

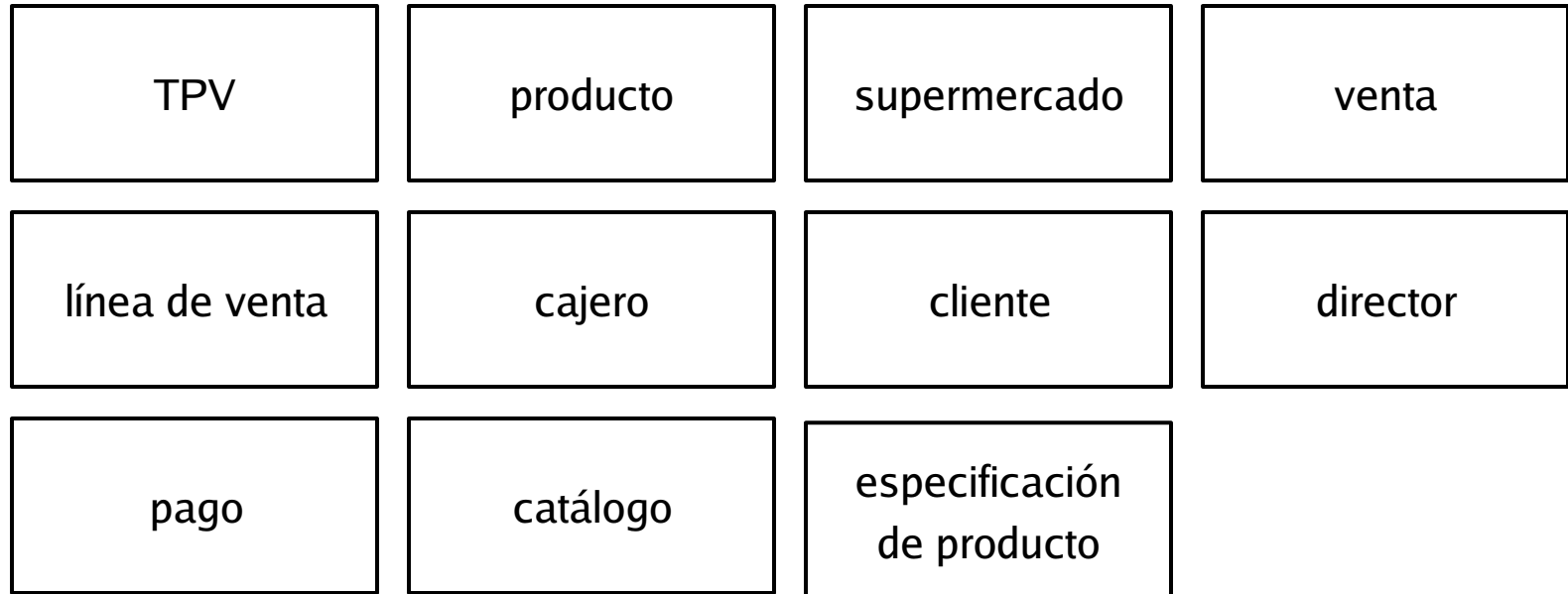
## Modelo de Dominio

- Representación de los conceptos (objetos) significativos en el dominio del problema
- Incluye:
  - Clases de objetos
  - Asociaciones entre clases de objetos
  - Atributos de las clases de objetos
- Objeto:
  - Entidad que existe en el mundo real
  - Tienen identidad y son distinguibles entre sí

## Clase de objeto

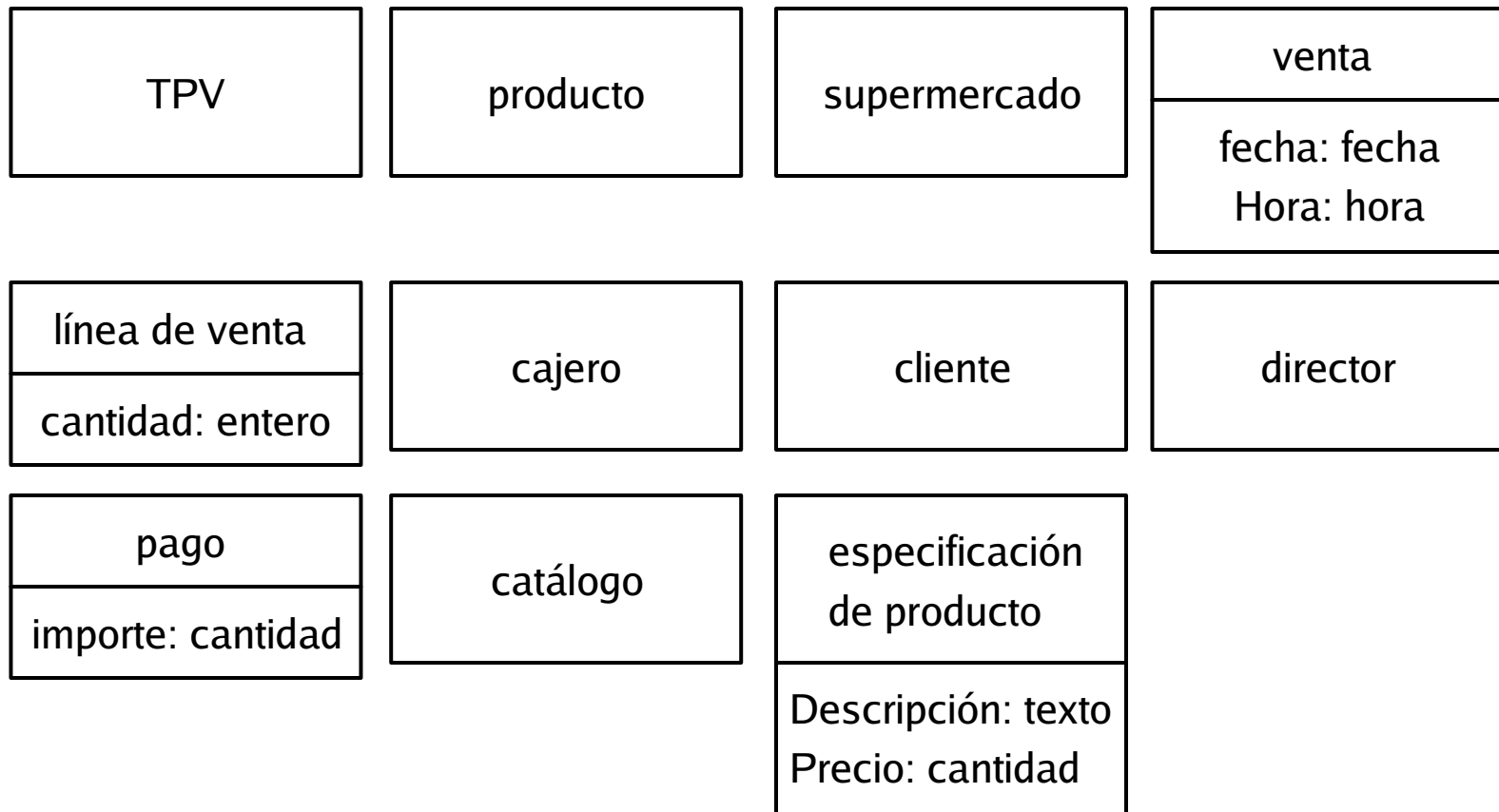
- **Agrupar un conjunto de objetos por tener:**
  - las mismas propiedades
  - un mismo comportamiento
  - la misma relación con otros objetos
  - una misma semántica
- **Abstracción:**
  - Ocultación de los detalles/características menos importantes para poder observar aspectos comunes
- **Los objetos de una clase tienen las mismas propiedades y los mismos patrones de comportamiento**

## Diagrama de clases



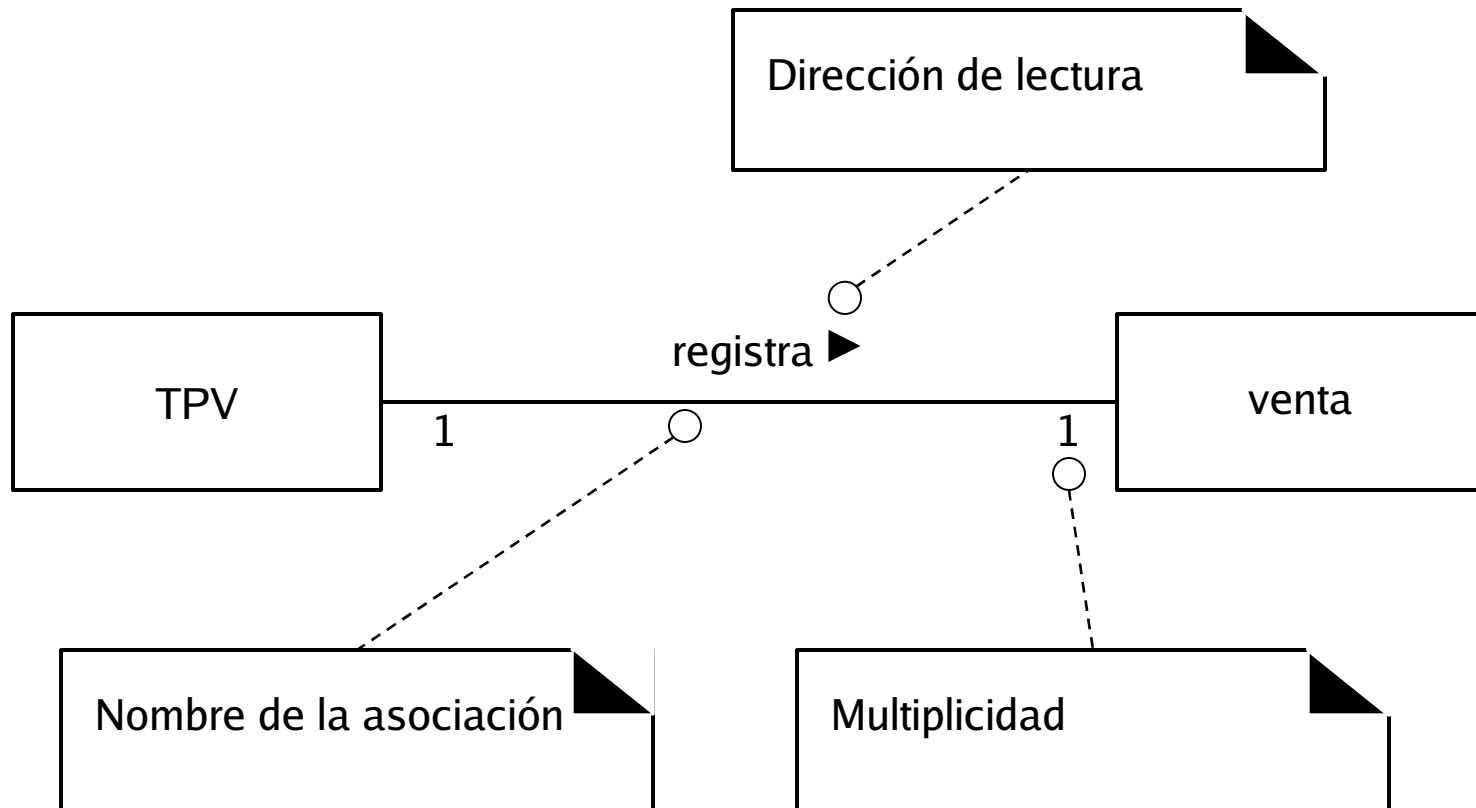
## Atributos

- Propiedades compartidas por los objetos de una clase



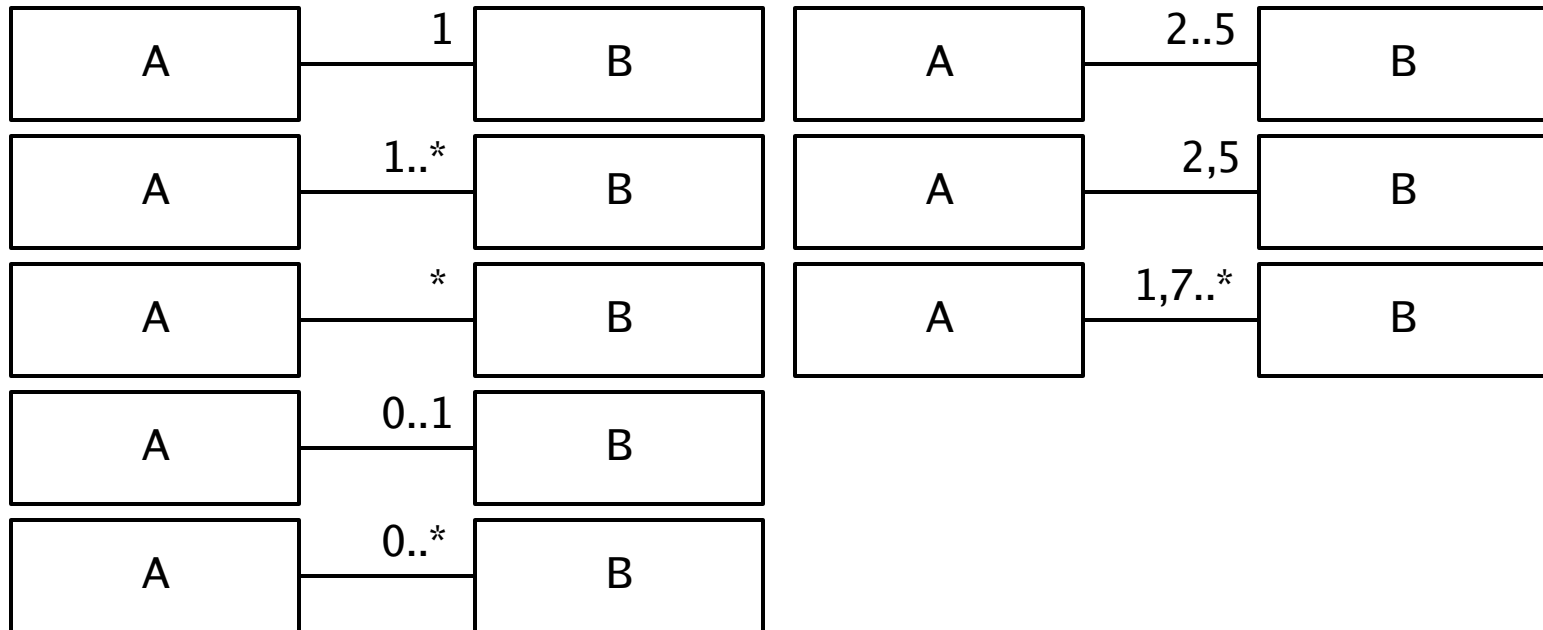
## Asociaciones

- Representan las relaciones entre dos o más objetos

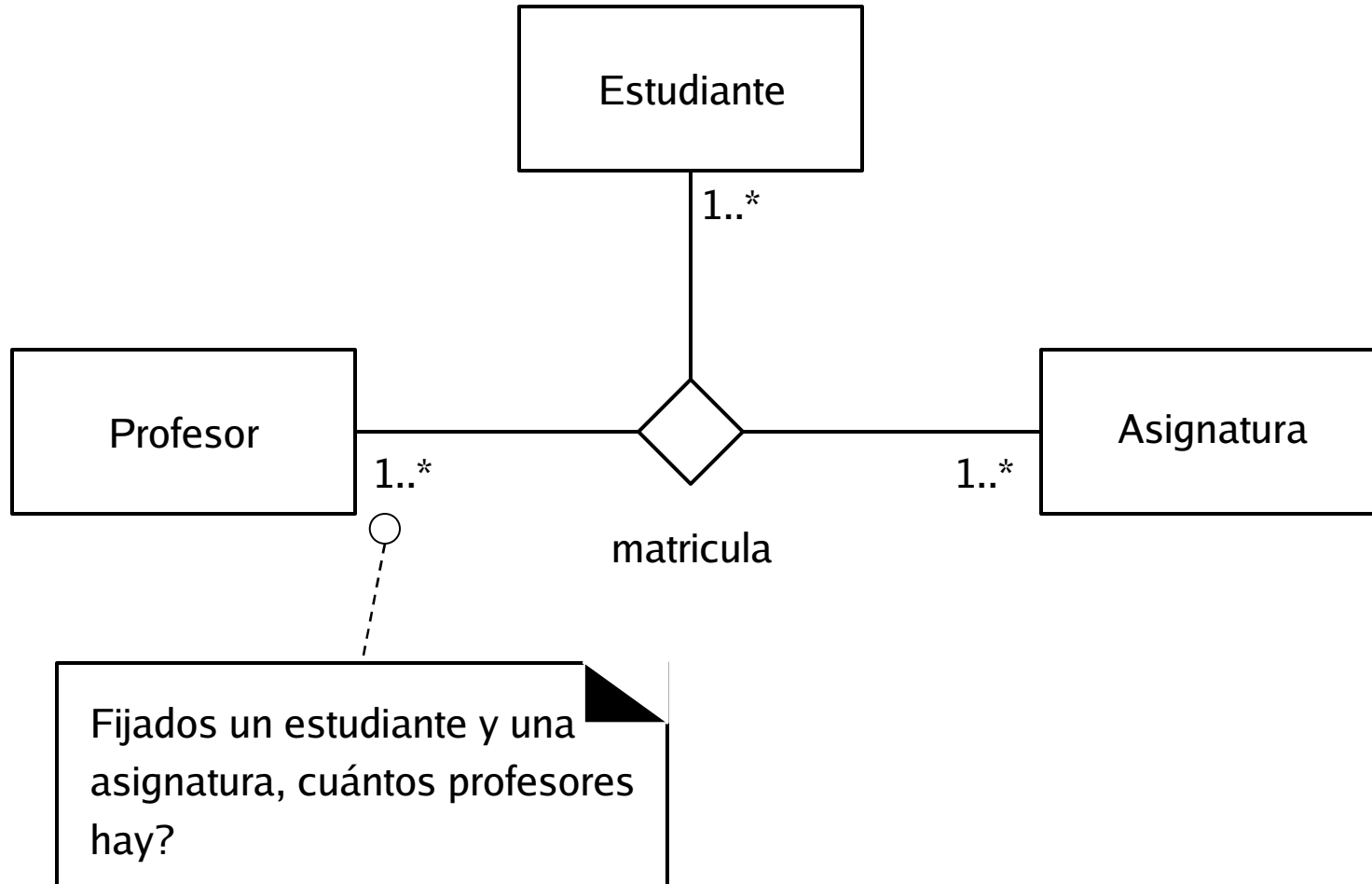


## Multiplicidad en las asociaciones

- Define cuantas instancias de una clase B pueden asociarse con una instancia de la clase A en un instante de tiempo determinado

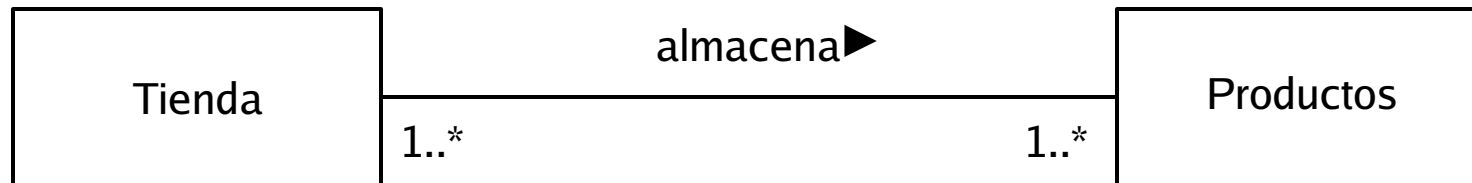


## Asociaciones de orden superior a dos



## Rol de las asociaciones

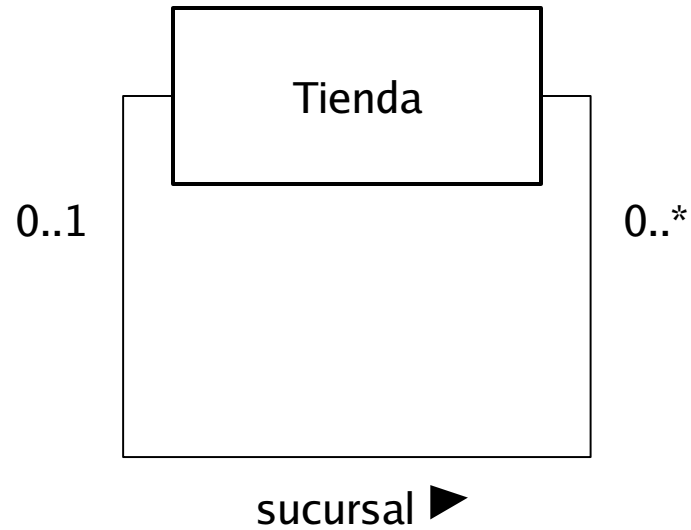
- Cada extremo de la asociación es un rol, que tiene como propiedades:
  - nombre
  - multiplicidad





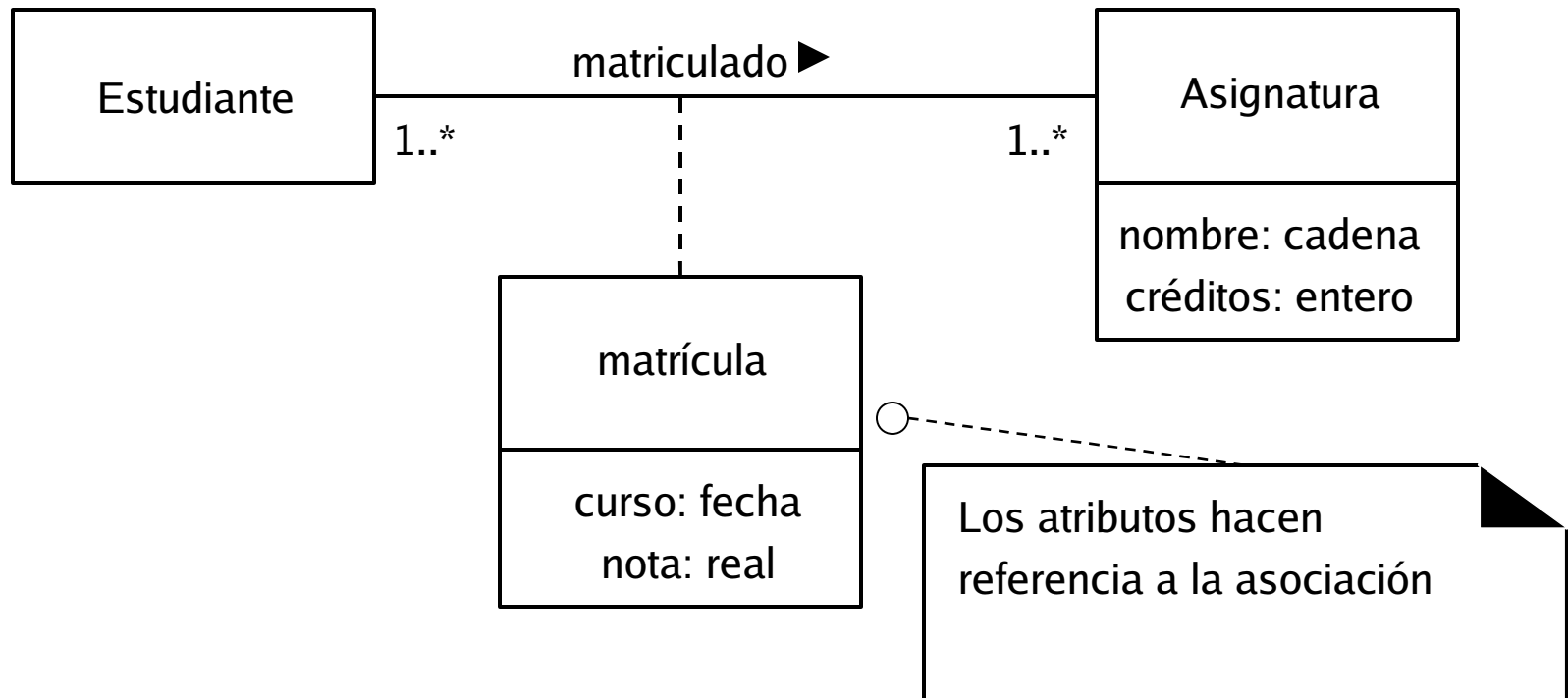
## Asociaciones recursivas

- Asociaciones con una misma clase de objetos



## Clase asociaciones

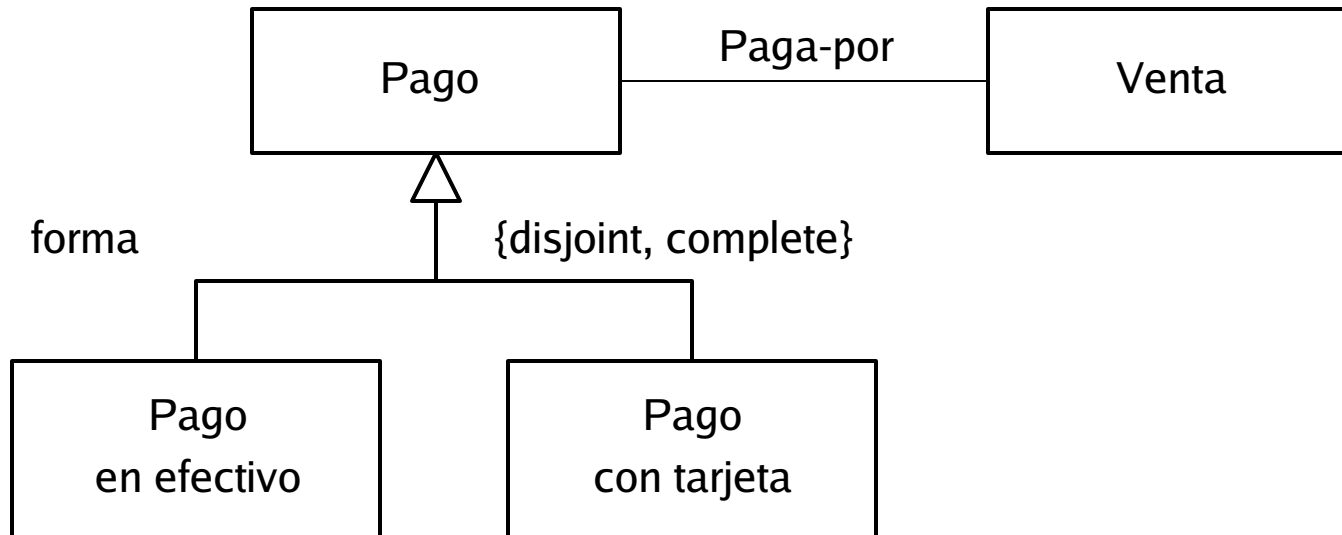
- Representa una asociación que puede verse como una clase



## Generalización/Especialización (1)

- Agrupan propiedades comunes entre los objetos definiendo relaciones clase/subclase
- Todos los miembros de la subclase son miembros de la clase y comparten la misma definición
  - atributos, asociaciones, operaciones
- Discriminador: nombre de la partición
- disjoint: un objeto sólo pertenece a una subclase
- overlapping: un objeto puede ser de más de una subclase
- complete: se detallan todas las subclases posibles
- incomplete: pueden haber más clases

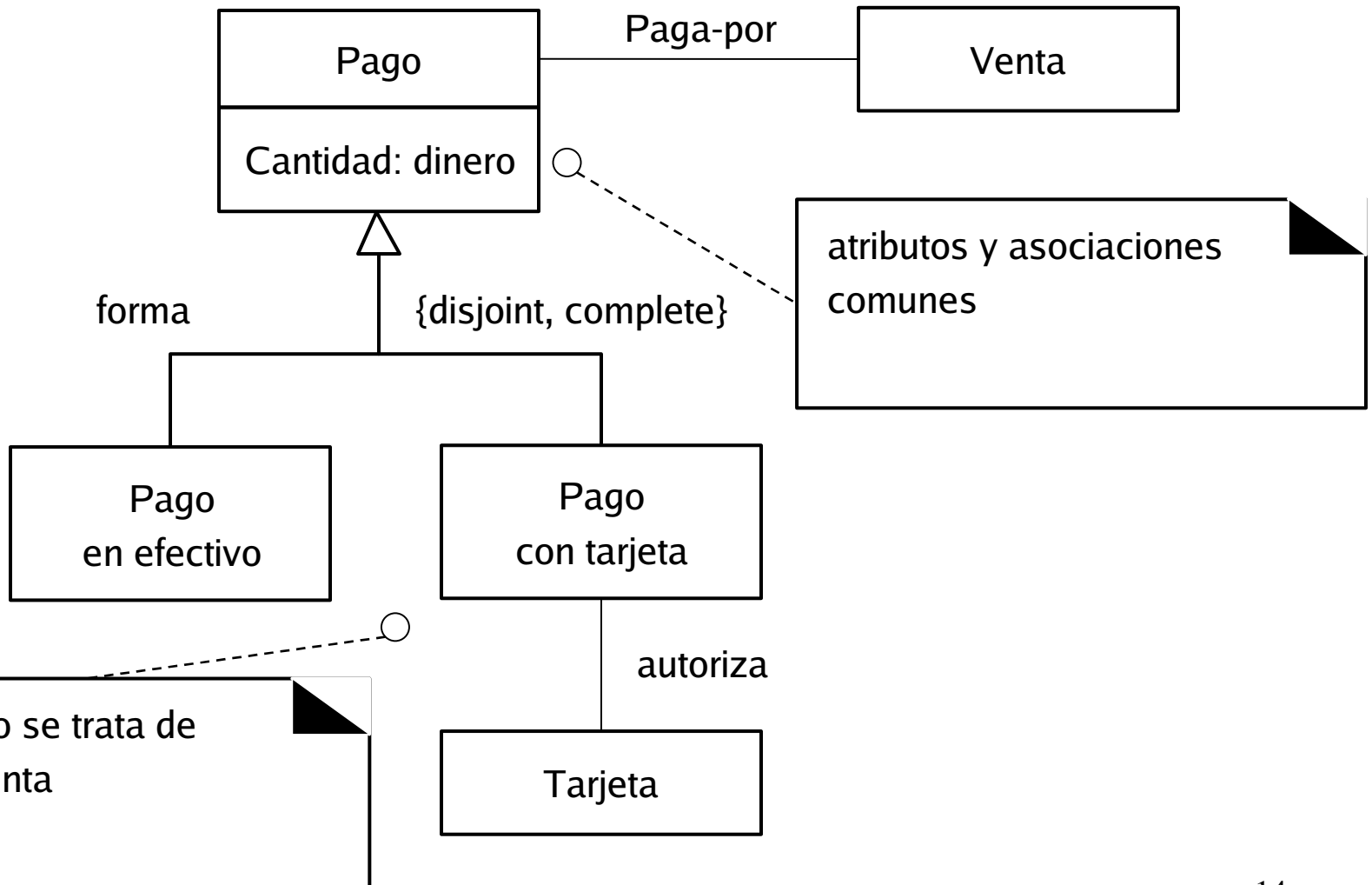
## Generalización/Especialización (2)



## Generalización/Especialización (3)

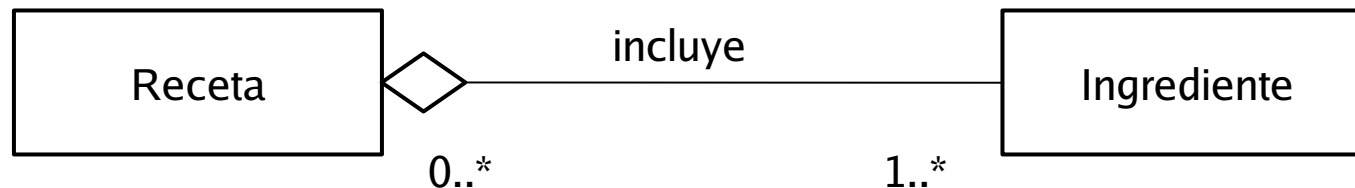
- **Motivos para crear subclases**
  - La subclase tiene atributos adicionales
  - La subclase tiene asociaciones adicionales
  - Se opera sobre la subclase de forma distinta al resto de subtipos
  - La subclase se comporta de forma diferente a la superclase o a las otras subclases
- **Motivos para crear superclases**
  - Las subclases son variaciones de un mismo concepto
  - Las subclases tienen atributos factorizables (comunes)
  - Las subclases tienen asociaciones factorizables

## Generalización/Especialización (4)



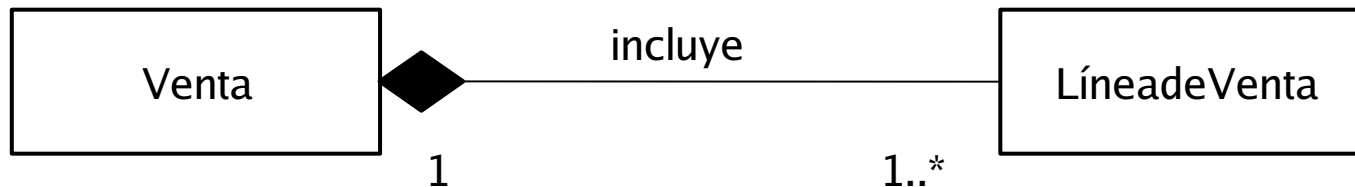
## Agregación

- Tipo de asociación que permite modelar relaciones parte-todo
- Las cadenas de agregaciones no pueden formar ciclos
- La multiplicidad del extremo del compuesto puede ser más de una



## Composición

- Tipo de agregación donde:
  - La multiplicidad del extremo del compuesto no puede ser mayor a uno
  - Ninguna “parte” puede existir sin el “todo”



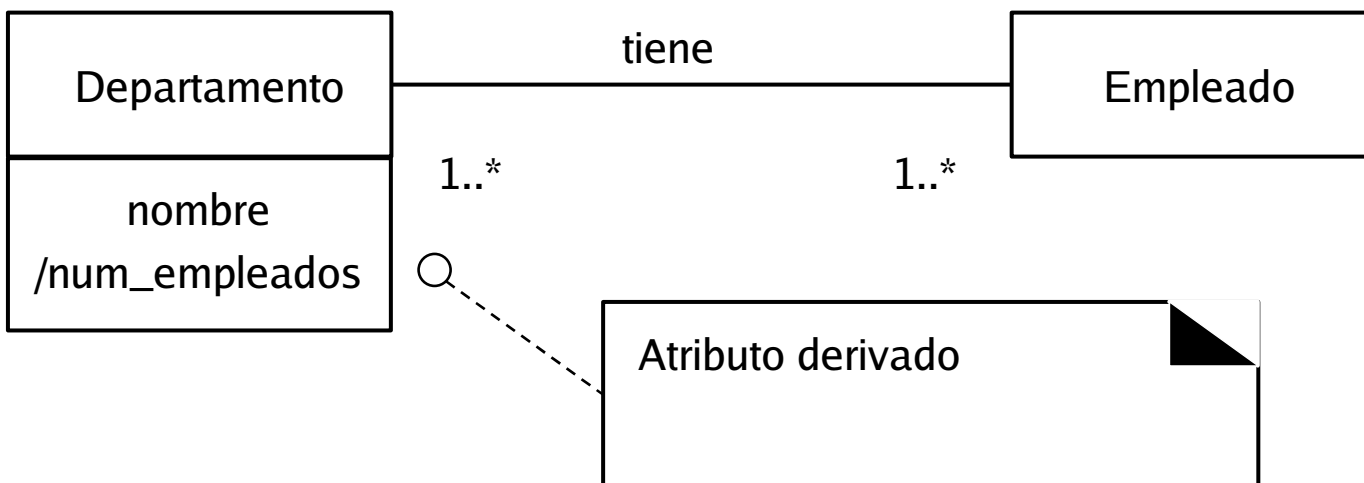


## Agregación/Composición

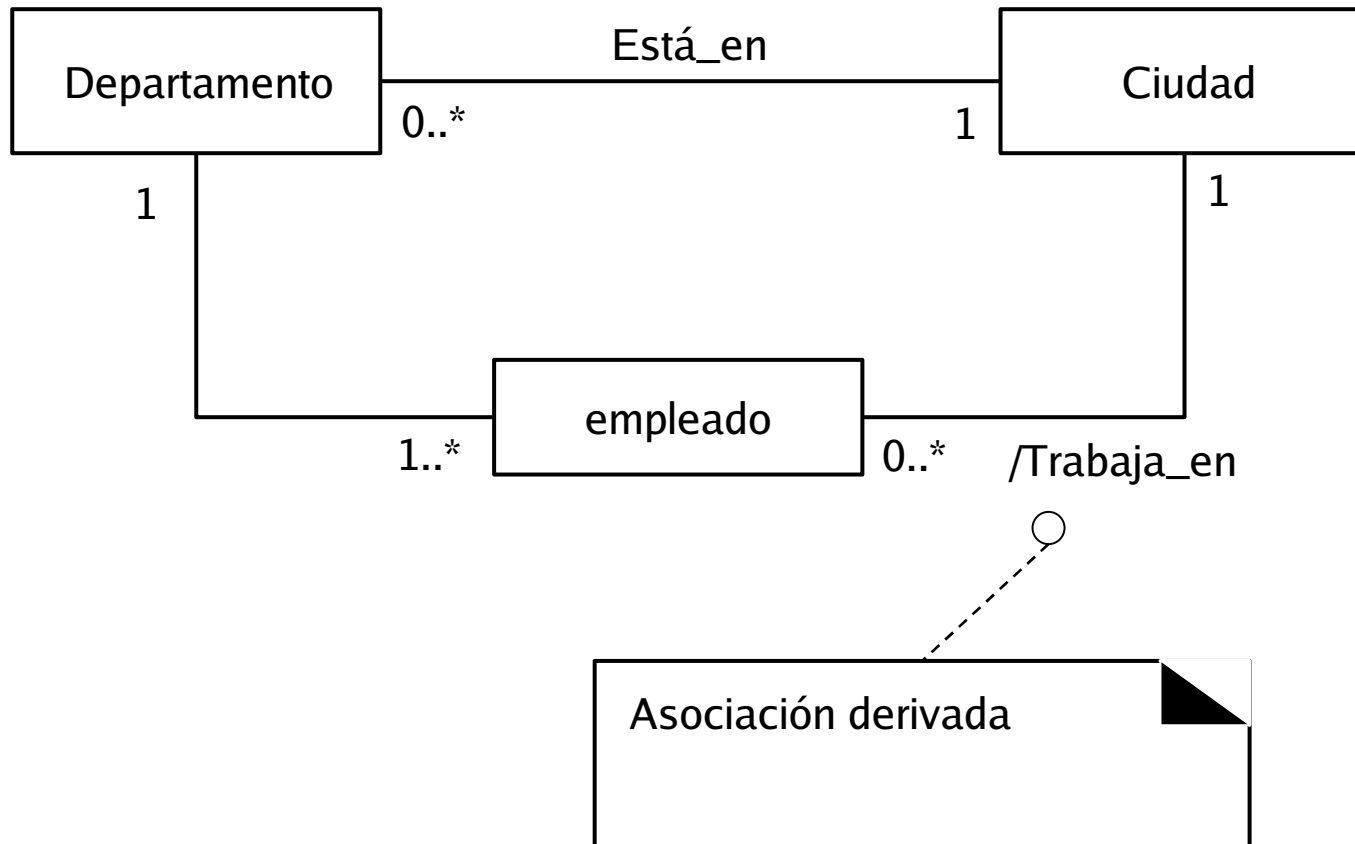
- **Motivos para crear agregaciones o composiciones:**
  - Existe una evidente conexión física parte-todo
  - Algunas propiedades del compuesto se difunden hacia las partes
  - Algunas operaciones aplicadas al compuesto se propagan a las partes: p.e. destrucción
  
- **Motivos para crear composiciones:**
  - La duración de la parte es dependiente de la que tiene el compuesto: la parte muestra una dependencia de crear-eliminar respecto al todo

## Información derivada (1)

- Un atributo (o asociación) es derivado si puede calcularse a partir de otra información presente
- Se añade para mejorar la claridad del modelo
- Debe ir acompañada de una restricción “constraint” o regla de derivación para especificar cómo se obtiene

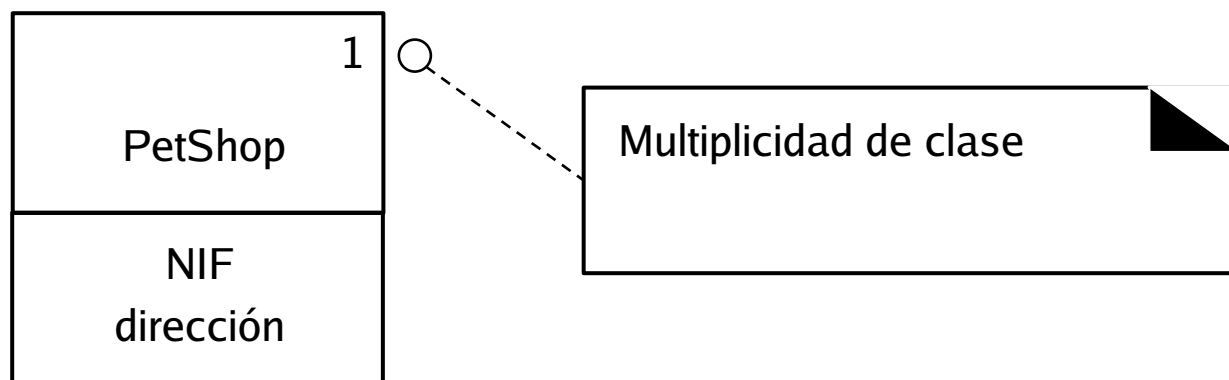


## Información derivada (2)



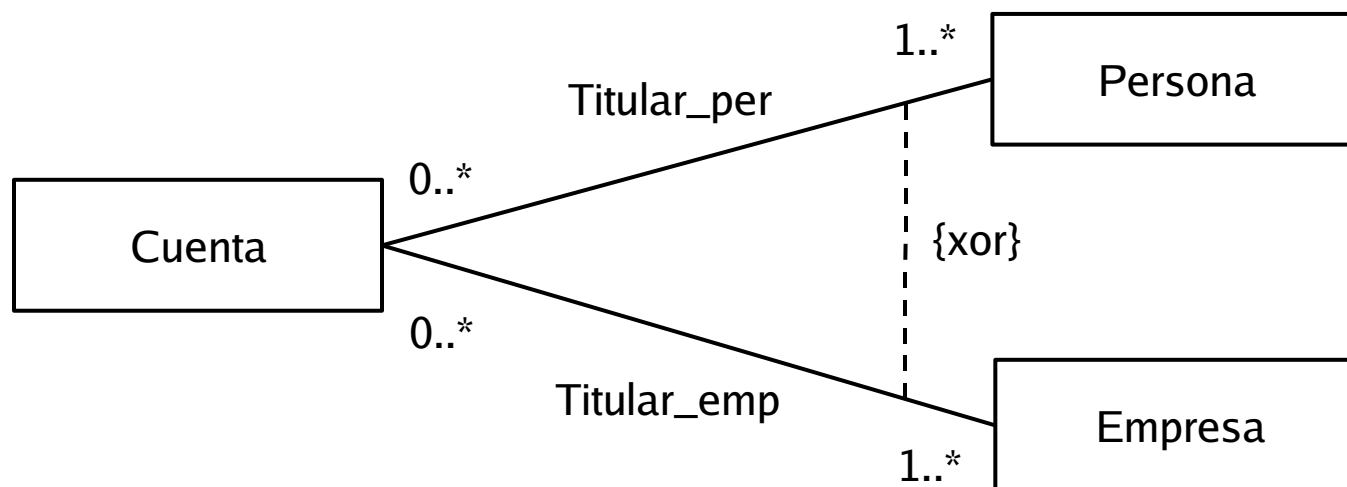
## Multiplicidad de clase

- Establece el rango de posibles instancias de una clase. Por defecto, es indefinida
- En algunos casos, es útil establecer una multiplicidad finita, especialmente en clases que puedan tener una sola instancia (llamada “singleton”)



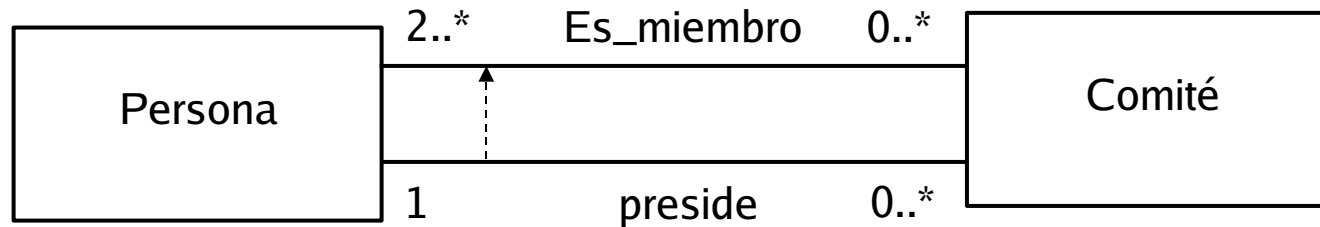
## Otras restricciones entre asociaciones (1)

- Además de la multiplicidad, es posible expresar otras restricciones sobre las asociaciones: xor, subset
- xor
  - Une varias asociaciones a una misma clase básica
  - Una instancia de la clase básica participa exactamente de una de las asociaciones unidas por xor



## Otras restricciones entre asociaciones (2)

- subset
  - Indica que una asociación es un subconjunto de otra



## Modificabilidad

- La modificabilidad indica si los valores de un atributo o los extremos de una asociación pueden cambiar o no: changeable, frozen, addOnly

