2.7. Relaciones de herencia (relaciones ISA)

Este tipo de relaciones las encontramos cuando necesitamos unificar entidades agrupándolas en una principal (generalización) o dividir una general en otras más específicas (especialización). Hoy en día se suele llamar a ambas **relaciones de generalización o relaciones de herencia.**

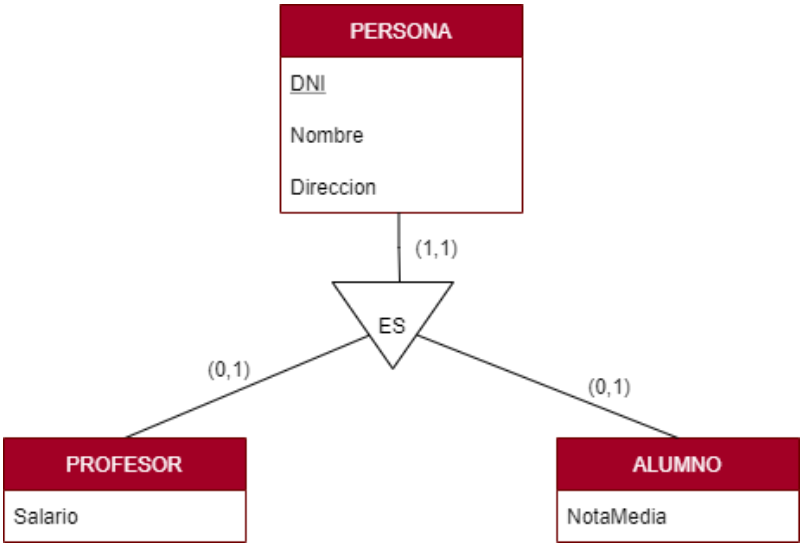
A veces podemos encontrarnos con entidades que son muy parecidas y se diferencian en pocos atributos.

Ejemplo

Queremos guardar información sobre las personas de un instituto.
Para todas las personas queremos guardar un identificador, un nombre y una dirección. Pero dependiendo de quién sea querremos guardar otra información:

- De los profesores queremos guardar su salario
- De los alumnos queremos guardar su nota media

En este caso deberíamos hacer una jerarquía con las entidades que intervienen:

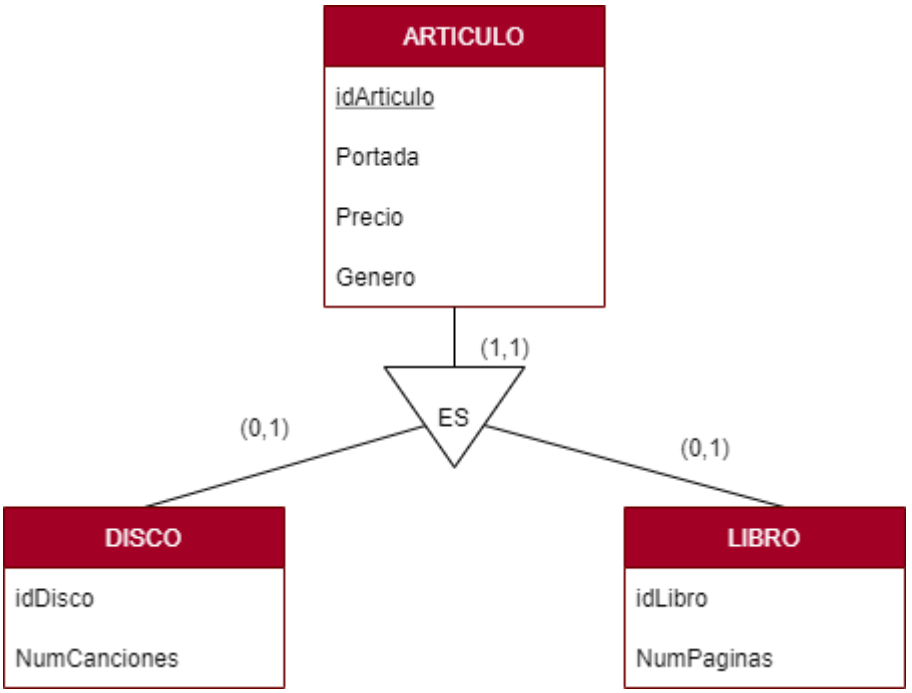


Ejemplo

Veamos ahora el caso de una tienda que vende discos y libros.

- De los discos queremos almacenar un identificador, portada, precio, número de canciones y género.
- De los libros queremos guardar un identificador, portada, precio, número de páginas y género.

Si observamos los atributos de las dos entidades, nos damos cuenta de que tenemos atributos muy similares y del mismo tipo, esto nos puede llevar a generalizar en una entidad que recoja los atributos comunes:



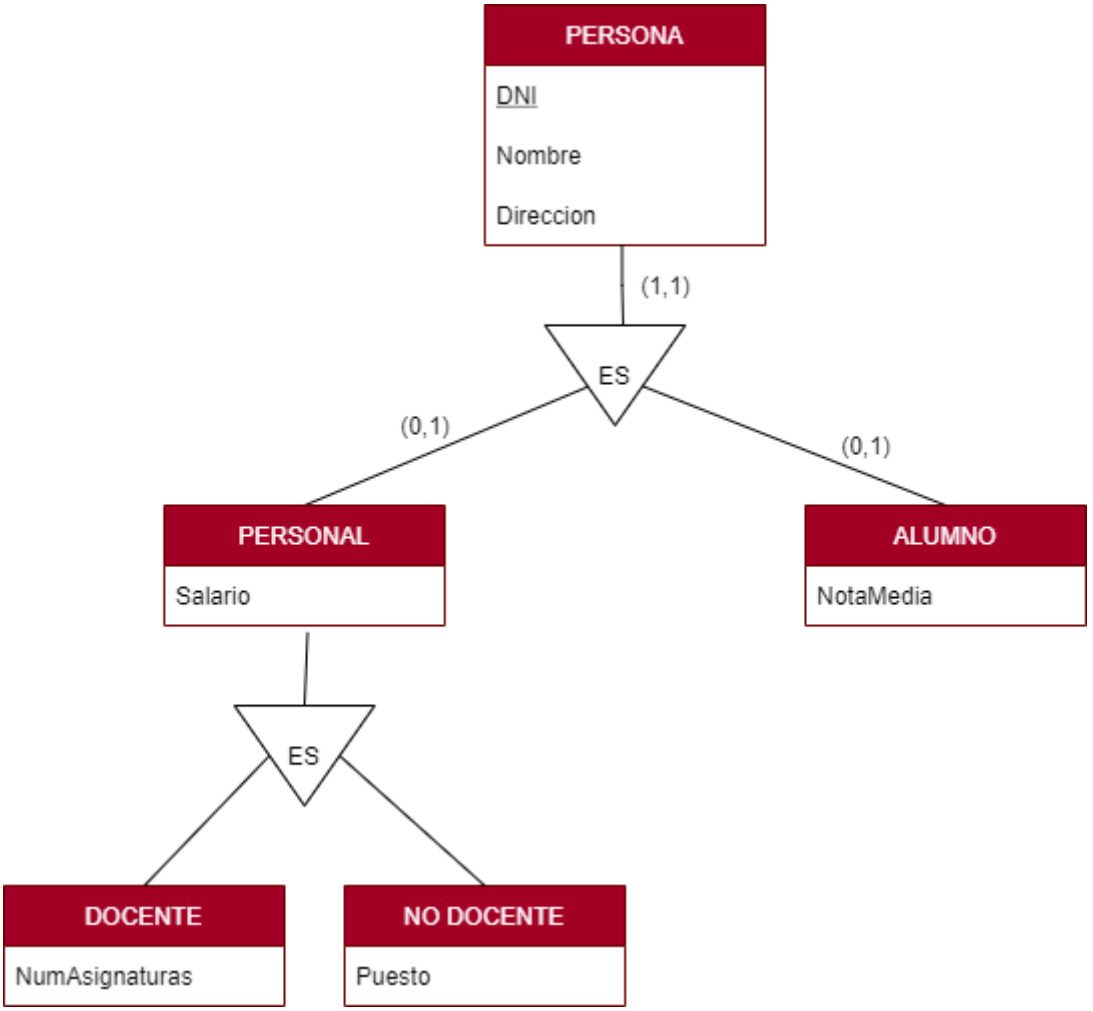
En conclusión podemos decir que las **relaciones tipo ISA**:

ISA procede del término inglés ***is a =es un/a***
Especifica relaciones en base a un tipo:

- Un alumno es una persona
- Un profesor es una persona
- Un disco es un artículo
- Un libro es un artículo

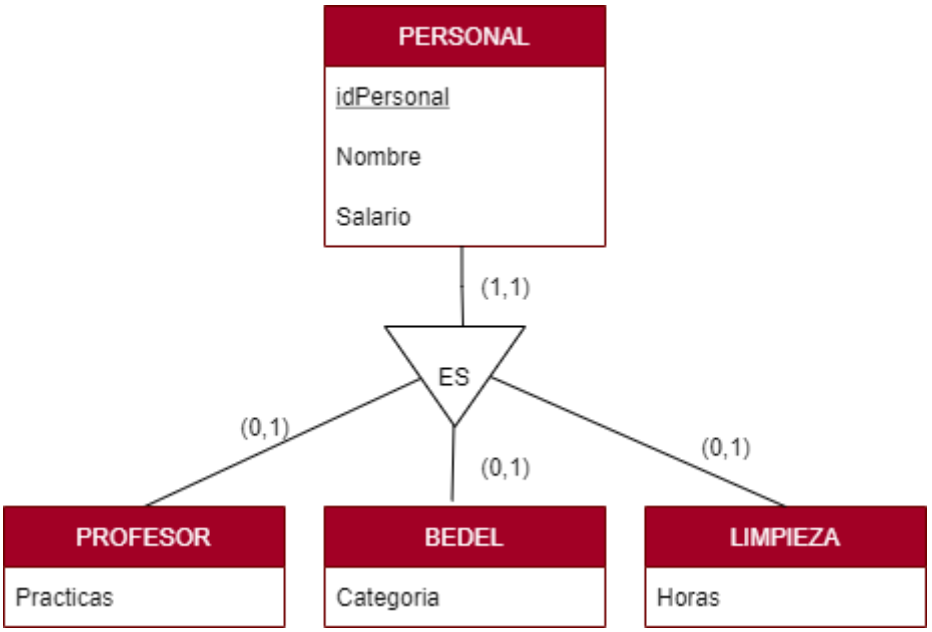
Es decir, los alumnos y profesores son tipos de PERSONA, y los discos y libros son tipo de ARTÍCULO.
La relación de jerarquía la representamos con un triángulo invertido ▽
A la entidad principal *PERSONA o ARTICULO* la llamamos ***superentidad*** o ***supertipo***.
A los diferentes ramales *PROFESOR, ALUMNO* son ***subentidad*** o ***subtipo de*** *PERSONA y, DISCO, LIBRO* son subentidad o subtipo de *ARTICULO.*
Las subentidades **heredan** los atributos de las superentidades, es decir, los DISCOS y LIBROS tienen los atributos de los ARTICULOS

Debemos señalar que puede haber subentidades que se comporten como superentidades de otras entidades, por ejemplo:



Las subentidades *DOCENTE* y *NO DOCENTE* heredan los atributos de *PERSONAL* y de *PERSONA.*

Las relaciones de tipo ISA también tienen cardinalidad y suele ser:



- Si vamos de *PROFESOR* a *PERSONAL*, la cardinalidad es (1,1), eso significa que un *PROFESOR* será sí o sí un miembro de *PERSONAL* (como mínimo y como máximo)
- Si vamos de *PERSONAL* a *PROFESOR*, la cardinalidad es (0,1), lo que significa que cada miembro del personal puede ser o no ser un *PROFESOR*.
- Estas cardinalidades se dan por hechas y muchas veces no se representan y por tanto no aparecen en los diagramas, pero se sobreentiende.

2.7.1. El Identificador

Otra cuestión en el tipo de relaciones ISA es el **Identificador**.

Si la superentidad tiene identificador, las subentidades también lo heredan, en el ejemplo anterior, como *PERSONAL* tiene el identificador *idPersonal, PROFESOR, BEDEL y LIMPIEZA* también tendrán el identificador *idPersonal*, en este caso la cardinalidad mínima de la superentidad será 1 y hablaremos de ***Especialización***

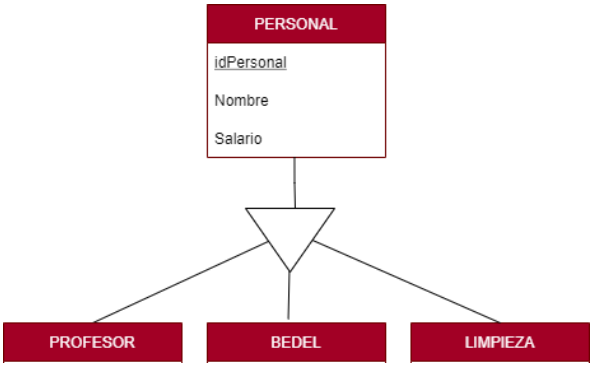
Esto no impide que por ejemplo PROFESOR tenga su propio identificador CodProfesor, que servirá para identificarle como *PROFESOR* aunque no para distinguirle de *BEDEL o LIMPIEZA.* En este caso la cardinalidad mínima de la superentidad será 0 y hablaremos de **Generalización**

2.7.2. ¿Cuándo utilizar relaciones tipo ISA?

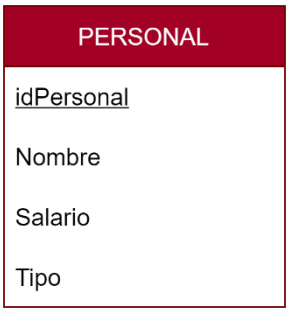
Cuando conocemos el concepto de relaciones ISA, tendemos a utilizarlas en exceso, pero debemos ver si en nuestro caso es o no necesario.

Ejemplo

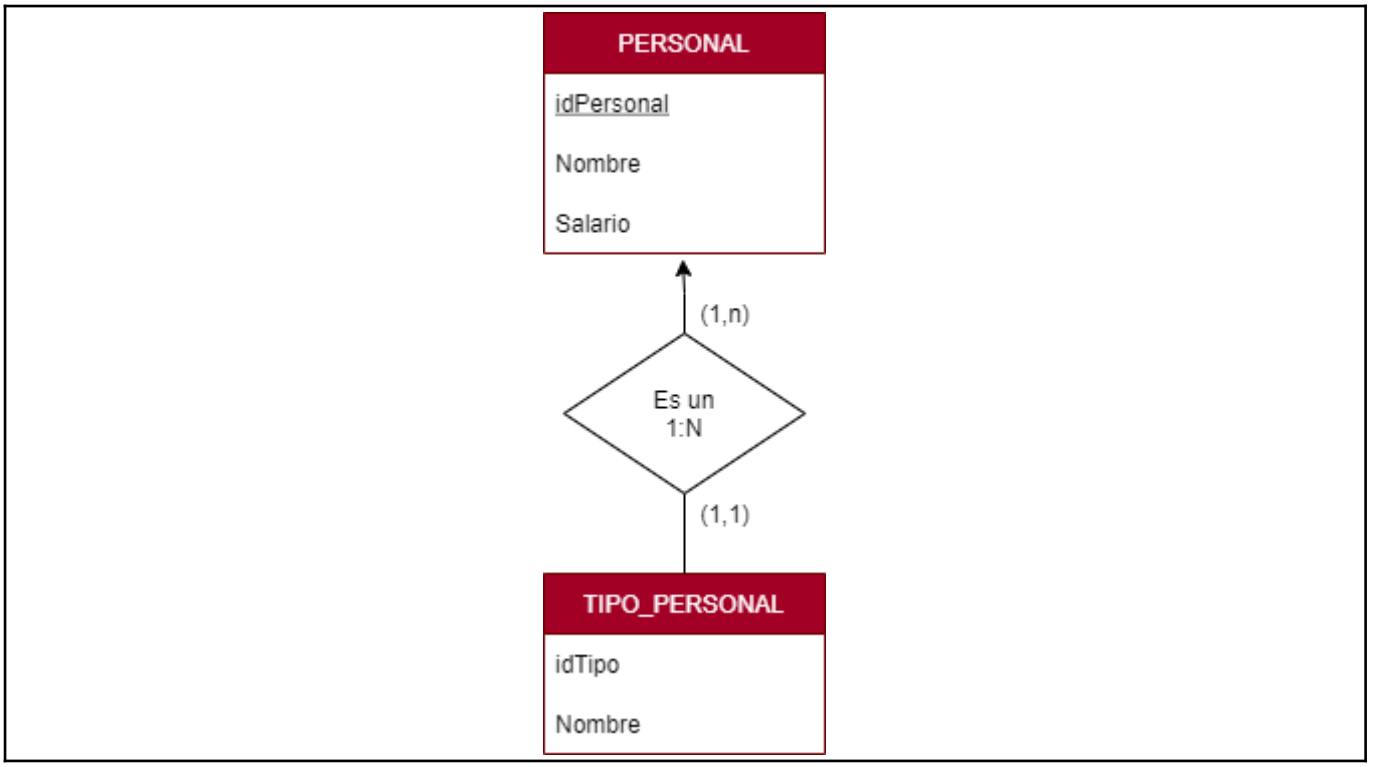
Queremos guardar un identificador, el nombre y el salario del Personal que trabaja en un instituto, teniendo en cuenta que en el instituto trabajan Profesores, bedeles y personal de limpieza.
Lo primero que pensamos es en un diagrama del tipo:



Pero no tenemos ningún atributo que nos distinga estas entidades, en este caso sería mejor crear un atributo en PERSONAL que marque el *Tipo* de *PERSONAL*:



Aunque una mejor solución sería:



En definitiva, debemos usar relaciones ISA cuando se cumpla alguna de las siguientes condiciones:

- Si las subentidades tienen atributos distintos
- Si las subentidades tienen relaciones distintas (aunque pueden no tener distintos atributos), por ejemplo los *PROFESORES* dan clase a los alumnos, y son los únicos miembros del personal que dan clase.

2.7.3. Tipos de relaciones en ISA

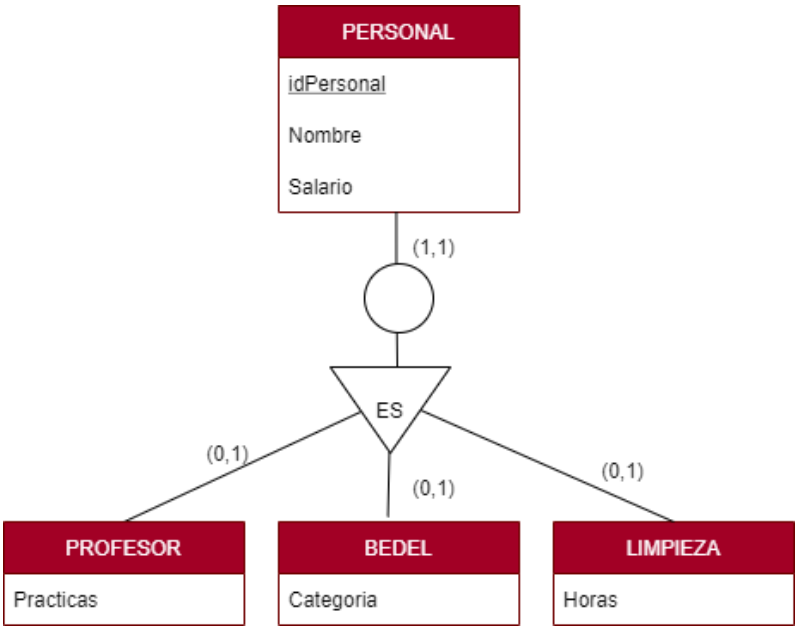
2.7.3.1. Obligatoriedad

Indica si los ejemplares de la superentidad se relacionan obligatoriamente con ejemplares de sus subentidades. Dentro de esta distinción podemos tener:

- Relaciones de Jerarquía total
- Relaciones de Jerarquía parcial

Relaciones de jerarquía total

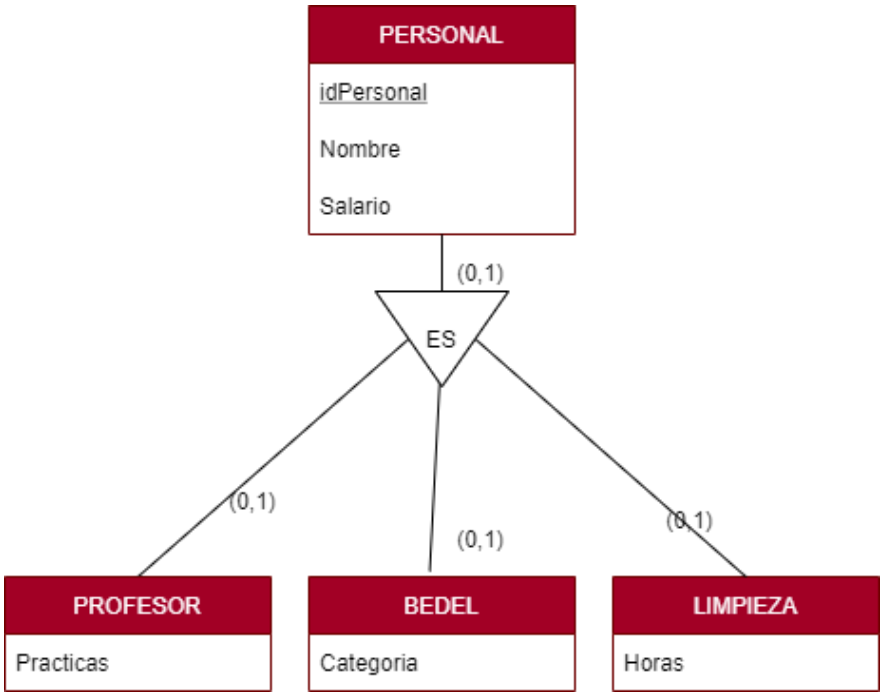
Indica que todos los ejemplares de la superentidad se relacionan sí o sí con alguna subentidad, es decir, en el ejemplo anterior, no pueden existir personas que no sean ni profesor ni bedel ni limpieza, siempre tiene que estar relacionados con alguno de ellos. Se representa con la cardinalidad mínima a 1 en la superentidad y con un círculo en la línea donde une la superentidad con la relación:



Relación de jerarquía total

Relaciones de jerarquía parcial

Indica que hay ejemplares de la superentidad que no se relacionan con ninguna subentidad, en el ejemplo anterior, diríamos que puede haber ejemplares de Personal que no sean Profesores, ni bedeles ni de limpieza. Se representa con cardinalidad mínima a 0 en la superentidad:



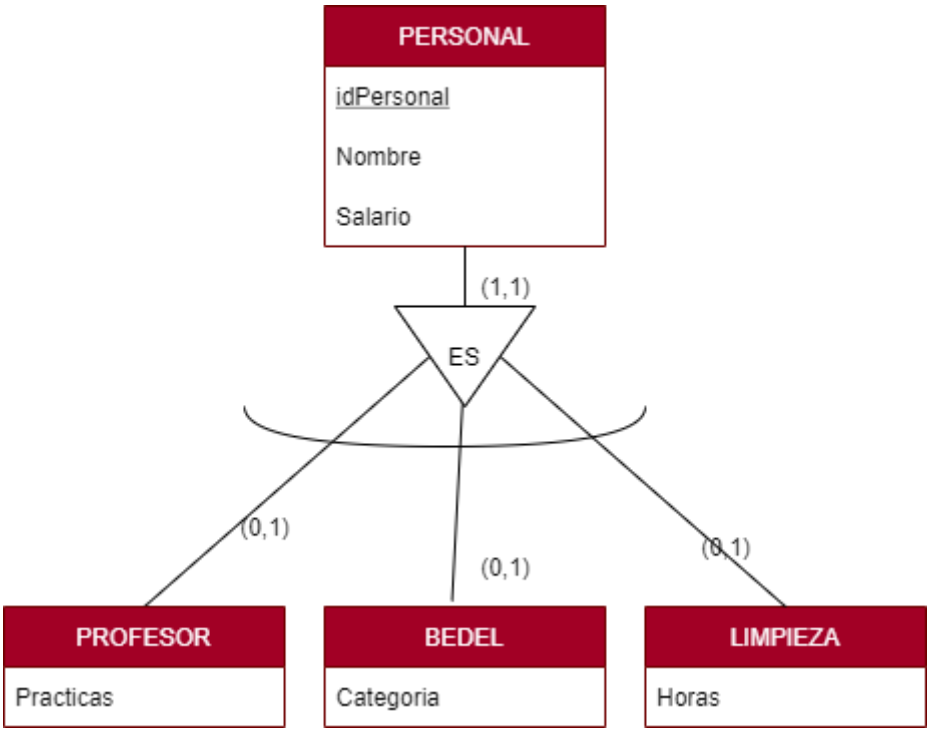
Relación de jerarquía parcial

2.7.3.2. Número de relaciones

Mide con cuántas subentidades se relaciona la superentidad, es decir, si hay personal que pueda ser profesor y bedel a la vez o si sólo puede ser una cosa.

Relaciones de jerarquía exclusiva

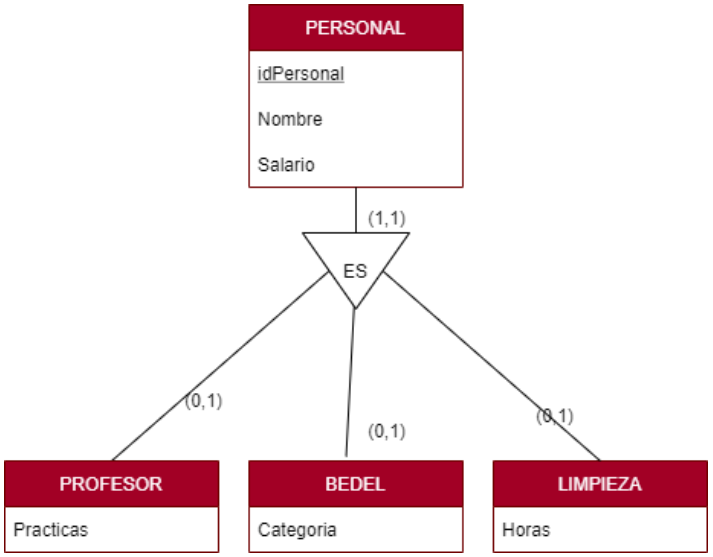
Ocurre cuando un ejemplar de la superentidad sólo puede estar relacionado exclusivamente con una de las subentidades, es decir, cuando un ejemplar de Personal sólo puede ser Profesor o Bedel o de Limpieza. Se representa con un arco debajo del triángulo invertido.



Relación de jerarquía exclusiva

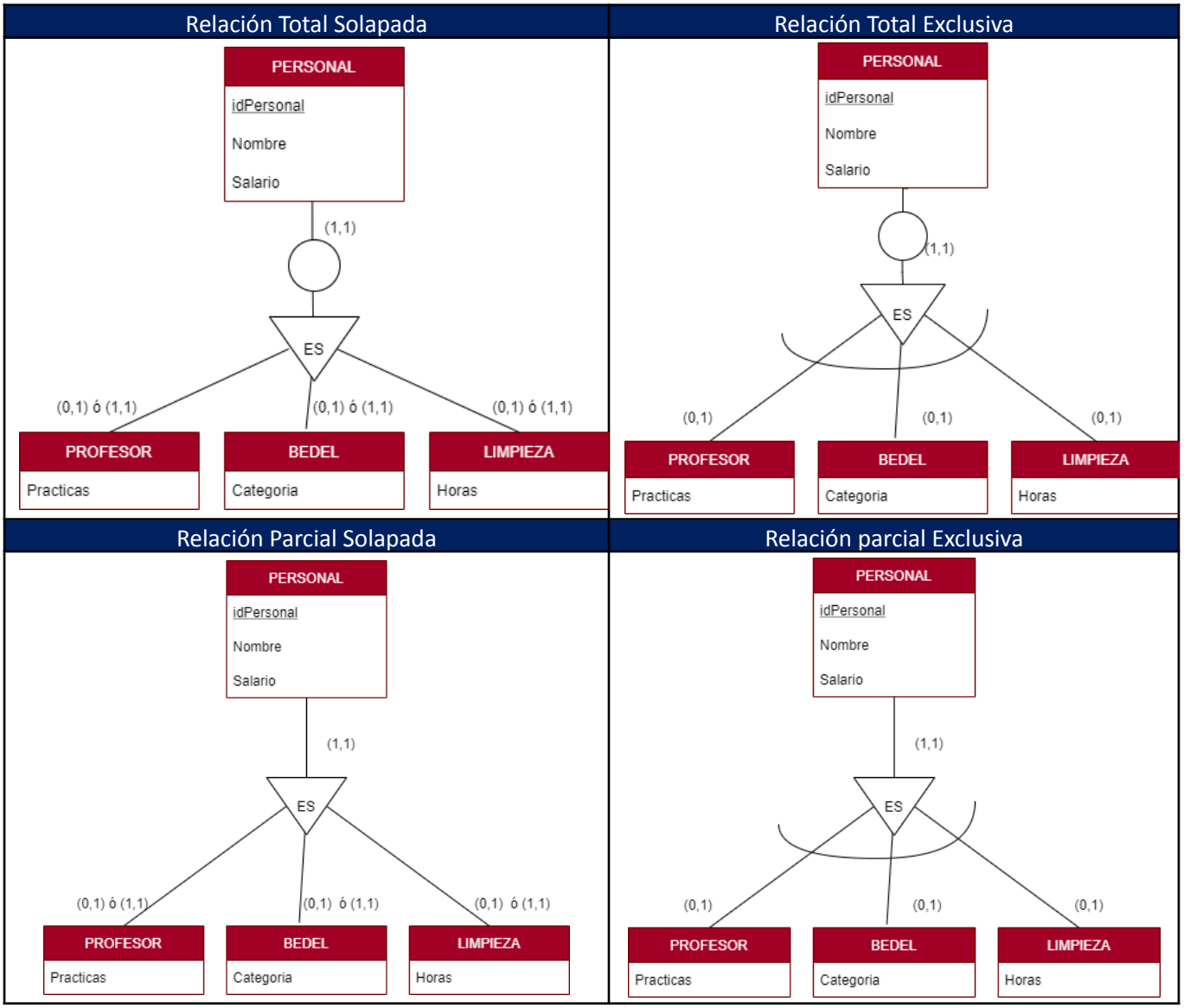
Relaciones de jerarquía solapada

Esto sucede cuando un ejemplar de la superentidad puede estar relacionado con más de un ejemplar de las subentidades, por ejemplo una persona que pueda ser Bedel y de Limpieza a la vez. Obviamente esto pasa en relaciones que no son exclusivas y son totales, ya que obligamos a que esté relacionada con alguna de ellas.



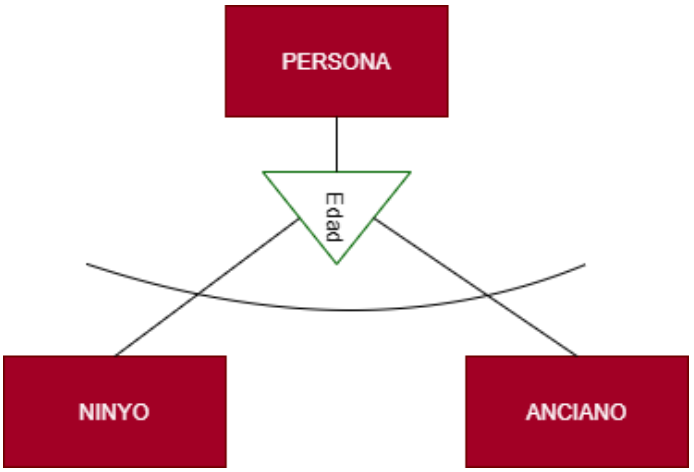
Relación de jerarquía solapada

En conclusión podemos tener las siguientes 4 relaciones ISA:



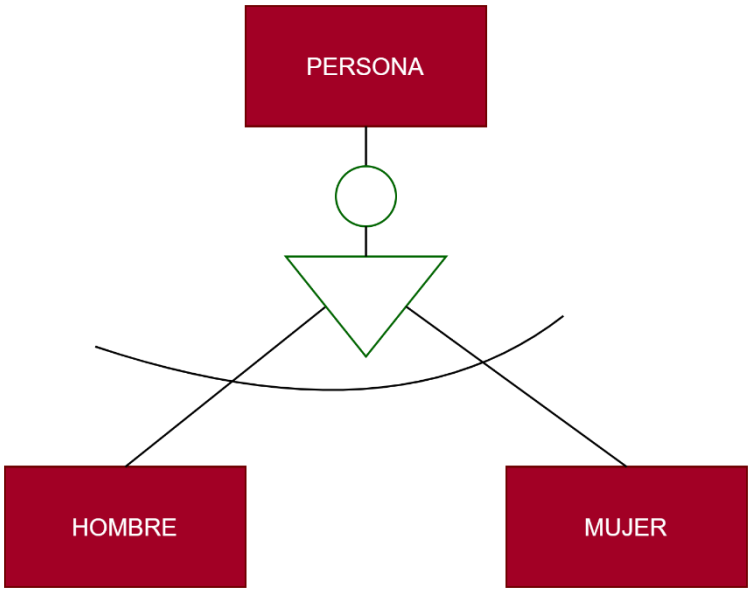
Más ejemplos

Jerarquía parcial exclusiva



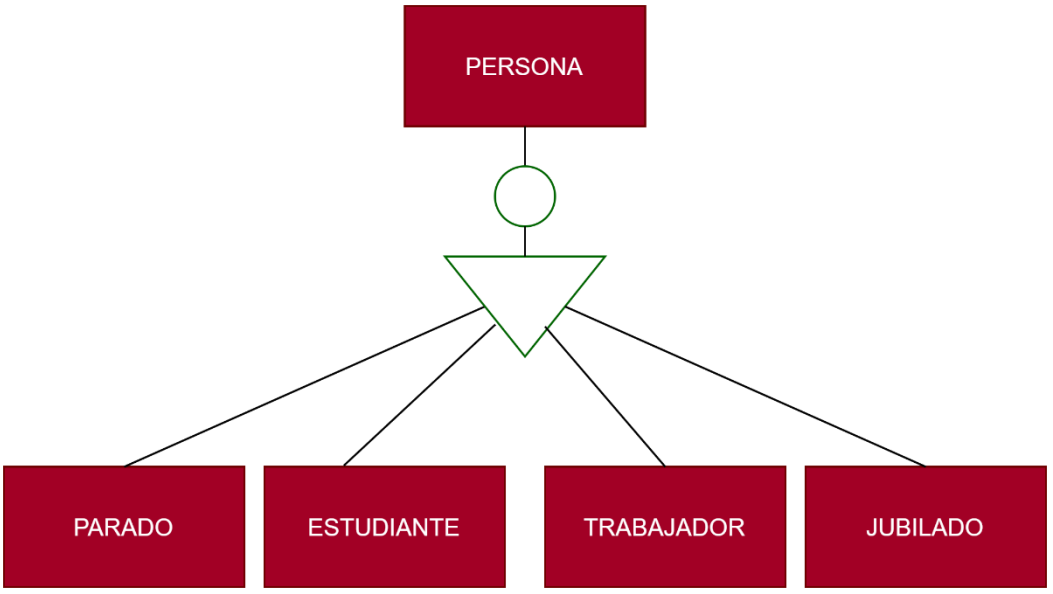
Una persona puede ser adulta sin ser niño o anciano, pero si es niño no puede ser anciano

Jerarquía total exclusiva



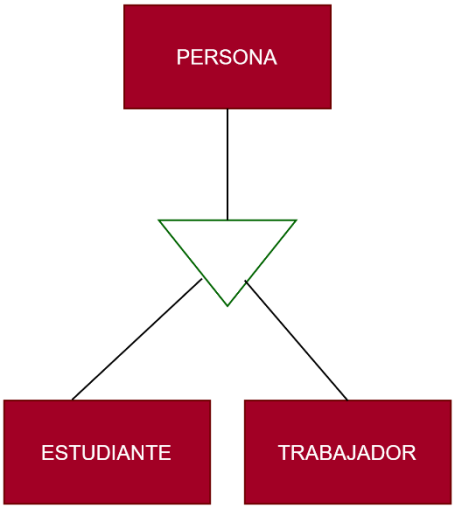
Cualquier persona sólo puede ser hombre o mujer

Jerarquía total solapada



Según su situación laboral, una persona sólo puede ser Parado, estudiante, trabajador o jubilado. Pero también puede ser estudiante y trabajador a la vez.

Jerarquía parcial solapada

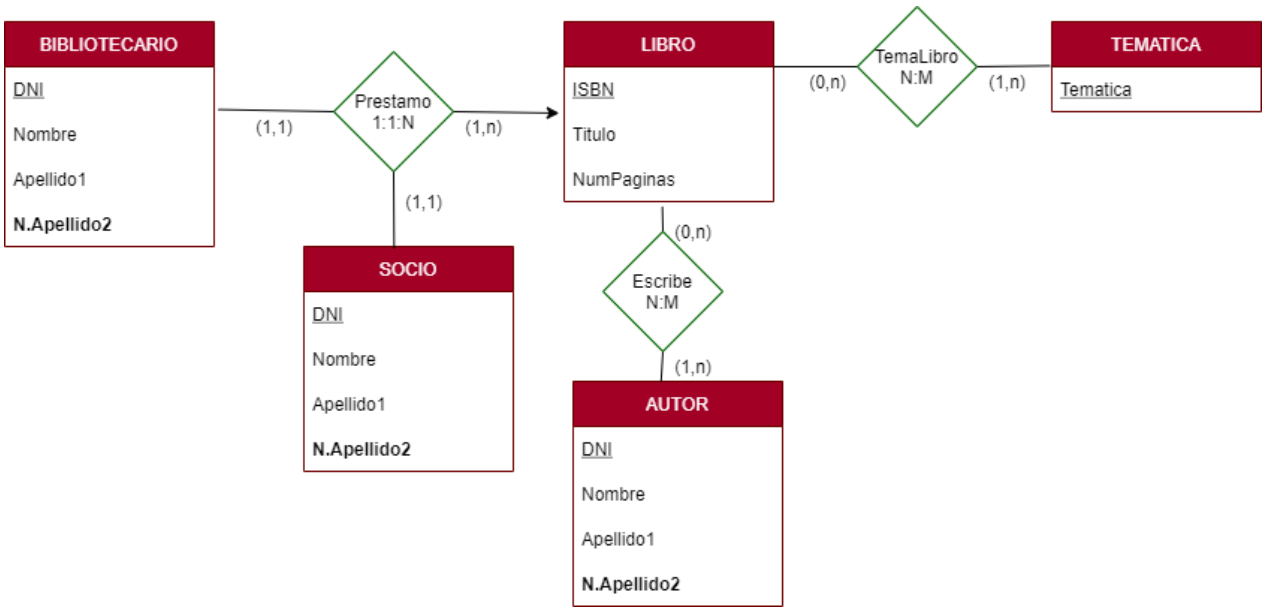


Según su situación laboral una persona puede ser estudiante y trabajador a la vez, pero también puede ser Jubilado

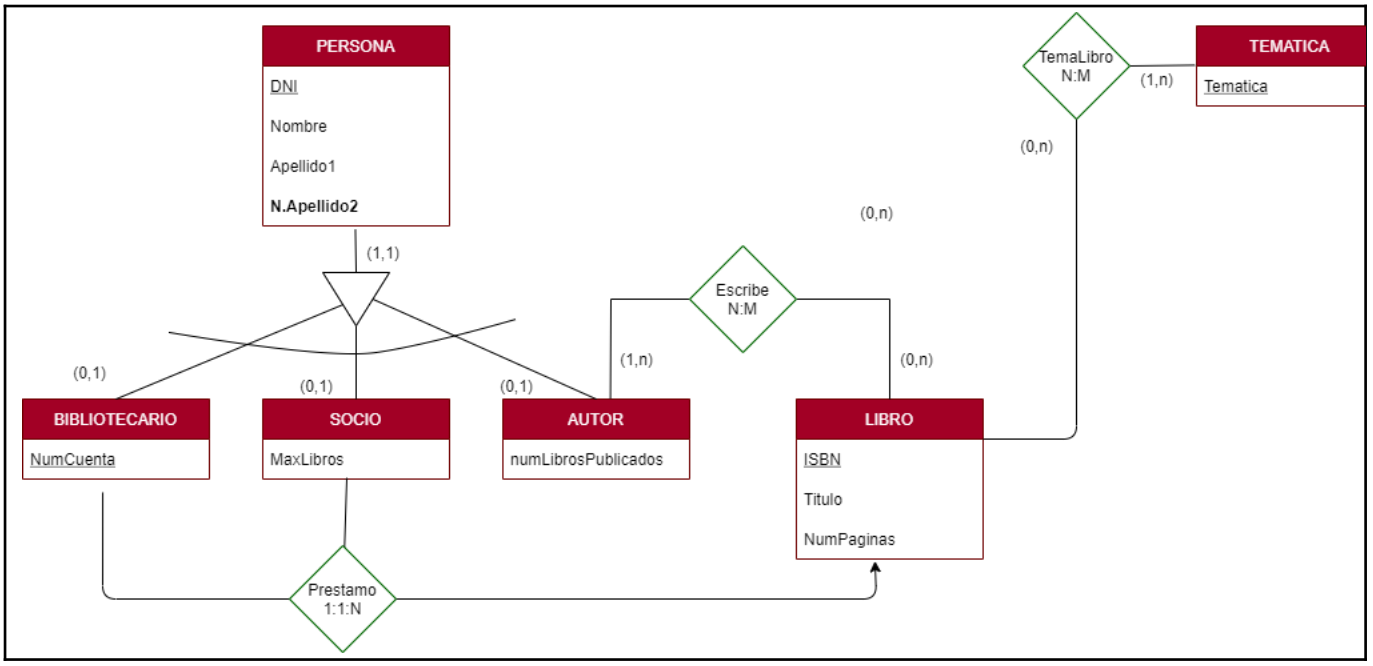
Ejercicio 1

Utilizando las relaciones ISA vamos a modificar el diseño que hicimos en el ejercicio de la Biblioteca del documento "*T2_2_Ejercicios_Diseño_Conceptual". Vamos a tener en cuenta que:*

- *De los bibliotecarios conocer su número de cuenta para ingresarles la nómina*
- *De los socios queremos saber el número máximo de libros que se les puede prestar.*
- *De los autores se quiere saber el número de libros que ha publicado*



Solución



2.8. Limitando el dominio de los atributos

Cuando se indica un atributo en una entidad éste puede tomar cualquier valor. Para limitar esos valores se utilizan declaraciones que se conocen como **dominio de un atributo.**

Si tenemos una entidad *LIBRO* con el atributo *NumPaginas* podríamos indicar que este atributo sólo puede tomar valores entre 10 y 10.000, ya que seguramente no existan libros que tengan menos de 10 páginas ni más de 10.000.

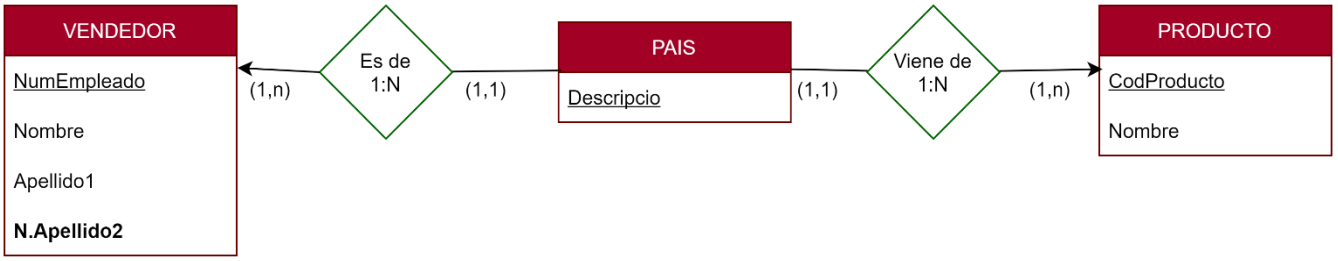
También podríamos declarar el dominio del atributo *DiaSemana* indicando que solo puede tomar los valores "Lunes", "Martes, "Miércoles", "Jueves", "Viernes", "Sábado", "Domingo".

A veces puede suceder que tenemos un atributo del que podemos conocer sus posibles valores y es utilizado por varias entidades, en este caso, puede interesarnos considerarlo como una entidad.

Por ejemplo el atributo País puede pertenecer a varias entidades, si no tiene un dominio explícito y sólo se declara como una cadena de 20 caracteres, podemos encontrar muchas maneras de escribir España, dependiendo que quién lo escriba (España, Espagna, Spanien, Spain, …), pero también puede que por error se escriba Espana o Ispaña en vez de España y tendríamos varios países diferentes.

En estas entidades el atributo País sería introducido por un usuario de la aplicación, por tanto cada vez puede introducirlo de una manera distinta y podría parecer que no tiene nada que ver.

Si creamos una entidad *PAIS* con el atributo *Descripcion* como atributo principal obligamos a relacionar un ejemplar de la entidad VENDEDOR con un ejemplar de la entidad PAIS. La relación entre las entidades que contienen el atributo y las nuevas entidades suele haber una multiplicidad de 1:N



El atributo país en VENDEDOR y PRODUCTO se sigue almacenando pero ahora ya no está abierto, habrá que elegirlo de una lista de países propuesta en la entidad PAIS

Con la creación de la entidad *PAIS* conseguimos limitar los valores del atributo *País* a los valores que existan en los ejemplares de la entidad *PAIS* y además esos valores pueden ser compartidos por otras entidades como *VENDEDOR y PRODUCTO.*

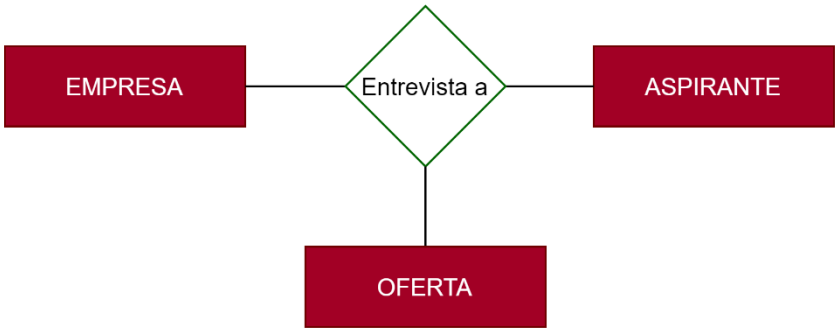
Durante el proceso de modelado suele ocurrir esto con bastante frecuencia lo que nos llevará a crear un gran número de entidades haciendo que el diagrama sea bastante engorroso. Para reducir esta representación se utiliza la técnica de **codificación:** pondremos delante del atributo que vamos a convertir en entidad el símbolo © o la letra **C.** De esta manera el diagrama anterior quedaría:

No es necesario decir que si en los requisitos se indicara que se quiere almacenar información sobre los países no se dudaría en crear PAIS como una entidad con los atributos correspondientes.

2.9. La relación agregación: Composición y colección

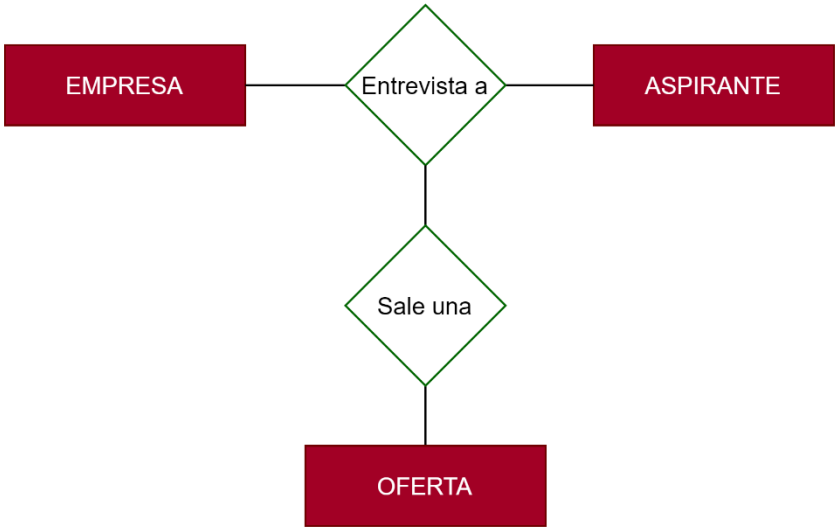
Con las reglas básicas del modelo ER sólo se pueden crear relaciones entre entidades, pero no se puede expresar la posibilidad de que una relación participe con otra relación. Hay un mecanismo llamado **agregación** que nos permite saltarnos esta limitación considerando una relación entre entidades como una entidad y utilizándola como tal.

Ejemplo
Supongamos que tenemos que modelar la siguiente situación: una empresa de selección de personal realiza entrevistas a diferentes aspirantes. Puede ser que de algunas de estas entrevistas se derive una oferta de empleo o puede ser que no.
Solución 1:



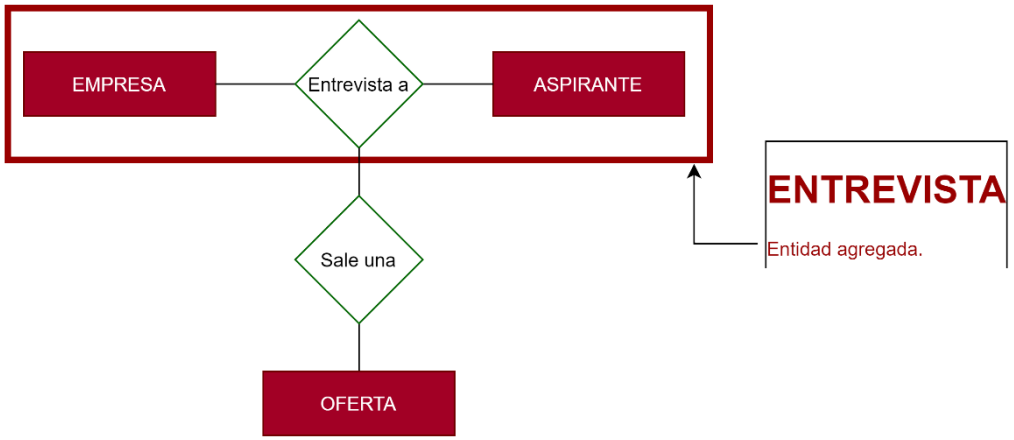
Esta solución no sería válida porque estamos representando que de cada entrevista realizada por una empresa a un aspirante sale una oferta de trabajo

Solución 2:



Esta solución tampoco sería correcta porque en el modelo ER no pueden establecerse relaciones entre relaciones.

Solución 3:



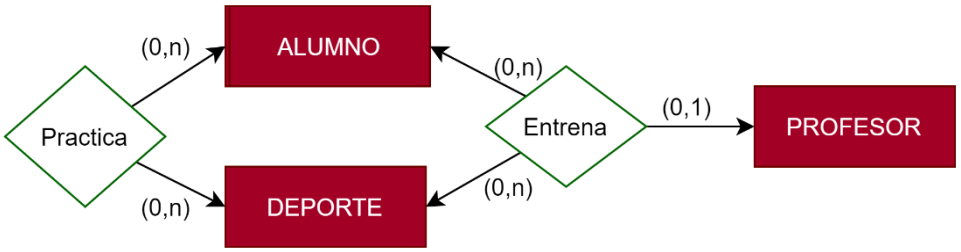
En el modelo ER podemos crear una **entidad agregada** llamada *ENTREVISTA* compuesta por la relación *Entrevista a* entre *EMPRESA* y *ASPIRANTE,* y una relación *Sale una* entre esta nueva entidad agregada y la entidad *OFERTA*

Como vemos la representación gráfica de una agregación se caracteriza por englobar con un rectángulo las entidades y la relación a abstraer. De este modo, se crea una nueva entidad agregada que puede participar en otras relaciones con otras entidades.

Ejemplo

Consideremos la relación binaria de cardinalidad N:M entre *ALUMNO y DEPORTE* que hemos utilizado en alguna ocasión. Supongamos que ahora queremos saber qué profesor (si es que lo hay) se encarga de entrenar a un alumno en un deporte determinado.

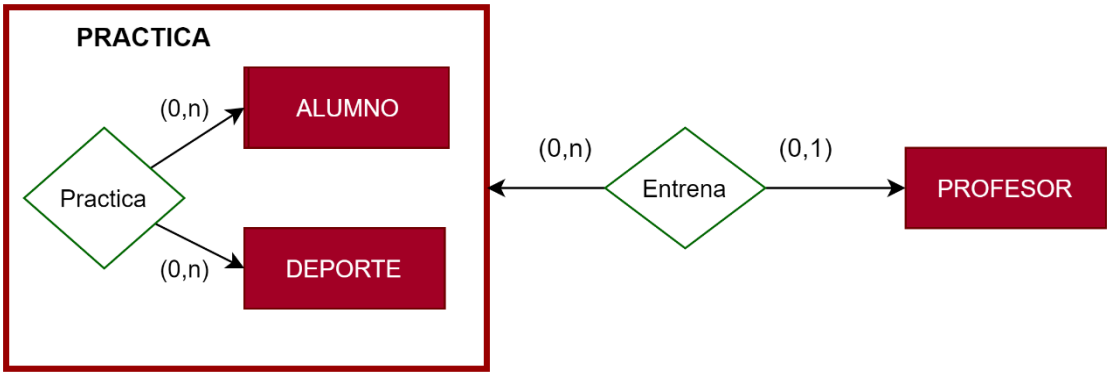
Solución 1



De esta manera puede parecer que las relaciones *Practica* y *Entrena* se pueden combinar en una sola relación, pero esto no es del todo cierto porque habrá relaciones entre *ALUMNO* y *DEPORTE* que no necesiten *PROFESOR* que actúe como entrenador.
En este esquema tendremos información redundante ya que cualquier combinación de instancias de *ALUMNO* y *DEPORTE* que hay en *Entrena* también estará en *Practica.*
Si PROFESOR fuera sólo un valor, podríamos plantearnos añadir simplemente un atributo *Entrenador* en la relación *Practica,* pero tenemos la entidad *PROFESOR,* así que mejor no hacerlo de esta manera.

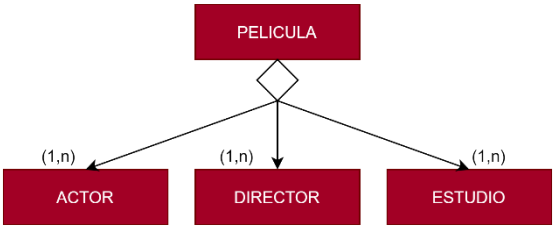
Solución 2

Por tanto la mejor manera de reflejar todas estas circunstancias es utilizando una relación de agregación considerando la relación *Practica* entre *ALUMNO y DEPORTE* como otra entidad de nivel más alto llamada *PRACTICA*

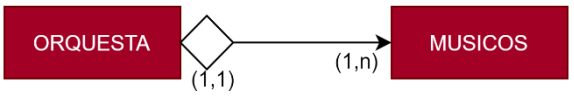


Existe dos tipos de agregaciones:

- **Composición/componente**: un todo se obtiene por la unión de varias partes que pueden ser entidades distintas y desempeñar papeles distintos en la agregación
- **Colección/miembro**: el todo está formado por una colección de miembros todos del mismo tipo de entidad y jugando el mismo rol.



Agregación composición/componente



Agregación colección/miembro

2.10. Pistas para elegir entre varios constructores

Para saber qué constructor elegir en función del resultado que queramos obtener podemos tener en cuenta los siguientes puntos:

- Elección de entidad en vez de atributo
- Transformación de relaciones ternarias en binarias sin perder información
- Codificación de atributos que se comparten o que se quieren limitar
- Reunir relaciones de la misma o distinta multiplicidad en relaciones jerárquicas

2.10.1. Atributos que son entidades

Para decidir si un atributo es una entidad o una propiedad:

Si existe información descriptiva sobre un concepto u objeto es una entidad

Si sólo se necesita un identificador para un objeto es un atributo, siempre que no necesite codificarlo

Por ejemplo, queremos almacenar la ciudad en la que se encuentra una oficina , ¿Atributo o entidad?

¿Necesitamos guardar información de la ciudad como nº habitantes o solamente su nombre? Si solo necesitamos su nombre, será un atributo de la entidad *OFICINA*, si necesitamos guardar más información, entonces será una entidad. Si queremos limitar los valores que puede tomar el atributo ciudad, codificaremos dicho atributo a través de su nombre o un código único.

Si un atributo se relaciona con más de una entidad, lo más conveniente sería crear una entidad que se relacione con las anteriores. Por ejemplo si queremos guardar información sobre las ciudades donde un proveedor tiene su sede, tendremos que ciudad es atributo de *PROVEEDOR* y de *OFICINA*, en este caso lo mejor es construir la entidad *CIUDAD* que se relacione con *PROVEEDOR* y *OFICINA*.

Cuando se modele un atributo como multivaluado, lo mostraremos en el diseño conceptual como tal si:

- tiene un número limitado y poco elevado de ocurrencias

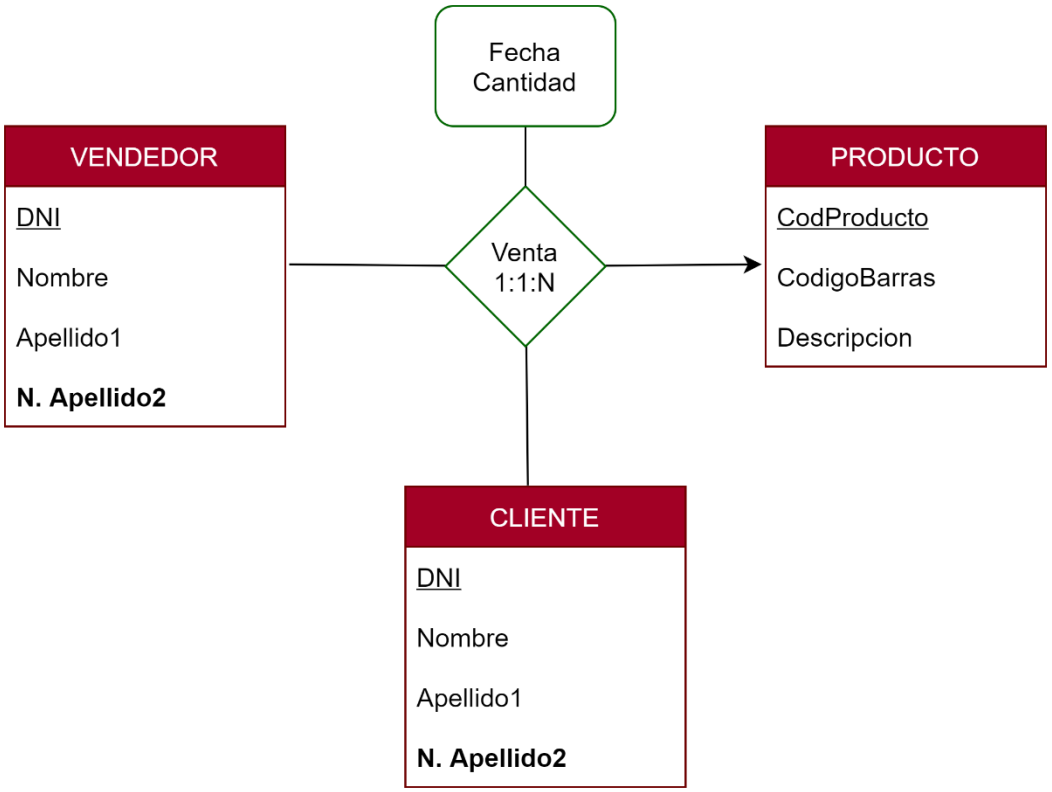
- no está relacionado con otros tipos de entidad

Si no es así, lo crearemos como entidad.

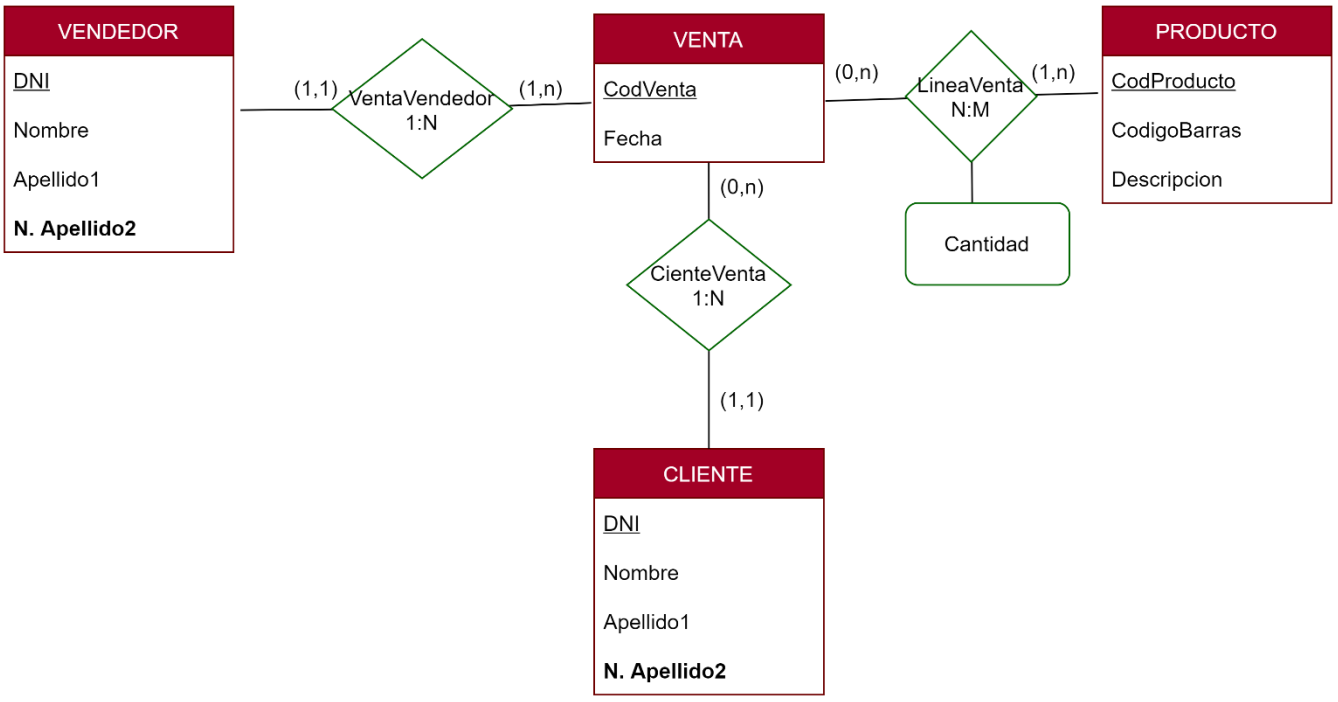
2.10.2. Consideraciones con las relaciones ternarias

Muchas veces las relaciones ternarias complican la representación semántica entre 3 entidades. Además suelen tener atributos que complican aún más su descripción. Por eso a veces es más sencillo transformar esa relación en una nueva entidad.

Por ejemplo, si queremos representar las ventas de una empresa, normalmente participan las entidades *CLIENTE, PRODUCTO y VENDEDOR.* En una misma venta el número de productos es normalmente mayor que uno e incluso de un mismo producto pueden comprarse varias unidades, lo que genera un atributo *cantidad*



Si convertimos la relación *Venta* en una entidad VENTA:



Este tipo de construcción es mejor realizarla cuando las relaciones ternarias no nos dan información con suficiente claridad o cuando tienen demasiados atributos.

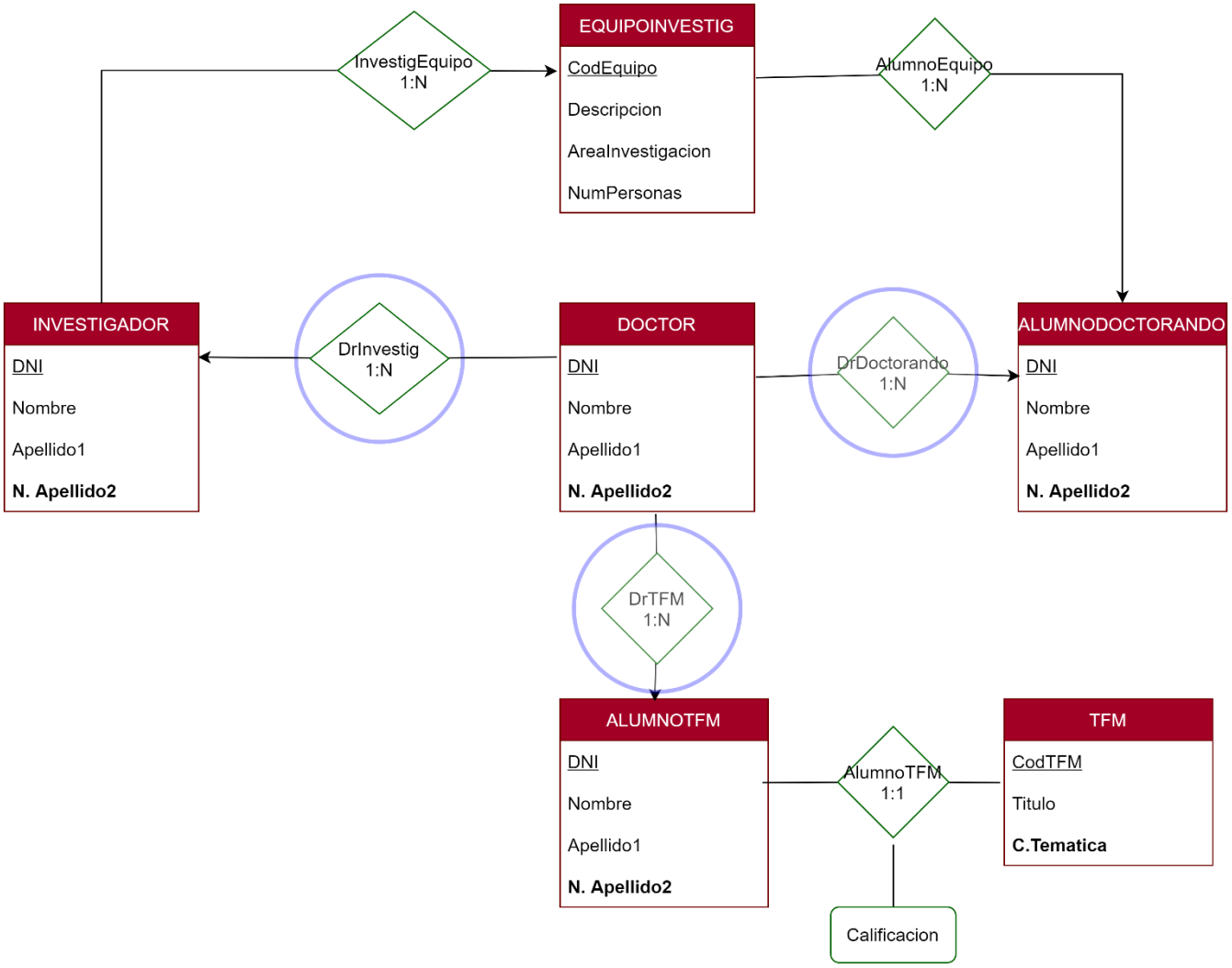
2.10.3. Codificación de atributos

Debemos tener en cuenta los casos en los que una entidad posee un atributo cuyos valores están en un dominio limitado y que además pueden usarse en otros atributos de otras entidades o relaciones. En estos casos es mejor convertir esos atributos en una entidad, es decir, codificamos el atributo.

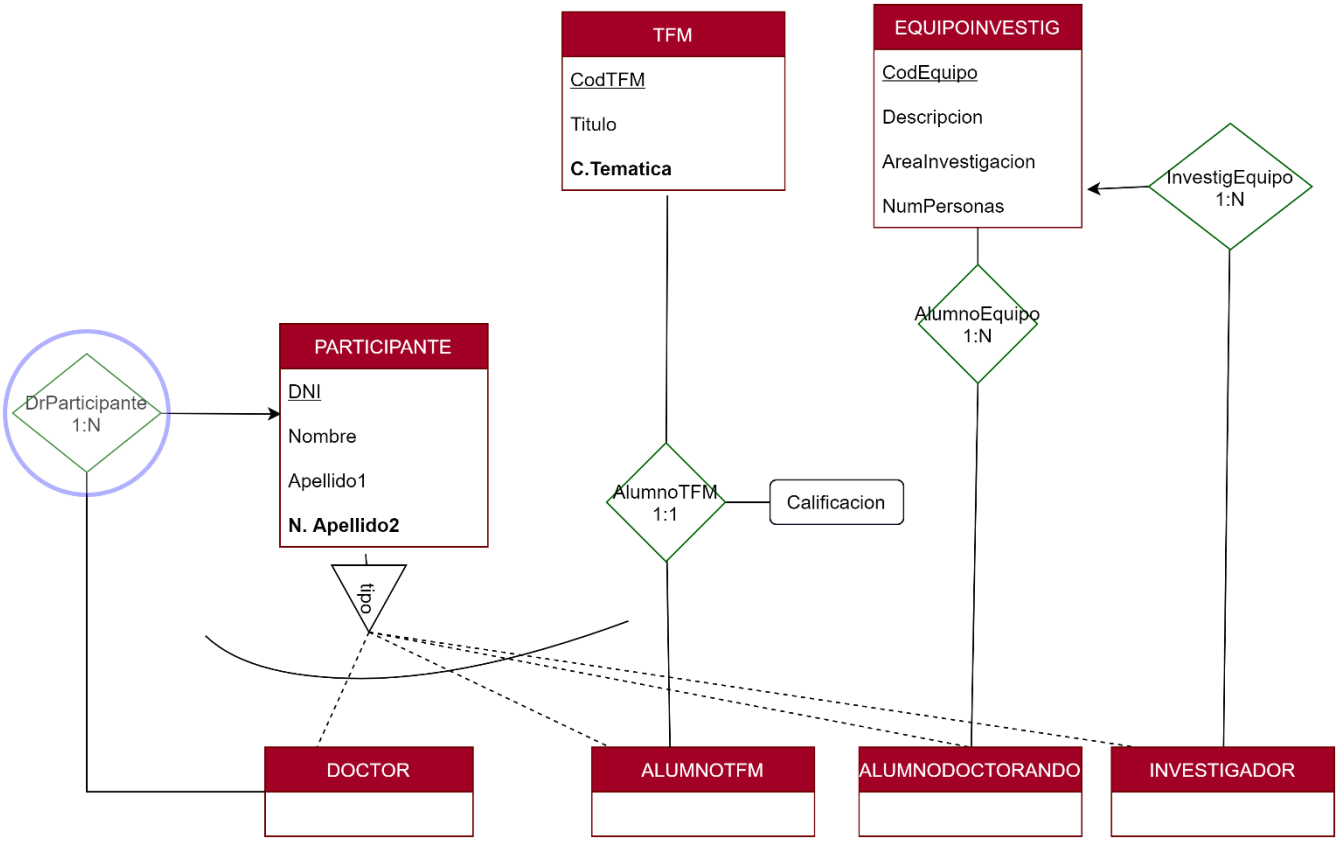
2.10.4. Agrupar relaciones de igual o diferente multiplicidad

Si tenemos los siguientes requisitos:

- En el departamento de informática de una universidad se quiere implantar una base de datos para gestionar los trabajos de fin de máster (TFM), los trabajos que llevan los alumnos de doctorado y los equipos de investigación que se dirigen. De los doctores, alumnos e investigadores se quiere almacenar su DNI, nombre y apellidos.
- Los alumnos que estudian un máster deben finalizar su trabajo fin de máster. Se quiere almacenar los datos del trabajo de fin de máster, título, temática, calificación y doctor que la dirige.
- Los alumnos doctorandos que están realizando sus estudios para la tesis doctoral pertenecen a un equipo de investigación y un doctor del departamento los dirige. De estos equipos se almacena su código, descripción, área de investigación y número de personas que pertenecen a él.
- Los equipos de investigación están formados por alumnos doctorandos y un investigador del departamento. Un investigador dirige a un equipo de investigación y éste es supervisado por un doctor del departamento.



Si vemos las relaciones *DrInvestig, DrDoctorando y DrTFM*, todas relacionan la entidad *DOCTOR* con otras 3 entidades y en las tres relaciones se da la misma multiplicidad, por tanto podemos especializar estas 3 entidades en una sola llamada *PARTICIPANTE,* y recoger las 3 relaciones entre *DOCTOR y PARTICIPANTE.*
Si la multiplicidad de las relaciones que se quieren agrupar no es la misma, se puede hacer la especialización considerando como óptima la multiplicidad máxima.


TFM
CodTFM
Titulo
C.Tematica
EQUIPOINVESTIG
CodEquipo
Descripcion
AreaInvestigacion
NumPersonas
InvestigEquipo
1:N
AlumnoEquipo
1:N
PARTICIPANTE
DNI
Nombre
Apellido1
N. Apellido2
DrParticipante
1:N
AlumnoTFM
1:1
Calificacion
tipo
DOCTOR
ALUMNOTFM
ALUMNODOCTORANDO
INVESTIGADOR