Diagramas de clases

1. Introducción
2. Objetos y clases
3. Interfaces
4. Asociaciones y dependencias
5. Generalización
6. Agregación y composición
7. Realización
8. Mecanismos de extensión

INTRODUCCIÓN

Los diagramas de clases

- Son las representaciones más usadas en el modelado de sistemas orientados a objetos
- Son representaciones gráficas de la parte estática de un sistema
- Describen la estructura de un sistema mostrando un conjunto de clases, interfaces y colaboraciones, así como las relaciones entre esos elementos
- Permiten representar modelos en distintas etapas de desarrollo
  - **Modelo de análisis**
    Diagramas de clases conceptuales
  - **Modelo de diseño**
    Diagramas de clases de diseño
  - **Modelo de implementación**
    Diagramas de clases de implementación

INTRODUCCIÓN

Los diagramas de clases están compuestos por

- Elementos estructurales
  - Clases e interfaces
- Relaciones entre esos elementos
  - Asociación
  - Dependencia
  - Generalización
  - Agregación y composición
  - Realización
- Notas y estereotipos

OBJETOS Y CLASES

¿Qué es un objeto?

Una entidad discreta con un límite bien definido que encapsula el estado y comportamiento; una instancia de una clase

Manual de Referencia de UML

Propiedades de los objetos

- **Identidad**
  Se refiere a la existencia única de un objeto en tiempo y espacio
- **Estado**
  Esta determinado por los valores de atributos y las relaciones con otros objetos en punto particular del tiempo
- **Comportamiento**
  Tareas u operaciones que los objetos pueden hacer por nosotros

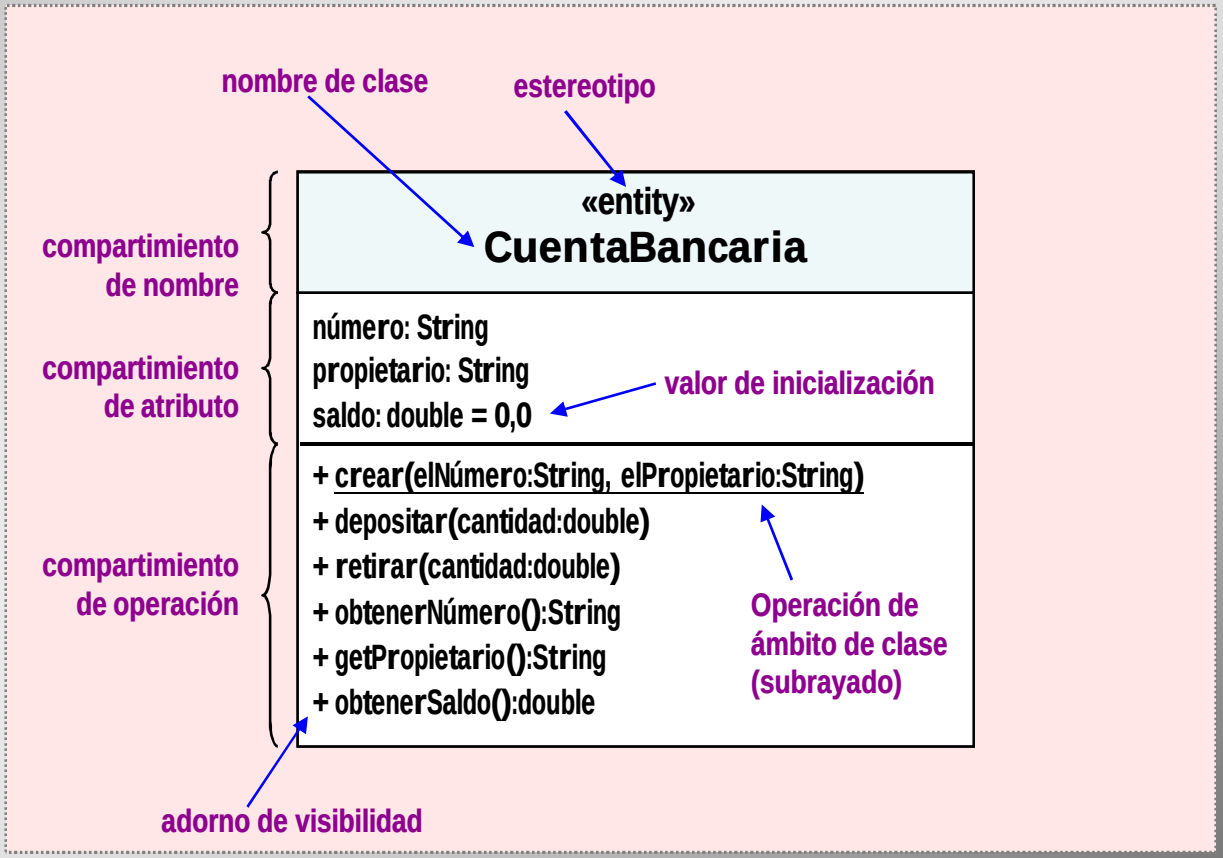
OBJETOS Y CLASES

¿Qué es una clase?

Un descriptor para un conjunto de objetos que comparten los mismos atributos, operaciones, métodos, relaciones y comportamiento

Manual de Referencia de UML

Notación



OBJETOS Y CLASES

- **Nombre de clase**
  Expresión nominal extraída del vocabulario del problema a modelar; debe comenzar con mayúscula

- **Estereotipo**
  Permite establecer la categoría de la clase

- **Atributo**
  Propiedad del elemento que se está modelando

  `[visibilidad]nombre[multiplicidad][:tipo][= valor inicial]`

- **Valor de inicialización**
  Permite especificar el valor que tomará un atributo cuando se instancia un objeto de la clase

- **Operación**
  Abstracción de un servicio que pueden prestar los objetos de la clase

  `[visibilidad]nombre[(lista de parémtros)][:tipo retorno`

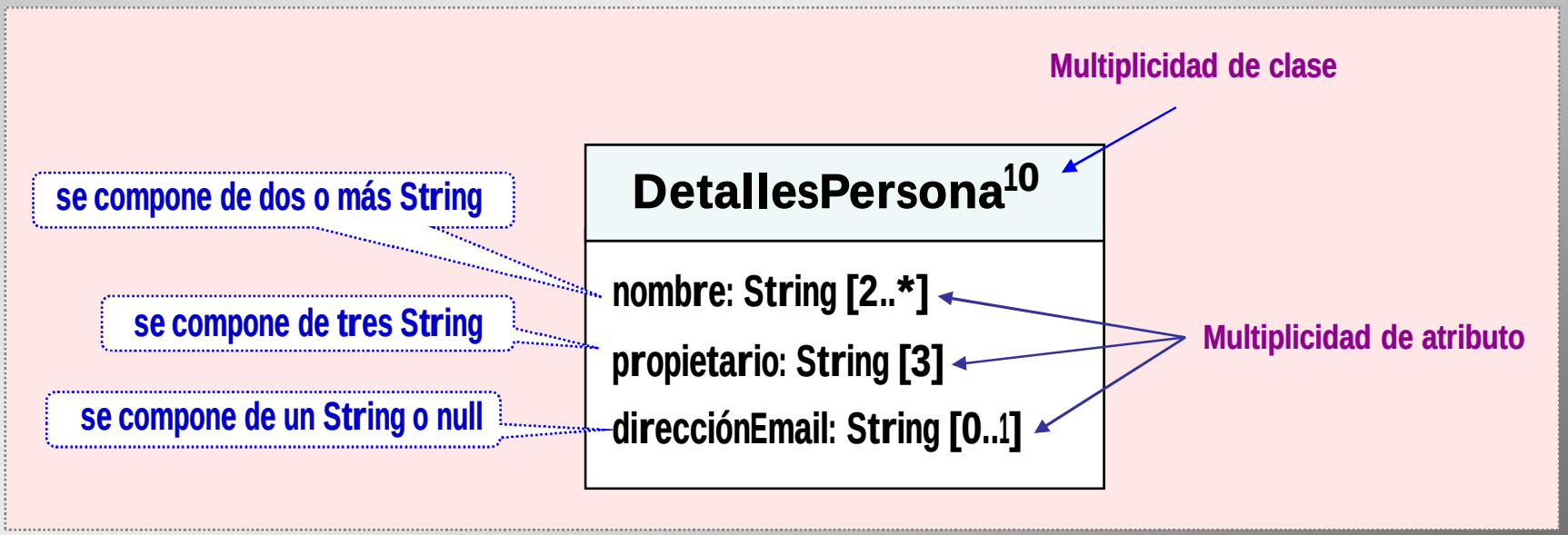
OBJETOS Y CLASES

- Visibilidad de atributos y operaciones
  - **Pública (+)**
    Cualquier clase puede usar el atributo u operación
  - **Paquete (~)**
    Sólo el paquete en el que está definida la clase puede usar el atributo u operación (valor por defecto)
  - **Protegida (#)**
    Cualquier clase hija puede usar el atributo u operación
  - **Privada (-)**
    Sólo la propia clase puede usar el atributo u operación
- Ámbito o alcance de atributos y operaciones
  - **De instancia**
    Cada instancia posee su propio atributo u operación
  - **De clase**
    Existe un único valor para todas las instancias de la clase; se representa subrayando el nombre

OBJETOS Y CLASES

- Multiplicidad
  - De clase
    Número de instancias que puede tener una clase
  - De atributo
    Número de instancias de un atributo

Ejemplo

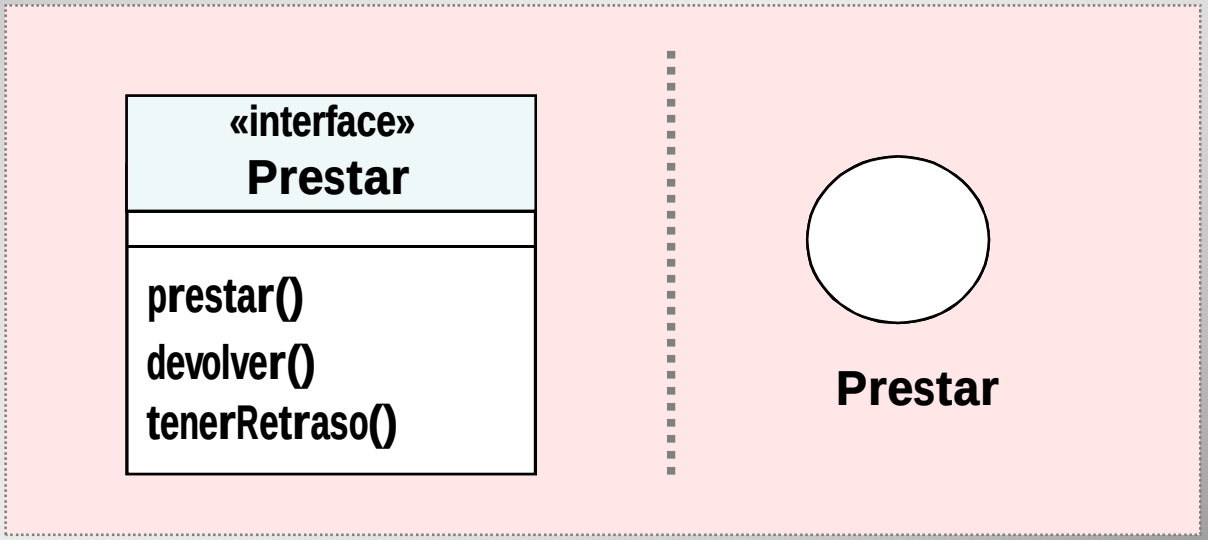


INTERFACES

Especifican un conjunto de características públicas

- Sirven para separar la especificación de la funcionalidad de su implementación
- No se pueden instanciar
- Compuestas por un nombre y la definición de un conjunto de operaciones

Notación



ASOCIACIONES Y DEPENDENCIAS

¿Qué es una relación?

Una conexión semántica (significativa) entre elementos de modelado; es la forma de conectar elementos

¿Qué es una enlace?

Una conexión semántica entre objetos que permite enviar mensajes de un objeto a otro

¿Qué es una asociación?

Una relación entre clases; las instancias de una asociación son los enlaces

Adornos de una asociación

Propiedades que modelan su semántica

- Nombre de asociación
- Navegabilidad
- Nombre de rol
- Multiplicidad
- Visibilidad

ASOCIACIONES Y DEPENDENCIAS

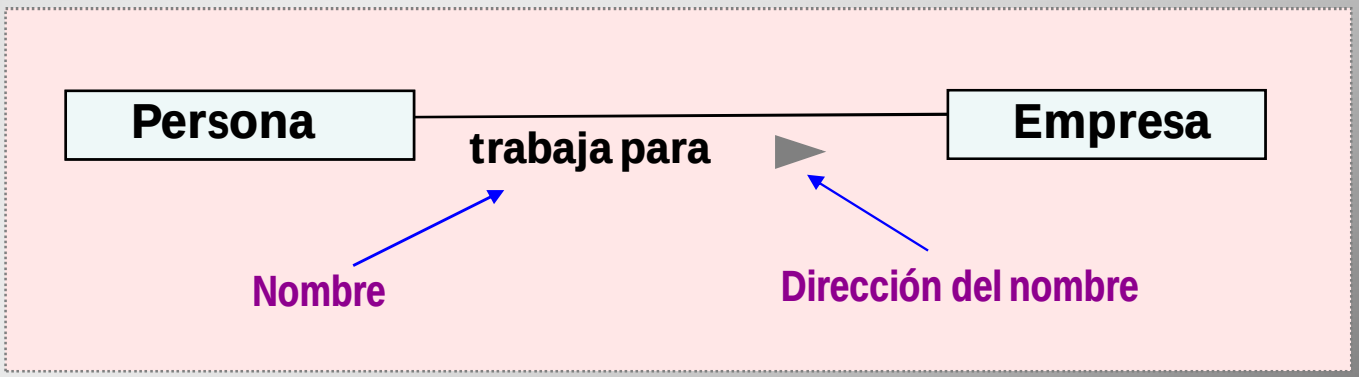
Nombre de la asociación

- Identifica la asociación que se establece entre dos clases
- Deberían ser frases verbales

  Indican una acción que el objeto fuente realiza sobre el objeto destino

Para su representación

- El nombre de asociación es opcional
- El nombre de asociación se escribe en minúsculas
- Puede llevar una flecha que indica el sentido en el que se debería leer la asociación (opcional)



ASOCIACIONES Y DEPENDENCIAS

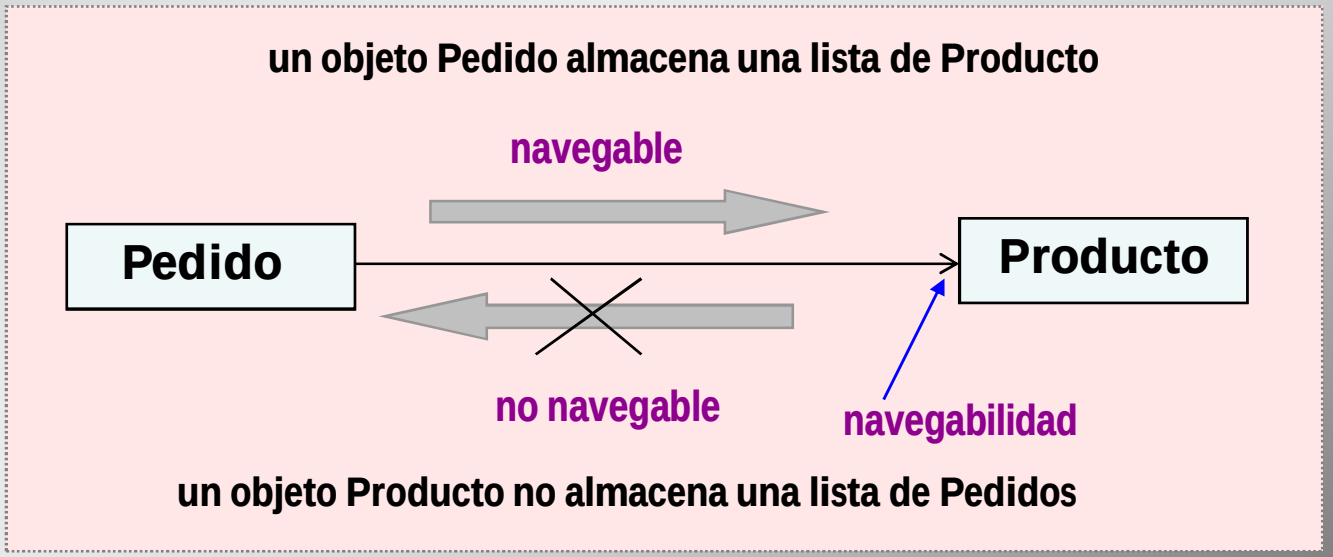
Navegabilidad

- Muestra que es posible pasar desde objetos de la clase fuente a objetos de la clase destino

"Mensajes que solamente se pueden enviar en la dirección de la flecha"

Para su representación

- La navegabilidad se muestra con una punta de flecha en un extremo de la asociación
- Cuando la asociación no lleva punta de flecha indica que la navegabilidad es en los dos sentidos

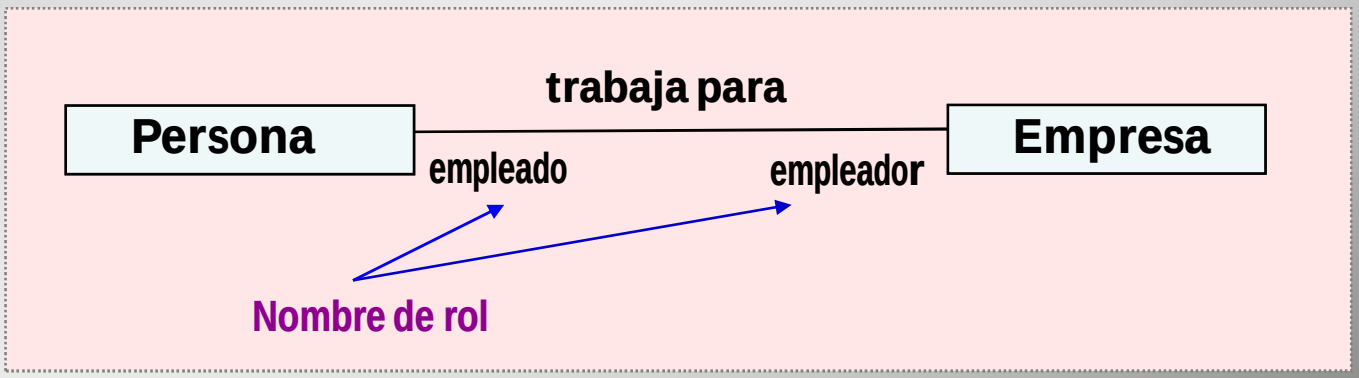


ASOCIACIONES Y DEPENDENCIAS

Nombre de rol

- El rol indica el papel que la clase de un extremo de la asociación juega con respecto a la clase del otro extremo
- Debería ser un nombre o frase nominal

  Indica el rol que pueden desempeñar los objetos
- Es una característica de las asociaciones

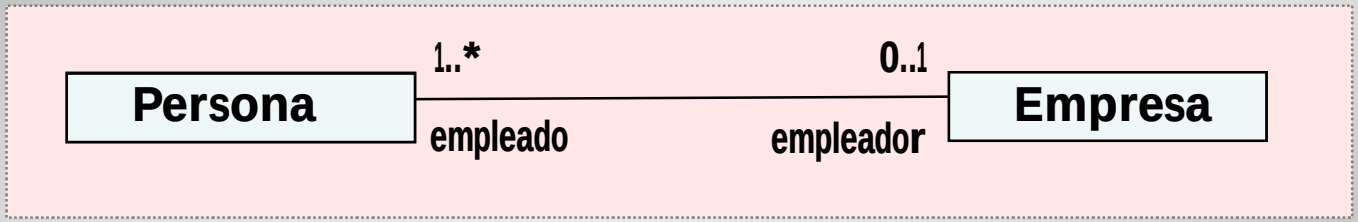


ASOCIACIONES Y DEPENDENCIAS

Multiplicidad

- Indica cuánto objetos de un extremo de la asociación pueden conectarse con un objeto del otro extremo

Restringe el número de objetos de una clase que se pueden implicar en una relación determinada en cualquier momento en el tiempo

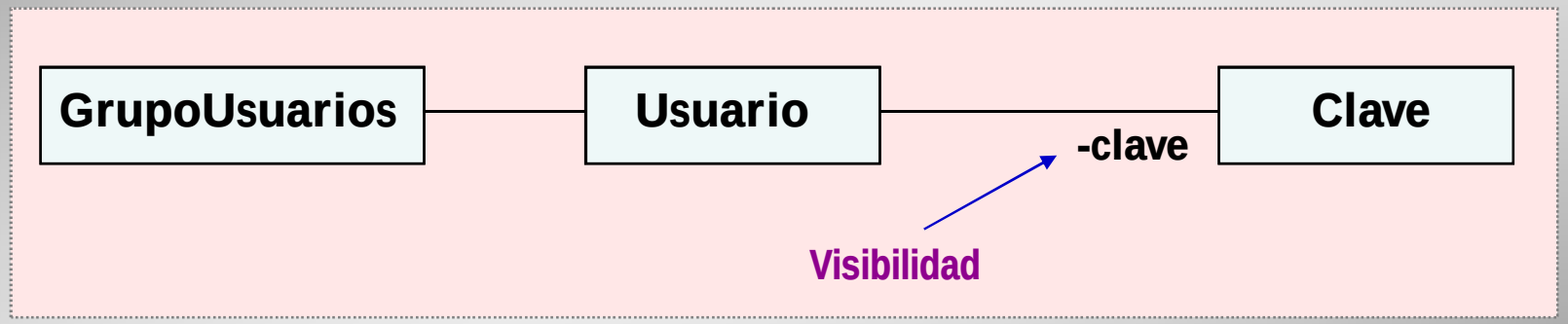


Expresión de la multiplicidad (mínimo..máximo)			
1	Exactamente uno	*	Cero o muchos
0..1	Cero o uno	0..*	Cero o muchos
m..n	Desde m hasta n	1..*	Uno o muchos
0..1, 3..4, 6..*	Cualquier número excepto 2 y 5		

ASOCIACIONES Y DEPENDENCIAS

Visibilidad

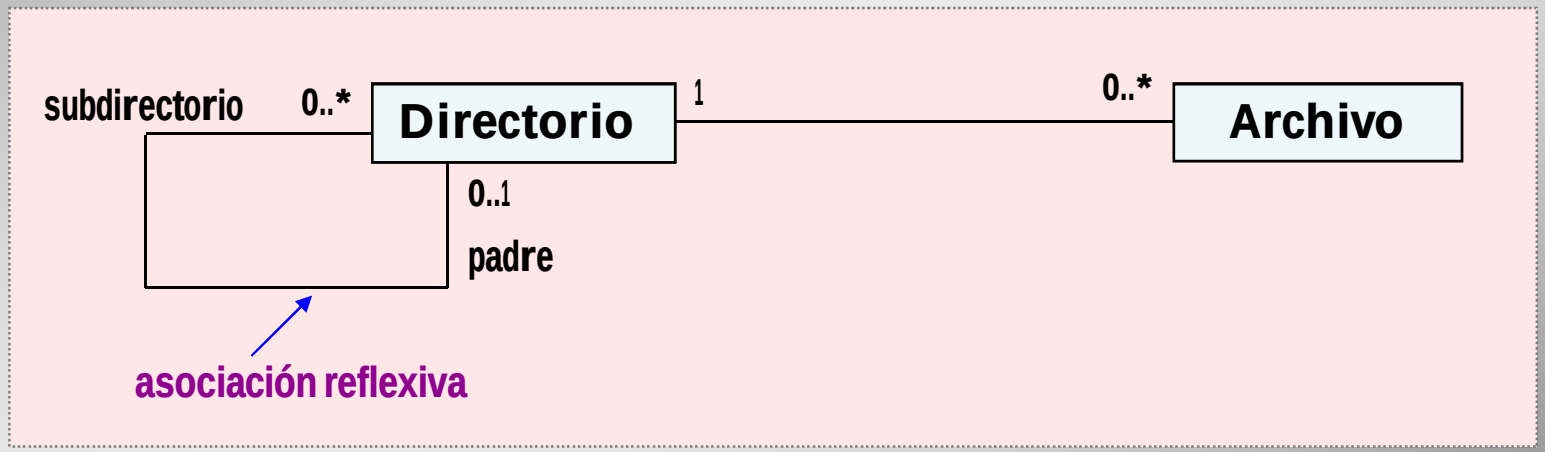
- Posibilita el acceso desde los objetos de una clase a los objetos de otras clases a través de asociaciones (+. -)



ASOCIACIONES Y DEPENDENCIAS

Asociaciones reflexivas

Asociación en la que los objetos de una clase tienen enlaces con objetos de la misma clase

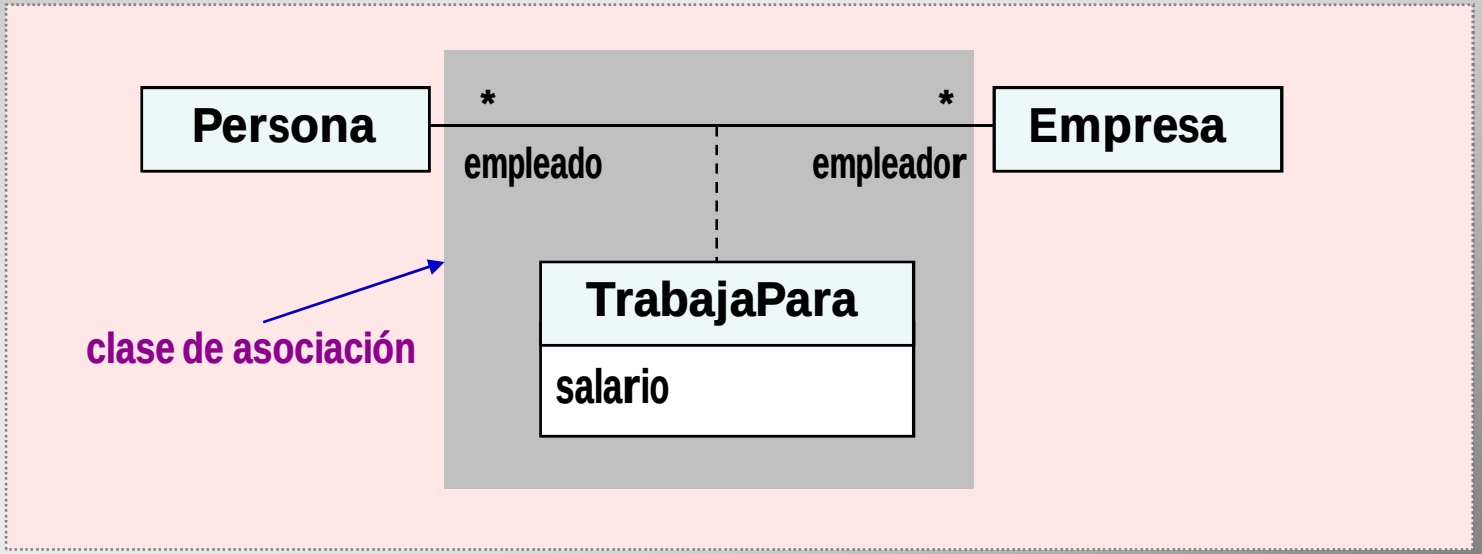


ASOCIACIONES Y DEPENDENCIAS

Clases de asociación

Modelan una relación muchos a muchos entre dos clases en la que existen atributos que no se pueden acomodar fácilmente en ninguna de las clases

La clase de asociación es la línea de asociación (incluidos los nombres de rol y las multiplicidades), la línea de puntos y el cuadro de la clase en el extremo de la línea de puntos

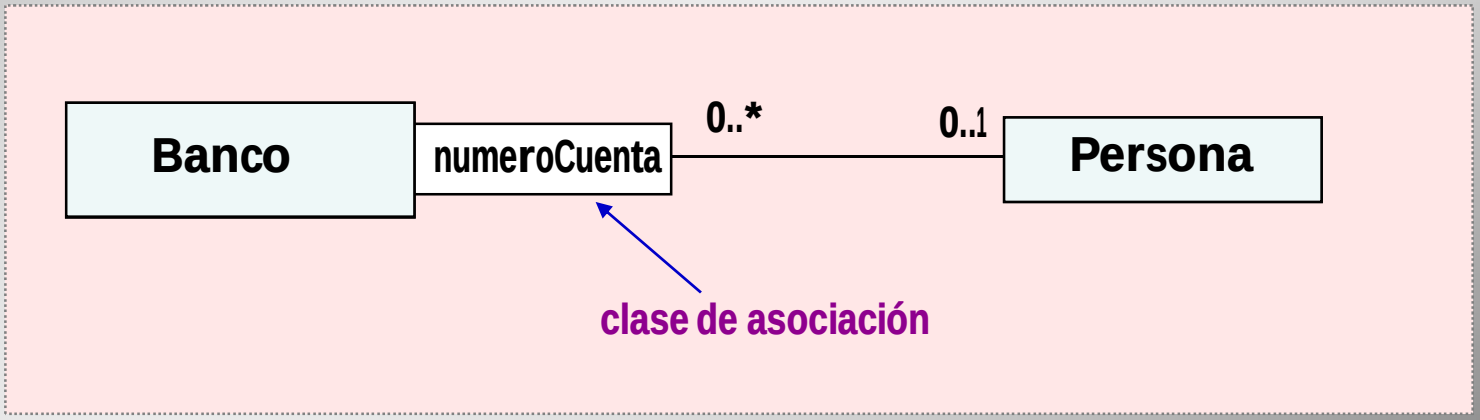


ASOCIACIONES Y DEPENDENCIAS

Asociaciones cualificadas

El **calificador** (o cualificador) es un atributo de alguna de las clases de la asociación que pasa a ser un atributo asociado a la clase del otro extremo

- Estas asociaciones se pueden usar para reducir una asociación n-a-muchos a una asociación n-a-1
- La cualificación reduce la multiplicidad del extremo opuesto al que cualifica

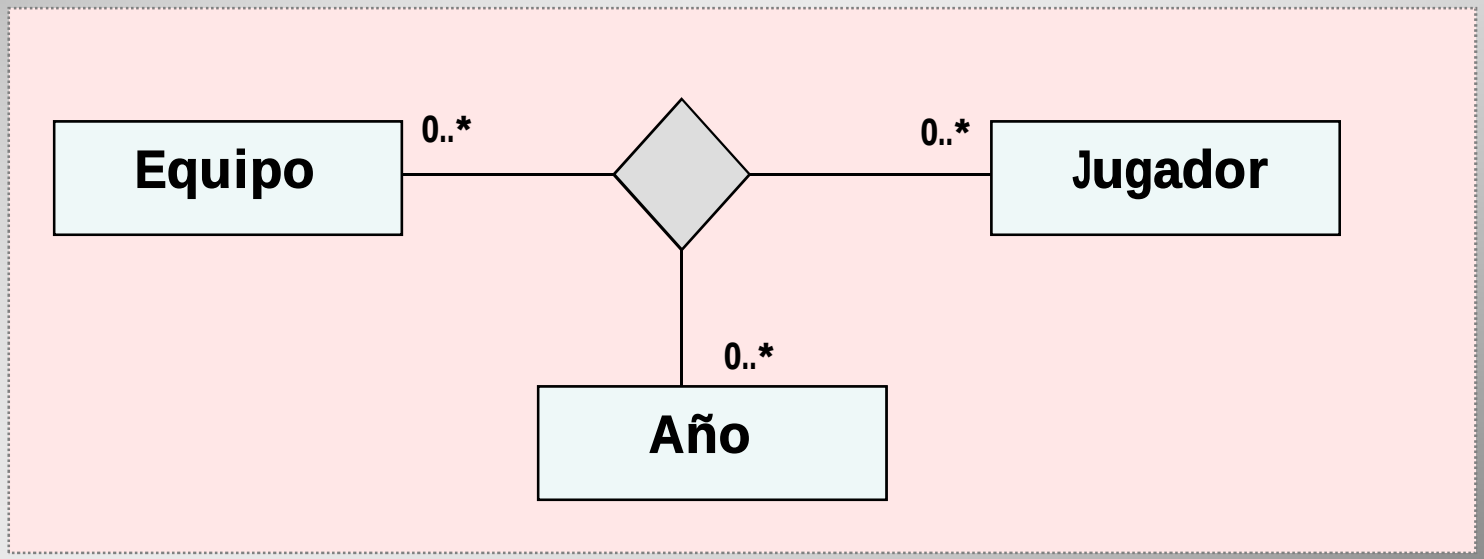


ASOCIACIONES Y DEPENDENCIAS

Asociaciones n-arias

Asociaciones que se establecen entre tres o más clases

Cada instancia de la asociación es una n-tupla de valores, uno por cada una de las clases que componen la asociación

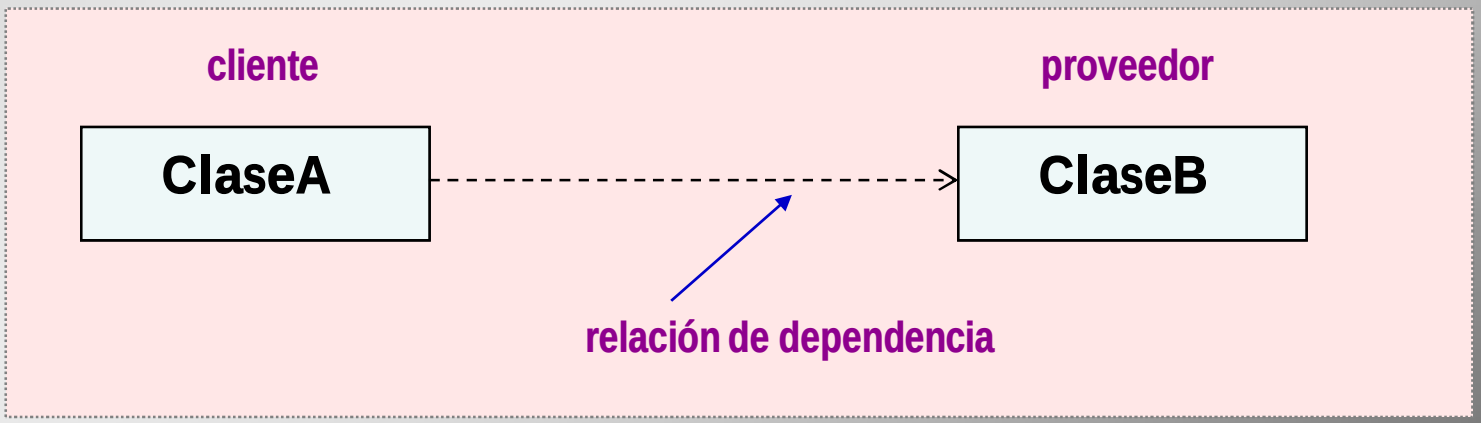


ASOCIACIONES Y DEPENDENCIAS

¿Qué es una dependencia?

Es una relación semántica de uso entre dos o más elementos de modelo en la que un cambio en un elemento (el proveedor) puede causar cambios en el elemento dependiente (el cliente)

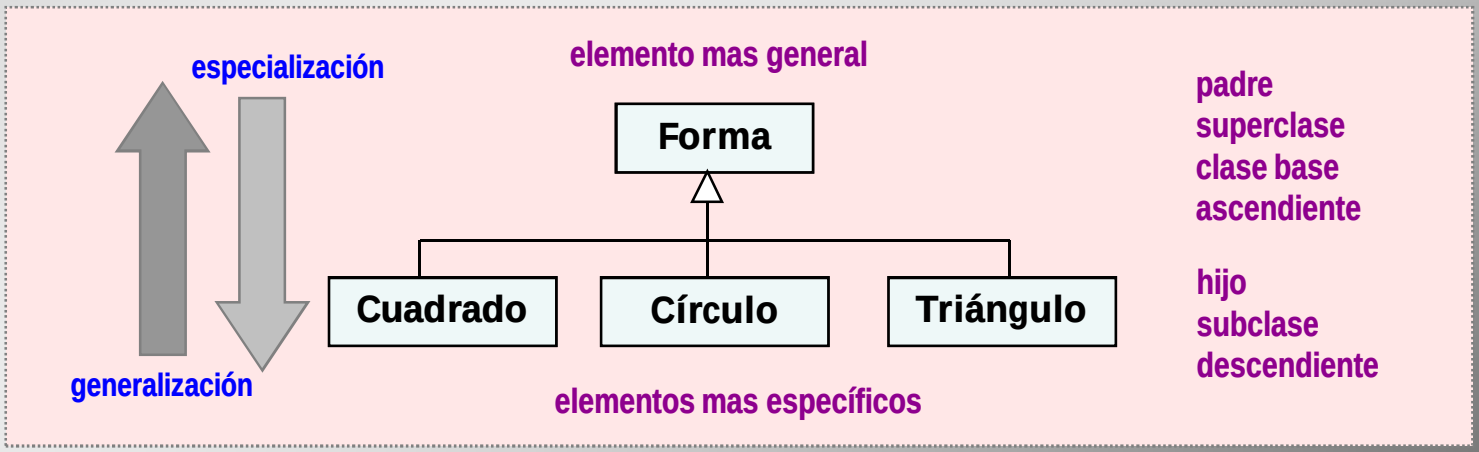
- La dependencias no necesitan tener un nombre
- Pueden ocurrir dependencias entre
  - Clases
  - Paquetes
  - Objetos y clases



GENERALIZACIÓN

Es una relación (“es un”) entre un elemento general (padre o superclase) y un elemento más específico de éste (hijo o subclase)

- Las subclases heredan
  - Atributos, operaciones, relaciones y restricciones de las superclases
- Las subclases pueden
  - Añadir nuevas características
  - Redefinir la implementación de las operaciones heredadas
- Superclases y subclases obedecen el principio de sustitución
  - Se puede utilizar el elemento más específico en cualquier lugar que se espere al elemento más general

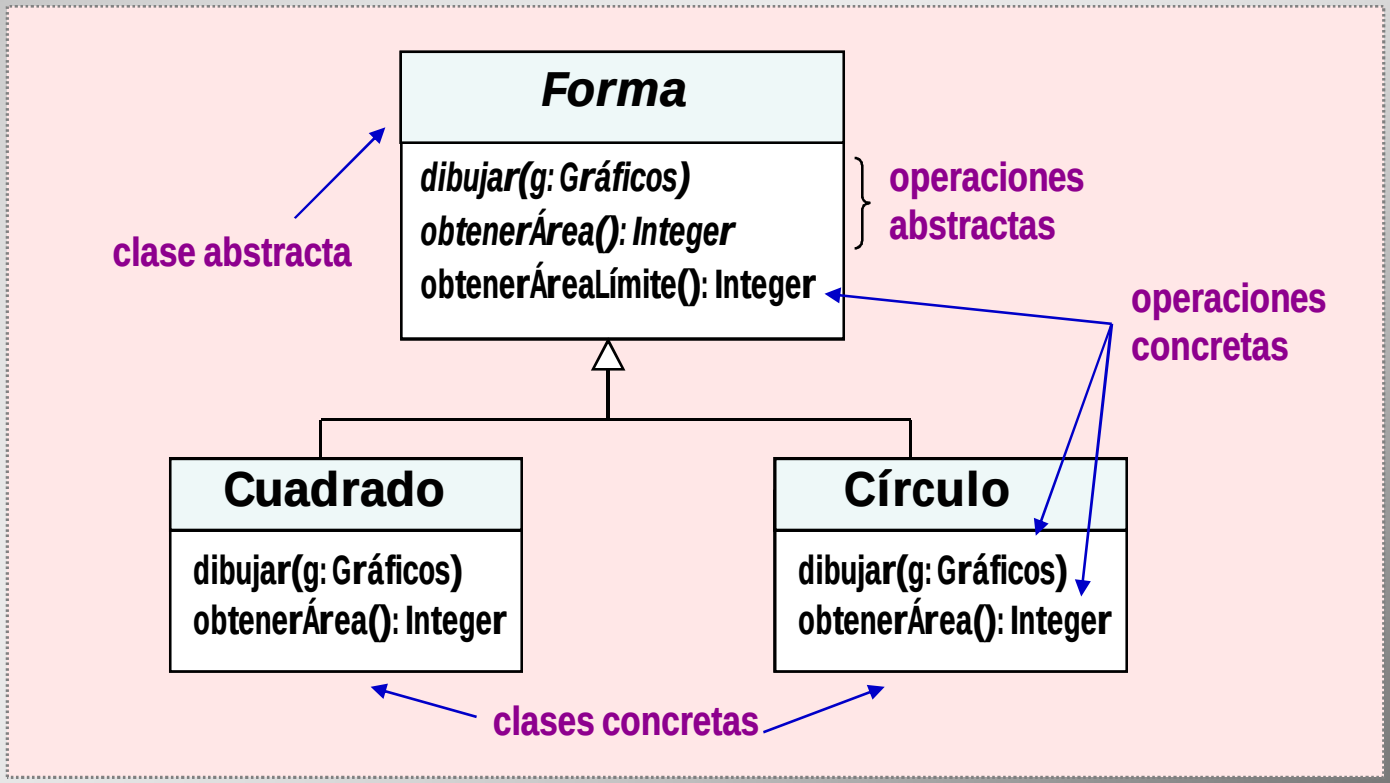


GENERALIZACIÓN

Clases abstractas

Clases no instanciables que poseen al menos un método abstracto (sin implementación, es decir, aún no definido), que se implementan a través de especializaciones en subclases

Se especifican escribiendo el nombre en cursiva



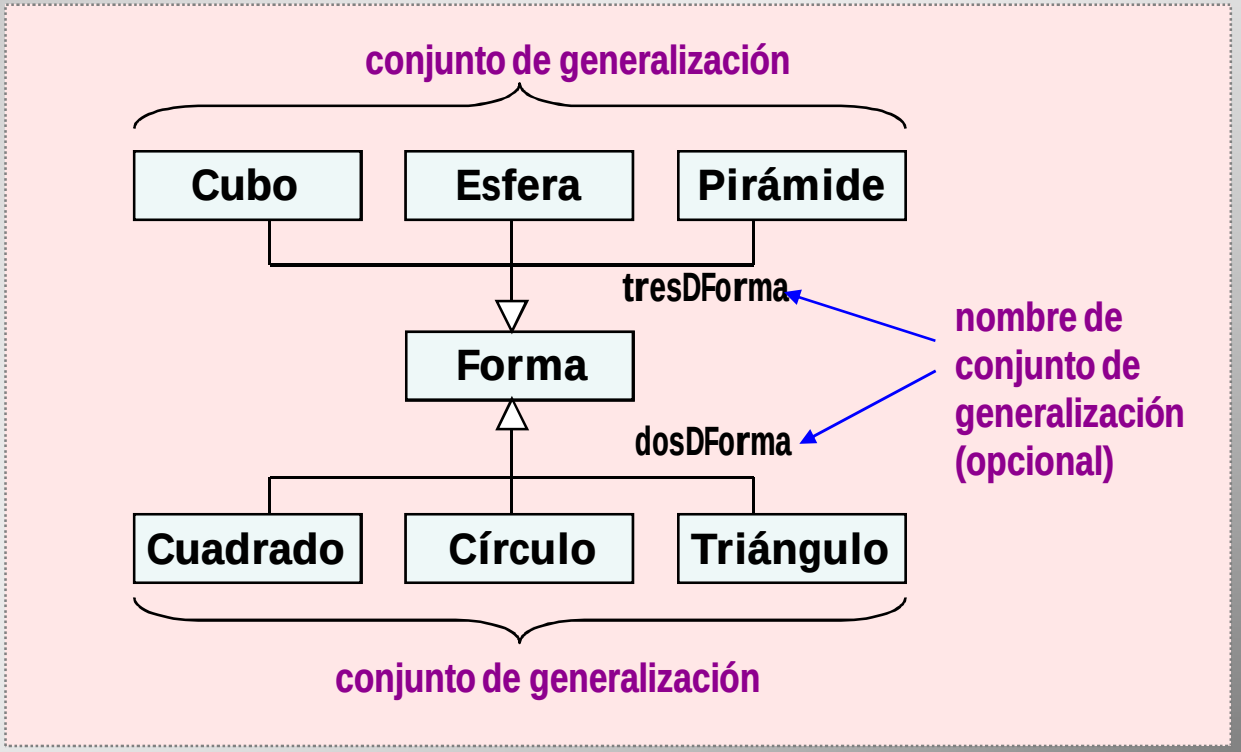
GENERALIZACIÓN

Conjuntos de generalización

Agrupan subclases de acuerdo a una regla particular, o base de especialización

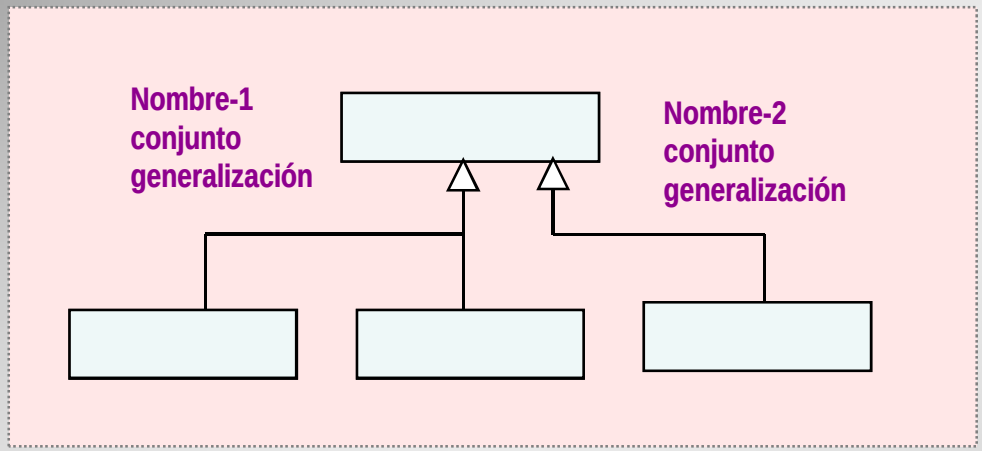
- Las subclases se pueden organizar en uno o más conjuntos de generalización

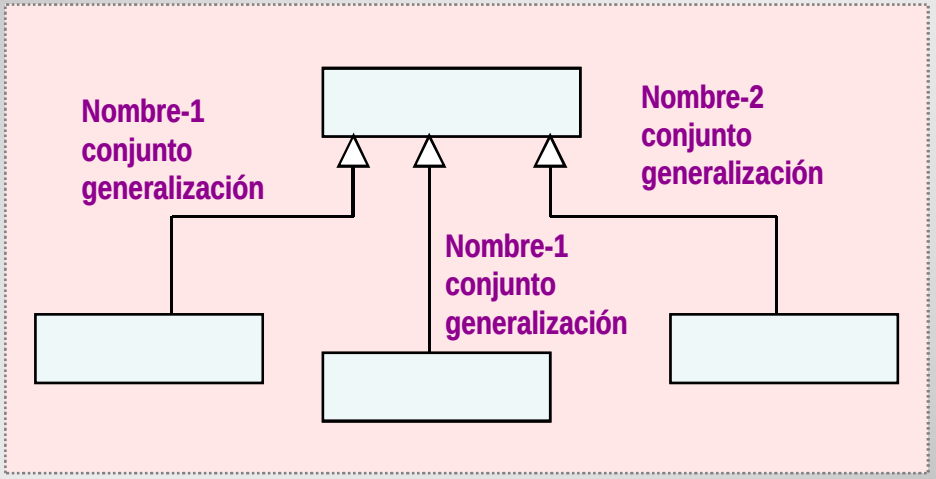
Ejemplo

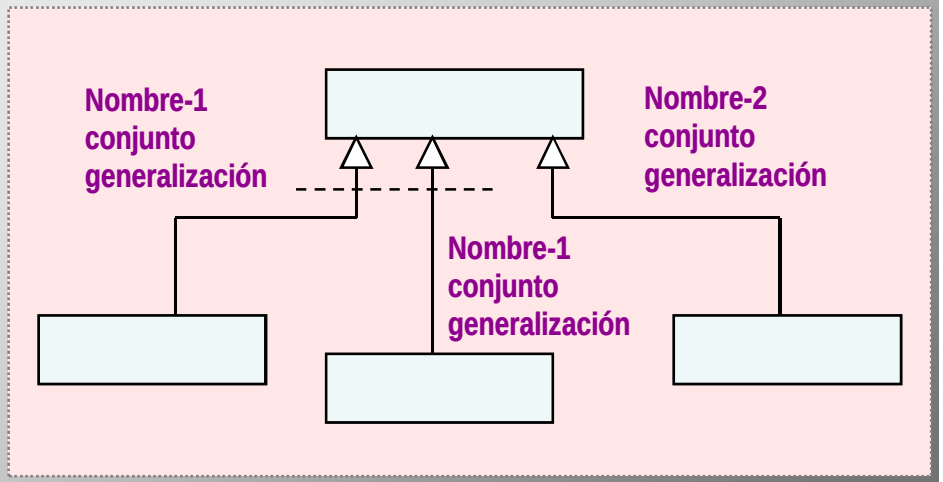


GENERALIZACIÓN

- Notación







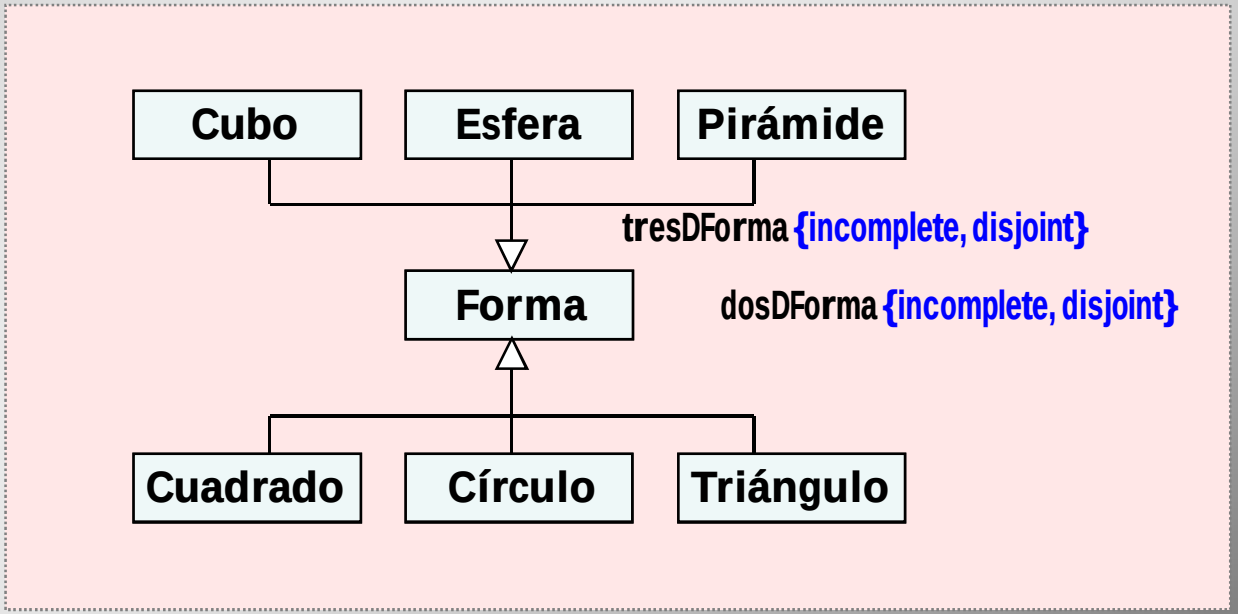
GENERALIZACIÓN

- Los conjuntos de generalización pueden tener restricciones aplicadas
  - **{complete}**
    Las subclases en el conjunto de generalización abarcan todas las posibilidades
  - **{incomplete}**
    Puede haber subclases que no estén en el conjunto
  - **{disjoint}**
    Un objeto puede ser instancia de uno y solamente uno de los miembros del conjunto de generalización
    Es el caso más común
  - **{overlapping}**
    Un objeto puede ser instancia de más de uno de los miembros del conjunto de generalización
    Es bastante poco común y requiere herencia múltiple

GENERALIZACIÓN

- las restricciones se combinan como
  - {incomplete, disjoint}
  - {complete, disjoint}
  - {incomplete, disjoint}
  - {incomplete, overlapping}

Ejemplo



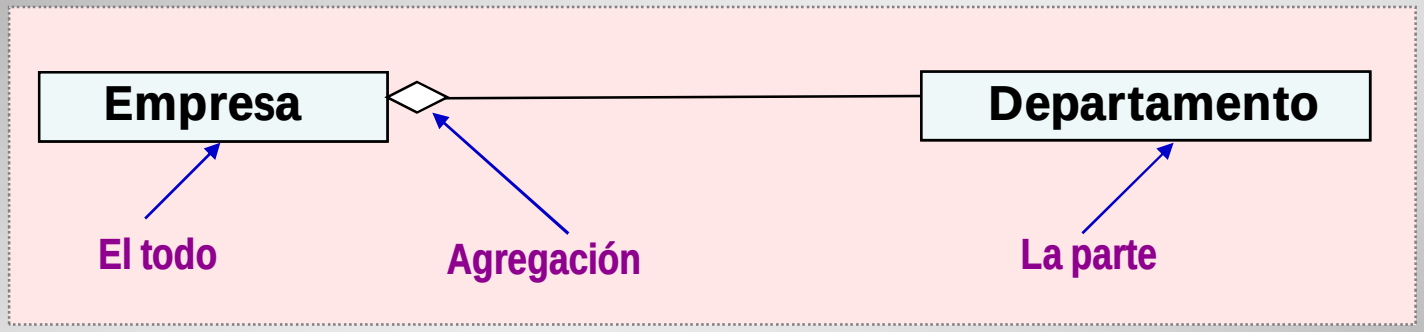
AGREGACIÓN Y COMPOSICIÓN

Agregación

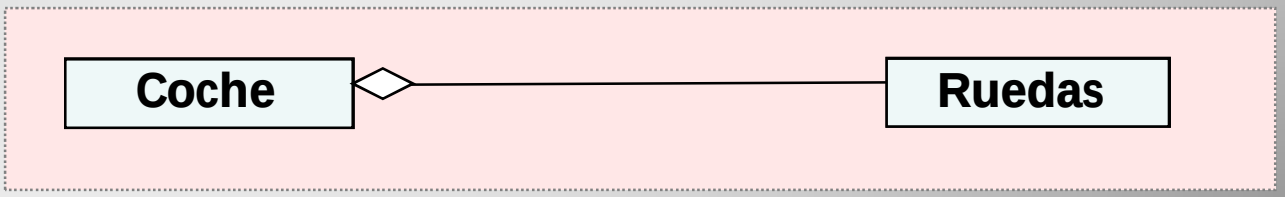
Es una tipo de relación (“parte de”) en la que una de las clases representa el “todo” y la otra la parte

Un objeto (el todo) utiliza los servicios de otro objeto (la parte)

Agregación conceptual



Agregación física



AGREGACIÓN Y COMPOSICIÓN

- La semántica de la agregación permite que
  - El conjunto puede existir algunas veces de manera independiente de las partes, algunas veces no
  - Las partes pueden existir independientemente del conjunto
  - El conjunto esta en cierto sentido incompleto si falta alguna de las partes
  - Es posible tener propiedad compartida de las partes por varios conjuntos
- La agregación es transitiva
- La agregación es asimétrica

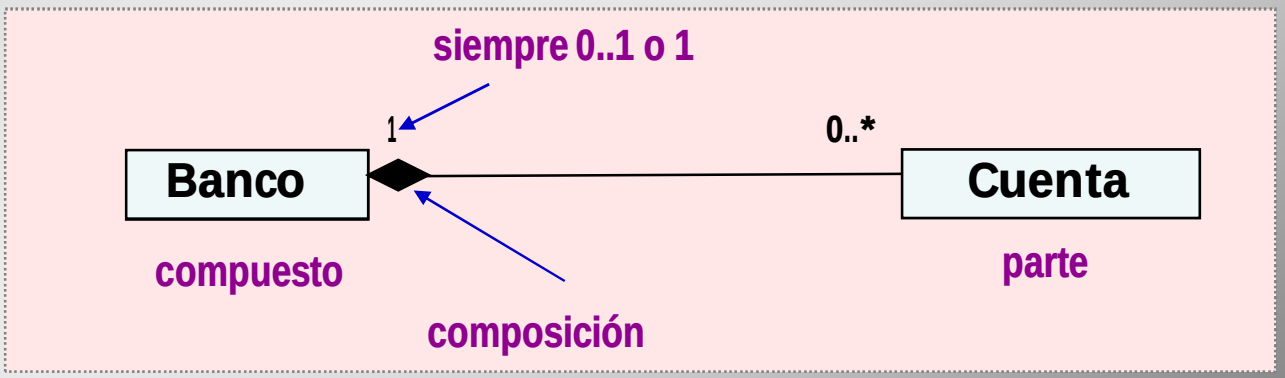
  Un objeto nunca puede ser, directa o indirectamente parte de sí mismo

AGREGACIÓN Y COMPOSICIÓN

Composición

Es una forma de agregación con una fuerte relación de pertenencia y vida de las partes con respecto al todo

- La semántica de la composición permite que
  - Las partes pueden solamente pertenecer a un conjunto
  - El conjunto es responsable de la creación y destrucción de las partes
  - El conjunto puede liberar partes siempre y cuando la responsabilidad la asuma otro objeto
  - Si se destruye el conjunto, se deben destruir todas sus partes
- La composición es transitiva y asimétrica

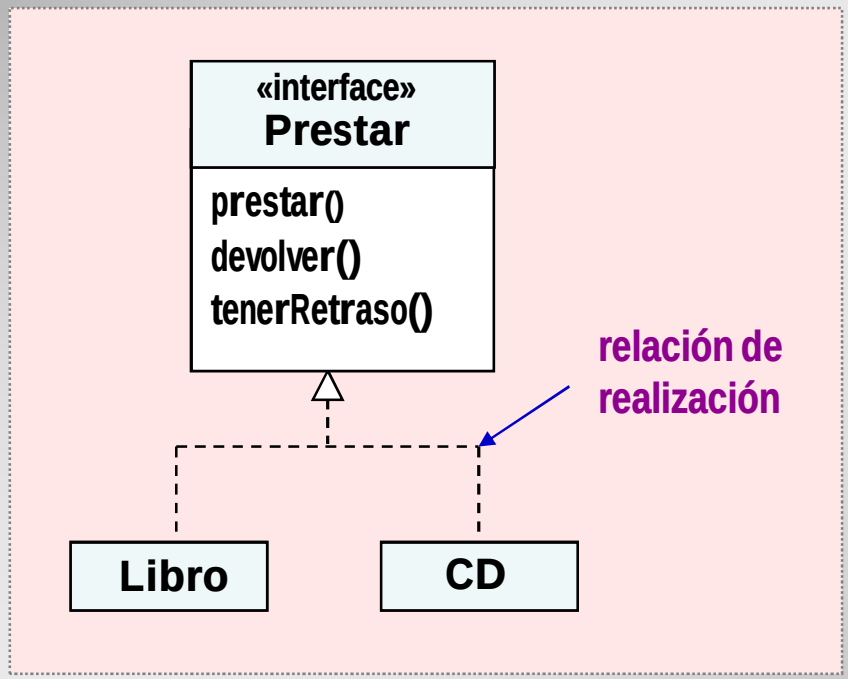


REALIZACIÓN

Es una relación que se establece entre dos elementos cuando uno de ellos especifica un contrato y el otro garantiza que se cumple

Ejemplo

Una interfaz presenta una relación de realización con las clases que la implementan

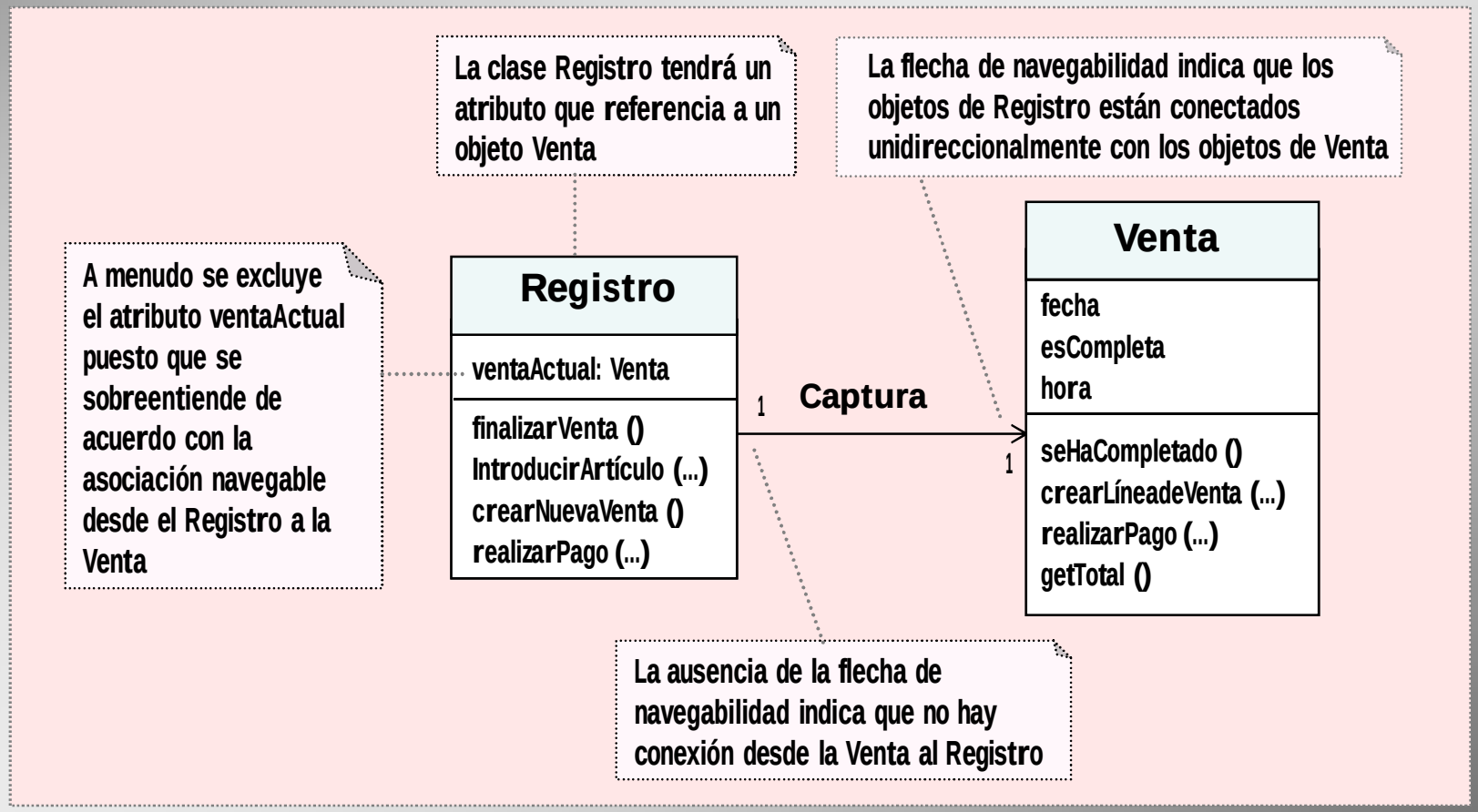


- La implantación de los métodos se realiza mediante una o varias clases concretas, subclases de la interfaz
- Una misma clase puede realizar varias interfaces (es un caso particular de herencia múltiple)
- Una interfaz puede ser implementada por varias clases

MECANISMOS DE EXTENSIÓN

Notas

Símbolos gráficos o textuales para representar comentarios asociados a uno o varios componentes



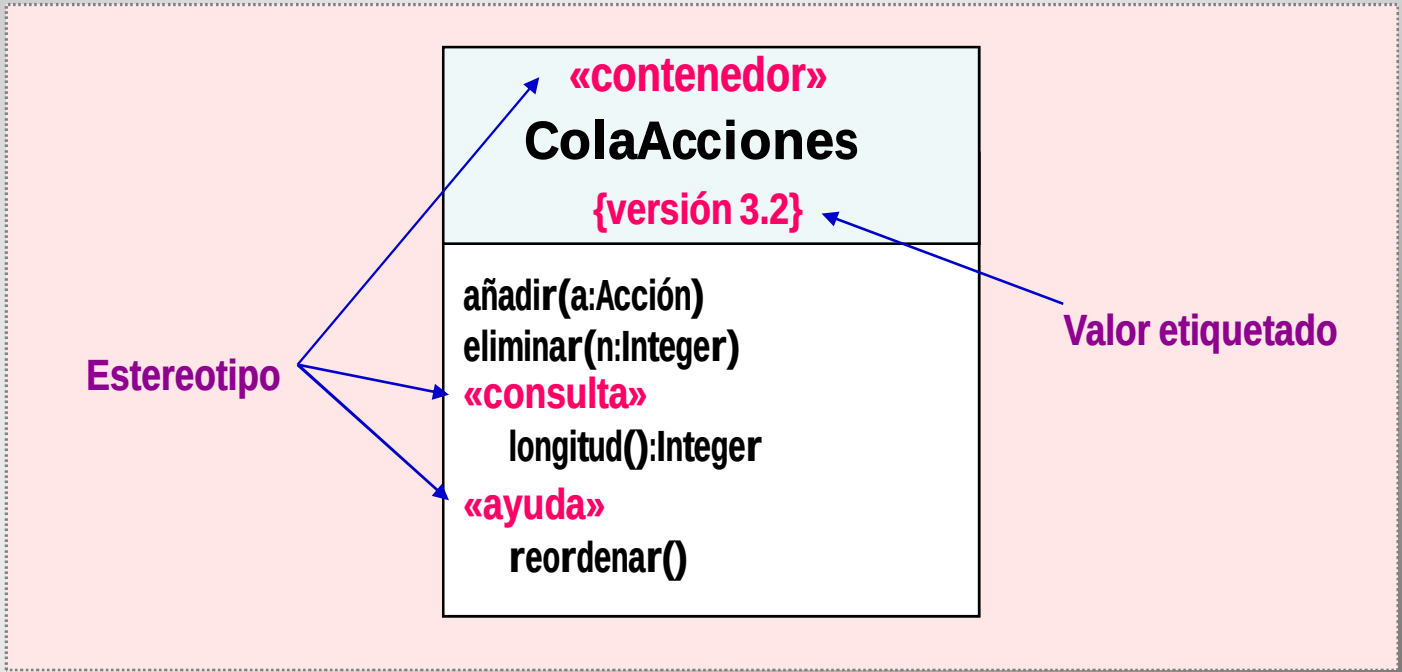
MECANISMOS DE EXTENSIÓN

Estereotipos

Añaden nuevos elementos de construcción y se representan entre «»

Valores etiquetados

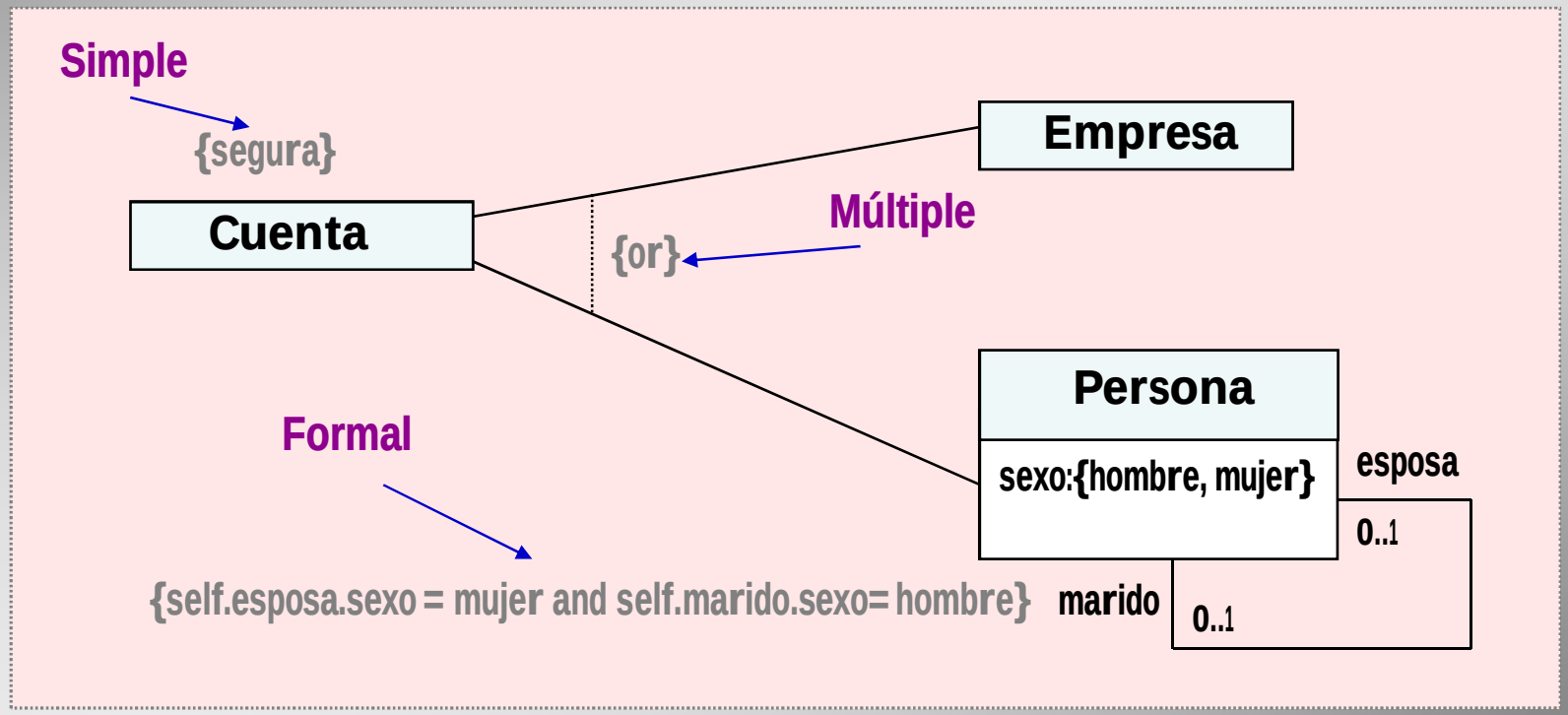
Añaden nuevas propiedades sobre los elementos del lenguaje y se representan entre {}



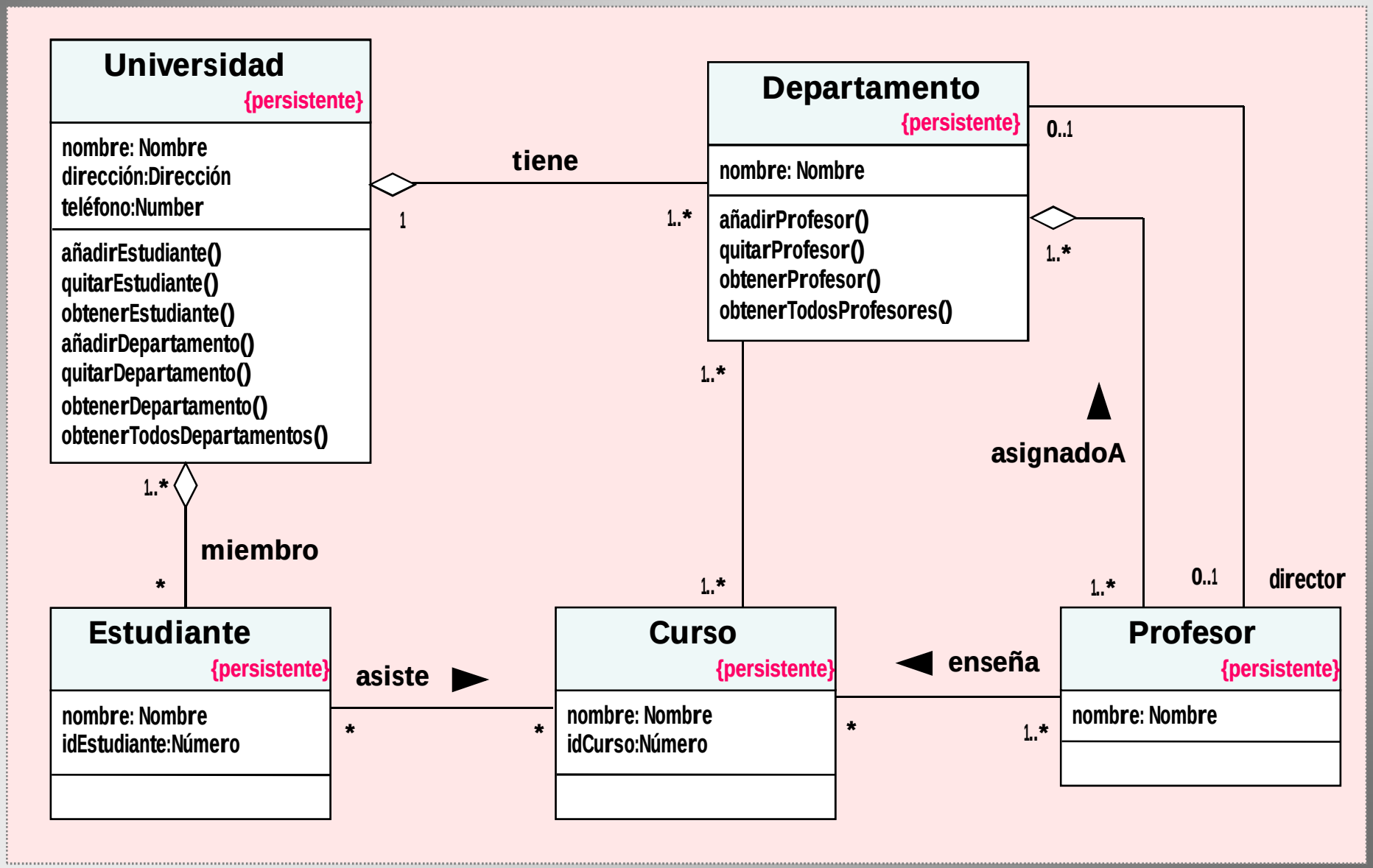
MECANISMOS DE EXTENSIÓN

Restricciones

Extensiones de la semántica de los elementos del lenguaje que añaden nuevas reglas o modifican las existentes



EJEMPLO DE DIAGRAMA DE CLASES



EJEMPLO DE DIAGRAMA DE CLASES