

## 2. Transformación del modelo Entidad-Relación a Modelo Relacional

---

### 2.1. Paso del modelo conceptual Entidad-Relación al modelo lógico Relacional

El modelo Entidad-Relación y el modelo relacional producen representaciones abstractas del mundo real, pero debido a que utilizan principios de diseño similares, se puede convertir un diseño Entidad-Relación en un diseño relacional siguiendo una serie de reglas que se resumen en los siguientes apartados. Se seguirá fundamentalmente la metodología propuesta por Adoración de Miguel y Mario Piattini.

Durante el proceso de transformación, puede surgir confusión en la utilización del término relación ya que:

- En el modelo Conceptual, una relación del diagrama Entidad-Relación representa un vínculo entre entidades.
- En el modelo Relacional, la relación es el elemento básico del modelo y se puede considerar como un conjunto de datos que se pueden representar como una tabla, en la que a cada fila se le llama tupla y cada columna corresponde a un atributo. Por extensión, se puede utilizar el término tabla en lugar de relación del modelo Relacional, aunque no son exactamente lo mismo.

El propósito del diseño lógico estándar es que sea aplicable a diferentes tecnologías de BD, por lo que no debe emplear características específicas de los SGBDR comerciales. El resultado del diseño debe ser altamente transportable, no sólo para ser empleado por diferentes gestores, sino también para facilitar la propagación entre versiones de un mismo sistema. Por otra parte, los SGBD poseen características propias que los diferencian, por lo que es obligatorio realizar adaptaciones del esquema lógico antes de crear la base de datos. Este es el motivo por el que se puede considerar que existen dos etapas dentro del diseño lógico:

- Primera etapa. En esta etapa se obtiene el esquema lógico estándar a representar mediante un grafo relacional, totalmente independiente del SGBDR en el que se va a crear la base de datos.
- Segunda etapa. A partir del esquema lógico estándar obtenido en la etapa anterior, se elaborará el esquema lógico específico a implementar en un SGBDR concreto empleando las herramientas proporcionadas por el propio SGBDR.

### 2.2. Transformaciones del modelo Entidad-Relación (MER) a modelo Relacional (MR)

Se distinguirá entre la transformación básica que permitirá iniciar las transformaciones del MER al MR y el resto de transformaciones que completan la básica.

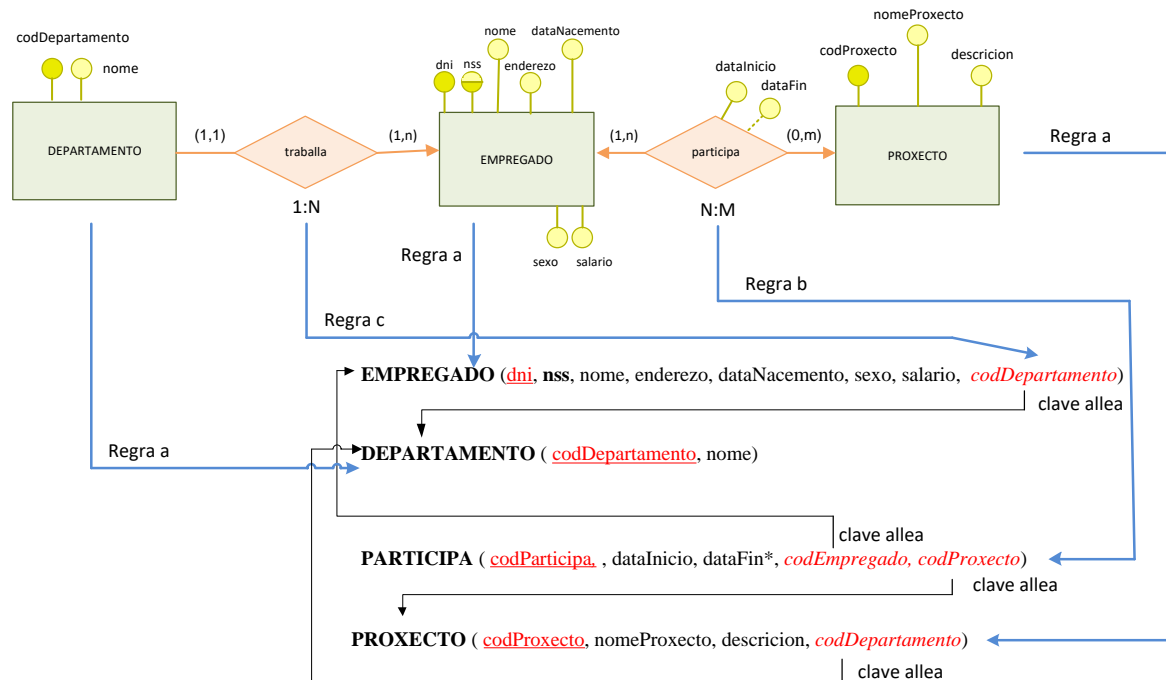
#### 2.2.1. Transformación básica

El paso de un esquema en el modelo entidad-relación al modelo relacional se fundamenta en las tres reglas básicas siguientes:

- Regla a. En general, todo tipo de entidad del MER se convierte en una relación (tabla) en el MR que mantiene atributos y claves respecto a la entidad de origen.

- Regla b. Todo tipo de relación del MER (también llamado interrelación) con tipo de correspondencia N:M, se transforma en una relación del MR mediante la técnica de propagación de clave; en la misma, figurarán como claves foráneas las claves principales de las relaciones procedentes de la conversión de las entidades que participan en la interrelación.
- Regla c. Todo tipo de relación del MER con tipo de correspondencia 1:N, o bien, se traduce en la propagación de clave, o bien, genera una nueva relación del MR.

Por ejemplo, en el siguiente diagrama Entidad-Relación:



Al aplicar la regla a, resultan las relaciones del MR: *EMPLEADO*, *PROYECTO* Y *DEPARTAMENTO*. Al aplicar la regla b sobre el tipo de relación *participa* del modelo Entidad-Relación, se obtiene la relación *PARTICIPA* que debe tener como claves foráneas las claves primarias de *PROYECTO* y *EMPLEADO*. Al aplicar la regla c a la relación *trabaja* del modelo Entidad-Relación, se propaga la clave de *DEPARTAMENTO* a la relación *EMPLEADO*, es decir, se añade un atributo en la relación *EMPLEADO* que es una clave ajena o foránea que hace referencia a la clave principal de *DEPARTAMENTO*.

En la relación *PARTICIPA* del MR aparece un nuevo atributo como clave primaria, como consecuencia de la aplicación de una regla general, que se explicará más adelante, que indica que las relaciones que surgen de una interrelación N:M deben tener como clave principal la concatenación de las claves principales de las entidades relacionadas; en este caso la razón de la conducta radica en evitar una clave compuesta de tres atributos (las dos foráneas más la fecha de inicio para que cumpla el principio de unicidad) para mejorar el rendimiento.

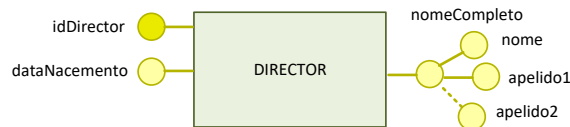
El modelo Relacional pierde semántica con relación al modelo Entidad-Relación ya que no distingue entre entidades e interrelaciones, al representar ambos mediante relaciones:

- Las interrelaciones N:M del MER no se distinguen de las entidades del MER, ya que ambas se transforman en tablas del MR.
- Las interrelaciones 1:N del MER se suelen representar mediante una propagación de clave, desapareciendo el nombre de la relación, aunque este nombre puede ser asignado a las claves ajenas generadas.

### 2.2.1.1. Regla 1. Transformación de entidades

En general, cada entidad se transforma en una relación (tabla) en el grafo relacional, con el mismo nombre que la entidad de la que proviene y conserva sus atributos y sus claves. Más adelante se verán algunos casos en los que esta regla se complementa fundiendo dos o más entidades en una única entidad. Ejemplo:

#### MER



#### Esquema MR (intensión)

**DIRECTOR** ( idDirector, nome, apellido1, apellido2\*, dataNacimiento)

#### Extensión MR

<u>idDirector</u>	nome	apellido1	apellido2	dataNacimiento
987654987	Antón	Reixa	Rodríguez	17-04-1957
236785439	José Luís	Cuerda	Martínez	18-02-1947
879654190	Isabel	Coixet	Castillo	09-04-1960
368965478	Vincenzo	Natali	NULO	06-01-1969
456952134	Steve Allan	Spielberg	NULO	18-12-1946
567423908	Hans Christian	Tomas	Alfredson	01-04-1965

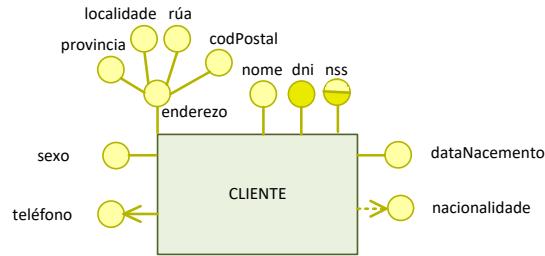
clave primaria

### 2.2.1.2. Regla 2. Transformación de atributos multivalor

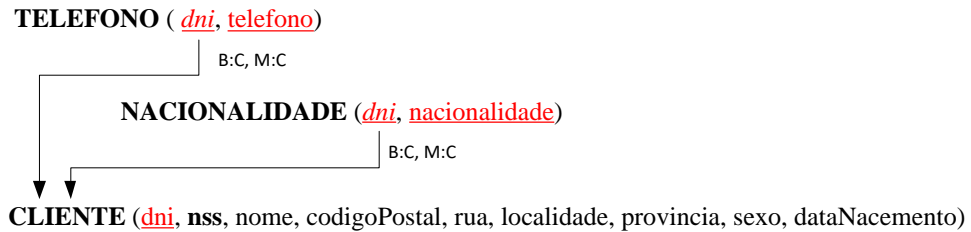
La restricción de dominio inherente al modelo relacional establece que cada atributo sólo puede tomar un valor para cada tupla. Teniendo en cuenta esto, la forma de representar el tipo de atributos multivaluados es creando, para cada uno de ellos, una nueva relación que tiene como atributos la clave primaria de la entidad y el atributo multivaluado. La clave primaria de la nueva relación es compuesta y está formada por la combinación de los dos atributos. El atributo que corresponde a la clave primaria de la entidad funciona en esta nueva relación como clave foránea que permite enlazar cada tupla de la nueva relación con una tupla de la relación resultante de transformar la entidad, y para mantener la integridad relacional debería tener una política de borrado y modificación en cascada (B:C,M:C).

Por ejemplo, en la siguiente imagen queda plasmado que un cliente puede tener uno o más teléfonos y más de una nacionalidad, aunque es posible que se desconozca su nacionalidad. Para que se mantenga la integridad referencial, el vínculo entre *TELEFONO* y *NACIONALIDAD* con la relación *CLIENTE* lleva asociado el comportamiento B:C, de tal manera que al borrar un cliente, se borran las filas que contengan el dni de ese cliente en las relaciones *TELEFONO* y *NACIONALIDAD*.

## MER



## Esquema MR



En el grafo relacional anterior, no se recoge la diferencia entre un atributo multivaluado opcional (puede que no tome ningún valor) y un atributo multivaluado obligatorio (tiene que tomar al menos un valor). Esta información tiene que ser detallada en un documento que acompaña el grafo relacional, en el que se plasman las restricciones que no aparecen en el grafo relacional pero que hay que tener en cuenta en el momento de la creación de la base de datos durante la fase del diseño físico.

### 2.2.1.3. Regla 3. Transformación de relaciones del MER

La transformación de una relación depende de si es reflexiva, binaria con tipo de correspondencia N:M, 1:N o 1:1, o si es n-aria.

#### Regla 3.1. Transformación de relaciones de tipo N:M

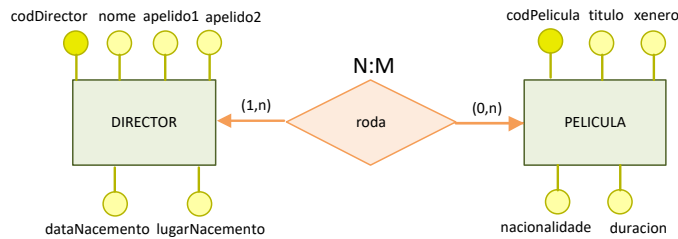
Los tipos de relación de muchos a muchos o correspondencia tipo N:M, dan lugar a una nueva relación o tabla en el MR, que tendrá claves foráneas apuntando a cada una de las claves primarias correspondientes a las entidades que asocia, y los atributos descriptivos de la relación (si es que existen).

El nombre de la nueva relación en el esquema relacional debería ser el nombre de la relación del diagrama Entidad-Relación del que procede, o la concatenación de los nombres de las entidades que une para evitar la pérdida semántica que aparece al no poder diferenciar relaciones procedentes de la transformación de relaciones.

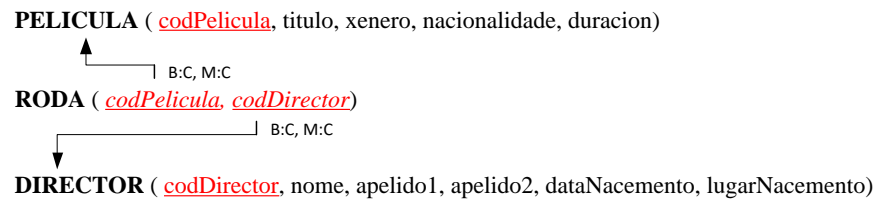
La clave principal de la relación surgida, habitualmente, está compuesta por la concatenación de las claves primarias de las entidades tipo asociadas. En algunos casos, no llegan estos dos atributos para identificar únicamente las tuplas de la relación, precisando anexar algún atributo de la relación para diferenciarlas.

Por ejemplo, el siguiente diagrama Entidad-Relación recoge información de películas rodadas por directores, en la que se refleja que un director puede tener rodadas varias películas, y en el rodaje de una película pueden participar varios directores. El diagrama Relacional obtenido tiene tres relaciones, una por cada entidad y otra para la relación *rueda*.

## MER



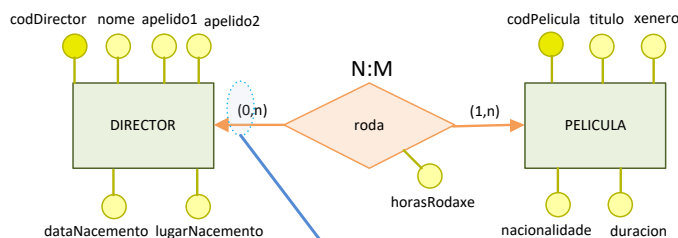
## Esquema MR



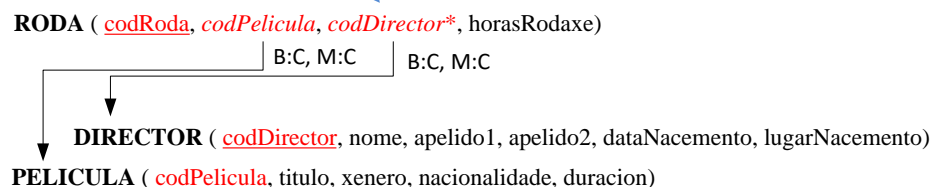
Es recomendable añadir un nuevo atributo que haga de clave primaria para evitar una clave compuesta en los siguientes casos:

- Cuando el tamaño de la clave compuesta es tan grande, que es preferible la creación de una clave propia para una manipulación más eficiente de los datos.
- Cuando hay atributos en la relación y alguna de las entidades participa de forma opcional, es decir, representada en el diagrama Entidad-Relación con una cardinalidad mínima 0. La clave ajena que hace referencia a la relación que representa esa entidad, debería poder almacenar valores nulos (NULL), pero esto no estaría permitido si forma parte de la clave primaria. Por ejemplo, en el siguiente diagrama Entidad-Relación, se admite que pueden existir en la relación *roda* tuplas de las que no se conoce el director, es decir, que tengan el atributo *codDirector* con valor nulo, pero interesa almacenar igualmente las horas de rodaje.

## MER

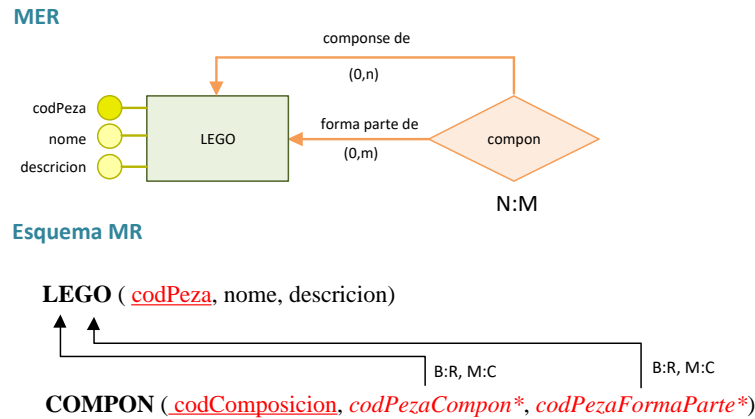


## Esquema MR



En el caso de que la relación de tipo N:M sea reflexiva, es decir, cuando en la relación participa una sola entidad, se procede de igual manera que en las binarias, creando una nueva relación en el grafo relacional, pero teniendo en cuenta que al propagar la clave principal dos veces, el nombre de la misma debe modificarse para que no aparezca repetida en la nueva relación. Por ejemplo, en la siguiente transformación, la relación *COMPONE* del MR no puede tener el atributo *codPieza* dos veces con el mismo nombre, por lo que se

adopta el sistema de añadir en su lugar los atributos *codPiezaCompone* y *codPiezaFormaParte* como claves foráneas.



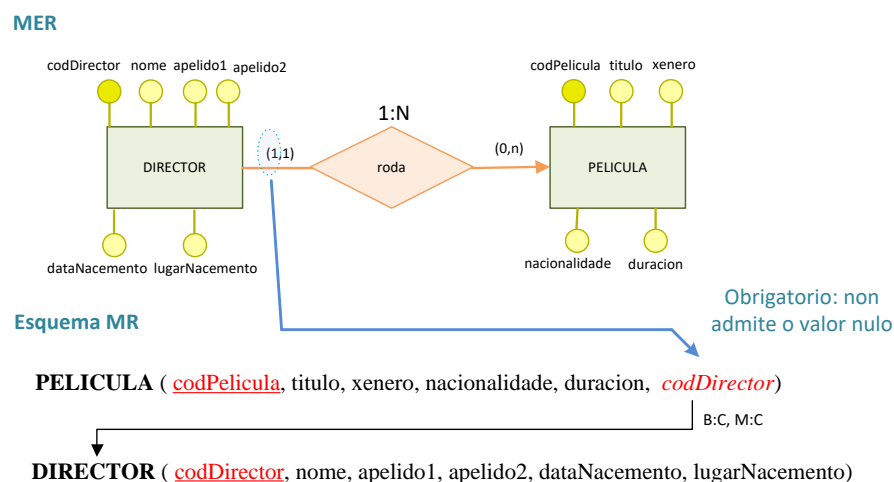
### Regla 3.2. Transformación de relaciones de tipo 1:N

Las relaciones con tipos de correspondencia 1:N se pueden transformar mediante dos métodos:

- Mediante propagación de clave.

Se añade un nuevo atributo en la relación que se crea para la entidad señalada por la flecha que simboliza el lado N, que representa una restricción de clave foránea y hace referencia a la otra entidad que participa en la relación. Este método es el recomendado cuando la entidad del lado de la N tiene una participación obligatoria, es decir, tiene cardinalidad mínima 1, y por tanto la clave foránea creada no admite valores nulos.

Por ejemplo, en el siguiente gráfico, la entidad *PELICULA* se va a transformar en una relación a la que se añade el atributo *codDirector* que hace referencia a la relación *DIRECTOR*. Toda película tiene obligatoriamente un director porque la entidad *PELICULA* tiene una participación obligatoria en la relación *rueda*, lo que significa que la clave foránea *codDirector* no puede tomar el valor nulo y tiene que tomar un valor que coincida con un valor de la clave primaria de la relación *DIRECTOR*.



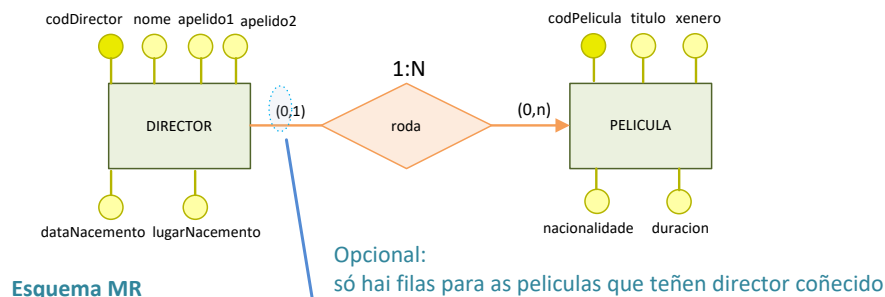
- Mediante la creación de una nueva relación.

Es un método similar al de las relaciones de tipo N:M. Se crea una nueva relación que tendrá como atributos las claves de ambas entidades, pero la clave primaria será únicamente la de la entidad señalada por la flecha que simboliza el lado N. Este método es el recomendado cuando la entidad del lado de la N tiene una participación opcional, es

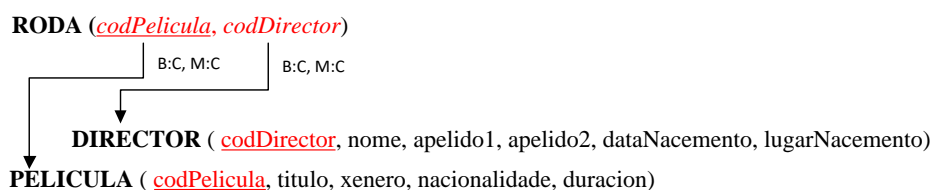
decir, tiene cardinalidad mínima 0, y por lo tanto pueden existir muchos valores nulos en la clave foránea, lo que no es deseable.

Por ejemplo, en el siguiente gráfico, en la relación *rueda*, puede haber películas que no tengan asociado un director. De utilizarse el método de propagación de clave, podrían existir tuplas en la relación *PELICULA* en las que la clave foránea tomase el valor nulo. El método de creación de una nueva relación en el grafo relacional sería lo más aconsejable, ya que entonces en *RUEDA* sólo habría tuplas para aquellas películas que tengan asociado un director.

#### MER



#### Esquema MR



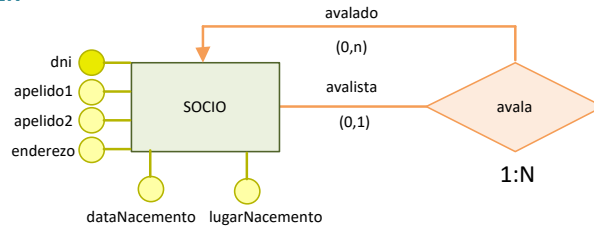
Otros casos en los que puede ser recomendable este método:

- Cuando se prevé que la relación del MER se convierta en el futuro en una de tipo N:M y se quiera dejar abierta esta posibilidad, ya que la transformación de tipo N:M es más flexible. Por ejemplo, de querer dejar abierta la posibilidad de que una película sea rodada por varios directores.
- Cuando la relación del MER tiene atributos propios y no se desea su propagación para conservar así la semántica original. Este proceder es el recomendable en las relaciones reflexivas. Por ejemplo, de querer conocer ciertos datos del rodaje.

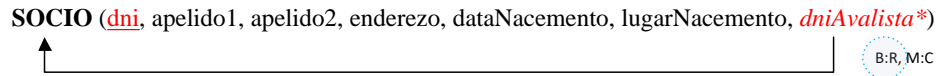
En las relaciones de tipo 1:N reflexivas, se procede de igual manera que en las binarias 1:N. Esto es, o se propaga la clave o se genera una nueva relación. En el caso de propagar la clave, se debe nombrar la clave foránea de tal modo que no aparezca un nombre de atributo repetido. Otra precaución a tener en cuenta con las relaciones reflexivas es que no se debe utilizar la política de borrado en cascada (B:C) porque podría provocar el borrado de todas las tuplas de la relación.

Por ejemplo, en el siguiente diagrama Entidad-Relación, se representa el hecho de que un socio puede ser avalista de 0, 1 o varios socios y que un socio puede ser avalado por 0 o 1 socio. De adoptar el método de propagación de clave, se obtendría el diagrama relacional que aparece a continuación, pero entonces podría haber varias tuplas que tuvieran nulo en la clave foránea. Hay que tener en cuenta que no se debería de utilizar la política B:C ya que entonces cuando se borra un socio, se borrarían todos los socios avalados por él, y si esos socios fueran avalistas de otros, provocaría que también se borrarán los socios avalados, y se podrían seguir realizando borrados en cascada hasta llegar a borrar todos o casi todos los socios.

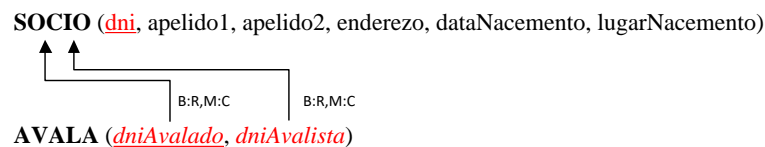
## MER



## Esquema MR



Una solución mejor que la anterior es utilizar el método de creación de una nueva relación, ya que entonces, en la relación *AVALA* solo habría tuplas para los socios avalados por otro socio, y los que no tienen evaluación ya no aparecerían en la relación, tal y como se muestra en el siguiente grafo.



## Regla 3.3. Transformación de relaciones de tipo 1:1

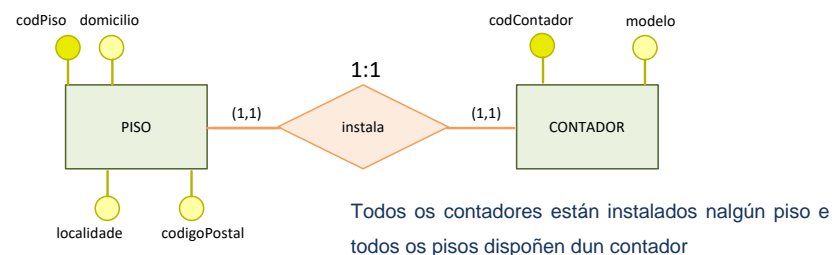
Para la transformación de las relaciones de tipo 1:1 del diagrama Entidad-Relación hay que tener en cuenta el tipo de participación de las entidades en la relación observando las cardinalidades mínimas. Se pueden dar los siguientes casos:

- El tipo de participación es obligatoria para todas las entidades que intervienen en la relación.

En el caso de una relación binaria, esto significa que cada ocurrencia de la primera entidad ha de estar relacionada obligatoriamente con una ocurrencia de la otra entidad, y viceversa. La solución es crear dos relaciones en el grafo relacional y añadir en una de ellas una clave ajena que relaciona cada tupla propia con una tupla de la otra relación; es mejor que reciba la clave la relación a la que se va a acceder con más frecuencia.

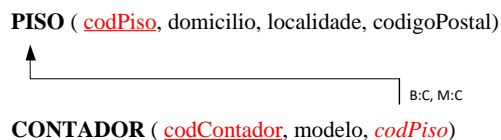
Por ejemplo, en el siguiente diagrama Entidad-Relación, se representa información sobre contadores de la luz y pisos en los que están instalados, indicando que todos los pisos tienen un contador instalado obligatoriamente y todos los contadores de los que tenemos información están instalados en un piso. Se crean en el diagrama relacional las relaciones *PISO* y *CONTADOR* y se añade la clave ajena *codPiso* en esta última.

## MER





### Esquema MR



En el caso de no existir relación de esas entidades con otras entidades diferentes, se podría adoptar la solución de crear una única relación que tuviera todos los atributos de las entidades que intervienen en la relación, cogiendo como clave primaria la de cualquiera de las entidades. Por ejemplo, para el diagrama Entidad-Relación anterior, el diagrama relacional sería el siguiente.

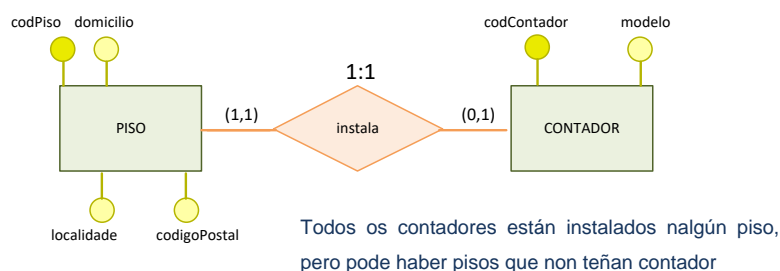
### Esquema MR

**PISO** ( codPiso, domicilio, localidad, codigoPostal, codContador, modelo)

- El tipo de participación es obligatoria para una de las entidades que interviene en la relación y opcional para la otra.

La solución es tratar el caso como si fuera una relación de tipo 1:N, mediante propagación de clave en la entidad que tiene participación obligatoria. Por ejemplo, modificando el anterior diagrama Entidad-Relación para indicar que hay pisos que no están relacionados con ningún contador pero todos los contadores están relacionados con algún piso. Es decir, la entidad *CONTADOR* tiene participación obligatoria y *PISO* tiene participación opcional. Se obtendría el siguiente diagrama relacional:

### MER



### Esquema MR

**PISO** ( codPiso, domicilio, localidad, codigoPostal)

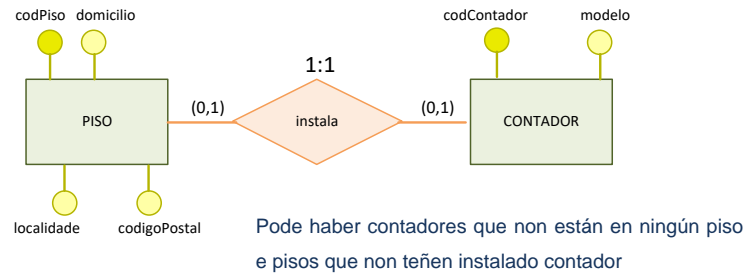
**CONTADOR** ( codContador, modelo, codPiso)

B:R, M:C

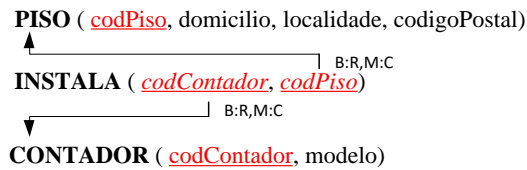
- El tipo de participación es opcional para todas las entidades que intervienen en la relación.

Este caso se trata como si fuera una relación de tipo N:M, creando una nueva relación que debe contener los atributos que son clave primaria de las dos entidades y además los atributos propios de la relación en el caso de que los hubiera. Por ejemplo, modificando el anterior diagrama Entidad-Relación para indicar que puede haber contadores que no están en ningún piso y pisos que no tienen instalado contador, se obtendría el siguiente diagrama relacional.

### MER

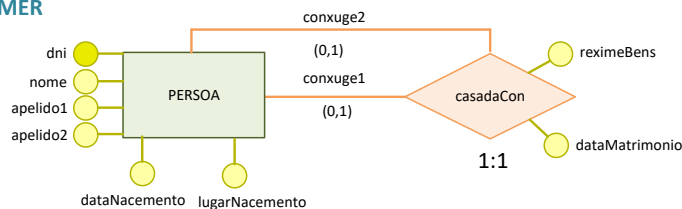


### Esquema MR

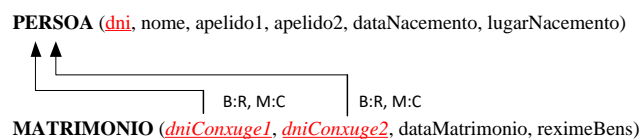


Sobre las relaciones de tipo 1:1 reflexivas, se aplican las mismas recomendaciones que para las binarias, recordando que si hay propagación de clave sobre la misma relación, no pueden permitirse los borrados en cascada. Por ejemplo, en el siguiente diagrama Entidad-Relación, hay información de personas que pueden estar casadas o no e indica que la participación de la entidad *PERSONA* en la relación es opcional por los dos lados. El diagrama relacional resultante es el siguiente.

### MER



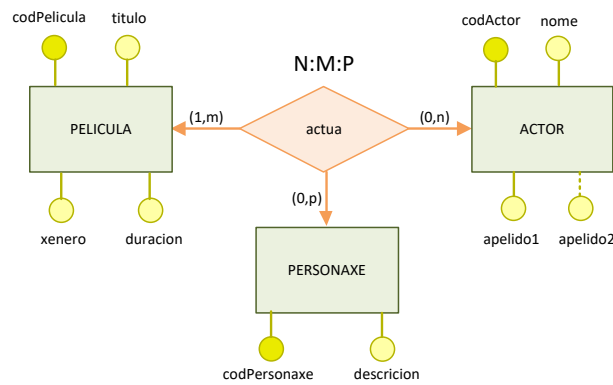
### Esquema MR



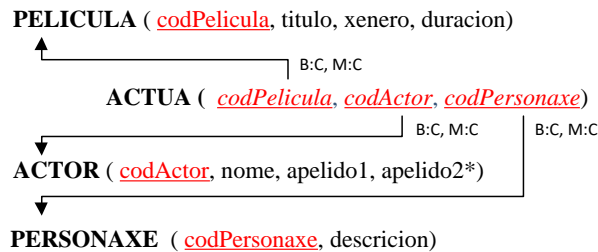
## Regla 3.4. Transformación de relaciones n-arias

Las relaciones ternarias, cuaternarias etc. se transforman del mismo modo que las relaciones N:M, es decir, generando una nueva relación que tendrá varias claves foráneas apuntando a cada una de las claves correspondientes a las entidades que vincula.

Por ejemplo, en el siguiente diagrama Entidad-Relación se representa la relación *actúa* entre las entidades *PELICULA*, *ACTOR* y *PERSONAJE* de tipo N:M:P. Esta relación no puede ser representada mediante relaciones binarias, porque en ese caso se produciría una pérdida de semántica, ya que se podría conocer los actores que actúan en una película, los personajes que son interpretados por un actor, pero no habría forma de obtener las películas en las que el actor interpretó esos personajes. Una transformación inicial del diagrama Entidad-Relación de esta relación ternaria en grafo relacional, crearía una relación *ACTUA* con una clave principal surgida de la concatenación de las claves tipo asociadas y sólo existirán tuplas con código de película, actor y personaje.

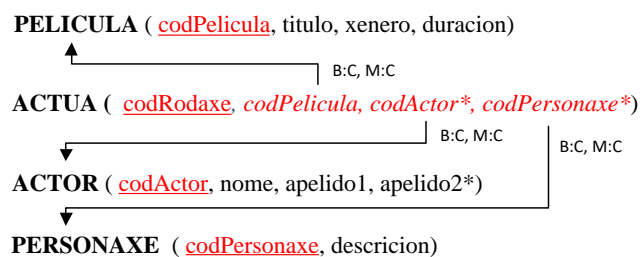


#### Esquema MR: solución 1



Una solución más eficiente que refleje la cardinalidad (0,N) con relación a *ACTOR* y el *PERSONAJE* consiste en crear un nuevo atributo clave principal específico al que se asociará un tipo autoincremental que maneja automáticamente el SGBD. De esta manera, las claves ajenas *codActor* y *codPersonaje* podrían admitir valores nulos para reflejar la situación de un personaje que no es representado por ningún actor (por ejemplo, un personaje de animación), o de un actor que actúa en una película pero no representa ningún personaje (por ejemplo, un actor que hace una voz en off). La transformación final daría lugar al siguiente diagrama relacional:

#### Esquema MR: solución 2



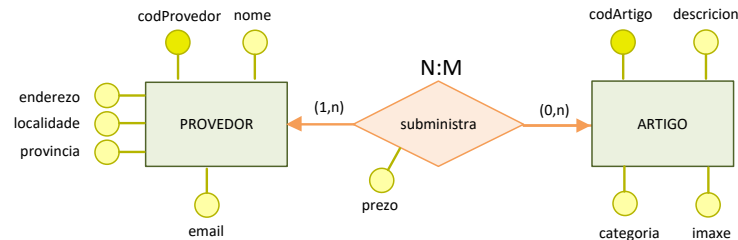
### Regla 3.5. Transformación de atributos de relaciones

La transformación de los atributos descriptivos de una relación del diagrama Entidad-Relación puede hacerse de dos maneras, en función del tipo de relación y la forma en que se transforma en el grafo relacional:

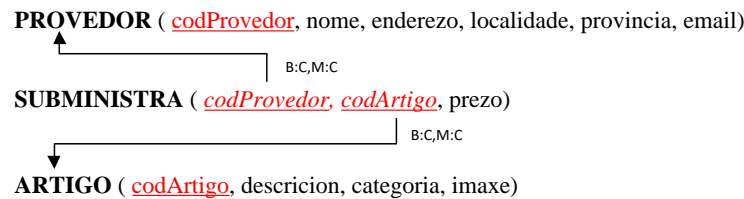
- Si los atributos pertenecen a una relación de tipo N:M o de tipo 1:1 que da lugar a la creación de una nueva relación en el grafo relacional, pasan a ser atributos de la nueva relación que se crea.

Por ejemplo, en el siguiente diagrama Entidad-Relación, se indica que puede haber un proveedor que suministre varios artículos, un artículo puede ser suministrado por varios proveedores y el precio del artículo varía en función del proveedor que lo suministra, la transformación da lugar al grafo relacional siguiente:

### MER



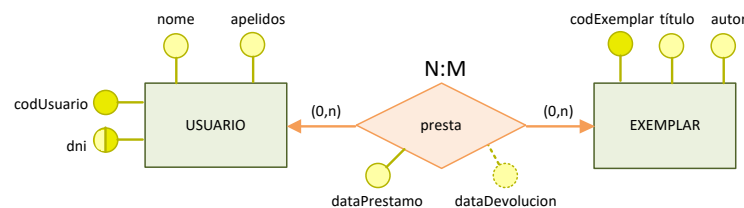
### Esquema MR



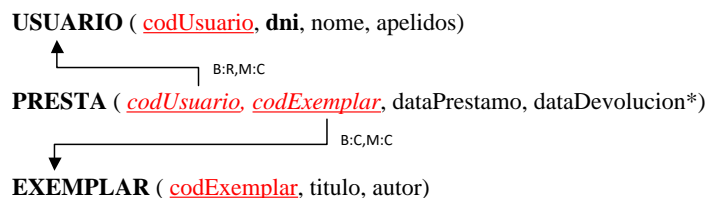
Un atributo de la relación en el diagrama Entidad-Relación puede representar una dimensión temporal, para recoger información puntual o histórica. En el caso de recoger información histórica, alguno o todos esos atributos deberán formar parte de la clave primaria de la nueva relación en el grafo relacional. Se explicará este caso particular con dos ejemplos.

Primer ejemplo: los atributos *fechaPrestamo* y *fechaDevolucion* no son multivaluados y recogen la fecha en la que se prestó un ejemplar a un usuario y la fecha en la que fue devuelto por el usuario; de esta manera, sólo queda registrada información de la última vez que un usuario llevó prestado un ejemplar. En el grafo relacional que se obtiene, se ve que en la relación *PRESTA*, la clave primaria está compuesta por los atributos *codUsuario* y *codEjemplar*. Por lo tanto, sólo permite una tupla para cada combinación de valores de esos atributos.

### MER

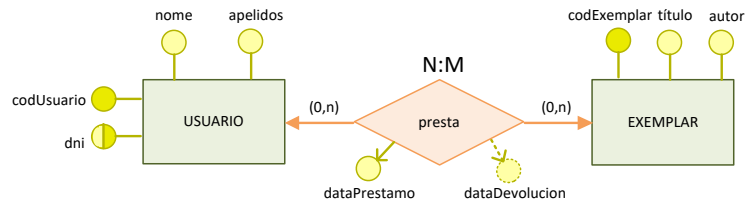


### Esquema MR

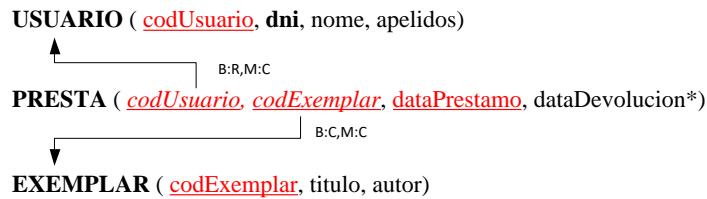


Segundo ejemplo: los atributos *fechaPrestamo* y *fechaDevolucion* son multivaluados, por lo que pueden recoger información histórica de todos los préstamos con sus fechas de préstamo y de devolución. En la transformación, el atributo que no es opcional pasa a formar parte de la clave primaria de la nueva relación del grafo relacional. El atributo *fechaDevolucion* no forma parte de la clave primaria ya que no es necesaria (no sería mínima) y además puede admitir valores nulos, lo que impediría que haya formado parte de la clave primaria.

### MER

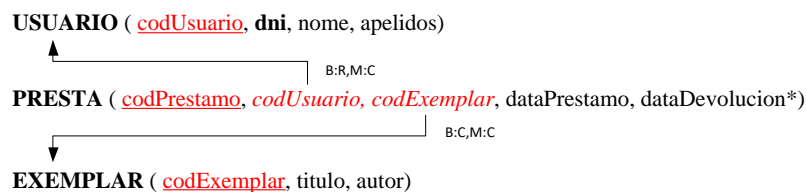


### Esquema MR



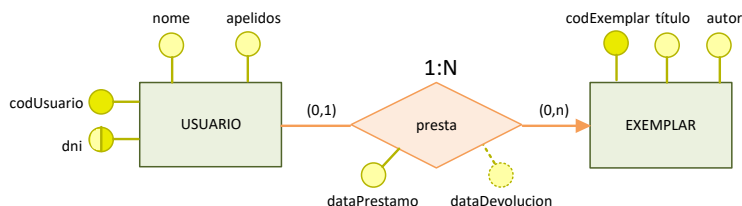
Conviene recordar que al explicar la manera de representar a la dimensión temporal en el modelo Entidad-Relación, se recomendaba cambiar una relación que tuviera varios atributos por una entidad y así asociarle varias características. También conviene recordar que no es recomendable utilizar claves primarias muy largas, por lo que en el ejemplo anterior, se puede optar por añadir un atributo nuevo en la relación *PRESTA* que sea clave primaria, como se muestra en el siguiente grafo relacional.

### Esquema MR

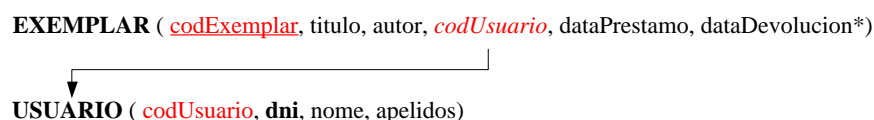


- Si los atributos pertenecen a una relación de tipo 1:N o cualquier otra relación que se transforme mediante la propagación de clave, se pueden incluir junto con la clave foránea en la relación que recibe la propagación. No es recomendable utilizar atributos en relaciones que no sean de tipo N:M, ya que podrían considerarse atributos de una de las entidades que participan en la relación. Por ejemplo, en el siguiente diagrama un ejemplar sólo puede ser prestado a un usuario. Sólo se guarda información del último usuario al que se le presta un ejemplar y no se recoge la información de los otros usuarios a los que se les prestó.

### MER

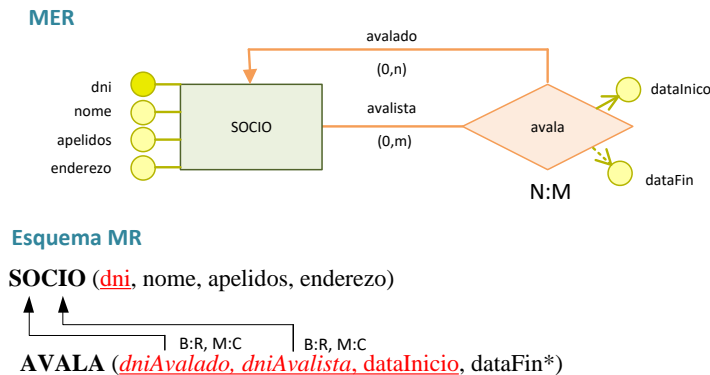


### Esquema MR



En el caso de que el atributo de la relación del diagrama Entidad-Relación fuese multivaluado, se trata como cualquier otro atributo multivaluado de la relación, que resulta de la transformación de la entidad que lleva la clave foránea.

La transformación de atributos de relaciones reflexivas, se realiza de la misma manera que en el resto de relaciones. Por ejemplo, el siguiente diagrama Entidad-Relación permite guardar información de los socios que fueron avalados por un socio y del período de tiempo en el que avaló a cada uno; permite también que un socio pueda avalar a varios socios al mismo tiempo o en distintos períodos de tiempo. La transformación en grafo relacional sigue las mismas pautas que si fuese una relación binaria.



## 2.2.2. Transformación del modelo Entidad-Relación extendido (MERE) a modelo Relacional (MR)

### 2.2.2.1. Regla 4. Transformación de entidades débiles

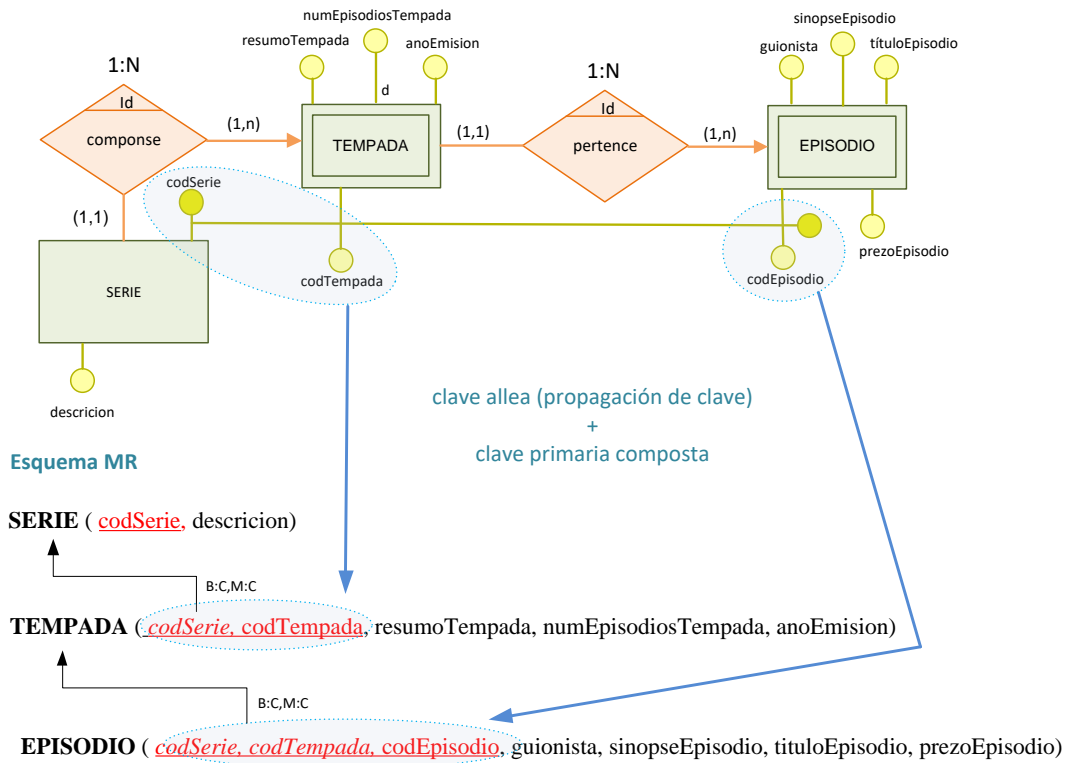
Una entidad débil del modelo Entidad-Relación se transforma en una relación en el grafo relacional, que se denominará con el mismo nombre que la entidad de la que proviene. Conservará sus atributos y tendrá una nueva clave foránea que hará referencia a la clave primaria de la entidad fuerte de la que depende.

En el caso de debilidad por identificación, la clave primaria de la relación creada será una clave compuesta formada por la clave foránea más uno o más atributos de la entidad débil.

En el caso de debilidad por existencia, la entidad débil ya tiene una clave primaria fuerte que será la clave primaria de la relación creada.

Para mantener la restricción de integridad asociada a la clave foránea es necesario establecer una política de comportamiento en la que las operaciones de borrado y modificación en la entidad fuerte obliguen a una modificación y borrado en cascada en la entidad débil.

Por ejemplo, en el siguiente diagrama Entidad-Relación, la entidad *TEMPORADA* depende por identificación de la entidad *SERIE* y la entidad *EPISODIO* depende por identificación de la entidad *TEMPORADA*; esto da lugar a que en el grafo relacional, la relación *TEMPORADA* necesite para su identificación del atributo de identificación principal de la relación *SERIE* y que la relación *EPISODIO* necesite para su identificación de los atributos de identificación principal de la relación *TEMPORADA*, lo que daría lugar al siguiente grafo relacional:



### 2.2.2.2. Regla 5. Transformación de jerarquías

Se describen tres maneras de transformar las jerarquías (especialización-generalización) representadas en un diagrama Entidad-Relación, teniendo en cuenta algunas consideraciones como las pérdidas semánticas, los tipos de jerarquías y cuestiones de rendimiento.

- Crear una única relación que tendrá los atributos de la entidad de nivel superior (super-tipo), los atributos de los subtipos (si los tuvieran) que deben permitir valores nulos y además, un atributo discriminante que hace referencia al subtipo que corresponde a cada tupla.

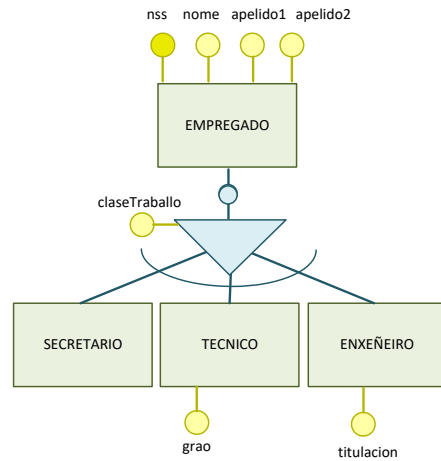
Esta solución es recomendable cuando los subtipos se diferencian entre sí en muy pocos atributos y cuando las relaciones que asocian los distintos subtipos con el resto de las entidades se pueden unificar para todos los subtipos.

En lo que se refiere a la eficiencia, esta opción es la que ofrece más velocidad en el acceso a los datos de un objeto, ya que no es preciso efectuar uniones de tablas para la recuperación de la información.

También puede contemplarse la posibilidad de no incluir el atributo discriminante en la relación, y deducir al subtipo o subtipos a partir de la existencia o no de los valores de los atributos pertenecientes a dichas entidades subtipo. Esta solución no se aconseja ya que aunque reduce el número de atributos de la relación, aumenta las operaciones necesarias para obtener el subtipo o subtipos a los que pertenece una ocurrencia.

Por ejemplo, en el siguiente diagrama E-R, hay una jerarquía (especialización) total y disjunta, atributos comunes a todos los empleados (*nss*, *nombre*, *apellido1* y *apellido2*) y muy pocos específicos de cada clase de trabajo (*grado* para los que trabajan como técnicos y la *titulación* para los ingenieros). Por ser una jerarquía total, el atributo *claseTrabajo* no puede admitir valores nulos ya que los empleados sólo pueden ser uno de los tres subtipos y su dominio estaría formado por el conjunto {secretario, técnico, ingeniero}. Por ser una jerarquía disjunta, los atributos *grado* y *titulación* no pueden tener valores distintos de nulo en la misma tupla y pueden ser ambos nulos en el caso de que el empleado sea secretario.

## MER



## Esquema MR

**EMPREGADO** ( nss, nome, apelido1, apelido2, claseTraballo, grao\*, titulacion\*)

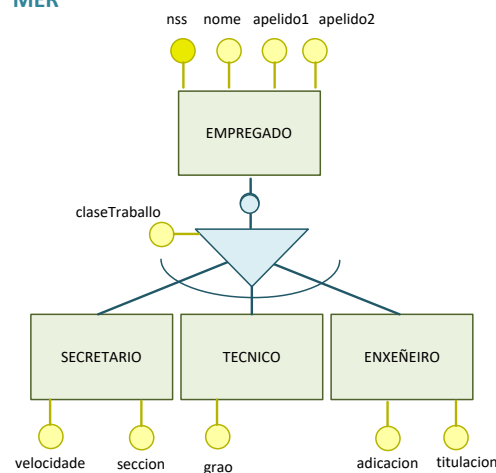
- Crear una relación para el supertipo con sus atributos y una relación para cada uno de los subtipos haciendo propagación de clave, de manera que tendrían los atributos propios del subtipo más la clave primaria del supertipo, que haría de clave primaria de cada relación subtipo.

Es la solución más adecuada y la más flexible, aplicable a todo tipo de jerarquía y que permite recoger más semántica. Es la más respetuosa con el modelo conceptual original, aunque es preciso crear las restricciones oportunas para que se refleje de forma fehaciente la semántica original del modelo conceptual de datos.

En cuanto al rendimiento, esta solución ofrece menos velocidad que la primera en el acceso a los datos de una relación, puesto que para la recuperación de la información es preciso unir varias relaciones. Asimismo, es la que más espacio ocupa, ya que tener atributos en una única relación siempre ocupa menos que tenerlos repartidos o repetidos en tablas diferentes.

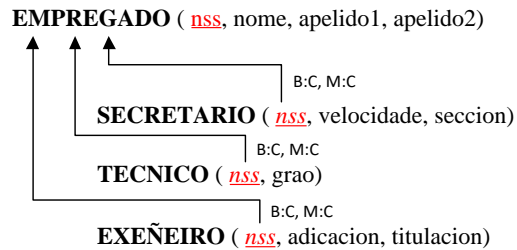
Por ejemplo, modificando el ejemplo anterior para que los subtipos *SECRETARIO*, *TECNICO* e *INGENIERO* tengan más atributos propios.

## MER





### Esquema MR



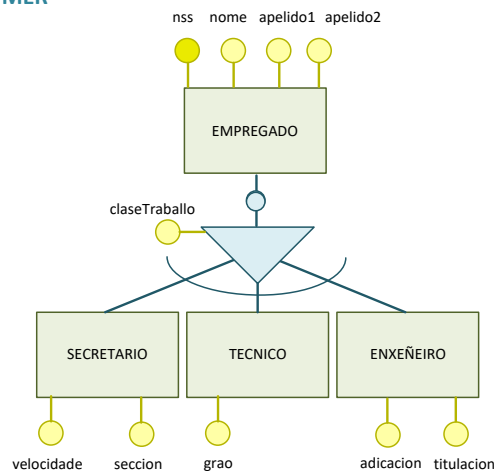
- Crear una relación sólo para los subtipos con los atributos propios del subtipo y, además, los atributos del supertipo. La clave primaria de las relaciones sería la del supertipo.

Es la opción menos recomendable y se aplica sólo cuando se cumplen las siguientes condiciones:

- Los subtipos disponen de un elevado número de atributos, y/o relaciones propias con otras entidades.
- Los accesos realizados a los datos de los subtipos afectan mayoritariamente a los atributos comunes.
- La jerarquía es exclusiva o disjunta, ya que de ser solapada los atributos comunes podrían estar repetidos en más de una tupla, dando lugar a una redundancia que habría que controlar para evitar inconsistencias.
- La jerarquía es total, ya que de ser parcial habría que crear una nueva tabla con los atributos del supertipo para aquellos empleados que no pertenecen a ningún subtipo.

Por ejemplo, usando el ejemplo anterior:

### MER



### Esquema MR

**SECRETARIO** ( nss, nome, apelido1, apelido2, velocidade, seccion)

**TECNICO** ( nss, nome, apelido1, apelido2, grao)

**EXEÑEIRO** ( nss, nome, apelido1, apelido2, adicacion, titulacion)

### 2.2.3. Documentación de las restricciones que no se pueden plasmar en el esquema lógico relacional

Algunas restricciones del diagrama Entidad-Relación no pueden ser representadas en el esquema lógico en el momento de hacer la transformación y deben detallarse en un documento que acompañará al esquema lógico. No existe un modelo estándar para este documento y sólo se pide que recoja todas las restricciones de forma clara y precisa. Esta documentación se utilizará en la creación de la base de datos empleando un SGBDR concreto, en la que mediante una serie de cláusulas del lenguaje SQL que se estudiarán en posteriores unidades, se asegurará el cumplimiento de las restricciones.

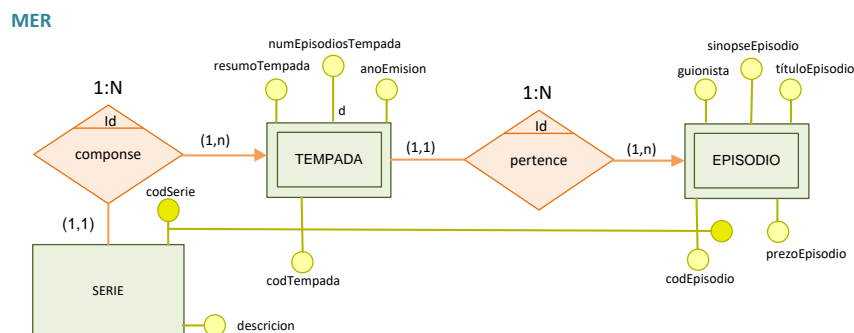
#### 2.2.3.1. Restricciones de dominios

Algunos atributos presentan restricciones de los valores que son válidos y constituyen su dominio. Los dominios no suelen representarse en el modelo Entidad-Relación pero pueden ser deducidos de la información recogida en el diccionario de datos. Por otra parte, forman parte del modelo relacional estándar pero no son representados en el grafo relacional. Por ejemplo, se podría suponer que dominio del atributo edad está formado por los números entre 18 y 80, ambos incluidos.

#### 2.2.3.2. Restricciones de atributos derivados

Los atributos derivados son aquéllos en los que su valor se calcula a partir de otros atributos de entidades o relaciones o del recuento de sus ocurrencias. Tienen representación gráfica en el modelo Entidad-Relación y pueden incluirse o no en el modelo relacional, en función del uso que van a tener o por cuestiones de eficiencia; en caso de incluirlos, se tratan como atributos normales dando lugar a un atributo de la relación a la que pertenecen. En el momento de la creación de la base de datos, se debe programar una rutina que actualice el atributo cada vez que se inserta o se borra una fila en la tabla *EPISODIO*. Este tipo de rutinas se conocen con el nombre de disparadores o triggers y se estudiarán en unidades posteriores.

Por ejemplo, *numEpisodiosTemporada* se puede calcular a partir del número de ocurrencias de la entidad *EPISODIO* relacionados con una *TEMPORADA*. Si el dato se va a utilizar con mucha frecuencia y su cálculo hace más lentos los procesos que lo manejan, se puede decidir incluirlo como un atributo más; en este caso, se debería de crear un procedimiento para que el SGBDR lo mantenga actualizado permanentemente.



#### Esquema MR

```
SERIE ( codSerie, descripcion )
    ↑
    B:C,M:C
TEMPADA ( codSerie, codTempada, resumoTempada, numEpisodiosTempada, anoEmission )
    ↑
    B:C,M:C
EPISODIO ( codSerie, codTempada, codEpisodio, guionista, sinopseEpisodio, tituloEpisodio, prezoEpisodio )
```

### 2.2.3.3. Restricciones en la transformación de jerarquías (especialización/generalización)

Las restricciones aplicadas en la transformación de jerarquías, relacionadas con la jerarquía total y/o disjunta, que no se pueden plasmar en el modelo relacional, están expresadas en la siguiente tabla y dependerán de la solución adoptada en la transformación.

Solución de transformación	Restricción de jerarquía total	Restricción de jerarquía disjunta
Crear una sola relación con los atributos del supertipo, un atributo discriminante para saber a que subtipo pertenece cada tupla y los atributos propios de los subtipos.	El atributo no discriminante puede tomar el valor nulo.	Sólo los atributos de un subtipo pueden tener valores distintos de nulo para cada tupla.
Crear una relación para el supertipo y una para cada uno de los subtipos.	No existe ninguna tupla en la tabla supertipo que no esté relacionada con una tupla de una tabla subtipo.	No permitir insertar nuevas tuplas en la tabla supertipo, sino que se insertan automáticamente cada vez que se inserta una tupla en una tabla subtipo. De esta manera, cuando se intenta insertar una tupla en una tabla subtipo con la misma clave primaria de otra que ya existe en otra tabla subtipo, ocurrirá un error al intentar insertar una tupla en la tabla supertipo con una clave duplicada.
Crear una relación para cada subtipo.	No es necesario.	No permitir insertar una tupla en una tabla subtipo con una clave primaria que existe en alguna de las otras tablas subtipo.

Este tipo de restricciones se resuelven en la creación de la base de datos utilizando aserciones, que son procedimientos que permiten validar cualquier atributo de una o varias tablas y que se explicarán en unidades posteriores.

### 2.2.3.4. Restricciones asociadas a la inclusividad y exclusividad

Las restricciones de exclusividad, inclusividad, exclusión e inclusión de los diagramas Entidad-Relación no pueden ser plasmadas en el grafo relacional, pero deben ser documentadas a fin de que en el momento de crear la base de datos con un SGBDR se realicen las comprobaciones necesarias creando módulos programados en la base de datos a tal fin.

Por ejemplo, la restricción de inclusión del siguiente diagrama Entidad-Relación representa que para que un investigador pueda solicitar un proyecto, debió participar con anterioridad en otro, que no tiene que ser el que solicitó. Habrá que comprobar, antes de insertar una fila en la tabla *SOLICITA*, que exista alguna fila en la tabla *PARTICIPA* para ese investigador; en caso contrario, no se permitirá la inserción.

