

Análisis Orientado a Objetos en UML

TEMA 2

Tema 2. Análisis Orientado a Objetos en UML

2.1. Actividades del análisis de requisitos del software

2.2. Principios fundamentales del análisis de requisitos del software

2.3. Especificación de requisitos del software

2.4. Análisis orientado a objetos en UML

2.5 Modelo de casos de uso en UML

2.6. Modelo conceptual de datos en UML

**Contenido de este
fichero**

2.7. Modelo de comportamiento del sistema en UML

Bibliografía

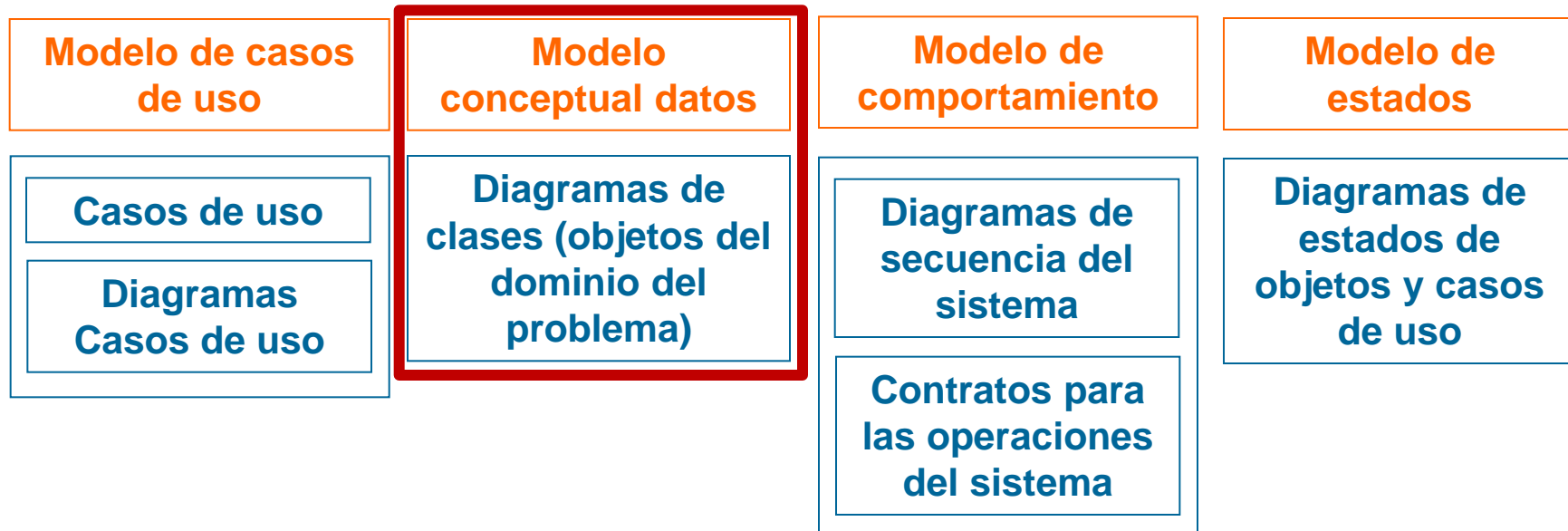
El alumno debe ser capaz de:

- Describir el objetivo del **modelo conceptual de datos**.
- Definir qué es un **objeto** y una **clase de objetos**.
- Definir qué es un **atributo** de una clase de objetos.
- Definir qué es una **asociación** entre clases de objetos.
- Definir los siguientes conceptos relacionados con la asociación: **rol, multiplicidad, enlace y atributo de enlace**.
- Diferenciar **rol como extremo de una asociación** y **rol como concepto**.
- Definir los conceptos **clase de asociación, clasificación y asociación calificada**.

El alumno debe ser capaz de:

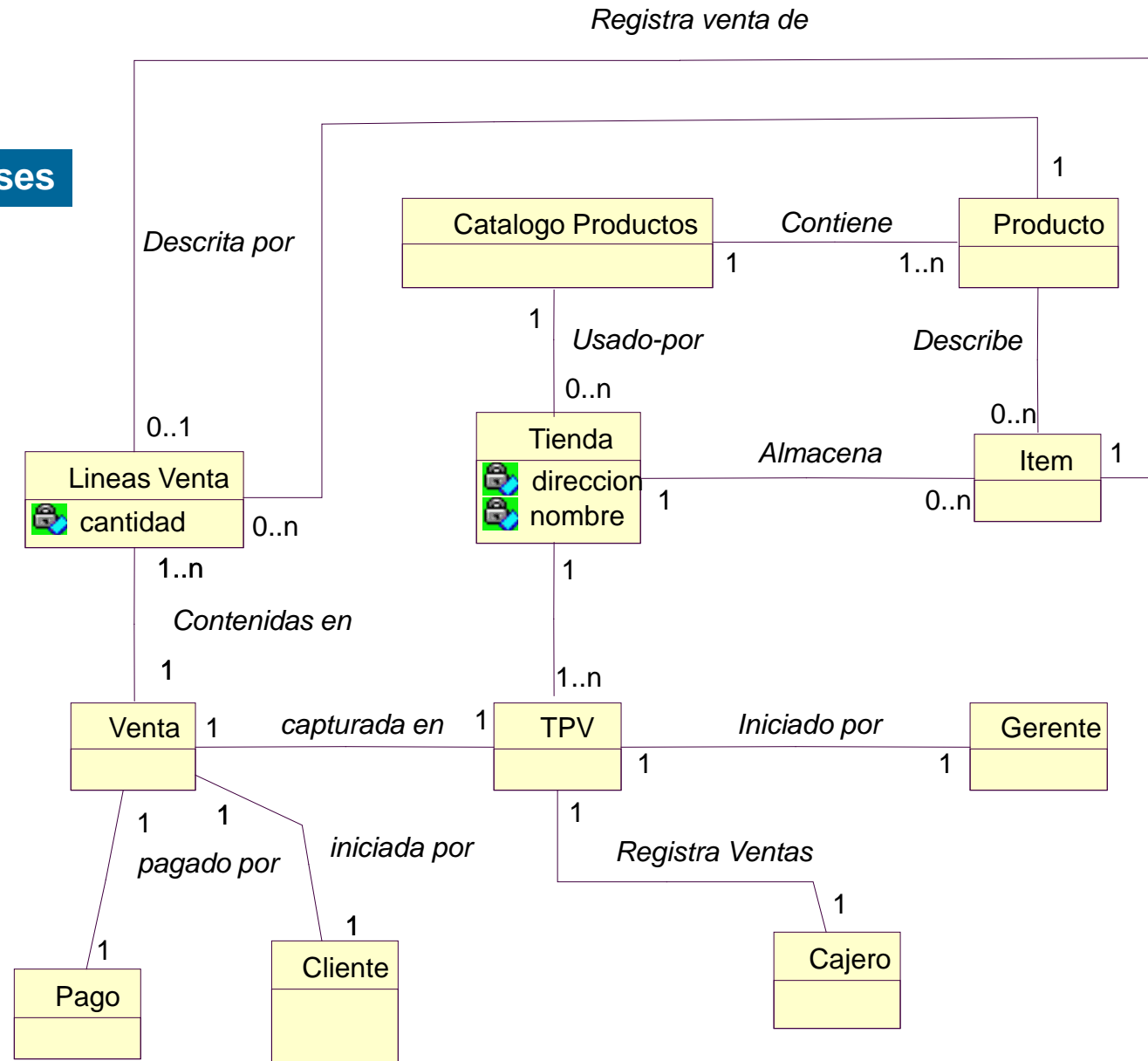
- Definir y diferenciar los conceptos **agregación y composición**.
- Definir los conceptos **generalización/especialización**.
- Enumerar y describir las **restricciones semánticas de la generalización**.
- Explicar las **reglas Es-Un y la regla del 100%**.
- Definir los conceptos **herencia múltiple y clasificación múltiple**.
- Definir qué es información derivada (**atributo derivado y asociación derivada**).
- Definir qué es una **restricción sobre atributo, una restricción sobre asociación (XOR y Subset) y una restricción textual**.
- Realizar modelos conceptuales de datos de sistemas propuestos.

Modelo conceptual de datos



Modelo conceptual de datos

Diagrama de clases



Modelo conceptual de datos

- Modela los **requisitos de datos** del software (datos y relaciones estáticas entre ellos).
- **Representa los conceptos significativos** en el dominio del problema (**objetos**).

- Normalmente contienen:
 - **Clases** de objetos.
 - **Relaciones** entre clases de objetos.
 - **Atributos** de las clases de objetos.
 - **Restricciones de integridad**

Objeto

- Entidad que existe en el mundo real.
- Tienen **identidad propia** y son **distinguibles** entre ellos.

Ejemplos

- El avión con matrícula 327.
- El avión con matrícula 999.
- La factura 3443.
- Una manzana.
- Una mesa.
- Un ordenador con nº inventario C-1122.
- Etc.

Clase

Describe un **conjunto de objetos** con:

- Las mismas **propiedades**.
- **Patrones de comportamiento** común.
- Idéntica **relación** con otros **objetos**.
- **Semántica** común.

- Avión con matrícula 327
- Avión con matrícula 999

Eliminar diferencias entre objetos para poder observar aspectos comunes (abstracción)

Clase avión

Avión

Clase

Ejemplo: Venta de productos

“Un terminal de punto de venta (TPV) es un sistema que se usa para registrar las ventas de productos a clientes y para gestionar los pagos. Se usa principalmente en supermercados y grandes superficies. Incluye componentes hardware (ordenador y escáner del código de barras) y software para ejecutar el sistema”.



¿Sabrías identificar las clases?

Clase

Ejemplo: Venta de productos

“Un terminal de punto de venta (TPV) es un sistema que se usa para registrar las ventas de productos a clientes y para gestionar los pagos. Se usa principalmente en supermercados y grandes superficies. Incluye componentes hardware (ordenador y escáner del código de barras) y software para ejecutar el sistema”.

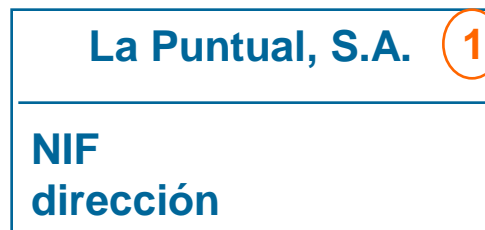
Clases

- TPV
- Venta (Línea de Venta)
- Producto
- Cliente
- Pago
- Supermercado

Multiplicidad de clase

- Establece el rango de **posibles cardinalidades** de las **instancias** de una clase (número posible de objetos de la clase).
- Por defecto, es **indefinida**.
- En algunos casos es útil establecer una **multiplicidad finita**, especialmente si la clase sólo puede tener una instancia.

Ejemplo



Atributo

- **Propiedad** compartida por los **objetos de una clase**
- Puede tomar **valores nulos** (ej. descripción)
- Puede ser **multivaluados** (ej. telfs)

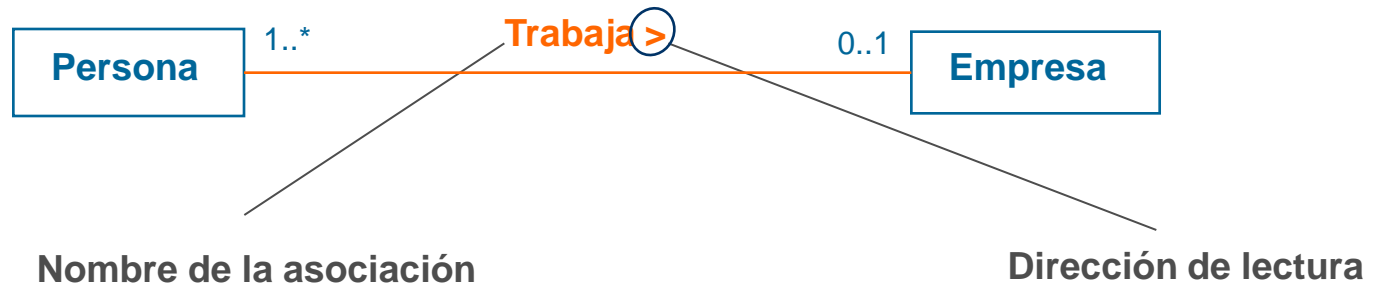
Ejemplo

| | | |
|-----------------------|---|---|
| TPV | Supermercado | Venta |
| num-pv: integer | dirección: string nombre: string | fecha: date hora: time |
| Línea de venta | Cliente | Producto |
| cantidad: integer | nombre: string telfs [1..*]: integer tipcli: string | upc: integer descripción [0..1]: string precio: integer |
| Pago | | |
| importe: integer | | |

Asociación

- Es la representación de **relaciones** entre **dos o más objetos** de diferentes clases.
- La **navegación** de una asociación por defecto es **bidireccional**.
- Al **nombrar una asociación** hay que elegir una **determinada dirección**.

Ejemplo

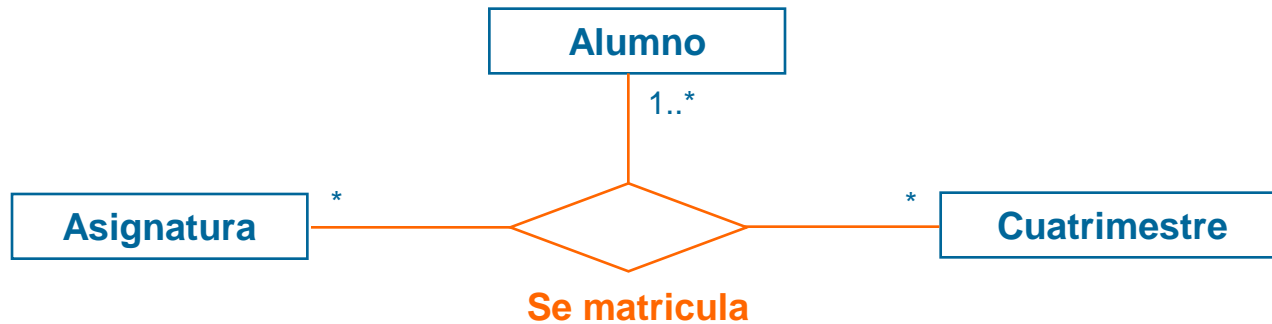


Asociación

Orden superior a dos

Cada instancia de la asociación es una n-tupla de valores de cada una de las clases relacionadas.

Ejemplo



Asociación

Recomendación

Si una clase A puede tener simultáneamente muchos valores para el mismo de atributo b , no ponga el atributo b en A , póngalo en otra clase que esté asociada con A .



¿Podrías poner algún ejemplo?

Asociación

Ejemplo

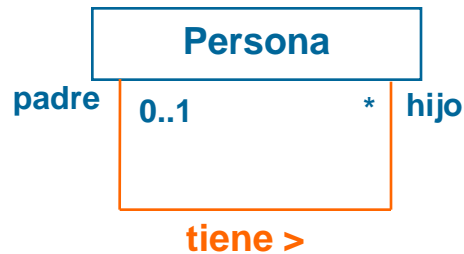
Una **persona** puede tener **varias tarjetas de crédito**



Asociación recursiva

Una clase participa más de una vez en la asociación (con papeles diferentes o no)

Ejemplo



Multiplicidad de asociación

Dada una *instancia a* de una *clase A*, la *multiplicidad* del extremo de la *clase B* asociada con *A* define *cuántas instancias de B* se pueden asociar con *a* en un momento dado.

| | | | |
|---------|-----------|---------|-----------------------|
| Clase A | 1 | Clase B | Exactamente una |
| Clase A | 1..* | Clase B | Muchas (1 o más) |
| Clase A | * | Clase B | Muchas (cero o más) |
| Clase A | 0..* | Clase B | Muchas (cero o más) |
| Clase A | 0..1 | Clase B | Opcional (cero o una) |
| Clase A | 5 | Clase B | Exactamente 5 |
| Clase A | 5+ | Clase B | 5 o más |
| Clase A | 1..10, 15 | Clase B | Entre 1 y 10, ó 15 |

Cota inferior: n+

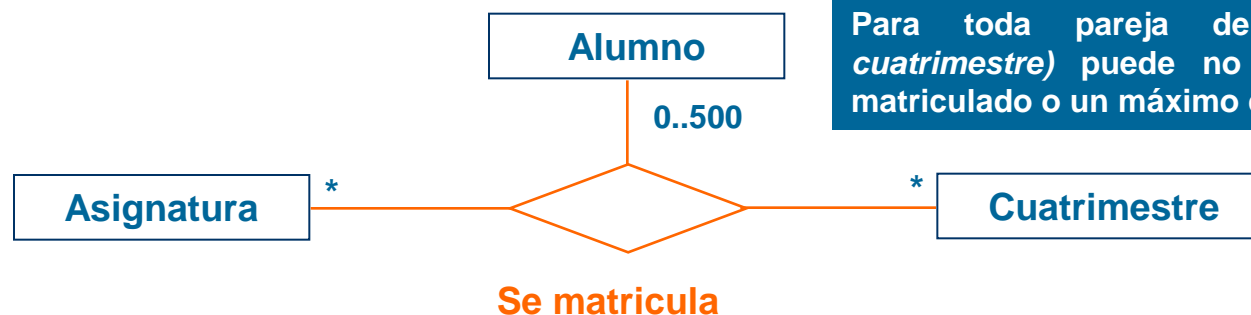
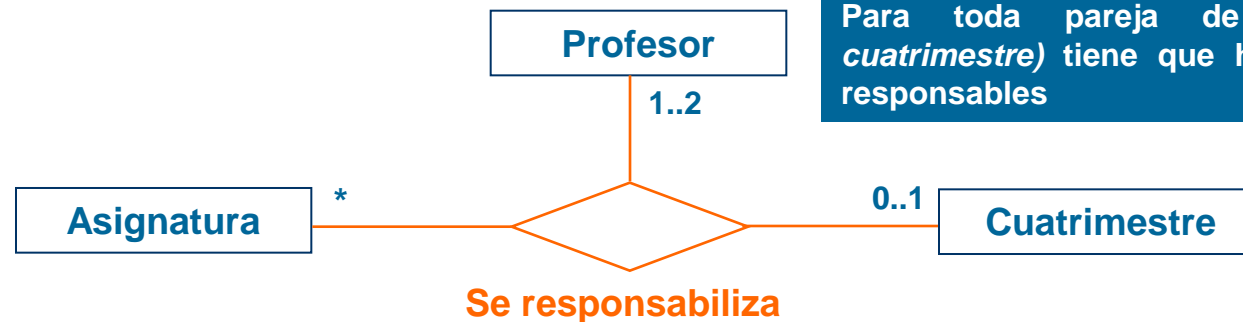
Cota cjto.: n₁, n₂...

Cota rango: m..n

Multiplicidad de asociación

Dada una *instancia a* de una *clase A* y una *instancia b* de una *clase B*, la multiplicidad en el extremo de la *clase C* asociada a *A* y *B* indica **cuántas instancias de C se pueden asociar con la pareja (a,b) en un momento dado**.

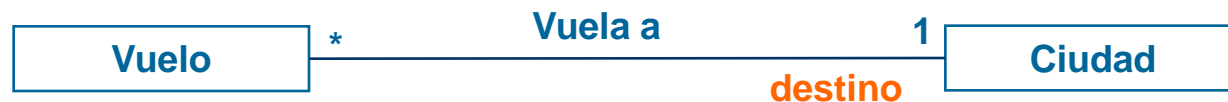
Ejemplo



Rol de asociación

- Cada **extremo** de una **asociación** es un **rol**.
- Un rol tiene diversas propiedades: **nombre** y **multiplicidad**.
- El **nombre de rol** identifica un extremo de la asociación y describe el **papel** que desempeñan los **objetos en la asociación**.

Ejemplo

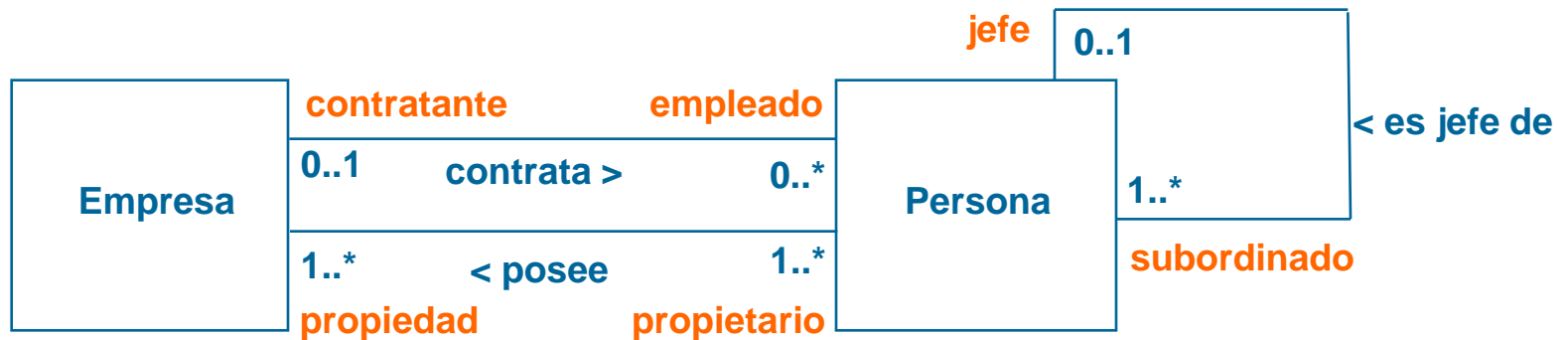


destino (nombre de rol): describe el rol de una ciudad en la asociación “*Vuela a*”

Modelo conceptual de datos

Rol de asociación

Ejemplo



Conviene indicar los nombres de los roles en **asociaciones reflexivas** y cuando hay **más de una asociación** entre dos **clases A y B**.

Rol como concepto

- En ocasiones un rol (especialmente un rol humano) se puede modelar como concepto.
- Facilita inclusión de semántica, atributos y asociaciones específicas.
- Facilita la implementación.



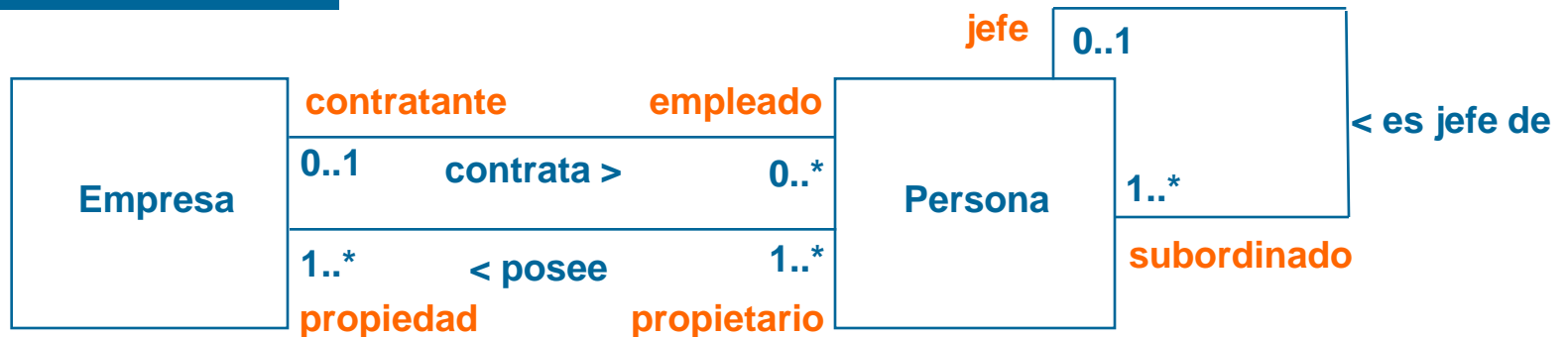
¿Sabrías modelar los roles “empleado” y “propietario” del ejemplo anterior como conceptos?

Modelo conceptual de datos

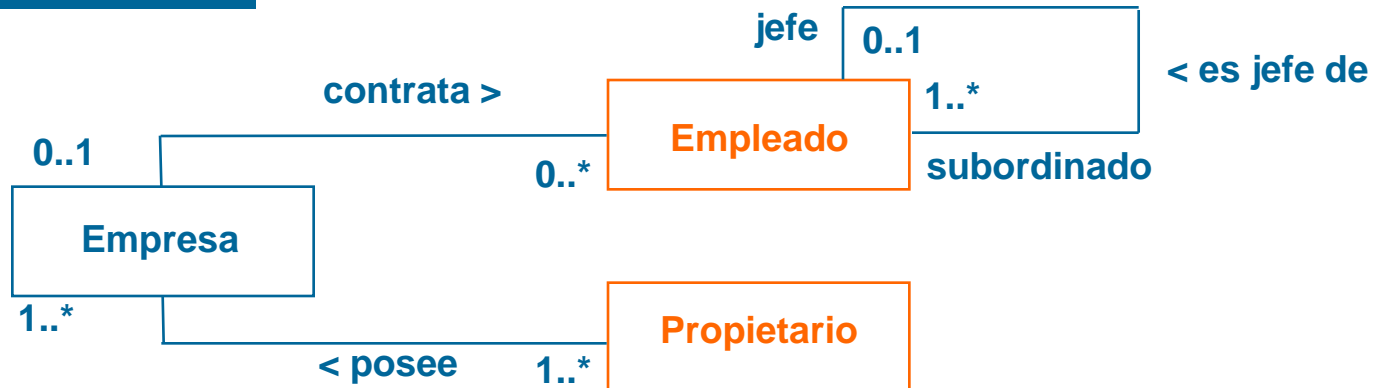
Rol como concepto

Ejemplo

Rol de asociación



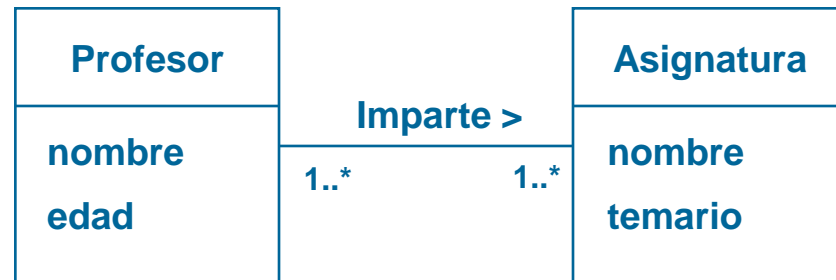
Rol como concepto



Atributo de enlace

- **Enlace:** instancia de una asociación
- **Atributo de enlace:** propiedad de los enlaces de una asociación

Ejemplo

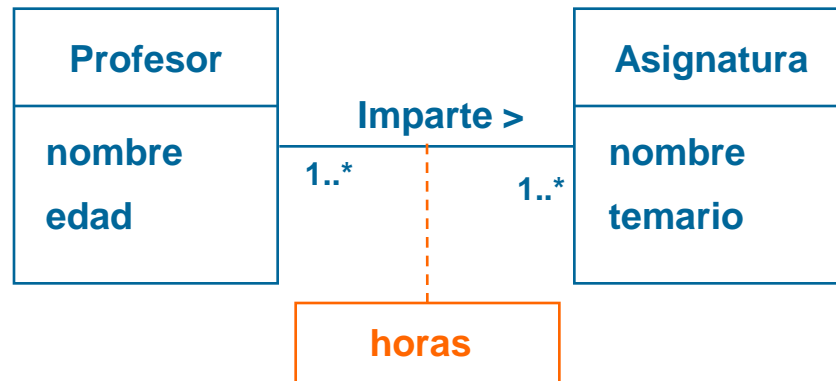


¿El nº horas que un profesor imparte una asignatura es atributo de Profesor? ¿Y de Asignatura?

Atributo de enlace

En el **modelo conceptual** los atributos de enlace **NO** se pueden modelar como **atributos** de ninguna de las **clases** que participan en la **asociación**

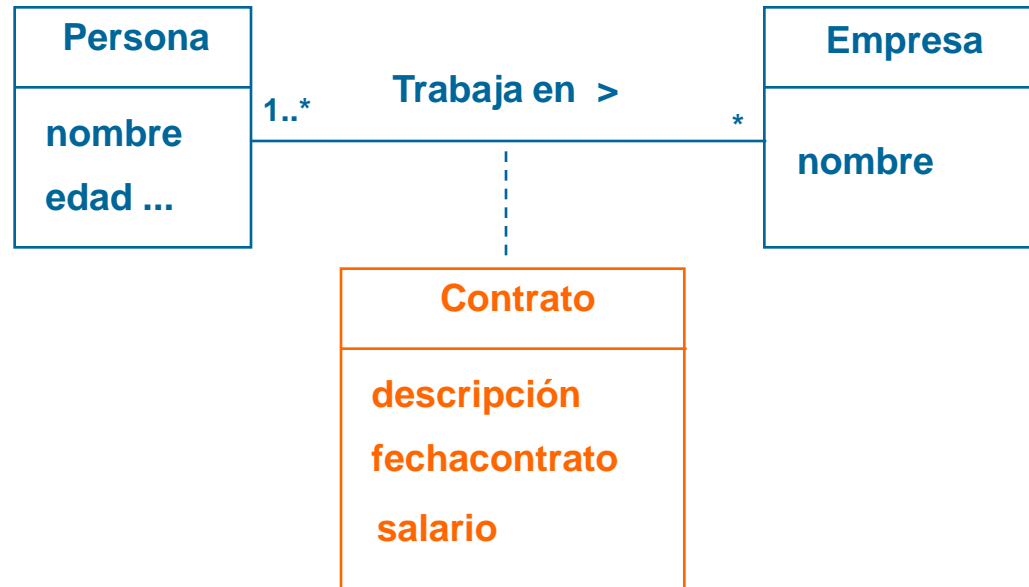
Ejemplo



Clase de asociación

- **Asociación modelada** como una **clase** que tiene **atributos y operaciones**.
- Su **existencia** depende de la asociación.
- Una clase de asociación **puede participar en otras asociaciones**.

Ejemplo



Clase de asociación

Guías

- Existe al menos un **atributo** que está **relacionado con una asociación**.
- El **tiempo de vida de las instancias** de la **asociación** depende de la **asociación**.
- Existe una **asociación n:m** entre dos **clases** e **información asociada** con la **propia asociación**.

Clasificación

- **Restricción semántica** de una **asociación** que indica que los **objetos** de la clase del lado “*muchos*” tienen un **orden explícito**.

Ejemplo

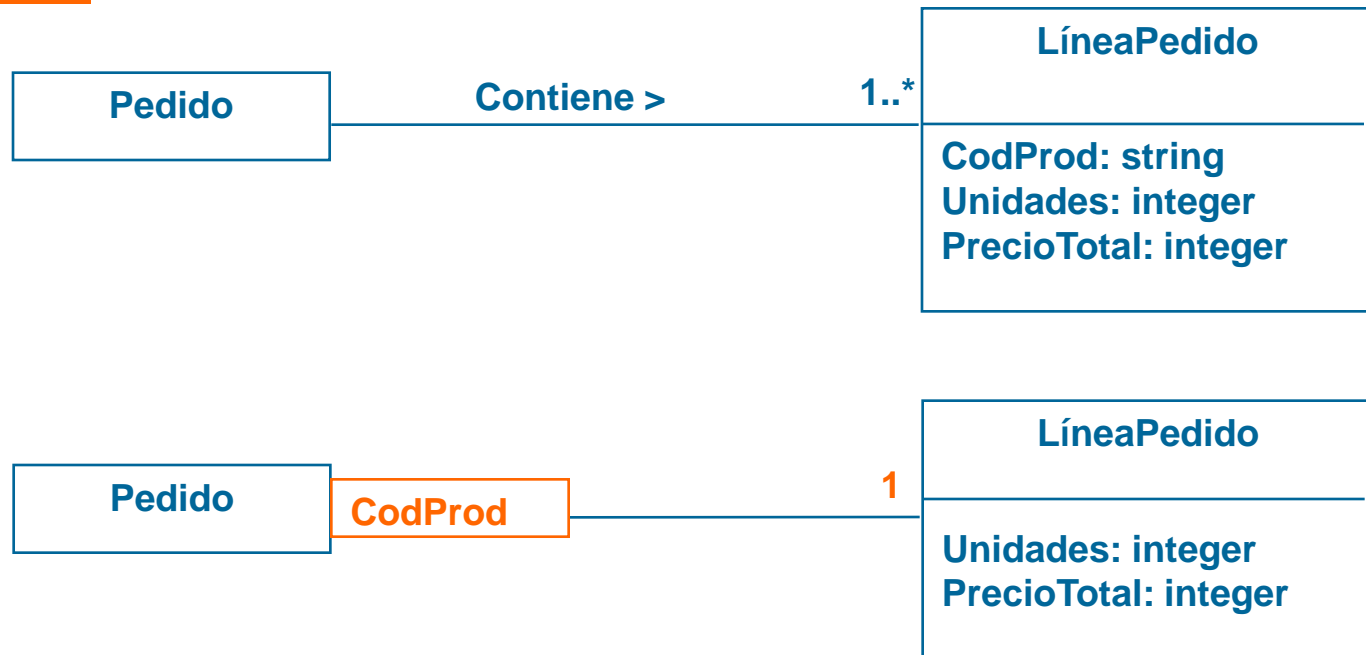


{ordenado}: Las claves asociadas con un usuario están ordenadas de menos a más recientemente usada.

Asociación calificada (calificación)

- Relaciona **dos clases** (asociación 1:m ó n:m) y un **calificador**.
- **Calificador**: distingue entre el conjunto de objetos del lado muchos (reduce la multiplicidad de la asociación).

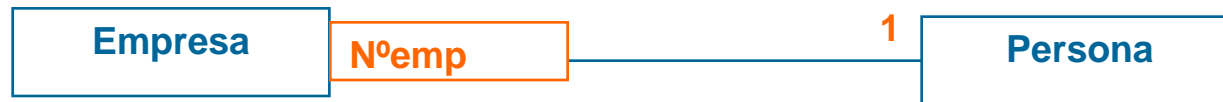
Ejemplo



“Dentro del mismo pedido no pueden existir dos líneas con el mismo producto”.
Pueden distinguirse las líneas de pedido en un pedido según su **CodProd**.

Asociación calificada (calificación)

Ejemplo

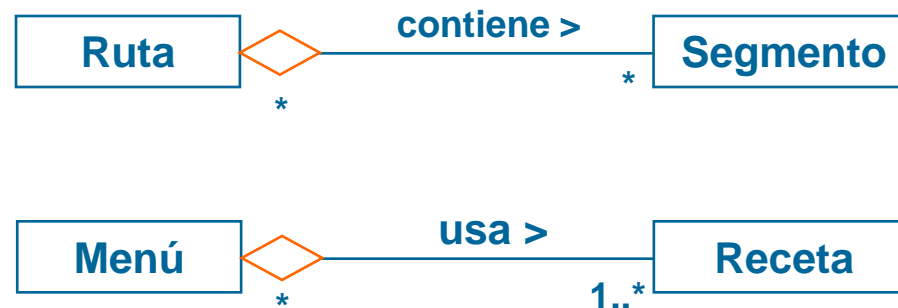


“En una determinada empresa no pueden existir dos empleados con el mismo número”. Pueden distinguirse los empleados de una empresa por su número de empleado.

Agregación

- Es un **tipo de asociación** usada para modelar relaciones “*parte-todo*” entre objetos.
- El “*todo*” se denomina **compuesto** y las “*partes*” **componentes**.

Ejemplo



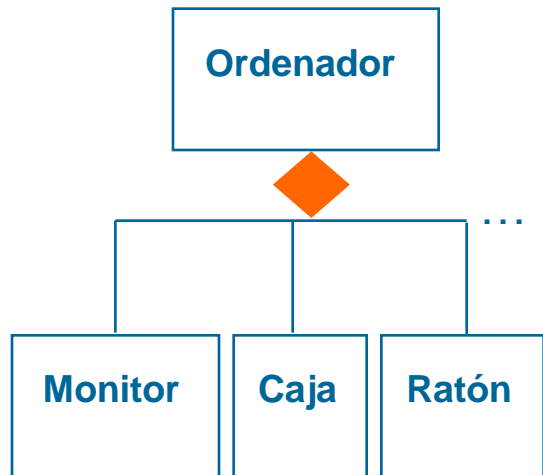
La **distinción** entre **asociación** y **agregación** es a menudo **subjetiva**.

Modelo conceptual de datos

Agregación

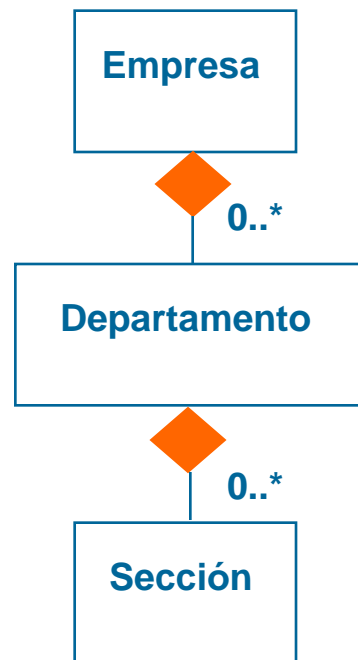
Fija

Niveles y partes fijas



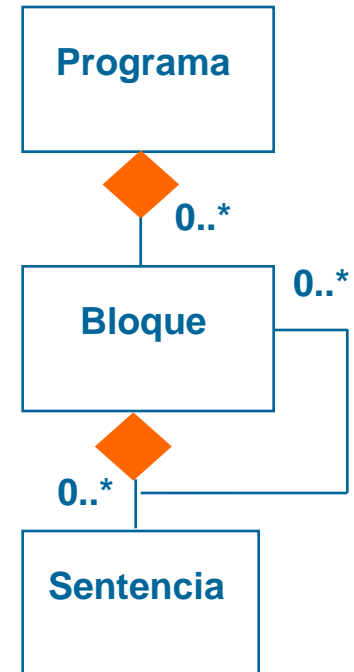
Variable

Niveles fijos y partes variables



Recursiva

Niveles ilimitados



Agregación

Propiedades

Las agregaciones tienen dos propiedades:

- **Dependencia:** ¿La existencia de una parte va ligada a la del agregado?
- **Exclusividad:** ¿Una parte puede pertenecer a más de un agregado?

Tipos de agregación

Según las propiedades de **dependencia** y **exclusividad** se diferencian **cuatro tipos de agregaciones**:

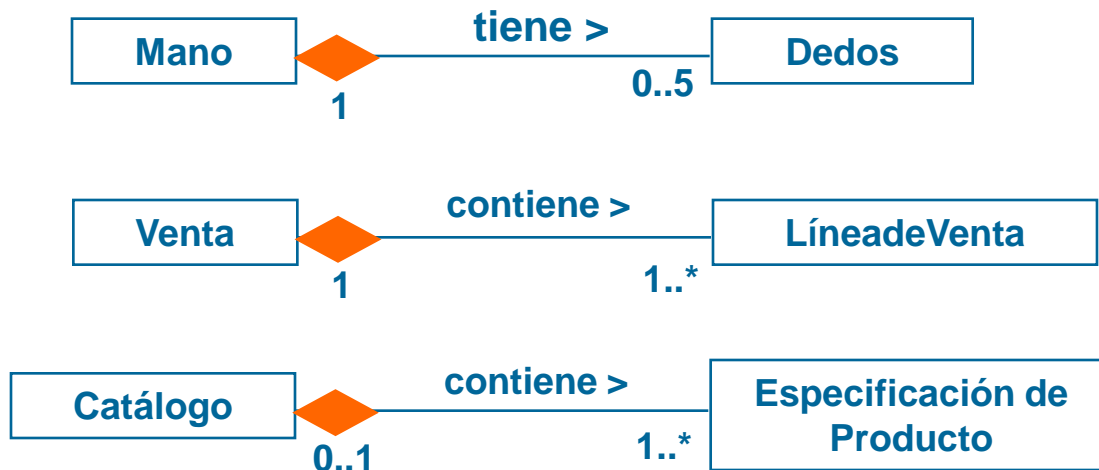
| Dependencia | Exclusividad |
|-------------|--------------|
| Sí | Sí |
| Sí | No |
| No | Sí |
| Sí | Sí |

Composición

Composición

- Es un tipo de **agregación exclusiva y dependiente**
- La **multiplicidad del extremo compuesto** puede ser **como máximo 1** (como máximo un componente lo es de un sólo compuesto)
- Si un componente está asociado a un compuesto y el **compuesto se borra** entonces el **componente también se ha de borrar** (no lo puede sobrevivir)
- Los **componentes se pueden borrar antes de borrar el compuesto**

Ejemplos



Agregación y composición

Guías agregación y composición

- Existe una **relación todo-parte física o lógica**.
- Algunas **propiedades del compuesto se propagan a los componentes**. (destrucción, movimiento, etc).
- Si hay **duda**, dejar la decisión para la **fase de diseño**.

Guías composición

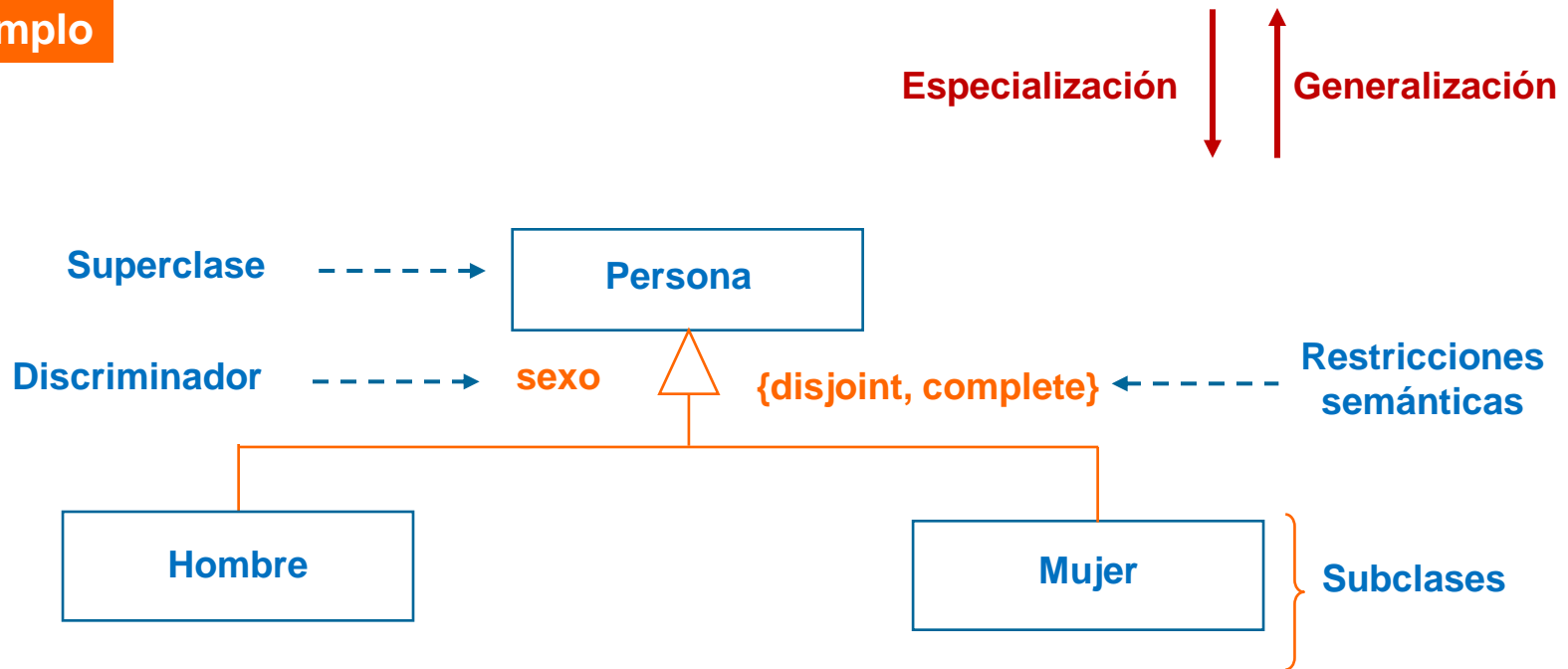
- La **vida del componente depende de la vida del compuesto**.
- Existe una **dependencia crear-borrar del componente respecto del compuesto**.

Modelo conceptual de datos

Generalización

- Identificar **elementos comunes** entre los **objetos** definiendo **relaciones** entre **superclase** (objeto general) y **subclases** (objetos especializados).

Ejemplo



Discriminador: propiedad de la superclase en base a la cual se realiza la especialización.

Generalización

Restricciones semánticas

Disjoint y overlapping

- **Disjoint:** Un objeto de la superclase no puede ser objeto de más de una subclase.
- **Overlapping:** un objeto de la superclase puede ser objeto de más de una subclase.

Complete e incomplete

- **Complete:** se especifican todas las posibles subclases de la superclase en base a los posibles valores del discriminador.
- **Incomplete:** no se especifican todas las posibles subclases de la superclase en base a los posibles valores del discriminador.

Generalización

Reglas

Regla 100%

- El **100%** de la **definición** de la **superclase** se debe poder **aplicar** a **todas las subclases**.
- La subclase debe ajustarse al **100%** de los **atributos**, **asociaciones** y **restricciones** de Superclase.

Regla “es-un”

- **Todos los objetos** de una **subclase** deben ser **objetos** de la **superclase**.

Toda **subclase** debe estar de acuerdo con la **regla 100%** y con la **regla “es-un”**

Generalización

Razones para especializar una clase

- La **subclase** tiene **atributos** adicionales **específicos**.
- La **subclase** tiene **asociaciones** adicionales **específicos**.
- La **subclase** es **tratada** o manipulada de **manera diferente** a la superclase o a las otras subclases.

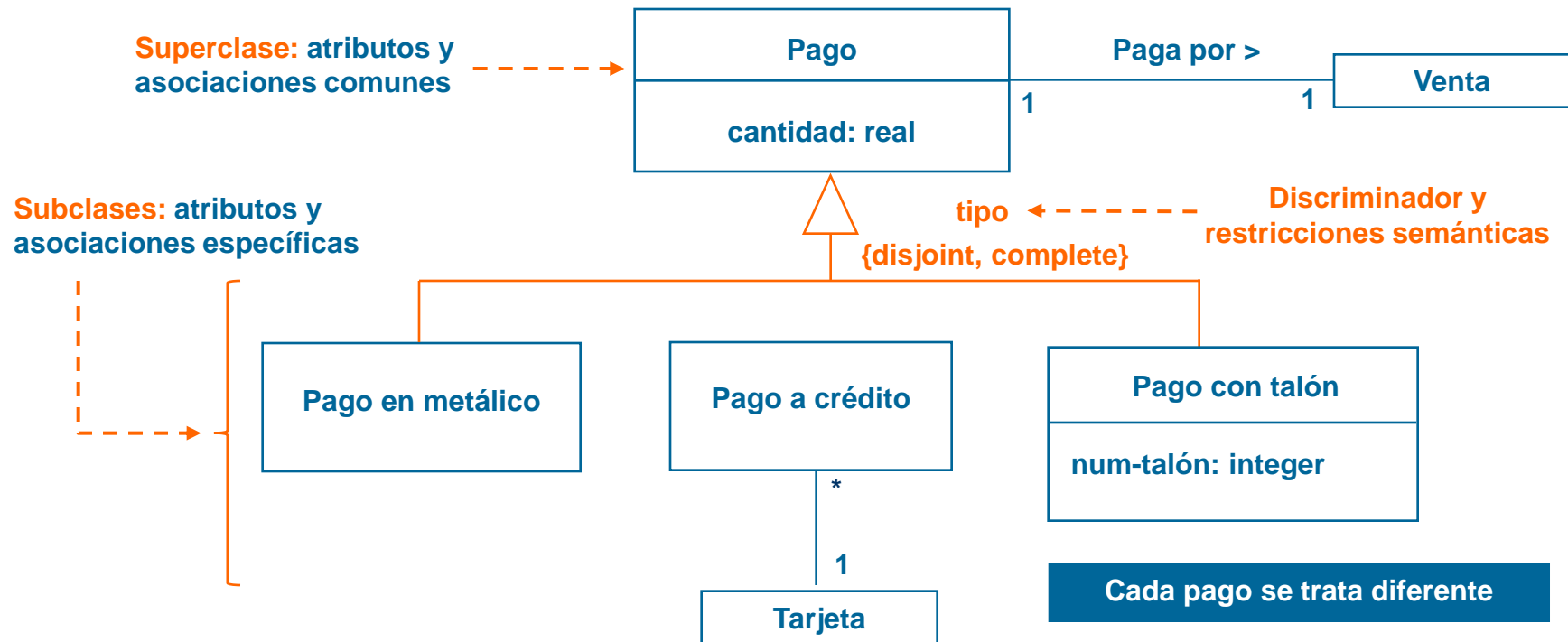
Razones para generalizar clases

- Las **subclases** representan variaciones de un **mismo concepto**.
- Las **subclases** tienen **atributos comunes** que pueden ser factorizados y modelados en las **superclases**.
- Las **subclases** tienen **asociaciones comunes** que pueden ser factorizadas y relacionadas con la **superclase**.
- Las **subclases** se ajustan a las reglas **100%** y “**es-un**”.

Modelo conceptual de datos

Generalización

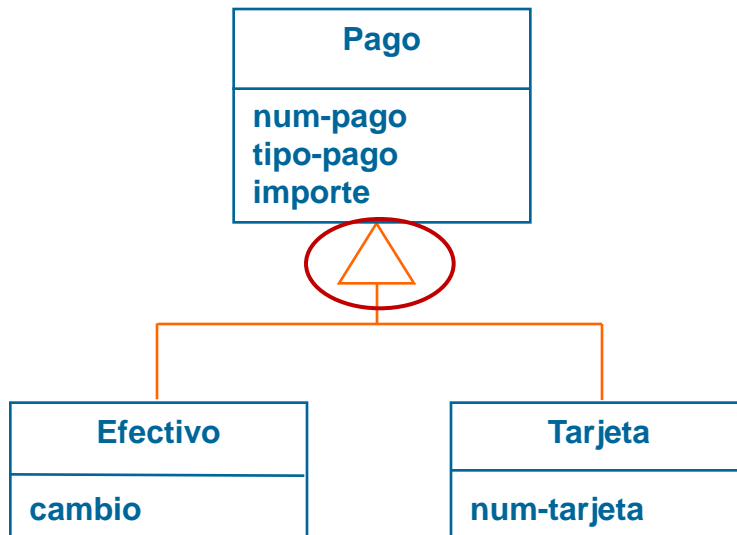
Ejemplo



Generalización

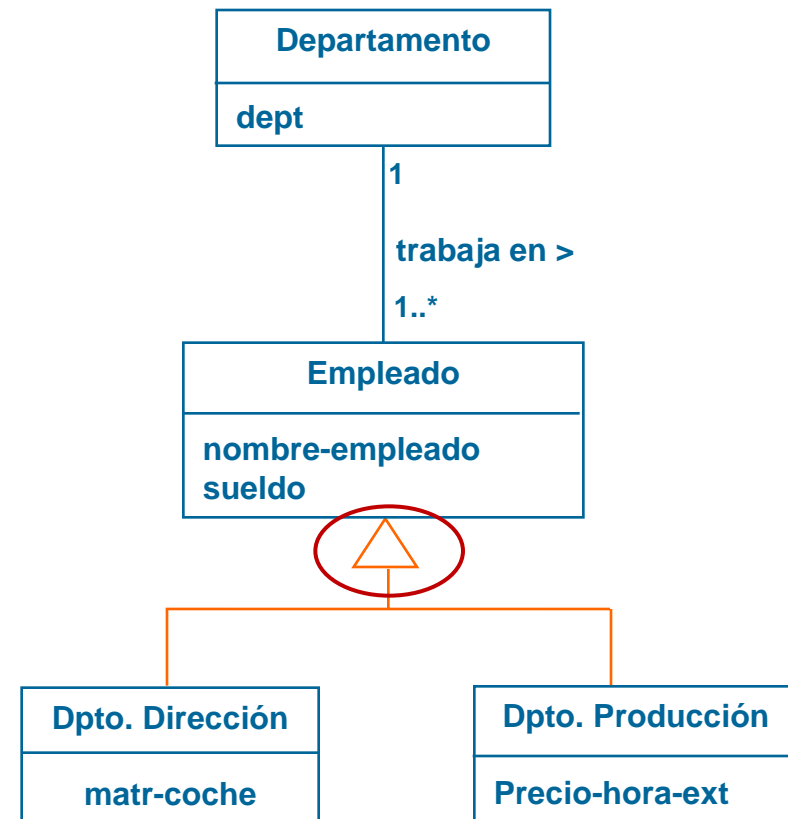
Ejemplos

A



¿Sabrías decir cuáles son los discriminadores y restricciones semánticas de los ejemplos A y B?

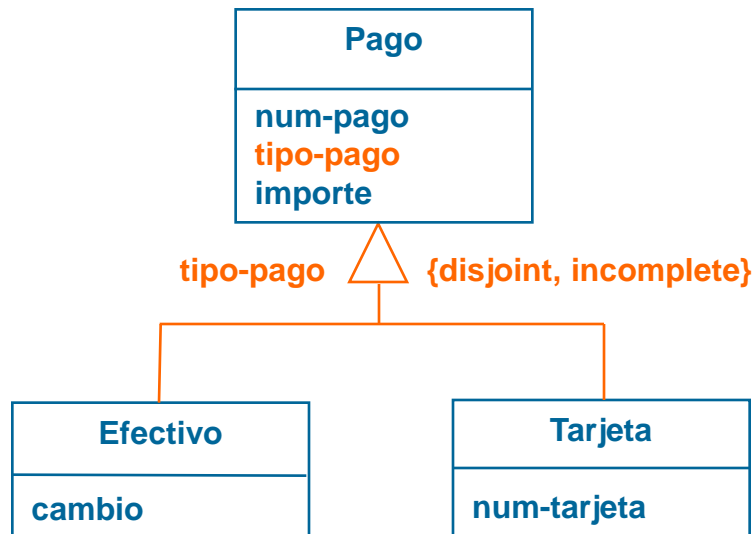
B



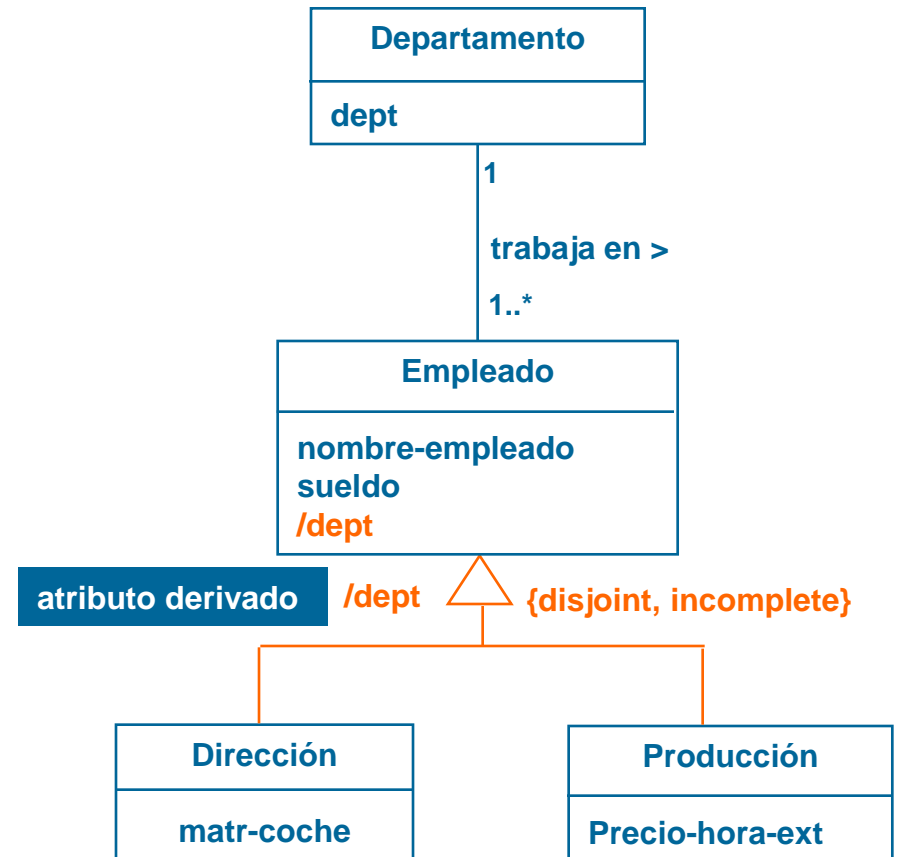
Generalización

Ejemplos

A

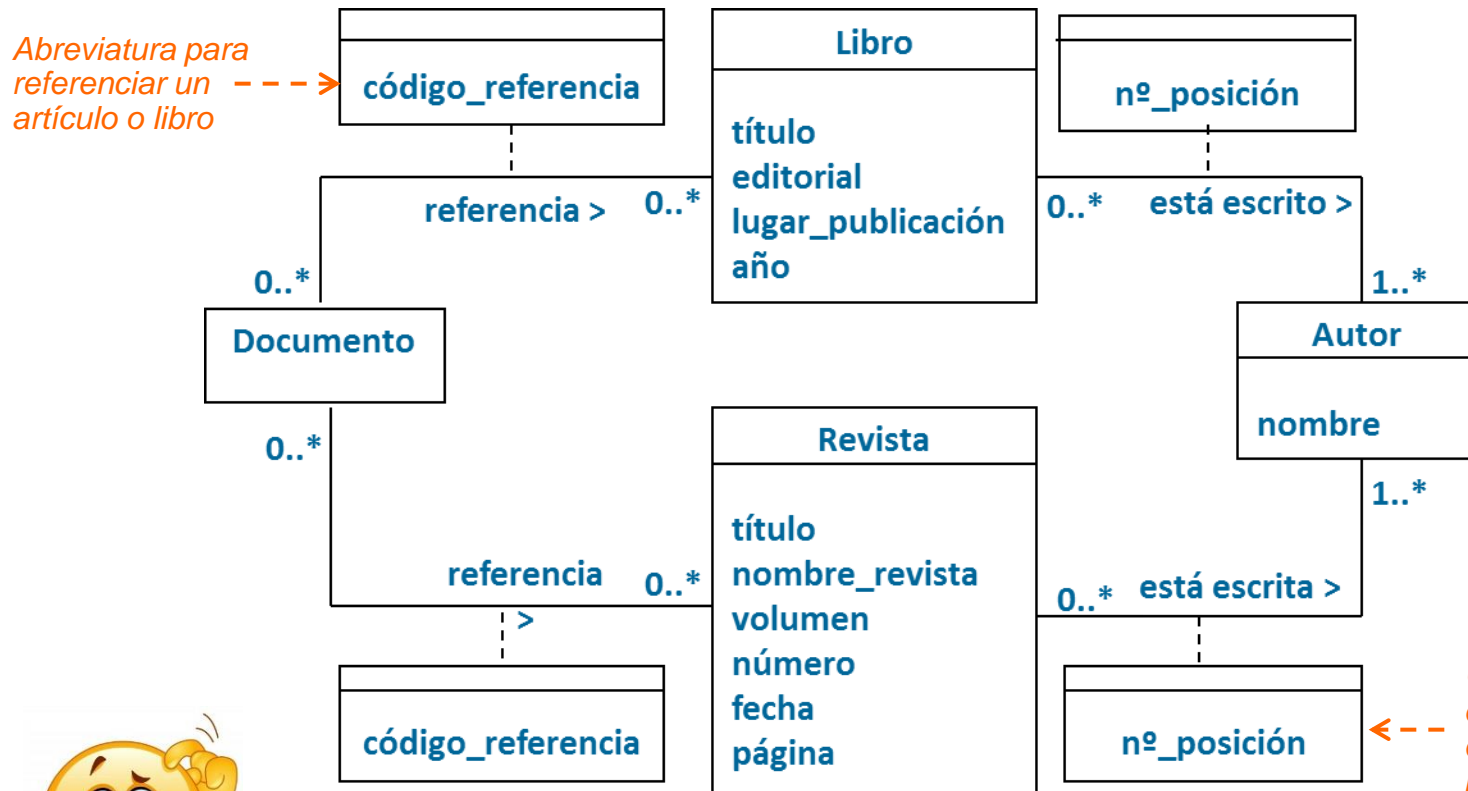


B



Generalización

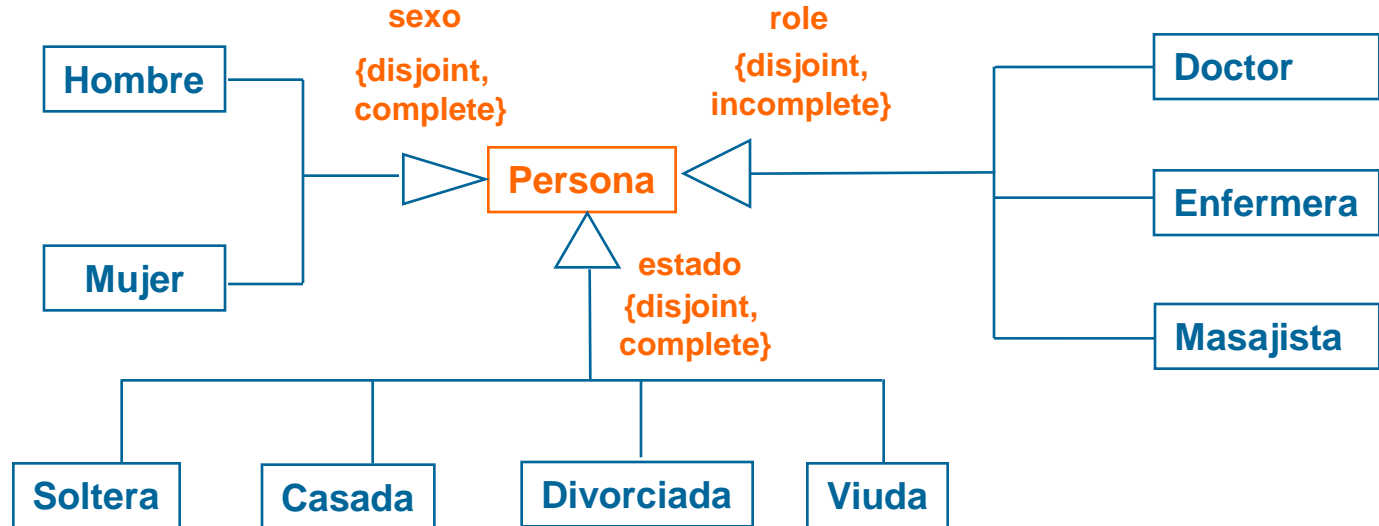
Refina el diagrama de clases de la figura utilizando la relación de generalización



Clasificación múltiple

Una **superclase** tiene **diversas jerarquías de especialización** en base a diferentes discriminadores.

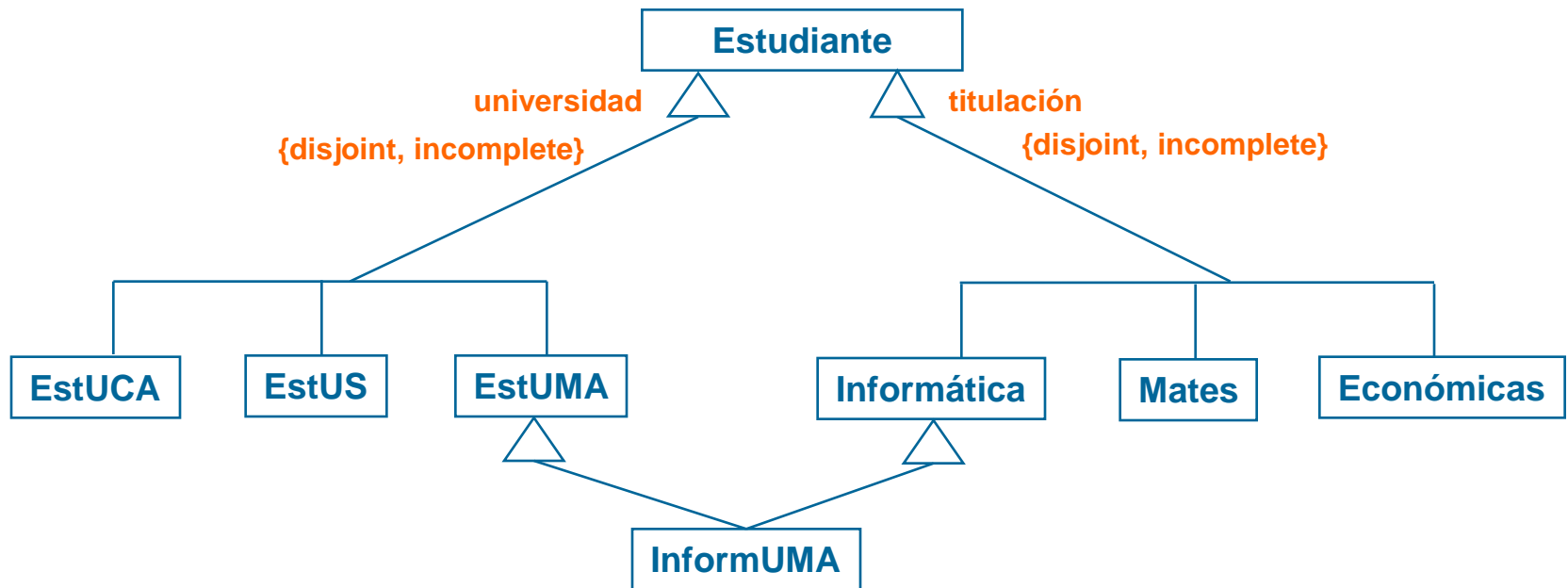
Ejemplo



Herencia múltiple

Una **subclase** tiene **más** de **una superclase** como consecuencia de realizar diferentes generalizaciones.

Ejemplo

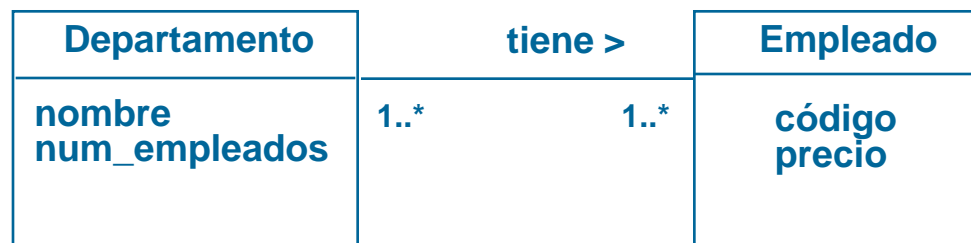
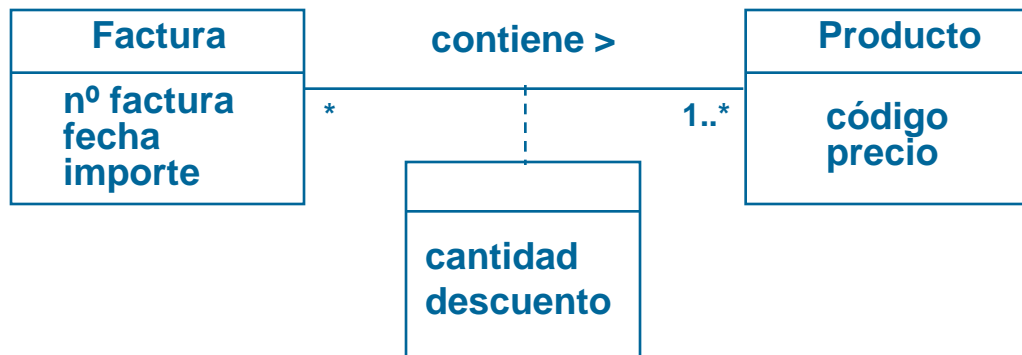


Sólo se puede utilizar si no hay **conflictos de herencia**: subclase con un atributo o asociación común a varias superclases.

Información derivada

- **Elemento derivado (atributo o asociación):** se puede obtener a partir de otros elementos (no añaden información fundamental).
- **Regla de derivación:** especifica cómo se obtiene el elemento.
- Se incluyen para **mejorar la calidad** del modelo.

Ejemplo atributo derivado

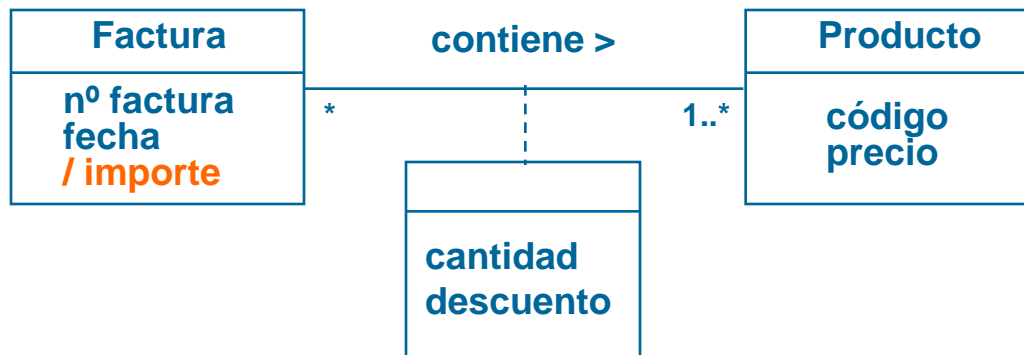


¿Hay algún atributo derivado?

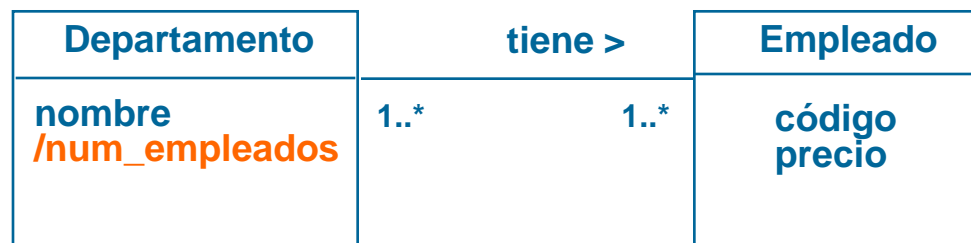
Información derivada

- **Elemento derivado (atributo o asociación):** se puede obtener a partir de otros elementos (no añaden información fundamental).
- **Regla de derivación:** especifica cómo se obtiene el elemento.
- Se incluyen para **mejorar la calidad** del modelo.

Ejemplo atributo derivado



/importe: El importe de una factura *F* es igual a la suma del importe (*cantidad* x *precio*) de todos productos que contiene la factura *F*.

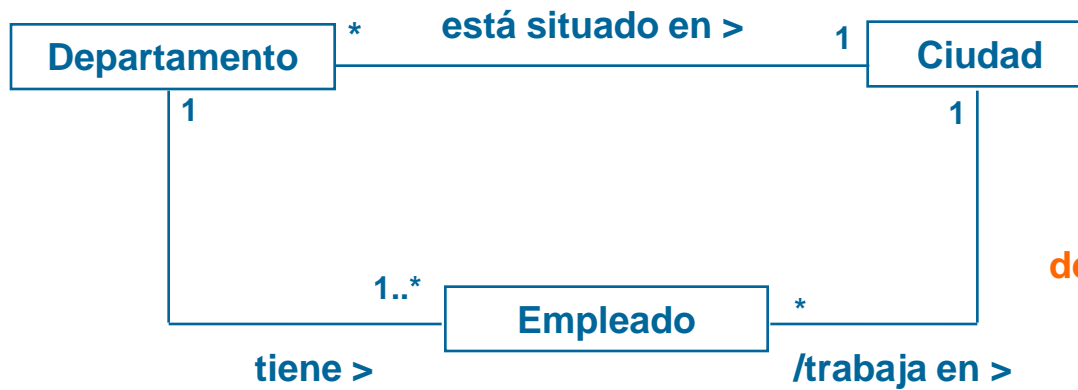


/num_empleados: El número de empleados de un departamento *D* es igual al número de enlaces de la asociación *tiene* en los que interviene *D*.

Información derivada

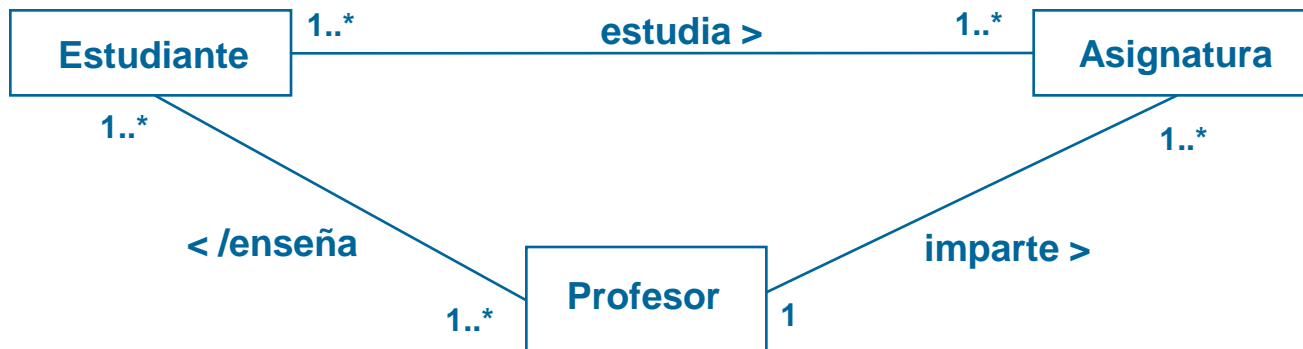
Ejemplo asociación derivada

A



¿Hay alguna asociación derivada en los ejemplos A y B?

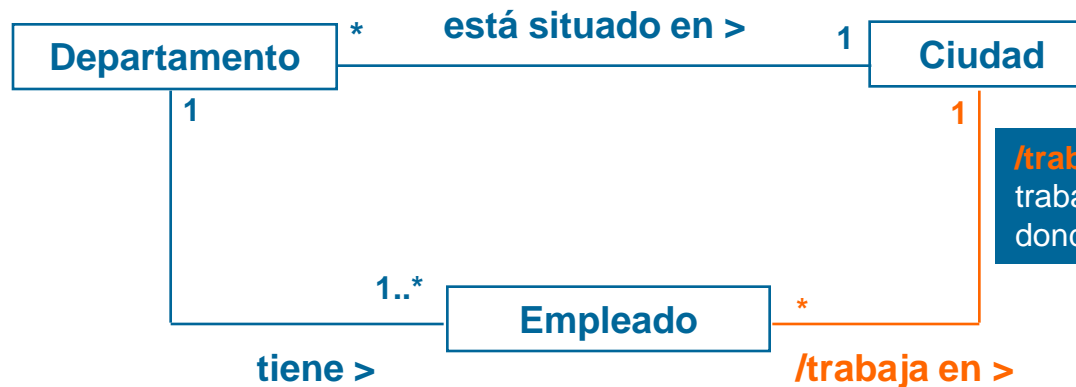
B



Información derivada

Ejemplo asociación derivada

A



/trabaja en: La ciudad en la que trabaja un empleado es la ciudad donde está situado su departamento.

B



/enseña: Los estudiantes a los que enseña un profesor son los estudiantes que estudian las asignaturas que imparte el profesor.

Restricciones

Limitan los valores que pueden tomar los elementos del modelo (atributos, asociaciones, etc.)

Ejemplo restricción de atributo

| Producto |
|----------------------------------|
| Código Stock min Stock max |

{Stock max > Stock min}

{Stock max = 1000}

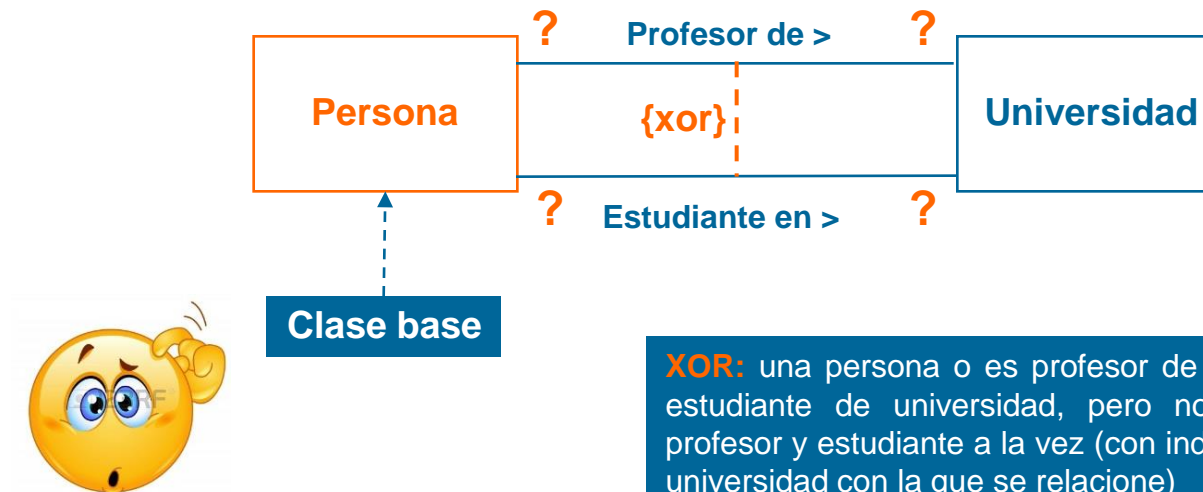
{Stock min > = 10}

Restricciones

Restricción de asociación: XOR

- Restricción de varias asociaciones en las que interviene una misma **clase base**.
- Una **instancia** de la **clase base** (objeto) puede participar como **máximo en una** de las **asociaciones** unidas por **xor**.

Ejemplo



XOR: una persona o es profesor de universidad o es estudiante de universidad, pero no puede ser las profesor y estudiante a la vez (con independencia de la universidad con la que se relacione)

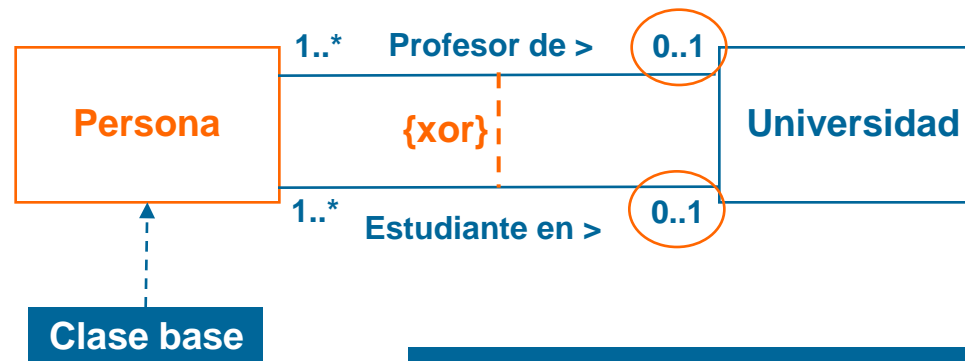
¿Sabrías poner la multiplicidad de las asociaciones?

Restricciones

Restricción de asociación: XOR

- Restricción de varias asociaciones en las que interviene una misma **clase base**.
- Una **instancia** de la **clase base** (objeto) puede participar como **máximo** en una de las **asociaciones** unidas por **xor**.

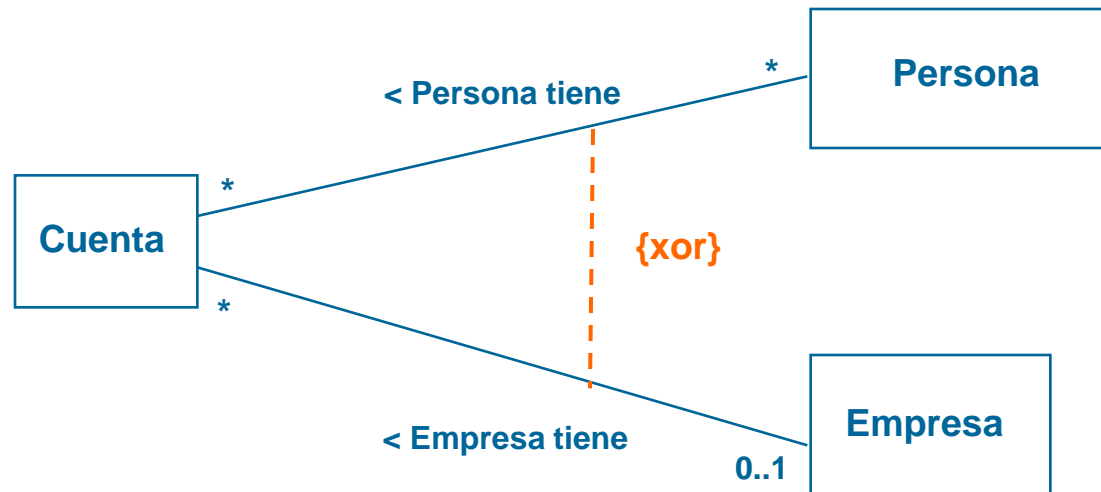
Ejemplo



XOR: una persona o es profesor de universidad o es estudiante de universidad, pero no puede ser las profesor y estudiante a la vez (con independencia de la universidad con la que se relacione)

Restricciones

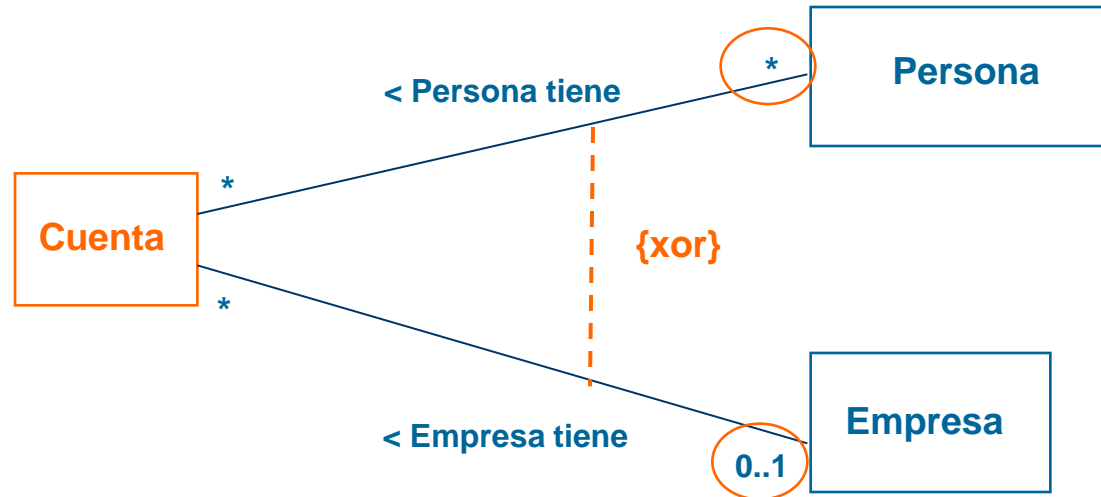
Ejemplo



¿Sabes el significado del modelo? ¿cuál es la clase base?

Restricciones

Ejemplo



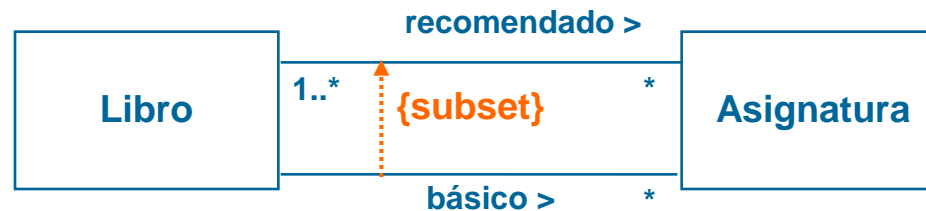
XOR: una cuenta o es de personas (una o varias) o es de empresa (sólo una), pero no puede ser de ambos a la vez.

Restricciones

Restricción de asociación: Subset

- Restricción de dos asociaciones A y B en las que intervienen las **mismas clases**.
- Indica que las **instancias** (enlaces) de una de las clases (A) son un **subconjunto** de las **instancias** de la otra clase (B).

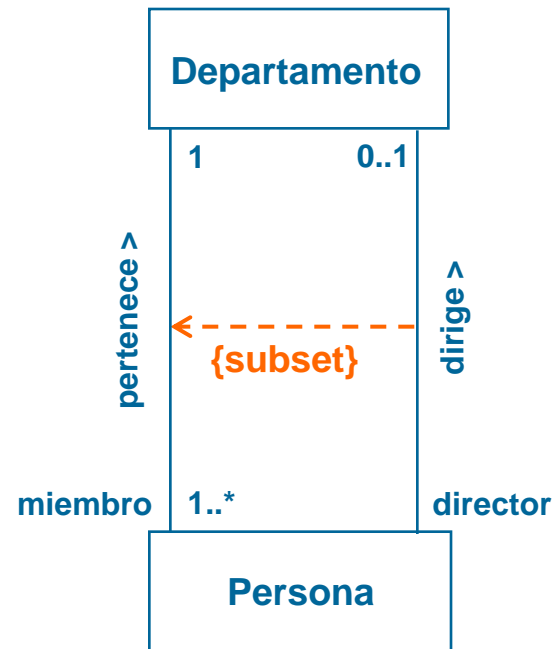
Ejemplo



Subset: los libros básicos de una asignatura tienen que un subconjunto de los libros recomendados en la asignatura.

Restricciones

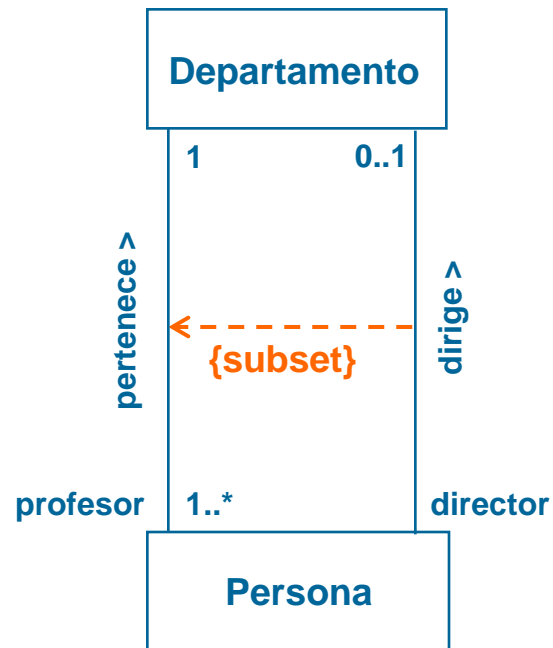
Ejemplo



¿Podrías explicar el significado de la restricción subset en el ejemplo?

Restricciones

Ejemplo

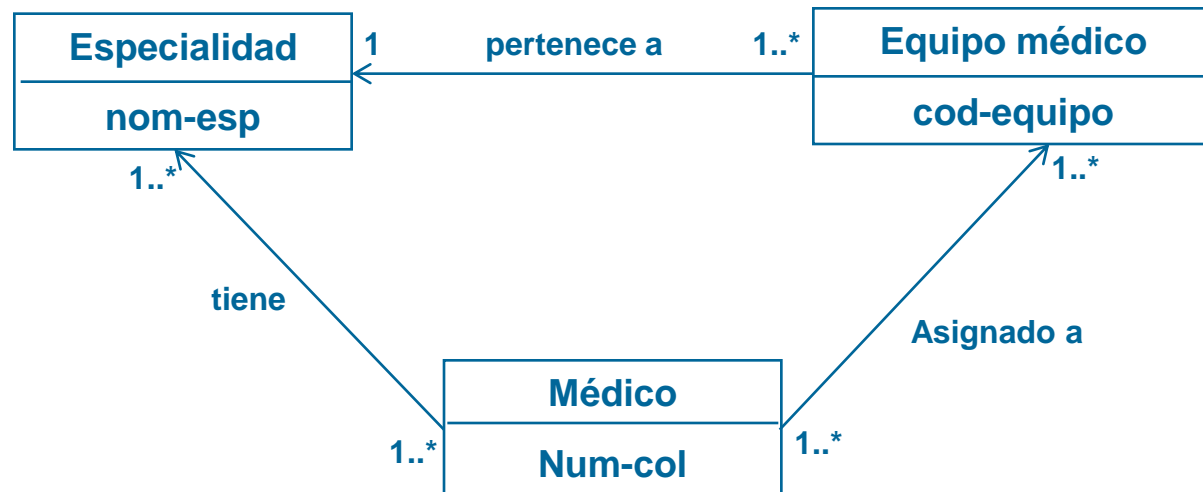


Subset: el director de un departamento tiene que ser alguno de los profesores del mismo.

Restricciones textuales

- Las restricciones que **no se pueden especificar gráficamente** con la notación UML se especifican de forma textual (lenguaje natural, OCL, etc.).
- Restricción de **clave externa** de una clase: no puede haber dos instancias (objetos) de la clase que tengan los mismos valores de los atributos que forman la clave.

Ejemplo



Restricción textual: un médico no puede estar asignado a ningún equipo médico que pertenezca a una especialidad que el médico no tenga

Restricciones clave externa:

{Especialidad, nom-esp}, {Equipo médico, cod-equipos} y {Médico, Num-col}

Ejemplo 1

Realizar un **modelo conceptual de datos** que responda a las siguientes especificaciones:

Las áreas metropolitanas tienen una serie de hoteles, algunos de los cuales pertenecen a una determinada cadena de hoteles. Los hoteles aceptan varias tarjetas de crédito. De las áreas metropolitanas interesa conocer el nombre del área, el nombre del estado o provincia a la que pertenece y el nombre del país. De los hoteles interesa conocer el nombre, la dirección, el nº de habitaciones, el nº de teléfono, el nº de estrellas, el precio de la habitación simple y el precio de la habitación doble. De las cadenas de hoteles interesa conocer el nombre y el director. De las tarjetas de crédito sólo interesa conocer el nombre.



¿Sabrías identificar las clases?

Ejemplo 1

Realizar un **modelo conceptual de datos** que responda a las siguientes especificaciones:

Las **áreas metropolitanas** tienen una serie de **hoteles**, algunos de los cuales pertenecen a una determinada **cadena de hoteles**. Los hoteles aceptan varias **tarjetas de crédito**. De las áreas metropolitanas interesa conocer el nombre del área, el nombre del **estado o provincia** a la que pertenece y el nombre del **país**. De los hoteles interesa conocer el nombre, la dirección, el nº de habitaciones, el nº de teléfono, el nº de estrellas, el precio de la habitación simple y el precio de la habitación doble. De las cadenas de hoteles interesa conocer el nombre y el **director**. De las tarjetas de crédito sólo interesa conocer el nombre.



Ejemplo 2 (problema 2)

Realizar un **modelo conceptual de datos** que responda a las siguientes especificaciones de un sistema de vuelos:

Los aeropuertos dan servicio a varias ciudades y hay ciudades que tienen más de un aeropuerto. Los vuelos entre los aeropuertos los gestionan las líneas aéreas y se describen como se muestra en el siguiente ejemplo: “vuelo *TW250* del aeropuerto de *S. Pablo de Sevilla* al aeropuerto *Reina Sofía de Tenerife*, tiene prevista la salida a las *7:42 am* y una duración estimada de *1 hora y 80 minutos*; el vuelo se realizará en un *DC9* (modelo de avión) *todos los días de la semana, excepto los sábados*, y estará vigente desde *febrero del 2015* hasta *junio del 2015*”. De los modelos de aviones interesa conocer el código del modelo (ej. DC9) y el fabricante, de los aeropuertos el código y el nombre, y de las ciudades y las líneas aéreas sólo interesa conocer el nombre.



¿Sabrías identificar las clases?

Ejemplo 2 (problema 2)

Realizar un **modelo conceptual de datos** que responda a las siguientes especificaciones de un sistema de vuelos:

Los **aeropuertos** dan servicio a varias **ciudades** y hay ciudades que tienen más de un aeropuerto. Los **vuelos** entre los aeropuertos los gestionan las **líneas aéreas** y se describen como se muestra en el siguiente ejemplo: “vuelo *TW250* del aeropuerto de *S. Pablo de Sevilla* al aeropuerto *Reina Sofía de Tenerife*, tiene prevista la salida a las *7:42 am* y una duración estimada de *1 hora y 80 minutos*; el vuelo se realizará en un *DC9* (modelo de avión) *todos los días de la semana, excepto los sábados*, y estará vigente desde *febrero del 2015* hasta *junio del 2015*”. De los **modelos de aviones** interesa conocer el código del modelo (ej. DC9) y el **fabricante**, de los aeropuertos el código y el nombre, y de las ciudades y las líneas aéreas sólo interesa conocer el nombre.



Ejemplo 3 (problema 3)

Realizar un **modelo conceptual de datos** que responda a las siguientes especificaciones del catastro de viviendas de un determinado municipio:

En el municipio existen diferentes zonas urbanas en cada una de las cuales se han edificado varias viviendas. De las personas que habitan las viviendas, además de sus datos (dni, nombre y apellidos) interesa conocer quién es su cabeza de familia. De cada vivienda, además de la información correspondiente a sus características (calle, número, piso, código_postal, metros, etc.), es necesario conocer quién es el propietario. Considerar los siguientes supuestos semánticos:

- Cada persona habita en una vivienda, la cual es considerada como su vivienda o residencia principal.
- Cada vivienda tiene un propietario.
- Una persona puede ser propietaria de más de una vivienda.
- Las viviendas se encuentran en una única zona urbana correspondiente al municipio.
- Las zonas urbanas en las que está dividido geográficamente el municipio tienen nombres diferentes.



¿Sabrías identificar las clases?

Ejemplo 3 (problema 3)

Realizar un **modelo conceptual de datos** que responda a las siguientes especificaciones del catastro de viviendas de un determinado municipio:

En el **municipio** existen diferentes **zonas urbanas** en cada una de las cuales se han edificado varias **viviendas**. De las **personas** que habitan las viviendas, además de sus datos (dni, nombre y apellidos) interesa conocer quién es su **cabeza de familia**. De cada vivienda, además de la información correspondiente a sus características (calle, número, piso, código_postal, metros, etc.), es necesario conocer quién es el **propietario**. Considerar los siguientes supuestos semánticos:

- Cada persona habita en una vivienda, la cual es considerada como su vivienda o residencia principal.
- Cada vivienda tiene un propietario.
- Una persona puede ser propietaria de más de una vivienda.
- Las viviendas se encuentran en una única zona urbana correspondiente al municipio.
- Las zonas urbanas en las que está dividido geográficamente el municipio tienen nombres diferentes.



Ejemplo 4 (problema 4)

Realizar un **modelo conceptual de datos** que responda a las siguientes especificaciones de un centro universitario:

El centro está integrado por diferentes departamentos y, de cada uno de ellos, interesa saber el código que lo identifica, el número de profesores que lo forman, quién es el director del departamento (tiene que ser uno de los profesores del departamento) y las áreas de conocimiento que incluye. Estas áreas están codificadas y cada departamento incluye una como mínimo: así por ejemplo, el Departamento de Lenguajes y Sistemas Informáticos incluye dos áreas de conocimiento: Lenguajes y Sistemas Informáticos (LSI) y Ciencias de la Computación e Inteligencia Artificial (IA). Cada profesor sólo puede pertenecer a un departamento, y de cada uno de ellos interesa conocer su dni, nombre, apellidos, dirección, teléfono y asignaturas que imparte. Una misma asignatura puede ser impartida por más de un profesor e interesa saber cuántas horas imparte cada profesor cada asignatura. Así mismo, de cada asignatura se conoce su código de identificación, descripción, créditos totales, créditos teóricos, créditos prácticos y los libros recomendados, diferenciando los que son básicos y los que son complementarios. Los libros recomendados están en la biblioteca del centro y de cada uno de ellos se conoce el código que lo identifica, el título, el primer autor y el número de ejemplares disponibles. Por último, las asignaturas pueden ser obligatorias, troncales y optativas. De las asignaturas troncales, interesa saber la materia troncal a la que corresponden y de las asignaturas optativas, en qué cursos se puede elegir y qué recomendaciones hay que tener en cuenta al elegir una asignatura optativa en un determinado curso. Así mismo, interesa saber el aula en la que se imparten las clases de cada curso.

¿Sabrías identificar las clases?



Modelo conceptual de datos

Ejemplo 4 (problema 4)

Realizar un **modelo conceptual de datos** que responda a las siguientes especificaciones de un centro universitario:

El **centro** está integrado por diferentes **departamentos** y, de cada uno de ellos, interesa saber el código que lo identifica, el número de **profesores** que lo forman, quién es el **director** del departamento (tiene que ser uno de los profesores del departamento) y las **áreas de conocimiento** que incluye. Estas áreas están codificadas y cada departamento incluye una como mínimo: así por ejemplo, el Departamento de Lenguajes y Sistemas Informáticos incluye dos áreas de conocimiento: Lenguajes y Sistemas Informáticos (LSI) y Ciencias de la Computación e Inteligencia Artificial (IA). Cada profesor sólo puede pertenecer a un departamento, y de cada uno de ellos interesa conocer su dni, nombre, apellidos, dirección, teléfono y **asignaturas** que imparte. Una misma asignatura puede ser impartida por más de un profesor e interesa saber cuántas horas imparte cada profesor cada asignatura. Así mismo, de cada asignatura se conoce su código de identificación, descripción, créditos totales, créditos teóricos, créditos prácticos y los **libros** recomendados, diferenciando los que son básicos y los que son complementarios. Los libros recomendados están en la **biblioteca** del centro y de cada uno de ellos se conoce el código que lo identifica, el título, el primer **autor** y el número de ejemplares disponibles. Por último, las asignaturas pueden ser **obligatorias**, **troncales** y **optativas**. De las asignaturas troncales, interesa saber la **materia troncal** a la que corresponden y de las asignaturas optativas, en qué **cursos** se puede elegir y qué **recomendaciones** hay que tener en cuenta al elegir una asignatura optativa en un determinado curso. Así mismo, interesa saber el **aula** en la que se imparten las clases de cada curso.

