

1. Caso Práctico

Caso Práctico

Una **compañía de seguros de automóviles** quiere crear una aplicación para gestionar el funcionamiento de su negocio. Contrata el trabajo con **SI Andalucía**, que se encargará de desarrollarla.

María, como casi siempre, es la encargada de realizar el análisis y diseño necesarios para poner en marcha la aplicación. Tras la reunión con los representantes de la empresa, a la que también acudió **José**, ha decidido que la aplicación va a sustentarse claramente sobre una base de datos, y lo primero que necesitan es hacer **un modelo conceptual de la misma**, un diseño, que es algo parecido a los planos sobre los que luego se implantará la base de datos con un **SGBD** concreto. En esa reunión, fundamentalmente han estado recogiendo los requerimientos que los encargados de la empresa le han dado sobre el funcionamiento de su negocio y que, en lo relativo a los datos, se refleja en una lista de supuestos semánticos, (se detallan en el ejemplo del apartado 7 de la unidad) a partir de los cuales deben iniciar el trabajo.

Tanto **María** como **José** son conscientes de que hacer un buen diseño de la base de datos les va a facilitar la tarea de desarrollar las aplicaciones que usarán esos datos, y también facilitará el mantenimiento de esas aplicaciones, de la base de datos, y de los propios datos que contiene. Por tanto, se proponen hacerlo con cuidado y meticulosidad, para que no se les escape ninguna información relevante, y poder cumplir con todos los requerimientos que los clientes les han planteado.

José le propone a **María** que aproveche la ocasión para que **Víctor** tome contacto con el modelo Entidad-Relación, que es el que van a usar, y que se vaya familiarizando con los conceptos de Entidades, Relaciones, Atributos, Cardinalidad, etc., al mismo tiempo que aprende a realizar por sí mismo el **diagrama E-R**, y el **esquema relacional** que de él se deriva a partir de los requerimientos que les plantean los clientes. A fin de cuentas, **María** empieza a tener mucho trabajo, y no le viene mal contar con una ayuda en sus tareas de análisis y diseño. **Carmen** suele ayudarle, pero también suele estar bastante atareada con la codificación de distintas aplicaciones, por lo que no siempre puede echarle una mano. Ha llegado la hora de formar a **Víctor** para que asuma nuevas responsabilidades profesionales dentro de la empresa, y es imprescindible que aprenda todo lo relativo a la modelización conceptual.

2. Introducción

Introducción

En la unidad anterior hemos visto cómo se realiza el análisis de los sistemas, recogiendo una serie de datos que se utilizan en los mismos.

- ¿Te has preguntado en algún momento cómo se representan esos datos?
- ¿Cómo podemos recoger toda la información relativa a una determinada porción de la realidad (la [semántica](#) de un determinado sistema de información)?

Pues la respuesta está en esta unidad, donde nos centraremos en el análisis de datos o modelización de datos desde el punto de vista de la ingeniería del software y del análisis estructurado.

La modelización de datos es una técnica del análisis que trata de representar los datos que aparecen en el sistema, sin molestarse por el procesamiento que se debe aplicar para transformar los datos.

Proporciona al analista y al diseñador una amplia **visión de los datos** y de las relaciones que gobiernan los datos. Dentro del contexto del análisis estructurado, se puede usar la modelización de datos para representar el contenido de los almacenes de datos (que vienen reflejados en el DFD) y de las relaciones que existen entre ellos.

La **modelización** de datos también proporciona una forma de estructurar los datos, eliminar inconsistencias y verificar el uso de los datos antes del diseño detallado.

Los modelos de datos son abstracciones mediante las cuales puede realizarse una representación de los problemas en estudio. Están basados en el uso de la abstracción, permitiendo una descripción del sistema y sus elementos como un todo sin que sea necesario describir cada uno de los elementos particulares ni cada uno de los estados del sistema.

De forma general, todos los **modelos** presentan:

- Un conjunto de **reglas** mediante las cuales puede ser representado gráficamente el problema. Permite representar objetos y sus relaciones así como las [restricciones](#) de los mismos.
- Un **pseudolenguaje** que describe las propiedades dinámicas y estáticas del sistema.
- Un conjunto de **restricciones** que limitan el ámbito en el cual el modelo puede ser utilizado.

Los modelos de datos permiten la **representación** del problema a tres niveles de abstracción diferentes. Tres visiones de un mismo problema, que permiten describir dicho problema de forma idealmente independiente para su representación y tratamiento.

Estos tres niveles de abstracción son:

- **conceptual,**
- **lógico**
- **físico.**

En esta unidad trataremos el primero de ellos.

3. Modelización conceptual de datos

Modelización conceptual de datos

¿Qué podemos representar desde el punto de vista **conceptual**?

En el nivel conceptual de abstracción de datos, son representados los tipos o clases de objetos, y sus relaciones desde un punto de vista estructural.

- En el nivel conceptual se representa un [modelo de datos](#) del sistema en el que se describen cada uno de los tipos de objetos o elementos del mismo.
- Para cada uno de estos tipos de objetos se describen sus **propiedades** y el dominio o tipo de datos básico en el cual pueden ser medidas, así como las restricciones o límites de los valores que pueden presentarse para cada una de estas propiedades.
- Además, son descritas las **relaciones** entre los tipos de objetos, relaciones jerárquicas o no, apoyándose para ello en los principios de la abstracción.

A este nivel se representa el problema tal y como es; es decir, se representa el mundo real del problema tal y como se percibe, sin tener en cuenta cómo este problema puede ser representado para que sea entendido por los programas de ordenador (lo que se denomina diseño de datos).

Un modelo conceptual de un problema o la visión conceptual de éste, **es independiente de los procedimientos manuales o automatizados que se utilicen o se vayan a utilizar para el mantenimiento y tratamiento de la información** correspondiente al problema.

La visión **conceptual** sólo es dependiente de:

- Las características del problema o sistema que se desea representar.
- El detalle de la representación, que depende de la "parte" o "partes" del problema que se desea representar para su posterior tratamiento.

Pero el modelo conceptual es independiente de las herramientas y mecanismos (del [S.G.B.D.](#) y del hardware) que se vayan a utilizar para esa representación y tratamiento.

Ejemplos: El modelo Entidad/Relación, que es el más utilizado como veremos a continuación, y el modelo de estructura de datos.

Los modelos lógico y físico no son materia de nuestro módulo, los aprenderás en otros, pero a grandes rasgos para que te vayan sonando los definimos así:

- **El Modelo lógico representa el problema bajo las limitaciones impuestas por la representación y el tratamiento de la información que se vaya a realizar (del hardware y del software).** Es dependiente del SGBD. Describe los objetos lógicos, sus atributos y relaciones. Un ejemplo es el Modelo Relacional.
- **El Modelo físico representa el problema bajo las limitaciones impuestas por el soporte y los tratamientos utilizados, no como se ve en el mundo real. Describe los objetos físicos y sus relaciones para su posterior almacenamiento, recuperación y tratamiento.** Un ejemplo serían los scripts SQL

El Modelo lógico es dependiente del SGBD utilizado, por ello es necesario traducir el modelo conceptual al modelo lógico del SGBD y éste se encargará de generar su modelo físico.

Para saber más

Si quieres conocer algo más acerca de la modelización conceptual, selecciona este enlace:

[Modelizacion conceptual](#) [\[Versión en caché\]](#)

4. El Modelo Entidad-Relación

El Modelo Entidad-Relación.

CASO.

María le explica a Víctor que el primer paso para elaborar el diagrama E-R para la base de datos con la información de la compañía de seguros de automóviles, que representará la solución diseñada para esta base de datos en concreto, es:

- encontrar las **entidades** que intervienen en el problema,
- las **relaciones** existentes entre esas entidades, y
- los **atributos** que tienen esas entidades y esas relaciones.

Le comenta que la tarea no es siempre fácil, y que incluso puede ser bastante complicada, pero que todo consiste en seguir el proceso adecuado, y en gran medida, usar la lógica y el sentido común, algo que requiere haber comprendido perfectamente el problema y los requerimientos que nos proporcionan los clientes. Es la información de partida sobre la que debemos comenzar el trabajo. Pero también es necesario haber entendido la teoría del modelo Entidad-Relación.

Tras las oportunas explicaciones, Víctor cree haber comprendido bastante bien, los conceptos de Entidad y Relación, y sigue con atención las explicaciones de María sobre lo que podemos hacer para determinar cuales son las entidades y relaciones del problema, así como los atributos y sus cardinalidades

María y Víctor siguen completando el diseño de la base de datos que quedará reflejado en el correspondiente diagrama Entidad-Relación. La próxima tarea que les ocupa es encontrar todos los atributos significativos para cada una de las entidades y relaciones que han encontrado previamente.

María le explica el proceso necesario para identificar esos atributos, con las peculiaridades que deben tenerse en cuenta para el caso de las entidades y para el caso de las relaciones. (El proceso completo para el ejemplo de la compañía de seguros lo encontrarás en el apartado 7 de la unidad, concretamente en los subapartados 7.1., 7.2. y 7.3.)

Carmen le ha echado una mano a Víctor identificando todas las llaves candidatas de las entidades y relaciones, pero una vez que han hecho esa tarea, y que se ha asegurado de que Víctor ha entendido bien lo que es una llave candidata y lo que es una llave primaria o principal, lo ha dejado solo para que se encargue de elegir directamente las claves principales de las entidades del problema, y el resultado que Víctor ha obtenido y que le ha pasado a María es el que puedes ver en ejemplo del subapartado 7.4. de la unidad.

Una vez que han identificado convenientemente todos los elementos, el paso final es dibujar el diagrama E-R que representa el diseño de la base de datos que se ha hecho. Le comenta los símbolos que se emplean para representar cada uno de esos elementos, que existen distintas estrategias para crear el diagrama, y que una vez hecho hay que verificar que el diagrama cumple con todos los criterios de calidad establecidos para los diagramas E-R.

La tarea de dibujar el diagrama le lleva un tiempo a Víctor, que todavía no tiene demasiado dominio de esas estrategias que María le ha contado, y que ha tenido que hacer varios borradores a papel y lápiz antes de hacer el diseño definitivo, pero el resultado es satisfactorio para todos. José, que se ha encargado de revisarlo, ha felicitado a Víctor, y le ha dicho que aunque conviene que practique un poco con otros ejemplos más variados y más complejos, muy pronto estará en condiciones de ayudar a María con el diseño de los diagramas E-R en todos los proyectos. El resultado puedes verlo en el ejemplo del subapartado 7.5. de la unidad.

¿Qué vamos a representar mediante este modelo conceptual?

El modelo **Entidad-Relación** (o entidad-interrelación) fue propuesto por Peter Chen a mediados de los años setenta para la representación conceptual de los problemas y como un medio para representar la visión de un sistema de forma global. Se trata de un modelo muy extendido que ha experimentado a lo largo de los años una serie de ampliaciones, lo que ha originado un medio muy potente para la representación de los datos correspondientes a un problema.

Las características actuales de este modelo permiten la representación de cualquier tipo de sistema y a cualquier nivel de abstracción o refinamiento, lo cual da lugar a que se aplique tanto a la representación de problemas que vayan a ser tratados mediante un sistema computerizado como manual.

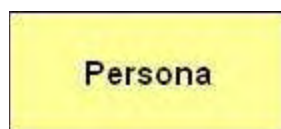
El modelo EntidadRelación, (a partir de ahora E/R), mediante un conjunto de símbolos, y haciendo uso de un conjunto reducido de reglas, representa los elementos que forman parte del sistema y las relaciones existentes entre ellos, siendo estos elementos descritos mediante un pseudolenguaje basado en una gramática sencilla.



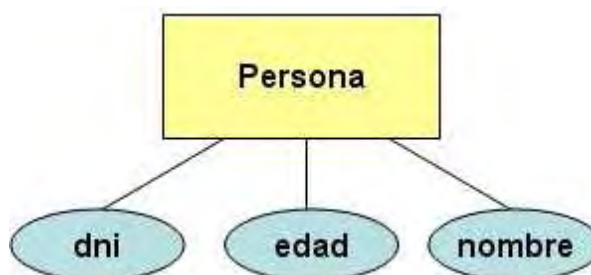
Los principales **elementos** del modelo E/R, como su nombre indica, son las entidades y las relaciones a las que hay que añadir los atributos. También se puede extender el modelo con el concepto de generalización entre entidades como veremos después.

Conceptos básicos del modelo E/R:

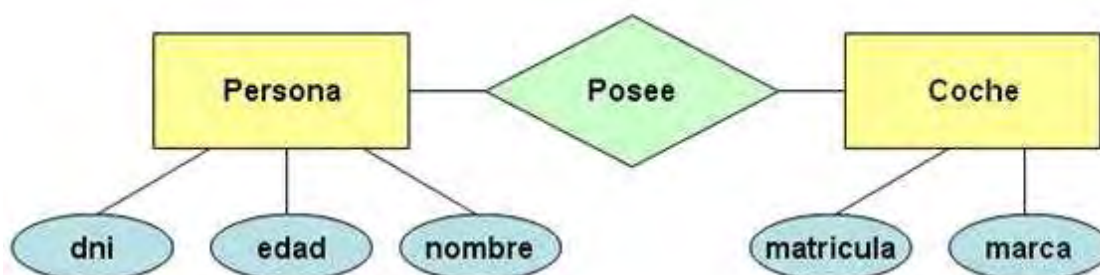
- **Entidad:** un tipo de objeto (un conjunto de objetos) definido en base a la agregación de una serie de atributos. Corresponde a la caracterización de objetos del mundo real, los cuales se distinguen del resto por el conjunto de atributos que poseen.



- **Atributo:** cada una de las propiedades o características de una entidad. Se representa gráficamente mediante un círculo con su nombre. Ejemplo: Para una entidad **Persona** tenemos los atributos **dni**, **edad**, **nombre**.



- **Relación o interrelación:** asociación entre ocurrencias de entidades. La suma de todas las ocurrencias de relación entre dos entidades son el conjunto de relaciones. Ejemplo: entre persona y coche existe la relación posee para indicar que una persona es poseedora de uno o varios coches.



- **Dominio:** conjunto de valores de datos que puede contener un atributo. Ejemplo: para la edad de una persona, es un número natural entre 0 y 99 (suponemos que no va a haber personas mayores de 99 años con coche)

Normalmente, se utilizan los términos Entidad y Relación cuando en realidad estamos hablando de Conjunto de Entidades y de Relaciones.

Para saber más

Si quieres conocer algo más acerca del trabajo del creador del Modelo Entidad-Relación, pulsa en los siguientes enlaces:

[Publicaciones de Peter Chen](#)

[Texto en el que aparece por primera vez el concepto de Modelo Entidad-Relación](#)

[Enlace al artículo original que publicó P.Chen para definir el modelo E/R](#) [\[Versión en caché\]](#)

5. Entidad

Entidad

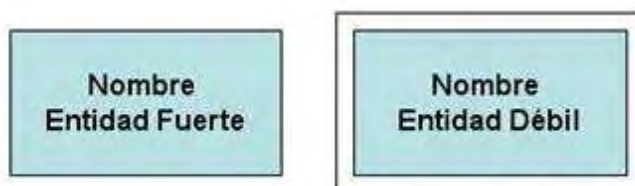
Se puede definir como aquel objeto (real o abstracto) acerca del cual queremos almacenar información en la base de datos. Según [ANSI](#), se define como "una persona, lugar, cosa, concepto o suceso, real o abstracto, de interés para la empresa" (por ejemplo, EMPLEADO).

Una entidad cumple las siguientes **propiedades**:

- Tiene **existencia propia**. Es decir, la entidad existe como un elemento que interviene en el comportamiento global del sistema.
- Es **distinguible** del resto de las entidades (objetos) que intervienen en el sistema.
- Las entidades de un mismo tipo están definidas en base a un mismo conjunto de atributos, cada uno de ellos definido en un mismo dominio.

Existen dos **tipos de entidades**:

- **Fuertes**, que son aquéllas que tienen existencia por sí mismas dentro del contexto del problema en estudio (por ejemplo **PROYECTO**, **DEPARTAMENTO**), y
- **Débiles**, cuya existencia depende de otra entidad (por ejemplo **FAMILIAR** depende de **EMPLEADO** y la desaparición de un empleado de la base de datos lleva consigo el que desaparezcan también todos los familiares que estaban a su cargo).



Las entidades **fuertes** se representan mediante un rectángulo, mientras que para las **débiles** se utilizan dos rectángulos concéntricos.

Para las entidades **débiles**, existen dos tipos de dependencia:

■ Dependencia en existencia (entre entidades).

Si desaparece una instancia del tipo de entidad fuerte deben desaparecer las instancias de la entidad débil que dependen de ella. Se nota con una "E" en la relación débil.

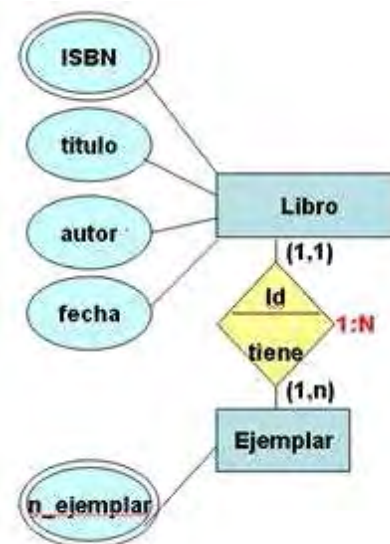
Es muy común que la dependencia en existencia se represente únicamente con las dos entidades involucradas, unidas mediante una línea y distinguiendo la entidad débil mediante un rectángulo doble.

- Si consideramos la entidad fuerte **EMPLEADO**, y la débil **FAMILIAR**, está claro que cada instancia de **FAMILIAR** queda perfectamente identificada sin necesidad de utilizar la clave primaria de la entidad **EMPLEADO**, aunque si eliminamos cualquier instancia de la entidad **EMPLEADO**, es evidente que no tiene sentido mantener las instancias de la entidad **FAMILIAR** que estaban asociadas dicha instancia.



■ Dependencia en identificación.

- Se produce cuando además de la dependencia en existencia, una instancia del tipo de entidad débil no se puede identificar por sí misma, y debe hacerse mediante la clave de la entidad fuerte asociada. Su clave es (clave_entidad_fuerte, clave_parcial). Se nota con una "ID" en la relación débil. Si consideramos la entidad fuerte **LIBRO**, y la débil **EJEMPLAR**, está claro que cada instancia de **EJEMPLAR** no se puede identificar únicamente mediante sus atributos propios, y exige añadir la clave de la entidad **LIBRO** de la que depende, es decir para identificar las instancias de un libro necesitamos el código del libro y un contador que nos diferencie cada instancia de **EJEMPLAR**.



Tanto las entidades fuertes como las débiles se nombran habitualmente con sustantivos en singular.

6. Descripción de entidades

Descripción de entidades

Como ya hemos visto en ocasiones anteriores en esta unidad, los nombres de los distintos componentes del modelo Entidad-Relación son muy intuitivos, y dan una idea muy explícita de su naturaleza.

¿Qué es para ti un **atributo**?

Un atributo es cualquier detalle que sirve para calificar, identificar, clasificar, cuantificar o expresar el estado de algo, en nuestro caso de una entidad, es decir un atributo es cualquier descripción de una característica de importancia.

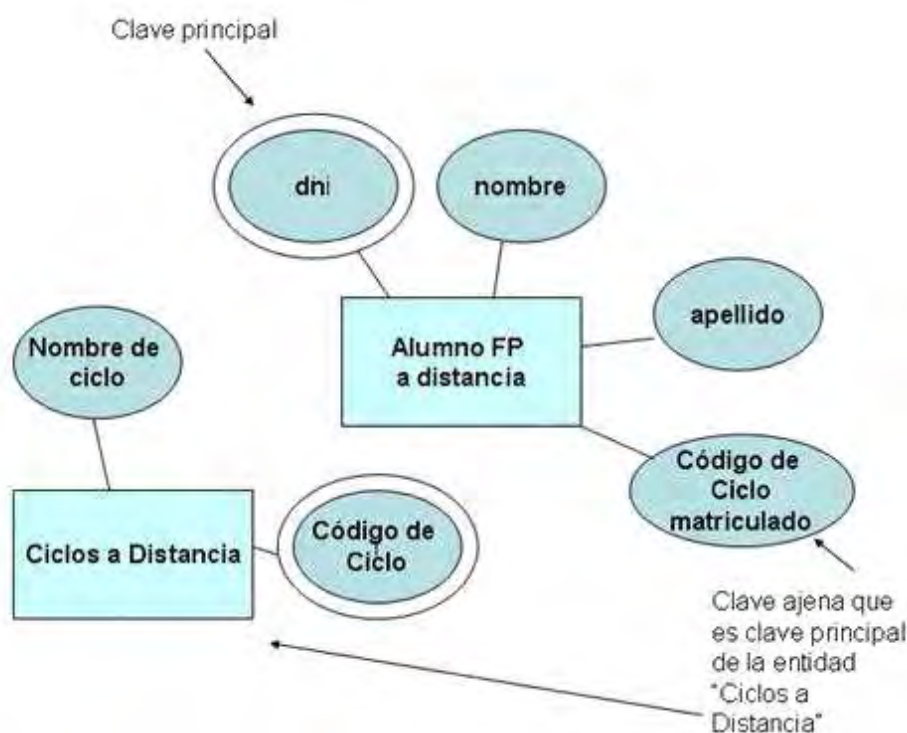
Un atributo puede ser un texto, un color, un dibujo, un sentimiento, etc., según la información que sea necesaria sobre la entidad que queremos guardar. Cuando los atributos toman un valor, señalan una instancia particular de la entidad.

Los atributos de una entidad se representan mediante elipses o círculos etiquetados, que se conectan por una línea recta a la entidad que califica, cada uno de los cuales tiene que tener un nombre único y que haga referencia a su contenido. Los nombres de los atributos deben ir en minúsculas.

Para distinguir sin ambigüedad una ocurrencia de entidad o relación de otra es necesario que nunca dos ocurrencias tengan los mismos dominios para todos sus atributos, es necesario que tengan una clave o identificador.

Se denomina **Superclave** a cada conjunto de atributos de una entidad que permiten distinguirla del resto de entidades del mismo tipo. Como por ejemplo el DNI de una persona es único para cada entidad persona. Las Claves Candidatas son aquellas superclaves que además contienen un conjunto de atributos mínimo. De todas las Claves Candidatas, el diseñador ha de elegir una que denominaremos **Clave Principal o Primaria** siendo todas las demás claves candidatas consideradas como claves alternativas. Por último, definimos como **Clave Ajena** a aquel conjunto de atributos de una entidad que son clave primaria en otra. Las claves principales se representan en el diagrama E/R con un círculo doble o subrayando el atributo.

En el caso anterior de la entidad **EMPLEADO**, pueden ser clave primaria tanto **dni**, como **Código_empleado**, depende del criterio del diseñador de la base de datos la que se elija. Pero una vez que el diseñador elige uno, sólo ése es clave primaria.



7. Relación

Relación

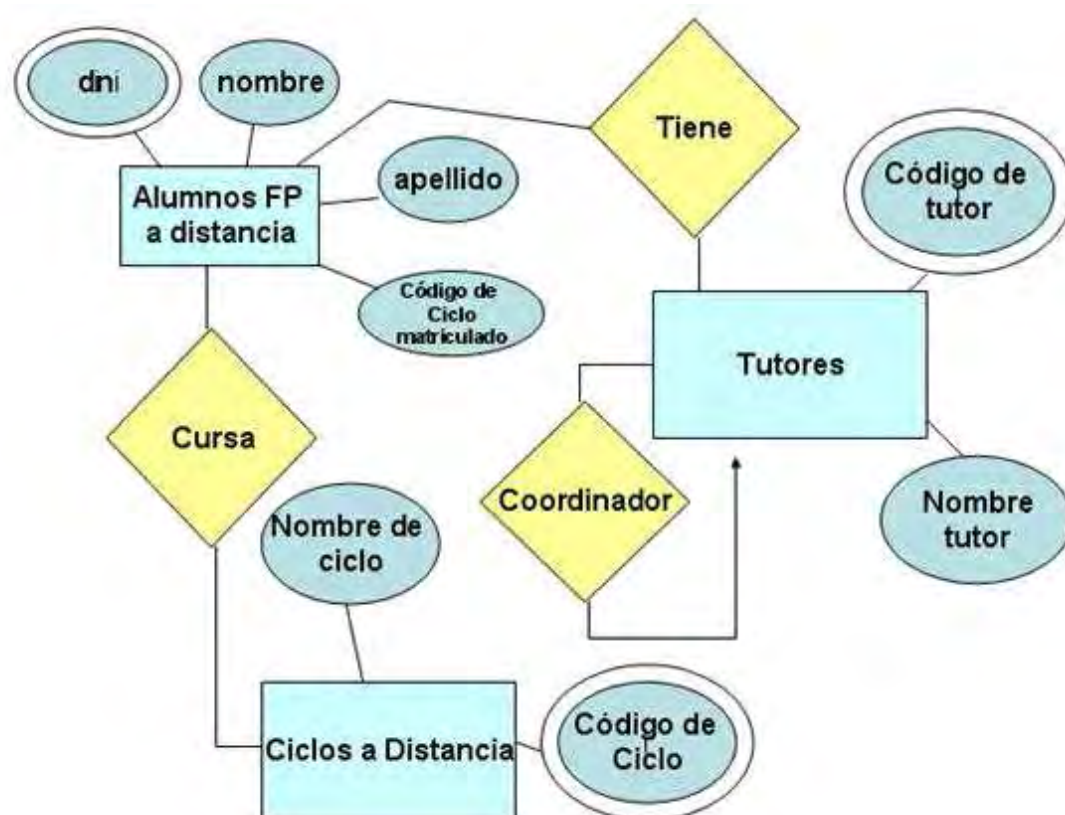
¿Te imaginas qué se puede representar mediante el nombre "RELACION"?

Efectivamente, los objetos de cualquier sistema de información se asocian unos con otros, siendo también de interés modelar estas conexiones. Para ello se utilizarán las relaciones, al igual que hemos hecho en el caso de las entidades.

Se entiende por relación aquella asociación o correspondencia existente entre entidades. Se representa mediante un rombo.

En la definición de una relación existen los siguientes elementos:

- **Nombre:** Como todo objeto del modelo E/R, cada relación tiene un nombre que la distingue claramente del resto y mediante el cual ha de ser referenciada.
- **Grado:** Es el número de entidades que participan en una relación. Por ejemplo, una relación puede asociar dos entidades distintas (grado 2 o relación binaria), como en el caso de la relación "**CURSA**" entre "**Alumnos FP Distancia**" y "**Ciclos a Distancia**", o bien puede asociar una sola entidad consigo misma (grado 1, denominándose en este caso relación reflexiva o unaria). Por ejemplo, en el caso de "**COORDINADOR**" en la entidad "**Tutores**", asociamos un tutor con otros, reflejando la posibilidad de que uno de ellos sea el coordinador de los demás. Puede existir también una relación que asocie más de dos entidades (grado n).



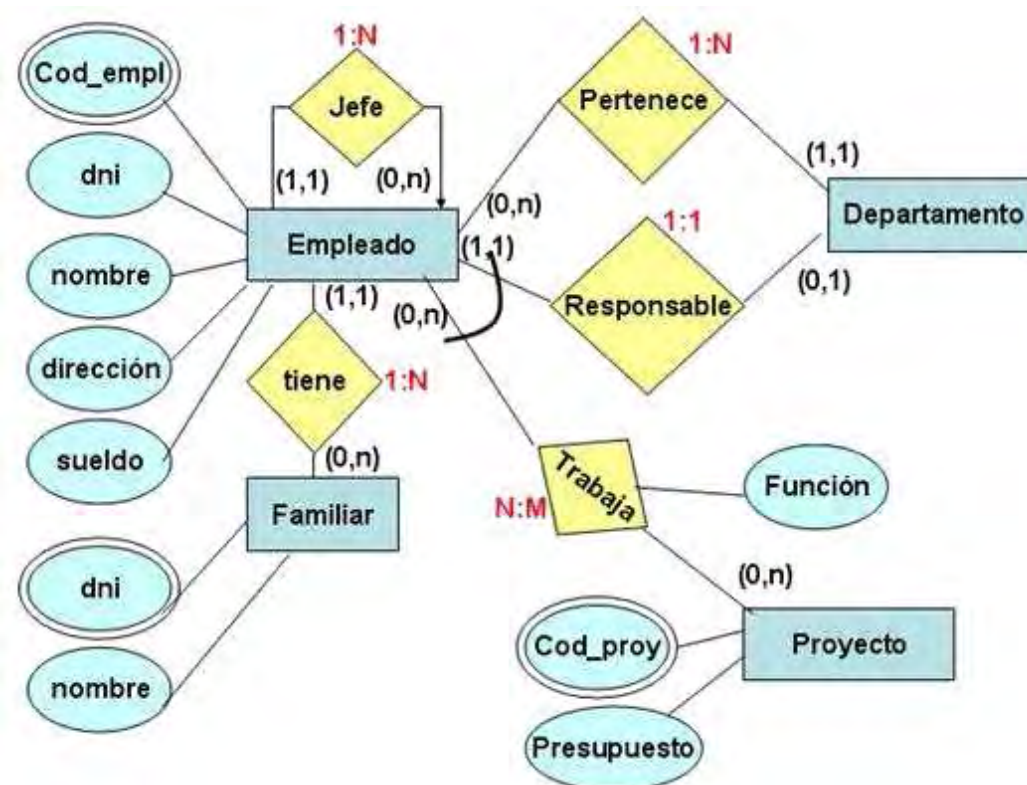
- **Tipo (de correspondencia) de la relación:** Es el número máximo de ocurrencias de cada entidad que pueden intervenir en una ocurrencia de relación que se está tratando. En la presentación gráfica aparece como una etiqueta con 1:1, 1:N, N:1 o N:M, según corresponda:
 - 1:1 A cada ocurrencia de una entidad le corresponde sólo una ocurrencia de la otra entidad.
 - 1:N A cada ocurrencia de una entidad le corresponden N ocurrencias de la otra entidad y a cada ocurrencia

de la segunda sólo le corresponde una de la primera.

■ N:M A cada ocurrencia de la primera le corresponde varias de la segunda y viceversa.

- **Cardinalidades de las entidades:** se definen como el número máximo y mínimo de ocurrencias de una entidad que pueden estar relacionadas con una ocurrencia de otra u otras entidades que participan en la relación. Su representación gráfica es una etiqueta del tipo (0,1), (1,1), (0,N) o (1,N), según corresponda. Así, por ejemplo, si a un departamento pertenecen de 0 a n empleados, en la entidad **EMPLEADO** aparecerán las cardinalidades (0,n); en el otro sentido, podemos suponer que un empleado tiene que pertenecer obligatoriamente a un departamento y a lo sumo a uno, por lo que las cardinalidades de **DEPARTAMENTO** serán (1,1).

Como en el caso de las entidades, las relaciones se clasifican también en **fuertes** o **débiles**, según estén asociando dos entidades fuertes o una entidad débil con otra fuerte respectivamente.



Un **atributo descriptivo** es aquel atributo que pertenece a una relación, no a una entidad. Por ejemplo, el atributo Función de la relación Trabaja.

Para saber más

Para tener más información acerca de la cardinalidad de las relaciones, pulsa en el siguiente enlace:

[Cardinalidades de las relaciones](#) [Versión en caché]

8. Tipos de relación

Tipos de relación

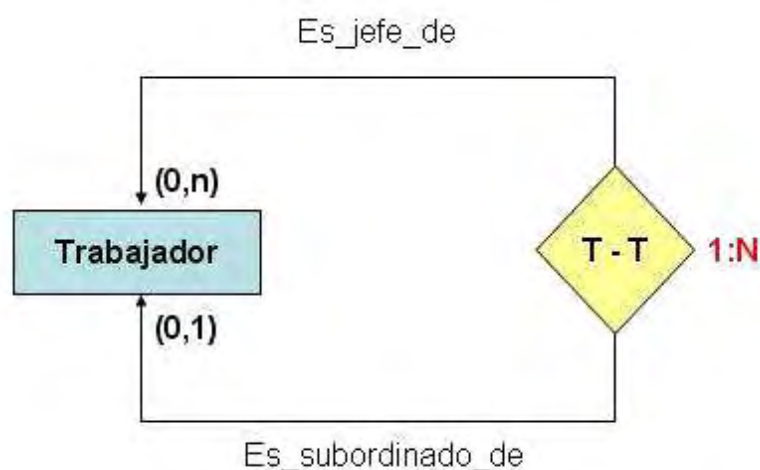
Hemos visto qué son las relaciones. ¿Pero cuántos **tipos** de relación existen?

El modelo E/R permite la representación de cualquier tipo de relación existente entre los objetos del mundo real que forman

parte del dominio del problema en estudio. Las últimas actualizaciones del modelo E/R, que han dado lugar a lo que se denomina **Modelo Entidad-Relación Extendido** (EE/R), permiten la representación de cualquier tipo de relaciones existentes entre clases de objetos que considera los principios de la abstracción.

■ Relaciones reflexivas.

Las relaciones reflexivas son [relaciones unarias](#) y, por tanto, consideran que en el tipo de relación se ve involucrado un único tipo de entidad. Por **ejemplo**, consideraremos el tipo de relación que describe la relación existente entre el tipo de entidad **TRABAJADOR** con ella misma, y que representa que un trabajador **es jefe** de 0 o varios trabajadores, mientras que un trabajador sólo es dirigido por 0 (nadie, por ser el jefe de todos) o 1 trabajador.

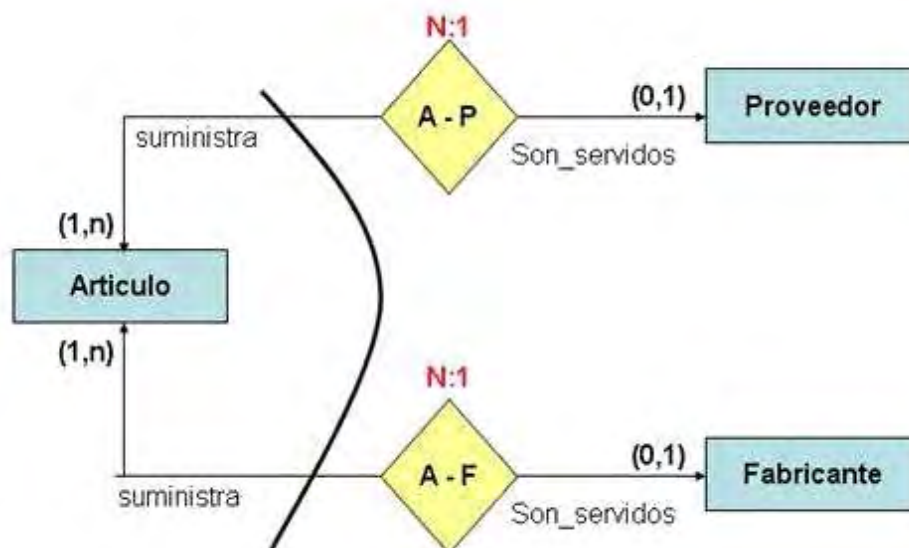


Se trata de un tipo de relación reflexiva en la que interviene un único tipo de entidad que desempeña dos papeles diferentes en el mismo tipo de relación.

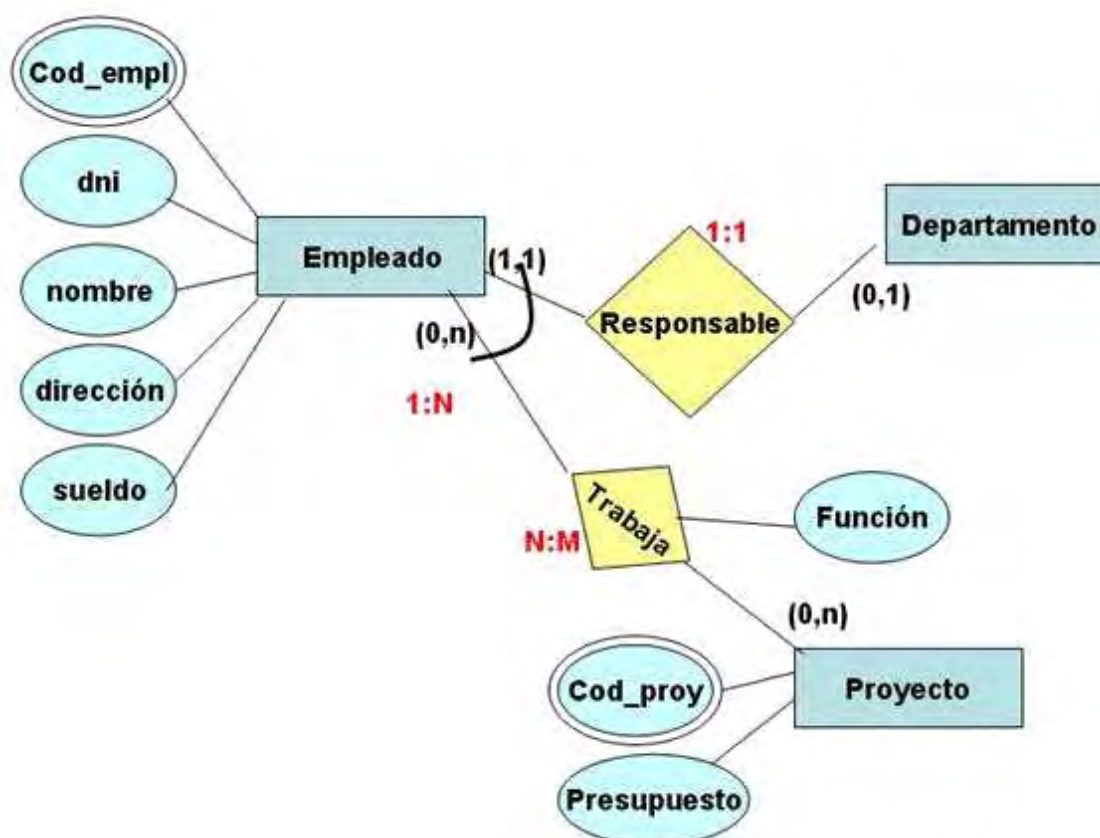
■ Relaciones exclusivas.

En un problema del mundo real un tipo de entidad puede mantener relaciones con un conjunto de otros tipos de entidad, pero no siempre estas relaciones son independientes. Consideremos el **ejemplo** en el que se presentan tres tipos de entidad: **ARTICULO**, **PROVEEDOR** y **FABRICANTE**, y el problema en el que los artículos son suministrados por los proveedores o por los fabricantes, pero un artículo nunca puede ser suministrado por un proveedor que no fabrica el artículo, de forma que si el fabricante puede suministrarlo, en ningún momento será solicitado ese artículo a ningún proveedor.

Para indicar la **exclusividad** entre dos tipos de relación que mantiene un mismo tipo de entidad se procede a representar un segmento que corta a los dos arcos que representan la relación del tipo de entidad con los tipos de relación exclusivas.



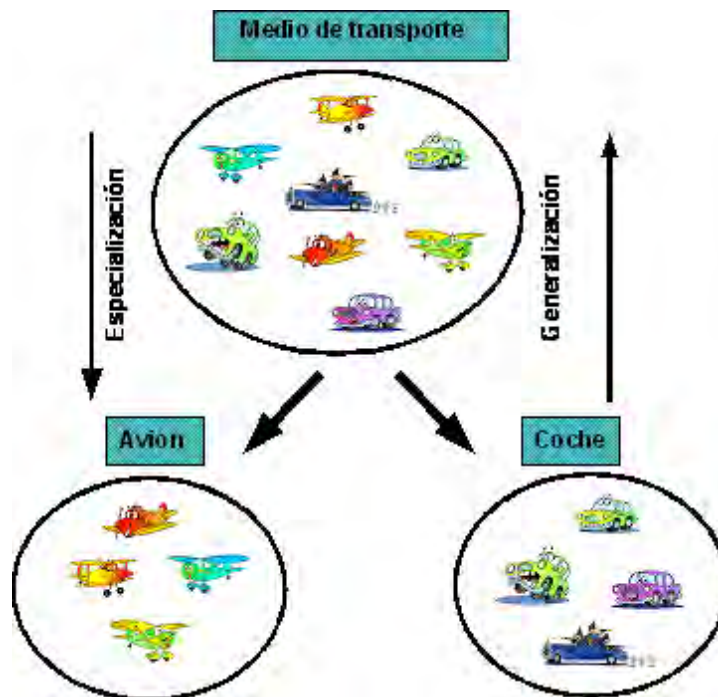
Otro ejemplo de relaciones exclusivas lo vemos en el ejemplo del apartado anterior, donde se supone que si un empleado es responsable de un departamento no puede trabajar en proyectos.



9. Generalización y especialización

Generalización y especialización

¿Qué te sugiere la palabra "**generalización**" si estamos hablando de entidades y de la representación de una parte de la realidad que nos rodea?

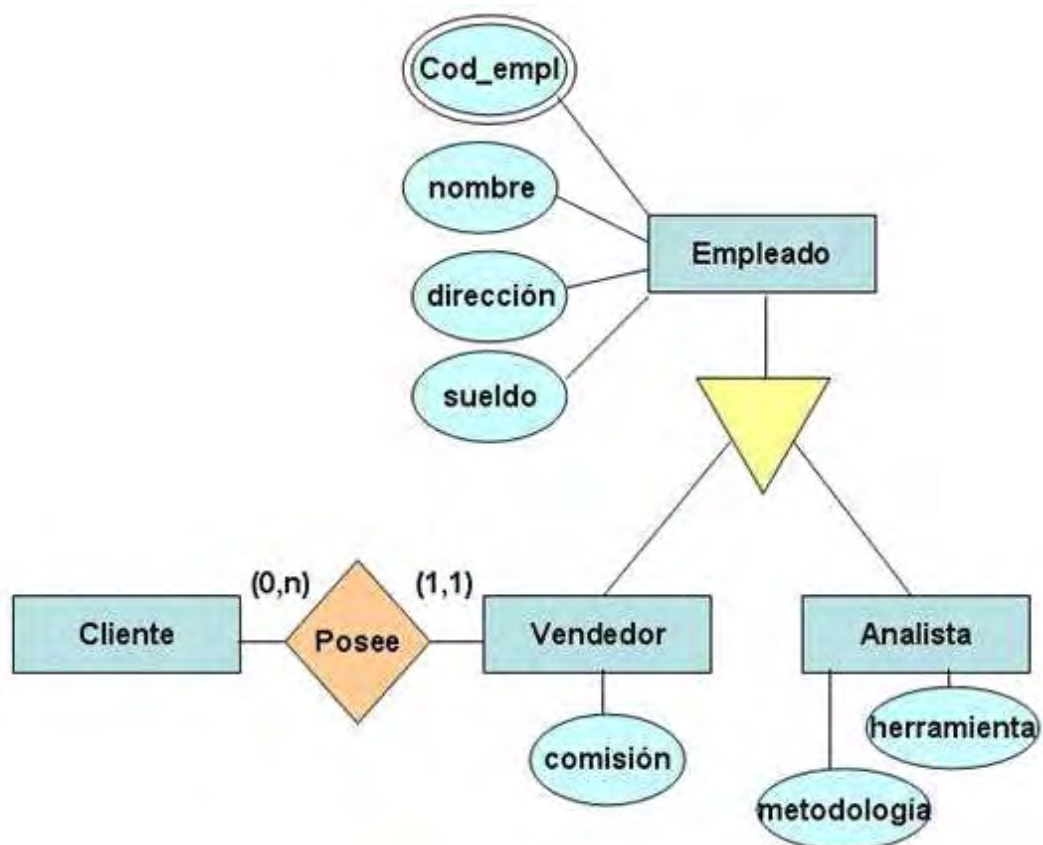


Efectivamente, lo que se te está ocurriendo, la **generalización** no es más que la reunión en un "supertipo" de entidad de una serie de "subtipos" de entidades, que tienen ciertos aspectos en común, pero que también se diferencian en algunos otros y que, para que la representación de la realidad sea lo más fiel posible, es importante que se reflejen en el modelo E/R dichas diferencias. Para ello se desarrolló el **modelo Entidad-Relación Extendido**, modelo ERE.

Resulta fácil de ver que el concepto de **especialización** es algo parecido al de generalización, pero considerado justo desde el punto de vista contrario.

La **descomposición de entidades** en varios subtipos es una necesidad muy habitual en la modelización de bases de datos. En efecto, en el mundo real se pueden identificar varias "jerarquías" de entidades. La relación que se establece entre el "supertipo" de entidad y sus "subtipos" corresponde a la noción de "**ESUN**" (conocida por sus siglas inglesas "ISA") o, más exactamente, "ESUNTIPODE".

Para su representación se utiliza un **triángulo invertido**, con la base paralela al rectángulo que representa el supertipo y conectado a los subtipos. En esta relación toda ocurrencia de un subtipo es una ocurrencia del supertipo, aunque no sucede lo contrario, con lo que las cardinalidades serán siempre (1, 1) en el supertipo y (0,1) o (1,1) en los subtipos.



Para saber más

Si quieres profundizar en los detalles del modelo ERE, pulsa en los siguientes enlaces:

[Modelo Entidad Relación Extendido \(I\)](#) [\[Versión en caché\]](#)

[Modelo Entidad Relación Extendido \(II\)](#) [\[Versión en caché\]](#)

10. Construcción de un esquema E/R

Construcción de un esquema E/R

¿Qué tenemos que hacer para crear un diagrama E/R? ¿Qué hay que saber y tener en cuenta?



Por lo pronto, conocer los **símbolos** que puedes usar en el diagrama y saber lo que significan, que es a lo que hemos dedicado gran parte de los apartados anteriores. Pero hay algo más...

Se parte del **análisis de la realidad empresarial**, analizando los listados, pantallas, normativas, etc., y realizando un conjunto de entrevistas a varios niveles de la empresa.

Posteriormente se elabora un **esquema** percibido, expresado en lenguaje natural que nos facilita la obtención del esquema conceptual, con lo que se hace preciso:

- "interpretar" las frases del lenguaje natural en el que está descrito el esquema percibido, y
- convertirlas en elementos del modelo E/R (entidades, atributos y relaciones).

Si bien no existen reglas que nos digan qué elemento va a ser una entidad o cuál otro una relación, sí se pueden enunciar unos **principios generales** que, junto al buen criterio del diseñador y a su experiencia, puedan ayudar a elaborar un esquema conceptual:

- Un **sustantivo** (nombre común) que actúa como sujeto o complemento directo en una frase es, en general, una entidad, aunque podría ser un atributo. Por ejemplo, en la frase "Los empleados trabajan en proyectos", existen dos posibles entidades: **EMPLEADO** y **PROYECTO**.
- Los **nombres propios** nos suelen indicar ocurrencias de una entidad, por ejemplo, "Narciso", "Alberto" o "Alfonso" indican una ocurrencia de **EMPLEADO**.
- Un **verbo** es una relación. En la frase anterior "trabajar" indica una relación entre las dos entidades: **EMPLEADO** y **PROYECTO**.
- Una **preposición** o frase preposicional entre dos nombres suele ser una relación, o también puede establecer la asociación entre una entidad y sus atributos. Por ejemplo, al decir "el departamento del empleado", estamos indicando la relación entre las entidades **DEPARTAMENTO** y **EMPLEADO**, mientras que si decimos "la dirección del empleado", estamos asociando el atributo **DIRECCIÓN** a la entidad **EMPLEADO**.

Por tanto, basándonos en conceptos lingüísticos podemos llegar a perfilar un primer esquema conceptual.

Existen **cuatro estrategias** a la hora de construir un esquema E/R

- **Estrategia descendente.** Es análoga a la descomposición en niveles de un DFD, partiendo de una única entidad que describe el universo del discurso (por ejemplo, EMPRESA) que se va descomponiendo sucesivamente con mayor nivel de detalle.
- **Estrategia ascendente.** Se parte del nivel más bajo, es decir, los atributos, que se van agrupando en entidades.

Posteriormente se crean relaciones entre las entidades y las jerarquías de generalización hasta obtener el esquema completo.

- **Estrategia de "mancha de aceite".** Es el método más usual. Se empieza creando un esquema E/R en una parte del papel, completándose, a medida que se examina, el esquema percibido en lenguaje natural y el resto de entidades y relaciones hasta "ocupar" todo el papel, de forma análoga a la manera en la que se extiende el aceite en un mantel.
- **Estrategia mixta.** Consiste en aplicar un enfoque que engloba características de varias estrategias anteriores. Se hace una división de requisitos atendiendo a una estrategia descendente, para luego aplicar una estrategia ascendente en cada una de las divisiones o particiones que se han hecho.

ESTRATEGIA	DESCRIPCIÓN	VENTAJAS	DESVENTAJAS
DESCENDENTE	Se analizan los conceptos de forma progresiva	No aparecen sorpresas de última hora	Requiere un alto nivel de abstracción
ASCENDENTE	Se construyen los conceptos a partir de otros elementales	Facilita las decisiones de diseño	Necesidad de reestructuración conforme se progresa
INSIDE-OUT	"Mancha de aceite"	Facilita describir nuevos conceptos relacionados	Sólo se tiene una visión global al final del proceso
MIXTO	Partición de requisitos descendente, dentro de cada partición se usa estrategia ascendente.	"Divide y vencerás"	Requiere decisiones importantes acerca del esqueleto inicial

Para saber más

Si quieres conocer algo más acerca del Modelo Entidad-Relación, pulsa en los siguientes enlaces:

[Modelado de datos](#) [\[Versión en caché\]](#)

11. Ejemplos resueltos de E/R: Ej 1 - Hospital

Ejemplos resueltos de E/R: Ej 1 - Hospital

Hasta ahora hemos visto toda la teoría que nos explica cómo hacer un diagrama E/R, pero ¿sabemos hacer realmente un diagrama?

La mejor manera de aprender, como ya hemos dicho, es haciéndolos... Pues ¡¡¡manos a la obra y vamos con el primer ejemplo!!!

Enunciado del primer ejemplo:

Con el objeto de crear un **software** para la administración de un hospital, se pretende diseñar una base de datos. Tras un estudio de la información existente en dicho centro, se obtuvieron los siguientes datos:

1. En el hospital se almacena información relativa a los **enfermos**: código de enfermo, nombre, dirección, DNI, fecha de nacimiento, teléfono de contacto, edad.
2. También se almacena la información relativa a los **trabajadores** del hospital (médicos y enfermeras), de los que se desea conocer: Nombre, dirección, DNI, sueldo, teléfono de contacto. En el caso de los médicos, además se desea conocer la especialidad.
3. Los enfermos están alojados en **habitaciones**, siendo una habitación asignada al enfermo en la fecha que se

hospitaliza y deja de estar asignada en la fecha de salida.

4. En una misma habitación, puede haber más de un enfermo.
5. Para cada habitación se almacena el número de camas y se anotan observaciones.
6. Las habitaciones pertenecen a las distintas plantas del hospital.
7. Para cada planta se almacena el número de habitaciones y la especialidad de la planta.
8. Los enfermos son curados por médicos según la especialidad. Dichos médicos pueden curar a varios enfermos.
9. Existe un conjunto de **enfermeras**. Cada enfermera, está asignada a una única planta.
10. Para cada enfermo existe un conjunto de **diagnósticos** emitidos por los médicos que los curan. Para cada diagnóstico del enfermo, se guarda la fecha del diagnóstico, y el informe, junto con el código correspondiente. Cada diagnóstico es emitido por un único médico.

Se pide realizar el análisis y obtener el diagrama E/R (habrá que indicar, según notación, las entidades, los atributos, las relaciones, y las llaves primarias de las entidades).

12. Identificando Entidades y Relaciones

Identificando Entidades y Relaciones.

Vamos a ir paso a paso siguiendo las instrucciones que hemos dado en los apartados anteriores de la unidad.

Empezamos **identificando las entidades**, para ello leemos el texto e intentamos identificar los sustantivos que puedan representar algo importante en nuestro sistema de información. En este caso un hospital.

Después de una primera lectura nos encontramos con **ENFERMO**, **TRABAJADOR** (**MEDICO**, **ENFERMERA**), **HABITACION**, **PLANTA** y **DIAGNOSTICO** como posibles entidades.

¿Podríamos considerarlas todas como entidades fuertes, o hay alguna que en principio pueda ser una entidad débil?

Si leemos con detenimiento, la entidad **DIAGNOSTICO** podría ser una entidad débil que depende de **ENFERMO**, y en efecto así es, puesto que si desaparece una instancia de la entidad **ENFERMO**, no tiene sentido la existencia de ningún **DIAGNOSTICO** para ese paciente... luego **DIAGNOSTICO** depende débilmente de **ENFERMO**.

Continuamos nuestro estudio **identificando las relaciones**, para ello volvemos a leer el texto.

¿De qué manera se relacionan las entidades que hemos descubierto en el paso anterior?

Claramente tenemos que **ENFERMO** se relaciona con **HABITACION** mediante '**esta_alojado**', que **HABITACION** se relaciona con **PLANTA** mediante '**ubicada**', **PLANTA** con **ENFERMERA** mediante '**asignada**', y por último **MEDICO** con **DIAGNOSTICO** mediante '**emite**'.

13. Identificando cardinalidades de Entidades y Relaciones

Identificando cardinalidades de Entidades y Relaciones.

Ahora que ya sabemos cuáles son las entidades y las relaciones implicadas, **¿cómo calculamos la cardinalidad de las entidades y por consiguiente de las relaciones?**

Muy sencillo, haciéndonos la siguiente pregunta para cada entidad: **¿con cuántas instancias de la entidad que nos interesa, participa en la relación en estudio?** La respuesta para cada caso es la siguiente:

- Para la entidad **ENFERMO**, ¿cuántos enfermos pueden alojarse en una habitación? Es evidente que puede que no haya ningún enfermo alojado o bien que haya el máximo número de enfermos que permita la habitación, por lo que la cardinalidad de **ENFERMO** será **(0,n)**.
- Para la entidad **HABITACIÓN** teniendo en cuenta la relación 'alojado', ¿en cuántas habitaciones puede alojarse un enfermo? Es evidente que sólo en una, por lo que su cardinalidad será **(1,1)**.
- Para la entidad **HABITACIÓN** teniendo en cuenta la relación 'esta', ¿en cuántas plantas puede estar una habitación? Es evidente que una habitación sólo puede estar en una planta, por lo que su cardinalidad será en este caso **(1,1)**.
- Para la entidad **PLANTA** teniendo en cuenta la relación 'esta', ¿cuántas habitaciones puede tener una planta? La respuesta es inmediata, como mínimo una planta debe tener una habitación y como máximo **n** habitaciones. La cardinalidad en este caso será **(1,n)**.
- Para la entidad **PLANTA** teniendo en cuenta la relación '**asignada**', ¿a cuántas plantas puede estar asignada una enfermera? Por el enunciado que tenemos, a una única planta, por lo que su cardinalidad será **(1,1)**.
- Para la entidad **ENFERMERA** teniendo en cuenta la relación '**asignada**', ¿cuántas enfermeras pueden estar asignadas a una planta? Es evidente, que al menos debe haber una enfermera y puede haber hasta **m**. Su cardinalidad en este caso será **(1,m)**.
- Para la entidad **MEDICO**, ¿cuántos médicos pueden emitir un diagnóstico? Gracias al enunciado sabemos que un diagnóstico puede ser emitido por un único médico, por lo que su cardinalidad será **(1,1)**.
- Y por último, para la entidad **DIAGNOSTICO**, ¿cuántos diagnósticos puede emitir un médico? Está claro que un médico puede no emitir ningún diagnóstico y puede emitir hasta **m** diagnósticos. Por este motivo, su cardinalidad será **(0,m)**.

Una vez que tenemos calculadas estas cardinalidades, **¿cómo calculo las cardinalidades de las relaciones?**

La respuesta es muy sencilla, tomando los valores máximos de las cardinalidades de cada una de las entidades que une dicha relación. Por tanto, en nuestro caso tenemos:

- La cardinalidad de la relación '**alojado**' es **1:n**.
- La cardinalidad de la relación '**esta**' es **1:n**.
- La cardinalidad de la relación '**asignada**' es **1:m**.
- La cardinalidad de la relación '**emite**' es **1:m**.

14. Identificando los atributos

Identificando los atributos

Seguimos el estudio **identificando los atributos**. Al leer los requerimientos del sistema nos preguntamos: **¿Qué información necesitamos almacenar de las distintas entidades encontradas?**

- Para la entidad **ENFERMO** necesitamos conocer: '**codigo_enfermo**', '**nombre**', '**direccion**', '**DNI**', '**fecha_nacimiento**', '**telefono_contacto**' y '**edad**', por lo que todos ellos son los atributos de la entidad **ENFERMO**. Hay que hacer una especial mención al atributo '**edad**', que no es más que un atributo

derivado, ya que se puede obtener como resultado de una operación sobre el campo **'fecha_nacimiento'**. De la misma manera también tenemos que estudiar qué ocurre con el atributo **'direccion'**, ya que dependiendo de cómo lo consideremos puede tratarse de un atributo simple, o un atributo compuesto por la calle, el número, población y provincia. En este caso lo consideramos como un atributo simple.

- Para la **entidad DIAGNOSTICO** se va a almacenar la **'fecha'** y el **'informe'**, junto con el **'codigo_diagnóstico'** correspondiente.
- De la **entidad MEDICO** necesitamos almacenar la siguiente información: **'nombre'**, **'direccion'**, **'DNI'**, **'sueldo'**, **'telefono_contacto'** y **'especialidad'**.
- De la **entidad ENFERMERA** queremos almacenar: **'nombre'**, **'direccion'**, **'DNI'** y **'telefono_contacto'**.
- De la **entidad HABITACION** almacenaremos: **'numero_habitacion'**, **'numero_de_camás'** y **'observaciones'**.
- Para la **entidad PLANTA**, por último, almacenaremos: **'numero_de_planta'**, **'numero_de_habitaciones'** y **'especialidad'** de la planta.

Ya hemos identificado todos los atributos de las entidades pero, **¿las relaciones no pueden tener también atributos?**

Leyendo los requerimientos de nuestro problema podemos observar que uno de los puntos que nos dan es el siguiente: "Los enfermos están alojados en habitaciones, siendo una habitación la asignada al enfermo en la fecha que se hospitaliza y deja de estar asignada en la fecha de salida". De esta forma nos indican que tenemos que guardar la información de cuándo se asigna la habitación a un enfermo y cuándo deja de estar asignada a ese enfermo, y la única manera que tenemos para hacerlo es añadiendo dos atributos a la relación **'esta_alojado'**. Estos atributos no son más que **'fecha_ingreso'** y **'fecha_alta'**.

No se aprecian más atributos de otras relaciones.

15. Identificando claves candidatas y elección de clave primaria

Identificando claves candidatas y elección de clave primaria

El siguiente paso a dar es **identificar las claves candidatas y elegir la clave principal**.

¿Sabemos cómo identificar las claves en una entidad?

Estudiamos las entidades una a una, y recordamos que una clave es un atributo o conjunto de atributos que identifica de manera única cada instancia de una entidad:

- **ENFERMO**, como claves candidatas tenemos **DNI** y **'codigo_enfermo'**, podemos seleccionar cualquiera de ellas como clave principal. Elegimos **DNI**
- **DIAGNOSTICO**, tiene como única clave candidata **'codigo_diagnostico'**, por lo tanto es además su clave principal.
- **MEDICO**, como clave candidata sólo tenemos **DNI**, por lo que también es nuestra clave principal.
- **ENFERMERA**, como clave candidata sólo tenemos **DNI**, por lo que también es nuestra clave principal.
- **HABITACION**, como única clave candidata tenemos **'numero_habitacion'**, y por lo tanto es la clave principal.
- **PLANTA**, como clave candidata tenemos únicamente **'numero_planta'**, por lo que también es la clave principal.

16. Comprobando generalizaciones/especializaciones y dibujando la solución

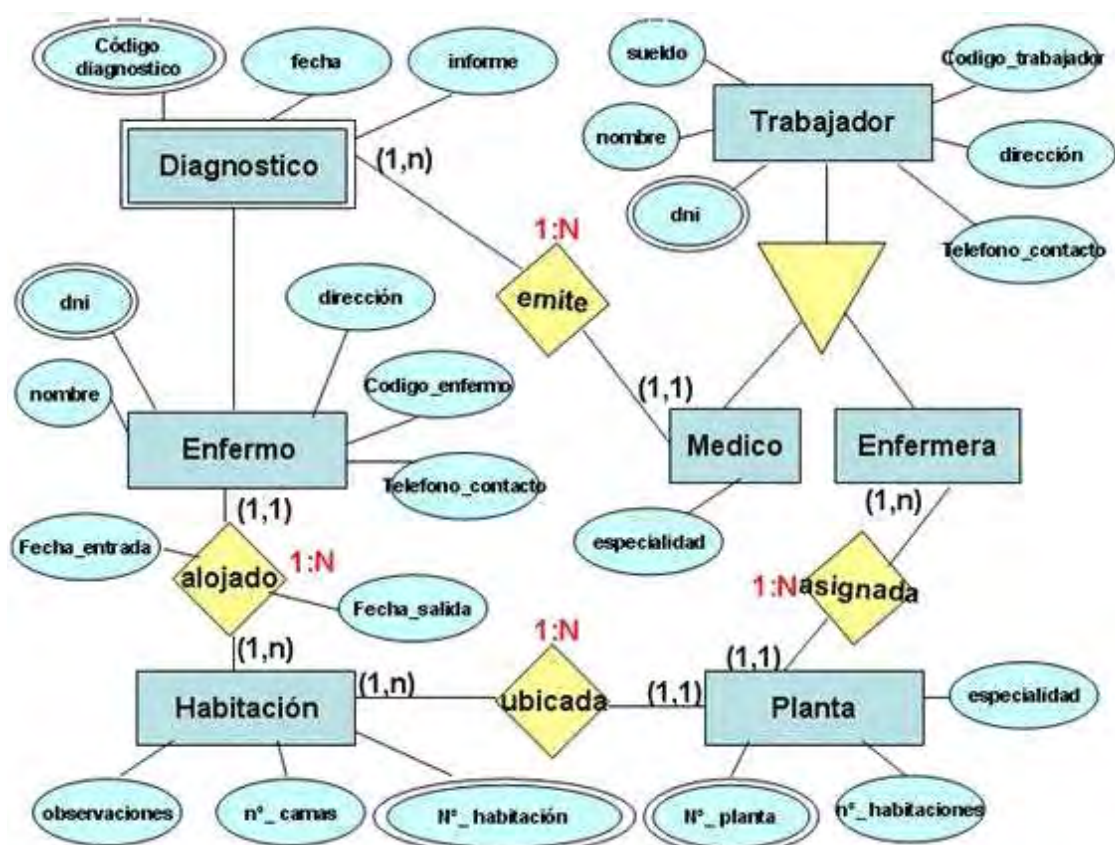
Comprobando generalizaciones/especializaciones y dibujando la solución.

¿Crees que ya hemos terminado con el estudio de nuestro diagrama?

Me temo que no, aún nos queda estudiar si **existe alguna generalización/especialización** en nuestro problema. Al leer con detenimiento el enunciado del problema nos percatamos de que existe una gran similitud entre las entidades **ENFERMERA** y **MEDICO**, tanto en atributos, como en el papel que juegan dentro del sistema de información. De ambas entidades se requiere almacenar la misma información, que sólo difiere en el caso de la entidad **MEDICO**, de la que deseamos guardar también el atributo '**especialidad**', por lo que podemos considerar una generalización de ambas entidades en una denominada **TRABAJADOR**.

Con todas estas consideraciones que hemos hecho, el diagrama ER que obtenemos es el que mostramos a continuación.

Solución:



17. Ejemplos resueltos de E/R: Ej 2 - Seguros.

Ejemplos resueltos de E/R: Ej 2 - Seguros.

Como es importante practicar, te vamos a proporcionar otro ejemplo resuelto más para que te pueda servir de guía en el proceso cuando tengas que enfrentarte a hacer tus propios diagramas E/R.

Enunciado:

Una compañía de seguros de automóviles quiere crear una **base de datos** que recoja el funcionamiento de su negocio y que se refleja en los siguientes supuestos semánticos:

- El elemento fundamental de información es la **póliza**, que se identifica mediante un número único, tiene un tipo de

seguro (a todo riesgo, a terceros, etc.), un importe de cobertura máxima y un estado (alta, baja, suspensión, etc.). La póliza pertenece a un único cliente (un cliente puede tener más de una póliza diferente) y referencia a un único vehículo y cada vehículo sólo puede tener una póliza.

- Los **clientes** se identifican mediante su NIF, además se quiere guardar su nombre y apellidos, su teléfono, fecha de nacimiento, fecha de obtención del permiso de conducir y su dirección completa (calle, número, ciudad, código postal y provincia).
- De los **vehículos** es importante conocer su número de chasis, su matrícula, la marca, el modelo, la potencia, el año de fabricación y el color. Además un vehículo puede tener una serie de extras (alarma, radio, etc.) de los que guardaremos un código identificador y el nombre del extra.
- Una póliza puede tener una serie de **personas autorizadas**, de las cuales se quiere tener su NIF, nombre y apellidos, fecha de nacimiento y relación con el cliente. Un autorizado sólo tendrá relación con un único cliente.
- Cuando se produce un **sinistro**, se crea un parte de accidente (identificado por un número de siniestro) donde se recoge la información del siniestro: datos de la póliza del cliente, datos del conductor (sólo puede ser el cliente, o alguien autorizado), fecha del siniestro, datos del taller donde se va a reparar el vehículo y fecha e importe de la reparación. Si el accidente es contra otro vehículo no se guardan ninguna información del vehículo contrario, si es de la misma compañía el cliente ya dará su propio parte de accidente.
- Los datos del **taller** que se almacenan serán, el nombre, la dirección y el teléfono además de un código identificador.

Se pide desarrollar el modelo E/R

18. Identificando Entidades y Relaciones

Identificando Entidades y Relaciones

Volvemos a repetir los pasos que hemos seguido en el ejemplo anterior, pero en esta ocasión te vamos a dar la oportunidad de que los razonamientos los hagas tú, nosotros te guiamos.

Empezamos **identificando las entidades**, para ello leemos el texto e intentamos identificar los sustantivos o sintagmas nominales que puedan representar algo importante en nuestro sistema de información. En este caso una compañía de seguros de automóviles.

¿Has tenido algún problema para identificar las entidades?

Podemos considerar las siguientes entidades: **POLIZA**, **CLIENTE**, **VEHICULO**, **EXTRA**, **AUTORIZADO**, **SINIESTRO** y **TALLER**.

¿Podríamos considerarlas todas como entidades fuertes, o hay alguna que en principio pueda ser una entidad débil?

Si analizas con detenimiento los requisitos, las entidades **AUTORIZADO** y **SINIESTRO** son entidades débiles de **CLIENTE** y **POLIZA** respectivamente.

Continuamos nuestro estudio **identificando las relaciones**, para ello volvemos a leer el texto.

¿De qué manera se relacionan las entidades que hemos descubierto en el paso anterior?

Claramente, **POLIZA** se relaciona con **CLIENTE** mediante una relación que podemos llamar '**tiene**', **POLIZA** se relaciona mediante '**referencia**' con **VEHICULO**, que a su vez se relaciona con **EXTRA** mediante '**esta equipado**'. Por otro lado, **CLIENTE** se relaciona mediante '**sufre cli**' con **SINIESTRO**, al igual que **AUTORIZADO** que se relaciona mediante '**sufre aut**' con **SINIESTRO**. Por último, **SINIESTRO** se relaciona mediante '**se repara**' con **TALLER**.

19. Identificando cardinalidades de Entidades y Relaciones

Identificando cardinalidades de Entidades y Relaciones

Ahora que ya sabemos cuáles son las entidades y las relaciones implicadas, **¿cómo calculamos la cardinalidad de las entidades y por consiguiente de las relaciones?**

Muy sencillo, haciéndonos la siguiente pregunta para cada entidad:

¿Con cuántas instancias participa la entidad que nos interesa en la relación en estudio?

De esta forma calculamos la cardinalidad de las entidades, y la cardinalidad de las relaciones la obtenemos de tomar el máximo de cada una de las cardinalidades de las entidades de dicha relación. La respuesta para cada caso es la siguiente:

- Para la entidad **POLIZA** tenemos una cardinalidad **(1,n)** en la relación '**tiene**' con la entidad **CLIENTE**. La entidad **CLIENTE** tiene una cardinalidad **(1,1)** en la relación '**tiene**' con la entidad **POLIZA**. Por lo tanto, la relación '**tiene**' tiene una cardinalidad **1:n**
- La entidad **POLIZA** tiene una cardinalidad **(1,1)** en la relación '**referencia**' con la entidad **VEHICULO**. Para la entidad **VEHICULO** tenemos una cardinalidad **(1,1)** en la relación '**referencia**' con la entidad **POLIZA**. Por lo tanto, la relación '**referencia**' tiene una cardinalidad **1:1**.
- La entidad **VEHICULO** tiene una cardinalidad **(0,n)** en la relación '**esta_equipado**' con la entidad **EXTRA**. Para la entidad **EXTRA** tenemos una cardinalidad **(0,m)** en la relación '**esta_equipado**' con la entidad **VEHICULO**. Por lo tanto, la relación '**esta_equipado**' tiene una cardinalidad **n:m**.
- La entidad **CLIENTE** tiene una cardinalidad **(0,1)** en la relación '**sufre_cli**' con la entidad **SINIESTRO**. Para la entidad **SINIESTRO** tenemos una cardinalidad **(0,n)** en la relación '**sufre_cli**' con la entidad **CLIENTE**. Por lo tanto, la relación '**sufre_cli**' tiene una cardinalidad **1:n>**.
- La entidad **AUTORIZADO** tiene una cardinalidad **(0,1)** en la relación '**sufre_aut**' con la entidad **SINIESTRO**. Para la entidad **SINIESTRO** tenemos una cardinalidad **(0,n)** en la relación '**sufre_aut**' con la entidad **AUTORIZADO**. Por lo tanto, la relación '**sufre_aut**' tiene una cardinalidad **1:n**.
- La entidad **SINIESTRO** tiene una cardinalidad **(1,n)** en la relación '**se_repara**' con la entidad **TALLER**. Para la entidad **TALLER** tenemos una cardinalidad **(0,1)** en la relación '**se_repara**' con la entidad **SINIESTRO**. Por lo tanto, la relación '**se_repara**' tiene una cardinalidad **1:n**.

20. Identificando atributos

Identificando atributos

Seguimos el estudio **identificando los atributos**. Al leer los requerimientos del sistema nos preguntamos: **¿Qué información necesitamos almacenar de las distintas entidades encontradas?**

- Para la entidad **POLIZA**, tenemos los atributos '**cod_poliza**', '**tipo_seguro**', '**importe_cobertura**' y '**estado**'.
- Para la entidad **CLIENTE**, tenemos los atributos '**NIF**', '**nombre**', '**apellidos**', '**telefono**', '**fecha_nacimiento**', '**fecha_carnet**' y '**direccion**'.
- Para la entidad **VEHICULO**, tenemos los atributos '**matricula**', '**marca**', '**modelo**', '**potencia**', '**chasis**', '**anno_fabricacion**' y '**color**'.
- Para la entidad **EXTRA**, tenemos los atributos '**cod_extra**' y '**nombre**'.
- Para la entidad **AUTORIZADO**, tenemos los atributos '**NIF**', '**nombre**', '**apellidos**', '**fecha_nacimiento**' y '**relacion_cliente**'.
- Para la entidad **SINIESTRO**, tenemos los atributos '**nº_siniestro**' y '**fecha**'.

- Para la entidad **TALLER**, tenemos los atributos **'cod_taller'**, **'nombre'**, **'direccion'** y **'telefono'**.

Ya hemos identificado todos los atributos de las entidades pero, ¿**las relaciones no pueden tener también atributos**?

Por supuesto que sí, y de hecho la relación **'se_repara'** tiene dos atributos, **'importe'** y **'fecha'** (de la reparación), debido a que al considerar estos dos atributos en la relación puede contemplarse el caso de que una reparación tenga lugar en distintas fechas y talleres. Si estos atributos estuvieran en la entidad **SINIESTRO** no cabría dicha posibilidad, y todas las reparaciones deberían efectuarse en el mismo momento y por un único importe.

21. Identificando claves candidatas y elección de clave primaria

Identificando claves candidatas y elección de clave primaria

El siguiente paso a dar es **identificar las claves candidatas** y **elegir la clave principal**, pero en esta ocasión vamos a elegir directamente las claves principales de nuestras entidades.

- **POLIZA**, clave principal: **'cod_poliza'**
- **CLIENTE** clave principal: **'NIF'**
- **VEHICULO** clave principal: **'matricula'**
- **EXTRA** clave principal: **'cod_extra'**
- **AUTORIZADO** clave principal: **'NIF'**
- **SINIESTRO** clave principal: **'n°_siniestro'**
- **TALLER** clave principal: **'cod_taller'**

22. Dibujando la solución

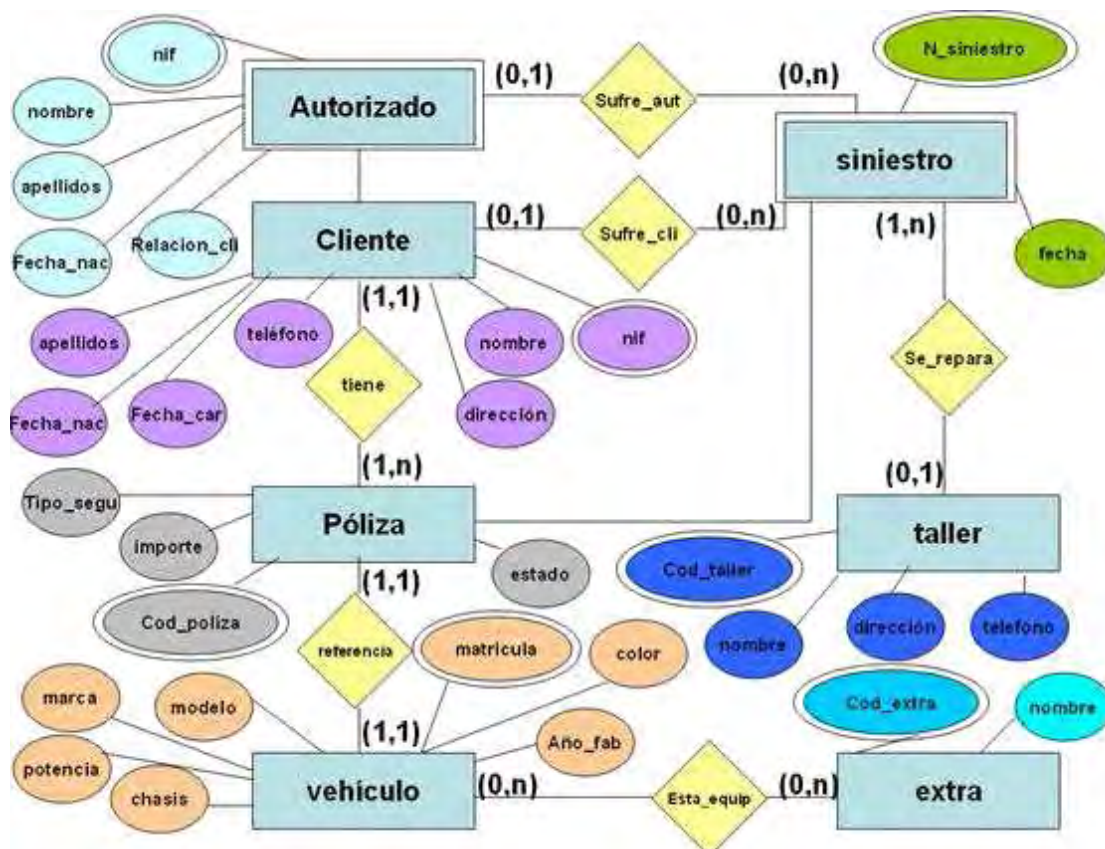
Dibujando la solución

¿Crees que ya hemos terminado con el estudio de nuestro diagrama?

Me temo que no, aún nos queda estudiar si **existe alguna generalización/ especialización** en nuestro problema. En esta ocasión no aparece ningún dato que nos haga pensar que tenemos alguna generalización/especialización en nuestro diagrama.

Después de todas estas comprobaciones podemos construir ya nuestro diagrama E/R.

Solución:



En los siguientes recursos encontrarás tanto enunciados de ejercicios, como sus soluciones. No todos los ejercicios vienen resueltos, pero justamente se trata de que tú intentes resolver algunos de ellos por ti mismo. En cualquier caso, también es interesante que intentes resolver algunos antes de mirar las soluciones.

[Ejercicio 1](#)

[Ejercicio 2](#)

[Ejercicio 3](#)

En el siguiente enlace encontrarás las soluciones a los enunciados del último fichero de los enlaces anteriores.

[Soluciones](#)

PARA SABER MÁS

En el siguiente enlace encontrarás otro ejercicio resuelto de un diagrama Entidad-Relación.

[Ejercicio Entidad-Relación resuelto](#) [Versión en caché]

23. Traducción del modelo E/R al modelo relacional

Traducción del modelo E/R al modelo relacional

En SI Andalucía continúan trabajando en la aplicación para la compañía de seguros. Ya elaboraron el diagrama E-R,

pero el proceso no termina ahí. No es más que una primera aproximación. **María** le cuenta a **Víctor** que a partir del diagrama E-R habrá que aplicar una serie de normas de transformación para obtener **el diseño del Modelo Relacional**. Ese diseño es un nuevo diagrama, al que llamamos **esquema relacional**, y que representa el modelo lógico de la base de datos. **Víctor** le pregunta qué es lo que muestra ese esquema, y **Carmen**, que también participa en la formación de **Víctor**, le cuenta que el modelo relacional nos indicará de forma clara y esquemática qué tablas tendrá finalmente la base de datos, qué campos o columnas tendrá cada tabla, cómo se relacionan unas tablas con otras, y cuales serán las claves primarias de cada tabla. De esta forma, el diseño de la base de datos estará completo, y pasar esas tablas a un SGBD concreto será una tarea rutinaria.

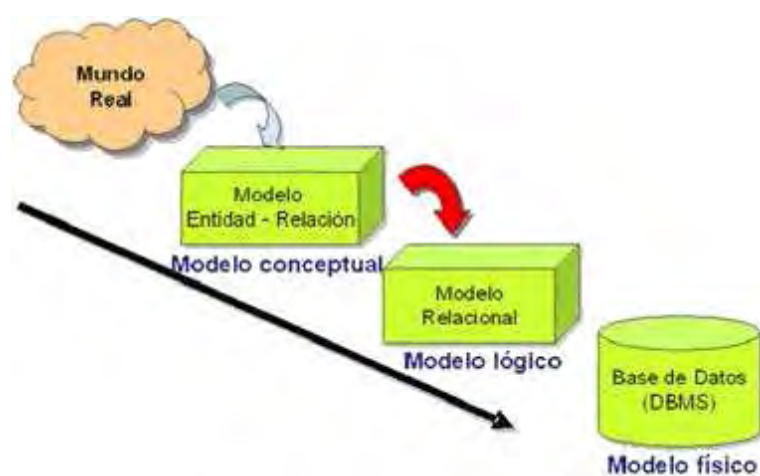
María le explica a **Víctor** que al igual que el modelo Entidad-Relación, el modelo Relacional también está basado en conceptos matemáticos. Concretamente se basa en el concepto matemático de relación, que **se representa gráficamente mediante una tabla**, lo que lo hace muy intuitivo. Es en definitiva un modelo matemático, pero afortunadamente **la mayoría de las operaciones de transformación que permiten pasar del modelo entidad-relación al modelo relacional son bastante "automáticas"**, y pueden aplicarse incluso sin dominar esos conceptos matemáticos, aunque conocer en qué se basan esas reglas ayuda a aprender a usarlas correctamente con más rapidez.

Carmen le propone a **Víctor** que la invite a cenar, y de esta forma podrá explicarle con más detalle **cómo se realiza ese proceso de transformación**, y en qué reglas se basa. Naturalmente **Víctor** acepta, porque realmente tiene ganas de aprender y de aplicar lo aprendido al diseño de la base de datos de la compañía de seguros. Y porque cenar en compañía de **Carmen** también le resulta por sí sólo suficientemente estimulante.

La verdad es que **Víctor** esperaba que las **reglas de transformación** fueran extremadamente complicadas, pero sin embargo se ha sorprendido de lo fácil y lógico que es aplicar esas reglas de transformación del diagrama E-R al modelo relacional. Como **Carmen** le dice, esa es una de sus grandes ventajas, que piensa siempre de forma lógica, y por eso entiende casi siempre todo a la primera.

Una vez vistas estas reglas, no queda más que aplicarlas, y **Víctor** se pone a elaborar su primer modelo relacional: el de la base de datos de la compañía de seguros.

María le llama la atención sobre algunos detalles que debe tener en cuenta, y al final tuvo que corregirle algún pequeño fallo, pero en general puede decirse que para ser el primer ejercicio de este tipo que hace, **Víctor** lo ha resuelto brillantemente, y en un tiempo bastante reducido. (Puedes ver el resultado obtenido para el ejemplo de los seguros en el apartado 10 y sus correspondientes subapartados)



La traducción del **modelo conceptual** al modelo **lógico** es la transformación de los esquemas E/R al modelo lógico que soporte los datos: jerárquico, Codasyl o relacional. Nos centraremos en el modelo relacional, por ser el más extendido en la actualidad.

24. 1 Reglas de transformación.

Reglas de transformación.

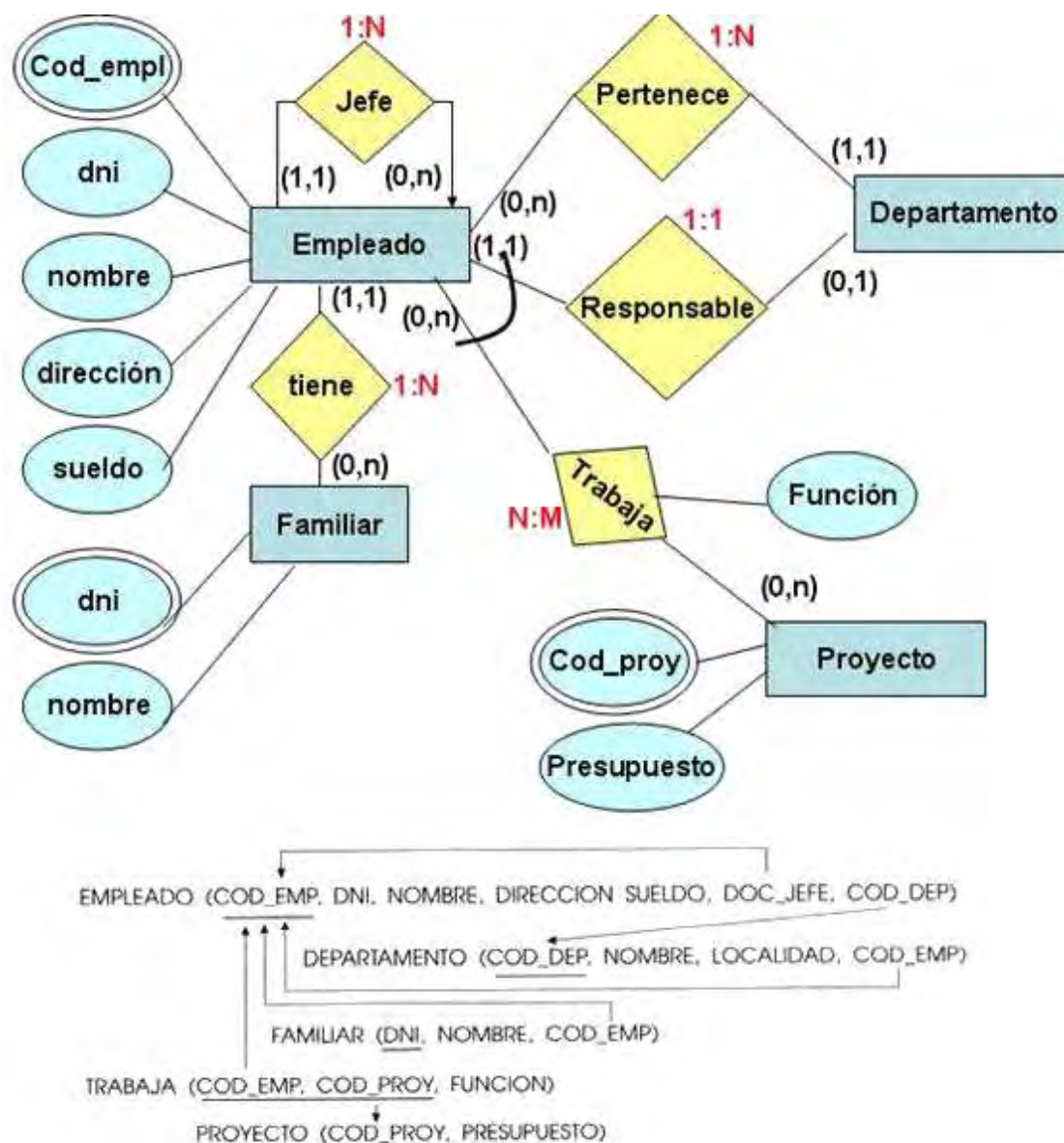
El paso de un esquema en el modelo E/R al relacional está basado en los **tres** principios siguientes:

1. Toda entidad se convierte en una relación (tabla).
2. Toda interrelación N:M se transforma en una relación (tabla).
3. Toda interrelación 1:N se traduce en el fenómeno de "[propagación de clave](#)".

A primera vista se observa que en el paso del Modelo E/R al relacional, **se pierde semántica**, puesto que tanto las entidades como las interrelaciones se transforman en relaciones, ya que en el modelo relacional sólo existe la relación para representar ambos tipos de objetos. También se constata pérdida de semántica en la propagación de clave, pero en este caso desaparece incluso el nombre de la interrelación 1:N. Es preciso llamar la atención sobre el hecho de que la pérdida de semántica no implica, sin embargo un peligro para la integridad de la base de datos, ya que se definen restricciones de integridad referencial que aseguren la conservación de la misma.

En el **ejemplo** puede observarse que:

- las tres entidades **EMPLEADO**, **PROYECTO** y **DEPARTAMENTO** se han transformado en otras tantas relaciones:
 - la interrelación N:M **TRABAJA** da lugar a una nueva relación cuya clave es la concatenación de las claves primarias de las entidades que participan en ella (**COD_EMP** y **COD_PROYECTO**), y éstas son, además, claves ajenas de **TRABAJA** que referencian a las tablas **EMPLEADO** y **PROYECTO** respectivamente;
 - la interrelación 1:N **PERTENECE** se ha transformado mediante el mecanismo de propagación de clave, por el que se ha incluido en la tabla **EMPLEADO** el atributo clave de la entidad **DEPARTAMENTO** (**COD_DEP**), que se constituye así en clave ajena de la relación **EMPLEADO** referenciando a la tabla **DEPARTAMENTO**;
 - la interrelación 1:N **"JEFE DE"** también se transforma de manera análoga, con la única diferencia de que al propagar la clave de **EMPLEADO** (**COD_EMP**) a la misma tabla, es necesario cambiar el nombre (**COD_JEFE**).
- Por lo que respecta a las interrelaciones 1:1, se pueden recoger en el esquema relacional de distintas maneras atendiendo a las cardinalidades de las entidades que participan en la misma:



Si las entidades que se asocian poseen **cardinalidades** (0,1), en este caso la interrelación 1:1 se transformará en una **relación**, además de las dos relaciones que representan cada una de las entidades. Por **ejemplo**, si tenemos la asociación **MATRIMONIO** entre las entidades **HOMBRE** y **MUJER**, que se transforma en una relación, evitando así los valores nulos que aparecerían en caso de propagar la clave de la entidad **MUJER** a la tabla **HOMBRE** o viceversa, ya que como reflejan las cardinalidades no todos los hombres ni todas las mujeres se encuentran casados.

Si una de las entidades que participa en la interrelación posee cardinalidades (0,1), mientras que en la otra son (1,1), conviene propagar la clave de la entidad con cardinalidades (1,1) a la tabla resultante de la entidad de cardinalidades (0,1). En el **ejemplo** tenemos una interrelación que recoge el empleado que es responsable de un departamento, suponiendo que un empleado puede ser responsable como máximo de un departamento y que cada departamento tiene que tener un responsable pero sólo uno. En este caso propagamos la clave de **EMPLEADO** a la tabla de **DEPARTAMENTO**, evitando así valores nulos, y captando más semántica recogemos la cardinalidad mínima 1, que en caso de realizar la propagación en sentido contrario no podríamos captar directamente.

En el caso de que ambas entidades presenten cardinalidades (1,1), se puede propagar la clave de cualquiera de ellas a la tabla resultante de la otra, teniendo en cuenta en este caso, los accesos más frecuentes y prioritarios a los datos de las tablas. Se puede plantear -también por motivos de eficiencia- la propagación de las dos claves. Sin embargo, esto introduce redundancias que deben ser controladas por medio de restricciones.

Las **dependencias en existencia y en identificación** no son recogidas directamente en el modelo relacional. En el **ejemplo**, vemos que la manera de transformar una dependencia en existencia es utilizar el mecanismo de propagación de clave,

creando una clave ajena con nulos no permitidos en la relación de la entidad dependiente, con la característica de obligar a una modificación y un borrado en cascada. Además, en el caso de dependencia en identificación, la clave primaria de la relación de la entidad débil debe estar formada por la concatenación de las claves de las dos entidades participantes en la interrelación.

En lo que respecta a los tipos y subtipos, no son objetos que se puedan representar explícitamente en el modelo relacional estándar. Ante una entidad y sus subtipos caben varias soluciones de transformación en el modelo relacional, con la subsiguiente pérdida de semántica dependiendo de la estrategia elegida. Destacamos tres principalmente:

- **Opción a: Englobar todos los atributos de la entidad y sus subtipos en una sola relación.** En general, adoptaremos esta solución cuando los subtipos se diferencien en muy pocos atributos y las interrelaciones que los asocian con el resto de las entidades del esquema sean las mismas para todos los subtipos. Por **ejemplo**, la diferencia que existe entre un analista y un vendedor podemos considerarla como mínima desde el punto de vista de que son empleados de la empresa. Por este motivo, la solución adecuada en este caso será la creación de una sola tabla que contenga todos los atributos del supertipo y los de los subtipos, añadiendo un atributo adicional que indique el tipo de empleado que realiza el atributo discriminante de la jerarquía. También se tendrán que especificar las restricciones semánticas correspondientes que obliguen a que los empleados que sean analistas no puedan tener comisión (que COMISIÓN sea nulo).
- **Opción b: Crear una relación para el supertipo y tantas relaciones como subtipos haya con sus atributos correspondientes.** Ésta es la solución adecuada cuando existen muchos atributos distintos entre los subtipos, y se quieren mantener de todas maneras los atributos comunes a todos ellos en una relación. Al igual que en el caso anterior habrá que crear las restricciones y/o aserciones oportunas.
- **Opción c: Considerar relaciones distintas para cada subtipo que contengan además los atributos comunes.** Se elegiría esta opción cuando se den las mismas condiciones que en el caso anterior (muchos atributos distintos) y los accesos realizados sobre los datos de los distintos subtipos afectan casi siempre a atributos comunes.

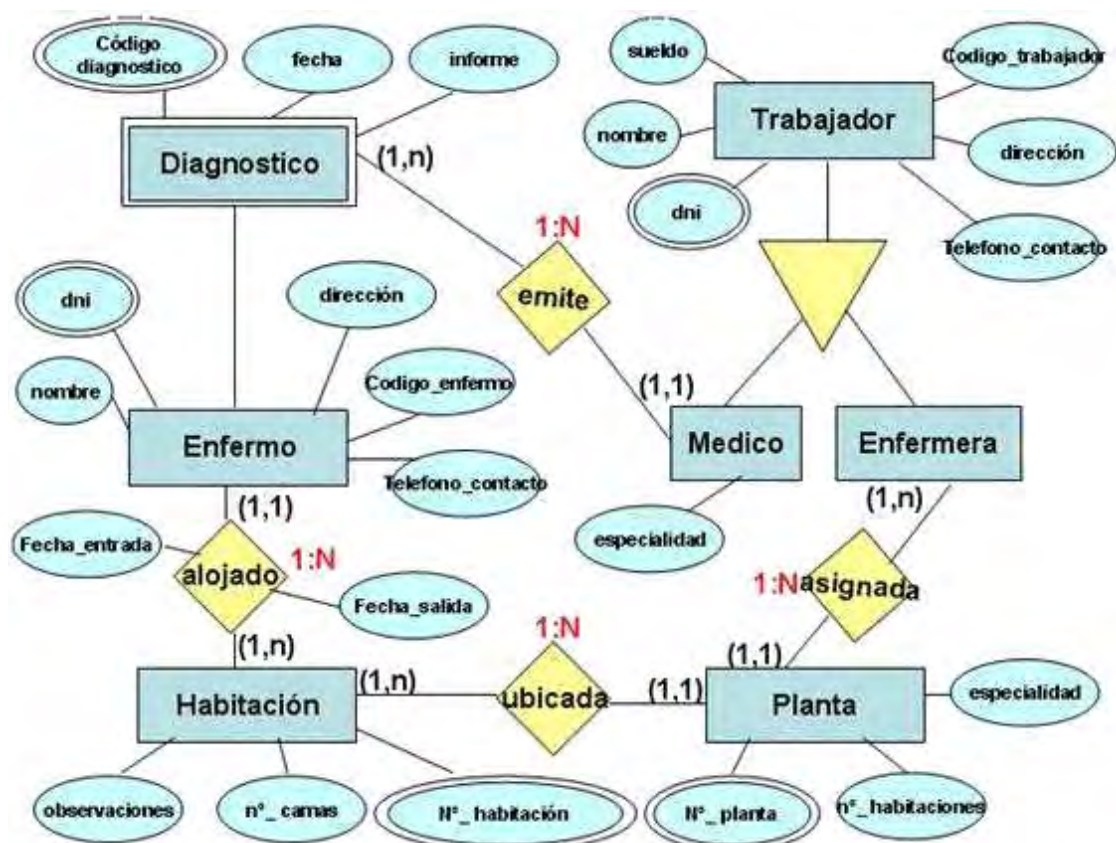
Elegiremos una estrategia u otra dependiendo de que sea la semántica o la eficiencia las que primen para el usuario en un momento determinado.

25. Ejemplo 1 de Transformación: Hospital

Ejemplo 1 de Transformación: Hospital.

Veamos todo lo anterior con un **ejemplo** concreto.

Para ello vamos a utilizar el ejemplo del hospital que hicimos en un apartado anterior. A continuación volvemos a reproducir el diagrama E/R que obtuvimos, y que será el punto de partida sobre el que comenzaremos la tarea de obtener el esquema relacional usando las reglas de transformación que hemos ido viendo en los apartados anteriores:



26. Transformando las entidades

Transformando las entidades.

Empezamos nuestra transformación al Modelo Relacional:

- **Entidades que nos encontramos:** **HABITACION**, **PLANTA** y **ENFERMO** como entidades fuertes, y **DIAGNOSTICO** como entidad débil.

Las tablas serían:

```
HABITACION ( )
PLANTA ( )
ENFERMO ( )
DIAGNOSTICO ( )
```

- **Consideramos los atributos para cada entidad:** Debemos recordar que las entidades débiles heredan los atributos de la clave primaria de la entidad fuerte de la que dependen, por lo que la entidad **DIAGNOSTICO** hereda el campo **DNI** de la entidad **ENFERMO** de la que depende.
- **Es habitual modificar el nombre del campo heredado si es necesario para evitar posibles ambigüedades o confusiones en la identificación de los atributos heredados.** Siguiendo esta recomendación notamos el **DNI** de **ENFERMO** como **DNI_enfermo** en la tabla **DIAGNOSTICO** para tener presente en todo momento de dónde procede dicho atributo.

```
HABITACION(n°_habitacion, n°_camas, observaciones)
PLANTA(n°_planta, n°_habitaciones, especialidad)
ENFERMO(DNI, codigo_enfermo, Nombre, direccion, telefono_contacto)
DIAGNOSTICO(codigo_diagnostico, DNI_enfermo, fecha, informe)
```

- **Seleccionamos las claves primarias:** Sabemos que coinciden con las claves primarias definidas en el diagrama E/R.

El único caso que merece especial atención es el de la entidad débil **DIAGNOSTICO**, que al ser débil en existencia y no en identificación, hereda la clave primaria de la entidad fuerte de la que depende, es decir, de **ENFERMO**, pero dicho atributo no forma parte de la clave primaria de **DIAGNOSTICO**.

HABITACION (nº_habitacion, nº_camapas, observaciones)
PLANTA (nº_planta, nº_habitaciones, especialidad)
ENFERMO (DNI, codigo_enfermo, Nombre, direccion, telefono_contacto)
DIAGNOSTICO (codigo_diagnostico, DNI_enfermo, fecha, informe)

27. Transformamos las relaciones

Transformamos las relaciones

- **Ahora pasamos las relaciones a tablas:** Sabemos que, en principio, salvo propagación de claves, cada relación pasa a ser una tabla en el Modelo Relacional, de esta manera tenemos las siguientes tablas:

alojado()
 emite()
 ubicada()
 asignada()

- **Añadimos atributos y seleccionamos las claves primarias:** Podemos observar que todas las relaciones son del tipo uno a muchos, con cardinalidad mínima uno de la entidad que participa a uno, por lo que todas las relaciones son candidatas a propagar la clave primaria de la relación que participa con cardinalidad 1 a la relación que participa con cardinalidad N, salvo que la relación tenga atributos propios, como es el caso de "alojado".

alojado (nº_habitacion, DNI, fecha_entrada, fecha_salida)
 emite (codigo_diagnostico, DNI_medico)
 ubicada (nº_habitacion, nº_planta)
 asignada (nº_planta, DNI_enfermera)

28. Transformamos los casos especiales y fusionamos

Transformamos los casos especiales y fusionamos

Por último tenemos que ocuparnos de la transformación de aquellas entidades que presenten generalización (o especialización, según se mire). En este caso, tanto **ENFERMERA** como **MEDICO** pueden considerarse claramente como casos de especialización sobre la entidad **TRABAJADOR**. (O bien **TRABAJADOR** como una generalización de **MEDICO** y **ENFERMERA**)

- **Pasamos la especialización a tablas:** De entre las distintas posibilidades que tenemos para pasar a tablas una especialización nos quedamos con la opción de convertir las entidades especializadas en tablas y no la generalización, es decir, pasamos a tablas las entidades **MEDICO** y **ENFERMERA** y no **TRABAJADOR** debido a que existen atributos específicos de una de las entidades y además cada una de ellas establece una relación concreta con otras entidades, lo que le da peso específico para existir de manera independiente.

Debemos recordar que las entidades que componen la especialización heredan los atributos de la entidad más general y se le añaden los atributos propios si los tuviera.

Con estos recordatorios podemos considerar:

MEDICO (DNI_medico, codigo_trabajador, Nombre, sueldo, direccion, telefono_contacto, especialidad)
 ENFERMERA (DNI_enfermera, codigo_trabajador, Nombre, sueldo, direccion, telefono_contacto)

- Fusionamos las tablas siguiendo el criterio de propagación de claves visto con anterioridad, es decir, las claves primarias de las entidades que participan con uno en la relación uno a muchos se propagan a la entidad que participa con muchos.

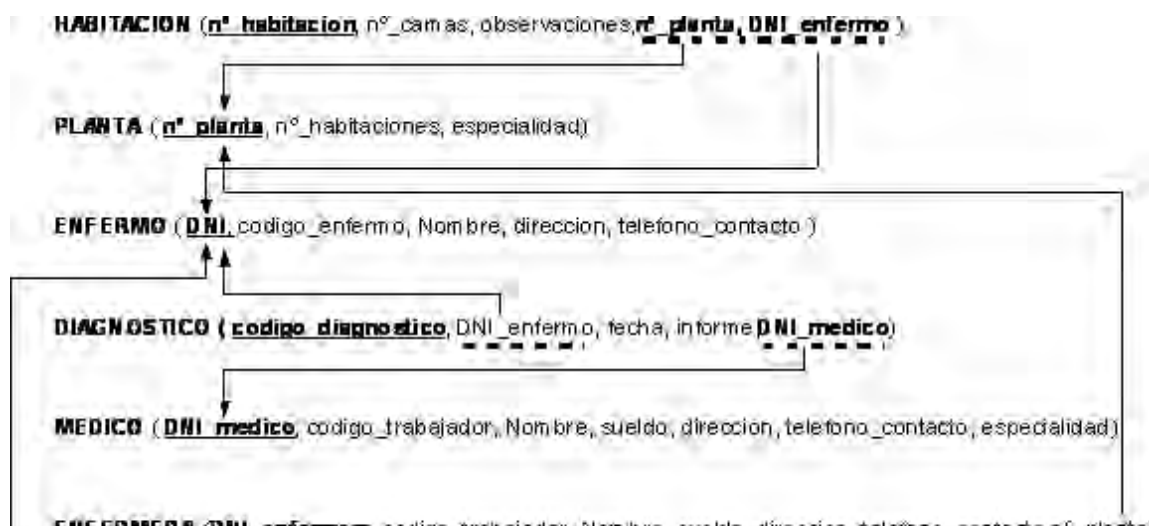
HABITACION (n° habitacion, n°_camas, observaciones, n°_planta, DNI_enfermo)
 PLANTA (n°_planta, n°_habitaciones, especialidad)
 ENFERMO (DNI, codigo_enfermo, Nombre, direccion, telefono_contacto)
 DIAGNOSTICO (codigo diagnostico, DNI_enfermo, fecha, informe, DNI_medico)
 MEDICO (DNI_medico, codigo_trabajador, Nombre, sueldo, direccion, telefono_contacto, especialidad)
 ENFERMERA (DNI_enfermera, codigo_trabajador, Nombre, sueldo, direccion, telefono_contacto)
 alojado (n° habitacion, DNI, fecha_entrada, fecha_salida)
 emite (codigo diagnostico, DNI_medico)
 ubicada (n° habitacion, n°_planta)
 asignada (n°_planta, DNI_enfermera)

Como puedes observar, las tres últimas tablas las hemos tachado. Eso se debe a que desaparecen como consecuencia de la propagación de las claves primarias.

29. Resultado final

Resultado final

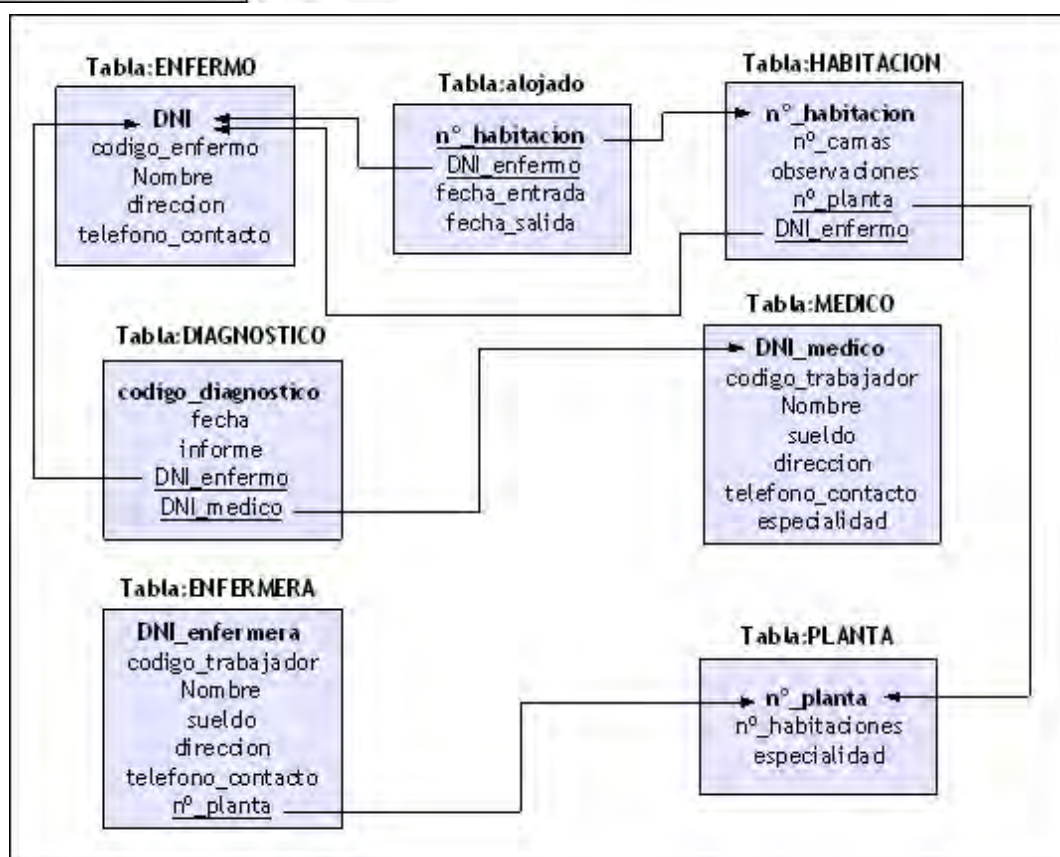
Hasta este momento no hemos tendido en cuenta las claves ajenas del Modelo Relacional que estamos construyendo. Ahora es el momento de especificar estas claves, y lo hacemos de dos maneras distintas, subrayando con línea discontinua el atributo, y vinculándolo con la clave primaria que referencia. De esta manera nos queda el siguiente esquema relacional:



Otra manera de expresar el esquema relacional es la siguiente. Como podrás

comprobar es más visual e intuitiva que la anterior:

`alojado (n°_habitacion, DNI_enfermo, fecha_entrada, fecha_salida)`

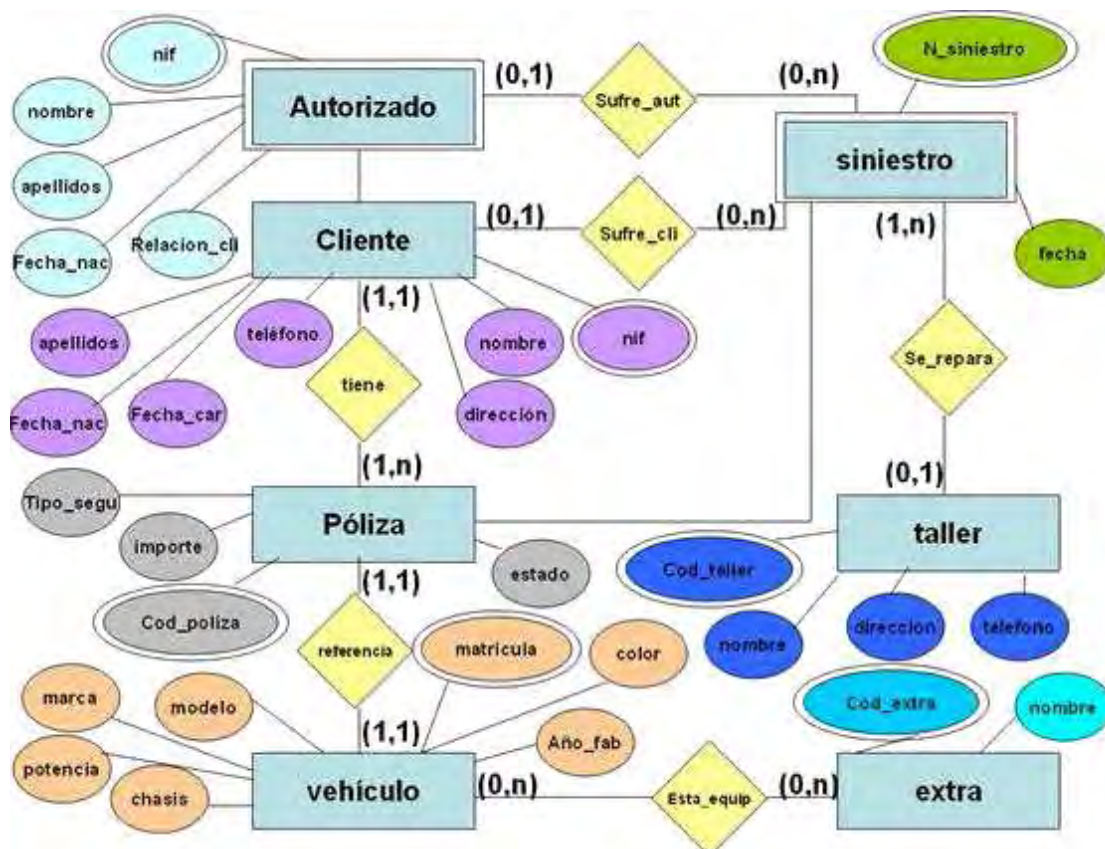


30. Ejemplo 2 de Transformación: Seguros

Ejemplo 2 de Transformación: Seguros

Igual que para aprender a hacer diagramas E/R la mejor manera era practicar y hacer varios de ellos, para aprender a transformar los diagramas E/R en esquemas relacionales, también hay que practicar y hacer varios ejemplos de transformaciones antes de saber hacerlas bien. Por eso es conveniente que te proporcionemos algún ejemplo más resuelto.

En esta ocasión también vamos a utilizar el otro ejemplo que vimos en el apartado anterior.



31. Transformamos las entidades

Transformamos las entidades.

Empezamos nuestra transformación al Modelo Relacional con este segundo ejemplo:

Entidades, atributos y claves primarias: Como entidades fuertes tenemos **CLIENTE**, **POLIZA**, **VEHICULO**, **EXTRA** y **TALLER**, mientras que como entidades débiles tenemos **AUTORIZADO** y **SINIESTRO**.

Volvemos a recordar que las entidades débiles heredan los atributos de la clave primaria de la entidad fuerte de la que dependen y que hay que tener en cuenta si la dependencia es en existencia o en identificación para decidir cual es la clave principal de la tabla que representa la entidad débil.

Te recordamos también que es habitual modificar el nombre del campo heredado si es necesario para evitar posibles ambigüedades o confusiones en la identificación de los atributos heredados.

Teniendo en cuenta todas estas consideraciones, las tablas de este ejemplo son las siguientes:

CLIENTE (NIF, nombre, apellidos, direccion, telefono, fecha_nacimiento, fecha_permiso)
POLIZA (num_poliza, tipo, cobertura, estatus, matricula)
VEHICULO (matricula, num_chasis, marca, modelo, potencia, anno_fabricacion, color)
EXTRA (cod_extra, descripcion)
TALLER (cod_taller, nombre)
AUTORIZADO (NIF, nombre, apellidos, telefono, fecha_nacimiento, relacion, NIF_cliente)
SINIESTRO (num_siniestro, num_poliza, fecha_siniestro)

Los únicos casos que merecen especial atención en cuanto a la elección de la clave primaria son los de las entidades débiles **AUTORIZADO** y **SINIESTRO**. La entidad débil **SINIESTRO** es débil en existencia, por lo que hereda la clave primaria de **POLIZA**, pero no para formar parte de su clave primaria, sin embargo, la entidad débil **AUTORIZADO** es débil en identificación, ya que un mismo autorizado, lo puede ser para varios clientes, con lo que además de heredar la clave primaria de **CLIENTE**, ésta forma parte de su propia clave primaria.

32. Transformamos las relaciones

Transformamos las relaciones.

Pasamos ahora las relaciones a tablas, con los atributos y las claves primarias: Sabemos que, en principio, salvo propagación de claves, cada relación pasa a ser una tabla en el Modelo Relacional, de esta manera tenemos las siguientes tablas:

```
sufre_aut (num_siniestro, NIF_cliente, NIF_autorizado)
sufre_cliente (num_siniestro, NIF_cliente)
tiene (NIF_cliente, num_poliza)
referencia (num_poliza, matricula)
esta_equip (cod_extra, matricula)
se_repara (num_siniestro, cod_taller, fecha, importe)
```

La relación **referencia**, al ser de cardinalidad uno a uno, puede tomar como clave primaria cualquiera de los dos atributos.

Para decidir cuales son las claves primarias de estas tablas hemos observado las cardinalidades de cada relación. Sabemos que las relaciones **sufre_aut**, **sufre_cliente** y **se_repara** son del tipo uno a muchos, con cardinalidad mínima cero de la entidad que participa a uno, por lo que todas las relaciones se tratan como si fueran del tipo muchos a muchos, y por consiguiente heredan las claves primarias de las entidades que relacionan y además dichos atributos constituyen la clave primaria de la relación.

Como podemos ver en el diagrama E/R, la relación **esta_equip** es de muchos a muchos, por lo que tiene como clave primaria la unión de las claves primarias de las entidades que relaciona.

33. Fusionamos las tablas

Fusionamos las tablas

Fusionamos las tablas siguiendo el criterio de propagación de claves visto con anterioridad, es decir:

- Las claves primarias de las entidades que participan con uno en la relación uno a muchos y tienen cardinalidad mínima uno, se propagan a la entidad que participa con muchos.
- Además también tenemos en cuenta que las relaciones uno a uno propagan también la clave primaria de una de las entidades que relaciona a la otra.

```
CLIENTE (NIF,nombre,apellidos,direccion,telefono, fecha_nacimiento,fecha_permiso)
POLIZA (num_poliza,tipo,cobertura,estatus,matricula, NIF_cliente)
VEHICULO
(matricula,num_chasis,marca,modelo,potencia,anno_fabricacion,color,num_poliza)
```

```

EXTRA (cod_extra,descripcion)
TALLER (cod_taller,nombre)
AUTORIZADO (NIF,nombre,apellidos,telefono,fecha_nacimiento,relacion, NIF_cliente)
SINIESTRO (num siniestro,num poliza,fecha siniestro)
sufre_aut (num siniestro,NIF_cliente,NIF autorizado)
sufre_cliente (num siniestro, NIF_cliente)
tiene (NIF_cliente,num poliza)
referencia (num poliza,matricula)
esta Equip (cod_extra,matricula)
se repara (num siniestro>,cod taller,fecha,importe)

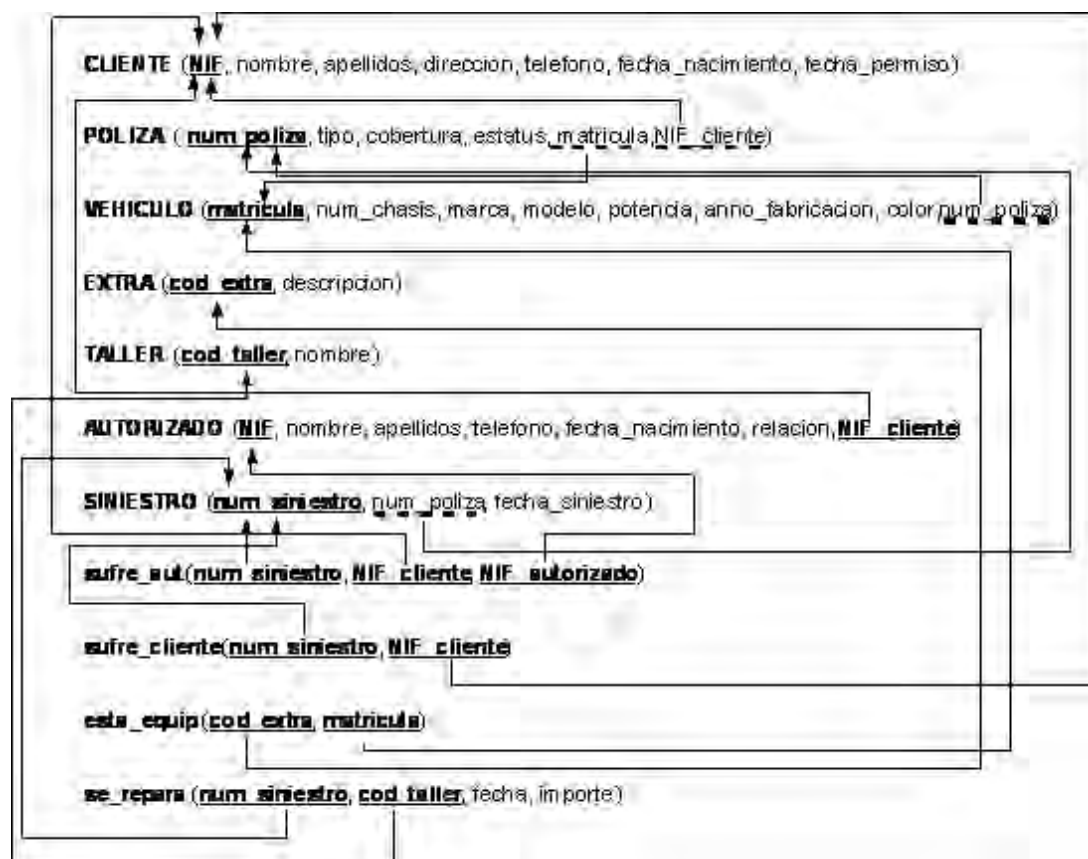
```

De nuevo, las dos tablas que aparecen tachadas porque desaparecen como consecuencia de la propagación de las claves primarias.

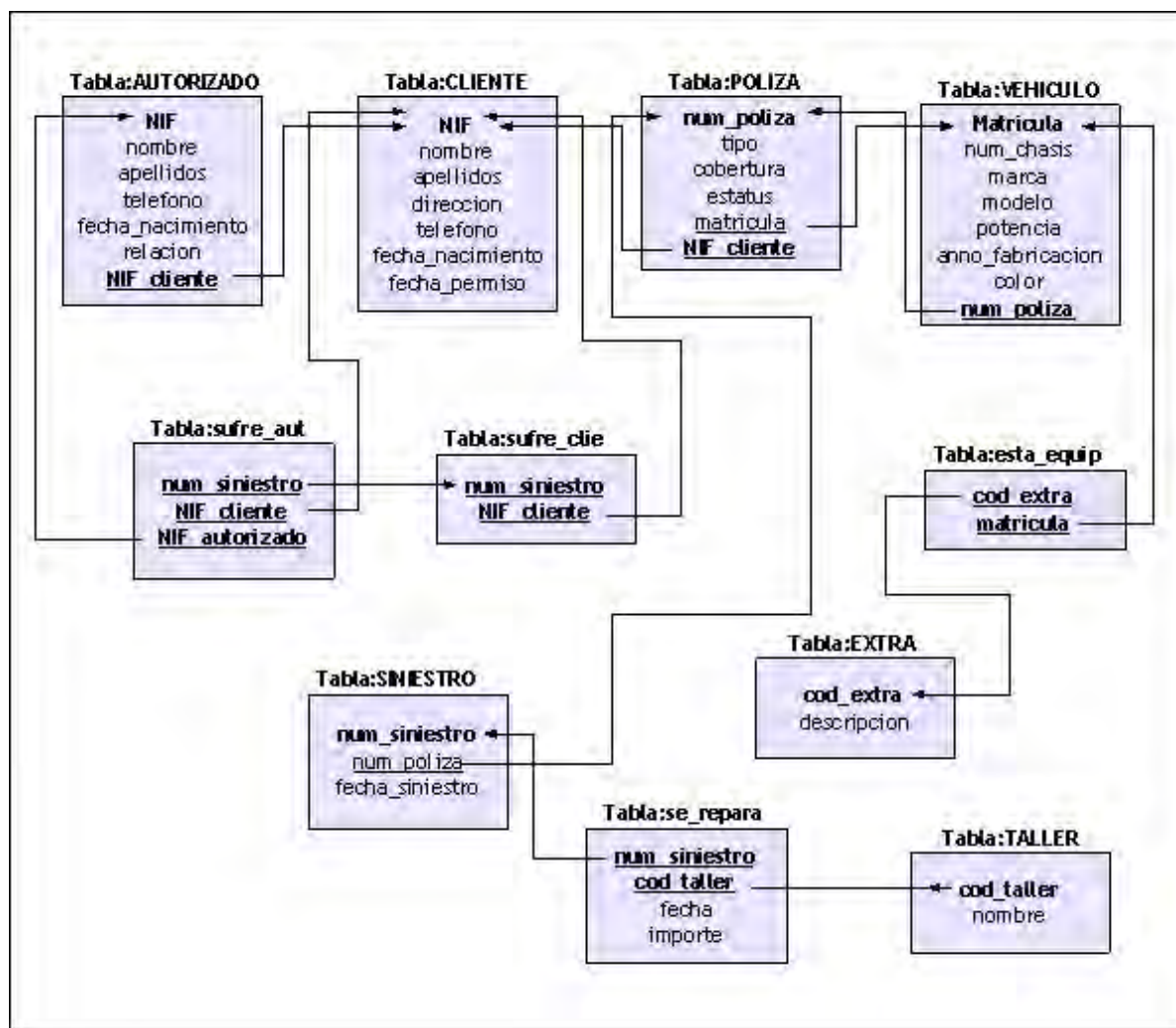
34. Resultado final

Resultado final

De nuevo no hemos tendido en cuenta las claves ajenas del Modelo Relacional que estamos construyendo. Ahora es el momento de especificar estas claves, y lo hacemos de dos maneras distintas, subrayando con línea discontinua el atributo, y vinculándolo con la clave primaria que referencia, como hemos hecho en el ejemplo anterior. De esta manera nos queda el siguiente esquema relacional:



Y ahora vemos la otra manera de representar el Modelo Relacional:



¿Te acuerdas de la herramienta CASE que vimos en la unidad anterior, llamada Studio Case? Es la que utilizamos para realizar DFD's, pues bien, acuérdate que había dos dimensiones:

- la funcional que usábamos con los DFD's y
- la de información, que será la utilizada para Diagramas Entidad/Relación.

Pues ahora vamos a ver mediante una pequeña animación cómo se utiliza para realizar los diagramas Entidad Relación, que además tendrás que usar para la tarea de esta unidad.

[Mira cómo se usa Studio Case para realizar diagramas E/R](#)