

Bases de Datos

Tema 1 *Modelos de Datos*

Francisco Ruiz
sep-2000

*documentación preparada con ayuda de Esperanza Marcos (Universidad Rey Juan Carlos) y Mario Piattini
(Universidad de Castilla-La Mancha)*

Tema 1

Modelos de Datos

Complementar con:

** capítulo 1 del libro “Diseño de Bases de Datos Relacionales”.
De Miguel, A.; Piattini, M.; Marcos, E.; Ra-Ma, 1999.*

** capítulo 3 del libro “Fundamentos y Modelos de Bases de
Datos” (2ª edición). De Miguel, A.; Piattini, M.; Ra-Ma, 1999.*

- *En el diseño, gestión y utilización de Bases de Datos, como en cualquier otra actividad científica o tecnológica, es fundamentalmente trabajar con conceptos basados en un cierto modelo teórico o conceptual.*
- *En este tema se presentan los principales conceptos de modelos de datos: esquemas, objetos, propiedades, asociaciones, operaciones, restricciones, etc.*
- *También se realiza una clasificación de los diversos tipos de modelos de datos existentes y se estudian los principales mecanismos de abstracción utilizados en esta área.*

- **Principales:**
- [de Miguel et al, 1999]
 - cap. 1
 - De Miguel, A.; Piattini, M.; Marcos, E.; Diseño de Bases de Datos Relacionales. Ra-Ma, 1999.
- [de Miguel y Piattini, 1999]
 - cap. 3
 - De Miguel, A.; Piattini, M.; Fundamentos y Modelos de Bases de Datos (2ª edición). Ra-Ma, 1999.
- **Otras:**
 - - Tsichritzis, D.; Lochovsky, F.; Data Models. Prentice Hall, 1982.
 - - Puerba, S.(editor); Handbook of data management (3rd edition). Auerbach Pub., 1999.

Indice

1. Introducción.
2. Modelo, Esquema y Ejemplar.
 - 2.1. Clases de Modelos de Datos.
3. Mecanismos de Abstracción.
 - 3.1. Clasificación.
 - 3.2. Agregación.
 - 3.3. Generalización.
 - 3.4. Asociación.
 - 3.5. Jerarquías de abstracciones.
4. Concepto de Modelo de Datos.
 - 4.1. Estática.
 - 4.2. Dinámica.
5. Restricciones de Integridad.
 - 5.1. Componentes de una restricción.
 - 5.2. Clasificación de las restricciones.
6. Los Modelos de Datos en el proceso de diseño de una Base de Datos.

- **Repasar** los conceptos básicos siguientes:
 - Datos vs Información,
 - Sistema de Información,
 - Base de Datos,
 - Sistema de Gestión de Bases de Datos (SGBD),
 - Arquitectura ANSI para un SGBD.
 - Independencia de datos lógica y física.
- *Referencia:*
 - *Capítulos 1 y 2 del libro “Fundamentos y Modelos de Bases de Datos” (2ª edición). De Miguel, A.; Piattini, M.; Ra-Ma, 1999.*

Modelo

“Construcción mental a partir de la realidad en la que se reproducen los principales componentes y relaciones del segmento de la realidad analizada”

Dos acepciones:

- **Reproducción simplificada de la realidad** (ciencias empíricas)
- **Realidad propiamente dicha** (pintor)

Modelo de Datos

“Dispositivo de abstracción que nos permite ver el bosque (esto es, la información contenida en los datos) en oposición a los árboles (valores individuales de los datos)”; Tsichritzis y Lochovsky (1982).

“Conjunto de herramientas conceptuales para describir la representación de la información en términos de datos. Los modelos de datos comprenden aspectos relacionados con: estructuras y tipos de datos, operaciones y restricciones”; Dittrich (1994)

“Conjunto de conceptos, reglas y convenciones que permiten describir y manipular los datos de la parcela de un cierto mundo real que deseamos almacenar en la base de datos”; de Miguel, Piattini y Marcos (1999).

$$\mathbf{LD = MD + Sintaxis}$$

Ejemplos:

$$\text{SQL} = \text{MDR} + \text{Sintaxis}$$

$$\text{QBE} = \text{MDR} + \text{Sintaxis (distinta)}$$

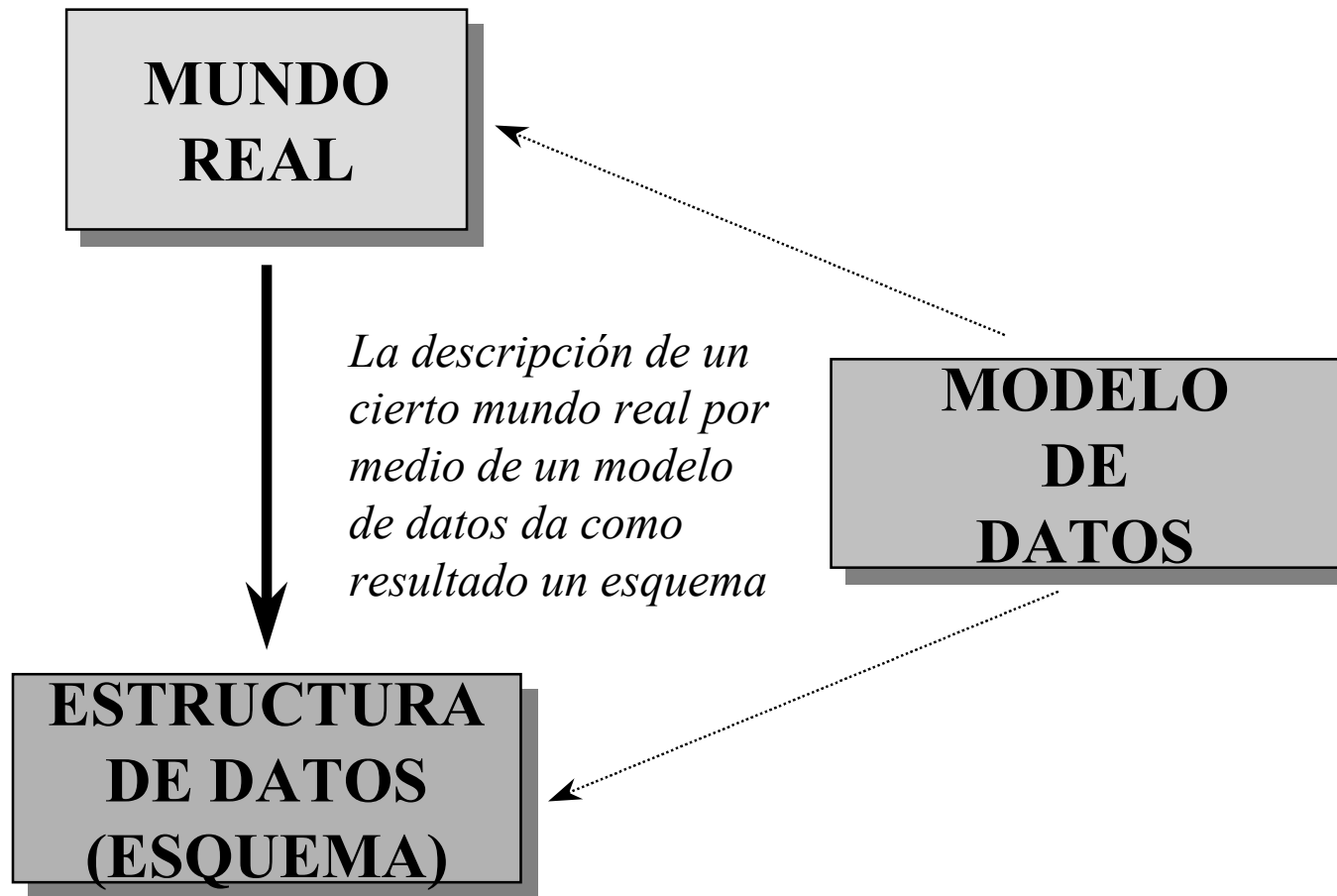
$$\text{OQL} = \text{MO} + \text{Sintaxis}$$

Modelo de Datos vs Lenguaje de Datos

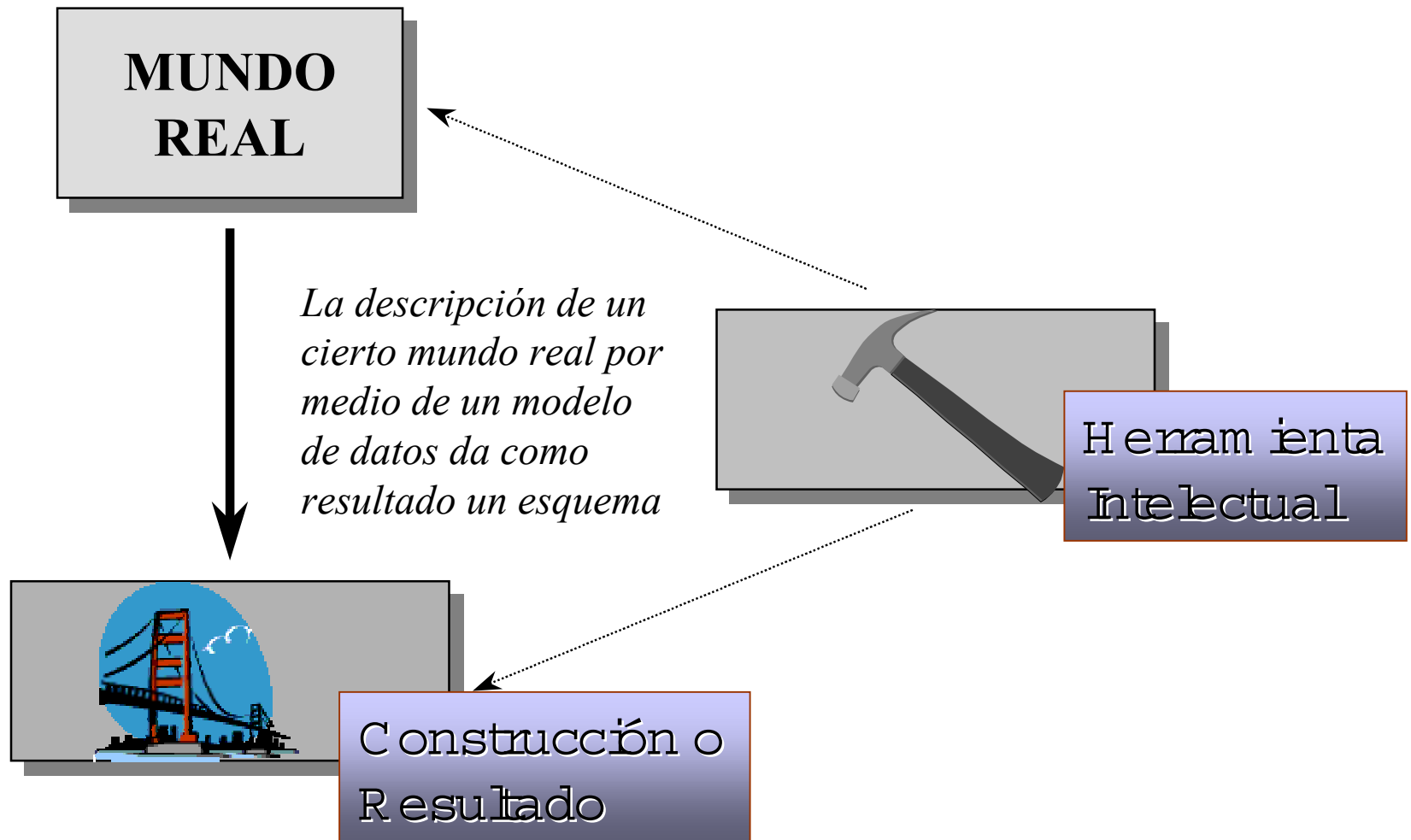
Esquema

"La descripción específica de un determinado mini-mundo en términos de un modelo de datos se denomina esquema (o esquema de datos) del mini-mundo. La colección de datos que representan la información a cerca del mini-mundo constituye la base de datos", Dittrich (1994).

"Representación de un determinado mundo real (universo del discurso) en términos de un modelo de datos"; de Miguel, Piattini y Marcos (1999).

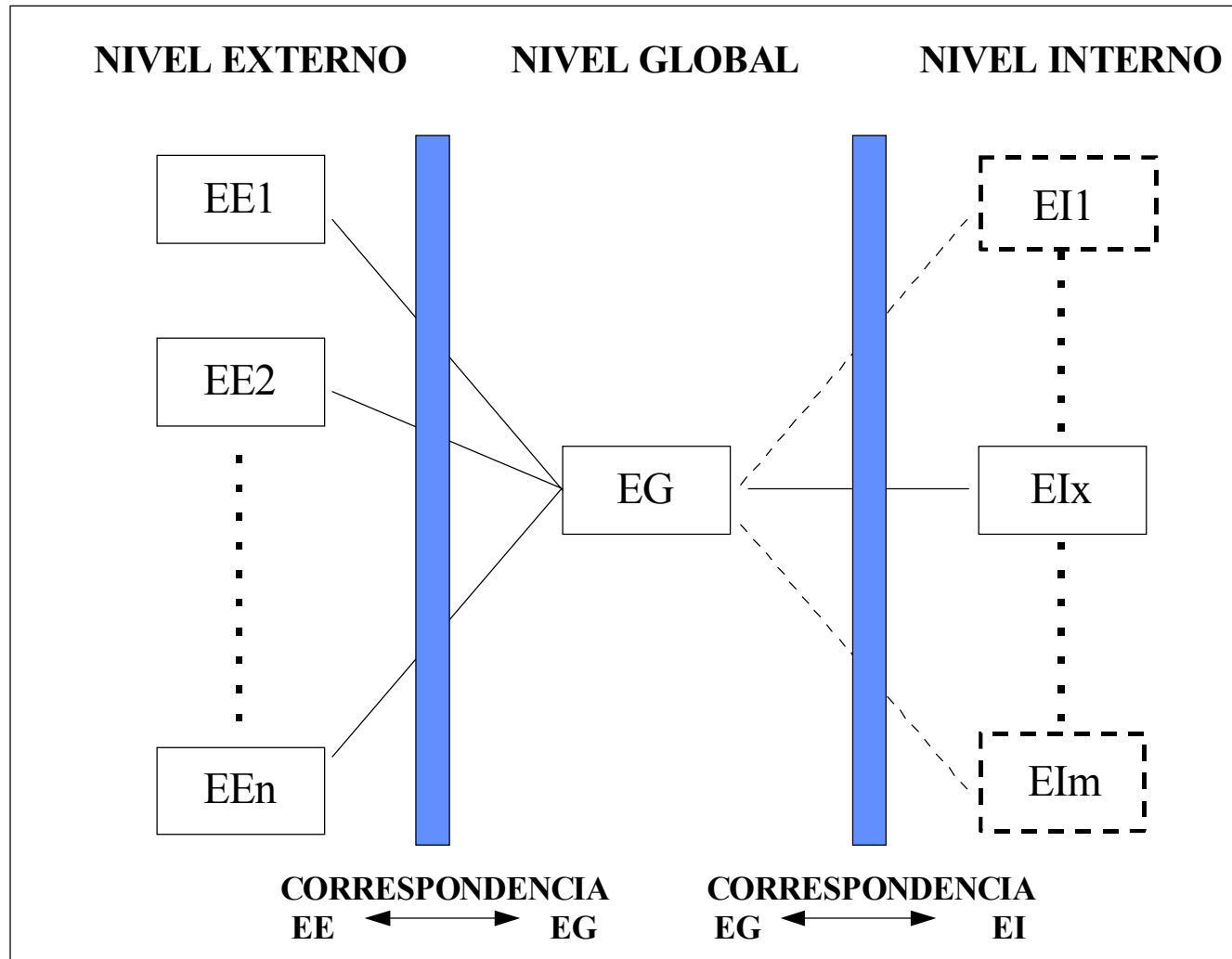


Diferencia entre modelo y esquema en Bases de Datos
(1)



Diferencia entre modelo y esquema en Bases de Datos
(2)

- También es preciso distinguir entre **esquema** y **ejemplar**.
 - Un ejemplar de un esquema son los datos que en un determinado momento se encuentran almacenados en el esquema.
 - La colección de ejemplares de todos los elementos de un esquema en un momento determinado constituyen un ejemplar del esquema.
 - Al igual que en los lenguajes de programación existen *variables* (constituidas por un *tipo* y un *contenido*), las cuales tienen, en un momento determinado, un cierto *valor*; en las bases de datos se debería hablar de *variables de base de datos*, cuyo tipo sería el *esquema* y su contenido todos los posibles valores del esquema; su valor, en un momento determinado, sería un *ejemplar* del esquema.
 - Nosotros utilizaremos la expresión “base de datos” en el sentido abstracto de todos los posibles ejemplares, y cuando queramos referirnos a su contenido en un cierto momento o bien hablaremos de un ejemplar o bien diremos “*la base de datos en el momento i*” (BD_i).



Diferentes esquemas para los tres niveles de la arquitectura ANSI

a) ESQUEMA GLOBAL***CURSO DE DOCTORADO******/* Tipo de Objeto */*****CURSO**

CODIGO Character (5)
 NOMBRE Character(50)
 NUM_HORAS Numérico (3)
 DESCRIPCION Carácter
 variable (200)
 Clave CODIGO

PROFESOR

CODIGO Character (3)
 NOMBRE Character(30)
 DNI Character (10)
 DIRECCION Character(50)
 SALARIO Numérico (7)
 Clave CODIGO

IMPARTE

PROFESOR Character (3)
 CURSO Character (5)
 FECHA INICIO Fecha
 FECHA FINAL Fecha
 Clave PROFESOR, CURSO

b) ESQUEMA INTERNO***CURSOS DE DOCTORADO******/* Registro Almacenado */*****CURSO**

COD_CURSO Byte (3)
 NOMBRE Byte (50)
 NUM_HORAS Byte (2)
 DESCRIPCION Byte (200)
 Índice de 2 niveles sobre
 COD_CURSO

PROFESOR

COD_PROFE Byte (2)
 NOMBRE Byte (30)
 DNI Byte (10)
 DIRECCION Byte (50)
 SALARIO Byte (4)
 Índice 1 nivel sobre
 COD_PROFE

IMPARTE

FECHA_INI Byte (8)
 FECHA_FIN Byte (8)
 PUNTERO_CURSO Byte (4)
 PUNTERO_PROFESOR Byte (4)

c) ESQUEMAS EXTERNOS**EN ORACLE FORMS**

(listado de cursos)

CODIGO Varchar2 (5)
 NOMBRE Varchar2 (50)
 HORAS Number (3,0)
 DESCRIPCION Varchar2 (200)

EN PASCAL

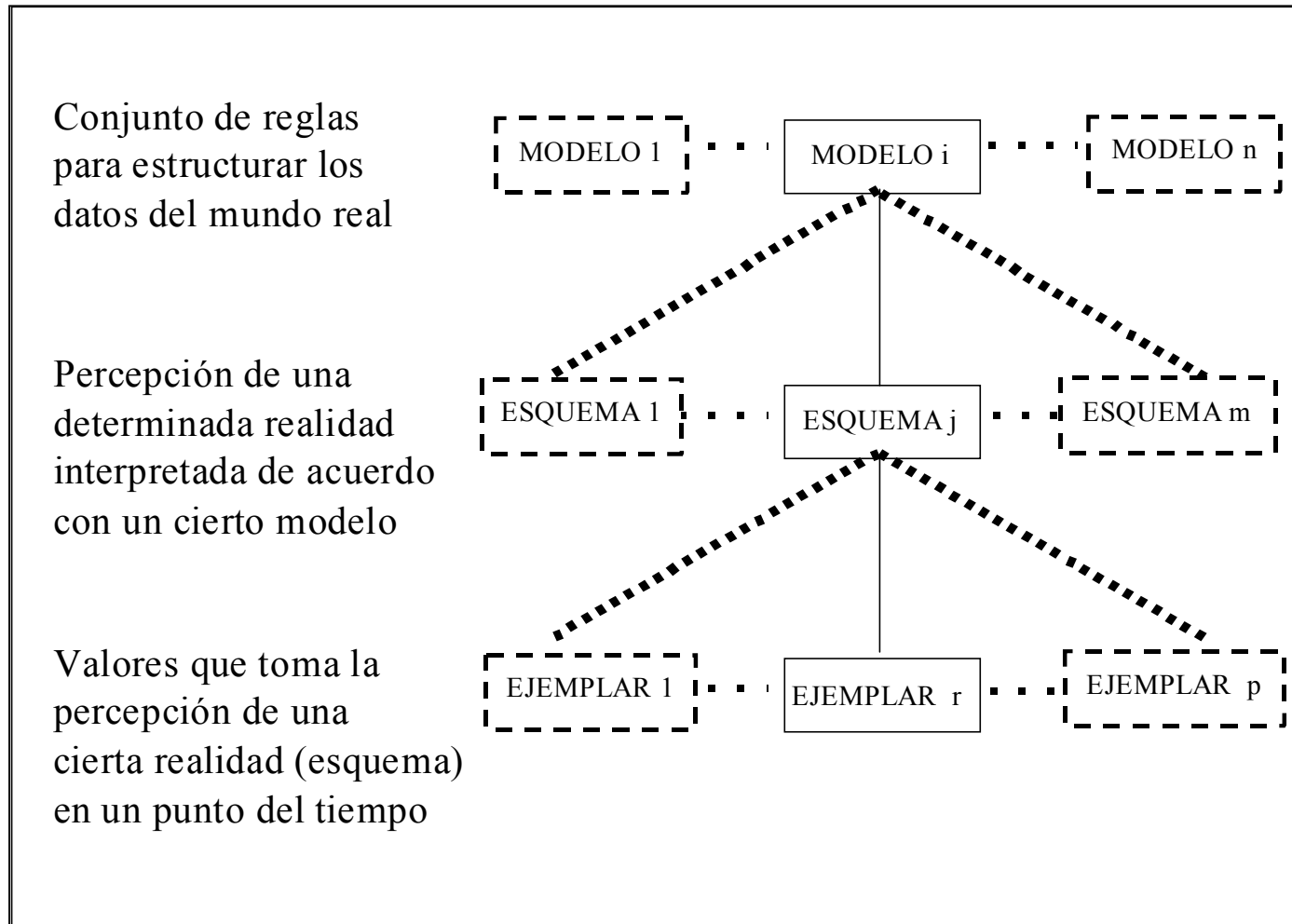
(asignación cursos a profesores)

CURSO Char (5)
 NOMBRE Char (30)
 HORAS Integer(10)
 COD_PROFE Char (3)
 PROFESOR Char (30)
 INICIO String (10)
 FIN String (10)

Ejemplos de esquemas global, interno y externos

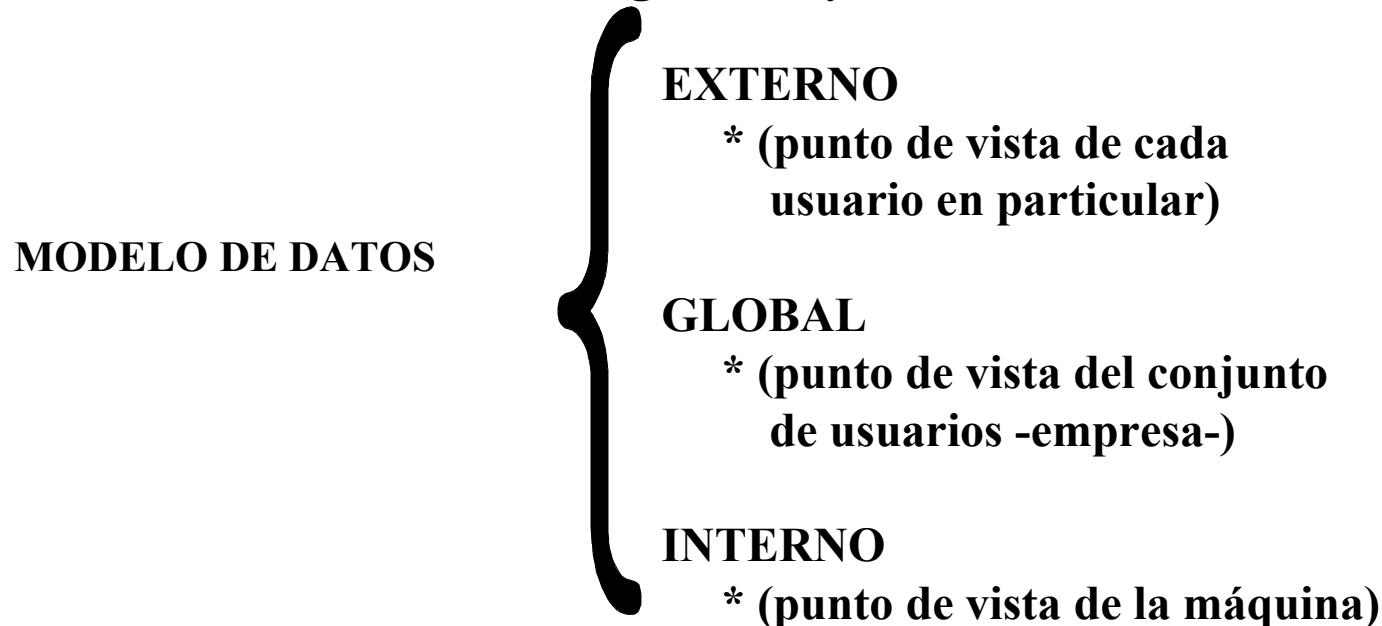
EJEMPLARES DE CURSO	EJEMPLARES DE PROFESOR	EJEMPLARES DE IMPARTE
00101	001	00101
Introducción a las Bases de Datos	Andrés García Ruiz	001
030	12312330	12/12/1997
Este curso tiene como objetivo...	C/ Conde de Vistahermosa	20/12/1997
	5.621.333	
00102	002	00101
Seguridad de la información	Mercedes García Arias	003
020	50179030	01/03/1998
La seguridad en la informática...	C/ Río Miño	12/03/1998
	3.928.352	
00203	003	00203
Diseño de Bases de Datos	Julio López Pérez	002
100	52342876	05/11/1997
Dentro de las bases de datos...	C/ Segovia	07/12/1997
	6.564.125	

Ejemplares que pueblan los esquemas anteriores

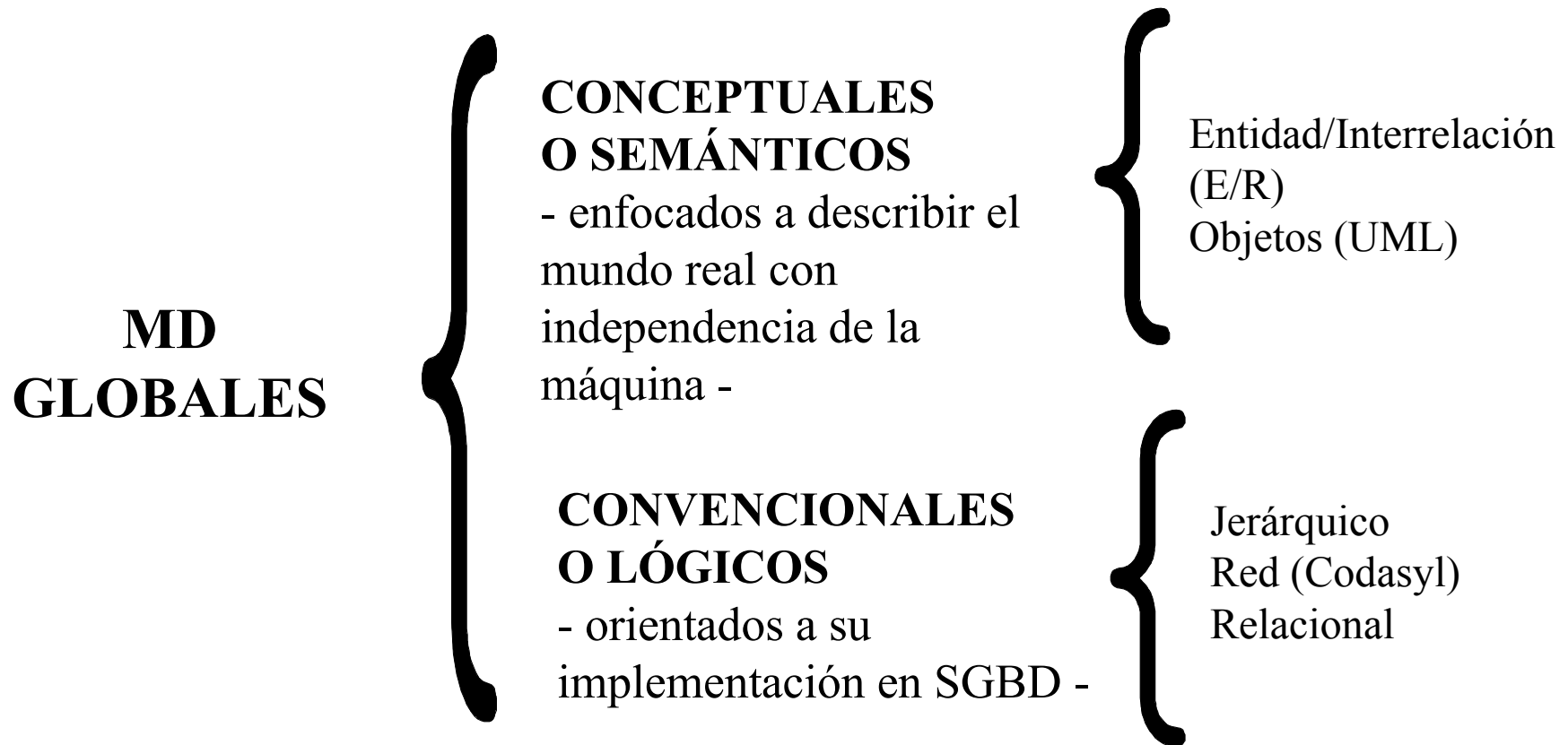


Relación entre modelo, esquema y ejemplar

- Según el nivel de abstracción que consideremos en la arquitectura ANSI, tendremos modelos **internos**, **globales** y **externos**:



- También se utiliza la expresión “modelos **lógicos**” para hacer referencia tanto a los globales como a los externos, ya que ambos describen aspectos lógicos de los datos -frente a los modelos internos que describen aspectos **físicos**-.



Diferencias entre modelos convencionales y conceptuales

CONVENCIONALES

- Implementados en SGBD comerciales
- Dependen del SGBD
- Más próximos al ordenador
- Poca capacidad semántica
- Más enfocados a la implementación
- Interfaz informático/sistema
- Nivel de “mediación” entre el nivel externo e interno

CONCEPTUALES

- No suelen estar implementados en SGBD
- Independientes del SGBD
- Mayor nivel de abstracción
- Mayor capacidad semántica
- Más enfocados al diseño de alto nivel (modelado conceptual)-
- Interfaz usuario/informático

- *El proceso de **abstracción** nos ayuda a modelar los datos al hacer que nos centremos en lo esencial, pasando por alto aspectos que no consideramos relevantes para nuestros objetivos en la representación del mundo real.*

Ejemplo:

El concepto de **ambulancia** como una abstracción en la que únicamente recogemos aquellas características (chasis, ruedas, sirena, etc.), comunes a todas las ambulancias y que la distinguen de otros vehículos, que son de interés para nuestros fines.



- Los MD ofrecen distintos mecanismos de abstracción a fin de facilitar la representación de los datos; siendo el esquema el resultado de aplicar un proceso de abstracción a un determinado mundo real. Los principales que aparecen en la bibliografía son:
 - **Clasificación**,
 - **Agregación**,
 - **Generalización** y
 - **Asociación**. $\leq$ (algunos autores piensan que es un tipo especial de agregación)
- Pueden combinarse entre sí ofreciendo interesantes mecanismos semánticos para estructurar los datos.
- Permiten establecer vinculaciones entre los elementos de un modelo.
- La clasificación establece una vinculación entre una categoría de objetos y cada objeto en particular (ejemplar) que pertenece a dicha categoría, mientras que en las otras tres (agregación, generalización y asociación) la relación se establece entre categorías de objetos y, por tanto, también entre los correspondientes ejemplares de dichas categorías.

Los mecanismos de abstracción los utilizamos - consciente o inconscientemente - de manera continua:

- El vehículo de matrícula CR-0978-Z es de la clase ambulancia.
- Está formada por cuatro ruedas, un chasis, un motor, ...
- Una ambulancia es un vehículo para recoger y transportar enfermos.
- Su propietario es la empresa CUASER; su conductor es Fernández, ...

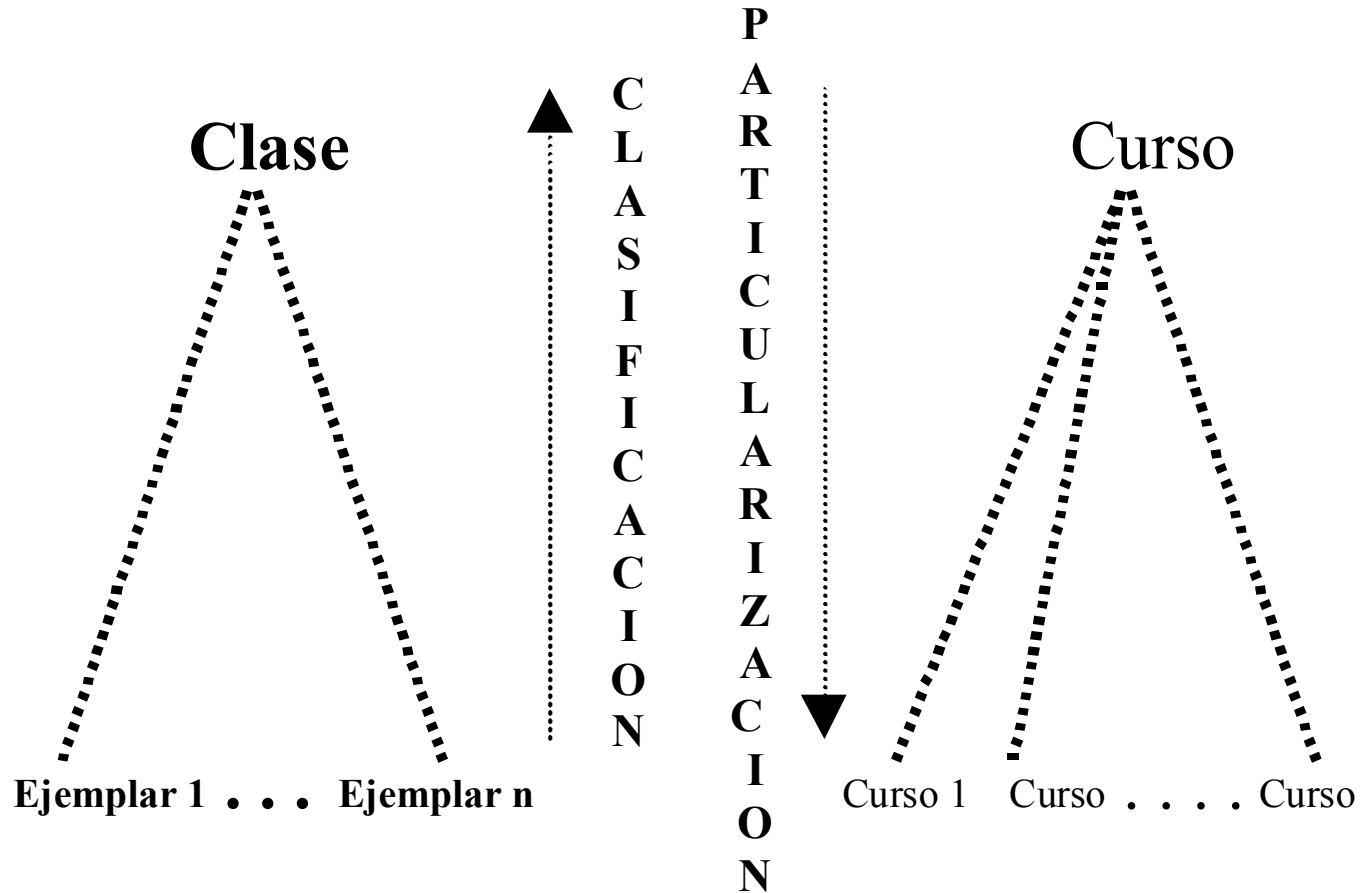


Los mecanismos de abstracción los utilizamos - consciente o inconscientemente - de manera continua:

- **Clasificación:**
 - El vehículo de matrícula CR-0978-Z es de la clase ambulancia.
- **Agregación:**
 - Está formada por cuatro ruedas, un chasis, un motor, ...
- **Generalización:**
 - Una ambulancia es un vehículo para recoger y transportar enfermos.
- **Asociación:**
 - Su propietario es la empresa CUASER; su conductor es Fernández, ...



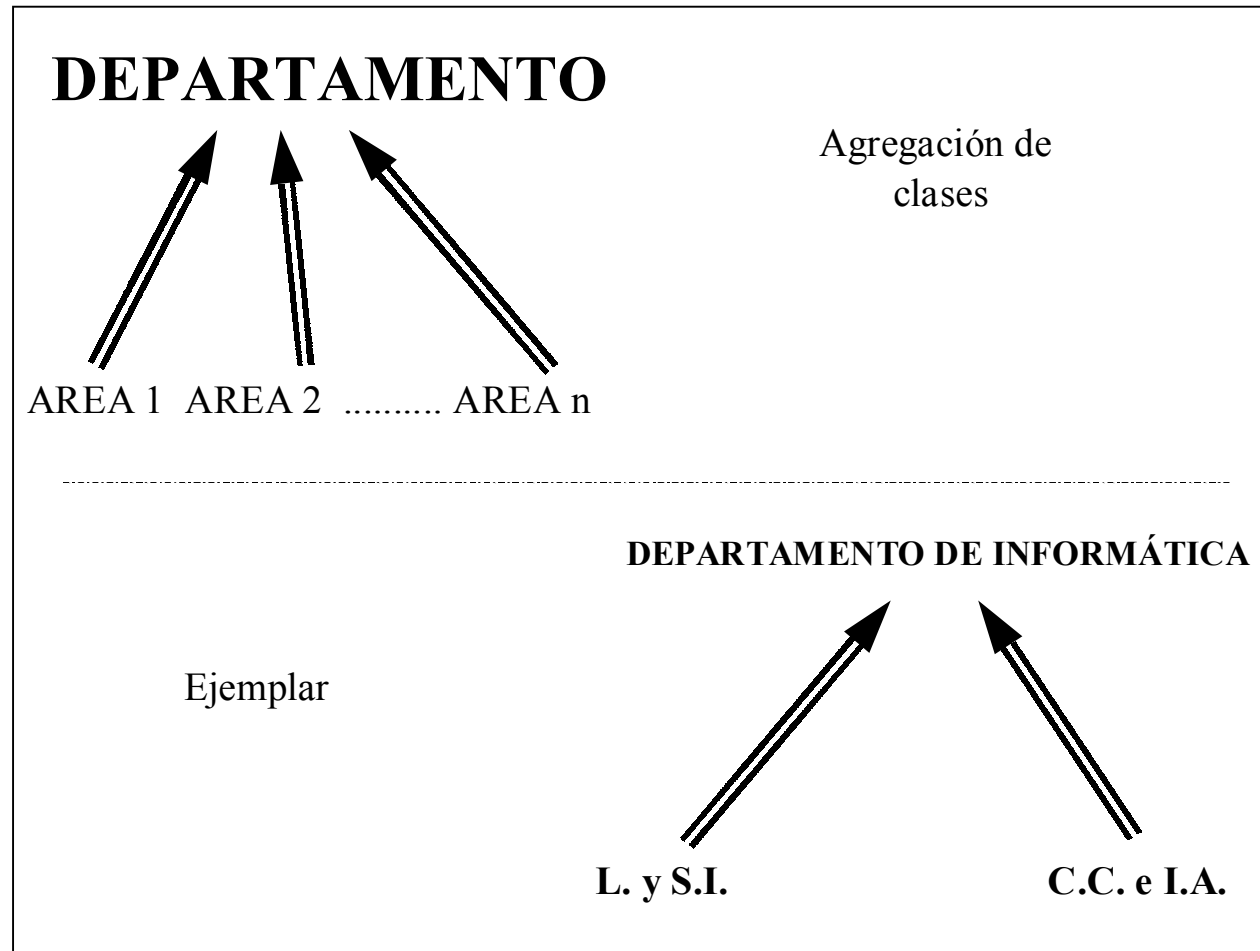
- La **Clasificación** es la acción de abstraer las características comunes a un conjunto de ejemplares para crear una categoría a la cual pertenecen dichos ejemplares.
- El mecanismo contrario se llama **Particularización**.
- BRODIE (1984) define la clasificación como:
 - *Una forma de abstracción en la que una colección de objetos se considera como una clase de objetos de más alto nivel.*
 - *Una clase de objetos es una caracterización precisa de todas las propiedades compartidas por todos los objetos en la colección.*
 - *Un objeto es un ejemplar de una clase de objetos si tiene las propiedades definidas en la clase.*
- Ejemplo:
 - Clasificamos como **Vehículos** a las máquinas, animales o cosas, con medios de propulsión propios, que sirven para desplazar seres u objetos desde una posición a otra.
 - Ambulancia => SI es un vehículo
 - Grúa => NO es un vehículo (incumple la propiedad de autopropulsión).



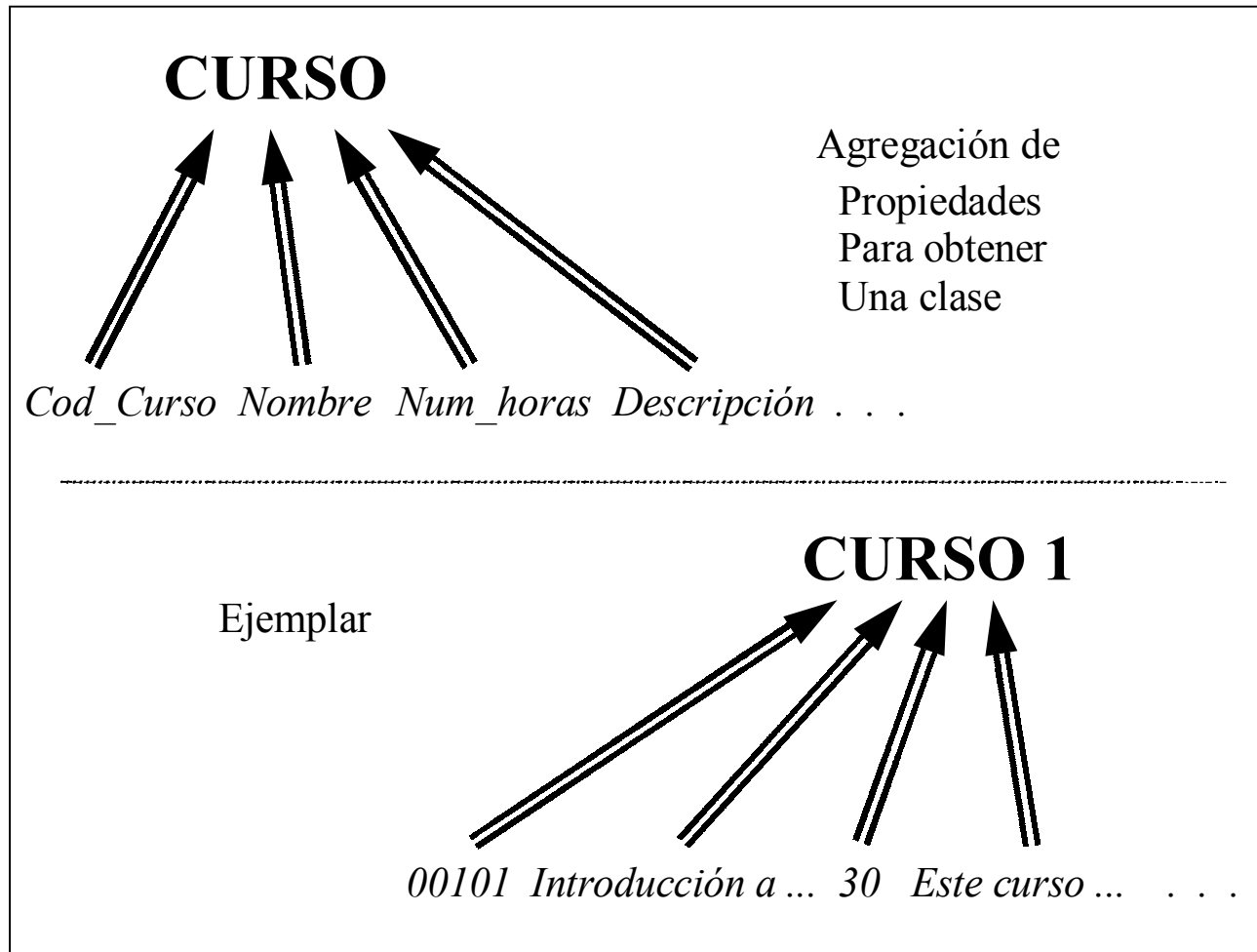
Representación de la clasificación/particularización

- La clasificación se corresponde con el concepto de **pertenencia** a un conjunto (*es miembro de*):
 - entre el elemento clase y los elementos miembros se establece una relación ES_MIEMBRO_DE.
- Los ejemplares de una clase tienen características similares, por medio de las cuales describimos la correspondiente clase; estas características toman valores concretos para cada uno de los ejemplares que pertenecen a la clase.
- Los mismos objetos admiten clasificaciones distintas. Por ejemplo, podemos clasificar las asignaturas de varias maneras:
 - obligatorias / optativas,
 - anuales / semestrales,
 - de primer curso, segundo curso, etc,
 - teóricas / aplicadas, etc.
- Todos los modelos de datos de las bases de datos admiten la abstracción de clasificación.

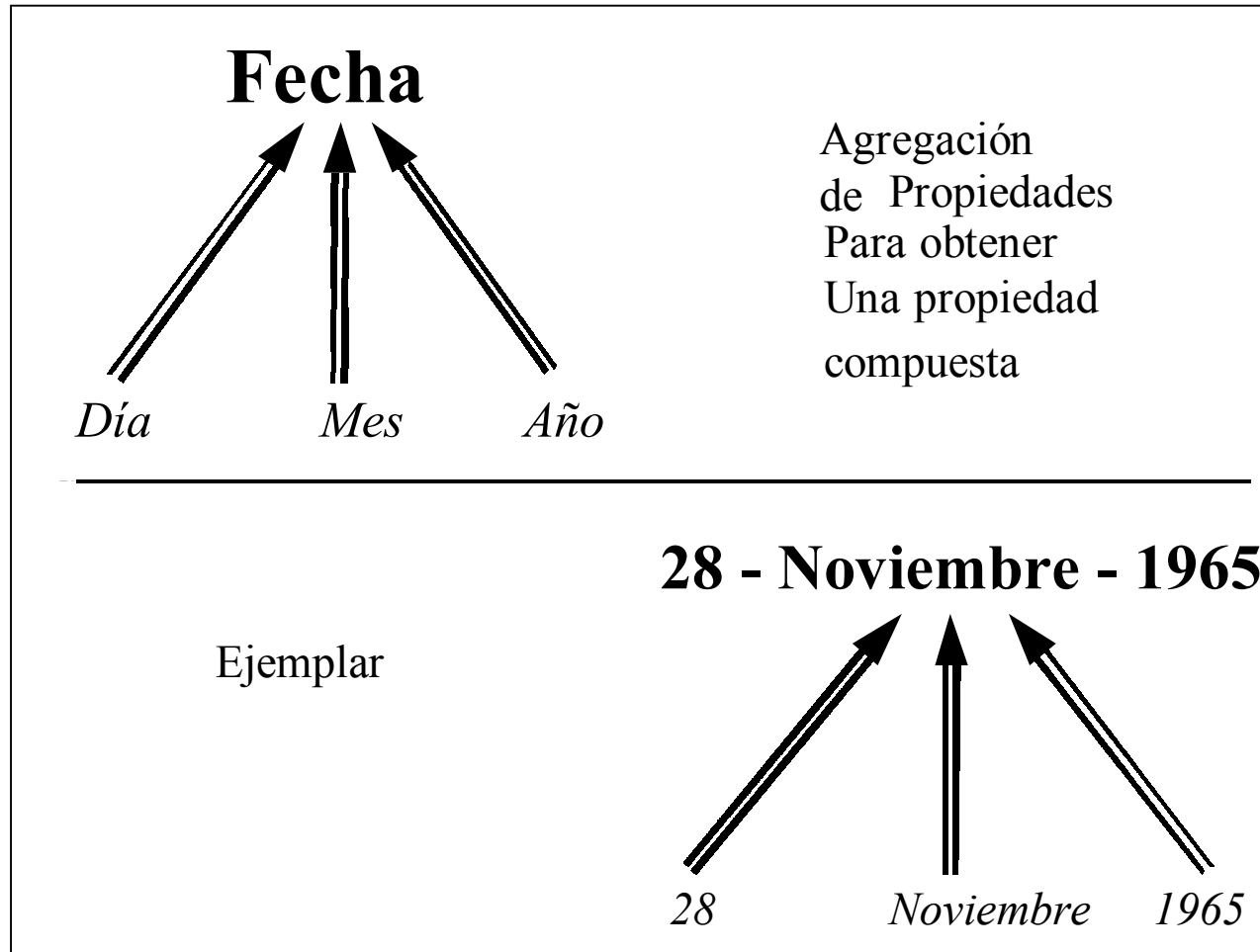
- La abstracción de **Agregación** consiste en construir un nuevo elemento del modelo como **compuesto** de otros elementos (**componentes**):
 - Se establece una relación ES_PARTE_DE entre los elementos componentes y el elemento compuesto.
- El mecanismo contrario se llama **Desagregación**.
- Se pueden considerar tres tipos distintos de agregación:
 - Agregación de clases para obtener una clase compuesta,
(sólo la incluyen los MD semánticos: ER, UML)
 - Agregación de propiedades para obtener una clase, y
(admitida explícita o implícitamente por todos los MD)
 - Agregación de propiedades para obtener una propiedad compuesta.
(admitida por algunos MD: Codasyl Sí, Relacional No)



Agregación de clases para obtener una clase compuesta

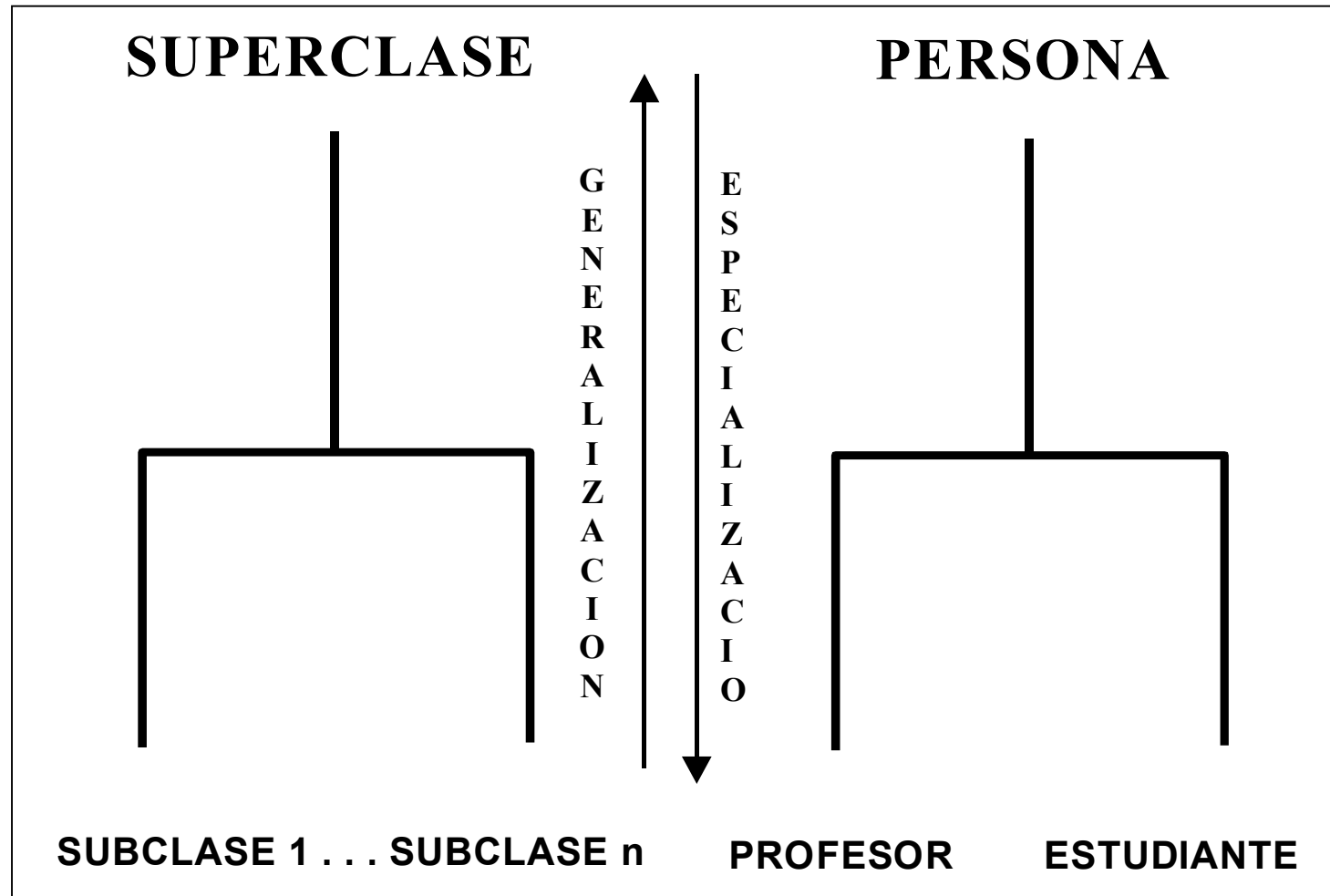


Agregación de propiedades para obtener una clase



Agregación de propiedades para obtener una propiedad compuesta

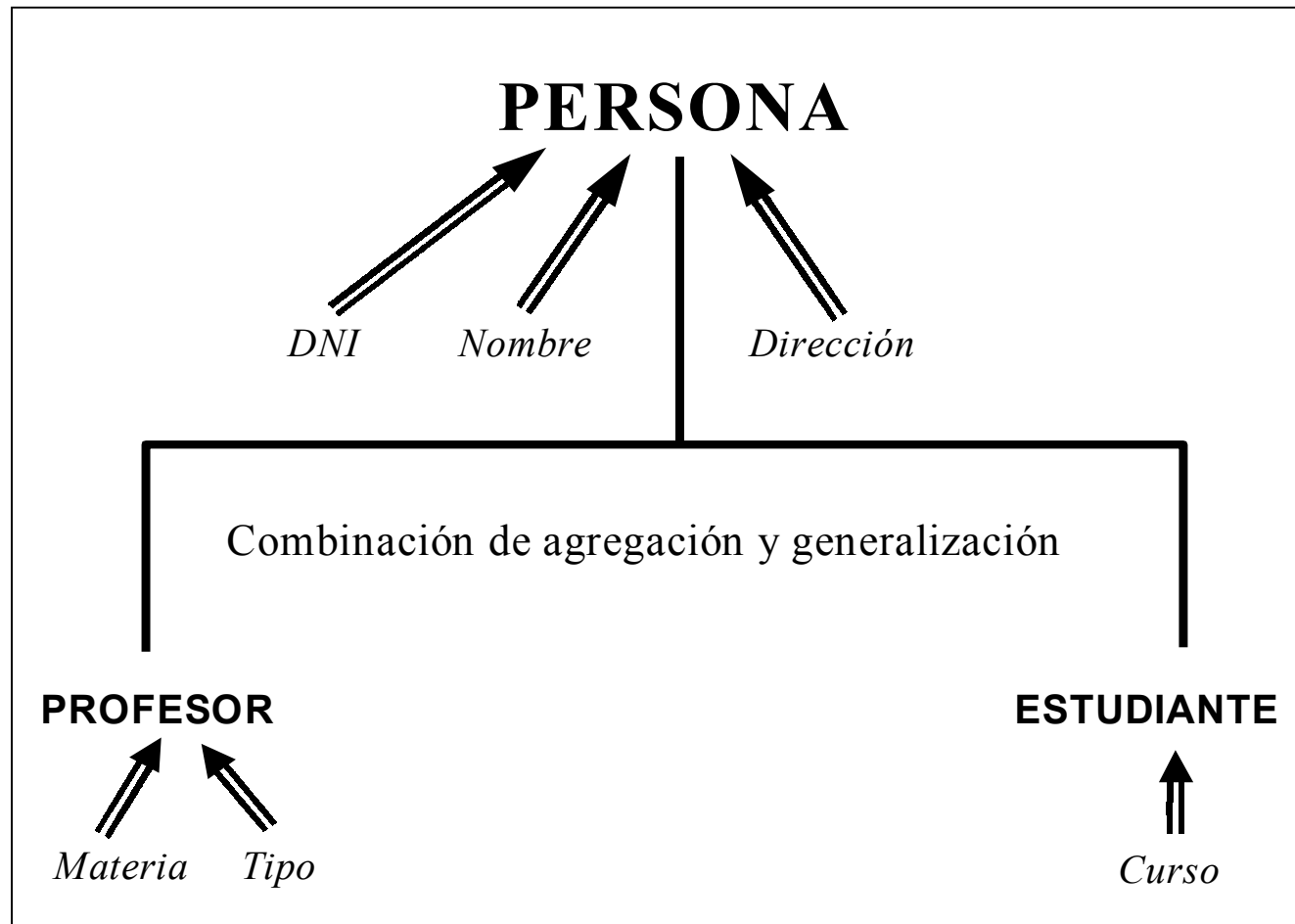
- La **Generalización** es la acción de abstraer las características comunes a varias clases (**subclases**) para constituir una clase más general (**superclase**) que las comprenda:
 - El conjunto de ejemplares de una subclase “es un” subconjunto de los ejemplares de la correspondiente superclase.
 - Entre los elementos subclase y el elemento superclase se establece una relación del tipo ES_UN.
 - Ejemplo: La superclase PERSONA es una generalización de las subclases PROFESOR y ESTUDIANTE.
- Cada generalización es un árbol (jerarquía) de un solo nivel, donde la raíz es la superclase y las hojas son las subclases.
- El mecanismo inverso de la generalización es la **Especialización**.
- Es un proceso parecido a la clasificación, pero mientras en ésta se pasa de los ejemplares a la clase (o viceversa), en la primera se pasa de una clase a otra clase.
- Todo ejemplar de una subclase es también ejemplar de la superclase y, además de poseer las características específicas de la subclase, *hereda* todas las correspondientes a la superclase.
- Aunque esta abstracción es muy intuitiva y muy útil, no se contempla en bastantes modelos de datos (p.e. Relacional).



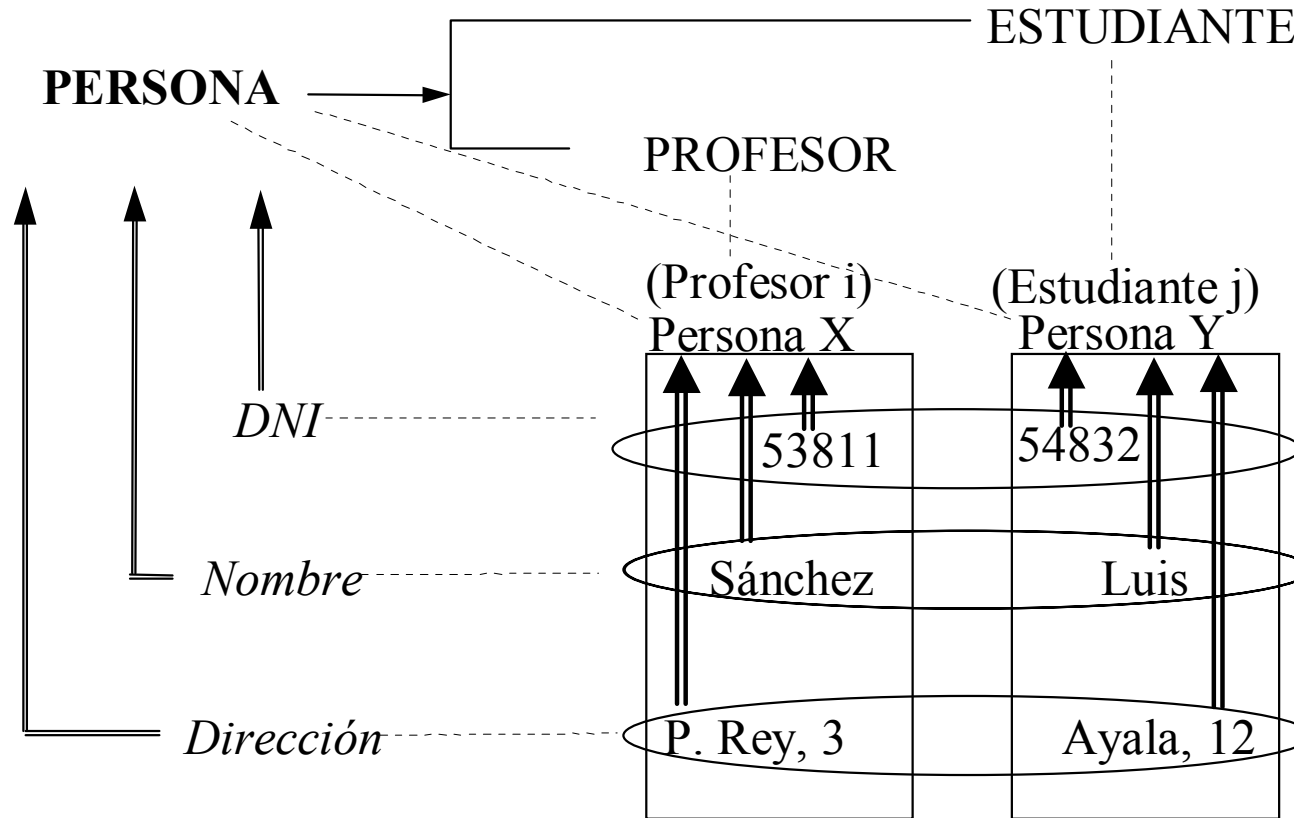
Representación de una generalización/especialización

- La **Asociación** es una abstracción que se utiliza para relacionar dos o más clases (y, por tanto sus ejemplares), creándose un elemento de un tipo distinto.
- En algunos MD no aparece esta abstracción como tal, no existiendo ningún concepto especial para representarla (p.e. Relacional).
- El mecanismo inverso es la **Disociación**.
- Aunque algunos autores consideran que la asociación es un tipo especial de agregación, en de Miguel, Piattini y Marcos (1999) se determinan las siguientes diferencias:
 - Cuando se asocian dos o más categorías, el nuevo elemento que aparece tiene determinadas características que lo distinguen de las categorías normales, por lo que, en general, los modelos de datos crean un nuevo concepto para representarlo.
 - El nuevo elemento no *está compuesto*, como en el caso de la agregación, por los elementos que asocia.
 - En la agregación puede existir herencia, y no así en la asociación.
- Ejemplo:

PROFESOR — **Imparte** — CURSO



En el proceso de modelado de una determinada realidad es preciso, a menudo, combinar distintas abstracciones formando lo que se conoce como una jerarquía de abstracciones



Ejemplo de abstracciones de clasificación, agregación y generalización

- Aunque existen muchos MD es posible abstraer una serie de características comunes a todos ellos, definiendo así el concepto de modelo de datos en general.
- Podemos ya definir de forma más precisa el concepto de MD como:
 - *“un conjunto de conceptos, reglas y convenciones bien definidos 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”*.
- Los MD facilitan la creación de categorías mediante la aplicación de los tipos de abstracción anteriormente considerados. Esto lleva a diferenciar dos tipos de modelos (de forma similar a como ocurre con los lenguajes de programación):
 - Fuertemente tipados, (son los utilizados en BD).
 - Débilmente tipados.

Propiedades de un Modelo de Datos:

a) ESTÁTICAS

a.1) Elementos permitidos

- Objetos
- Asociaciones
- Características de los objetos
- Dominios

a.2) Elementos no permitidos e restricciones

- Inherentes
- De integridad o semánticas

b) DINÁMICAS (conjunto de operadores). Cada operador tiene dos componentes:

- Localización
- Acción

- Un MD ha de proporcionar facilidades para recoger los aspectos estáticos y dinámicos de la realidad, por lo que se define formalmente como el par:

$$\text{MD} = \langle \mathbf{G}, \mathbf{O} \rangle$$

- donde **G** es el conjunto de reglas de generación que permiten representar la componente **estática**, es decir, describir las **estructuras** de nuestro universo del discurso, y
- **O** es el conjunto de **operaciones** autorizadas sobre la correspondiente estructura, operaciones que permiten representar la componente **dinámica**.
- La componente estática de un determinado MD expresado con una sintaxis determinada es el *Lenguaje de Definición de Datos (LDD)*, y la componente dinámica el *Lenguaje de Manipulación de Datos (LMD)*; ambos constituyen el *Lenguaje de Datos (LD)*.
- Los SGBD suelen tener además un *Lenguaje de Consulta* (en inglés Query Language -QL-) y un *Lenguaje de Control* (en inglés Control Language).

- Los **elementos permitidos** no son los mismos para todos los MD (varían especialmente en terminología), pero en general son:
 - **Objetos** (entidades, relaciones, registros, etc.)
 - **Asociaciones** entre objetos (interrelaciones, “set”, etc.)
 - **Propiedades** o características de los objetos o asociaciones (atributos, campos, elementos de datos, etc.)
 - **Dominios**, que son conjuntos nominados de valores homogéneos sobre los que se definen las propiedades.
- A estos elementos permitidos se les podrán aplicar aquellas abstracciones reconocidas por el modelo.
- La representación de estos elementos depende de cada modelo de datos, pudiendo hacerse en forma de **grafos** (E/R, UML) o de **tablas** (Relacional).

- Los **elementos no permitidos** se conocen como **restricciones**.
- Existen dos tipos:
 - Restricciones **inherentes** (del modelo): vienen impuestas por la misma naturaleza del modelo de datos, el cual no admite ciertas estructuras.
 - Restricciones de **integridad** o **semánticas** (de usuario): permiten captar la semántica del universo del discurso que se quiere modelar y verificar la corrección de los datos almacenados en la BD. Según los instrumentos que proporcione el modelo de datos para definir y gestionar las restricciones, éstas pueden ser:
 - *Propias al MD*: su definición le corresponde al diseñador, pero su gestión es responsabilidad del modelo de datos, el cual las reconoce y recoge en el esquema.
 - *Ajenas al MD*: son, por completo, responsabilidad del diseñador, ya que el modelo de datos no las reconoce ni proporciona instrumentos para manejarlas.

- Podemos definir la componente estática del modelo de datos como:

$$G = \langle G_e, G_r \rangle$$

- donde G_e es el conjunto de reglas de generación de estructuras (objetos del modelo y restricciones inherentes) y G_r es el conjunto de restricciones de usuario.
- La aplicación de la componente estática de un MD a un determinado Universo del Discurso (UD) nos da como resultado un esquema (E):

$$G[UD] = E$$

- E es la estructura de datos que describe, en el correspondiente modelo MD, las categorías que han resultado de las abstracciones aplicadas al mundo real (UD) que se trata de modelar.

- La componente **dinámica** consta de un conjunto de **operadores** que se definen sobre la estructura del correspondiente MD, ya que no todas las estructuras admiten el mismo tipo de operaciones.
- La aplicación de un operador a un ejemplar de un esquema transforma éste en otro ejemplar:

$$O [BDi] = BDj$$

- Tanto BDi como BDj deben ser ejemplares válidos de la BD, es decir, los valores de ambos deben pertenecer a alguna de las categorías definidas en el esquema y cumplir las restricciones de integridad (también deben cumplir, en caso de que existan, las posibles restricciones asociadas al cambio de estado).

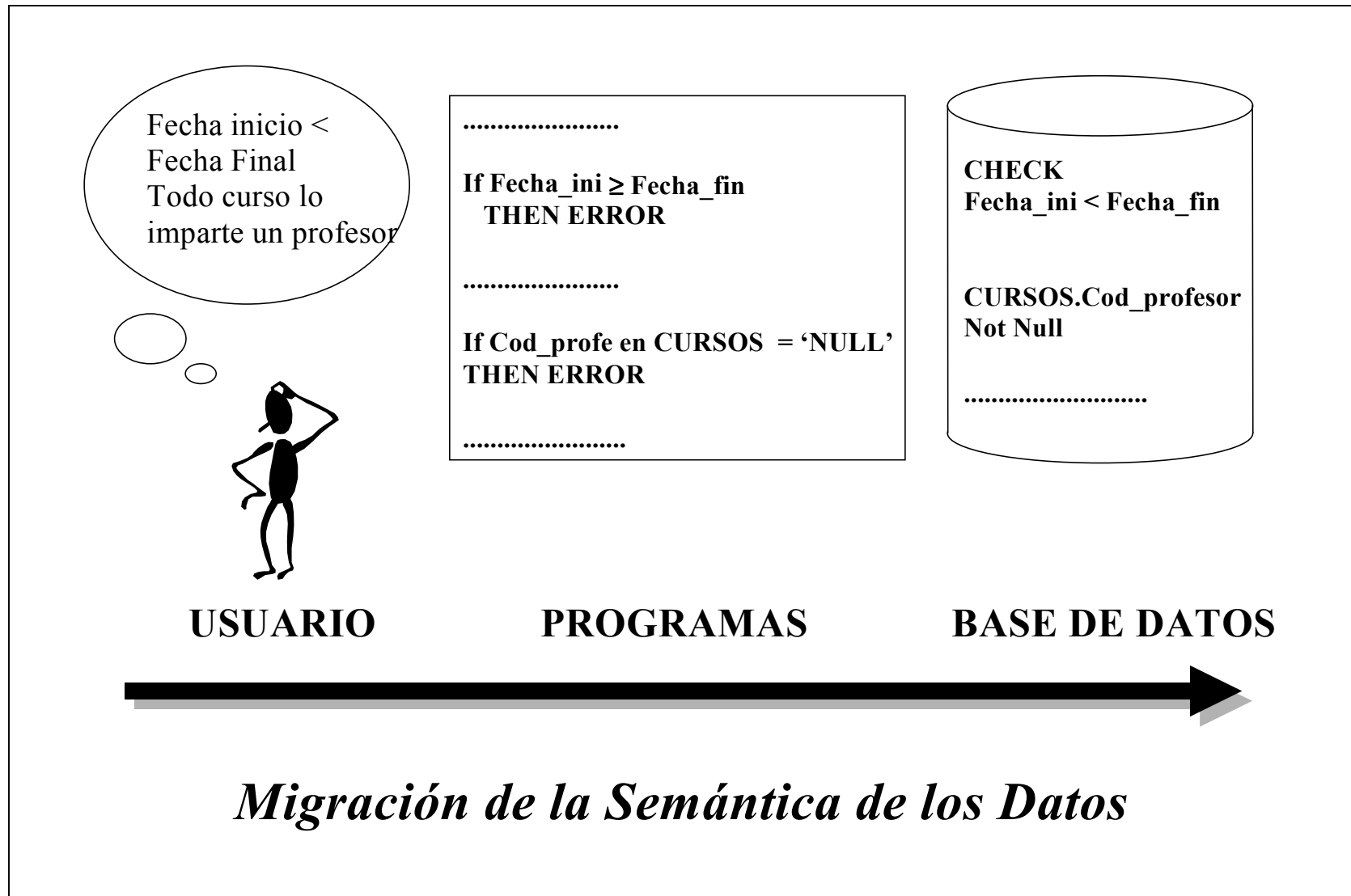
- En un plano conceptual, sin seguir una sintaxis concreta, podemos expresar una sentencia del LMD de la siguiente forma:

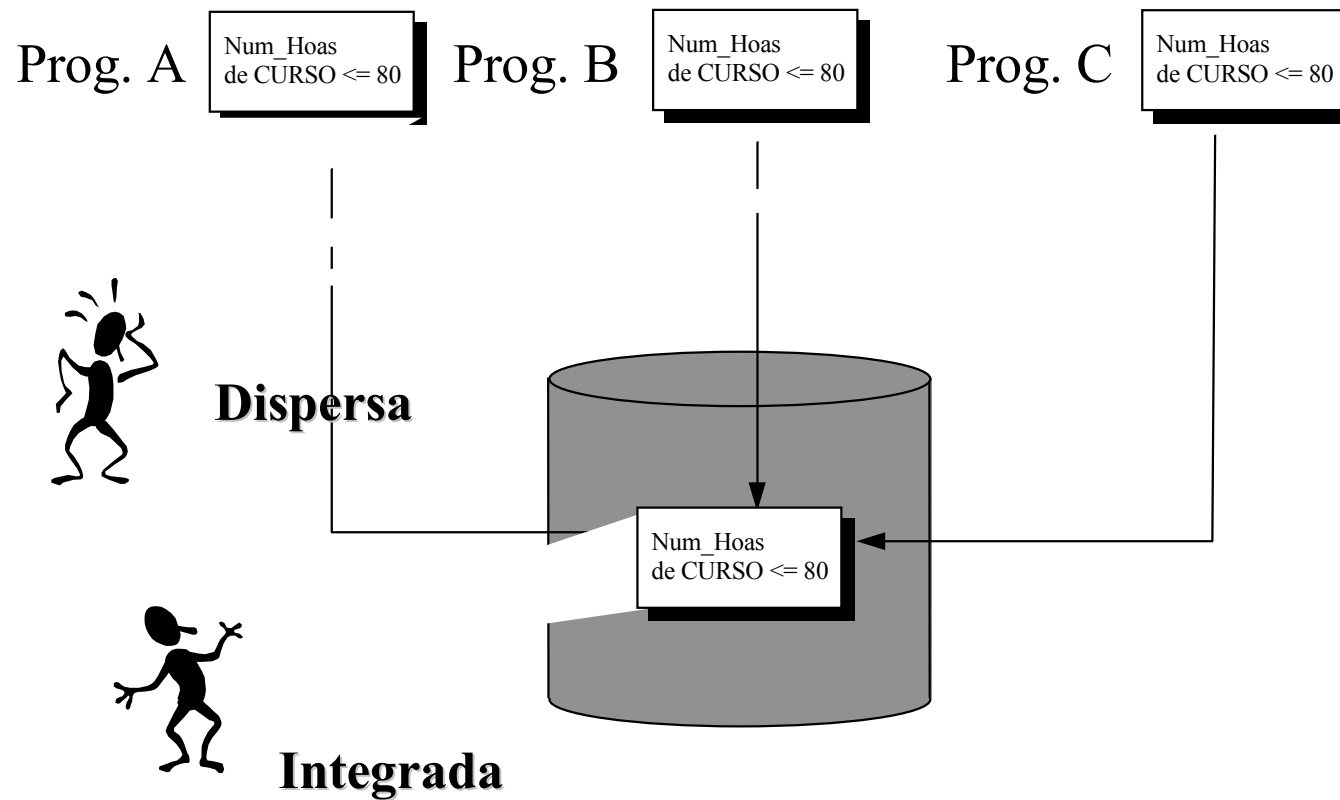
LOCALIZACIÓN <condición>

ACCIÓN <objetivo>

```
SELECT DNI, Nombre  
FROM Alumnos  
WHERE Curso=3
```

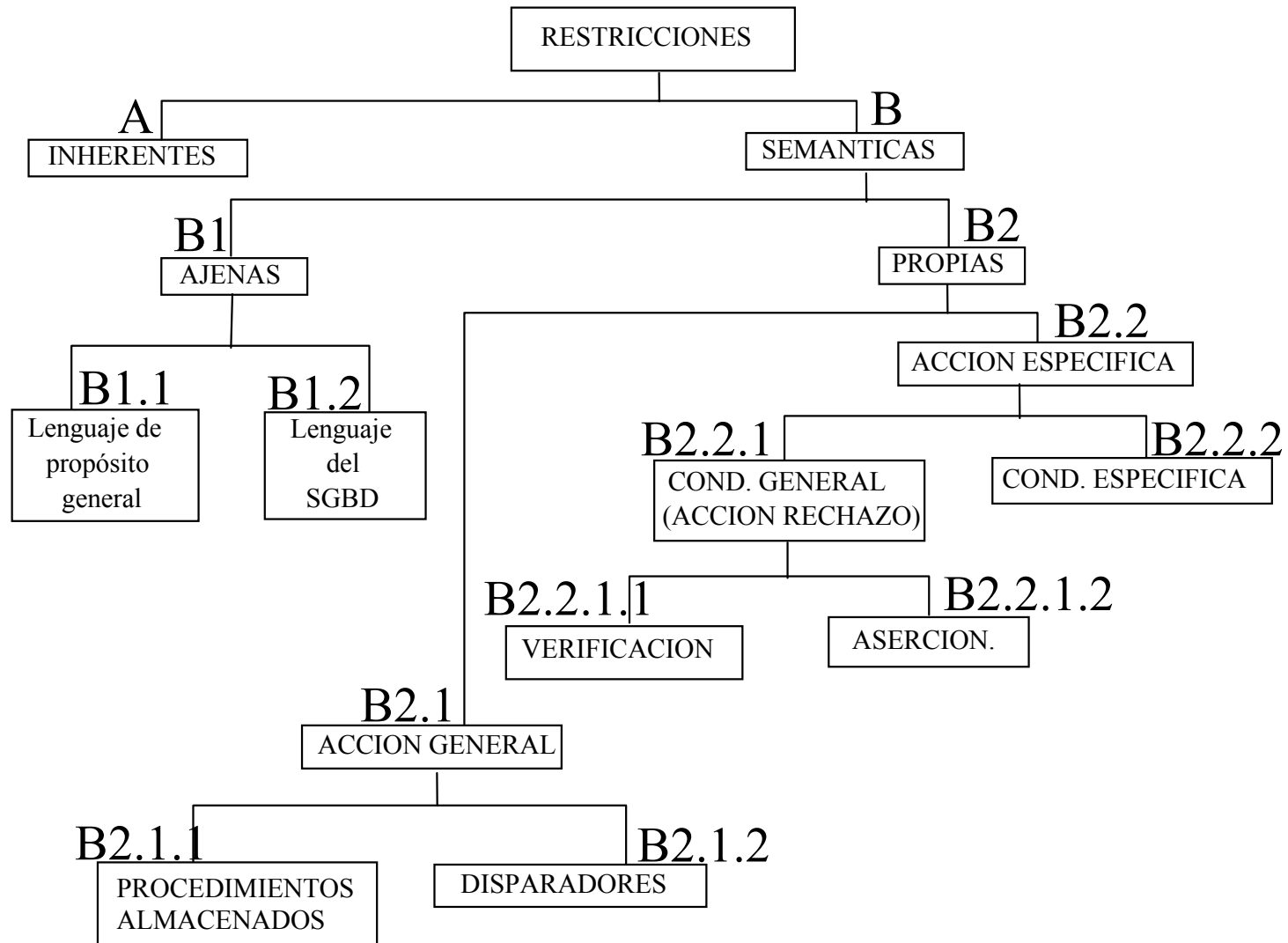
- donde
 - la **Localización** o “enfoque” consiste en localizar un ejemplar de un objeto indicando un camino (lenguaje navegacional), o un conjunto de ejemplares especificando una condición (lenguaje de especificación); y
 - la **Acción** se realiza sobre el(los) ejemplar(es) previamente localizado(s). Puede consistir en una recuperación o en una actualización (inserción, borrado o modificación).
- y
 - <condición> representa una expresión lógica que deben cumplir los objetos que se desea localizar o señala el camino que permite llegar a esos objetos; y
 - <objetivo> indica los objetos (o las propiedades de éstos) sobre los que se aplica la acción.





Semántica de los datos “dispersa” vs “integrada”

- En una restricción de integridad es posible distinguir los siguientes componentes:
 - La **operación** de actualización (inserción, borrado o modificación) cuya ejecución ha de dar lugar a la comprobación del cumplimiento de la restricción.
 - La **condición** que debe cumplirse, la cual es en general una proposición lógica, definida sobre uno o varios elementos del esquema, que puede tomar uno de los valores de verdad (cierto o falso).
 - La **acción** que debe llevarse a cabo dependiendo del resultado de evaluar la condición.
- Las restricciones de integridad se pueden considerar, en cierto modo, como reglas ECA (Evento, Condición, Acción):
 - al ocurrir un evento (en este caso una actualización), se comprueba una condición y dependiendo de su resultado se pone en marcha una acción (rechazar la operación, informar al usuario, corregir el error, etc.).
- Además de estos elementos, también pueden tener un *nombre*, por medio del cual es posible identificarlas, y también puede indicarse el *momento* en el que ha de evaluarse la condición.



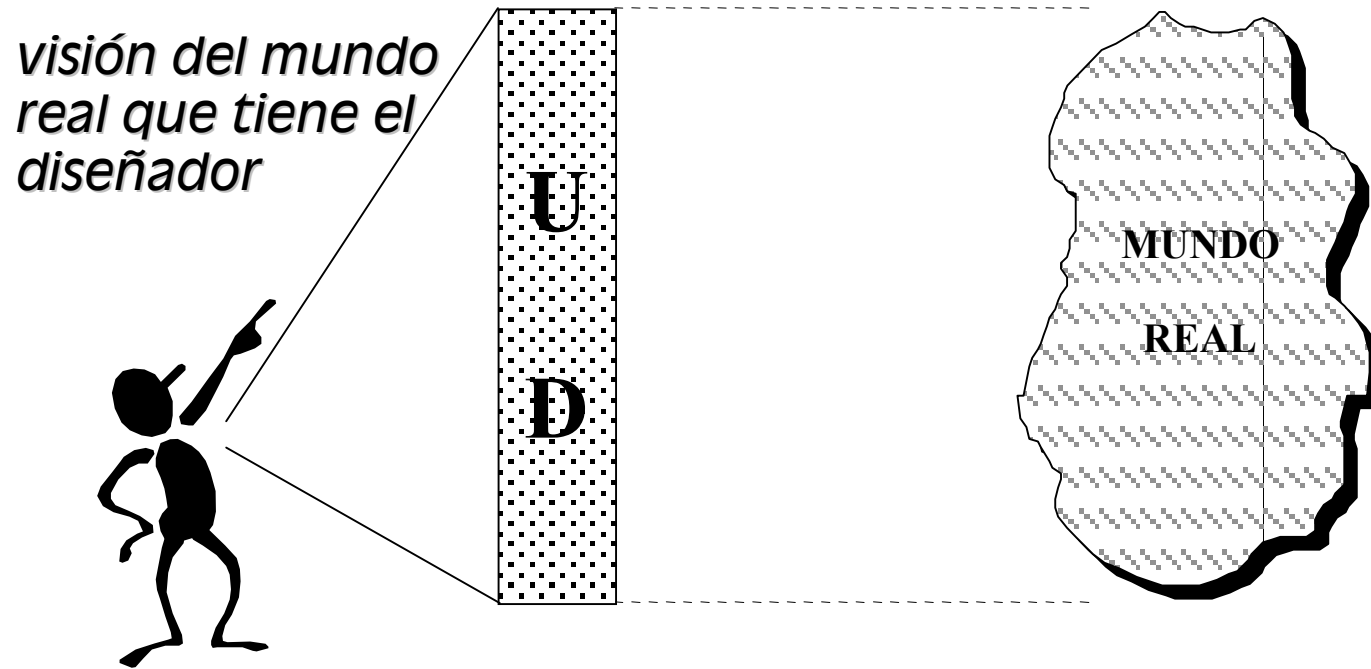
- Las ***restricciones propias*** se especifican al definir el esquema mediante las facilidades que proporciona la función de definición de datos, almacenándose en la base de datos (no en los programas), por lo que no pueden ser violadas por ninguna aplicación, es decir, cualquier actualización está obligada a respetarlas.
- Dependiendo de que sea o no preciso definir la acción tendremos restricciones propias de dos tipos:
 - De ***acción general***: es preciso programar un procedimiento (en cualquier lenguaje) que determine la acción que hay que llevar a cabo. Se subdividen en:
 - ***Procedimientos almacenados***: se definen totalmente de forma procedimental (tanto la acción como la condición).
 - ***Restricciones de disparo***: se definen mediante **disparadores** (triggers). En ellas se formula una condición de forma declarativa, mediante una proposición lógica; el *cumplimiento* de la misma "dispara" una acción especificada de forma procedimental.
 - De ***acción específica***: la acción (en general rechazo, aunque puede ser otra, bien predeterminada bien elegida mediante opciones) está implícita en la misma restricción.

- Dentro de las *restricciones de acción específica* es preciso distinguir especifican las siguientes clases:
 - De *condición general*: la condición se define mediante una proposición lógica. La operación será una actualización. No se declara la acción porque este tipo de restricción lleva siempre asociado el rechazo de la operación cuando no se cumple la condición, es decir, el sistema evalúa la condición y si el resultado es cierto se actualiza y si no es cierto no se lleva a cabo la operación. En SQL 92 se incluyen dos tipos de restricciones que son de condición general:
 - De *verificación*: La expresión lógica mediante la cual se formula la condición está definida sobre uno o varios atributos de un mismo elemento. Por ejemplo, una cláusula "CHECK" dentro de un CREATE TABLE.
 - De *aserción*: son análogas a las anteriores pero pueden estar referidas a más de un elemento del esquema ya que tienen existencia por sí mismas (por tanto tienen un nombre). Ejemplo: CREATE ASSERTION ..
 - De *condición específica*: también llamadas de "caso especial" ó "implícitas". Se refiere a las diversas opciones que facilitan los distintos MD cuando se definen los elementos de su esquema y que en realidad son restricciones. Por ejemplo, en el modelo Relacional: PRIMARY KEY, FOREIGN KEY, NOT NULL ...

Otros criterios de clasificación:

- A) Según la dimensión temporal:
 - De **Estado** o *Estáticas*: las restricciones se aplican a un determinado estado de una BD y no hay necesidad de conocer los estados anteriores para saber si se cumple o no la condición. Ejemplo: $\text{edad} \geq 0$.
 - De **Transición** o *Dinámicas*: la restricción hay que aplicarla a la transición entre dos estados. Ejemplo: el salario de un empleado no puede disminuir.
- B) Según el ámbito o alcance:
 - Afectan a un **único ejemplar** de un tipo dado: $\text{edad} > 14$.
 - Afectan a **varios ejemplares** de un tipo dado. A su vez, pueden afectar sólo a algunos o a todos los ejemplares de un cierto tipo.
- C) Según los criterios de comparación:
 - De **valor**: en la condición se comparan los valores que pueden tomar las propiedades; o
 - **Estructurales**: imponen limitaciones a la estructura de los elementos del modelo. Por ejemplo, que un atributo no puede tomar más que un valor.

- Se conoce como proceso de diseño de una BD al conjunto de tareas necesarias para pasar de una determinada realidad (Universo del Discurso) a la BD que la representa. Los MD juegan un importante papel en el proceso de diseño de una BD al ofrecernos facilidades de abstracción que nos ayudan a representar la realidad.
- Los **objetivos** que persigue todo MD son de dos tipos:
 - a) De **Formalización**: el MD permite definir formalmente las estructuras permitidas y las restricciones; también establece la base para la definición de un lenguaje de datos.
 - b) De **Diseño**: el MD es un elemento fundamental en el desarrollo de una metodología de diseño de BD, en el cual se basan los otros componentes de la metodología (lenguajes, documentación y otras herramientas).



Universo del Discurso y Mundo Real

6. Los Modelos de Datos en el proceso de diseño de una Base de Datos.

