

**MODELADO CONCEPTUAL DE BASES DE DATOS**  
**BASES DE DATOS II**  
**DEPARTAMENTO DE INFORMÁTICA**  
**UNSL**

## 1. Introducción

El desarrollo de una base de datos requiere de un proceso complejo que se inicia cuando se toma la iniciativa de crear una base y finaliza cuando la misma está operativa. Por lo que dada una realidad que se desea representar computacionalmente, se requiere de una metodología que nos permita en una serie de pasos lograr el objetivo. El siguiente diagrama muestra los pasos elementales de este proceso:

*Mundo Real → Universo del Discurso → Modelado Conceptual → Modelado Lógico →  
Modelado Interno → Almacenamiento Físico*

El primer paso en la concepción de una base de datos es definir el *universo del discurso*, fijando para ello una serie de objetivos sobre el mundo real que se va a analizar. Una vez definido el universo del discurso acerca del cual deseamos recoger información en nuestra base de datos, debemos proceder a su estructuración, paso a paso, hasta llegar al producto deseado.

La siguiente etapa es el modelado conceptual, que consiste en la descripción del mundo real en términos formales, no ambiguos, específicamente matemáticos. Este modelo es independiente del sistema administrador de bases de datos que posteriormente utilicemos para su implementación; aun más, es independiente de cualquier elemento computacional.

El próximo paso a la etapa del modelado conceptual, es el *modelado lógico*. Aquí se establece qué modelo de base de datos convencional se utilizará: *Relacional, Jerárquico, Red*, etc., apoyándose en el sistema de gestión de bases de datos que vaya a utilizarse.

Posteriormente, el *modelado interno* nos permite describir registros internos o almacenados, punteros, organizaciones secuenciales, indizadas, direccionadas, agrupamientos, etc.). Aquí el objetivo es lograr una implementación lo más eficiente posible del esquema lógico.

La última etapa se corresponde con el *almacenamiento físico*, en donde se determinan las estructuras físicas (registros físicos, bytes, bits, campos, ítems, etc.). La estructura física resultante del proceso de diseño se ha rellenar con los valores que se obtienen por observación de los sucesos del mundo real. Estas cadenas de bits estarían carentes de significado si no dispusiéramos de los medios que nos permiten recorrer el camino inverso, pasando de nuevo al mundo real con la ayuda de un lenguaje de manipulación, por medio del cual actualizaremos o recuperaremos los datos almacenados en la base reincorporándoles su contenido semántico y obteniendo la información que necesita el usuario.

A partir del esquema lógico, y teniendo en cuenta las restricciones, más el sistema de gestión a utilizar, los recursos físicos, etc., se procede al diseño, programación e implementación de procesos, carga de datos y explotación de la base.

### **Modelo Conceptual**

Para obtener el modelo conceptual, se pueden considerar dos subetapas:

- *Etapa Análisis de requisitos*: esta es la etapa de percepción, identificación y descripción de los fenómenos y componentes del mundo real a analizar. Aquí es donde debemos preguntarnos *¿qué representar?*. Entonces, mediante el estudio de las reglas que lo rigen, la recopilación documental y entrevistas a los usuarios de distintos niveles, llegamos a elaborar un esquema descriptivo de la realidad.

*Profesora Edilma Olinda Gagliardi*

- *Etapas de conceptualización:* en esta etapa se refina el esquema descriptivo, estructurándolo y nos respondemos la pregunta *¿cómo representar?* Y es aquí donde se presenta un modelo de datos expresado en términos matemáticos, satisfaciendo propiedades tales como coherencia, plenitud, no redundancia, simplicidad, fidelidad, etc.

Para la etapa análisis de requisitos se estudian y aplican las siguientes técnicas de recopilación de datos:

- *La Observación:* distintas técnicas de recopilación de datos a través de la observación; normas y recaudos para realizar una observación sistemática y controlada; sus diversas modalidades; diferentes instrumentos de observación; ventajas y limitaciones que posee.
- *La Recopilación Documental:* como técnica de recopilación de datos; diferentes clases de documentos (escritos, estadísticos, cartográficos, de imagen y sonido, objetos, etc.); fuentes primarias y secundarias; problemas de autenticidad, validez y significación.
- *La Entrevista:* diversas modalidades de la entrevista; su preparación y principios directivos; ventajas, dificultades y límites de la entrevista.
- *El cuestionario:* entrevista versus encuesta como modalidades del método de encuesta; modalidades en el uso de cuestionarios; aspectos formales referentes al envío y recepción de los mismos; ventajas y limitaciones de la técnica.

Para la etapa de conceptualización se estudia y utiliza el *Modelo Algebraico*, que nos permite representar las entidades identificadas como conjuntos; los datos propios, denominados atributos, como funciones; y las vinculaciones entre entidades como relaciones. Adicionalmente, se especifican todas las restricciones referentes a entidades, atributos y relaciones.

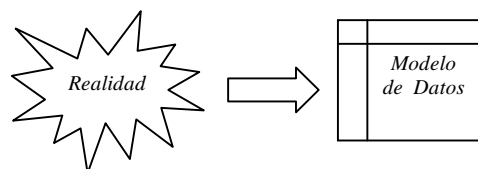
La práctica final consiste fundamentalmente en que dada una realidad que se requiere informatizar, llevando a cabo las dos etapas: análisis de requisitos y etapa de conceptualización, se obtenga el modelo algebraico de la misma.

Este reporte está dedicado al modelo conceptual, presentando el modelo algebraico. Las demás fases de modelado y etapas de metodología de desarrollo de base de datos, se deja para apuntes posteriores.

## 2. Modelo Algebraico

Utilizar el lenguaje natural como herramienta de modelización no nos sirve dado las diversas interpretaciones que se pueden encontrar para lo mismo y no se puede dejar a la computadora que seleccione la interpretación más adecuada.

Por ello, es necesario contar con herramientas formales, libres de diversas interpretaciones, que nos permitan modelar la realidad, mediante el uso de un lenguaje formal y abstracto, no ambiguo ni de programación que nos permitan representar las variables que conforman la realidad o los parámetros de interés. Esto es, dada una realidad, obtener su *modelo de datos*.



Un modelo es una representación simplificada de un sistema real. Si la representación es adecuada, podremos interrogar al modelo en caso de querer conocer alguna propiedad de la realidad. Debe poder representar tanto las características estáticas como las dinámicas. Podemos decir que contiene las siguientes

componentes: i) un conjunto de objetos y sus interrelaciones (características estáticas e invariantes y propiedades de los objetos).; ii) restricciones sobre los objetos, sus interrelaciones y las operaciones definidas sobre ellas; iii) un conjunto de operaciones, o lenguaje, que representan características dinámicas.

Entonces, un *Modelo de Datos* se puede definir como un conjunto de conceptos, reglas y convenciones bien definidos (matemáticamente) que nos permiten aplicar una serie de abstracciones a fin de describir y manipular los datos de un cierto mundo real que deseamos almacenar en la base de datos. Nosotros para modelar utilizaremos una herramienta denominada: *Modelo Algebraico*.

Podemos destacar un par de objetivos en los modelos de datos:

- *Formalización*: el modelo de datos permite formalizar estructuras permitidas y restricciones, nos proporciona una base para el lenguaje de datos y facilita una apreciación más objetiva de la rigidez o flexibilidad de las estructuras de datos, ayudando a la evaluación y selección de un sistema de gestión de bases de datos.
- *Diseño*: dado que el modelo de datos es fundamental en el desarrollo de una metodología de diseño de bases de datos, en el cual se basan los otros componentes de la metodología tales como lenguaje, documentación, etc., permite además prever el impacto de los cambios del mundo real en nuestro sistema de información.

El Modelo Algebraico utiliza conjuntos, funciones y relaciones para describir una realidad.

El conjunto de objetos y sus interrelaciones resultan abstracciones realizadas sobre la estructura del sistema real y constituye la estructura del modelo.

Las relaciones entre los objetos se analizan en términos de sus componentes (unarias, binarias, etc.).

En cuanto a las restricciones, permiten representar en forma más precisa y restringir el conjunto de posibles estados de una bases de datos. Tenemos restricciones *inherentes* (aspectos semánticos), *explícitas* (propiedades expresadas mediante un mecanismo); e *implícitas* (derivan de otras propiedades).

El conjunto de operaciones puede dividirse en operaciones de manipulación (crear, modificar y destruir objetos y relaciones); y operaciones de consulta que permitan identificar objetos por medio de sus propiedades.

A continuación describimos brevemente cada una de las componentes y en secciones posteriores profundizamos su estudio.

### 3. Entidades

Para ello consideramos a la realidad objeto de análisis como nuestro *Universo* de discurso. A partir de ello, comenzamos un proceso de clasificación para individualizar grupos de objetos cuyas características los hagan similares. Estos grupos o conjuntos son los que denominamos *entidades*.

Una *entidad* es en sí la ley de conformación que constituye un conjunto dado.

La ley de conformación puede estar dada por comprensión o extensión, según sea apropiado usar. Se puede definir un ente o elemento como cualquier objeto real o abstracto que existe en la realidad y acerca del cual queremos almacenar información en la base de datos. Hall (1976) la define como “algo en la realidad objetiva que existe o puede ser pensado”. Según ANSI (1977), es “una persona, lugar, cosa, concepto o suceso, real o abstracto, de interés para la empresa”. La estructura genérica que describe un conjunto de elementos aplicando la abstracción de clasificación se denomina *entidad*, mientras que cada uno de los ejemplares o elementos de esa entidad o conjunto suele denominárselo *ente*; por tanto la entidad es el resultado de la clasificación de un conjunto de elementos, obtenidos de un universo de discurso.

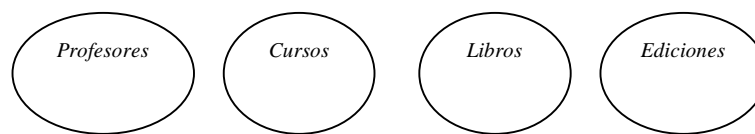
Por ejemplo, en el ámbito académico, *Cursos* es una entidad que describe las características comunes de un conjunto de cursos; un elemento de esta entidad *Cursos* será, por ejemplo, “Diseño de Bases de Datos

Relacionales” y otro “Introducción a lo Sistemas de Bases de Datos”. Otra entidad podría ser *Profesores* y un ejemplar de la misma sería “Sr. Sánchez”.

El conjunto de elementos que conforman una entidad en un momento dado conformará la *instancia* de esa entidad, mientras que ley de conformación que la define conforma el *esquema* de la entidad. Chen habla de conjunto de entidades (entity set), lo que para él es análogo a entidad. Nosotros nos referiremos a *Entidad* y a sus *elementos* (*entes*) de aquí en adelante.

Si un elemento pertenece a una entidad, debe satisfacer el predicado que describe la ley de conformación del conjunto.. Matemáticamente, se define entonces como:  $E = \{e / \phi(e)\}$ , siendo  $e$  un elemento de la entidad  $E$  y  $\phi$  el predicado asociado a  $E$ . Por ejemplo, la entidad *Profesores*, cuyo predicado asociado es “*Persona que ejerce o enseña una materia o arte*” tiene un elemento en su instancia “Sr. Sánchez”, que cumple dicho predicado.

La representación gráfica de una entidad es un círculo (pensando en los conjuntos) en cuyo interior está el nombre de la entidad:



Existen dos clases de entidades:

- *Regulares*, que son aquellas cuyos elementos tienen existencia por sí mismos (como *Cursos*, *Profesores* y *Libros*), y
- *Débiles*, en las cuales la existencia de un elemento depende de la existencia de un cierto elemento en otra entidad. Por ejemplo, *Ediciones* depende de *Cursos*, puesto que la desaparición de un determinado curso de la base de datos hace que desaparezcan también todas las ediciones de dicho curso. Las entidades débiles se representan de la misma manera .

Aunque es fácil entender el concepto de entidad, no lo es su definición formal; por esta razón, se ha afirmado a veces que es preferible dejar el término sin definir. El problema surge cuando cierto objeto del mundo real se cataloga en ocasiones como una entidad, mientras que en otras se considera una propiedad de una entidad o de una relación; por ejemplo, *color* puede verse como una entidad con sus propias propiedades o bien una propiedad de una entidad.

Algunos autores han intentado precisar el concepto de entidad. Así, Tardieu *et al.* (1979) propone tres reglas generales que debe cumplir una entidad:

- Tiene que tener existencia propia.
- Cada elemento de una instancia de una entidad debe poder distinguirse de las demás.
- Todos los elementos de una instancia de una entidad deben tener las mismas propiedades.

Sin embargo, la primera regla no es aplicable a las entidades débiles, cuya existencia depende de la existencia de una entidad regular de la cual dependen. La segunda de estas condiciones supone la obligación de un identificador que permita distinguir los distintos elementos de una entidad, lo que tampoco es universalmente aceptado (ni por los autores, ni por los modelos, ni por los productos). Respecto a la tercera: ¿hasta qué punto todos los elementos de una entidad tienen las mismas propiedades en el caso de que el modelo admita valores nulos (especialmente los inaplicables)?

#### 4. Atributos

Las distintas propiedades o características de una entidad o de una relación, se denomina *Atributo*. Los atributos toman valores de uno o varios *dominios*. Así, el conjunto de posibles valores que puede tomar una cierta propiedad se denomina *dominio para tal propiedad*. Por ejemplo, propiedades de *Profesores* pueden ser su nombre y apellido, DNI, domicilio, etc.; de *Cursos*, pueden ser nombre del curso, duración, requisitos, etc. y los dominios serán aquellos conjuntos de los cuales podemos obtener esos valores “Sr. Sánchez”, 25657789, etc.

Se define *dominio* como un conjunto de valores homogéneos con un nombre. Para saber si un valor pertenece a un dominio determinado, comprobaremos que cumple el predicado que el dominio lleva siempre asociado. Matemáticamente se expresa:  $D = \{v_i: p(v_i)\}$ ; donde  $D$  es el dominio,  $v_i$  es un valor y  $p$  es el predicado asociado a dicho dominio.

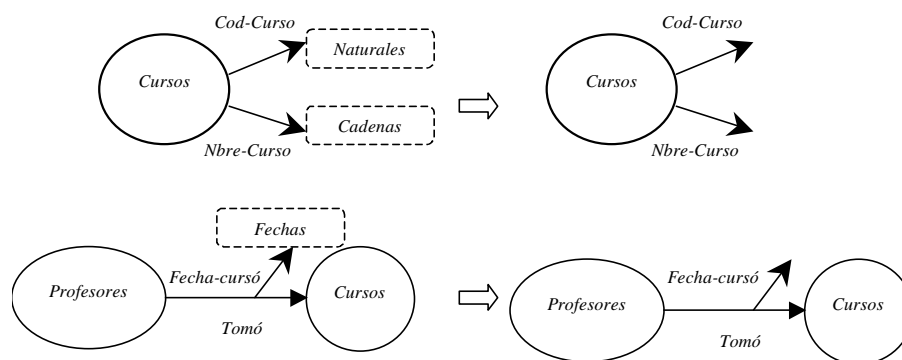
Por ejemplo, el valor “*inglés*” se toma del dominio *Idiomas*, y cumple el predicado de ser uno de los idiomas posibles del conjunto { “*español*”, “*inglés*”, “*francés*” }; el dominio *Nombre de Curso* es una cadena de caracteres de alguna longitud dada.

Un dominio puede definirse por *intensión*, especificando el tipo de datos, por ejemplo, carácter 30 para el *Nombre de Curso* o fecha para la *Fecha de Edición*; o por *extensión*, declarando el valor de cada elemento del dominio, como es el caso de *Idioma*. El dominio es una componente del modelo que tiene existencia propia independiente de cualquier otro componente. No debemos confundir dominio con entidad. Dominio siempre va asociado a un atributo (propiedad).

Podemos decir que el atributo le da una determinada interpretación al dominio (o a los dominios) en el contexto de una entidad o de una relación. Matemáticamente consiste en una función definida sobre una entidad o relación, cuyo rango es un conjunto de tipo dominio ( varios dominios):  $A_i: E_j \rightarrow D_k$  o  $A_i: E_j \rightarrow D_1 \times D_2 \times \dots \times D_n$ ; donde  $A_i$  es el atributo,  $E_j$  es la entidad o la relación, y  $D_i$  es un dominio.

La representación gráfica de un atributo consiste en cualificar con su nombre el arco que une el dominio con la entidad (o relación). Sin embargo, para simplificar la representación gráfica, será suficiente el arco con el nombre del atributo.

Los atributos pueden pertenecer a las relaciones, como hemos mencionado adelantadamente. Por ejemplo, el curso tiene dos atributos: su código de identificación y su nombre; también podríamos querer mantener información acerca de la fecha en que un profesor cursó un cierto curso. Entonces este atributo *fecha-cursó* está definido sobre la relación *Tomó*, cuyo rango es el dominio *Fechas*.



A diferencia de los dominios que tienen vida propia, es decir, existen por sí mismos, la existencia de un atributo está ligada a la existencia de la correspondiente entidad o relación. Así la fecha de nacimiento de un profesor *fecha-nac-prof* no tiene sentido si de nuestro esquema desaparece la entidad *Profesores*; sin embargo, el dominio *Fechas* puede existir con independencia de cualquier otro tipo de entidad o atributo.

## Atributos derivados

Es preciso, en el modelo algebraico, analizar la existencia de redundancias, por los problemas de inconsistencias a los que puede dar lugar. Decimos que un elemento cualquiera es redundante cuando puede ser eliminado sin pérdida semántica (o información –LLJ). Veamos a continuación la redundancia en los atributos, específicamente en los atributos derivados.

Entendemos por *atributos derivados* o *calculados*, aquellos que se obtienen a partir de otros ya existentes, por lo que, aunque son redundantes, no dan lugar a inconsistencias, siempre que en el esquema se indique su condición de derivados y la fórmula mediante la que han de ser calculados, como un dato más en el diccionario de datos.

Por ejemplo, el atributo *número de ediciones*, que puede ser calculado a partir de los ejemplares de edición mediante la relación *Tiene*.

Un atributo derivado puede ser calculado en dos momentos distintos:

- en actualizaciones: que pueden provocar cambios en su valor, puesto que el atributo derivado se calcula y almacena.
- cuando se recupera: en este caso no está almacenado y se calcula cuando se realiza una consulta.

Calcularlo de una u otra forma, dependerá del número de actualizaciones frente al de recuperaciones. Para un atributo derivado, su valor se calcula con las restricciones que comprueban la consistencia entre valores que están almacenados en la base de datos, y no por su introducción.

## Restricciones sobre atributos

Se consideran dos tipos de restricciones de integridad: *restricciones sobre valores* y *restricciones estructurales*.

Las restricciones sobre valores se establecen mediante la definición del dominio, que permite limitar los valores del dominio y, por ende, los de los atributos sobre él definidos, a los siguientes casos:

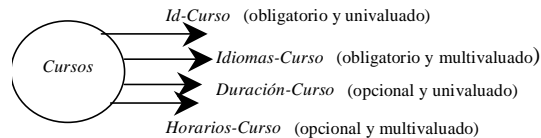
- a los de un determinado tipo de datos,
- o restringirlos a los comprendidos en un rango,
- o bien declarar los valores posibles en el caso de que la definición se haga por extensión.

Las *restricciones estructurales* se refieren tanto a atributos como a relaciones (referidas a la semántica de las relaciones que vemos en la siguiente sección). A continuación veamos las restricciones estructurales referidas a atributos:

- Cada entidad debe tener uno o varios atributos (simples y/o compuestos) que identifiquen unívocamente cada una de los elementos del conjunto, denominado *identificador candidato*(IC) o *clave*.  
Cuando un IC es compuesto, el número de los atributos que lo componen debe ser mínimo, en el sentido de que la eliminación de cualquiera de ellos le haría perder su carácter identificador. Luego todo IC debe cumplirla condición de ser unívoco y mínimo. Entre los IC se elige uno como *Identificador Principal* ( IP) o *clave primaria* y el resto serán *identificadores alternativos* (IA) o *claves secundarias*. La representación gráfica de estos atributos queda reflejada en la figura subrayándolos.
- Se puede permitir también atributos *multivaluados* y *opcionales* (nulos o faltantes).  
En general, un atributo toma para cada elemento de una entidad, un único valor de cada dominio(s) subyacente(s). Por ejemplo, un libro tiene un único título, un único ISBN, etc., pero también existen atributos

que pueden tomar más de un valor, como por ejemplo un curso puede impartirse en más de un idioma, o un profesor puede tener más de un teléfono; estos atributos reciben el nombre de *multivaluados* frente a los *univaluados* que toman un solo valor.

Por otro lado, puede ser obligatorio que un atributo de una entidad tome, como mínimo, un valor del (o de los) dominio (s) subyacente (s) para cada elemento de la entidad; en este caso decimos que el valor de ese atributo es *obligatorio* (no puede ser nulo) para todo elemento de la entidad; en otro caso es *opcional*. La prohibición de valores nulos para un atributo (no admitir la opcionalidad) y la de que un atributo puede tomar más de un valor (no admitir que sea multivaluado) son restricciones específicas sobre la estructura de los atributos, al igual que la declaración de atributos identificadores.

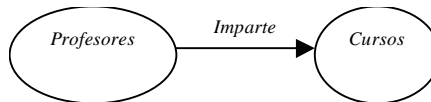


## 5. Relaciones

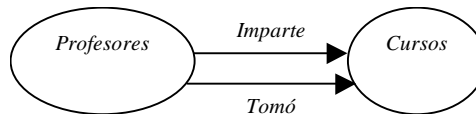
Se entiende por *Relación* una asociación, vinculación o correspondencia entre entidades. Denominaremos *esquema de relación* a la estructura genérica que describe un conjunto de relaciones. Desde un punto de vista matemático, una relación se define del siguiente modo: dados los conjuntos  $D1, D2, \dots, Dn$  llamados dominios, una relación  $R$  es un subconjunto del producto cartesiano de  $n$  dominios,  $R \subseteq D1 \times D2 \times \dots \times Dn$ ;  $n$  el grado de tipo de relación. Así,  $R = \{ \langle e1, e2, \dots, en \rangle / ei \in Di, i=1..n, \Psi(e1, e2, \dots, en) \}$ .

Por ejemplo, *Imparte* es la vinculación entre *Profesores* y *Cursos*; un elemento de la instancia la relación *Imparte* es la vinculación entre el profesor “Sr. Sánchez” ( $e1$ ) y el curso “Diseño de Bases de Datos Relacionales” ( $e2$ ); *Imparte*(Sr. Sánchez, Diseño de Bases de Datos Relacionales).

Representaremos la relación mediante un arco orientado y etiquetado con el nombre de la relación, que toque las entidades vinculadas, como se puede observar en la siguiente figura, en donde establecemos la relación *Imparte* entre *Profesores* y *Cursos*:

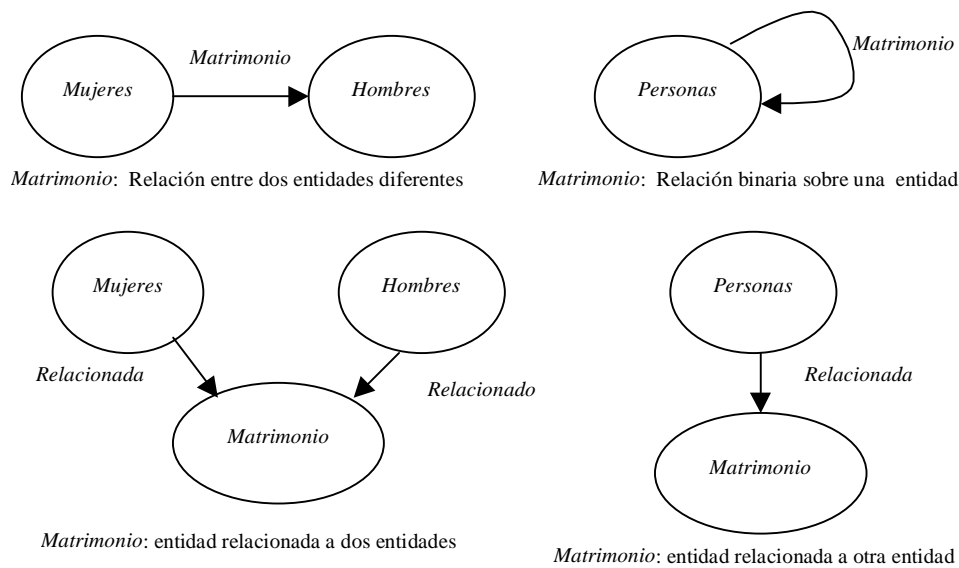


Entre dos entidades puede existir más de una relación, como por ejemplo: qué cursos *Tomó* cada profesor.



Varios autores no están de acuerdo en que se distinga entre entidades y relaciones. Así, para Date (1993), tal distinción no tiene sentido ya que un mismo objeto del mundo real puede ser visto como entidad o como relación; todo depende del dominio de la aplicación. Para Date, una relación es un tipo especial de entidad.

Por ejemplo, el matrimonio puede ser visto como una relación entre dos personas o como una entidad en sí misma. En el segundo caso, la relación puede considerarse un tipo especial de entidad cuya existencia depende de la existencia de las entidades a las que relaciona, pero su especificidad hace necesario definirla como un elemento del modelo de datos distinto de la entidad; puede tener atributos propios. Dependerá del universo del discurso que se esté analizando (al igual que ocurre, como ya se ha comentado, con los atributos y las entidades), para decidir si es entidad o relación.



## Semántica de las relaciones

En esta parte vamos a comenzar viendo los elementos de una relación que aparecen en el modelo básico así como algunos aspectos semánticos como las dependencias en existencia y en identificación. Posteriormente, en otras secciones, iremos extendiendo la semántica de las relaciones.

## Esquema de relación

En el *esquema de relación* se pueden distinguir los siguientes elementos:

- **Nombre:** Al igual que las entidades, los dominios y los atributos, cada relación tiene un nombre que la distingue unívocamente del resto, y mediante el cual ha de ser referenciada. Como hemos indicado anteriormente, en la representación gráfica de la relación (un arco dirigido etiquetado) siempre ha de aparecer el nombre, el cual aporta semántica al modelo.
- **Grado:** Es el número entidades que participan en una relación. Así, una relación es de grado 2 (o binaria) cuando asocia dos entidades. Un caso particular de relaciones de grado 2 son las reflexivas (también llamadas recursivas), las cuales asocian una entidad consigo misma; en la próxima figura se muestra el tipo de relación reflexiva *Consta* que asocia *Tema* con *Tema*, en la que se refleja la posibilidad de que un cierto tema (por ejemplo, *informática*) está compuesto por (sub)temas (por ejemplo, *bases de datos*, *sistemas operativos*, *lenguajes*, etc.).
- **Tipo de correspondencia:** Es el número máximo de ejemplares de una instancia de una entidad que pueden estar asociados, en una determinada relación, con un elemento de otra(s) entidades. Para representarlo gráficamente, bien se pone una etiqueta con (1:1), (n:1), (1:n) o (n:m) según corresponda, al lado la etiqueta de la relación y la orientación del arco.

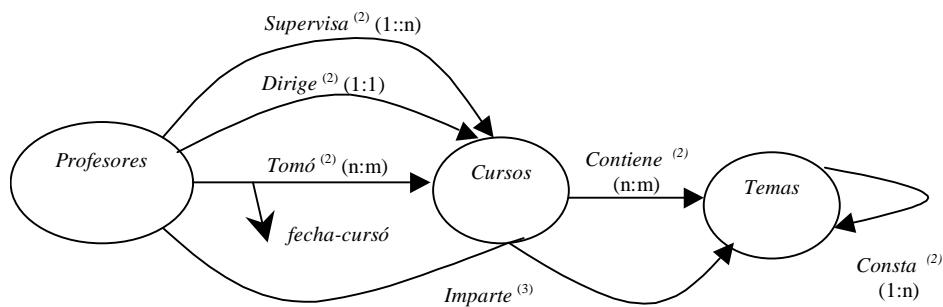
Pueden existir también tipos de relación que asocien más de dos tipos de entidad (grado n, n>2) como en la próxima figura. En este ejemplo se muestra un profesor con los temas y cursos que imparte.

- **Papel (“rol”):** Es el rol que cada una de las entidades tiene en la relación. Se define del siguiente modo:

$$R^n = \{(x_1, x_2, \dots, x_n) / x_1 \in E_1, x_2 \in E_2, \dots, x_n \in E_n, \text{“}x_1 \text{ es...tal que } x_2 \dots \text{es } x_n\text{”}\}$$

$$\text{Tomó} = \{(x_1, x_2) / x_1 \in \text{Profesores}, x_2 \in \text{Cursos}, \text{“}x_1 \text{ es un profesor que tomó el curso } x_2\text{”}\}$$





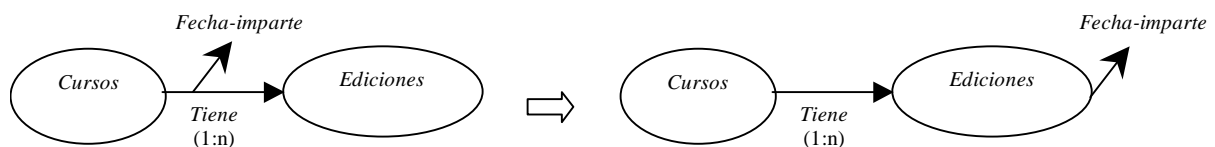
## Propiedades

Podemos considerar las siguientes propiedades que brindan información adicional sobre las relaciones, que también se pueden considerar restricciones de las mismas. Estas propiedades se estudian respecto de su semántica y no de su extensión (instancia). Supongamos una relación  $R \subseteq A \times B$ .

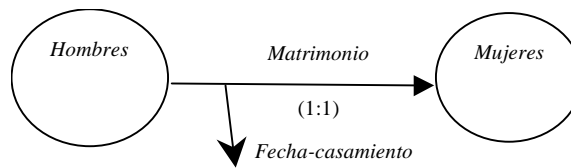
- **Unívoca**:  $R(a)$  tiene una sola imagen. Es decir,  $|R(a)| = 1$ .
- **Total**: Todo elemento de A está relacionado con algún elemento de B. Es decir,  $R(a)$  está siempre definido; sino es parcial.
- **Función**: unívoca y total.
- **Inyectiva**:  $R(x) = R(y)$  entonces  $x = y$ .
- **Suryectiva**: Todo elemento de B es imagen de algún elemento de A. Es decir, para todo  $b$  perteneciente a B, existe algún  $a$  tal que  $R(a) = b$ .
- **Multívoca**: Inyectiva y suryectiva.
- **Biunívoca**: unívoca e inyectiva.
- **Biyectiva**: Biunívoca, suryectiva y total.
- **Irrestricida**: sin restricciones, también de grado mayor que dos, y cualquier otro caso diferente.

## Atributos en las relaciones

Cuando una relación (1:n) tiene un atributo asociado es inmediata la demostración matemática de que el atributo puede llevarse a la entidad cuya cardinalidad máxima es  $n$ . En el ejemplo siguiente, el atributo *fecha\_imparte* debe llevarse a *Edición*, con plena independencia.



Semánticamente, sin embargo, puede ser, en ocasiones, de interés conservar el atributo dependiendo de la relación. Éste es el caso donde tenemos la relación *Matrimonio* (1:1) entre *Hombres* y *Mujeres*, vigentes, con el atributo *fecha-casamiento*. Por ser la relación (1:1), para cada par (*hombrex*, *mujery*) existe una sola fecha válida de celebración del matrimonio, fecha que no es una propiedad de ninguno de los dos ejemplares, sino del hecho de la unión entre ellos, es decir, de la relación.



Los atributos de las relaciones ( $n:m$ ), son propios de la misma y no de las entidades vinculadas por la relación; pueden incluso ser multivaluados, como en el ejemplo donde un profesor puede dar el mismo curso en varias fechas distintas, por lo que *fecha* es un atributo multivaluado.

### Dependencia en existencia y en identificación

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

Es interesante distinguir, dentro del tipo de relación débil, la dependencia en *existencia* y la dependencia en *identificación*.

- Se dice que hay dependencia en existencia cuando los elementos de una entidad débil no pueden existir si desaparece el elemento de la entidad regular del cual dependen.
- Se dice que existe dependencia en identificación cuando además de cumplirse la condición anterior, los elementos de la entidad débil no se pueden identificar por sí mismos, mediante sus propios atributos, y requieren añadir el identificador principal de la entidad regular de la cual dependen. Se ve claramente que una dependencia en identificación es siempre una dependencia en existencia (no ocurre lo contrario), la relación es débil en ambos casos.

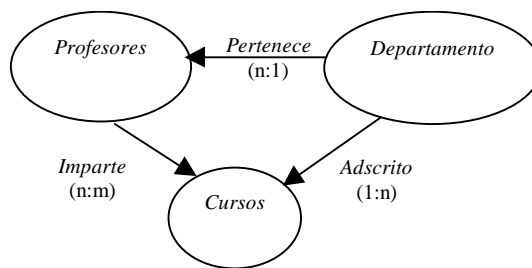
### Control de redundancia. Relaciones redundantes

Hemos tratado la redundancia en atributos (atributos derivados) en apartados anteriores. Ahora es preciso considerar la redundancia en las relaciones, denominadas también por algunos autores relaciones derivadas.

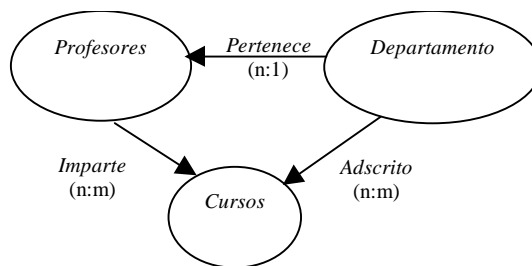
Se dice que una relación es redundante cuando su eliminación no implica pérdida de semántica porque existe la posibilidad de realizar la misma asociación de elementos por medio de otras relaciones.

Es condición necesaria, aunque que no suficiente, para que una relación sea redundante, que forme parte de un ciclo, por lo que hay que estudiar detenidamente los ciclos en el diagrama del modelo algebraico.

En el ejemplo siguiente se da un ciclo entre *Profesores*, *Cursos* y *Departamento*, por lo que en principio es posible que aparezca alguna relación redundante. Supongamos que un profesor sólo puede impartir cursos de doctorado que estén adscritos al departamento al que él pertenece; si se conocen los cursos de doctorado que imparte un profesor y el departamento al que está adscrito cada curso, se deduce inmediatamente a qué departamento pertenece dicho profesor. De forma análoga, dado un departamento, si sabemos qué cursos de doctorado tiene adscritos y los profesores que imparten dichos cursos, conoceremos qué profesores pertenecen a dicho departamento. Por lo expuesto la relación *Pertenece* entre las entidades *Profesores* y *Departamento* es redundante y su eliminación no produce pérdida de información.



En la siguiente figura, a pesar de que también existe un ciclo, no hay ninguna relación redundante. En este ejemplo la semántica es distinta y un departamento puede no tener adscritos cursos de doctorado; además un mismo curso puede estar adscrito a distintos departamentos y puede haber profesores que no impartan ningún curso. La relación *Pertenece* no puede deducirse en este caso de las otras dos, ya que aunque sepamos los cursos que ha impartido un profesor y los departamentos a los que están adscritos dichos cursos, no podemos saber a qué departamento en concreto pertenece dicho profesor; Tampoco se tiene esta información para los profesores que no imparten ningún curso. La relación *Imparte* tampoco es redundante ya que un curso de doctorado puede ser impartido por diversos departamentos a cada uno de los cuales pertenecen varios profesores, por lo que no se puede saber qué profesor en concreto imparte un determinado curso. Por último, la relación *Adscrito* tampoco es redundante, ya que un curso impartido por un profesor no tiene por qué estar necesariamente adscrito al departamento al que pertenece dicho profesor: hay departamentos que no tienen cursos adscritos y los profesores de estos departamentos pueden colaborar en cursos adscritos a otros departamentos distintos del suyo.



Existen otros casos en los que la relación, a pesar de poder ser deducida a partir de otras presentes en el esquema, no se puede eliminar porque posee atributos.

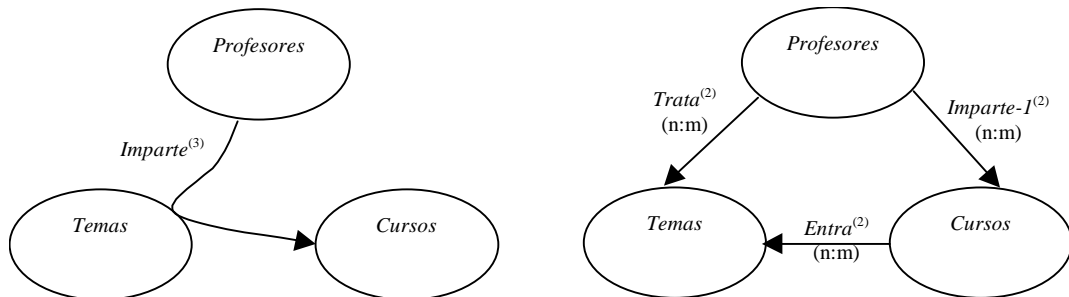
Como norma general, la existencia de un ciclo no implica la existencia de relaciones redundantes. Deben estudiarse con mucho detenimiento las categorías así como la semántica que aportan las relaciones, para poder afirmar con seguridad que existen relaciones redundantes. Habrá que analizar si al eliminar una relación es *siempre* posible el paso, tanto en un sentido o en el inverso, entre las dos entidades unidas por la relación que se considera redundante, y habrá que comprobar también que no se pierdan atributos.

En resumen, para que una relación pueda ser eliminada por redundante se tiene que cumplir:

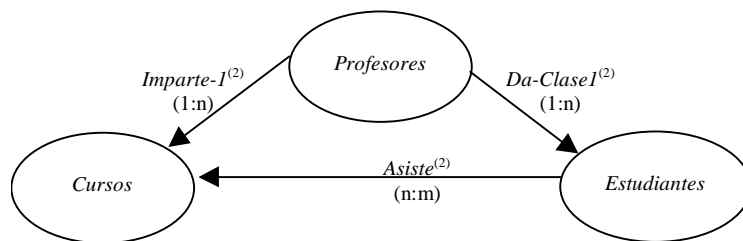
- Que exista un ciclo
- Que las relaciones que componen el ciclo sean equivalentes semánticamente
- Que se puedan asociar los ejemplares de las dos entidades que estaban relacionadas, incluso habiéndose eliminado la relación
- Que la relación o bien no tenga atributos o bien éstos puedan ser transferidos a otra a fin de no perder su semántica.

## Relaciones de grado superior a 2

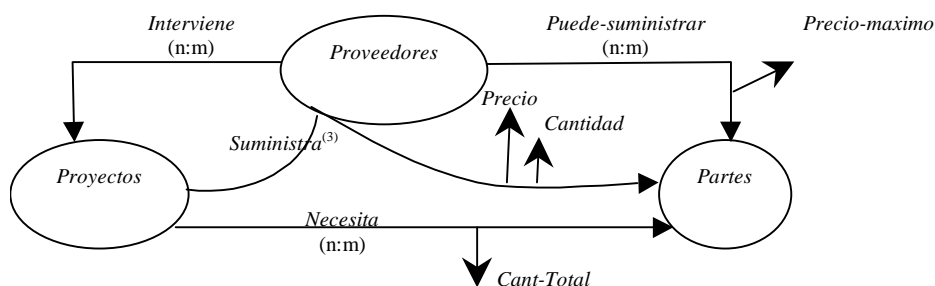
Cuando se presenta una relación de grado  $n$  ( $n > 2$ ), es preciso analizar si realmente es de tal grado, porque podría ser posible su descomposición, sin pérdida de información (semántica). Por ejemplo, en la relación *Imparte* podemos observar que la información almacenada que asocia tres entidades, se refiere a que un profesor imparte un tema en un curso; si sustituimos esta relación por las tres *Imparte-1*, *Trata* y *Entra*, de ellas no se puede deducir los temas que trata un profesor en un curso determinado, aunque sepamos los cursos que ha impartido ese profesor, qué temas entran en esos cursos y cuáles son los temas que trata ese profesor. Por tanto, no es posible la descomposición de esta relación de grado 3 en tres de grado 2 sin pérdida de semántica o información.



Sin embargo, en el ejemplo de la relación *Imparte* entre *Profesores*, *Cursos* y *Estudiantes*, sí puede ser descompuesta sin perder semántica en las relaciones *Imparte1*, *Da-clase* y *Asiste*, ya que éstas aportan la misma semántica que la relación de grado tres. Cuando un tipo de relación de grado  $n$  ( $n > 2$ ) puede ser sustituido por otros de grado menor, sin pérdida de información, se debe llevar a cabo tal sustitución.



La existencia de una relación de grado superior a 2 no es incompatible con la existencia de relaciones de menor grado en las que participen los mismos tipos de entidad. Por ejemplo, la relación de grado 3 *Suministra* coexiste con las tres relaciones de grado 2 (*Puede-suministrar*, *Interviene* y *Necesita*), ya que éstas recogen las piezas que puede suministrar, etc., mientras que la de grado 3 representa las piezas que, de hecho, están siendo suministradas para un cierto proyecto por un determinado proveedor; por tanto, la semántica de la relación ternaria es distinta de la de las relaciones binarias y el usuario podría necesitar que se mantuvieran tres relaciones (*Interviene* sí es redundante con respecto a *Suministra*).



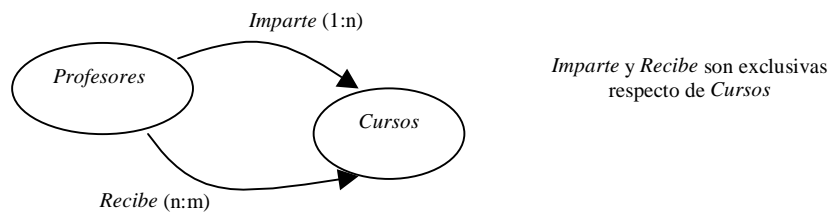
## Otras restricciones sobre relaciones

Existen otras restricciones que afectan a las relaciones, como son: restricción de exclusividad, restricción de exclusión, restricción de inclusividad y restricción de inclusión.

- *Restricción de Exclusividad*

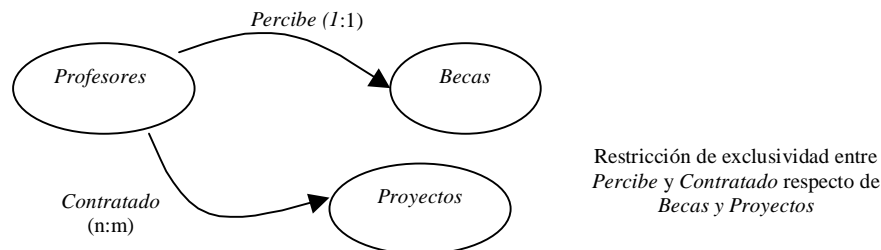
Decimos que dos o más relaciones tienen restricción de exclusividad con respecto a una entidad que participa en ambas relaciones cuando cada ejemplar de dicha entidad sólo puede pertenecer a una de las relaciones, y que en el momento en que pertenezca a uno ya no podrá formar parte de la otra relación.

Por ejemplo, si suponemos que un profesor puede impartir cursos de doctorado o recibirlos, pero no ambas cosas, tendríamos una relación *Imparte* y otra *Recibe*, entre *Profesores* y *Cursos*, con una restricción de exclusividad entre sí. En nuestro modelo, lo dejaremos indicado con un cartel cuáles son las relaciones exclusivas.



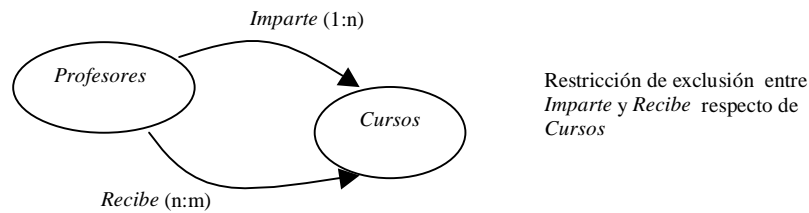
El significado de este modelo es el siguiente: un profesor puede impartir o no cursos de doctorado, y puede o no recibirlos, pero si un profesor imparte estos cursos no puede recibirlos y viceversa. Un curso de doctorado es impartido por un solo profesor, pero a él pueden asistir varios profesores o ninguno.

No es obligatorio que las relaciones exclusivas lo sean respecto a la misma entidad; en este caso *Cursos*, sino que podrían serlo respecto a distintas entidades. Por ejemplo, si un profesor percibe una beca no puede estar contratado en un proyecto.



- *Restricción de Exclusión*

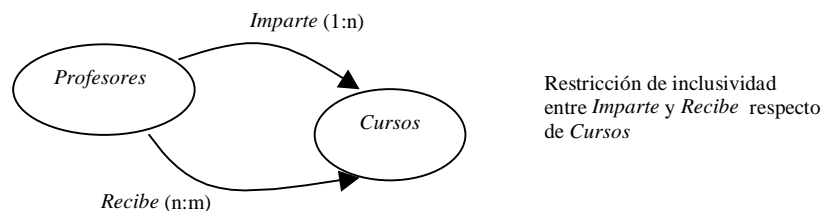
La restricción de exclusividad en el ejemplo anterior indicaba que un profesor podía impartir o recibir cursos, pero no ambas cosas; si el profesor no es doctor podrá recibir cursos de doctorado y en caso contrario impartirlos. Supongamos ahora que se permite a un profesor ya doctor matricularse en cursos aunque él, a su vez, esté impartiendo otros cursos. En este caso la restricción que debemos imponer es que un profesor no esté impartiendo y recibiendo el mismo curso. Es decir, que todo elemento de *Profesores* que esté unido a un elemento de *Cursos* mediante la relación *Imparte*, no podrá estar unido al mismo elemento de *Cursos* mediante la relación *Recibe*. En este caso decimos que existe una restricción de exclusión y la notamos como en el caso anterior.



- **Restricción de Inclusividad**

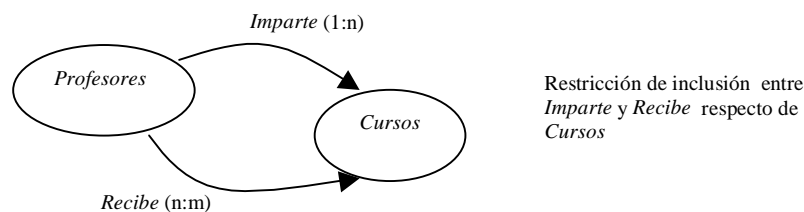
Supongamos ahora que se desea imponer la restricción de que sólo pueden impartir clases en nuestro programa de doctorado aquellos profesores que hayan realizado al menos un curso dentro de este mismo programa, aunque no tiene por qué ser el mismo que él imparte. Aplicamos entonces una restricción de inclusividad entre dos (o más) relaciones con respecto a uno de las entidades que participa en uno de las relaciones tiene necesariamente que participar en la otra. También lo dejamos indicado mediante un cartel.

En este ejemplo se representa que si un profesor participa en *Imparte* tiene que participar necesariamente en *Recibe*.



- **Restricción de Inclusión**

A veces es preciso imponer una restricción más fuerte: si un profesor imparte un curso es porque previamente ha tenido que recibir dicho curso. Aplicamos pues una restricción de inclusión, por la cual todo profesor que esté unido a un curso mediante la relación *Imparte*, tiene necesariamente que estar unido al mismo curso mediante la relación *Recibe*.



Si se considera la dimensión temporal se pueden tener casos más complejos de modelado, como por ejemplo que todo profesor que imparta un curso tiene que haberlo recibido antes (restricción de inclusión con el histórico recibe), pero no puede estar recibéndolo a la vez que lo imparte (restricción de exclusión con el actual de recibe).

## Generalización / especialización

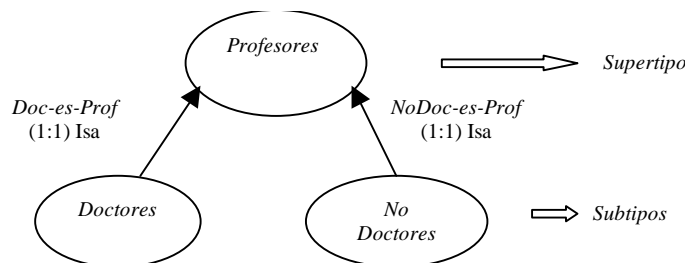
Considerando las redes semánticas (originadas en el campo de la inteligencia artificial, introducido por QUILLIAN (1968)), la *jerarquía de generalización / especialización* se considera como un caso especial de relación entre varias entidades (*subtipos*) y una más general (*supertipo*) cuyas características son comunes a todos los subtipos. La relación que se establece entre los subtipos y el supertipo corresponde a la noción de “*es\_un*” o más precisamente “*es\_un\_tipo\_de*”, también conocida como relaciones *Isa* por “*es un*”.

Aunque existen distintas convenciones para representar estas jerarquías de generalización / especialización, nosotros sólo dejaremos indicado junto a la etiqueta del nombre de la relación que es tipo Isa.

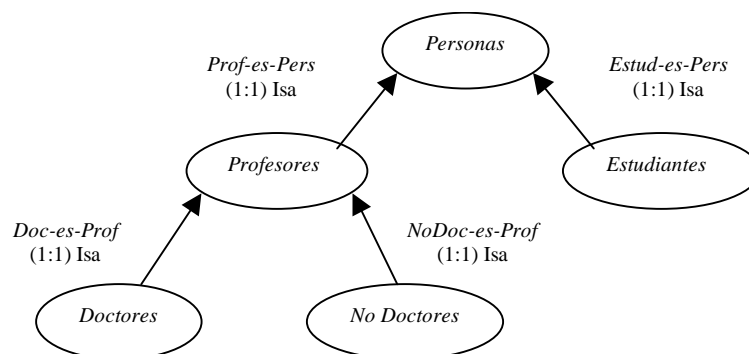
Esta clase de relación tiene la característica de que todo elemento de un subtipo es también un elemento del supertipo, aunque no sucede lo contrario. La aparición de estas jerarquías en el modelo de bases de datos puede surgir de dos formas distintas:

- *Generalización*: Se observa que dos o más entidades comparten varios atributos y/o relaciones, de donde se deduce la existencia de una entidad de nivel superior (supertipo) que contiene los atributos y las relaciones comunes a todos los subtipos.
- *Especialización*: Se observa que una entidad tiene ciertos atributos y/o relaciones que tiene sentido para unos elementos pero no para otros, por lo que es conveniente definir uno o varios subtipos que contengan estos atributos y/o relaciones específicas, dejando en el supertipo los que son comunes.

Por tanto, si nos movemos de los subtipos hacia el supertipo, se trata de una generalización; mientras que si primero identificamos el supertipo y, a partir de él, llegamos a los subtipos, se trata de una especialización.



Puede ocurrir que se formen, por generalización y/o especialización, jerarquías a más de un nivel donde un subtipo es, a su vez, supertipo de otros donde una jerarquía a dos niveles donde uno de ellos se ha obtenido por generalización de profesor y estudiante en persona, y el otro nivel por especialización de profesor en doctor y no doctor.



Otra característica muy importante de esta clase de relaciones es la *herencia*, ya que, en principio todo atributo del supertipo pasa a ser un atributo de los subtipos; por ejemplo, tanto los doctores como los no doctores son profesores, por lo que heredarán todos los atributos de *Profesores* (*Código*, *Nombre*, *DNI*, etc.).

En este tipo de abstracción los atributos comunes a todos los subtipos, incluidos los identificadores, se asignan al supertipo, mientras que los atributos específicos se asocian al subtipo al cual pertenecen. Del mismo modo, las relaciones que afectan a todos los subtipos se asocian al supertipo, dejándose para los subtipos las relaciones específicas en las que sólo participa el correspondiente subtipo.

La abstracción de generalización/especialización tiene algunas restricciones semánticas de las que nos ocuparemos a continuación. Según sea si los subtipos se solapan o son disjuntos, y si la unión de los subtipos recubre o no al supertipo, se pueden distinguir cuatro clases de generalización. Si un mismo elemento del

supertipo puede pertenecer a más de un subtipo habrá solapamiento, y si sólo puede pertenecer a uno de los subtipos existiría *exclusividad*; por otro lado, si todo elemento del supertipo tiene que pertenecer a algún subtipo tendremos *totalidad*, y si, por el contrario, no tiene obligatoriamente que pertenecer a algún subtipo habrá *parcialidad*.

En el ejemplo visto hay una jerarquía total de subtipos disjuntos, puesto que:

- Tanto un doctor como un no doctor son profesores (por tener una jerarquía de generalización)
- Un mismo profesor no puede ser a la vez doctor y no doctor (exclusividad)
- Todo profesor tiene que ser obligatoriamente un doctor o un no doctor (totalidad)

Por ejemplo, el supertipo *Documentos* y los subtipos *Libros* y *Artículos* forman una jerarquía disjunta y parcial, que se traduciría en lo siguiente:

- Tanto un artículo como un libro son documentos
- Un mismo documento no puede ser a la vez un artículo y un libro (exclusividad)
- Un documento puede no ser ni un artículo ni un libro (parcialidad)

Una jerarquía parcial sólo puede surgir por especialización, ya que en la generalización los ejemplares aparecen a nivel del subtipo y, por tanto, no puede existir ningún ejemplar en el supertipo que no pertenezca a alguno de los subtipos. Hay que observar que la parcialidad de la jerarquía significa la admisión de nulos en el atributo discriminante, mientras que el solapamiento implica que el atributo discriminante sería un grupo repetitivo.

Pueden existir jerarquías múltiples que parten de un supertipo común, como por ejemplo la división de la entidad *Cursos* en dos jerarquías distintas, una según el tema y la otra por el idioma; *Temas* e *Idiomas* son los atributos discriminantes, cada uno en su correspondiente jerarquía.

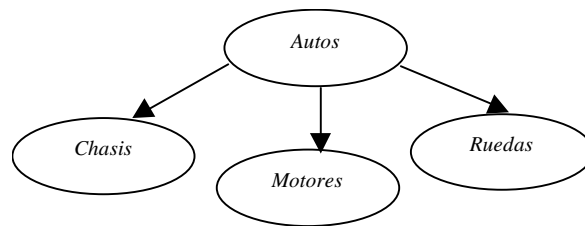
Hasta ahora hemos considerado que se trataba de *jerarquías* estrictas, es decir, que podían solaparse ejemplares de subtipos que dependían del mismo supertipo, pero no subtipos de ramas distintas; pueden ocurrir, sin embargo, que un subtipo tenga más de un supertipo, formándose un verdadero *retículo o red* de generalización. En este caso, la herencia ya no es posible, sino que se convierte en múltiple, pudiéndose presentar conflictos a la hora de heredar atributos. Existen modelos de datos que en caso de conflicto definen un orden de prioridad en la herencia; otros, por el contrario, permiten heredar atributos iguales de dos supertipos distintos pero teniendo que renombrar alguno de ellos.

## Agregación

La *agregación* es una abstracción que permite representar entidades compuestas que se obtienen por unión de otras más simples. A la entidad compuesta nos referimos como el *todo*, mientras que los componentes son las *partes*.

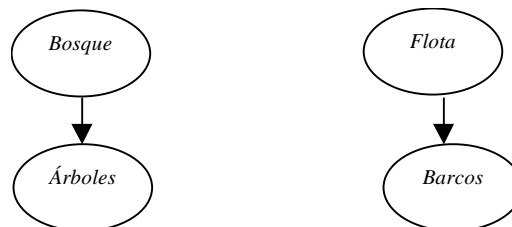
- La *agregación compuesto/componente*, como su propio nombre indica, es una abstracción que permite representar que un *todo* que se obtiene por la unión de diversas *partes*, las cuales pueden ser tipos de objetos distintos y que desempeñan diferentes papeles en la agregación. Por ejemplo, un coche puede verse como la unión del chasis, el motor y las cuatro ruedas.





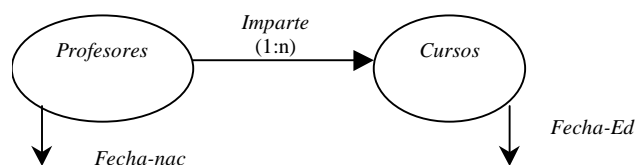
- La *agregación miembro/colección* es la abstracción que permite representar un *todo* como una colección de partes, donde todas las partes son de un mismo tipo y desempeñan el mismo papel. Por ejemplo, un bosque es un todo formado por la agregación de árboles; cada árbol es una parte, pero todos ellos son un mismo tipo y desempeñan el mismo papel.

A veces se desea establecer un orden entre las partes. Por ejemplo, una flota esta compuesta por barcos pero, a diferencia de lo que ocurre en el bosque, en la flota cada barco tiene un determinado orden. Esto se representa mediante una restricción de orden, donde los barcos se ordenan, dentro de la flota, según el valor del atributo *Núm\_barco*. Esta restricción se puede tomar, igualmente, en los actuales modelos de objetos. Sin embargo, en el diseño lógico en el modelo relacional, esta restricción no se puede tomar directamente.



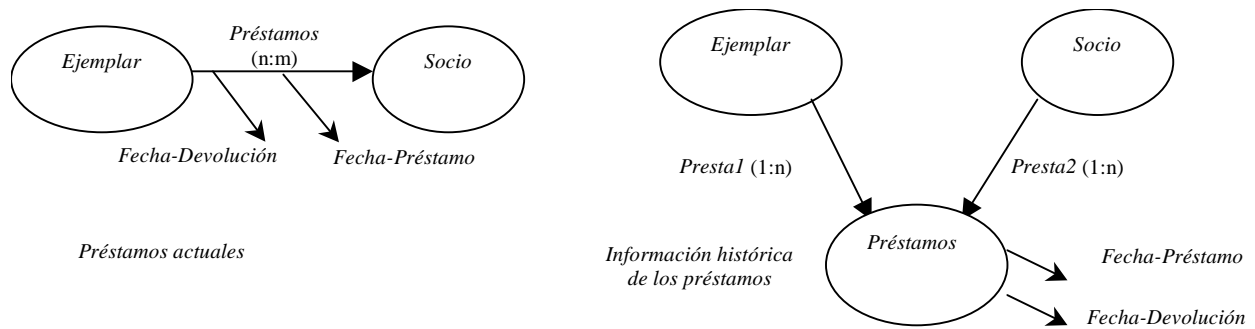
## Temporalidad

El tratamiento de la *dimensión temporal* en las bases de datos es un tema complejo sobre el cual hay una intensa labor de investigación. Es indudable la necesidad de establecer un método semántico y gráfico que recoja de algún modo, en el esquema conceptual, el transcurso de tiempo y su influencia en la forma en que varían los datos. La aproximación más simple la constituyen atributos de tipo fecha que aparecen asociados a algunas entidades.



En este caso, la fecha de nacimiento de un profesor o la fecha en la que se impartió un curso son datos temporales recogidos en el esquema, pero se trata sólo de atributos que han de recibir un tratamiento especial en cuanto a las operaciones, y no se puede considerar realmente una aproximación semántica a la dimensión temporal.

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 mismos. La diferencia entre estos tipos de esquemas se puede apreciar en la siguiente figura donde la parte izquierda se refiere a los préstamos actuales de libros en una biblioteca, de forma que una vez finalizado el préstamo la correspondiente información desaparece de la base de datos, sin que exista fichero histórico. En la parte derecha se representa el esquema conceptual de todos los préstamos que se han realizado en la biblioteca, recogiendo, además, el periodo del tiempo que duró el préstamo.



En caso de tratarse de datos históricos, los tipos de entidad o de relación correspondientes tendrán asociados siempre atributos de tipo *fecha*. Para sucesos puntuales, es decir, sin duración, bastará con un solo atributo de este tipo, mientras que para poder almacenar hechos que transcurren en un período de tiempo determinado necesitaremos una *fecha\_inicio* y una *fecha\_fin*.

A veces resulta interesante representar la evolución de un tipo de entidad a lo largo del tiempo y aparece la noción de *estado*. Por ejemplo, si deseamos reflejar si un libro está en la biblioteca o se encuentra prestado, añadiremos al tipo de entidad un atributo que denominamos *estado*, que indicará en qué estado concreto se encuentra el elemento y que en muchos casos lleva asociado otro atributo, que es la fecha en la que se ha producido el cambio de estado; es también habitual en este tipo de aplicaciones que se desee tener constancia de la evolución de los estados, en cuyo caso se podría crear una nueva entidad, como *Situación*, que tendría como atributos, entre otros posibles, *estado* y *fecha*. Observando el mundo real de los sistemas de información nos damos cuenta que este mecanismo se utiliza sobre todo en la gestión de expedientes.

## 6. Operaciones

Hemos dicho que tenemos dos clases de operaciones: de *manipulación* y de *consultas*.

El primer tipo de operaciones modifica el estado de la bases de datos, permitiendo crear, modificar y destruir objetos o relaciones. Así se modifica el estado de la base de datos alterando los objetos o propiedades de los objetos y/o relaciones.

El segundo grupo permite extraer o buscar datos en la base, no modifican el estado de la base.

Un conjunto perfectamente definido de operaciones se denomina una *transacción*. Una transacción se compone de acciones elementales sobre objetos individuales. Este concepto es de fundamental importancia para asegurar la consistencia entre operaciones atómicas concurrentes y requiere que todos los accesos a la base de datos estén dentro de transacciones.

En esta etapa usaremos operaciones del álgebra y para las estructuras de control utilizaremos el lenguaje de diseño visto en asignaturas de introducción a la computación. En etapas posteriores, veremos diversos lenguajes de manipulación y consultas a bases de datos (álgebra relacional, sql, etc.).

## 7. Notas Y Referencias Bibliográficas

- Modelo Entidad-Relación (MER), propuesto por Peter Chen [Chen1976] y [Chen1977]. Posteriormente, realizaron aportes otros autores tales como [Paul1980], Ferg[1984], [ShangShixuan1984, [ElsMari1985].
- Modelo Algebraico (MA): simplificación de las ideas propuestas en el Modelo Entidad-Relación, utilizando elementos matemáticos sencillos. Ideas propuestas por Dr. Turull Torres. Apuntes de cátedra.

- Introducción a las Bases de Datos Relacionales; Mendelzon y Ale; Prentice Hall 2000.
- El presente apunte se realizó tomando como base el Capítulo 2 del libro: “Diseño de Base de Datos Relacionales”, de los siguientes autores: A. de Miguel, M. Piattini, E. Marcos. Se adaptaron las ideas presentadas para MER a MA.