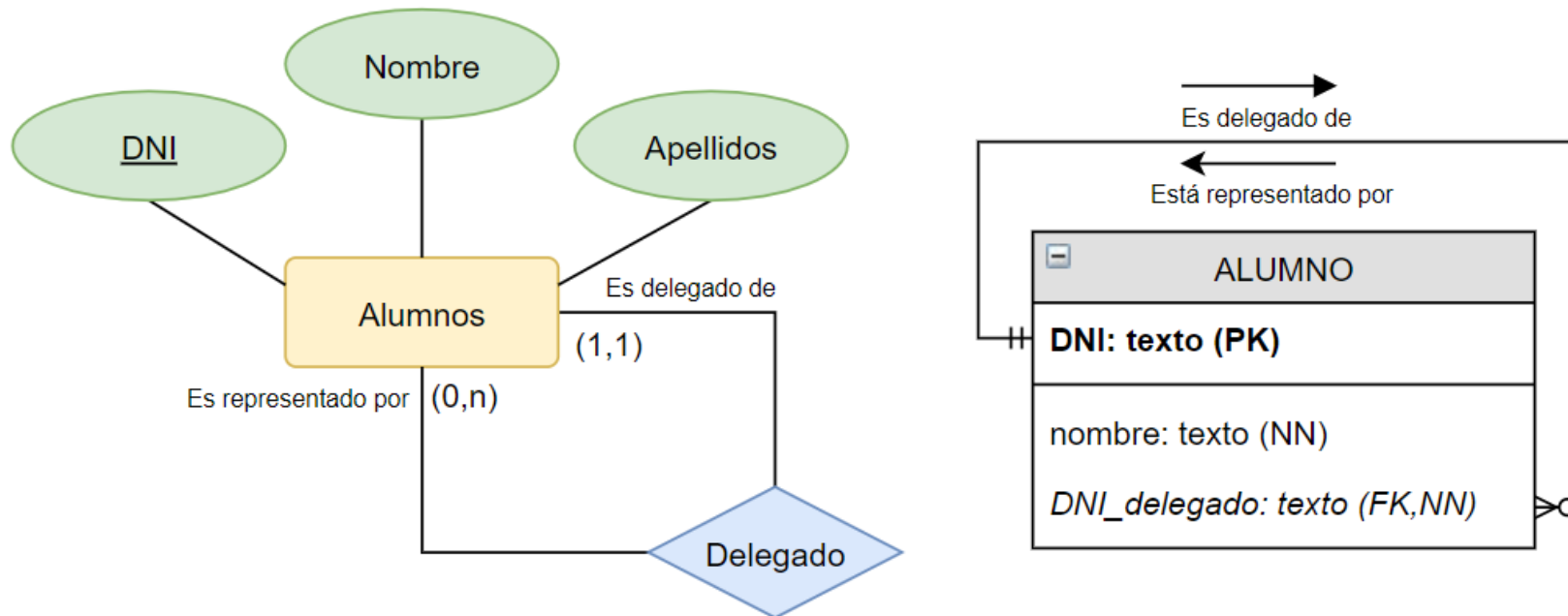
✓ La clave ajena debe ser la del lado 1 y **debe utilizarse el rol para denominarla**.

- **Reflexiva N:M** Como en la binarias la relación genera tabla. La nueva tabla contendrá dos veces la clave, **cada una con el nombre de su rol**.

4.4 Transformación de relaciones reflexivas

Ejemplo relación reflexiva 1:N

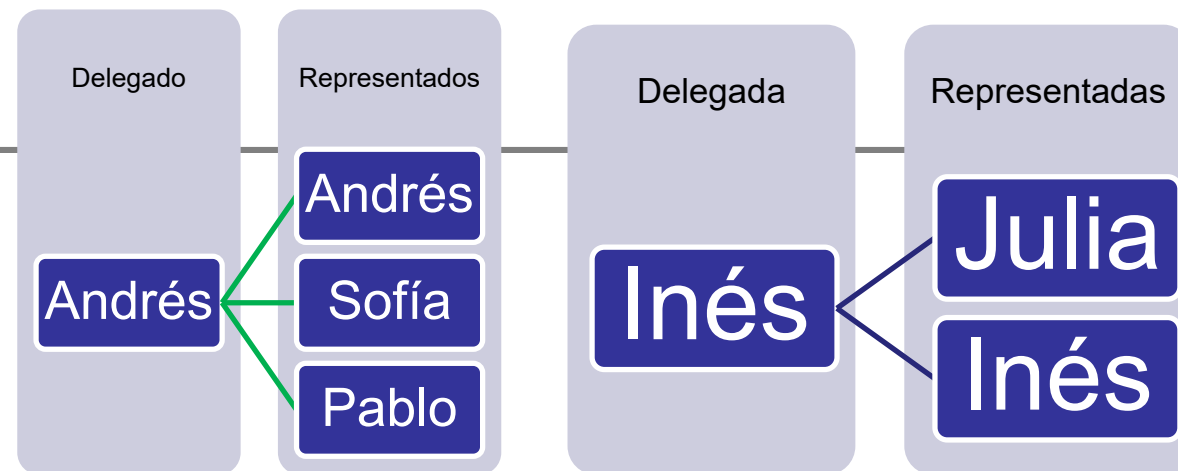
Un alumno tiene 1 y solo 1 delegado. Un alumno puede ser o no ser delegado.



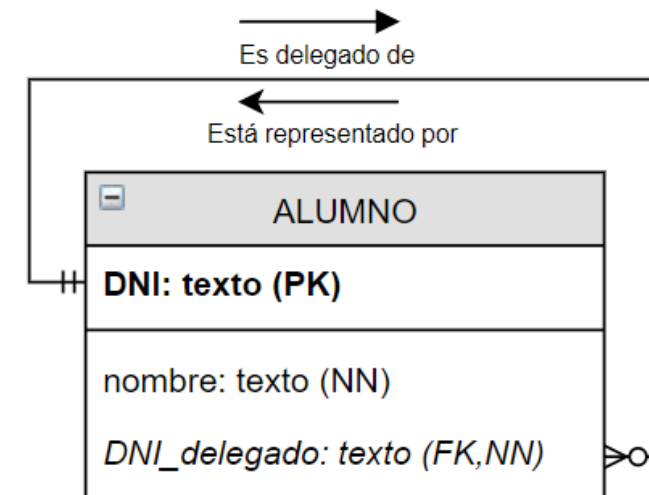
4.4 Transformación de relaciones reflexivas

Ejemplo relación reflexiva 1:N

Un alumno tiene 1 y solo 1 delegado. Un alumno puede ser o no ser delegado.



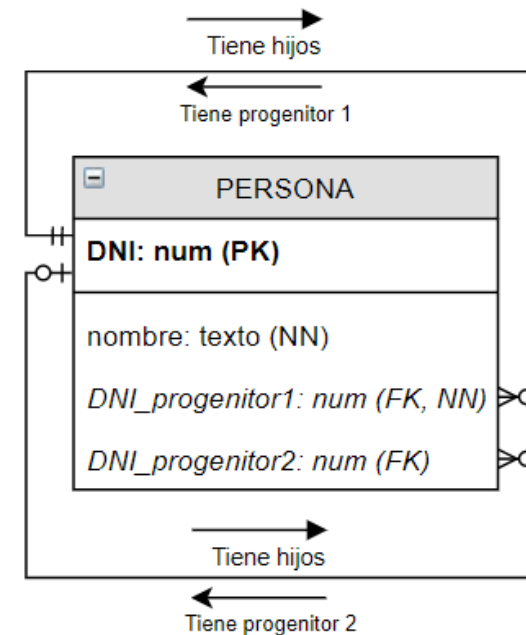
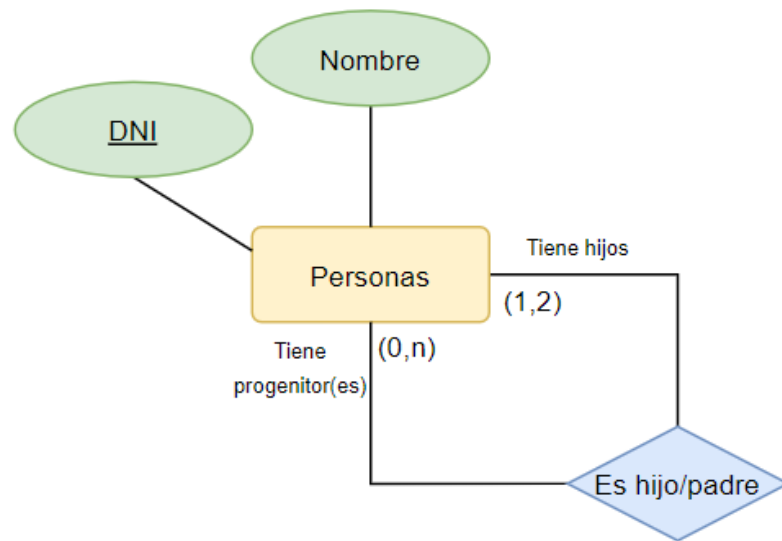
<u>DNI_Alumno</u>	Nombre	Ape1	DNI_delegado FK y NN
115C	Andrés	Dorado	115C
563F	Sofía	Benito	115C
X457	Pablo	Garcés	115C
159R	Julia	Delgado	754Z
754Z	Inés	Moreno	754Z



4.4 Transformación de relaciones reflexivas

Ampliación: Ejemplo relación reflexiva 2:N

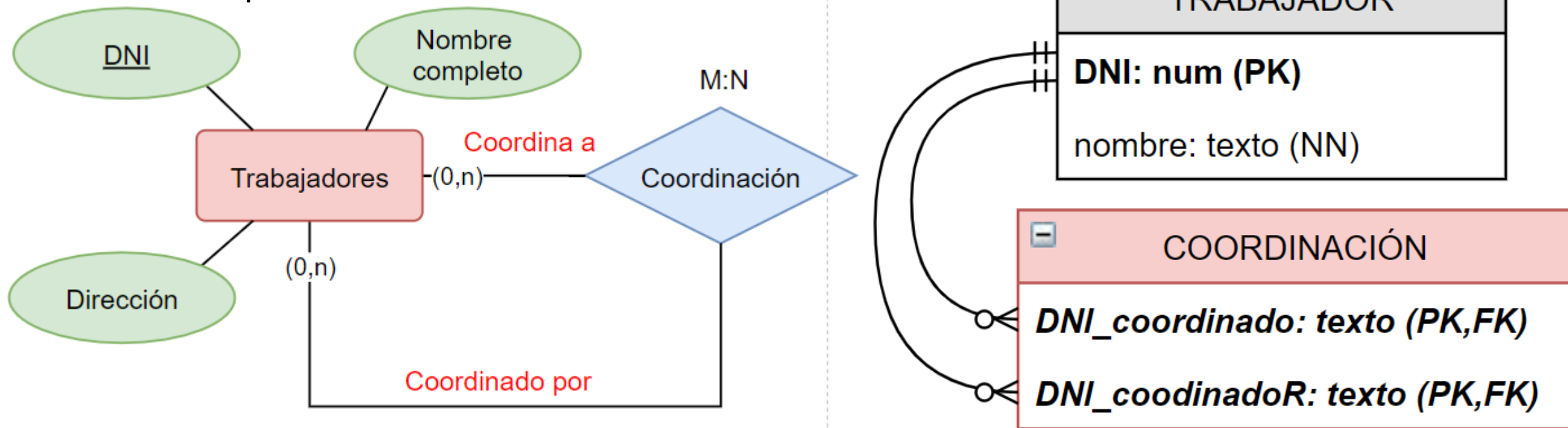
Una persona tiene uno (familia monoparental) o dos progenitores. Y, por supuesto una persona puede no tener hijos.



4.4 Transformación de relaciones reflexivas

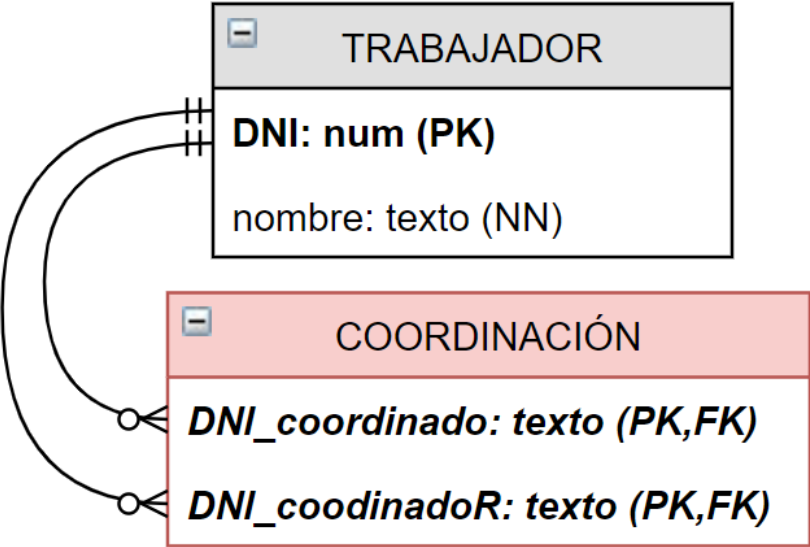
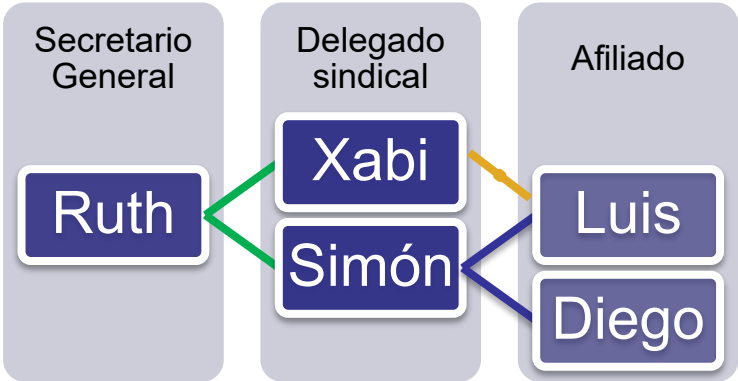
Ejemplo relación reflexiva M:N

Un trabajador puede coordinar a ninguno o a más compañeros y ser coordinado por 0 o más compañeros



4.4 Transformación de relaciones reflexivas

Ejemplo relación reflexiva M:N



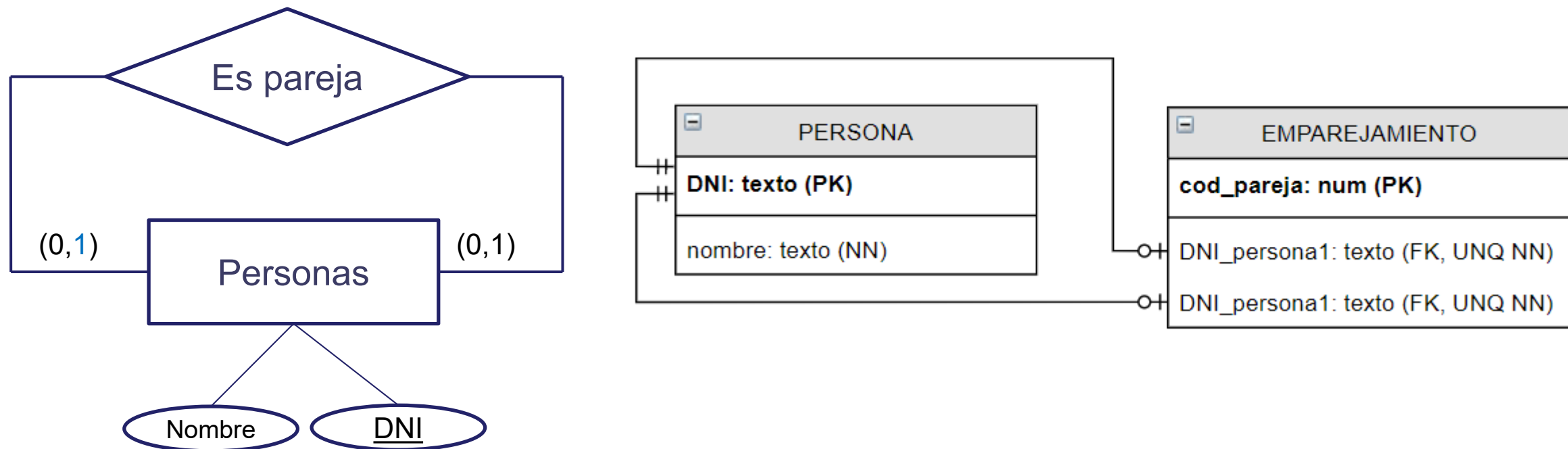
<u>DNI</u>	Nombre	Ape1
115C	Luis	Durruti
563F	Simón	Mera
X457	Ruth	Campoamor
159R	Diego	G. Oliver
478J	Xabi	Pestaña

<u>Código coordinadoR</u> PK y FK	<u>Código coodinadO</u> PK y FK
563F	115C
563F	159R
X457	563F
X457	478J
478J	115C

4.4 Transformación de relaciones reflexivas

Ejemplo relación reflexiva 1:1. Cardinalidades (0,1) y (0,1)

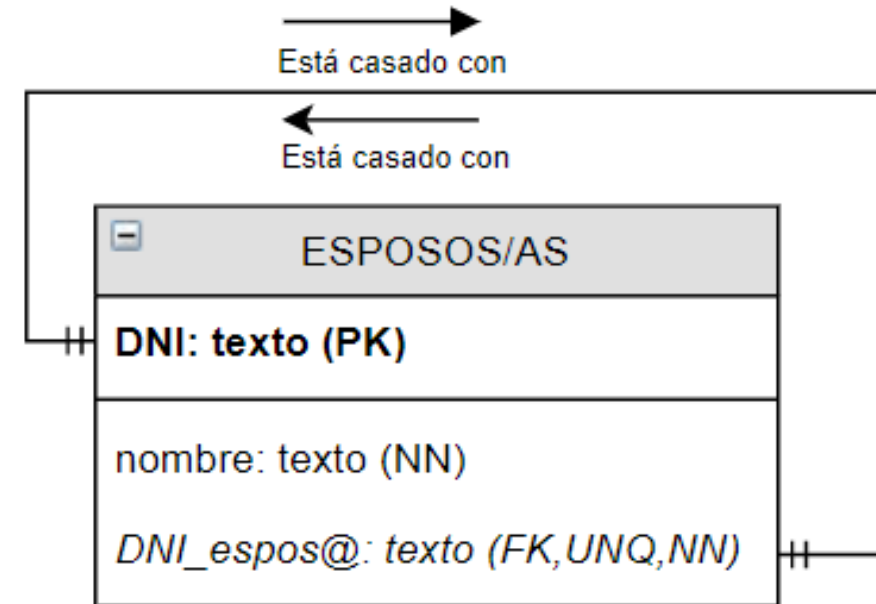
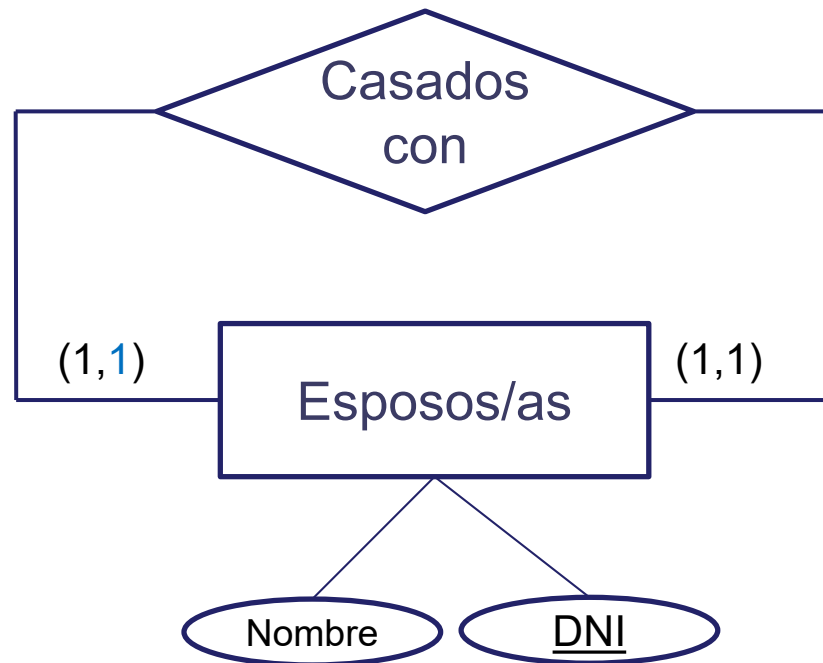
Una persona tiene como pareja a ninguna o a otra pareja (monogamia).



4.4 Transformación de relaciones reflexivas

Ejemplo relación reflexiva 1:1 Cardinalidades (1,1) y (1,1)

Un persona casada tiene que tener una y solo una esposa/o.

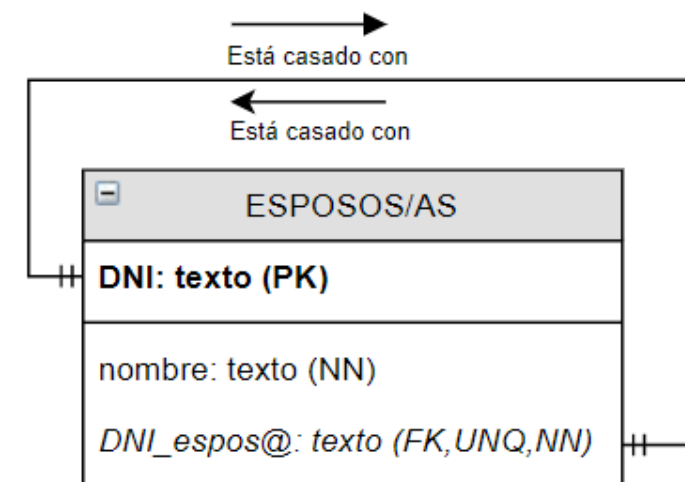


4.4 Transformación de relaciones reflexivas

Ejemplo relación reflexiva 1:1. Cardinalidades (1,1) y (1,1)

Un persona casada tiene que tener una y solo una esposa/o.

<u>DNI</u>	Nombre	Ape1	DNI_pareja FK UNQ NN
475G	Vanesa	Martínez	115C
115C	Andrés	Dorado	475G
159R	Sofía	Benito	X457
X457	Pablo	Garcés	159R



4.6 Transformación de jerarquías: Introducción

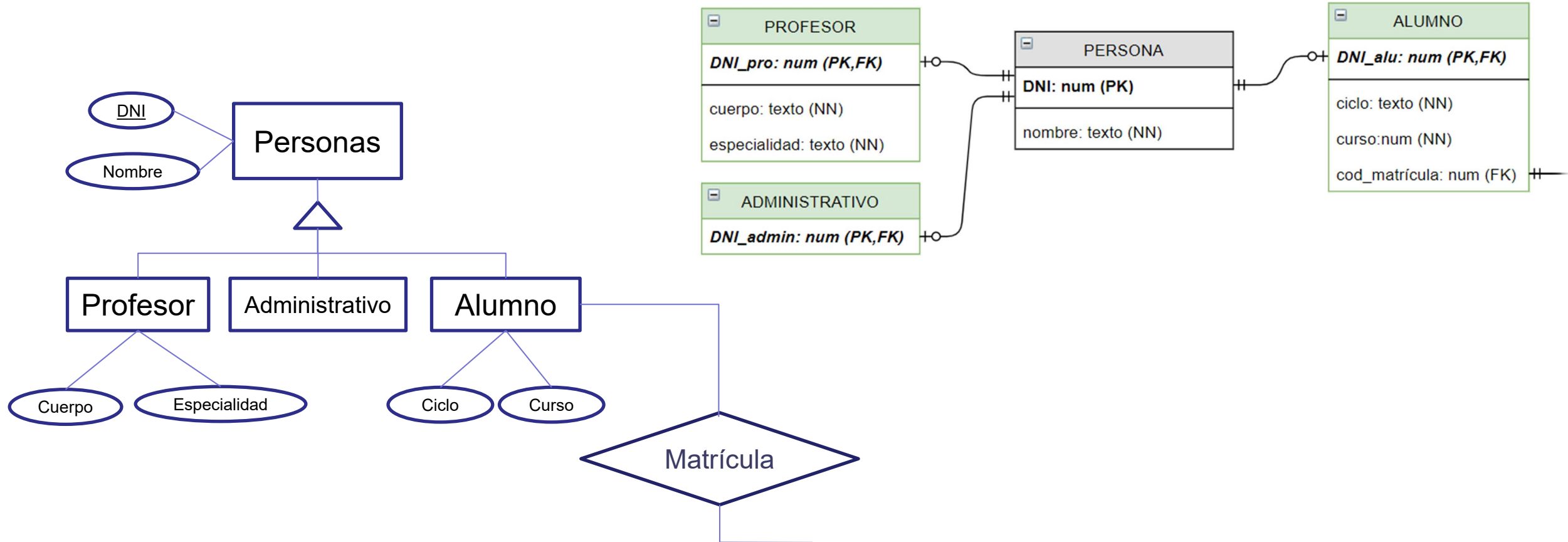
- El Modelo Relacional no dispone de instrumentos específicos que permitan representar tipos y subtipos.
- Existen distintos métodos de transformación, dependiendo de los objetivos perseguidos:
 - ✓ **Información semántica representada en el modelo (restricciones).**
 - ✓ **Eficiencia de acceso y almacenamiento de los datos.**
- Sólo cuando la jerarquía es parcial y solapada (sin restricciones) existe un método transformación en relaciones totalmente adecuada y eficiente.
- Nosotros usaremos este método y **dejaremos al modelo físico el control de restricciones** como la totalidad o la exclusividad.

4.6 Transformación de jerarquías: Método de transformación

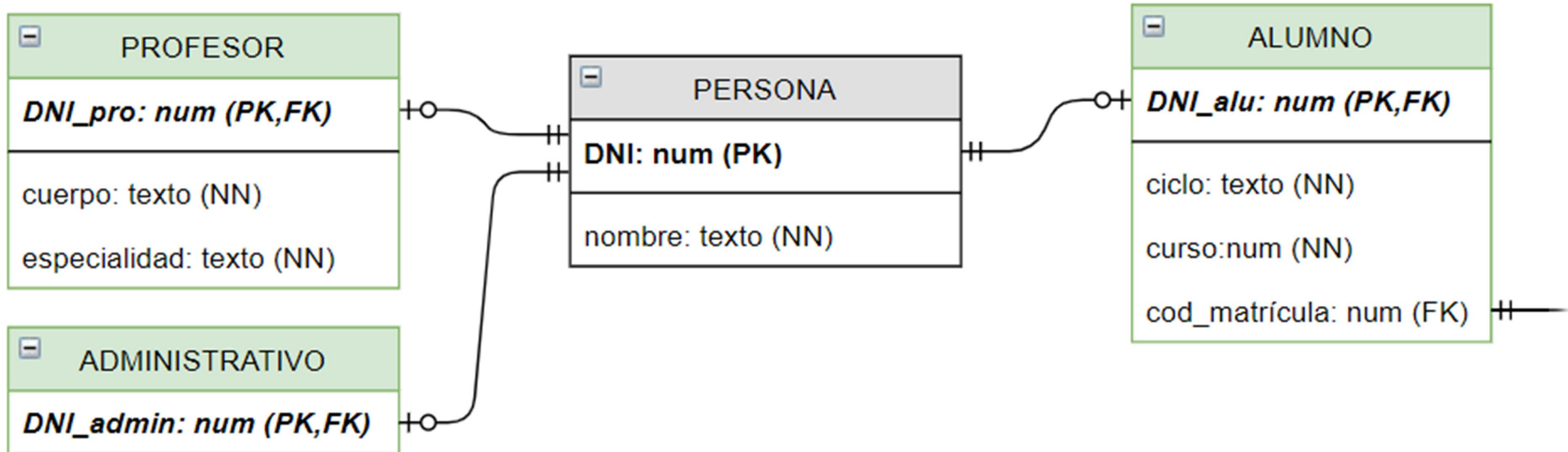
Crear **una tabla para la superentidad** con los atributos comunes y **una tabla por cada subentidad** con los atributos específicos de cada una (si los tuviera).

- ✓ Es un modelo que **por defecto es parcial y solapado** (sin restricciones).
- ✓ Por tanto, este modelo **no podrá implementar exclusividad ni totalidad explícitamente**.
- ✓ En los demás casos (totalidad y exclusividad) es necesario incluir ciertas restricciones como **CHECKS y TRIGGERS** (comprobaciones y disparadores) que corresponden al modelo físico.
- ✓ A pesar de lo anterior es el **modelo más eficiente** en almacenamiento y acceso a datos.

4.6 Transformación de jerarquías: Ejemplo



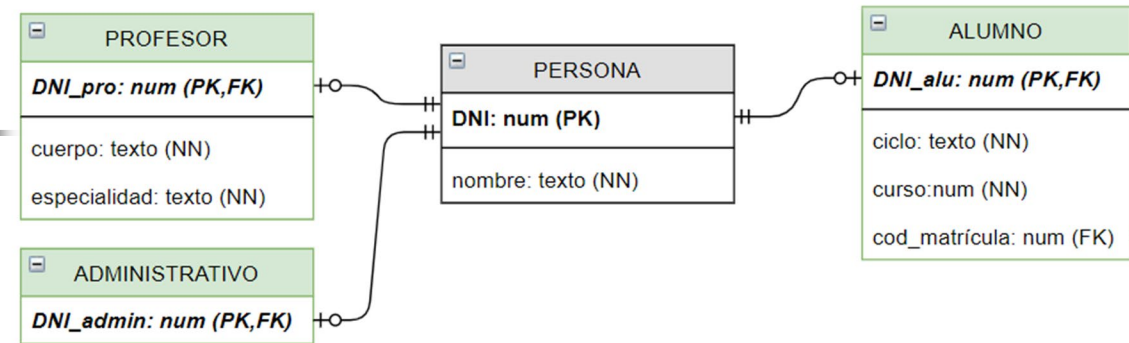
4.6 Transformación de jerarquías: Ejemplo



4.6 Transformación de jerarquías: Ejemplo

Recuerda Parcial y solapada por defecto.

- Luis es profesor y alumno.
- Simón es profesor.
- Ruth es alumna.
- Diego es administrativo.
- Vanesa no es ni alumna, ni administrativa, ni profesora.



<u>DNI</u>	Nombre	Ape1
115C	Luis	Durruti
563F	Simón	Mera
X457	Ruth	Campoamor
159R	Diego	G. Oliver
478F	Vanesa	Luxemburgo

<u>DNI Alumno</u> PK y FK	Ciclo	Curso	Cod Matrícula FK
115C	DAM	1º	001
X457	ASIR	2º	002

<u>DNI Profesor</u> PK y FK	Cuerpo	Especialidad
115C	Secundaria	Informática
563F	FP	SAI

<u>DNI Administrativo</u> PK y FK
159R

4.7 Normalización

- La **normalización** de bases de datos es un proceso que consiste **una serie de reglas a las tablas** obtenidas tras el paso del modelo entidad-relación al **modelo relacional**.
- Las bases de datos relacionales se normalizan para:
 - ✓ Minimizar la redundancia de los datos.
 - ✓ Disminuir problemas de actualización de los datos en las tablas.
 - ✓ Proteger la integridad de datos.

4.7 Normalización

Primera Forma Normal (1FN)

La primera regla de normalización se expresa generalmente en forma de dos indicaciones separadas.

1. Todos los atributos, valores almacenados en las columnas, **deben ser indivisibles**.
2. No deben existir **grupos de valores repetidos**.

4.7 Normalización

Primera Forma Normal (1FN)

1. Todos los atributos, valores almacenados en las columnas, **deben ser indivisibles**.

id	Nombre	Dirección	Teléfono	URL
1	Anaya	J:l: Luca	92199932	Anaya.com
2	Pericles	C/Luna # 20-28018 Tlaxcala	99299492	Pericles.com



Calle	Número	Puerta	CP	Población	Provincia
Luna	20		28018	Tlaxcala	Tlaxcala

Figura 1. Tabla con un atributo divisible en varias partes

4.7 Normalización

id	Nombre	calle	Número	Puerta	CP	Estado	Capital	Teléfono	URL
1	Anaya	Jl: Luca	15	2	28917	Tepic	Nayarit	93488345	Anaya.com
2	Pericles	Luna	20		28120	San Blas	Nayarit	88238188	Pericles.com
3	Mieres	Tajin	12	1	28120	San Blas	Nayarit	94989982	Mieres.es

Primera Forma Normal (1FN)

2. No deben existir **grupos de valores repetidos**.



Estado	CP	PK
CP	Estado	Capital
28917	Tepic	Nayarit
28120	San Blas	Nayarit
23009	Jean	Jaen

4.7 Normalización

Segunda Forma Normal (2FN)

- Una tabla está en 2FN si:
 1. La tabla está en la **primera forma normal (1NF)**.
 2. No existen **dependencias funcionales parciales**.
- ✓ Es decir, todos los valores de las columnas de una fila **deben depender de la clave primaria completa** de dicha fila.
- ✓ Las tablas en **1FN y con clave primaria simple ya cumplen la 2FN**.
- ✓ En caso de una claves primaria compuesta por dos columnas, se requeriría que todas las filas se identificaran por la **clave primaria completa y no por una sola columna** de la misma.

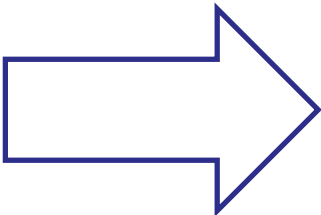
4.7 Normalización

Segunda Forma Normal (2FN)

HABILIDAD_EMP
Nombre: texto (PK)
Habilidad: texto (PK)
lugar_trabajo: texto (NN)

Habilidades de los empleados		
Empleado	Habilidad	Lugar actual de trabajo
Jones	Mecanografía	114 Main Street
Jones	Taquigrafía	114 Main Street
Jones	Tallado	114 Main Street
Bravo	Limpieza ligera	73 Industrial Way
Ellis	Alquimia	73 Industrial Way
Ellis	Malabarismo	73 Industrial Way
Harrison	Limpieza ligera	73 Industrial Way

EMPLEADO
Nombre: texto (PK)
lugar_trabajo: texto (NN)



HABILIDAD_EMP
Nombre: texto (PK)
lugar_trabajo: texto (NN)

Empleados	
Empleado	Lugar actual de trabajo
Jones	114 Main Street
Bravo	73 Industrial Way
Ellis	73 Industrial Way
Harrison	73 Industrial Way

Habilidades de los empleados	
Empleado	Habilidad
Jones	Mecanografía
Jones	Taquigrafía
Jones	Tallado
Bravo	Limpieza ligera
Ellis	Alquimia
Ellis	Malabarismo
Harrison	Limpieza ligera

4.7 Normalización

Tercera Forma Normal (3FN)

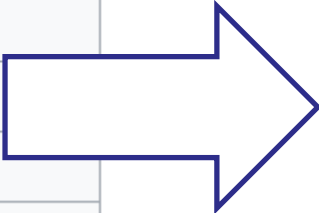
- Una tabla está en 3FN si:
 1. La tabla está en la **segunda forma normal** (2NF)
 2. Ningún atributo no-primario de la tabla es **dependiente transitivamente** de una clave primaria.
 - ✓ Es decir, no existen atributos que dependen de otro(s) atributos que no son la clave primaria.

4.7 Normalización

Tercera Forma Normal (3FN)

Ganadores del torneo

Torneo	Año	Ganador	Fecha de nacimiento del ganador
Indiana Invitational	1998	Al Fredrickson	21 de julio de 1975
Cleveland Open	1999	Bob Albertson	28 de septiembre de 1968
Des Moines Masters	1999	Al Fredrickson	21 de julio de 1975
Indiana Invitational	1999	Chip Masterson	14 de marzo de 1977



Ganadores del torneo

Torneo	Año	Ganador
Indiana Invitational	1998	Al Fredrickson
Cleveland Open	1999	Bob Albertson
Des Moines Masters	1999	Al Fredrickson
Indiana Invitational	1999	Chip Masterson

Fecha de nacimiento del jugador

Ganador	Fecha de nacimiento
Chip Masterson	14 de marzo de 1977
Al Fredrickson	21 de julio de 1975
Bob Albertson	28 de septiembre de 1968

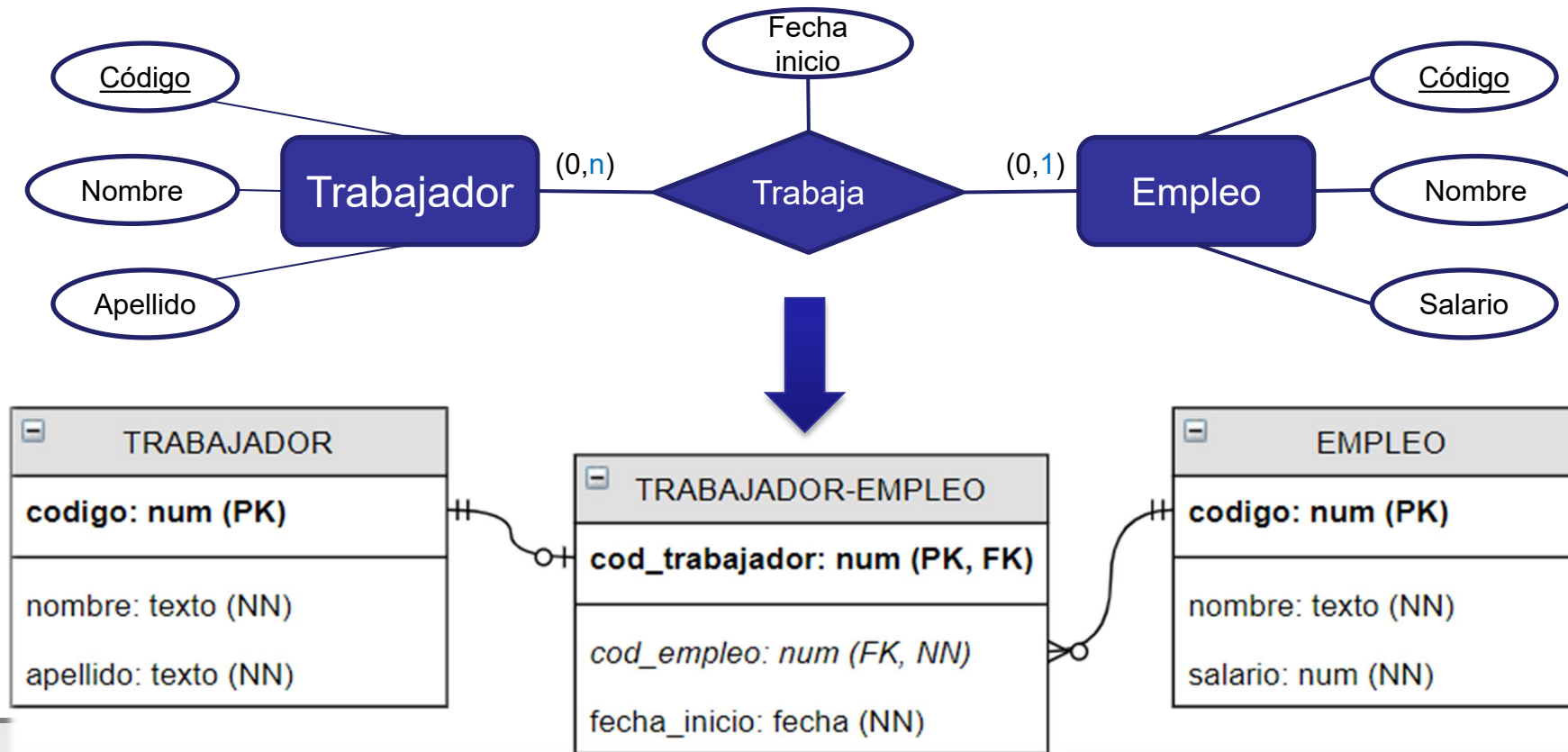
Ampliación 5.1: Transformaciones de relaciones uno a varios CON NUEVA TABLA

Relaciones UNO A VARIOS con nueva tabla: Método de transformación

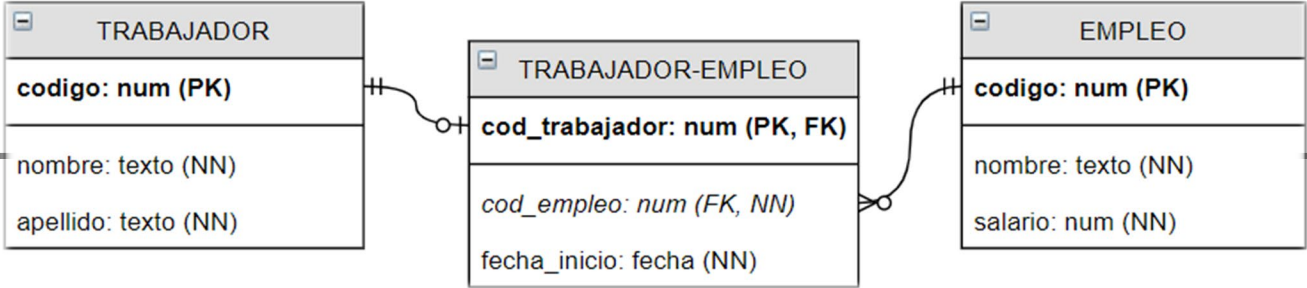
- En este caso **SÍ se crea ninguna tabla nueva.**
- Es decir, creamos una nueva tabla cuya **clave primaria será la clave ajena que apunta a la tabla-entidad que interviene en la con cardinalidad N.**
- A diferencia de la nueva tabla en las relaciones M:N (donde como clave primaria se usa la combinación de ambas claves ajenas), **en las relaciones 1:N se usa un solo campo** (el que apunta a la tabla del lado N) **como clave primaria.**

Ampliación 5.1: Transformaciones de relaciones uno a varios CON NUEVA TABLA

Relaciones con sólo una entidad con cardinalidad N: UNO A VARIOS



Ampliación 5.1: Transformaciones de relaciones uno a varios CON NUEVA TABLA



Código	Nombre	Ape1
T001	Andrés	Dorado
T002	Sofía	Benito
T003	Pablo	Garcés
T004	Julia	Delgado
T005	Inés	Moreno

Cód trabajador PK y FK	Cód_empleo FK	Fecha Inicio
T001	E001	5/5/21
T002	E001	3/3/20
T002	E002	4/4/19

Código	Nombre	Salario
E001	Barrendera	10
E002	Médico	5
E003	Ingeniera	30

✓ La clave primaria simple (Cod_trabajador) no incluye Cód_empleo y garantiza que no puede haber un mismo trabajador con dos empleos pero sí varios trabajadores con el mismo empleo.

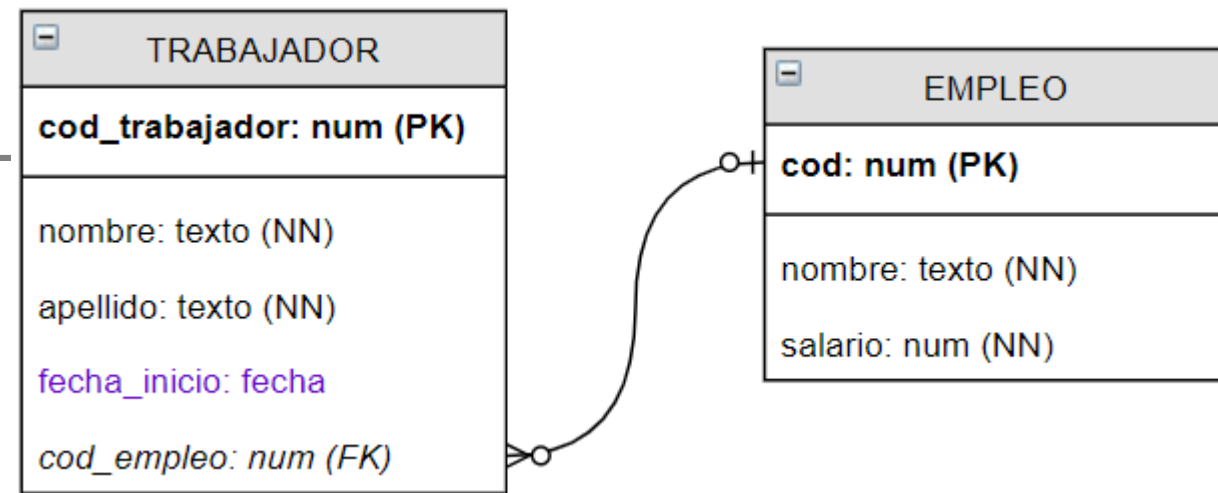
Ampliación 5.1: Transformaciones de relaciones uno a varios CON NUEVA TABLA

Relaciones UNO A VARIOS con nueva tabla: Casos de aplicación

1. Tenemos **atributos de relación** a pesar de ser 1:N.
2. La clave ajena de la tabla del lado muchos **tiene muchos registros con el valor NULL**.
3. Se prevé que la interrelación **podría ser N:M en el futuro**.

Ampliación 5.1: Transformaciones de relaciones uno a varios CON NUEVA TABLA

Problemas de aplicar el sistema ordinario con muchos nulos y atributos de relación



<u>Código</u>	Nombre	Ape1	Fecha Inicio	Cód_Empleo
T001	Andrés	Dorado	5/5/21	E001
T002	Sofía	Benito	3/3/20	E001
T003	Pablo	Garcés		
T004	Julia	Delgado		
T005	Inés	Moreno		

<u>Código</u>	Nombre	Salario
E001	Barrendera	10
E002	Médico	5
E003	Ingeniera	30

- ✓ Nos obliga a trasladar el atributo de relación a la tabla de la clave ajena (fallo de semántica).
- ✓ Si tenemos muchos trabajadores sin empleo perdemos mucho espacio (fecha y Cod_Empleo).

Ampliación 5.2: Otros métodos de transformación de jerarquías

Crear **una sola tabla** que contendrá los **atributos comunes** de la superentidad y los **atributos específicos** de las subentidades. Usaremos un **atributo discriminador** que indique la pertenencia de cada ocurrencia a una subentidad (o a ninguna).

Este modelo es útil cuando:

1. **Alguna de las subentidades no tenga atributos.**
 2. Queremos **controlar la totalidad y la exclusividad.**
- ✓ Es un **modelo poco eficiente** ya que la tabla principal casi siempre tiene atributos sin valor.

Ampliación 5.2: Otros métodos de transformación de jerarquías

Este método tendrá dos aproximaciones:

a. Totalidad/Parcialidad

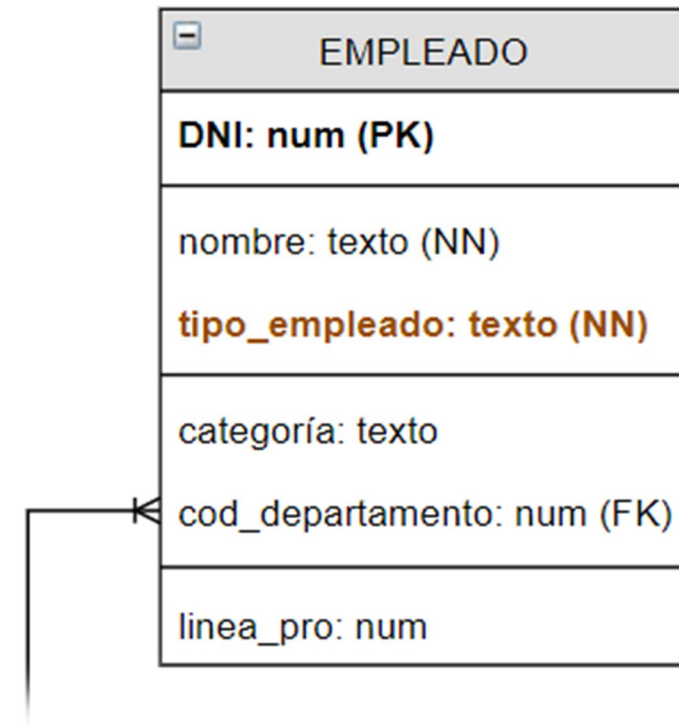
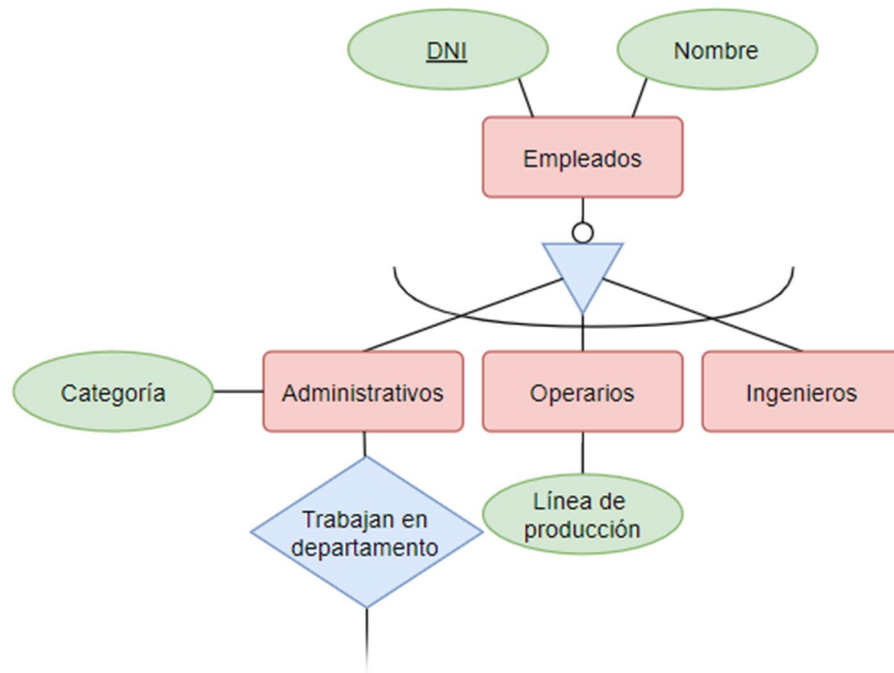
- La **totalidad** se implementaría **impidiendo el valor “ninguno”** en el atributo “tipo”.
- La **parcialidad** se implementaría **admitiendo el valor “ninguno”** en el atributo “tipo”.

b. Exclusividad/Solapamiento

- La **exclusividad** se implementaría con un atributo discriminador **univaluado**.
- El solapamiento se implementaría con un atributo discriminador **multivaluado**.
 - ✓ Y exigiendo que todas las ocurrencias de la tabla principal tengan su correspondencia en el atributo multivaluado.

Ampliación 5.2: Otros métodos de transformación de jerarquías. Ejemplo

Exclusividad



Ampliación 5.2: Otros métodos de transformación de jerarquías

Exclusividad

- ✓ Si no se permite “ninguno” la jerarquía será total.
 - Luis es administrativo. Simón es operario.
 - Ruth es ingeniera. Diego no está especializado en ninguna entidad.

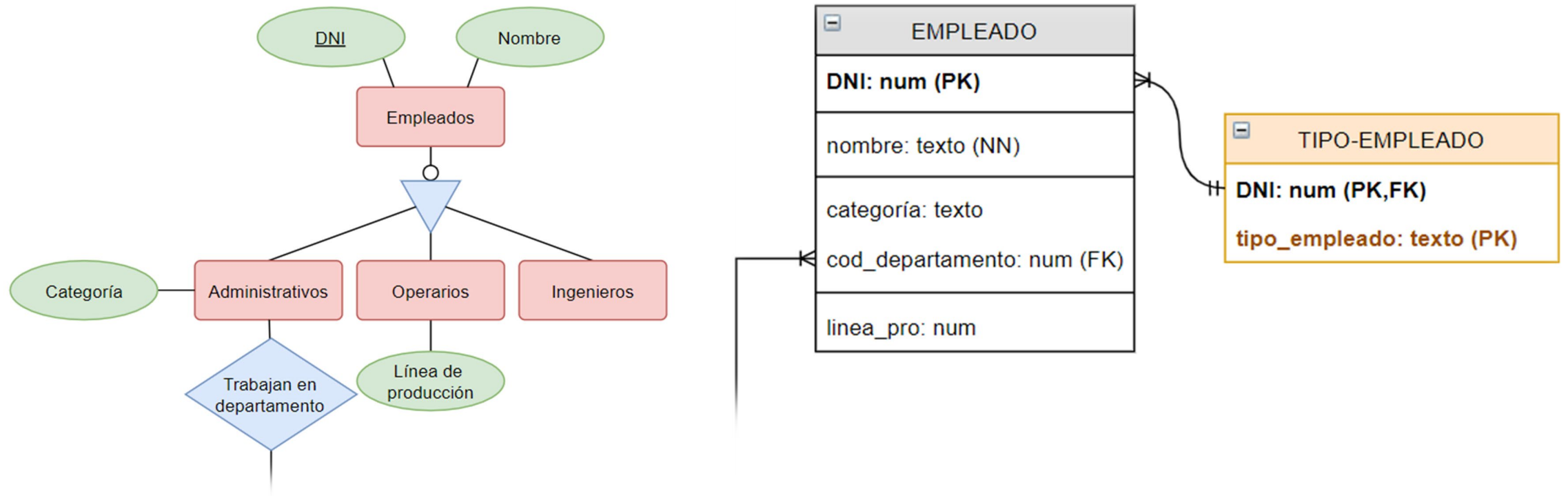
<u>DNI</u>	Nombre	Tipo_emp	Categoría	Cod_dep FK	Línea Prod
115C	Luis	Administrativo	Contable	Dep20	
563F	Simón	Operario			10
X457	Ruth	Ingeniera			
159R	Diego	Ninguno			



- ✓ Observad como, si no se permite el valor “Ninguno”, todos los empleados deben estar especializados.
- ✓ Observad todo el espacio que se pierde con los empleados no especializados o especializados sin atributos.

Ampliación 5.2: Otros métodos de transformación de jerarquías

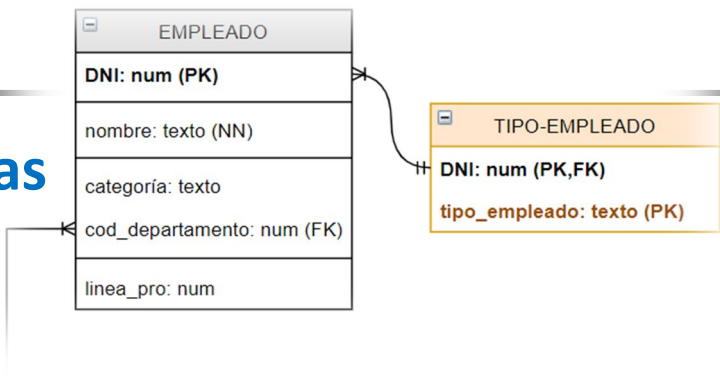
Solapamiento



Ampliación 5.2: Otros métodos de transformación de jerarquías

Solapamiento

- ✓ Si no se permite “ninguno” la jerarquía será total.
 - Luis es administrativo e ingeniero. Simón es operario.
 - Ruth es ingeniera. Diego no está especializado en ninguna entidad.



<u>DNI</u>	Nombre	Categoría	Cod_dep FK	Línea Prod	<u>DNI</u> PK y FK	<u>Tipo emp</u> PK
115C	Luis	Contable	Dep20		115C	Administrativo
563F	Simón			10	115C	Ingeniero
X457	Ruth				563F	Operario
159R	Diego				X457	Ingeniera
					159R	Ninguno