

**BASES DE DATOS**

**Desarrollo de Aplicaciones Web  
GESTIÓN DE BASES DE DATOS**

**Administración de Sistemas Informáticos en Red**

**MODELADO CONCEPTUAL**

**Apuntes**

**Luis Dorado Garcés**

**Basado en el trabajo de Alba Tortosa López**

ÍNDICE

[1 Introducción a los modelos de datos 1](#_Toc53745650)

[2 Fases de diseño de una BD 3](#_Toc53745651)

[2.1 Fase 1. Diseño Conceptual. 3](#_Toc53745652)

[2.1.1 Fase de Análisis: Especificación de requisitos Software (E.R.S.) 3](#_Toc53745653)

[2.1.2 Modelo Entidad/Relación (E/R) 3](#_Toc53745654)

[2.2 Fase 2. Diseño Lógico: Modelo Relacional 4](#_Toc53745655)

[2.3 Fase 3. Diseño Físico: Modelo Físico 4](#_Toc53745656)

[3 Modelo conceptual I (modelo E/R) 6](#_Toc53745657)

[3.1 Presentación e historia del modelo 6](#_Toc53745658)

[3.2 Elementos del Modelo Entidad/Relación 7](#_Toc53745659)

[3.3 Entidades 8](#_Toc53745660)

[3.4 Atributos 9](#_Toc53745661)

[3.4.1 Dominio de un atributo 9](#_Toc53745662)

[3.5 Clave primaria 10](#_Toc53745663)

[3.6 Elaboración de diagramas E/R: Entidades y atributos 11](#_Toc53745664)

[3.6.1 Identificación de entidades 12](#_Toc53745665)

[3.6.2 Identificación de atributos y claves 12](#_Toc53745666)

[3.7 Relaciones 13](#_Toc53745667)

[3.8 Elaboración de diagramas E/R: Relaciones 14](#_Toc53745668)

[3.8.1 Identificación de relaciones 14](#_Toc53745669)

[4 Modelo conceptual II (modelo E/R) 15](#_Toc53745670)

[4.1 Entidades Fuertes y Débiles 15](#_Toc53745671)

[4.1.1 Clave primaria en entidades débiles 16](#_Toc53745672)

[4.1.2 Entidad débil participante en dos relaciones 16](#_Toc53745673)

[4.2 Atributos complejos 17](#_Toc53745674)

[4.2.1 Atributos de una relación 17](#_Toc53745675)

[4.2.2 Atributos multivaluados 18](#_Toc53745676)

[4.2.3 Atributos derivados/calculados. 19](#_Toc53745677)

[4.2.4 Atributos simples o compuestos. 19](#_Toc53745678)

[*4.2.5* *Ampliación: Atributos obligatorios/opcionales.* 20](#_Toc53745679)

[4.3 Cardinalidad de las relaciones y entidades 20](#_Toc53745680)

[4.3.1 Cardinalidad de las relaciones 21](#_Toc53745681)

[4.3.1.1 Relaciones uno a uno (1:1) 21](#_Toc53745682)

[4.3.1.2 Relaciones uno a muchos (1:N) 22](#_Toc53745683)

[4.3.1.3 Relaciones muchos a uno (N:1) 22](#_Toc53745684)

[4.3.1.4 Relaciones muchos a muchos (N:M) 23](#_Toc53745685)

[4.3.2 Cardinalidad de entidades 23](#_Toc53745686)

[*4.3.3* *Ampliación: Cardinalidad en relaciones de debilidad* 24](#_Toc53745687)

[4.4 Relaciones complejas 24](#_Toc53745688)

[4.4.1 Relaciones reflexivas 24](#_Toc53745689)

[4.4.2 Relaciones ternarias 25](#_Toc53745690)

[4.5 Jerarquías 25](#_Toc53745691)

[4.5.1 Restricciones 26](#_Toc53745692)

[4.5.2 Representación 27](#_Toc53745693)

[4.5.3 Jerarquía total con exclusividad 28](#_Toc53745694)

[4.5.4 Jerarquía parcial con exclusividad 29](#_Toc53745695)

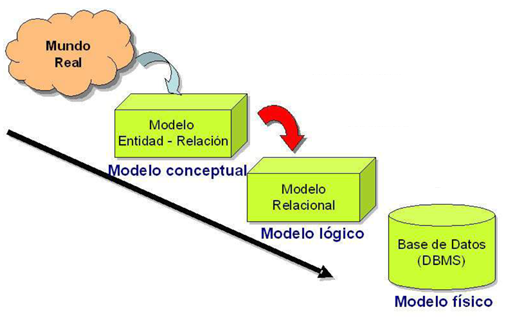
[4.5.5 Jerarquía total con solapamiento 30](#_Toc53745696)

[4.5.6 Jerarquía parcial con solapamiento 30](#_Toc53745697)

[4.5.7 Identificación de jerarquías 30](#_Toc53745698)

[4.6 Control de redundancia 31](#_Toc53745699)

# Introducción a los modelos de datos



En este tema veremos cómo elaborar el diseño conceptual y lógico de una base de datos. Empezaremos diseñando el modelo conceptual usando diagramas Entidad-Relación y Entidad-Relación extendidos. Este diseño es de más alto nivel, más próximo a la realidad y más alejado del diseño físico de la BD. A continuación, a partir del modelo Entidad-Relación (conceptual), procederemos a generar el modelo relacional (lógico), el cual ya se halla muy próximo al modelo físico de BD. Veremos las reglas de transformación que hemos de seguir para ello.

**¿Y qué es un modelo de datos?** No es sino una serie de **diagramas, especificaciones, metodologías**… que conforman un **modo de representar datos que describen una realidad**.

También lo podemos definir como un **esquema teórico** mediante el cual se expresan las **propiedades, estáticas y dinámicas, de una realidad** que se analiza con vistas a ser representada mediante una base de datos.

Al representar una base de datos mediante un modelo, se pueden emplear diferentes enfoques: uno orientado a los conceptos y otro, orientado a los datos (lógico).

Los modelos orientados al concepto, no hacen referencia a la estructura que se empleará para representar los datos, sino a los conceptos, propiedades y relaciones que se establecen entre dichos conceptos. Es cercano al pensamiento humano.

Sin embargo, al utilizar modelos orientados a los datos (lógico) se opera directamente con la información que contendrá la base de datos, es decir, con los datos.

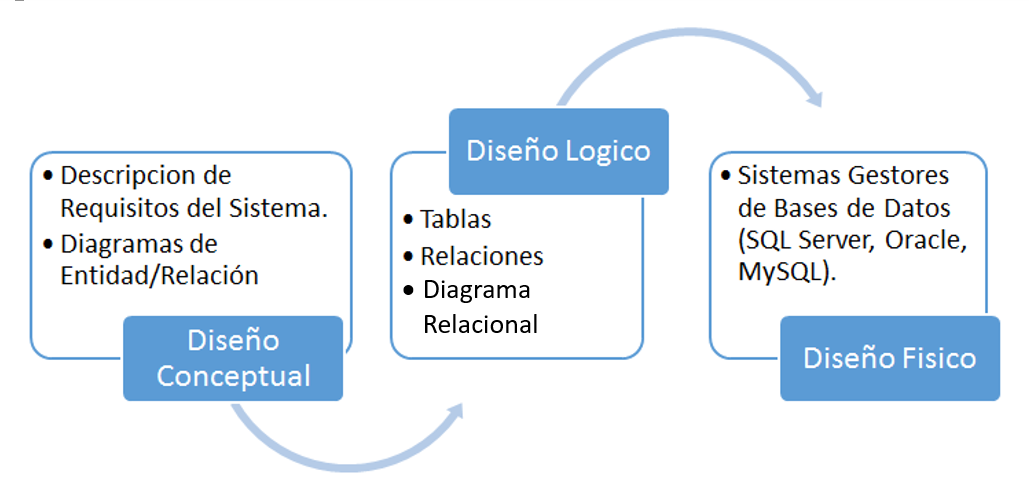
Resumiendo, los 2 modelos de datos, de mayor a menor nivel de abstracción, que veremos en esta área son:

* Nivel conceptual: Modelo Entidad-Relación (extendido)
* Nivel lógico: Modelo Relacional

Más adelante, realizaremos el diseño físico de la BD a partir del modelo relacional.

# Fases de diseño de una BD

El diseño de una base de datos consiste en extraer todos los datos relevantes de un problema, por ejemplo, saber que datos están implicados en el proceso de facturación de una empresa que vende artículos de informática, o, que datos son necesarios para llevar el control de pruebas diagnósticas en un centro de radiológico. Para extraer estos datos, se debe realizar un análisis en profundidad del problema, para averiguar qué datos son esenciales para la base de datos y descartar los que no sean necesarios. Tras ello iremos elaborando unos modelos de datos que vayan adaptando la realidad al modelo físico que define el SGBD específico.



## Fase 1. Diseño Conceptual.

### Fase de Análisis: Especificación de requisitos Software (E.R.S.)

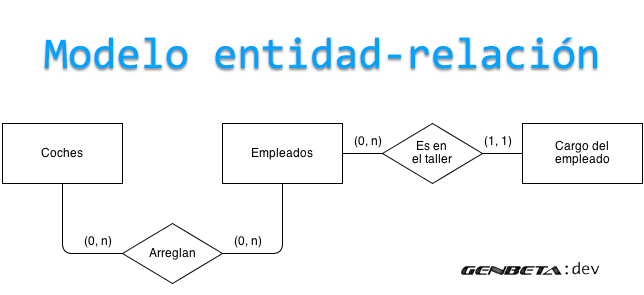
Antes de pasar a diseñar una BD hay que tener claro que es lo que queremos hacer. Para ello, típicamente los informáticos se reúnen con los futuros usuarios del sistema para recopilar la información que necesitan para saber que desean dichos usuarios.

De estas entrevistas, se extrae el documento más importante del análisis, el documento de Especificación de Requisitos Software o E.R.S. A partir de dicha E.R.S. Se extrae toda la información necesaria para la modelización de datos.

La información se extrae, por lo general, de clientes expertos en el ámbito a modelar pero igorantes del diseño de BBDDs. Tendremos que tener paciencia e irles guiando para extraer toda la información necesaria. En ocasiones habrá que desechar algunas de sus ideas para proponer otras más funcionales y eficientes.

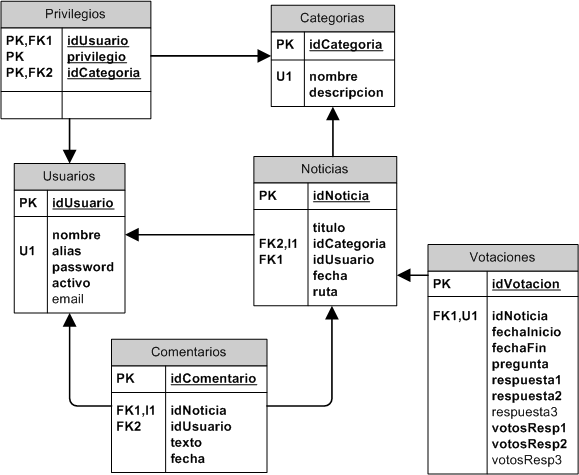
### Modelo Entidad/Relación (E/R)

El objetivo de esta fase del diseño consiste es representar la información obtenida del usuario final y concretada en el E.R.S. mediante estándares para que el resto de la comunidad informática pueda entender y comprender el modelo realizado. El modelo que se utiliza en esta primera fase del diseño tiene un gran poder expresivo para poder comunicarse con el usuario que no es experto en informática y se denomina Modelo Conceptual. El modelo conceptual que utilizaremos es el Modelo Entidad/Relación e iremos profundizando en él a lo largo de estos apuntes.



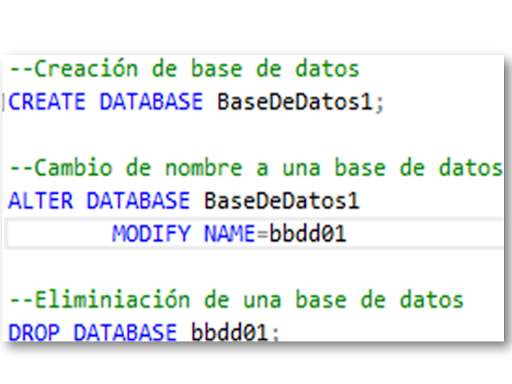
## Fase 2. Diseño Lógico: Modelo Relacional

Este modelo es más técnico que el anterior porque está orientado al personal informático y generalmente tiene traducción directa al modelo físico que entiende el SGBD. Se obtienen a partir del modelo conceptual y dependerá de la implementación de la BD. Así, no es lo mismo implementar una base de datos jerárquica u orientada a objetos que una BD relacional. El modelo que se usará en este módulo es el Modelo Relacional.



## Fase 3. Diseño Físico: Modelo Físico

Es el resultado de aplicar el modelo lógico a un SGBD concreto. Generalmente está expresado en un lenguaje de programación de BBDD tipo SQL. En este módulo, transformaremos el Modelo Relacional en el modelo físico mediante el sublenguaje DDL de SQL. Esto se estudiará más adelante.



# Modelo conceptual I (modelo E/R)

## Presentación e historia del modelo

El modelo Entidad/Relación fue desarrollado por Peter Chen en el año 1976 y, a pesar del tiempo transcurrido desde su presentación, es un modelo de datos de plena actualidad en el ámbito de la ingeniería en informática y, más concretamente, en el campo del diseño de bases de datos.

¿Cuál es entonces la clave de su éxito? Inicialmente podría esgrimirse como argumento el hecho de que este nuevo modelo de datos intenta aglutinar las ventajas de cada uno de los modelos de datos anteriores: modelo en red, modelo jerárquico y modelo relacional.

Sin embargo, el argumento anterior no justifica totalmente la utilidad del Modelo Entidad/Relación que lo ha convertido, por ejemplo, en un modelo valido para el diseño de Bases de Datos Orientadas a Objetos, ámbito éste tan moderno que no era previsible que un modelo de datos con veinte años de antigüedad pudiera ser utilizado, pese a sus evoluciones, en este nuevo campo de aplicación de los sistemas de información.

Para completar la respuesta a la pregunta anterior debemos argumentar que su potencia para representar prácticamente todas las restricciones posibles del diseño de datos, junto con su flexibilidad para admitir la evolución en el tiempo del sistema de información diseñado, son un combinado ideal para justificar tal éxito.

Una segunda pregunta a resolver sería: ¿Por qué es necesario un nuevo modelo de datos? En este caso, la respuesta hay que buscarla en el cambio sufrido en el proceso de desarrollo de los sistemas de información a finales de los años 60, principios de los 70. Inicialmente, el proceso de desarrollo de sistemas de información estaba centrado en el análisis y diseño de los tratamientos a realizar, dejando en un segundo plano el análisis y diseño de los datos.

El cambio fue la respuesta a la constatación de una realidad que se daba en todos los sistemas de información desarrollados: Los datos como estructura son más duraderos que los tratamientos.

Expresado en otros términos, el sistema de información de una determinada organización permanece casi invariable en lo que a sus datos de interés se refiere, mientras que es cambiante en el tratamiento dado a los datos.

Por tanto, podemos asegurar que el Modelo Entidad/Relación es la consecuencia de la necesidad de aportar soluciones al problema anteriormente mencionado.

Para ello el Modelo Entidad/Relación percibe el mundo real como una serie de objetos relacionados entre sí y pretende representarlos gráficamente, mediante un determinado mecanismo de abstracción. Este mecanismo de abstracción está basado en una serie de símbolos, reglas y métodos que nos permitirán representar gráficamente los datos de interés del mundo real. Es decir, el Modelo Entidad/Relación fue creado como una metodología gráfica para diseño de bases de datos.

Ahora bien, podría considerarse que el modelo E/R es un modelo intuitivo por el hecho de basarse en la representación gráfica de los objetos y asociaciones del mundo real. Sin embargo no debemos olvidar que este modelo nació como una generalización de los tres modelos de datos existentes en ese momento, intentando dar una visión más uniforme de los datos y mantener las ventajas de cada uno de dichos modelos.

De hecho, el modelo E/R permite una visión más natural de los datos, separando los objetos de sus asociaciones (al igual que el Modelo en Red), mantiene un alto grado de independencia de los datos respecto a los tratamientos (al igual que el modelo Relacional), y establece un cierto nivel de dependencia o jerarquía entre los distintos elementos componentes del Modelo (al igual que el Modelo Jerárquico).

Además, el Modelo E/R aporta un mayor contenido semántico sobre el universo objeto de estudio que cualquiera de los otros tres modelos de datos mencionados.

Por último, conviene destacar otra característica del modelo E/R derivada de su orientación al diseño de datos del sistema de información de una organización: vamos a realizar el diseño lógico de la base de datos ignorando consideraciones de almacenamiento físico de los datos y de eficiencia de los tratamientos. Es decir, el diseño que vamos a realizar mediante el Modelo E/R es desde todo punto de vista asimilable al esquema conceptual definido para la arquitectura ANSI/X3/SPARC.

La diferencia radica en su función:

El diseño obtenido no es el nexo de unión entre el mundo del usuario (esquema externo), y el mundo del computador (esquema interno), aunque a partir del mismo se puedan definir ambos. Solamente es una representación de las propiedades lógicas de los datos del universo objeto de estudio y, por tanto, dicha representación no es accesible directamente por el SGBD.

Es un método de representación abstracta del mundo real centrado en las restricciones o propiedades lógicas de una base de datos. Por tanto, no es directamente implantable en un SGBD, sino que necesita una transformación a las estructuras de datos del modelo de datos propio de dicho SGBD.

En un principio, el Modelo E/R sólo contemplaba los conceptos de “entidad”, “relación” y “atributo”. En la siguiente revisión del modelo ya se contemplaban otros conceptos como atributo multiocurrente y generalización.

## Elementos del Modelo Entidad/Relación

El Modelo Entidad/Relación, como cualquier modelo de datos, tiene sus estructuras propias que son conocidas como Diagramas Entidad/Relación.

De hecho, para describir el esquema conceptual de la base de datos del mundo objeto de estudio se construye su Diagrama Entidad/Relación.

Los elementos componentes del Modelo Entidad/Relación son los siguientes:

* Las **entidades**: objetos de estudio como clientes, productos, cuentas bancarias, etc.
* Los **atributos**: propiedades de las entidades como nombre, cantidad, saldo, etc.
* Las **relaciones** entre entidades: Por ejemplo, un cliente de un banco tiene asociada una o más cuentas bancarias.

Cada uno de estos elementos tiene asociado un modo gráfico de representación o símbolo específico, que lo distingue del resto de elementos. En los apartados siguientes describiremos cada uno de estos elementos, sus características y simbología.

## Entidades

Si utilizamos las bases de datos para guardar información sobre cosas que nos interesan o que interesan a una organización, ¿No crees que hay que identificar esas cosas primero para poder guardar información sobre ellas? Para ello, vamos a describir un primer concepto, el de **Entidad**.

Una **entidad** puede ser un objeto físico, un concepto o cualquier elemento que queramos modelar, que tenga importancia para la organización y del que se desee guardar información. Cada entidad debe poseer alguna característica, o conjunto de ellas, que lo haga único frente al resto de objetos.

Por ejemplo, podemos establecer una entidad llamada **ALUMNOS** que tendrá una serie de características. El alumnado podría ser distinguido mediante su número de identificación escolar (NIE), por ejemplo.

**Entidad**: objeto real o abstracto, con características diferenciadoras capaces de hacerse distinguir de otros objetos.

¿Ponemos otro ejemplo? Supongamos que tienes que desarrollar el esquema conceptual para una base de datos de mapas de montaña, los elementos: camping, pista forestal, valle, río, pico, refugio, etc., son ejemplos de posibles entidades. A la hora de identificar las entidades, hemos de pensar en nombres que tengan especial importancia dentro del lenguaje propio de la organización o sistema que vaya a utilizar dicha base de datos. Pero no siempre una entidad puede ser concreta, como un camping o un río, en ocasiones puede ser abstracta, como un préstamo, una reserva en un hotel o un concepto.

Asociado al concepto de entidad surge el concepto de **ocurrencia** de entidad. Una **ocurrencia de entidad** no es otra cosa que una realización concreta de una entidad. Porejemplo, si tenemos la entidad FRUTAS con características como nombre, color, árbol de procedencia… una ocurrencia de la misma será PERAS

La **representación gráfica** de una entidad es un rectángulo etiquetado.



Reglas que debe cumplir una entidad:

* Tiene que tener existencia propia.
* Cada ocurrencia de un tipo de entidad debe poder distinguirse de las demás a través de sus características concretas (valores de sus atributos).
* Todas las ocurrencias de un tipo de entidad deben tener los mismos tipos de características (atributos)

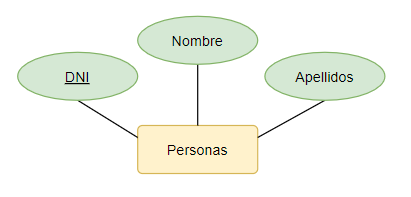
## Atributos

**¿Cómo guardamos información de cada entidad?** A través de sus **atributos**. Las entidades se representan mediante un conjunto de atributos. Éstos describen características o propiedades que posee cada miembro de un conjunto de entidades. El mismo atributo establecido para un conjunto de entidades o, lo que es lo mismo, para un tipo de entidad, almacenará información parecida para cada ocurrencia de entidad. Pero, cada ocurrencia de entidad tendrá su propio valor para cada atributo.

**Atributo**: Cada una de las propiedades o características que tiene un tipo de entidad o un tipo de relación se denomina atributo; los atributos toman valores de uno o varios dominios.

Por tanto, un atributo se utilizará para guardar información sobre alguna característica o propiedad de una entidad o relación. Ejemplos de atributos pueden ser: **altura**, **color**, **peso**, **DNI**, **fecha**, etc. todo dependerá de la información que sea necesaria almacenar.

En el modelo Entidad/Relación los atributos de una entidad son **representados** mediante elipses. El círculo se conecta con la entidad mediante una línea recta. Cada atributo debe tener un nombre único que haga referencia al contenido de dicho atributo. En la siguiente imagen se ven algunos ejemplos de atributos de la entidad **PERSONAS**.



### Dominio de un atributo

Al conjunto de valores permitidos para un atributo se le denomina **dominio**. Todos los posibles valores que puede tomar un atributo deberán estar dentro del dominio. Varios atributos pueden estar definidos dentro del mismo dominio. Por ejemplo, los atributos nombre, apellido primero y apellido segundo de la entidad **PERSONAS**, están definidos dentro del dominio de cadenas de caracteres de una determinada longitud.

Aunque los dominios suelen ser amplios (números enteros, reales, cadenas de caracteres, etc.), a la hora de llevar a cabo el desarrollo de una base de datos, es mejor establecer unos límites adecuados para que el sistema gestor de la base de datos lleve a cabo las verificaciones oportunas en los datos que se almacenen, garantizando así la integridad de éstos. Los dominios más utilizados y sus restricciones se pueden observar en la siguiente tabla:

|  |  |  |
| --- | --- | --- |
| **Dominio** | **Restricciones** | **Ejemplos** |
| **Texto** | Longitud máxima | Máx. 30 caracteres |
| **Número** | Longitud máxima  Entero o decimal  Positivo y/o negativo | Máx. 5 dígitos  Permite decimales  No permite negativos |
| **Temporal** | Formato | Año-mes-día  HH:MM  Año |
| **Lista de valores** | Lista de valores concretos | Meses del año  Rojo, azul, amarillo |
| **Si/No** | Restringido a 2 valores | Sí/No, Encendido/Apagado, 0/1, Alto/Bajo. |

¿Todos los atributos son iguales? Claro que no. Existen varias características que hacen que los atributos asociados a una entidad o relación sean diferentes, los clasificaremos según varios criterios.

## Clave primaria

En el apartado anterior hablábamos de un tipo de atributo especial obligatorio, **las claves o llaves**. Ahora es el momento de abordar con mayor detalle este concepto.

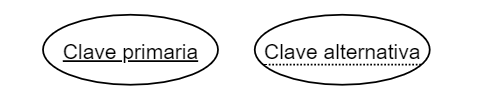
Está claro que es necesario identificar correctamente cada ocurrencia de entidad o relación, de este modo el tratamiento de la información que se almacena podrá realizarse adecuadamente. Esta distinción podría llevarse a cabo tomando todos los valores de todos los atributos de una entidad o relación. Pero, en algunas ocasiones, sabemos que puede no ser necesario utilizar todos, bastando con un subconjunto de ellos. Aunque puede ocurrir que ese subconjunto tenga idénticos valores para varias entidades, por lo que cualquier subconjunto no será válido.

Por tanto, los valores de los atributos de una entidad deben ser tales que permitan **identificar unívocamente** a la entidad. En otras palabras, no se permite que ningún par de ocurrencias de una misa entidad tengan exactamente los mismos valores de sus atributos. Teniendo en cuenta esto, presta atención a los siguientes conceptos:

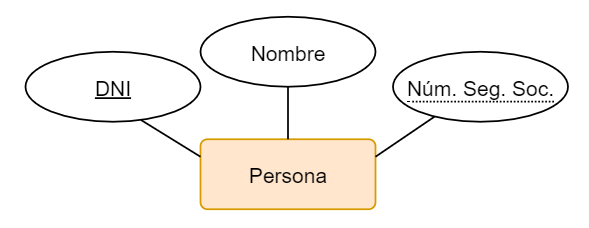
* **Clave candidata**: Cualquier conjunto mínimo de atributos que permite identificar de forma única una ocurrencia de una entidad.
* **Clave primaria (Primary Key)**: También llamada llave primaria o clave principal. De todas las claves candidatas, el diseñador de la base de datos ha de escoger una, que se denominará clave principal o clave primaria. La clave primaria es un atributo o conjunto de ellos, que toman valores únicos y distintos para cada ocurrencia de entidad, identificándola unívocamente. No puede contener valores nulos.
* **Claves alternativas**: son el resto de claves candidatas que no han sido escogidas como clave primaria.

Representación gráfica de atributos clave:

**Clave primaria Clave alternativa**



Ejemplo:



## Elaboración de diagramas E/R: Entidades y atributos

Llegados a este punto, te surgirán varias dudas ¿Cómo creo un diagrama E/R? ¿Por dónde empiezo? ¿Y qué puedo hacer con todo lo visto? Son cuestiones totalmente normales cuando se comienza, no te preocupes, vamos a darte una serie de orientaciones para que puedas aplicar todos los conceptos aprendidos hasta ahora en la elaboración de diagramas Entidad/Relación.

Sabemos que en la fase de diseño conceptual de la base de datos, en la que nos encontramos, hemos de generar el diagrama E/R que representará de manera más sencilla el problema real a modelar, independientemente del Sistema Gestor de Base de Datos. Este esquema será como un plano que facilite la comprensión y solución del problema. Este diagrama estará compuesto por la representación gráfica, a través de la simbología vista anteriormente, de los requisitos o condiciones que se derivan del problema a modelar.

Saltarnos este paso en el proceso de creación e implementación de una base de datos, supondría pérdida de información. Por lo que esta fase, requerirá de la creación de uno o varios esquemas previos más cercanos al mundo real, antes del paso a tablas del modelo relacional.

Lo primero que hemos de tener a nuestra disposición para poder generar un diagrama E/R adecuado es el conjunto de requerimientos, requisitos o condiciones que nuestra base de datos ha de cumplir. Es lo que se denomina el documento de especificación de requerimientos. En otras palabras, el **enunciado del problema a modelar**. Cuanto más completa y detallada sea la información de la que dispongamos, mucho mejor.

Suponiendo que conocemos la simbología del modelo Entidad/Relación y que entendemos su significado ¿Cómo empezamos? Las etapas para la creación del diagrama E/R se detallan a continuación.

### Identificación de entidades

Es un proceso bastante intuitivo. Para localizar aquellos elementos que serán las entidades de nuestro esquema, analizaremos la especificación de requerimientos en busca de **nombres o sustantivos**. Si estos nombres se refieren a objetos importantes dentro del problema probablemente serán entidades. Tendremos en cuenta que nombres referidos a características, cualidades o **propiedades no se convertirán en entidades** sino en atributos de estas entidades.

Otra forma de identificar entidades es localizando objetos o elementos que existen por sí mismos. Por ejemplo: **VEHICULO**, **PIEZA**, etc. En otras ocasiones, la localización de varias características o propiedades puede dejar ver la existencia de una entidad.

¿Esto puede ser una entidad o no? Es una pregunta que se repite mucho cuando estamos en esta etapa. Algunos autores indican que para poder considerarse como entidad se deben cumplir tres reglas:

* Existencia propia.
* Cada ejemplar de un tipo de entidad debe poder ser diferenciado del resto de ejemplares.
* Todos los ejemplares de un tipo de entidad deben tener las mismas propiedades.

El número de entidades obtenidas debe ser manejable y según se vayan identificando se les otorgará **nombres en plural, representativos** de su significado o función. De esta manera el diagrama será cada vez más legible.

### Identificación de atributos y claves

Sólo con la localización de entidades no está todo hecho. Hemos de completar el proceso realizando las siguientes tareas:

**Identificación de atributos**: Volvemos sobre el documento de especificación de requerimientos para buscar nombres relativos a **características, propiedades, identificadores o cualidades** de entidades o ~~relaciones~~ (esto último lo veremos más tarde). Resulta más sencillo si nos preguntamos ¿Qué información es necesario tener en cuenta de una u otra entidad? Quizás no todos los atributos estén reflejados directamente en el documento de especificación de requerimientos, aplicando el sentido común el diseñador podrá establecerlos en algunos casos y en otros, será necesario consultar e indagar en el problema.

Tendremos en cuenta si los atributos localizados son simples o compuestos, derivados o calculados, multivaluados...

Cada atributo deberá tener asignado un nombre representativo de su contenido o función. Además, siempre es recomendable recopilar la siguiente información de cada atributo:

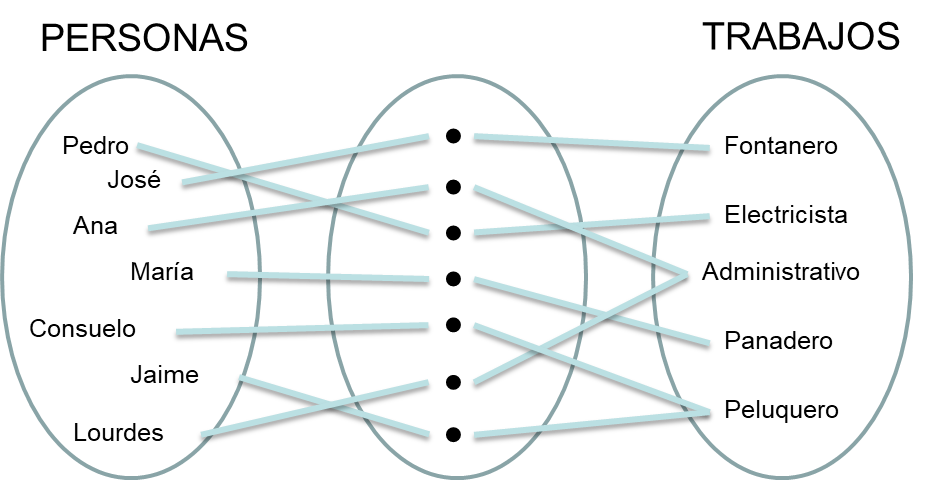
* Nombre y descripción.
* Atributos simples que lo componen, si es atributo compuesto.
* Método de cálculo, si es atributo derivado o calculado.

**Identificación de claves**: Del conjunto de atributos de una entidad se establecerán una o varias claves candidatas, escogiéndose una de ellas como clave o llave primaria de la entidad. Esta clave estará formada por uno o varios atributos que identificarán de manera unívoca cada ocurrencia de entidad.

**Identificación de dominios:** Para cada atributo, será necesario establecer su dominio, sus restricciones y si es opcional o no.

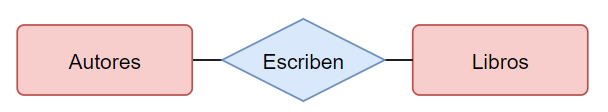
## Relaciones

¿Cómo interactúan entre sí las entidades? A través de las relaciones. La relación o interrelación es un elemento del modelo Entidad/Relación que permite relacionar datos entre sí. En una relación se asocia un elemento de una entidad con otro de otra entidad.



**Relación**: es una asociación entre diferentes entidades. En una relación no pueden aparecer dos veces relacionadas las mismas ocurrencias de entidad.

Ejemplos de relación: PROVEEDOR **suministra** PRODUCTO, PERSONA **ha nacido en** PAÍS, EMPLEADO **trabaja en** DEPARTAMENTO, etc… La Relación se representa gráficamente por medio de un rombo y en el interior del mismo se escribe la etiqueta que identifica la Relación. Esta etiqueta será normalmente un verbo. En el siguiente diagrama vemos la relación entre las relaciones LIBRO y AUTOR: “Un autor **escribe** libros”.



Las relaciones son **bidireccionales**. En el ejemplo anterior: “Un autor escribe libros” pero “Un libro es escrito por un autor”.

Asociado al concepto de relación surge el concepto de **ocurrencia de relación**. Una ocurrencia de relación es la asociación concreta de ocurrencias de diferentes entidades. Por ejemplo, si tenemos las entidades EMPLEADO y DEPARTAMENTO, y la relación **trabaja en**, una ocurrencia de Relación será: MARTA GARCÍA **trabaja en** elDEPARTAMENTO DE DIRECCIÓN.

## Elaboración de diagramas E/R: Relaciones

### Identificación de relaciones

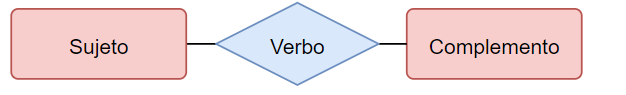
Localizadas las entidades, sus atributos y sus claves, debemos establecer qué relación existe entre ellas. Para ello, analizaremos de nuevo el documento de especificación de requerimientos en busca de **verbos o expresiones verbales que conecten unas entidades con otras**.

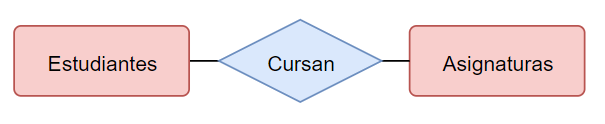
Debemos analizar cuidadosamente las anotaciones que se analizan a la hora de construir el modelo o, en vuestro caso, el enunciado que se os proporcionará. Una relación se identifica, generalmente, con los verbos o acciones como: vender, comprar, formado por, tener, etc.

Las relaciones se expresan mediante un verbo, procurando así formar frases que expresan un proceso de vínculo entre información contenida en las entidades, considerando que las entidades son sustantivos que actúan como sujeto y complemento cuando se asocian.

Ejemplo:

***Los/las estudiantes cursan asignaturas / Las asignaturas son cursadas por estudiantes.***





El siguiente paso será la representación de la cardinalidad (mínima y máxima) de las entidades participantes en cada relación y del tipo de correspondencia de la relación (1 a 1, 1 a muchos o muchos a muchos).

# Modelo conceptual II (modelo E/R)

## Entidades Fuertes y Débiles

Las entidades pueden ser clasificadas en dos grupos:

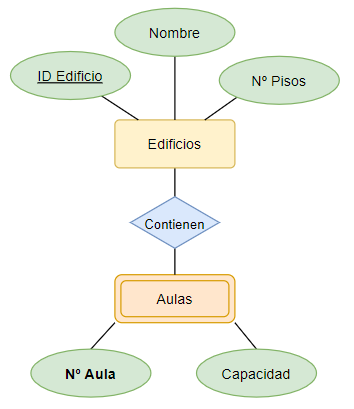
1. **Entidades Fuertes o Regulares:** Son aquellas que tienen existencia por sí mismas, es decir, su existencia no depende de la existencia de otras entidades. Por ejemplo, en una base de datos hospitalaria, la existencia de instancias concretas de la entidad **DOCTORES** no depende de la existencia de instancias u objetos de la entidad **PACIENTES**. En el modelo E/R las entidades fuertes se representan como hemos indicado anteriormente, con el nombre de la entidad encerrado dentro de un rectángulo.
2. **Entidades débiles:** Son aquellas cuya existencia e identificación depende de la existencia de otras instancias de entidad.

**Entidad Débil**: Es un tipo de entidad cuyas propiedades o atributos no la identifican completamente, sino que sólo la identifican de forma parcial. Esta entidad debe participar en una relación que ayude a identificarla.

En el modelo E/R una entidad débil se representa con el nombre de la entidad encerrado en un rectángulo doble. En el gráfico se muestra la representación de la entidad **AULAS**.

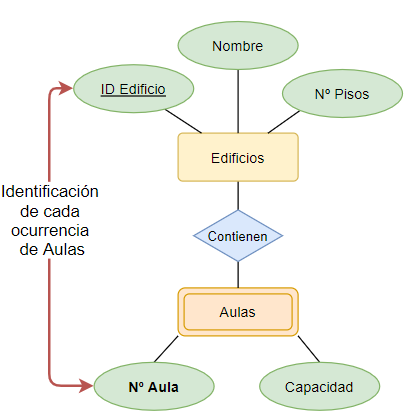


Por ejemplo, consideremos las entidades **EDIFICIO** y **AULA**. Supongamos que puede haber aulas identificadas con la misma numeración pero en edificios diferentes. La numeración de cada aula no identificará completamente cada una de ellas. Para poder identificar completamente un aula es necesario saber también en qué edificio está localizada. Por tanto, la existencia de una instancia de una entidad débil depende de la existencia de una instancia de la entidad fuerte con la que se relaciona.



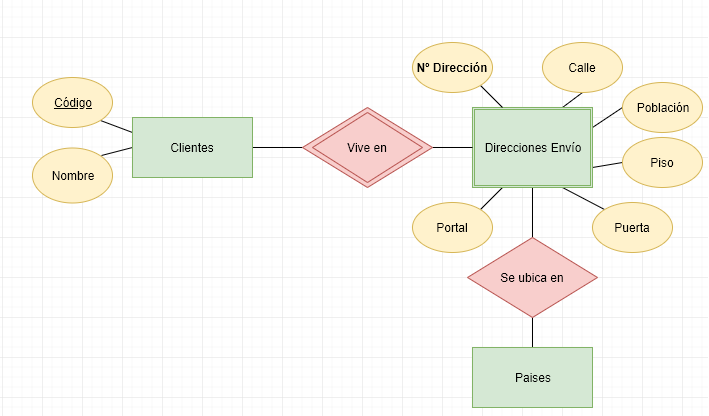
### Clave primaria en entidades débiles

Una entidad débil no tiene clave primaria, sino tan sólo un atributo discriminador y, por tanto, necesita obligatoriamente la clave primaria de la entidad fuerte para poder identificar de manera única sus ocurrencias de entidad. En este caso, la clave primaria de la entidad débil se forma por unión de la clave primaria de la entidad fuerte con el mencionado atributo discriminador. Este atributo irá en **negrita**.



### Entidad débil participante en dos relaciones

Cuando la entidad débil participa en dos relaciones se debe usar un **doble rombo** para indicar cuál es la relación de dependencia.

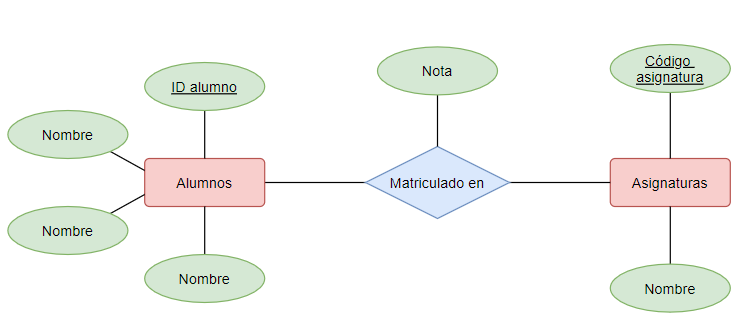


## Atributos complejos

### Atributos de una relación

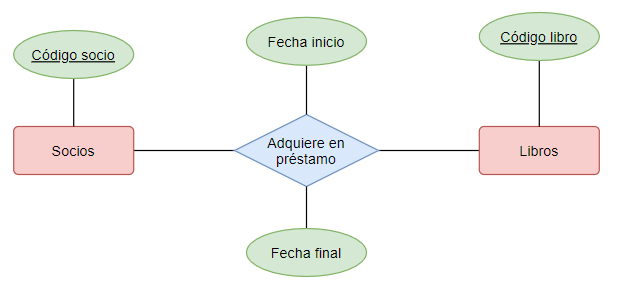
Las relaciones pueden tener atributos propios. Para ilustrar esta situación, observa el siguiente ejemplo.

Consideremos la relación **CURSA** entre las entidades **ALUMNO** y **ASIGNATURA**. Podríamos asociar a la relación **CURSA** un atributo **nota** para especificar la nota que ha obtenido un alumno/a en una determinada asignatura. En el modelo Entidad/Relación la representación de atributos asociados a relaciones es exactamente igual a la que utilizábamos para entidades: un círculo blanco conectado con una línea a la relación y junto a él, el nombre del atributo.

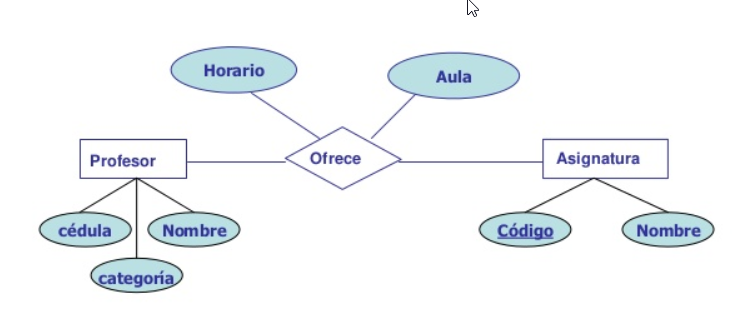


Otro ejemplo típico son las relaciones que representan **históricos**. Este tipo de relaciones suele constar de datos como fecha y/o hora y se da cuando se requiere almacenar la relación entre dos ocurrencias más de una vez en el tiempo.

Por ejemplo, un socio de una biblioteca puede tomar prestado un mismo libro en diferentes momentos de su vida. La biblioteca desea almacenar todos los préstamos, no sólo el más reciente. Es por ello que la relación entre los tipos de entidad LIBRO y SOCIO posee los atributos “Fecha\_inicio” y “Fecha\_fin” que especifican el periodo de tiempo en el cual un socio tiene un libro.



Otro ejemplo: Un profesor puede impartir diferentes asignaturas pero, además, puede impartir la misma asignatura a distintos grupos y, por lo tanto en diferentes aulas a diferentes horas. Por tanto una ocurrencia de la entidad Profesor (ej. Luis Dorado) puede estar relacionada dos veces con la misma ocurrencia de Asignatura (BBDD).



Serán los atributos de relación (Horario y Aula) los que distingan cada relación.

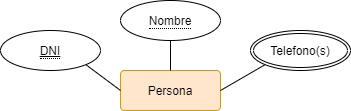
Para sucesos puntuales, es decir, sin duración, bastara con un solo a tributo de este tipo, mientras que para poder almacenar hechos que transcurren en un periodo de tiempo determinado necesitaremos una "fecha\_inicio” y una "fecha\_fin"

Por otro lado, podemos analizar si los datos que se pretenden almacenar van a constituir una base de datos histórica o si, por el contrario, sólo nos interesa el estado actual de los datos. Esta decisión se materializará a la hora de pasar el modelo E/R a modelo relacional.

### Atributos multivaluados

Son atributos que pueden poseer varios valores simultáneamente (siempre respetando su dominio). Por ejemplo, si necesitamos almacenar varios e-mail de una misma persona entonces deberemos utilizar un atributo multivaluado. Igual sucede con el teléfono. Si sólo necesitamos almacenar un sólo valor utilizaremos un atributo ordinario.

La **representación gráfica** de un atributo multivaluado es similar al del atributo ordinario pero usando una doble línea para el contorno de la elipse.



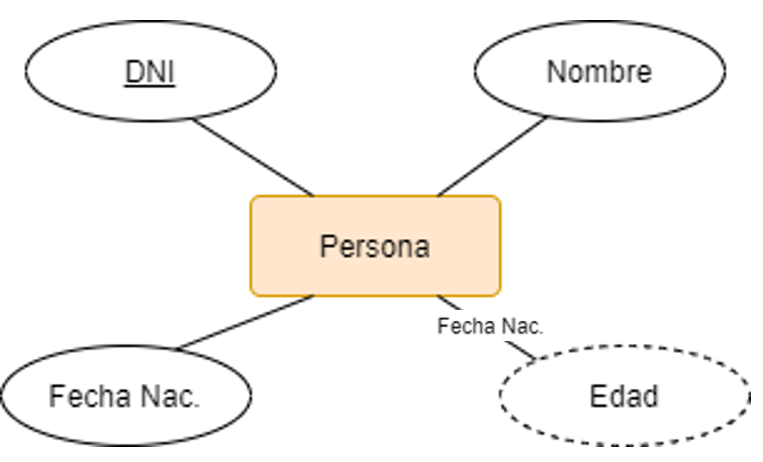
### Atributos derivados/calculados.

Un atributo **derivado** es aquel que puede calcularse a partir de otros. Por ejemplo, si tenemos la entidad PERSONA con los atributos DNI, Nombre, Fecha\_Nacimiento y Edad, el último atributo (Edad) puede obtenerse a partir de otro atributo (la fecha de nacimiento) y es, por lo tanto, redundante. Este tipo de atributos no se almacenan sino que se calculan en el momento de realizar la consulta a la BBDD. Además no tendría sentido almacenarlos porque su valor debería modificarse cada vez que variaran los atributos de los que depende.

Todos los atributos que no son derivados se consideran atributos almacenados.

La **representación gráfica** de un atributo calculado es similar al del atributo ordinario pero usando una línea discontinua para el contorno de la elipse y especificando en el nexo el atributo del que depende.

En la siguiente figura pueden apreciarse la representación gráfica del atributo derivado “edad”:



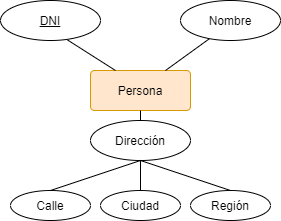
### Atributos simples o compuestos.

**Atributo simple o atómico**: es un atributo que no puede dividirse en otras partes o atributos, presenta un único elemento. No es posible extraer de este atributo partes más pequeñas que puedan tener significado. Un ejemplo de este tipo de atributos podría ser el atributo **dni** de la entidad **PERSONA** del gráfico.

**Atributo compuesto**: son atributos que pueden ser divididos en subpartes, éstas constituirán otros atributos con significado propio. Es decir, son aquellos que agrupan en sí mismos, por afinidad o por forma de uso, más de un atributo. Por ejemplo:

* Por su forma habitual de utilización, el atributo “dirección” engloba los atributos calle, número y ciudad.
* Por su significado, el atributo “nombre” de una entidad PERSONA engloba nombre de pila, primer apellido y segundo apellido.

La **representación gráfica** de un atributo compuesto se realiza dibujando el atributo compuesto y, unidos a él, los atributos que forman parte de él. En la siguiente figura pueden apreciarse la representación gráfica del atributo compuesto “dirección”:



**Ojo**: Si un atributo tiene sus propias propiedades también podría convertirse en una ENTIDAD. **Solo en casos muy particulares se usan atributos compuestos.**

### *Ampliación: Atributos obligatorios/opcionales.*

**Atributo obligatorio**: es aquel que ha de estar siempre definido para una entidad o relación. Por ejemplo, para la entidad PERSONA será necesario tener algún atributo que identifique cada ocurrencia de entidad, podría ser su DNI. Una **clave o llave** es un atributo obligatorio.

**Atributo opcional**: es aquél que podría ser definido o no para la entidad. Es decir, puede haber ocurrencias de entidad para las que ese atributo no esté definido o no tenga valor.

## Cardinalidad de las relaciones y entidades

¿Qué es eso de la cardinalidad? En matemáticas, el cardinal de un conjunto es el número de elementos que lo forman. Este concepto puede extrapolarse a las relaciones con las que estamos tratando.

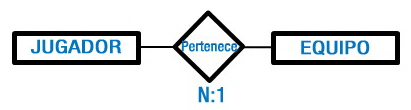
Una Relación queda caracterizada por tres propiedades:

* **Nombre**: Como todo objeto en el modelo E/R las relaciones deben tener unnombre que las identifique unívocamente.
* **Cardinalidad de la relación**: Número máximo de ocurrencias de cada tipo de entidad quepueden intervenir en una ocurrencia de relación. Dicho de otra manera, número de máximo ocurrencias de cada entidad que participa en la relación. Será un valor para cada una.
* **Cardinalidad de las entidades**: Número máximo **y mínimo** de ocurrencias de cada tipo de entidad quepueden intervenir en una ocurrencia de relación. Dicho de otra manera, número de máximo ocurrencias de cada entidad que participa en la relación. **La cardinalidad máxima será la calculada en la cardinalidad de la relación.**

### Cardinalidad de las relaciones

**Cardinalidad de una relación**: Es el número máximo de ocurrencias de cada entidad que pueden intervenir en una ocurrencia de relación. La cardinalidad vendrá expresada siempre para relaciones entre dos entidades. Dependiendo del número de ocurrencias de cada una de las entidades pueden existir relaciones **uno a uno, uno a muchos, muchos a uno y muchos a muchos**.

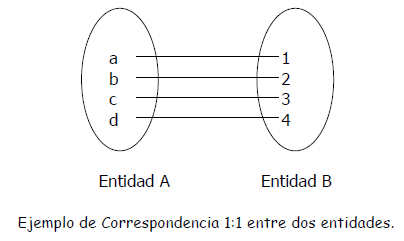
Observa el siguiente ejemplo, la cardinalidad indicará el número de ocurrencias de la entidad **JUGADOR** que se relacionan con cada ocurrencia de la entidad **EQUIPO** y viceversa. Podríamos hacer la siguiente lectura: un jugador pertenece a uno y solo a un equipo y a un equipo estar formado por varios jugadores y como mínimo 1.



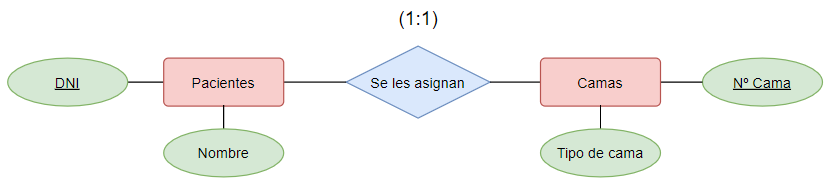
Una posible representación de la cardinalidad de las relaciones es la que hemos visto en el ejemplo anterior, aunque existen otras. Podríamos representar los cuatro tipos de cardinalidades mediante las etiquetas **1:1**, **1:N**, **N:1**, **N:M** que se leerían respectivamente: uno a uno, uno a muchos, muchos a uno y muchos a muchos.

Veamos en detalle el significado de cada una de estas cardinalidades…

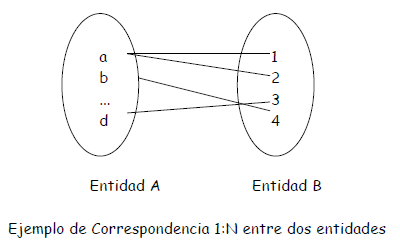
#### Relaciones uno a uno (1:1)



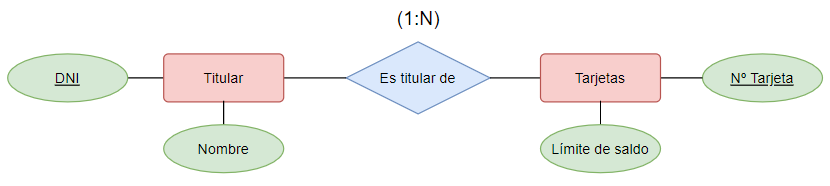
Sean las entidades A y B, una instancia u ocurrencia de la entidad A se relaciona únicamente con otra instancia de la entidad B y viceversa. Por ejemplo, para cada ocurrencia de la entidad PACIENTES sólo habrá una ocurrencia relacionada de la entidad CAMAS y viceversa. O lo que es lo mismo, a un paciente se le asigna una sola cama y una cama solo puede estar asignada a un paciente.



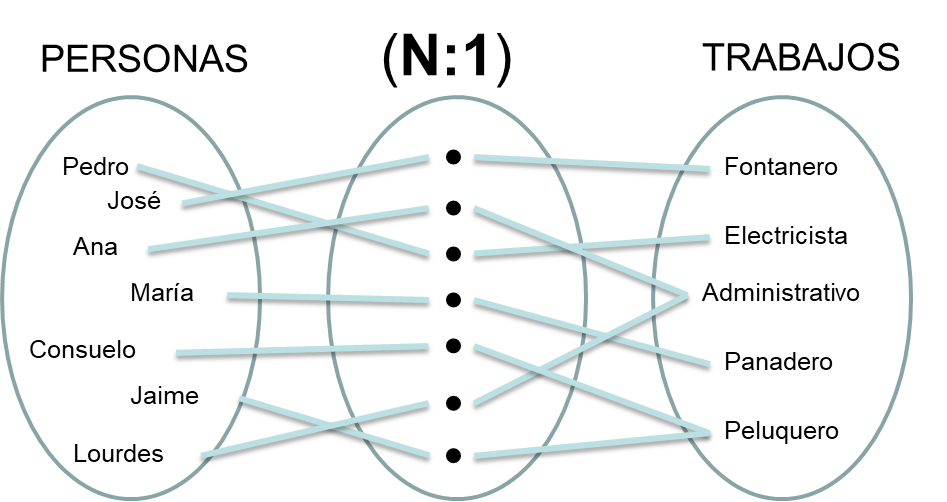
#### Relaciones uno a muchos (1:N)



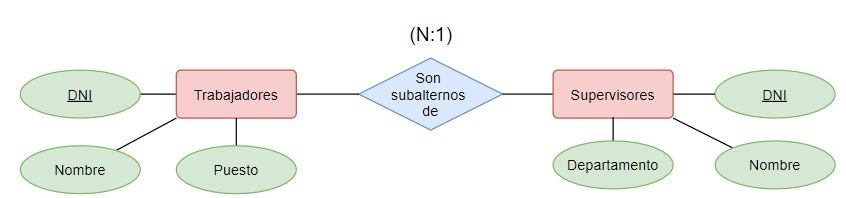
Sean las entidades A y B, una ocurrencia de la entidad A se relaciona con muchas ocurrencias de la entidad B y una ocurrencia de la entidad B sólo estará relacionada con una única ocurrencia de la entidad A. Por ejemplo, para cada ocurrencia de la entidad TITULARES puede haber varias ocurrencias de la entidad TARJETAS y para varias ocurrencias de la entidad TARJETAS sólo habrá una ocurrencia relacionada de la entidad TITULARES. O lo que es lo mismo, un cliente puede ser titular de varias tarjetas y una tarjeta sólo puede tener como titular un único cliente.



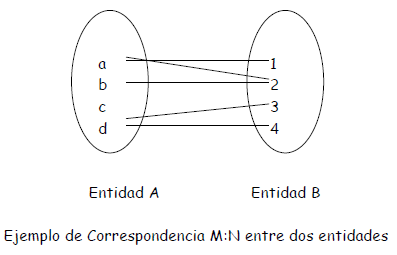
#### Relaciones muchos a uno (N:1)



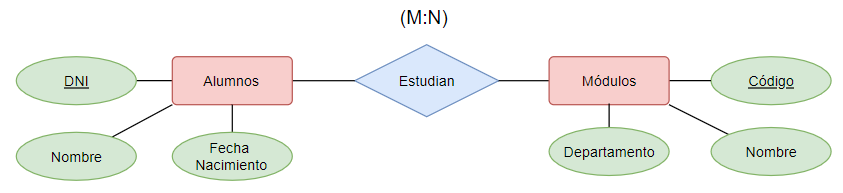
Sean las entidades A y B, una ocurrencia de la entidad A está asociada con una única ocurrencia de la entidad B y un ejemplar de la entidad B está relacionado con muchas ocurrencias de la entidad A. Por ejemplo cada trabajador tiene un solo supervisor pero cada supervisor tiene varios empleados a su cargo.



#### Relaciones muchos a muchos (N:M)



Sean las entidades A y B, un ejemplar de la entidad A está relacionado con muchas ocurrencias de la entidad B y viceversa. Por ejemplo, cada alumno estudia varios módulos y cada módulo es cursado por varios alumnos.



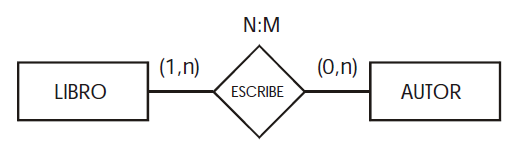
### Cardinalidad de entidades

Podemos definir la **Cardinalidad de una Entidad** como el número mínimo y máximo de ocurrencias de una entidad que pueden estar relacionadas con una ocurrencia de otra entidad que participan en el tipo de Relación. Su representación gráfica es una etiqueta del tipo (0,1), (1,1), (0,n) o (1,n) según corresponda. El significado del primer y segundo elemento del paréntesis corresponde a **(cardinalidad mínima, cardinalidad máxima)**:

* Si toda ocurrencia de la entidad A debe estar asociada con al menos una ocurrencia de la entidad B a la que está asociada por una determinada Relación, se dice que la clase de pertenencia es **obligatoria**, es decir, la cardinalidad mínima es 1.
* Por el contrario, si no toda ocurrencia de la entidad A necesita estar asociada con alguna ocurrencia de la entidad B asociada, se dice que la clase de pertenencia es **opcional**, es decir, la cardinalidad mínima es 0.
* La cardinalidad máxima puede ser uno, otro valor concreto mayor que uno (tres, por ejemplo) o muchos (se representa con n).

Veámoslo más claro a través del siguiente ejemplo: un libro puede estar escrito por ninguno (anónimo), uno o varios autores. Esto se representa con la cardinalidad (0,n).

Por otra parte, un autor escribe al menos un libro y puede escribir varios. Esto se representa con la cardinalidad (1,n).

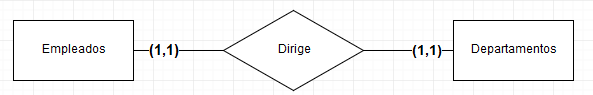


Ten en cuenta que cuando se representa la cardinalidad de una entidad, el paréntesis y sus valores han de colocarse junto a la entidad con la que se relaciona. Es decir **en el lado opuesto a la relación**.

Por ejemplo, en *un departamento pueden trabajar* ***uno o más*** *empleados* pero *un empleado sólo tiene que trabajar en* ***un y solo en un*** *departamento.*



Por ejemplo, *un departamento es dirigido* ***sólo por un*** *empleado* y *un empleado* ***sólo*** *puede dirigir* ***un*** *departamento*.

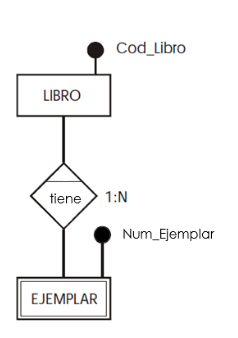


### *Ampliación: Cardinalidad en relaciones de debilidad*

La relación entre una entidad fuerte y una débil siempre tendrá **cardinalidad N:1** o **1:N**, según si la relación fuerte se encuentra a la izquierda o a la derecha. En una relación con cardinalidad N:M nunca habrá entidades débiles. La razón es que la supuesta ocurrencia de la entidad débil que se tuviera que borrar podría estar asociada a más de una ocurrencia de la supuesta entidad fuerte, lo que implicaría la imposibilidad de su borrado, hecho éste en clara contraposición con la definición de entidad débil.

Para representar gráficamente este tipo de relaciones, se utiliza el mismo rombo que cualquier otra relación, pero se dibuja partido por una raya horizontal.

En la figura siguiente puede apreciarse un ejemplo de este tipo de entidades. En este caso, el atributo Num\_Ejemplar por sí solo no permite distinguir cada una de las ocurrencias de la entidad EJEMPLAR (porque sus valores se pueden repetir para ejemplares de libros distintos), es decir, Num\_Ejemplar no es la clave de la entidad EJEMPLAR. La clave primaria será Cod\_Libro como clave primaria de la entidad fuerte LIBRO más Num\_Ejemplar como atributo discriminador de la entidad EJEMPLAR.

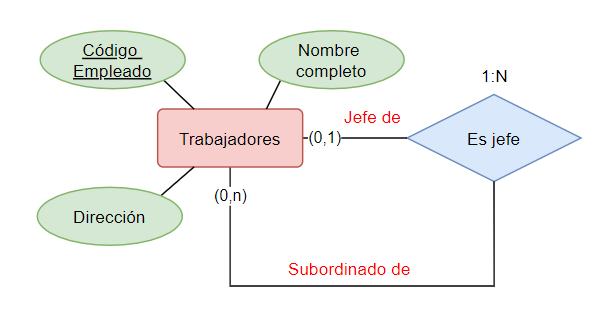


## Relaciones complejas

### Relaciones reflexivas

Son relaciones unarias y, por tanto, del tipo de Relación sólo participa un único tipo de entidad.

Ejemplo: Un trabajador puede ser jefe de ningún trabajador o puede serlo de varios trabajadores, mientras que un trabajador sólo es dirigido por ninguno o un trabajador.



Si hemos encontrado alguna relación  reflexiva, hemos de representar en nuestro esquema los **roles** desempeñados por la entidad en dicha relación.

### Relaciones ternarias

A diferencia de las entidades ordinarias o **binarias**, estas relacionan **tres entidades**.

Curso

Aula

Imparte

Profesor

Una relación ternaria une cada ocurrencia de una entidad se relaciona con otras dos ocurrencias de cada entidad.

Dicho de otra manera cada ocurrencia de una relación ternaria conecta tres ocurrencias de tres entidades.

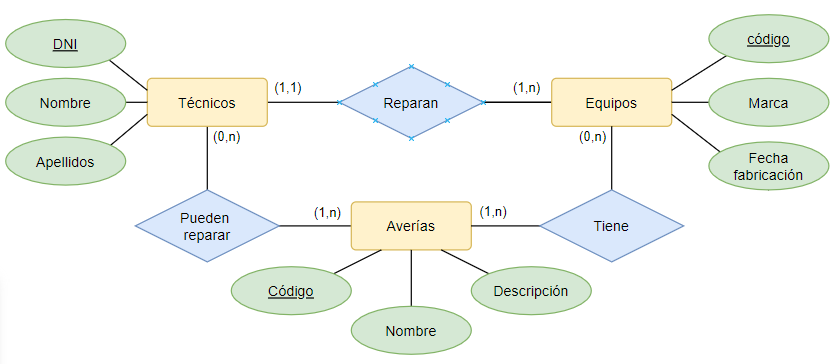
**Características**

* Representan relaciones que requieren de **tres relaciones binarias uniendo entre sí 3 entidades**.
* Muchas veces estas relaciones se representan más fácilmente como **relaciones ternarias** que como binarias.
* Se debe elegir la representación que **modele mejor la semántica** del problema, es decir, que capture mejor la relación existente en la realidad.
* Siempre se puede encontrar una **representación ternaria usando tres relaciones binarias que unan tres entidades.** Depende del gusto del diseñador.
* Lo importante es verificar que el diseño represente los requerimientos del problema.
* **Nunca** se debe usar para modelar una relación de **tres entidades con solo DOS relaciones**.

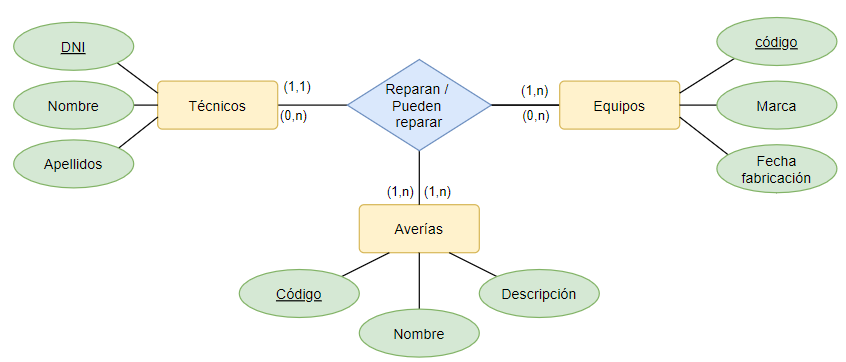
**Ejemplo de enunciado**

Los equipos sufren averías. Estos equipos son reparados por técnicos. Además los técnicos están habilitados para arreglar una serie de averías

**Ejemplo con relaciones binarias**



**Mismo ejemplo con relaciones ternarias**



**Ejemplo de relación ternaria errónea**

|  |  |  |
| --- | --- | --- |
|  |  |  |

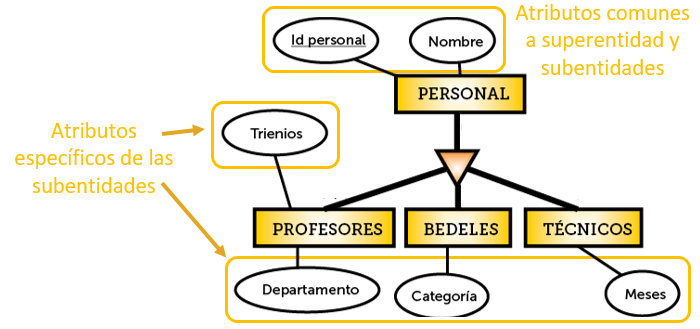
## Jerarquías

* Cuando estamos diseñando una base de datos puede que nos encontremos con **conjuntos de entidades que posean características comunes**, lo que permitiría crear un tipo de entidad de nivel más alto que englobase dichas características. A esto se le llama **generalización**.
* Y a su vez, puede que necesitemos **dividir una entidad en un conjunto de entidades** por tener éstas, características diferenciadoras. A esto se le llama **especialización**.
* Podemos decir que al aplicar una generalización o una especialización establecemos **una jerarquía**.

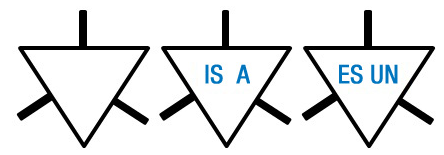
Este proceso de **refinamiento ascendente/descendente**, permite expresar mediante jerarquías la existencia de tipos de entidades de nivel superior que engloban a conjuntos de entidades de nivel inferior. A los conjuntos de entidades de nivel superior también se les denomina **superentidad**, superclase, supertipo o padre. A los conjuntos de entidades de nivel inferior se les denomina **subentidad**, subclase, subtipo o hijo.

**¿Cómo detectamos una jerarquía**? Podremos identificar una jerarquía cuando encontremos una serie de atributos comunes a un conjunto de entidades, y otros atributos que sean específicos. Los atributos comunes conforman la superclase o supertipo y los atributos específicos la subclase o subtipo.

Las jerarquías se caracterizan por un concepto que hemos de tener en cuenta, la **herencia**. A través de la herencia los atributos de una superclase de entidad son heredados por las subclases. Si una superclase interviene en una relación, las subclases también lo harán.



¿Cómo se **representa** una generalización o especialización? Existen varias notaciones, pero hemos de convenir que la relación que se establece entre una superclase de entidad y todos sus subtipos se expresa a través de las palabras ES UN, o en notación inglesa IS A, que correspondería con ES UN TIPO DE. Partiendo de este punto, una jerarquía se representa mediante un triángulo invertido, sobre él quedará la entidad superclase y conectadas a él a través de líneas rectas, las subclases.



### Restricciones

En las jerarquías, los **atributos comunes** a todos los subtipos se asignan al **supertipo**, mientras que los atributos específicos se asocian al subtipo correspondiente.

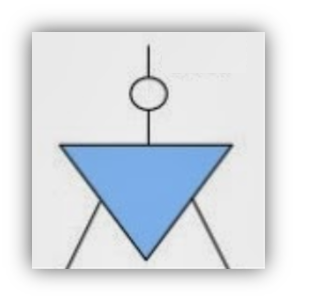
Las **relaciones que afectan a todos los subtipos se asocian al supertipo**, dejándose para los subtipos las relaciones específicas en las que el correspondiente subtipo participa.

Se pueden distinguir cuatro tipos de generalización, atendiendo a si los subtipos se solapan o son disjuntos, y a si la unión de los subtipos recubre o no el supertipo:

* **Totalidad**: una jerarquía será total si todo ejemplar de la superclase pertenece a alguna de las subclases.
* **Parcialidad**: una jerarquía será parcial si no todos los ejemplares de la superclase pertenecen a alguna de las subclases.
* **Solapamiento**: una jerarquía presentará solapamiento si un mismo ejemplar de la superclase puede pertenecer a más de una subclase.
* **Exclusividad**: una jerarquía presentará exclusividad si un mismo ejemplar de la superclase pertenece sólo a una subclase.

### Representación

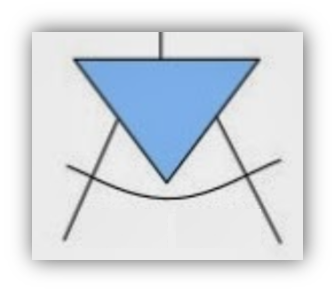
1. **Jerarquía Total**: Se define por **un círculo** entre la superentidad y el triángulo.



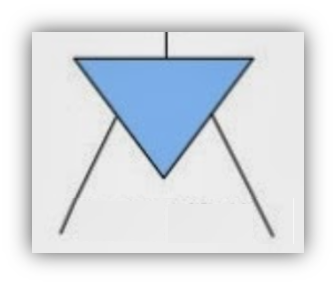
1. **Jerarquía Parcial**: **Ausencia de círculo** entre la superentidad y el triángulo.



1. **Jerarquía Exclusiva**: Se define por **un arco cubriendo los enlaces** entre triángulo y subentidades.

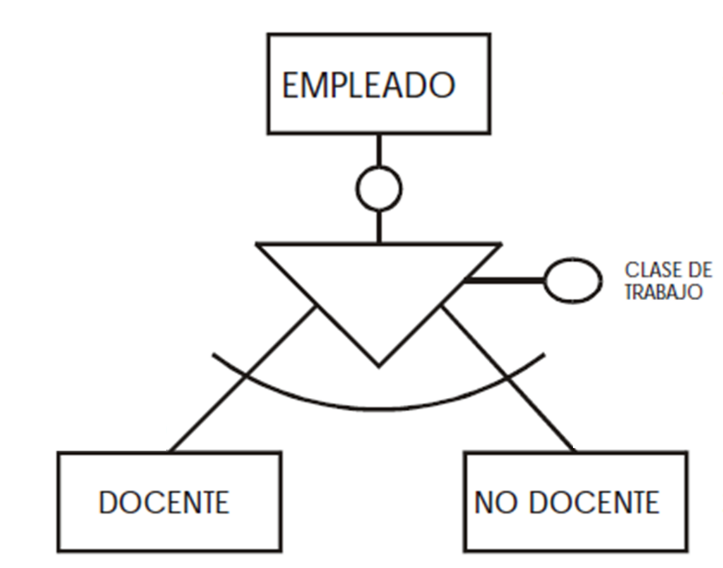


1. **Jerarquía Solapada**: Ausencia de arco.



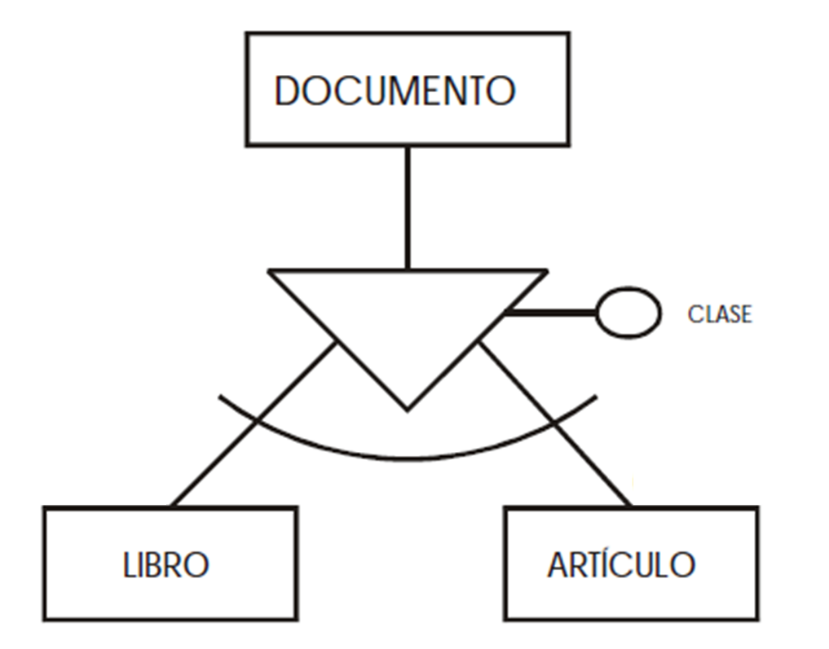
Las restricciones anteriores dan lugar a cuatro combinaciones distintas:

### Jerarquía total con exclusividad



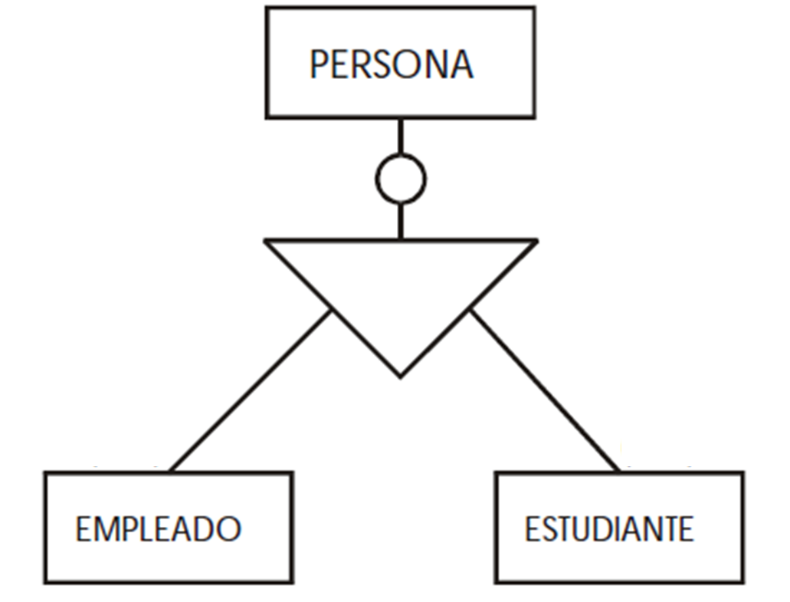
* Tanto un docente como un no docente son empleados.
* **Exclusividad**: Un mismo empleado no puede ser a la vez docente y no docente.
* **Totalidad**: Todo empleado tiene que ser obligatoriamente un docente o un no docente

### Jerarquía parcial con exclusividad



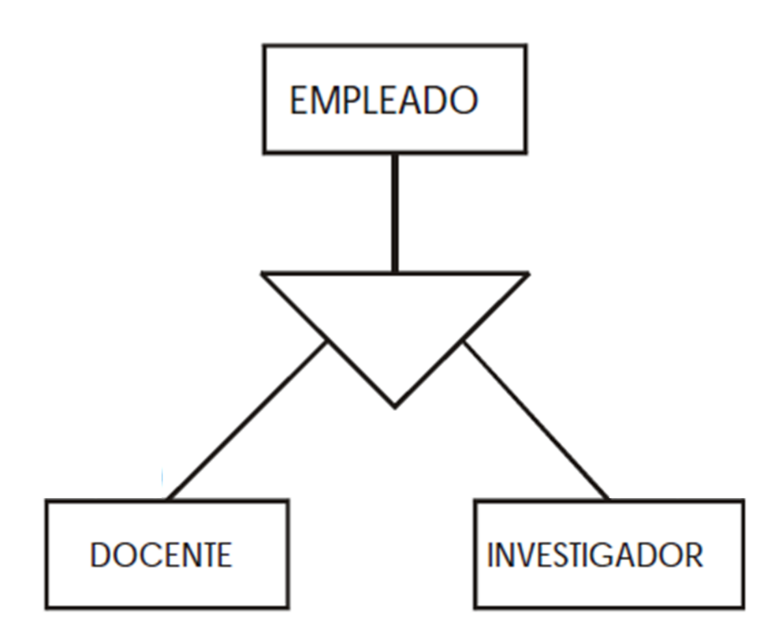
* Tanto un artículo como un libro son documentos.
* **Exclusividad**: Un mismo documento no puede ser a la vez un artículo y un libro.
* **Parcialidad**: Puede haber documentos que no sean ni artículos ni libros.

### Jerarquía total con solapamiento



* Tanto un empleado como un estudiante son personas.
* **Solapamiento**: Una misma persona puede ser estudiante a la vez que empleado
* **Totalidad**: Toda persona en nuestra BD tiene que ser obligatoriamente un estudiante y/o empleado

### Jerarquía parcial con solapamiento



* Tanto un docente como un investigador son empleados.
* **Solapamiento**: Un mismo empleado puede ser, y en general lo es, docente a la vez que investigador.
* **Parcialidad**: Puede haber empleados que no sean docentes ni investigadores.

### Identificación de jerarquías

Como se ha comentado anteriormente, es probable que existan entidades con características comunes que puedan ser generalizadas en una entidad de nivel superior o superclase (jerarquía de generalización). Pero también, puede ser necesario expresar en el esquema las particularidades de diferentes ejemplares de un tipo de entidad, por lo que se crearán subclases o subtipos de una superclase o supertipo (jerarquía de especialización). Para ello, habrá que analizar con detenimiento el documento de especificación de requerimientos.

Como normal general, crearemos subtipos en una jerarquía sólo si se cumple alguna de las siguientes condiciones:

* **Los subtipos tienen atributos diferenciadores.**
* **Los subtipos tienen relaciones diferenciadoras.**

Sólo se representarán los subtipos que cumplan alguna de las anteriores condiciones.

Si se identifica algún tipo de jerarquía, se deberá representar adecuadamente según el tipo de notación elegida, determinando si la jerarquía es total/parcial o exclusiva/con solapamiento.

## Control de redundancia

Una de las principales razones por las que las bases de datos aparecieron fue la eliminación de la redundancia en los datos ¿Y qué es la redundancia?

**Redundancia**: reproducción, reiteración, insistencia, reincidencia, reanudación. En bases de datos hace referencia al almacenamiento de los mismos datos varias veces en diferentes lugares.

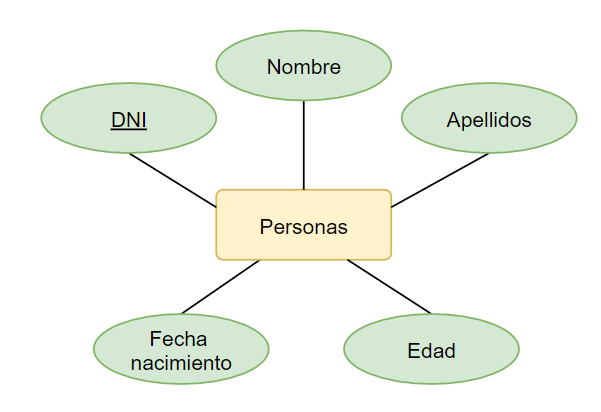
La redundancia de datos puede provocar problemas como:

* **Aumento de la carga de trabajo**: al estar almacenado un dato en varios lugares, las operaciones de grabación o actualización de datos necesitan realizarse en varias ocasiones.
* **Gasto extra de espacio de almacenamiento**: al estar repetidos, los datos ocupan mayor cantidad de espacio en el medio de almacenamiento. Cuanto mayor sea la base de datos, más patente se hará esta problema.
* **Inconsistencia**: se produce cuando los datos que están repetidos, no contienen los mismos valores. Es decir, se ha actualizado su valor en un lugar y en otro no, por lo que no se sabría qué dato es válido y cual erróneo.

Para que una base de datos funcione óptimamente, hay que empezar realizando un buen diseño de ella. Es imprescindible que nuestros diagramas E/R controlen la redundancia y, para ello, debemos analizar el esquema y valorar qué elementos pueden estar incorporando redundancia a nuestra solución.

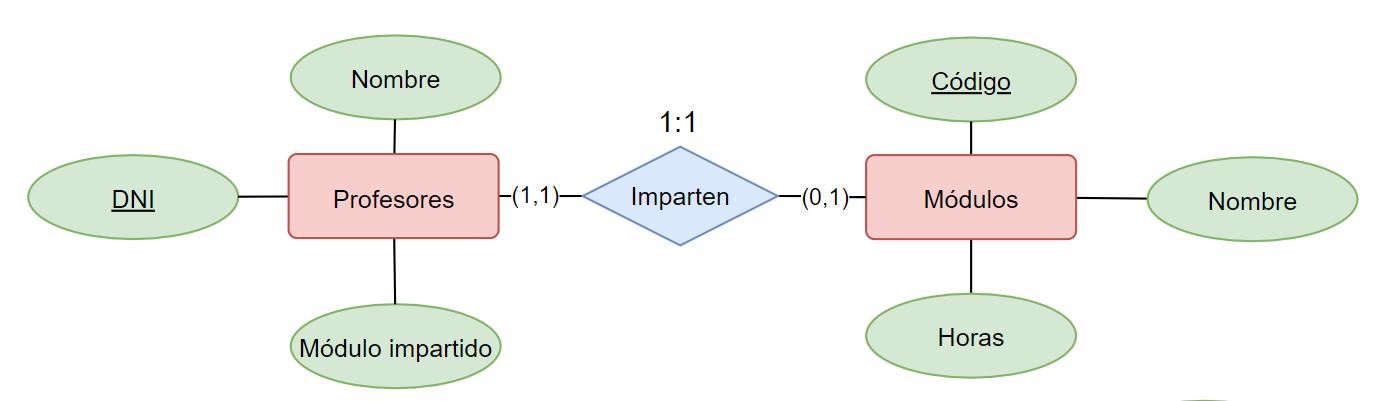
¿Dónde buscamos indicios de redundancia en nuestros esquemas? Existen lugares y elementos que podrían presentar redundancia, por ejemplo:

1. **Atributos redundantes** cuyo contenido se calcula en función de otros. Un atributo derivado puede ser origen de redundancia.



*Edad se calcula a partir de la fecha de nacimiento y no es necesario almacenarlo.*

1. Atributos cuya información **ya se halla en una entidad relacionada**.



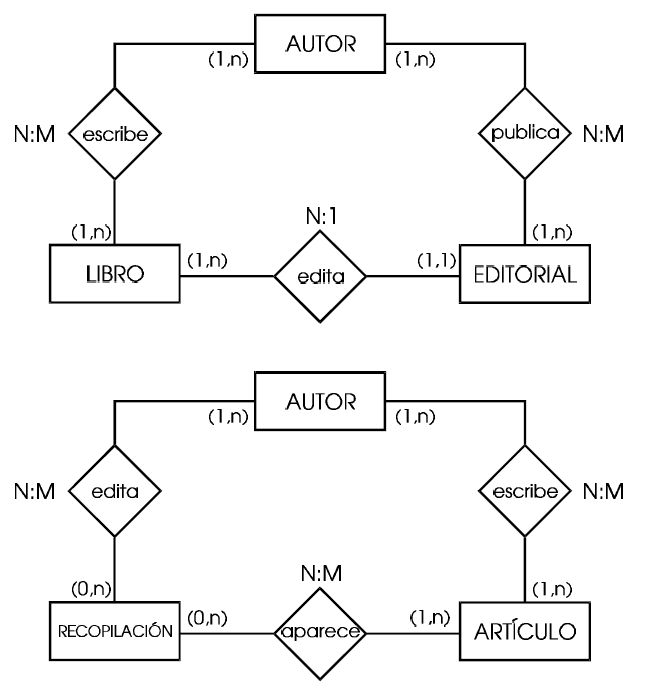
*El módulo que imparte cada profesor se almacena en la entidad Módulos.*

1. Varias entidades unidas circularmente o cíclica a través de varias relaciones, es lo que se conoce como **un ciclo**. En caso de existir un ciclo, deberemos tener en cuenta las siguientes condiciones, antes de poder eliminar dicha relación redundante:

* Que el significado de las relaciones que componen el ciclo sea el mismo.
* Que si eliminamos la relación redundante, el significado del resto de relaciones es el mismo.
* Que si la relación eliminada tenía atributos asociados, éstos puedan ser asignados a alguna entidad participante en el esquema, sin que se pierda su significado.

Pero hay que tener en cuenta que **no siempre que exista un ciclo estaremos ante una redundancia**. Es necesario analizar detenidamente dicho ciclo para determinar si realmente existe o no redundancia.

Para finalizar, una apreciación. **No toda redundancia es perjudicial.** Existen ciertas circunstancias y condiciones en las que es conveniente (sobre todo a efectos de rendimiento) introducir cierta **redundancia controlada** en una base de datos. Por ejemplo, si el método de cálculo del valor de un determinado atributo derivado es complejo (varias operaciones matemáticas o de cadenas de caracteres, varios atributos implicados, etc.) y ralentiza el funcionamiento de la base de datos, quizá sea conveniente definir dicho atributo desde el principio y no considerarlo como un atributo redundante. La incorporación o no de redundancia controlada dependerá de la elección que haga el diseñador.

Ejemplos:

En el **primer diagrama**, la relación “publica” es redundante, ya que podemos obtener dicha información a través del resto de relaciones. Podemos saber qué editoriales publican a un autor sabiendo qué libros ha escrito, y de cada libro, que editorial lo ha editado. También podemos saber qué autores ha publicado una editorial concreta sabiendo qué libros ha editado dicha editorial, y de cada libro, extrayendo sus autores. El resto de relaciones no son redundantes, ya que su información no se puede extraer a través de otras relaciones.

En el **segundo diagrama**, la relación “edita” NO es redundante. Un autor puede haber escrito un artículo que aparece en una recopilación que él ha editado. Pero también puede haber editado una recopilación en la que no aparezca ningún artículo escrito por él. No podemos saber qué recopilaciones ha editado solo con saber qué artículos ha escrito y en qué recopilaciones se encuentran. “Autor edita recopilación” NO es redundante.