



Unidad Didáctica 2:

ELABORACIÓN DEL DISEÑO  
CONCEPTUAL:

Modelo Entidad-Relación

## 1. Introducción

El **Diseño de Bases de Datos** es el proceso por el que se determina la organización de una Base de Datos, incluidas su estructura, contenido y las aplicaciones que se han de desarrollar.

El diseño de una Base de Datos es un proceso complejo que abarca decisiones a muy distintos niveles. La complejidad se controla mejor si se descompone el problema en subproblemas y se resuelve cada uno de estos subproblemas independientemente, utilizando técnicas específicas. Así, el diseño de una Base de Datos se descompone en: **diseño conceptual, diseño lógico y diseño físico**. En cada una de estas fases se utiliza el modelo de datos correspondiente.

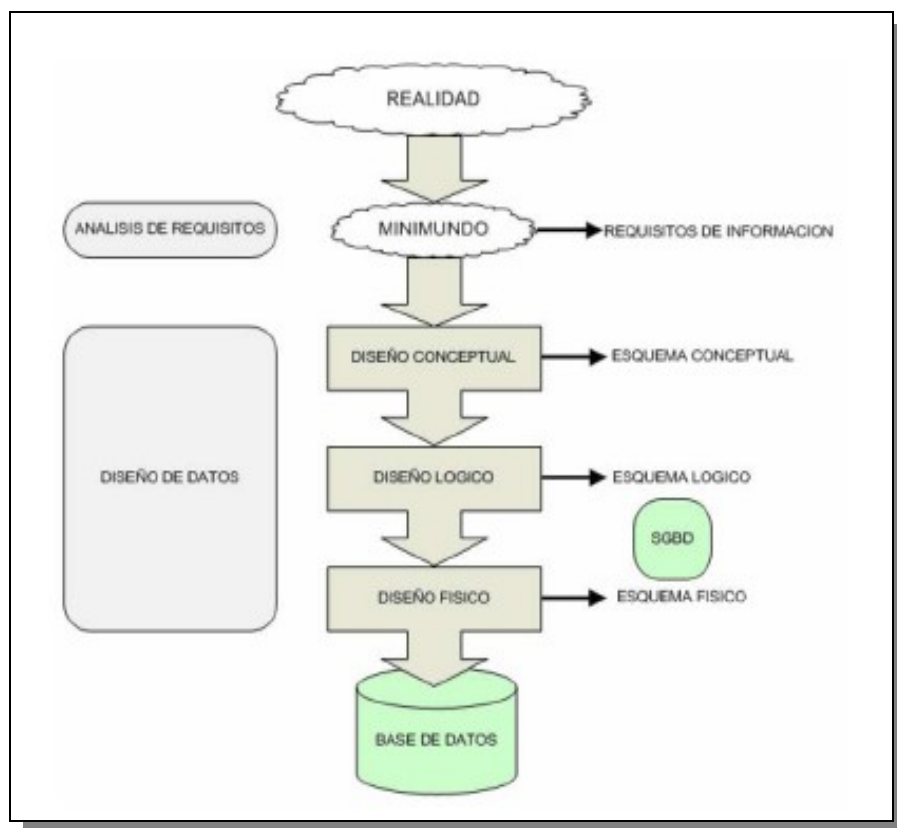


Figura 1. Fases del diseño de una BD

## 2. Fases del diseño de una Base de Datos

El **diseño conceptual** parte de las especificaciones de requisitos de usuario y su resultado es el esquema conceptual de la Base de Datos. Un esquema conceptual es una descripción de alto nivel de la estructura de la Base de Datos, independientemente del SGBD que se vaya a utilizar para manipularlo. Los procesos de definición de requisitos y del diseño conceptual exigen identificar las exigencias de información de los usuarios y representarlos en un modelo bien definido. Diseñaremos el esquema conceptual mediante el **modelo Entidad-Relación (MER)**.

Ejemplo de requisitos de la BD EMPRESA:

La empresa está organizada en **departamentos**. Cada uno con *nombre* y *código* único e interesa saber el *número* de *empleados* que trabaja en él. El departamento está localizado en una *localidad*.

Cada departamento controla cierto número de **proyectos**. Cada uno con *nombre* y *código* únicos. Interesa saber la *fecha de inicio* del proyecto.

De cada **empleado** se guarda su *nombre*, *número de seguridad social*, *dirección*, *salario*, *sexo* y *fecha de nacimiento*. También necesitamos saber el *departamento* al que pertenece, *proyectos* en los que trabaja (pueden estar controlados por departamentos distintos al que pertenece el empleado), *nº horas semanales* dedicadas a cada proyecto y quién es su *supervisor*.

...

Se obtiene el Esquema Conceptual mediante el MER:

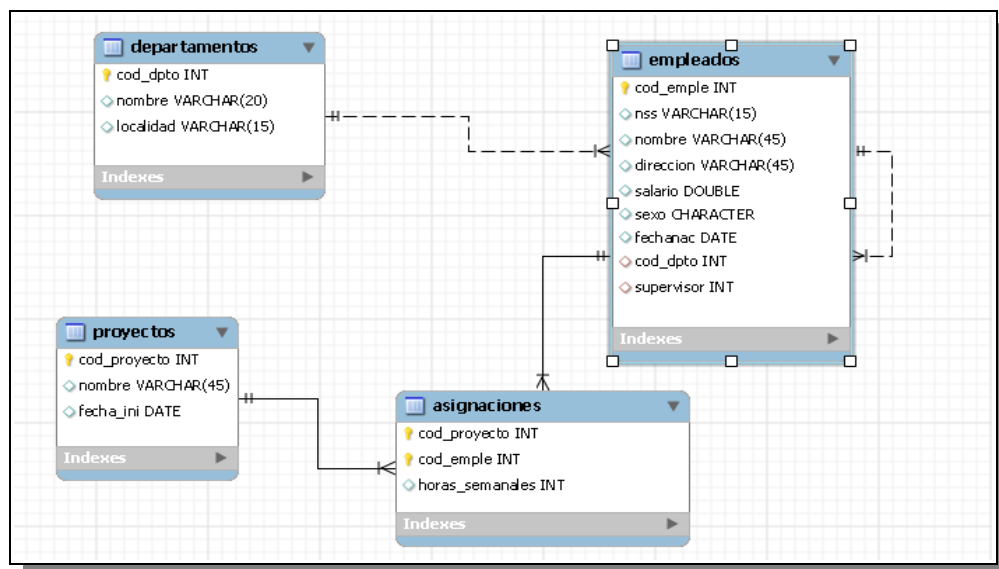


Figura 2. MER de la BD EMPRESA . Herramienta Workbench.

El **diseño lógico** es el proceso de construir un esquema de la información que utiliza la empresa, basándose en un modelo conceptual de base de datos específico, independiente del SGBD concreto que se vaya a utilizar (salvo en el modelo) y de cualquier otra consideración física.

En esta etapa, se transforma el esquema conceptual en un esquema lógico que utilizará las estructuras de datos del modelo de base de datos en el que se basa el SGBD que se vaya a utilizar, como puede ser el modelo relacional, el modelo en red, el modelo jerárquico o el modelo orientado a objetos.

Conforme se va desarrollando el esquema lógico, éste se va probando y validando con los requisitos de usuario. La *normalización* es una técnica que se utiliza para comprobar la validez de los esquemas lógicos basados en el modelo relacional, ya que asegura que las relaciones (tablas) obtenidas no tienen datos redundantes.

El Esquema Lógico sería:

DEPARTAMENTOS ( cod\_dpto, nombre, localidad)

EMPLEADOS ( cod\_emple, nss, nombre, direccion, salario, sexo,  
fechanac, cod\_dpto, supervisor)

ASIGNACIONES ( cod\_proyecto, cod\_emple, horas\_semanales)

PROYECTOS ( cod\_proyecto, nombre, fecha\_ini)

El **diseño físico** es el proceso de producir la descripción de la implementación de la base de datos en memoria secundaria: estructuras de almacenamiento y métodos de acceso que garanticen un acceso eficiente a los datos. Para llevar a cabo esta etapa, se debe haber decidido cuál es el SGBD que se va a utilizar, ya que el esquema físico se adapta a él. Entre el diseño físico y el diseño lógico hay una realimentación, ya que algunas de las decisiones que se tomen durante el diseño físico para mejorar las prestaciones, pueden afectar a la estructura del esquema lógico.

En general, el propósito del diseño físico es describir cómo se va a implementar físicamente el esquema lógico obtenido en la fase anterior. Concretamente, en el modelo relacional, esto consiste en:

- Obtener un conjunto de relaciones (tablas) y las restricciones que se deben cumplir sobre ellas.
- Determinar las estructuras de almacenamiento y los métodos de acceso que se van a utilizar para conseguir unas prestaciones óptimas.
- Diseñar el modelo de seguridad del sistema.

## 3. Modelo Entidad Relación

### 3.1. Introducción

Fue ideado por Peter Chen en los años 1976 y 1977 a través de dos artículos. Se trata de un modelo que sirve para crear esquemas conceptuales de bases de datos. De hecho es prácticamente un estándar para crear esta tarea.

Se le llama modelo E/R (entity-relationship) e incluso EI (Entidad / Interrelación). Sus siglas más populares son las E/R por que sirven para el inglés y el español. Inicialmente (en la propuesta de Chen) sólo se incluían los conceptos de entidad, relación y atributos. Después se añadieron otras propuestas (atributos compuestos, generalizaciones,...) que forman el llamado modelo entidad relación extendido (se conoce con las siglas ERE).

Los elementos básicos del modelo E-R original son:

- ENTIDAD (*entity*).
- ATRIBUTO (*attribute*).
- DOMINIO (*domain*).
- RELACION (*relationship*).

### 3.2. Entidades

Se denomina entidad a cualquier objeto (real o abstracto) que existe en la realidad y acerca del cual queremos almacenar información en la B.D.

Pueden ser objeto con existencia física o real (una persona, un libro, un empleado) o abstracta o conceptual (una asignatura, un viaje).

*"Persona, lugar, cosa, concepto o suceso, real o abstracto, de interés para la empresa"* (ANSI, 1977).

Las entidades poseen un predicado asociado que hace que los ejemplares lo cumplen. El modelo ER proporciona una notación diagramática para representar gráficamente las

entidades : las entidades se representan con un rectángulo. El nombre de la entidad (sustantivo) se escribe en mayúsculas dentro del rectángulo.

Llamaremos *tipo de entidad* a la estructura genérica y *ocurrencia* o *ejemplar* de entidad a cada una de las realizaciones concretas de ese tipo de entidad.

Ejemplo: La entidad **PROFESOR**, cuyo predicado asociado es "persona que enseña una materia", tiene un ejemplar 'Juana' que pertenece a ese tipo de entidad, ya que cumple dicho predicado (o al menos lo intenta). De este modo hablaríamos de la entidad **PROFESOR**. Mientras que cada persona en concreto sería una **ocurrencia** o un **ejemplar** de la entidad **PROFESOR**.



**PROFESOR**

Existen 2 clases de entidades:

- **Regulares (fuertes).** Son las entidades normales que tienen existencia por sí mismas sin depender de otras. Su representación gráfica es la que ya hemos visto.
- **Débiles.** Su existencia depende de otras. Por ejemplo si se desea almacenar la información de las copias de películas que hay en un videoclub, la entidad utilizada para ello será una entidad débil, ya que ésta depende de la entidad PELICULAS, puesto que si desaparece una película de nuestra BD, también desaparecerán todas sus copias.

Su representación gráfica es un rectángulo doble:



**COPIAS**

Entre las entidades débiles y las entidades fuertes existe dos tipos de dependencia:

- **En existencia:** Si desaparece una instancia del tipo de entidad regular deben desaparecer las instancias de la entidad débil que dependen de ella.
- Dependencia **en identificación:** Una instancia del tipo de entidad débil no se puede identificar por sí misma, es decir, nos hace falta la clave de la entidad regular de la que depende. Su clave es (clave\_entidad\_regular, clave\_parcial).

COPIAS es una entidad débil, pues depende en existencia de PELICULAS, y también depende en identificación.

### PERMISOS CONDUCCION

PERMISO\_CONDUCCION es débil, pues depende en existencia de EMPLEADOS, pero no depende en identificación, pues tiene clave primaria propia.

## 3.3. Atributos

Se denomina *atributo* a cada una de las propiedades o características que posee una entidad y de la que se desea guardar información. Un atributo debe tener un nombre significativo que debe ser único dentro de esa entidad.

Los diagramas entidad-atributo permiten presentar de forma simbólica los atributos de una entidad. Esta forma de representación varía de unos modelos a otros, siendo el soportado por *Designer/2000* el que nosotros utilizaremos.

Ejemplos de atributos:

Sobre una entidad EMPLEADO nos puede interesar, por ejemplo, tener registrados su DNI, su NSS, su nombre, su apellido y su sueldo como atributos.

### EMPLEADO

# dni  
\* nss  
\* nombre  
\* apellidos  
o sueldo

- \* Significa obligatorio.
- # Significa obligatorio y es (o es parte de) un identificador único o clave.
- o Significa opcional.

Los atributos se pueden clasificar en:

### Atributos Simples o Compuestos:

- Los **atributos simples** son atómicos y, por lo tanto, no pueden dividirse en otras unidades. Por ejemplo, el *salario* de un empleado.
- Los **atributos compuestos** pueden dividirse en otros con significado propio. Por ejemplo *fechanac* puede dividirse en *día*, *mes* y *año*.

### Atributos Almacenados o Derivados:

- Los **atributos derivados** son atributos cuyo valor se obtiene aplicando una fórmula (normalmente a partir del valor de otros atributos). Son atributos que no se almacenarán en la base de datos. Su valor se obtendrá en el momento en que sea necesario aplicando la fórmula asociada a ellos. En el diccionario de datos debe especificarse esta fórmula o método para calcular su valor. Son información redundante.

Como ejemplo tenemos el atributo *edad* de EMPLEADO, puesto que se puede calcular a partir de *fechanac*. Otro ejemplo sería *numcopias* de una PELICULA, puesto que podríamos calcularlo contando el número de ocurrencias de una determinada película que hay en la entidad COPIA, entidad relacionada con la entidad PELICULA.

- Los **atributos almacenados** son atributos que no se pueden obtener a partir de otros y, por lo tanto, tienen que estar almacenados en la base de datos. No son redundantes. Como ejemplos tenemos la *fechanac* de cada EMPLEADO o la *nacionalidad* de una PELICULA.

### Atributos Monovalorados o Multivalorados

- Los **atributos monovalorados** (monovaluados) son los que admiten sólo un valor para cada ocurrencia de una entidad. Por ejemplo, la *fechanac* de un EMPLEADO o el *añoestreno* de una PELICULA.
- Los **atributos multivalorados** (multivaluados) son los que admiten más de un valor para una ocurrencia de una entidad. Por ejemplo, la *nacionalidad* de una PELICULA coproducida por varios países o el *teléfono* de EMPLEADO cuando un empleado tiene varios teléfonos de contacto.



## Atributos Opcionales u Obligatorios

- Los **atributos opcionales** son atributos que admiten valores **nulos** (null value) y que son usados cuando:
  - Se desconoce el valor de un atributo para cierta ocurrencia de una entidad:
    - El valor existe pero falta, por ejemplo la *altura* de un EMPLEADO (todos los empleados tienen altura pero, en algún caso, la desconocemos).
    - No se sabe si el valor existe o no. Por ejemplo, el *teléfono* de un EMPLEADO (desconozco si hay algún empleado sin teléfono).
  - La entidad no tiene ningún valor aplicable para el atributo. Por ejemplo, *piso* del atributo *dirección* de la entidad EMPLEADO, no hay porque es una vivienda unifamiliar.
- Los **atributos obligatorios** son los que no admiten valores nulos. Por ejemplo, el *dni* de EMPLEADO es un atributo obligatorio (todos los empleados de la empresa tienen DNI y además es un atributo que me sirve para distinguir un empleado de otro).

## Atributo clave

Es un atributo con valor distinto para cada ocurrencia de un tipo de entidad y que identifica de forma unívoca a cada una de ellas. Por ejemplo, el *dni* en la EMPLEADO o el atributo *matricula* de la entidad COCHE. Son atributos que no admiten valores nulos.

Una clave puede estar formada por varios atributos. En este caso sería una **clave compuesta**, por ejemplo (*nombre, fechanac*) en el tipo de entidad EMPLEADO. Una clave compuesta debe ser mínima.

Un tipo de entidad puede tener más de una clave. Al los atributos que pueden ser clave en una entidad se les denomina **claves candidatas**. De los atributos que son claves candidatas se elige uno que se denominará **identificador principal** o **clave principal** y el resto serán **identificadores alternativos** o **claves alternativas**.

Ejemplo

**Claves o Identificadores Candidatos** de EMPLEADO:

- *dni*
- *nss*
- (*nombre, apellidos y fechanacim*)

**Clave Principal**, elegida (por el diseñador) de entre los identificadores candidatos, para ser el medio principal de identificación de las instancias del tipo de entidad.

- *dni*

Atributos **identificadores alternativos**,

- *nss*
- (*nombre, apellidos, fechanacim*)

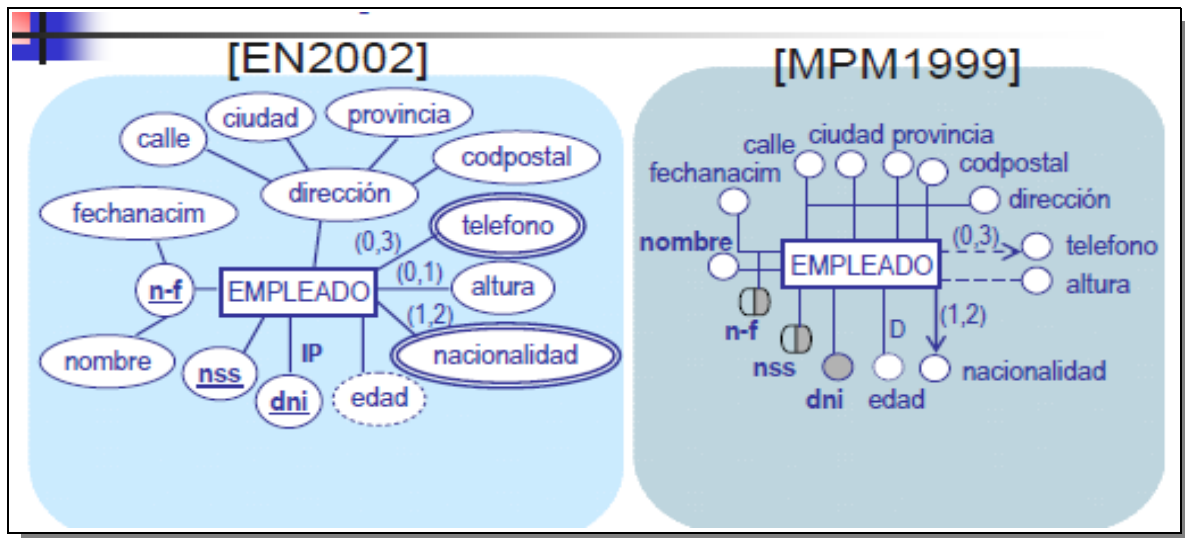


Figura 3. Representación de entidades y atributos en la notación de De Miguel, Piattini y Marcos y de Elmasri&Navathe

### 3.4. Dominio (values set)

Es el conjunto de posibles valores que puede tomar un atributo (su tipo de datos, en términos de lenguajes de programación). Cada atributo simple está asociado a un dominio, que especifica sus valores válidos. Por ejemplo, el dominio del atributo *DNI* sería los números enteros de 8 cifras.

Más de un atributo pueden compartir el mismo dominio. Por ejemplo, si incluimos el atributo *fechaincorporación* a la compañía en la entidad *EMPLEADO*, tiene el mismo dominio que *fechanac*.

Los dominios se especificarán en el diccionario de datos. Es obligatoria la especificación del nombre del dominio, el tipo y la descripción.

Atributo	Dominio	Descripción Dominio
nombre	NOMBRES	cadenas de hasta 30 caracteres alfabéticos
telefono	TELEFONOS	cadenas de hasta 9 caracteres numéricos
altura	MEDIDAS	números reales entre 0 y 2'5 (metros)
...	...	...

Figura 4. Ejemplos de posibles dominios para varios atributos

### 3.5. Relaciones (Interrelaciones)

Las relaciones representan asociaciones entre entidades. Es el elemento del modelo que permite relacionar en sí los datos del modelo. Por ejemplo, en el caso de que tengamos una entidad personas y otra entidad trabajos. Ambas se realizan ya que las personas trabajan y los trabajos son realizados por personas.

- el director "**Alejandro Amenábar**" ha rodado la película "**Mar adentro**".
- el empleado **87654321** trabaja en el local de videoclub "**principal**".
- la película "**El imperio contraataca**" es una continuación de la película "**La guerra de las galaxias**".

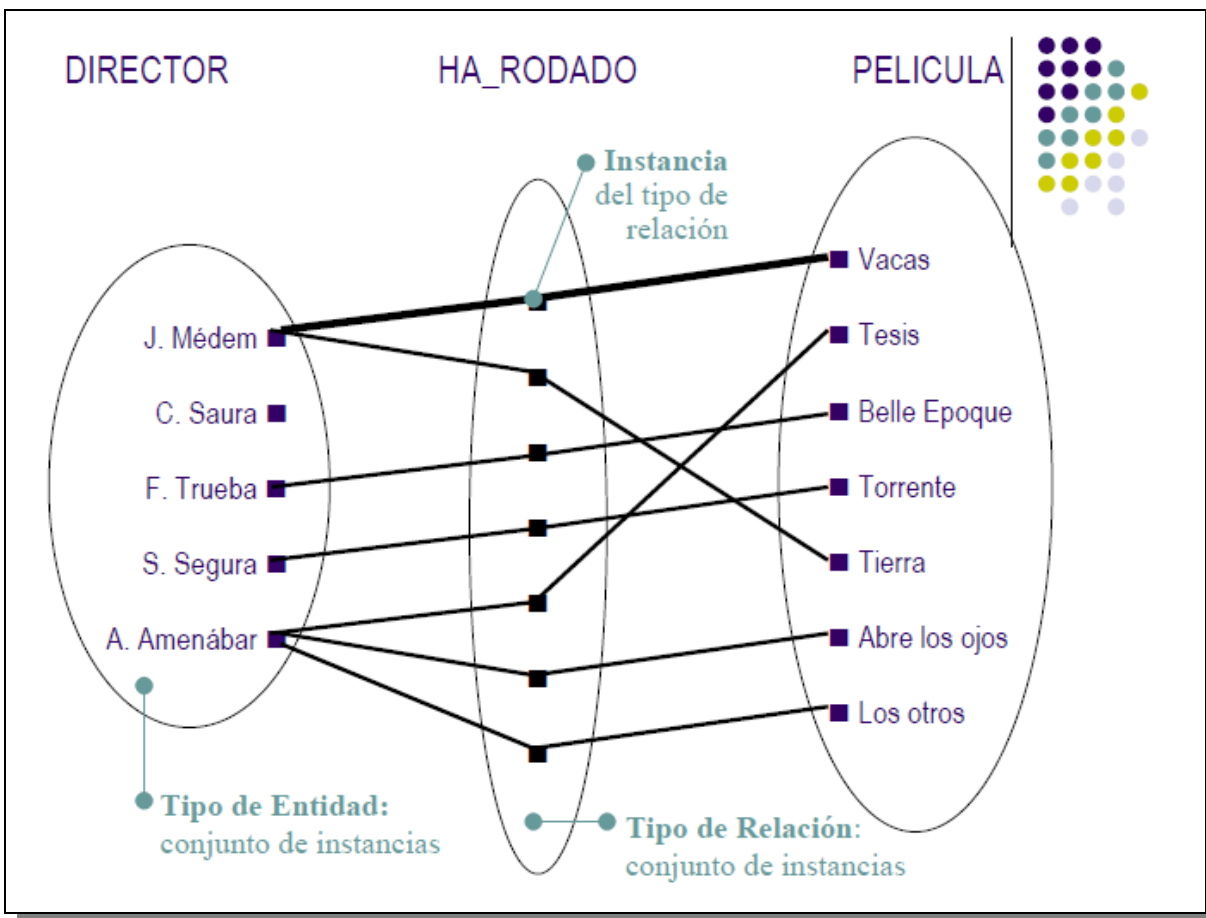


Figura 5. Ejemplo de relación

En este caso, la representación gráfica de esta relación sería:



A veces en las líneas de la relación se indican **roles**. Los roles representan el papel que juega una entidad en una determinada relación. Normalmente son verbos o formas verbales.

### 3.5.1. Grado de una relación

El número de entidades que participan en la interrelación se denomina grado.

Las interrelaciones de grado dos se denominan también **interrelaciones binarias**. Todas las interrelaciones de grado mayor que dos se denominan, en conjunto, interrelaciones n-arias. Así pues, una interrelación n-aria puede tener grado tres y ser una **interrelación ternaria**, puede tener grado cuatro y ser una **interrelación cuaternaria**, etc. Las más habituales son las relaciones binarias.

Cuando interviene una sola entidad la relación es **reflexiva** o **recursiva**.

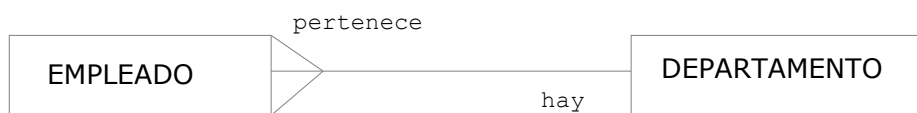


Figura 6. Ejemplo de relación binaria

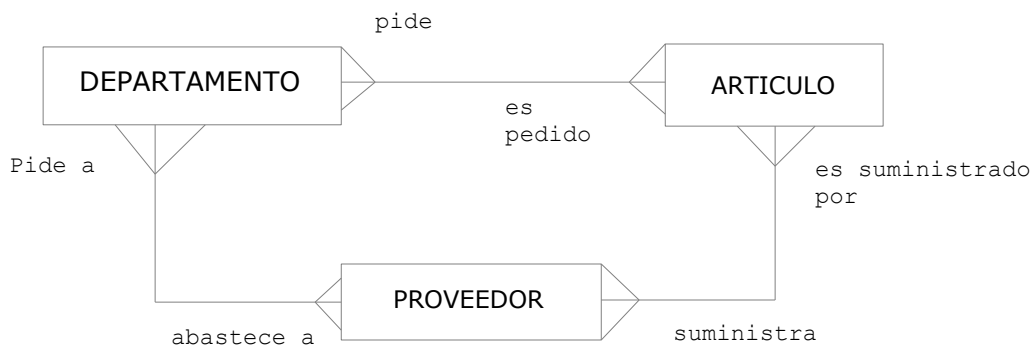


Figura 7. Ejemplo de relación ternaria



Figura 8. Ejemplo de relación recursiva (los empleados tienen un supervisor que también es empleado de la compañía)

### 3.5.2. Cardinalidad de una relación

La cardinalidad de una relación expresa el tipo de correspondencia que se establece entre las ocurrencias de entidades asociadas con la relación. En el caso de las relaciones binarias, expresa el número de ocurrencias de una de las entidades con las que una ocurrencia de la otra entidad puede estar asociada según la relación.

Podemos tener tres tipos de cardinalidad:

- **Cardinalidad uno a uno (1:1).** A cada ocurrencia de la entidad A le corresponde no más de una ocurrencia de la entidad B y viceversa. Por ejemplo, cada persona posee un historial clínico y cada historial clínico es de una única persona.
- **Cardinalidad uno a muchos (1:N).** A cada ocurrencia de la entidad A le pueden corresponder varias ocurrencias de la entidad B pero a cada ocurrencia de la entidad B le corresponde como máximo una ocurrencia de la entidad A. Un ejemplo: banco tiene muchas sucursales pero cada sucursal pertenece a un banco.
- **Cardinalidad muchos a muchos: (M:N).** A cada ocurrencia de la entidad A le pueden corresponder varias ocurrencias de la entidad B. Y a cada ocurrencia de la entidad B le pueden corresponder varias ocurrencias de la entidad A. Un ejemplo: un alumno está matriculado de varias asignaturas y en cada asignatura hay varios alumnos matriculados.

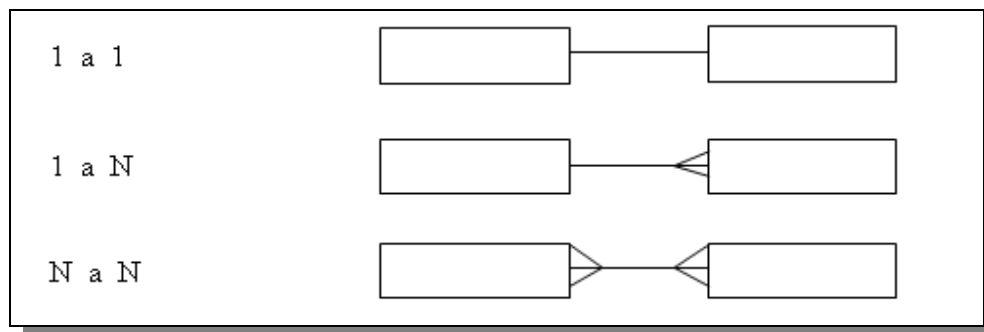


Figura 9. Cardinalidad de una relación

### 3.5.3. Participación de las entidades en las relaciones

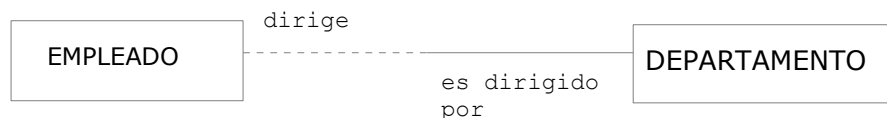
Es la participación obligatoria u opcional de la entidad en la relación. Esto responde al hecho de que a veces existen ocurrencias de una entidad que pueden no tomar parte en una relación como otras ocurrencias lo hacen.

Hay dos clases de participación:

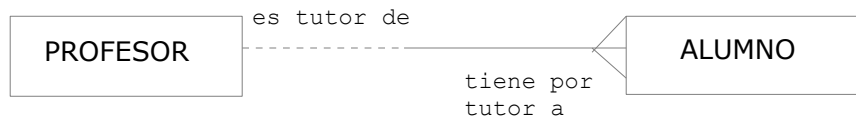
- **Participación total** (dependencia en existencia) o relación obligatoria, significa que toda ocurrencia de una entidad debe estar asociada a una ocurrencia de la otra entidad.
- **Participación parcial** o relación opcional, significa que puede haber ocurrencias de una entidad que no estén asociadas a ninguna ocurrencia de la otra entidad.

Las figuras siguientes nos servirá para entender el significado práctico de la dependencia de existencia u opcionalidad.

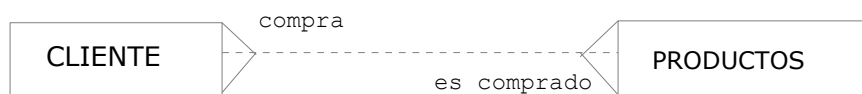
La entidad empleado es obligatoria en la relación *dirigir*. Esto indica que no puede existir un departamento que no tenga un empleado que actúe de director del departamento. La entidad departamento, en cambio, es opcional en la relación *dirigir*. Es posible que haya un empleado que no está relacionado con ningún departamento: puede haber -y es el caso más frecuente- empleados que no son directores de departamento.



Un profesor es tutor de 0 a n alumnos y un alumno tiene exactamente 1 tutor.



Un cliente puede comprar de 0 a n productos y un producto puede ser comprado por 0 a n clientes.

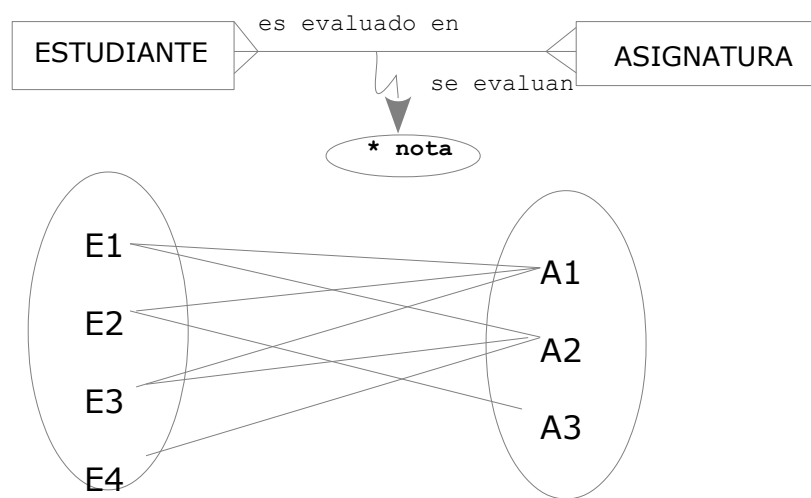


### 3.5.4. Atributos de una relación

Las relaciones también pueden tener atributos, igual que las entidades. Por ejemplo, *el nº de horas que un empleado trabaja en un proyecto es un atributo de la relación **trabaja** (no puede ser ni de Empleado ni de Proyecto).*

Ejemplo de atributo de una relación:

Observemos la entidad estudiante y la entidad asignatura que se muestran en la figura siguiente:



Entre estas dos entidades se establece la relación **evaluar** para indicar de qué asignaturas han sido evaluados los estudiantes. Esta relación tiene el atributo **nota**, que sirve para especificar qué nota han obtenido los estudiantes de las asignaturas evaluadas.

Conviene observar que el atributo **nota** debe ser forzosamente un atributo de la relación **evaluar**, y que no sería correcto considerarlo un atributo de la entidad ESTUDIANTE o un atributo de la entidad ASIGNATURA. Lo explicaremos analizando las ocurrencias de la relación evaluación que se muestran en la figura anterior.

Si **nota** se considerase un atributo de ESTUDIANTE, entonces para el estudiante 'E1' de la figura necesitaríamos dos valores del atributo, uno para cada asignatura que tiene el estudiante; por lo tanto, no sería univaluado. De forma similar, si **nota** fuese atributo de ASIGNATURA tampoco podría ser univaluado porque, por ejemplo, la asignatura 'A1' requeriría tres valores de **nota**, uno para cada estudiante que se ha matriculado en ella. Podemos concluir que el atributo **nota** está relacionado al mismo tiempo con una asignatura y con un estudiante que la cursa y que, por ello, debe ser un atributo de la relación que asocia las dos entidades.

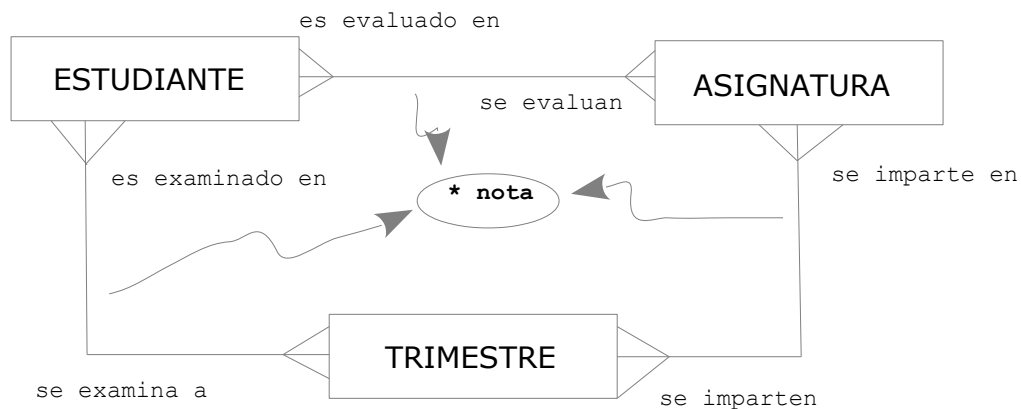
### 3.5.5. Tipo de relación con grado superior a dos

A continuación presentaremos un ejemplo que nos ilustrará el hecho de que, en ocasiones, las relaciones binarias no nos permiten modelizar correctamente la realidad y es necesario utilizar relaciones de mayor grado.

Consideremos la relación **evaluar** de la figura anterior, que tiene un atributo **nota**. Este atributo permite registrar la nota obtenida por cada estudiante en cada asignatura de la que ha sido evaluado. Una relación permite establecer una sola asociación entre unas entidades individuales determinadas. En otras palabras, sólo se puede relacionar una vez al estudiante 'E1' con la asignatura 'A1' vía la interrelación evaluación.

Observad que, si pudiese haber más de una relación entre el estudiante 'E1' y la asignatura 'A1', no podríamos distinguir estas diferentes ocurrencias de la relación. Esta restricción hace que se registre **una sola nota por estudiante y asignatura**.

Supongamos que deseamos registrar varias notas por cada asignatura y estudiante correspondientes a varios trimestres en los que un mismo estudiante ha cursado una asignatura determinada. La relación anterior no nos permitiría reflejar este caso. Sería necesario aumentar el grado de la relación, tal y como se muestra en la figura siguiente:



La relación ternaria evaluación-trimestral asocia estudiantes, asignaturas y una tercera entidad que denominamos trimestre. Su atributo **nota** nos permite reflejar todas las notas de una asignatura que tiene un estudiante correspondientes a diferentes trimestres.

De hecho, lo que sucede en este caso es que, según los requisitos de los usuarios de esta BD, una nota pertenece al mismo tiempo a un estudiante, a una asignatura y a un trimestre y, lógicamente, debe ser un atributo de una relación ternaria entre estas tres entidades.

Este ejemplo demuestra que una relación binaria puede no ser suficiente para satisfacer los requisitos de los usuarios, y puede ser necesario aplicar una relación de mayor grado.



### 3.5.6. Modelo Entidad-Relación Extendido MERE

Las aportaciones de diversos autores al modelo entidad-relación “básico” ha dado lugar al modelo entidad-relación extendido (MERE), que permitirá representar:

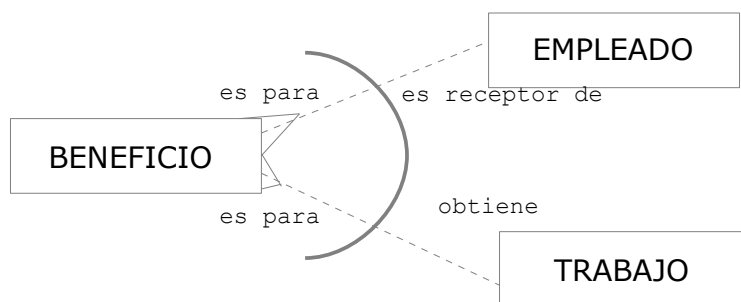
- **Relaciones exclusivas** entre sí.
- Jerarquías de **Especialización/Generalización**.
- **Agregación** de entidades.

#### Relaciones exclusivas (arcos exclusivos)

Dos (o más) tipos de relación son exclusivos, respecto de un tipo de entidad que participa en ambos, si cada instancia del tipo de entidad sólo puede participar en uno de los tipos de relación.

Ejemplo de relación exclusiva:

Cada beneficio de la empresa debe ser, o bien para uno y solo un EMPLEADO, o bien para uno y sólo un TRABAJO.



La exclusividad se representa con un *arco* y se debe cumplir:

- Las relaciones incluidas en un arco deben tener la misma “opcionalidad” y, normalmente, pero no necesariamente, el mismo nombre.
- Las relaciones encerradas deben originarse en la misma entidad.
- Una relación sólo puede formar parte de un arco.
- Un arco debe encerrar, al menos, dos relaciones y, normalmente, no encerrará más de 4.

## Especialización/Generalización

Son relaciones de tipo **IS A** (es un) aquellas en las que una entidad se descompone en entidades especializadas. Estas jerarquías pueden formarse por **especialización** o bien por **generalización**.

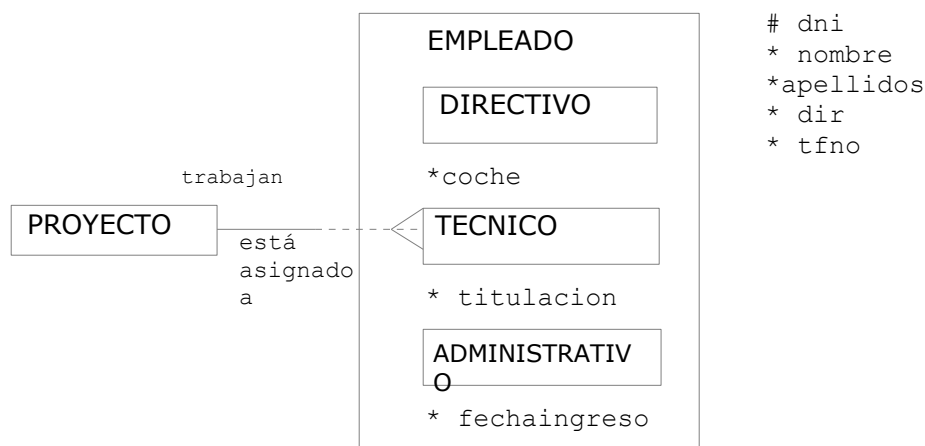
La **especialización** consiste en que una entidad se divide en entidades más concretas. La entidad general comparte con las especializadas sus atributos. Se observa una especialización cuando hay ejemplares para los que no tienen sentido algunos de los atributos, mientras que para otros sí.

Se denomina **generalización** si se agrupan varias entidades en una o más entidades generales. Se observa una generalización si en varias entidades se observan atributos iguales, lo que significa que hay una entidad superior que posee esos atributos.

La generalización/especialización permite reflejar el hecho de que hay una entidad general, que denominamos entidad **supertipo** (o superclase), que se puede especializar en entidades **subtipos** (o subclase):

- La entidad supertipo nos permite modelizar las características comunes de la entidad vista de una forma genérica.
- Las entidades subtipo nos permiten modelizar las características propias de sus especializaciones.

Por ejemplo, puede ocurrir que se quiera tener constancia de qué coche de la empresa tienen asignado los empleados que son **directivos**; también que, de los empleados **técnicos**, interese tener una interrelación con una entidad proyecto que indique en qué proyecto trabajan y se desee registrar su titulación. Finalmente, que convenga conocer la antigüedad de los empleados **administrativos**. Así mismo, habrá algunas características comunes a todos los **empleados**: todos se identifican por un DNI, tienen un nombre, un apellido, una dirección y un número de teléfono.



En este ejemplo, el conjunto de subtipos DIRECTIVO, TECNICO y ADMINISTRATIVO es una especialización del supertipo EMPLEADO mediante la distinción del tipo de trabajo en cada ocurrencia de entidad.

Hay que tener en cuenta que el proceso de generalización puede ser visto funcionalmente como el proceso inverso de especialización. Por tanto, en la figura 9 podemos ver COCHE, CAMION como una especialización de VEHICULO, así como VEHICULO puede verse como la generalización de COCHE y CAMION.

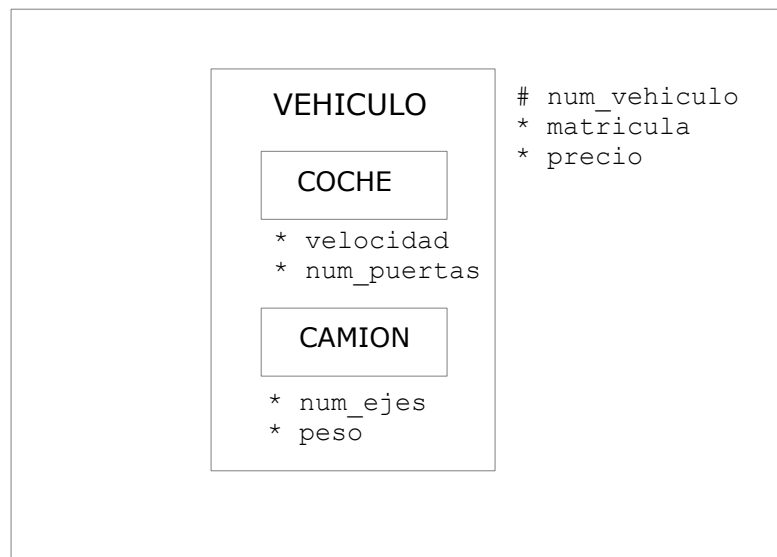


Figura 10. Especialización /Generalización

## Herencia de propiedades

En la generalización/especialización, las características (atributos o interrelaciones) de la entidad supertipo se propagan hacia las entidades subtipo. Es lo que se denomina herencia de propiedades.

Del mismo modo, las relaciones que afectan a todos los subtipos se asocian a los supertipos, dejándose para los subtipos las relaciones específicas en las que sólo participa el correspondiente subtipo.

Un subtipo, con sus atributos y relaciones específicos, más los atributos y relaciones que hereda del supertipo, es un tipo de entidad por derecho propio.

## Generalización vs. Especialización

Generalización:

- Énfasis en las similitudes.
- Cada instancia del supertipo es también una instancia de alguno de los subtipos.

Especialización:

- Énfasis en las diferencias.
- Alguna instancia del supertipo puede no ser instancia de ningún subtipo.

## Restricciones sobre la Especialización/Generalización

A la hora de hablar de restricciones sobre la E/G deberíamos contestar a unas cuantas preguntas:

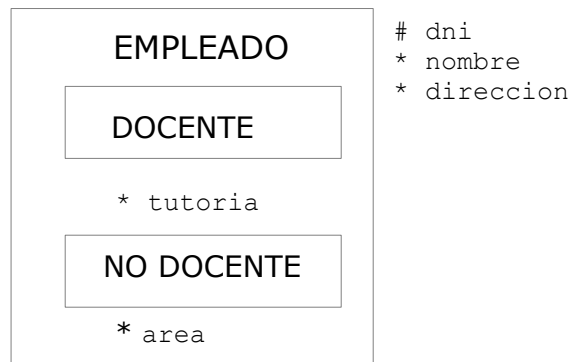
- ¿Qué instancias del supertipo pertenecen a cada subtipo?
- ¿A cuántos subtipos puede pertenecer (a la vez) una instancia del supertipo?
- ¿Debe toda instancia del supertipo pertenecer a algún subtipo?

Se pueden distinguir cuatro clases de generalización atendiendo a si los subtipos se solapan o son disjuntos y a si la unión de los subtipos recubre o no al supertipo.

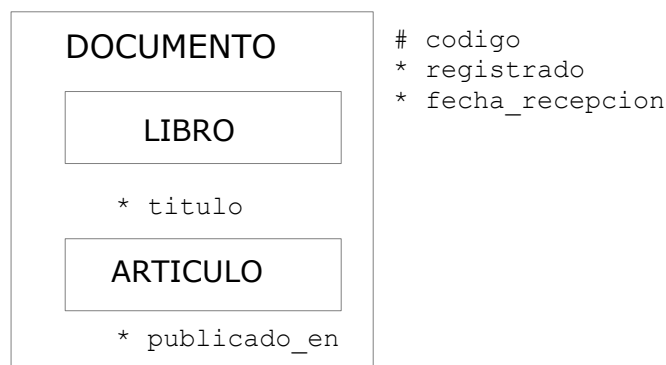
- Jerarquía **Total**: Toda ocurrencia del supertipo debe obligatoriamente pertenecer a algún subtipo.
- Jerarquía **Parcial**: Una ocurrencia del supertipo no tiene por que pertenecer a algún subtipo.
- Jerarquía **Solapada**: Una ocurrencia del supertipo puede pertenecer a varios subtipos.
- Jerarquía **Exclusiva** o **Disjunta**: Una ocurrencia del supertipo no puede pertenecer a varios subtipos.

Por tanto los posibles cuatro tipos de generalizaciones serán: total-exclusiva, parcial-exclusiva, total-solapada y parcial-solapada.

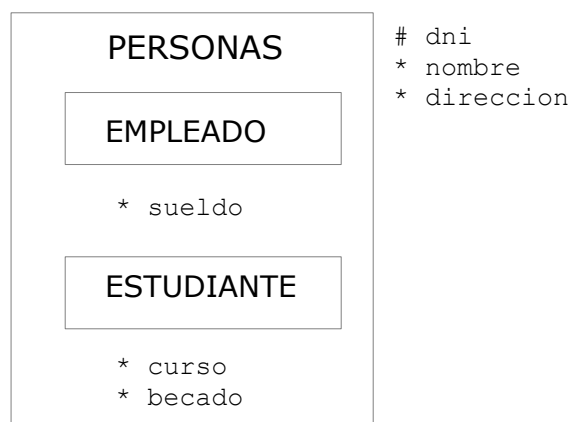
1. **Total-exclusiva**: Tanto un docente como un no docente son empleados. Un mismo empleado no puede ser a la vez docente y no docente (disjuntos). Todo empleado tiene que ser obligatoriamente un docente o un no docente (total).



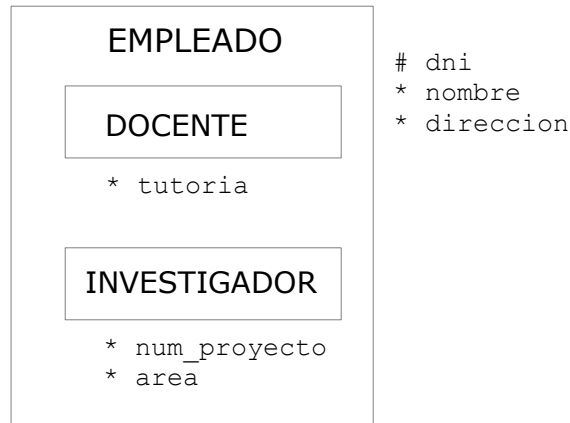
2. **Parcial-exclusiva:** Tanto un artículo como un libro son documentos. Un mismo documento no puede ser a la vez un artículo y un libro (disjuntos). Un documento puede no ser ni un artículo ni un libro (parcialidad) por tanto no tendrá los atributos específicos de los artículos y los libros (por ejemplo un documento puede ser una carta).



3. **Total-solapada:** Tanto un empleado como un estudiante son personas. Una misma persona puede ser estudiante y empleado a la vez (solapada). Toda persona registrada en la BD tiene que ser obligatoriamente un estudiante y/o un empleado (total).



4. **Parcial-solapada:** Tanto un docente como un investigador son empleados. Un mismo empleado puede ser, y en general lo es, docente a la vez que investigador (solapada). Un empleado puede no ser ni un docente ni un investigador (parcial).



## Reglas de inserción y eliminación

Deben aplicarse a la Especialización y la Generalización, debido a las restricciones definidas.

- **Insertar** una instancia en un supertipo implica insertarla en todos los subtipos definidos por predicado o por atributo, para los cuales satisface el predicado de definición.
- Insertar una instancia en un supertipo de una especialización total implica insertarla en, al menos, un subtipo.
- Y si la especialización es disjunta, entonces la instancia se insertará en un único subtipo.
- **Eliminar** una instancia de un supertipo implica eliminarla de todos los subtipos a los que pertenece.
- Eliminar una instancia de un subtipo implica eliminarla del supertipo si la especialización es disjunta y total, o bien solapada y total, y la instancia ya sólo pertenece al subtipo (se eliminó del resto).
- En el resto de casos, la instancia sólo se elimina del subtipo no del supertipo ( lo haría el usuario, si fuese necesario).

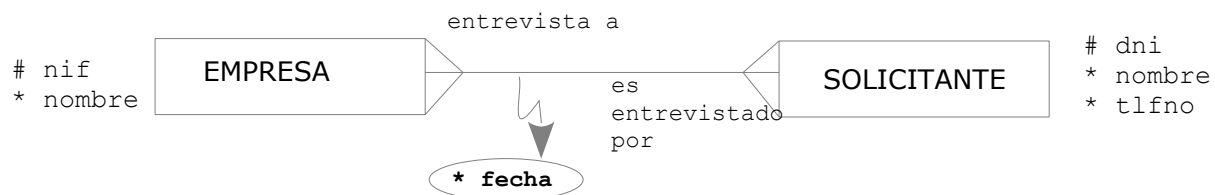
## Agregación de tipos de entidad

El Modelo E/R presenta una serie de restricciones inherentes a él como son:

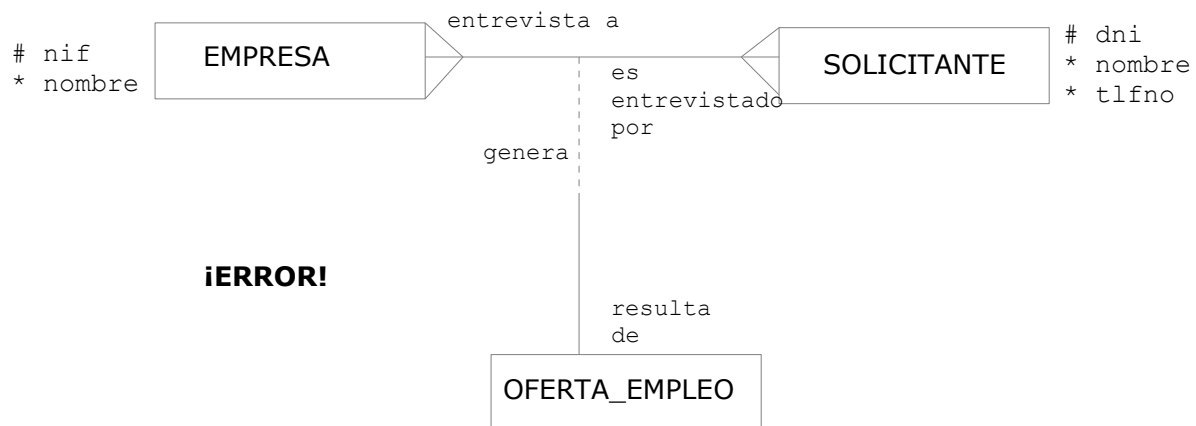
- No se puede expresar relaciones entre varias relaciones.
- No se puede establecer relaciones entre un tipo de relación y un tipo de entidad.

La agregación permite combinar varios tipos de entidad, relacionados mediante un tipo de relación, para formar un tipo de entidad agregada de nivel superior. Es útil cuando el tipo de entidad agregada debe relacionarse con otros tipos de entidad.

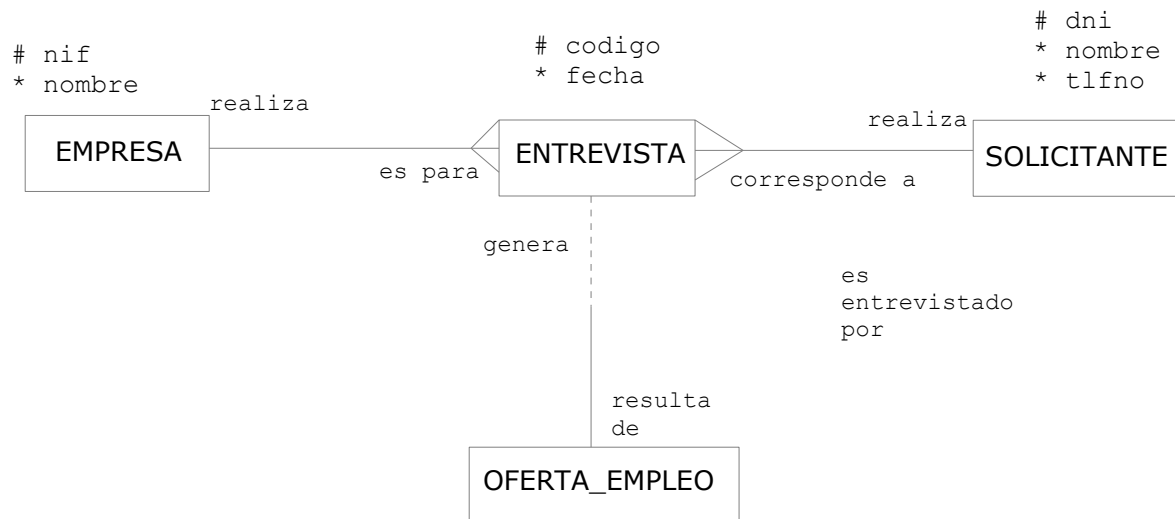
Supongamos una base de datos que almacena información sobre las entrevistas que una ETT organiza entre solicitantes de empleo y diferentes empresas.



Algunas entrevistas dan lugar a ofertas de empleos y otras no. ¿Cómo modelamos esto?



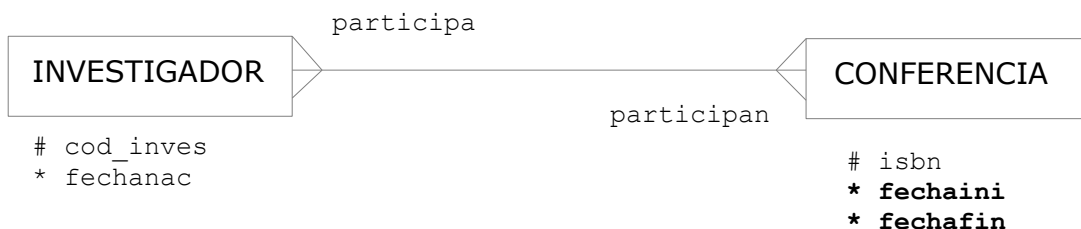
La solución sería crear una entidad agregada que resulte de la relación *entrevistar*, entidad que podremos relacionar con la nueva entidad OFERTA\_EMPLEO:



### 3.6. Dimensión temporal en el Modelo Entidad-Relación

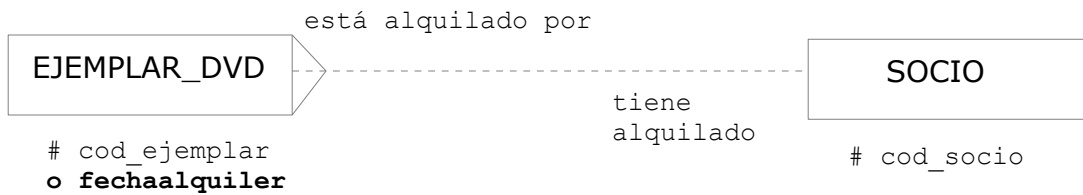
Según la base de datos que deseemos diseñar tendremos que indicar la dimensión temporal en algunas relaciones.

Una primera aproximación a la dimensión temporal en un esquema E/R sería:





En algunas ocasiones, diseñaremos una base de datos actual en la que no consideraremos el pasado:



Y en otras, necesitaremos una base de datos histórica en la que tendremos en cuenta el pasado:

