3.- Diseño estructural: Diagrama de clases

Mª Antonia Zapata

Máster Bases de Datos e Internet

Introducción

Los diagramas de clases sirven para

representar la estructura estática de un sistema

incluyendo

una colección de elementos de modelización estáticos, tales como clases y relaciones

Conceptos básicos

- Un objeto es algo distinguible que percibimos como que tiene existencia, sea física o conceptual

  Ejemplos: Pedro González,
  el libro “Cien años de soledad”,
  la luna,
  la asignatura de “Acceso a bases de datos”

- Una clase refiere genéricamente a los objetos de una familia que se perciben con propiedades y comportamiento comunes

  Ejemplos: persona, libro, satélite, asignatura

- Una instancia de una clase (objeto) refiere a la representación de un objeto de una clase.

Conceptos básicos

- Los objetos generalmente los percibimos relacionados entre sí

  Ejemplos: “Cien años de soledad” lo escribió Gabriel García Márquez,
  la luna es un satélite de la tierra

- Una asociación refiere genéricamente a las relaciones que existen entre objetos de clases

  Ejemplos: libro escrito por autor
  satélite de un planeta

- Un enlace refiere a la representación de una relación entre instancias de clases

Elementos básicos

Clase:

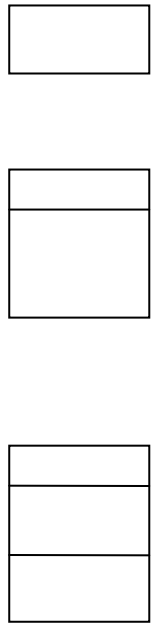
describe **genéricamente** a una familia de objetos que tienen en común una serie de **atributos** y **operaciones**

Atributo:

describe genéricamente una **propiedad** de los objetos de una clase (generalmente, describe hechos estáticos o estructurales)

Operación:

describe genéricamente un servicio que puede ser requerido a cualquier objeto de una clase para que muestre un comportamiento



Elementos básicos

Asociación:

describe una relación genérica entre objetos de clases

Multiplicidad:

describe el número mínimo y máximo de enlaces posibles

<min..max>

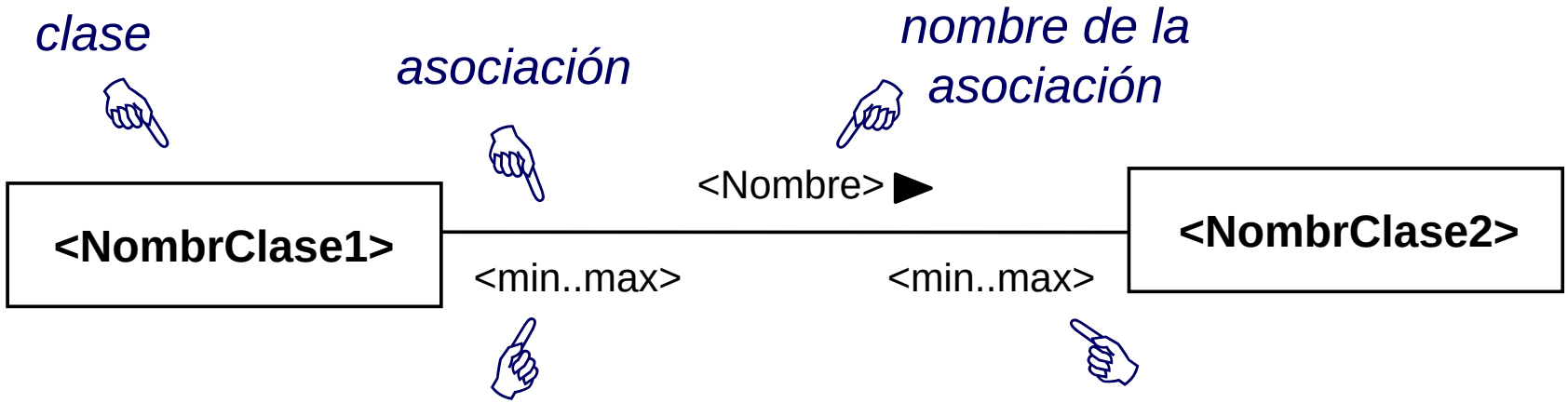
0..* *

1..*

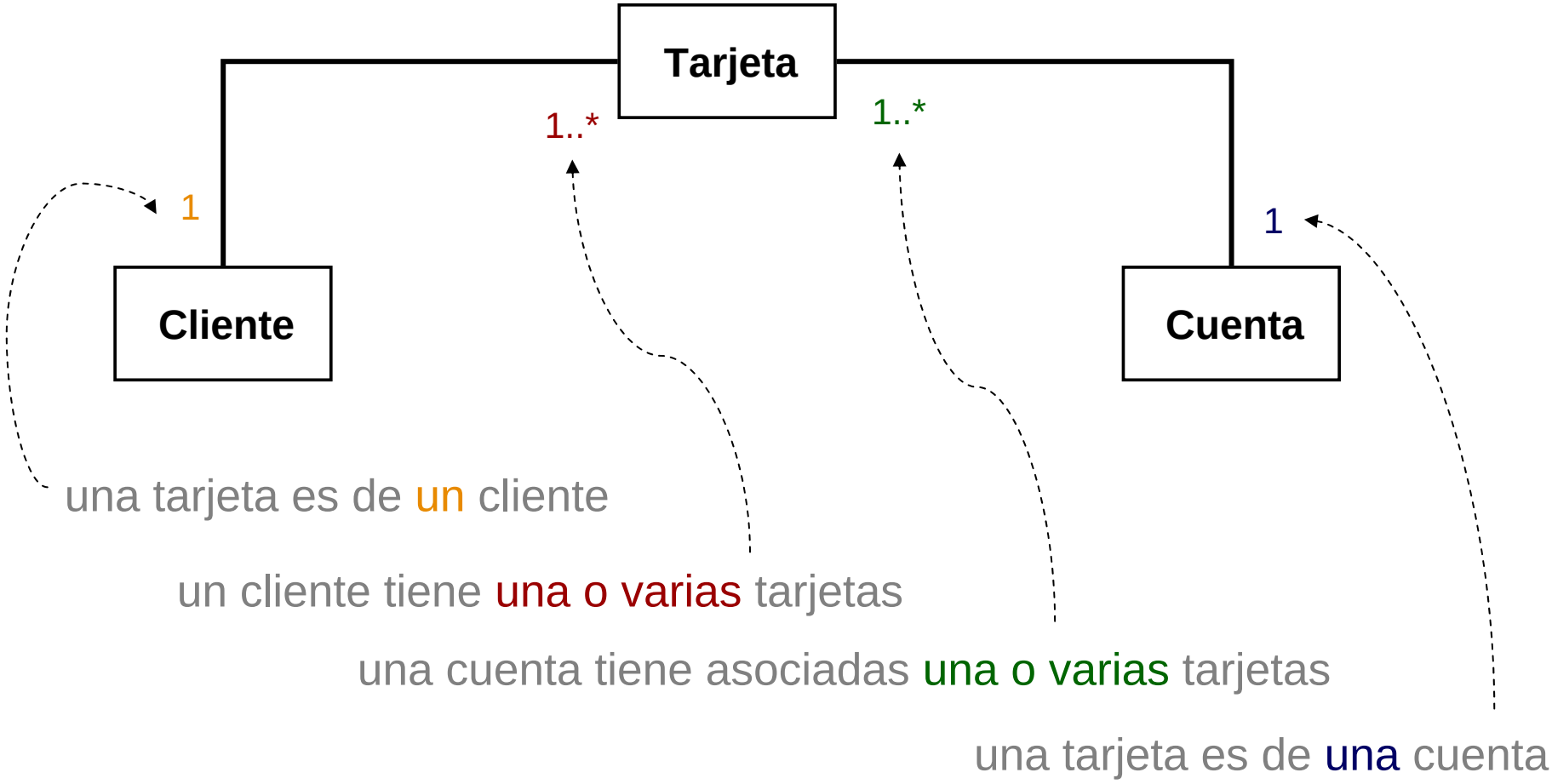
1..1 1

n..m

Representación gráfica



Ejemplo: cajero automático (versión 1)



Restricciones

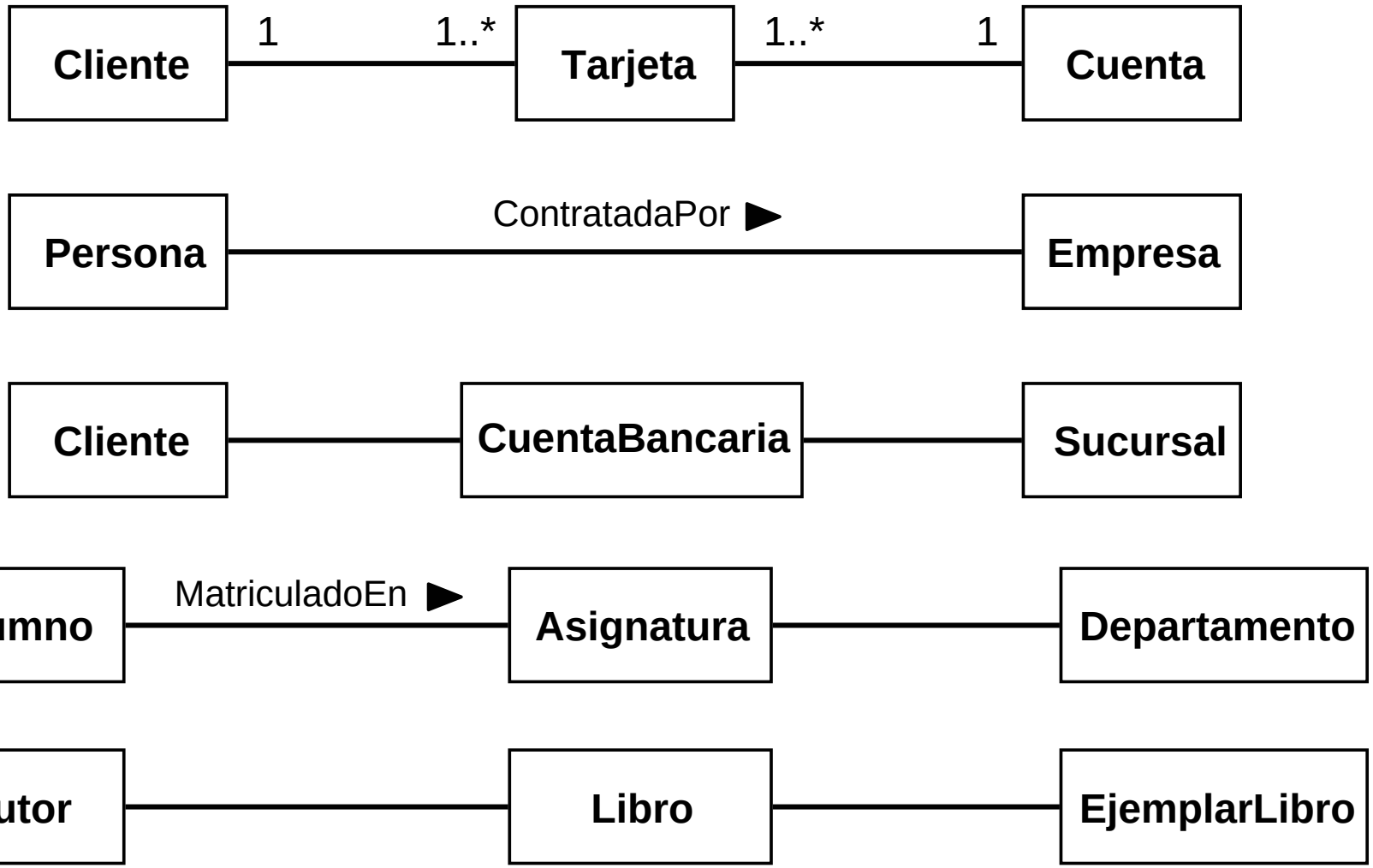
- Dentro de una misma clase, no se pueden repetir nombres de atributos
- La multiplicidad mínima no puede ser negativa
- La multiplicidad máxima tiene que ser mayor o igual que la mínima

Recomendaciones

- El nombre de las clases se escribe en negrita
- Los nombres de las clases y las asociaciones empiezan por mayúscula
- Los nombres de los atributos y las operaciones empiezan por minúscula
- Cuando un nombre está formado por más de una palabra, entonces la segunda y siguientes palabras empiezan por mayúscula

Ejemplos: Persona, Autor, EjemplarLibro, EscritoPor,
edad, teléfonoMóvil, expulsarTarjeta, dibujar

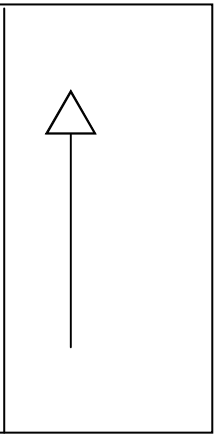
Ejercicios



Jerarquía de clases

Generalización:

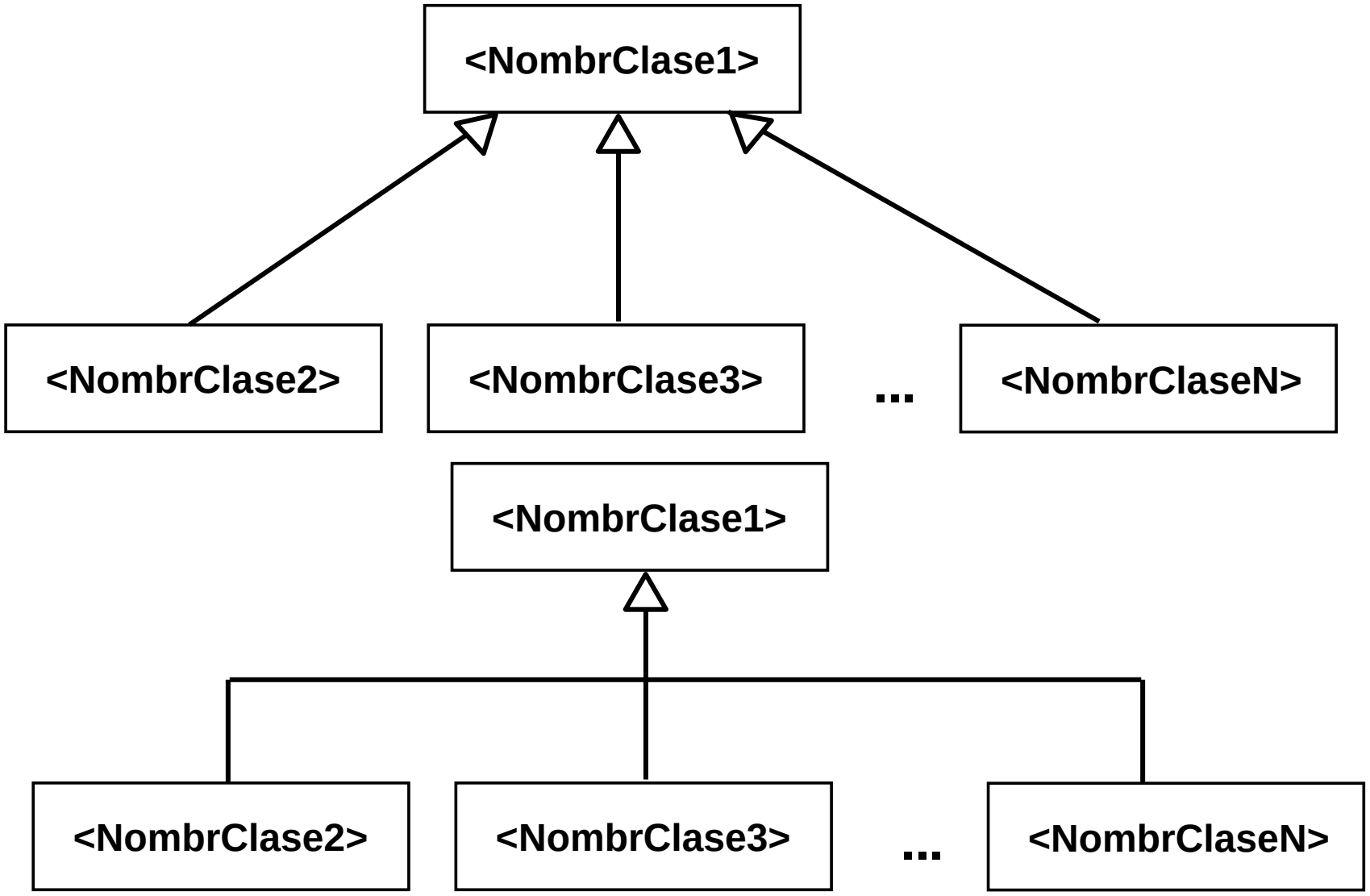
es una asociación entre una clase y otra más general de modo que la primera describe una subfamilia de objetos de esta última



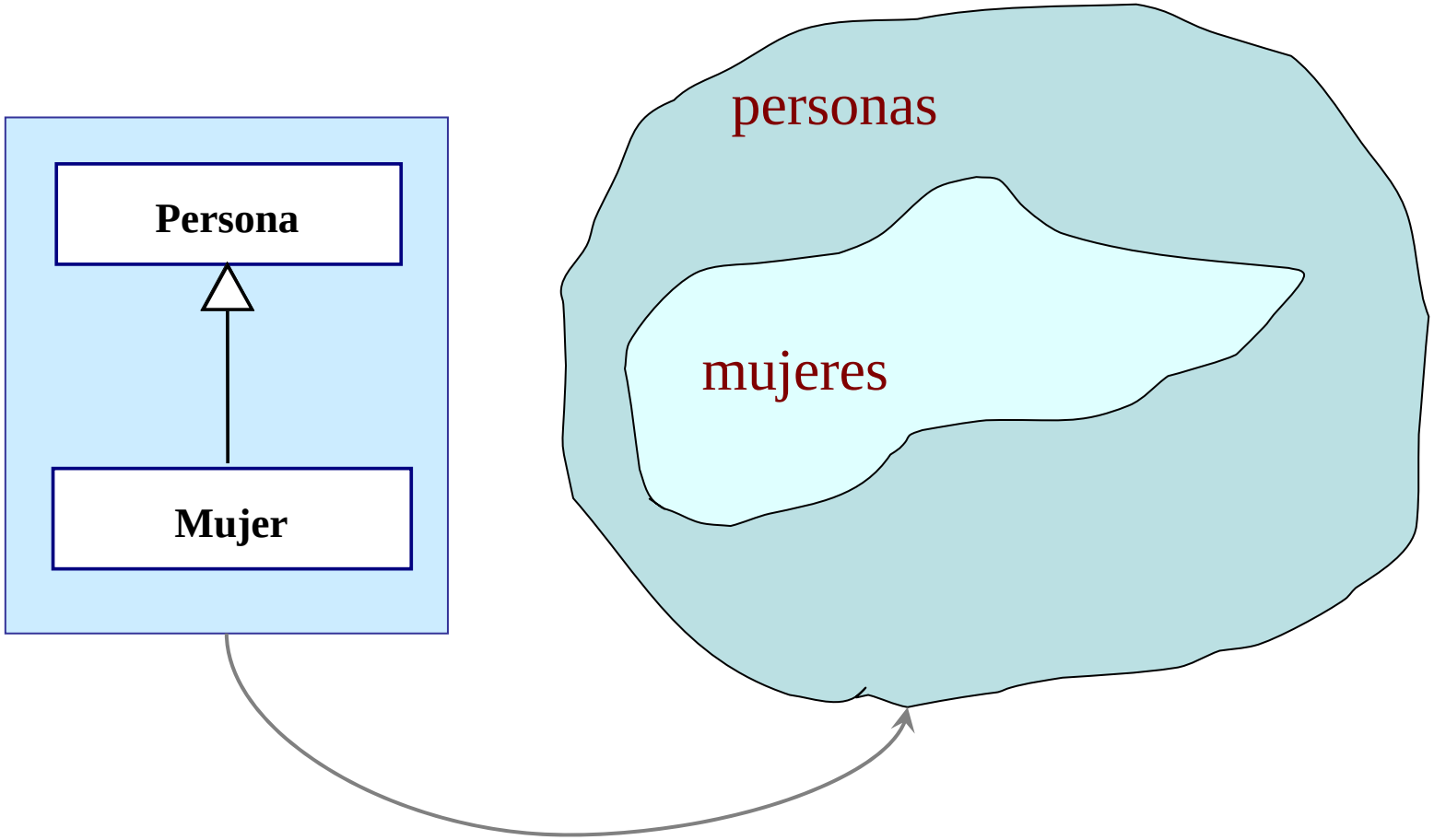
Se debe verificar la propiedad de sustituibilidad (substitutability), es decir, se puede usar una instancia de la especialización siempre que se espere una instancia de la generalización

El elemento más específico hereda las características del elemento más general

Representación gráfica



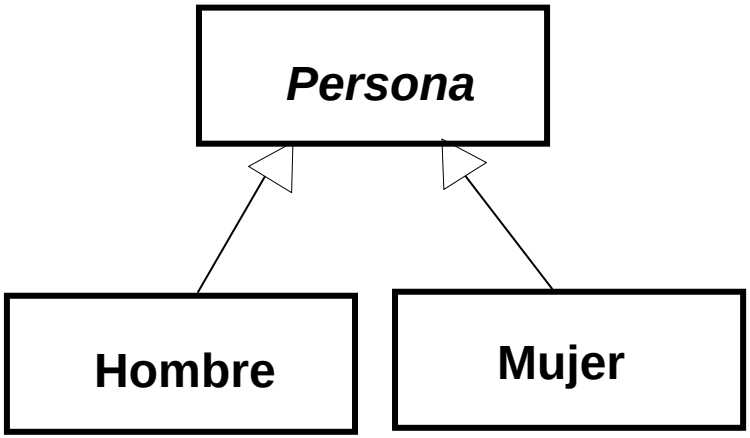
Ejemplo



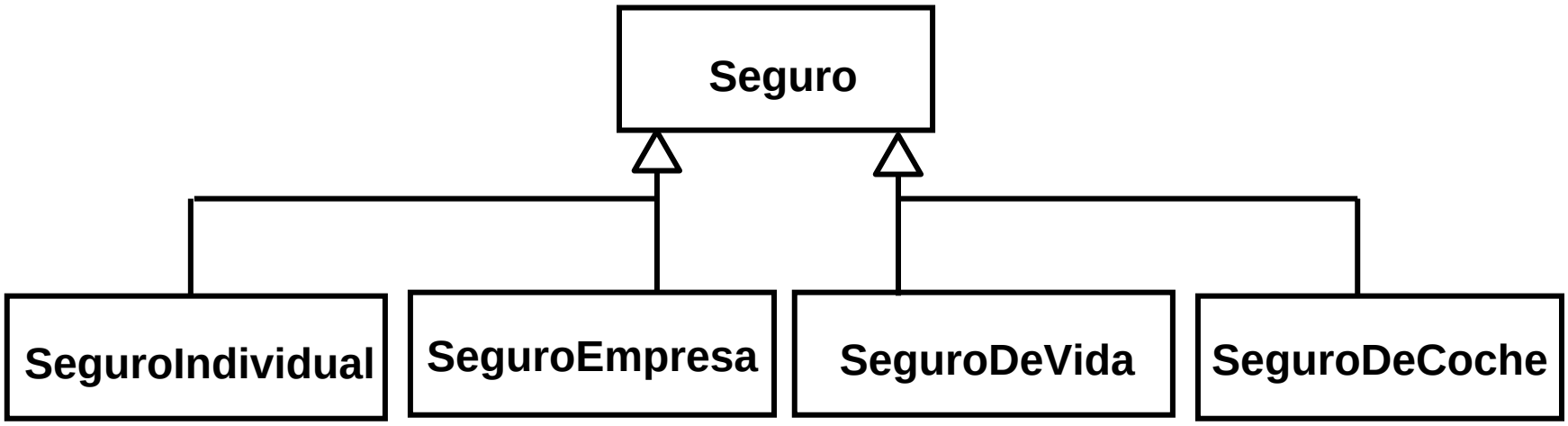
representa una relación de subconjunto

Recomendaciones

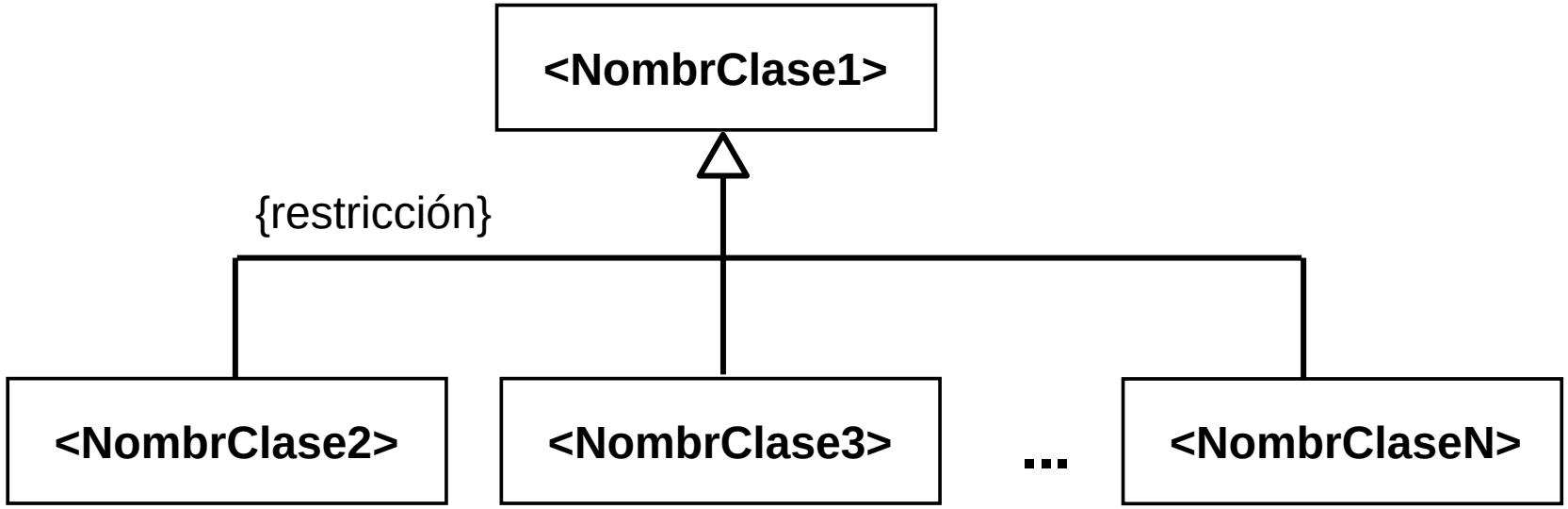
- El nombre de una clase abstracta se escribe en itálica



Se pueden realizar distintas clasificaciones



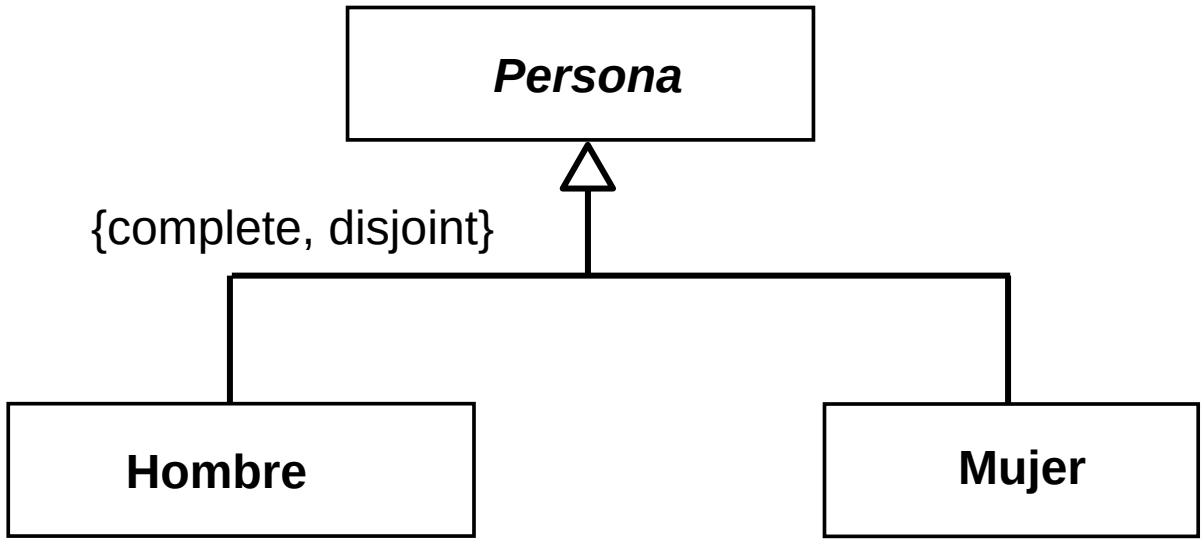
Tipos de clasificaciones



Las restricciones posibles son:

- complete o incomplete: si todas las instancias de la superclase están en una subclase o no.
- disjoint o overlapping: si cada instancia de la superclase está en como mucho una subclase o no

Ejemplo



Clases: atributos

Atributo: describe genéricamente una propiedad de los objetos de una clase (generalmente, describe hechos estáticos o estructurales)

Visibilidad: ámbito en el que el atributo es visible (+: público, -: privado, #: protegido o ~: paquete)
Nombre: identifica el atributo dentro de la clase (no se puede repetir)
Tipo: describe el tipo del valor del atributo (entero, real, ...)
Multiplicidad: describe el número mínimo y máximo de valores posibles de un atributo
Valor inicial: describe el valor que se asigna por defecto a un atributo cuando se instancia un objeto
Alcance: describe si es un atributo de instancia o de clase

Notación y ejemplos

Notación:

[visibilidad] nombreAtributo [: tipo] [[multiplicidad]] [= valorIncicial]

- Los atributos de clase se escriben subrayados
- Por defecto, la visibilidad es pública y la multiplicidad es [1..1]

Ejemplos:

nombrePersona

origen: Punto

idUnico: Long

+provincia: String =“Zaragoza”

-segundoApellido: Integer [0..1]

#prioridad: Entero =1

Ejemplos

Persona
nombrePersona primerApellido segundoApellido fechaNacimiento

Dirección
calle: String localidad: String [0..1] provincia: String [0..1] =“Zaragoza” CP: String

Libro
+título: String [0..1] +autor: String [0..*] +fechaPublicación: date

Producto
#id: Float +nombre: String +precio: Float

Clases: operaciones

Operación: describe genéricamente un servicio que puede ser requerido a cualquier objeto de una clase para que muestre un comportamiento

Visibilidad: ámbito en el que la operación es visible (+: público, -: privado, #: protegido o ~: paquete)
Nombre: identifica la operación dentro de la clase (no se puede repetir)
Parámetros: lista de parámetros de la operación
Tipo del valor devuelto: tipo del resultado si es que tiene
Alcance: describe si es una operación de instancia o de clase

Notación y ejemplos

Notación:

[visibilidad] nombreOperación [(listaParámetros)] [:tipoRetorno]

- Las operaciones de clase se escriben subrayadas
- Por defecto, la visibilidad es pública

Ejemplos:

mover( )

ponerAlarma(t:temperatura)

copiasEnEstantería( ):Entero

+añadirCurso(c:Curso):Booleano

-compactar( )

#comprobarErrores( )

Ejemplos

Figura
origen
mover( ) redimensionar( ) visualizar( )

Ventana
origen:Punto tamaño:Vector
abrir( ) cerrar( ) mover( )

SensorTemperatura
+reiniciar( ) +ponerAlarma(t:temperatura) +valor( ): temperatura

Transacción
+ejecutar( ) +rollback( ) #prioridad( ) #marcaDeTiempo( )

Instancias

Instancia de clase (objeto):

describe un objeto de una clase mediante valores de los atributos de la clase. El objeto responde a las operaciones de la clase

Instancia de asociación (enlace):

describe una relación entre objetos

Notación

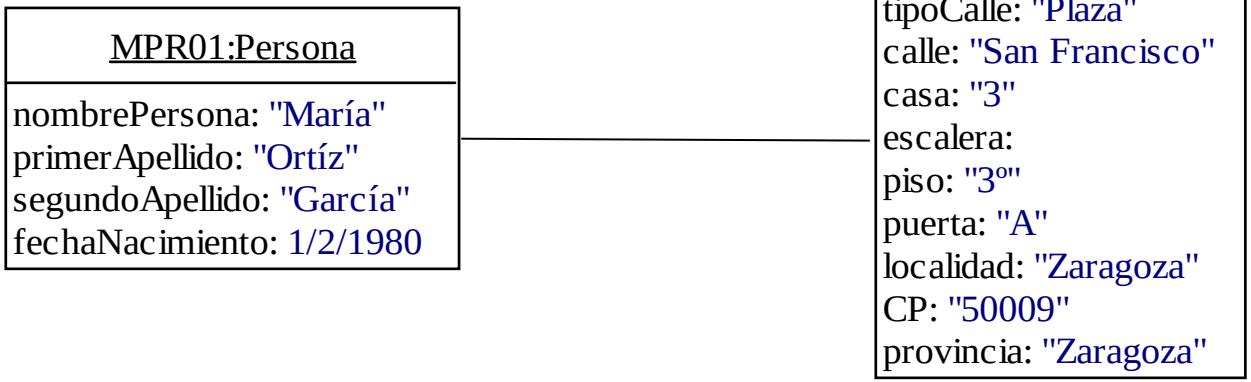
nombre:clasificador

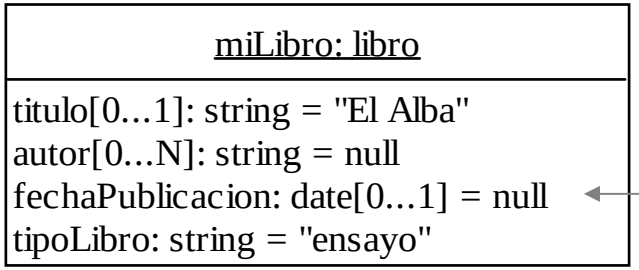
Ejemplos

objeto *sin clase*

objeto *anónimo*

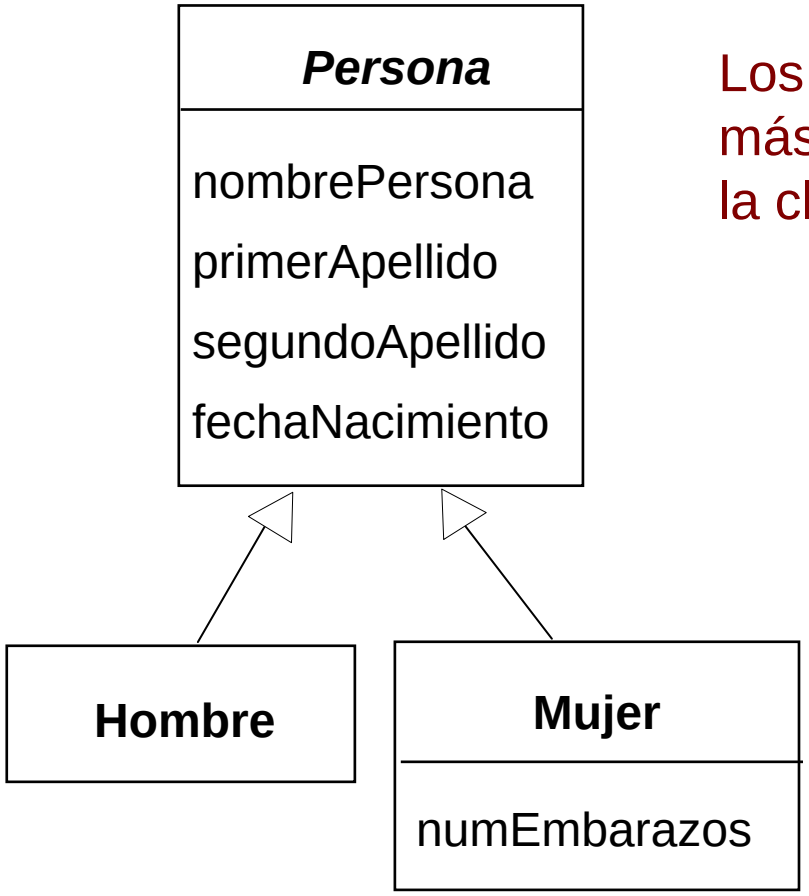
objeto *no anónimo*





Valor nulo describe la ausencia de valor

Propiedad de herencia



Los atributos y operaciones de la clase más general (superclase) son atributos de la clase especializada (subclase)

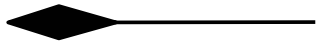
Una subclase puede redefinir una operación heredada

Ejemplos

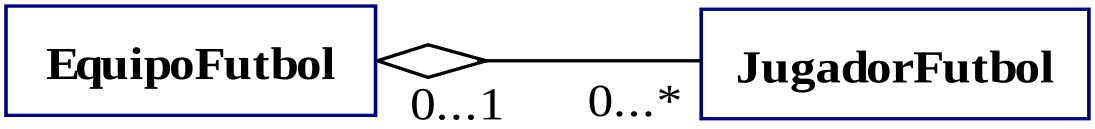
MPR01:Mujer
nombrePersona: "María" primerApellido: "Ortíz" segundoApellido: "García" fechaNacimiento: 1/2/1980 numEmbarazos: 0

PPGO3:Hombre
nombrePersona: "Pedro" primerApellido: "Pérez" segundoApellido: "García" fechaNacimiento: 7/10/2005

Tipos especiales de asociaciones

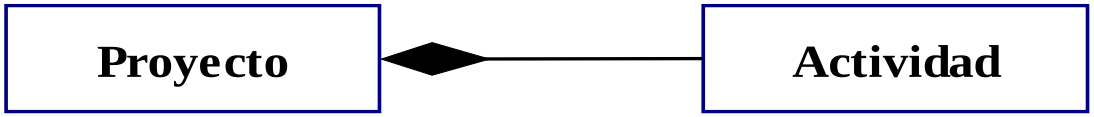
Agregación: es una asociación que describe una relación entre un todo y sus partes de modo que las partes pueden existir por sí mismas	
Composición: es una asociación que describe una relación entre un todo y sus partes de modo que las existencias de las partes se perciben como totalmente dependientes del todo	

Ejemplos



relación tipo ‘tiene un’

En una agregación una parte puede pertenecer a varios todos



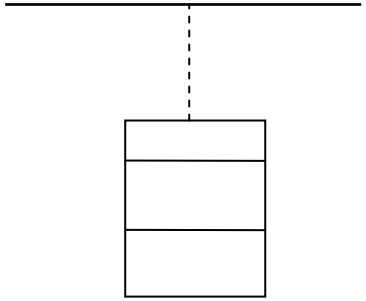
relación tipo ‘contiene a’

En una composición una parte sólo puede pertenecer a un todo

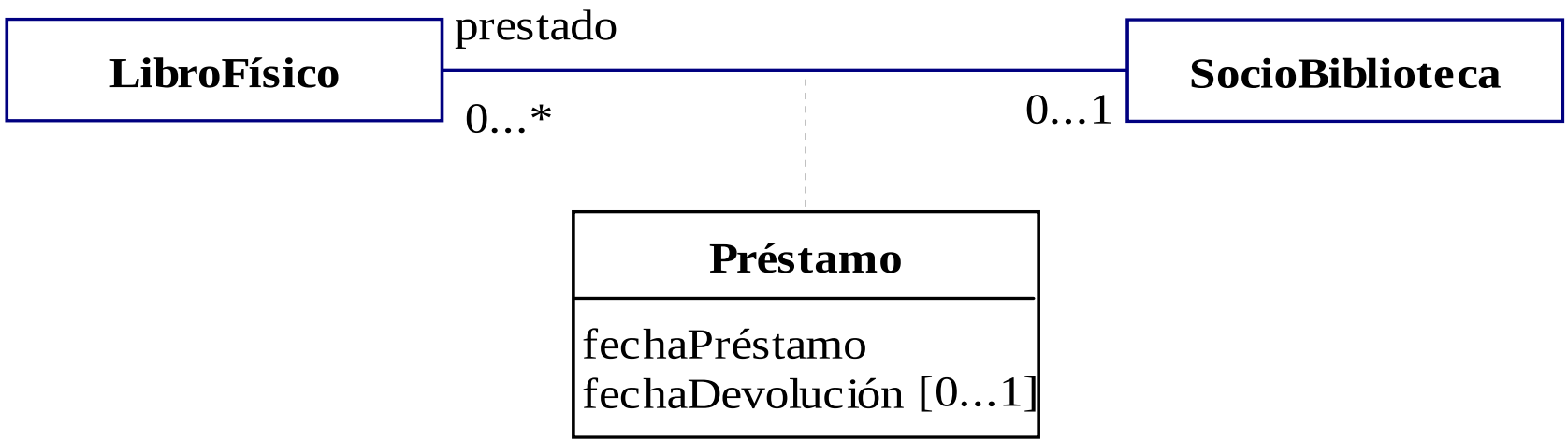
Clase asociativa

Clase asociativa:

describe una asociación que refiere a una familia de relaciones entre objetos sobre las que se perciben propiedades que son propias de las relaciones



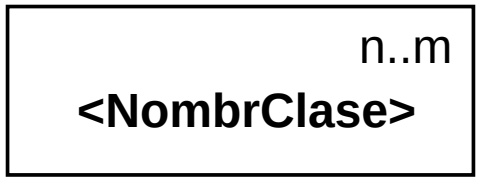
Ejemplo



Un atributo debe situarse en el elemento
al que directamente atribuye

Multiplicidad de clase

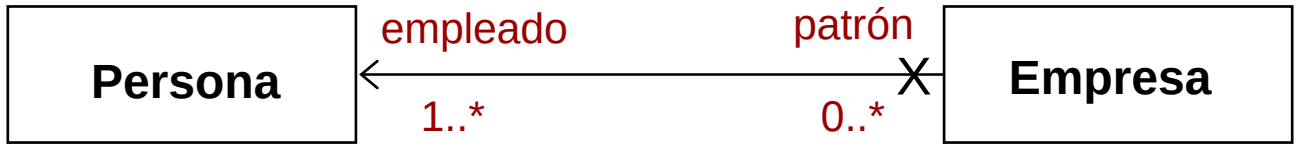
Las clases también pueden tener asociada una multiplicidad que limita el número mínimo y máximo de instancias.



Por defecto se toma el valor *.

Una clase que tiene sólo una instancia se denomina clase unitaria o ‘singleton’

Roles y navegación

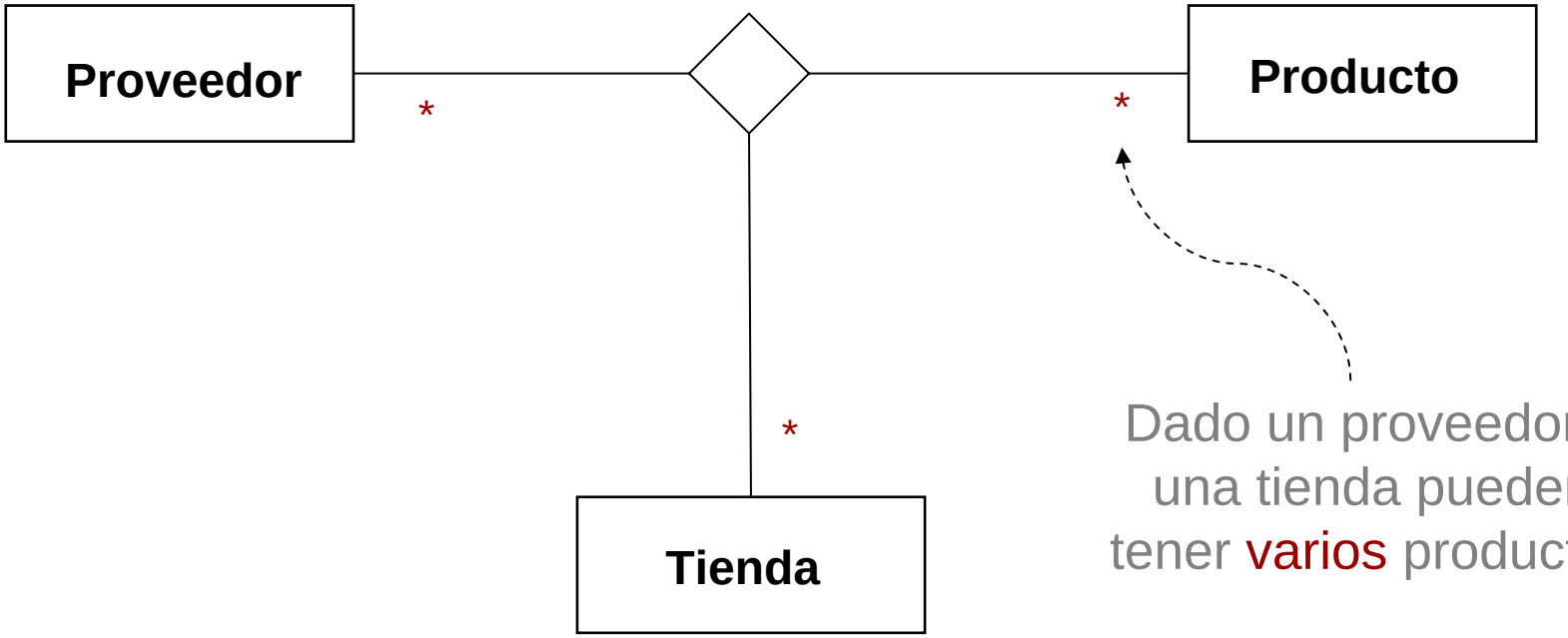


Un rol indica el papel que juega una clase en una asociación.

La navegación define la posibilidad (o no) de acceder a los objetos de una clase desde otra.

Asociaciones n-arias

La multiplicidad de una clase en una asociación n-aria especifica el número de instancias que pueden relacionarse con una instancia de cada una de las otras clases

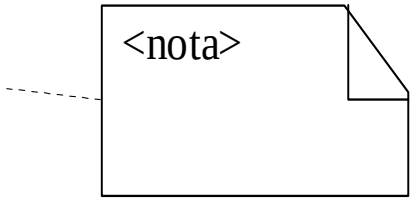


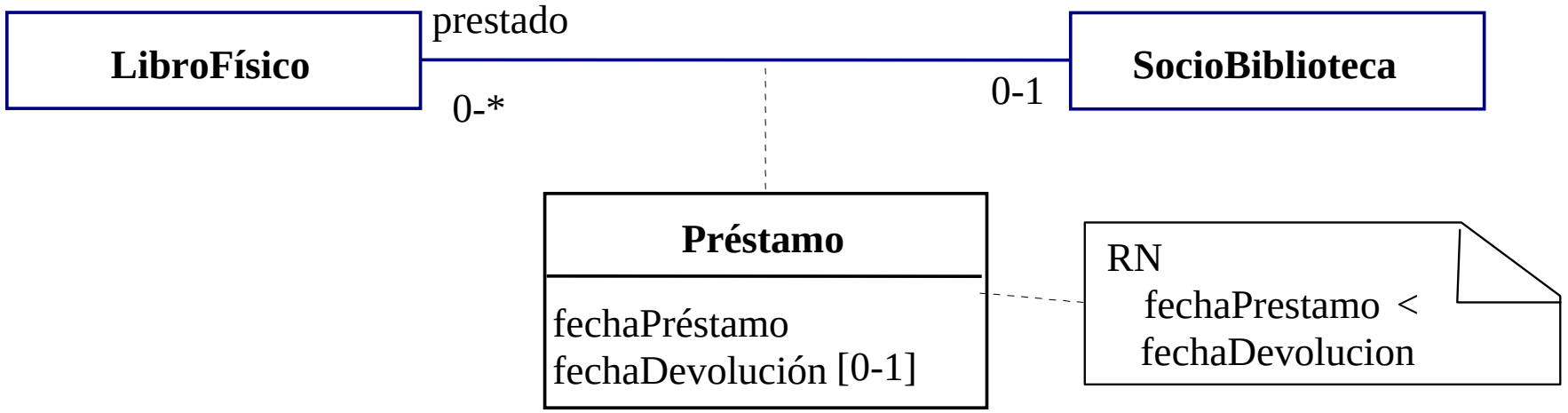
Dado un proveedor y una tienda pueden tener varios productos

Notas

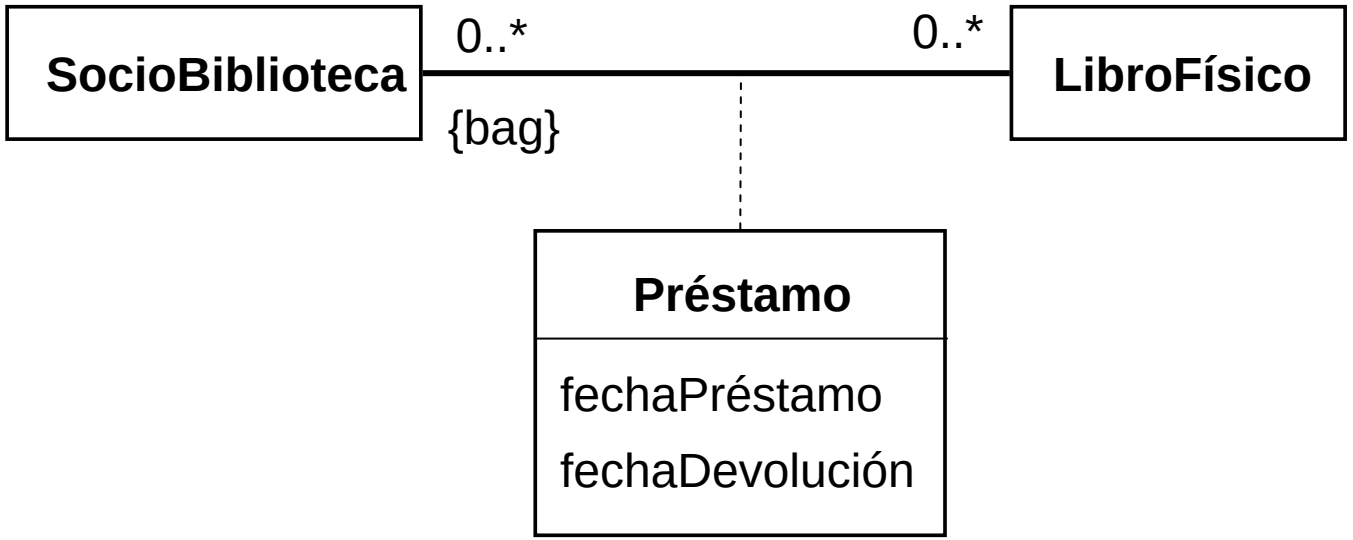
Nota:

es una observación, condición semántica o restricción



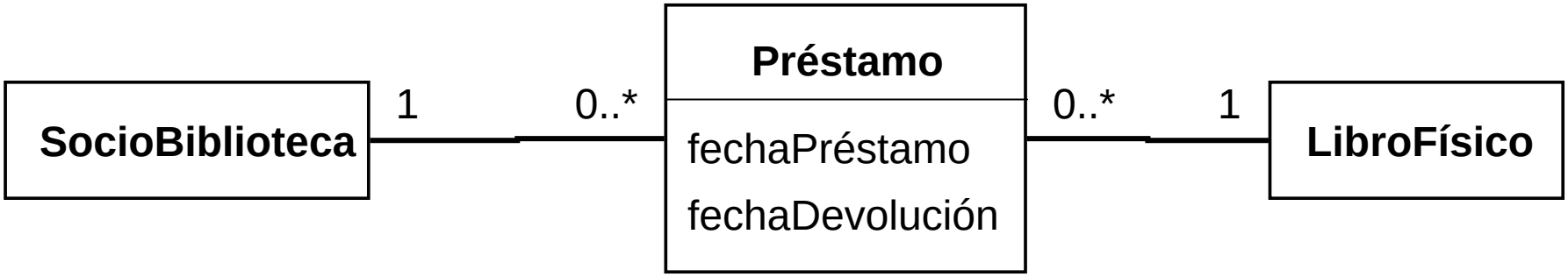


Ejemplo de préstamo con UML 2.0



- un préstamo es la asociación entre un único socio y un único libro (por la propia definición de clase asociativa).
- un socio puede haber cogido prestado varios libros.
- un libro ha podido ser cogido prestado por socios varias veces.
- un socio puede haber cogido prestado un mismo libro varias veces (correspondiendo a préstamos distintos).

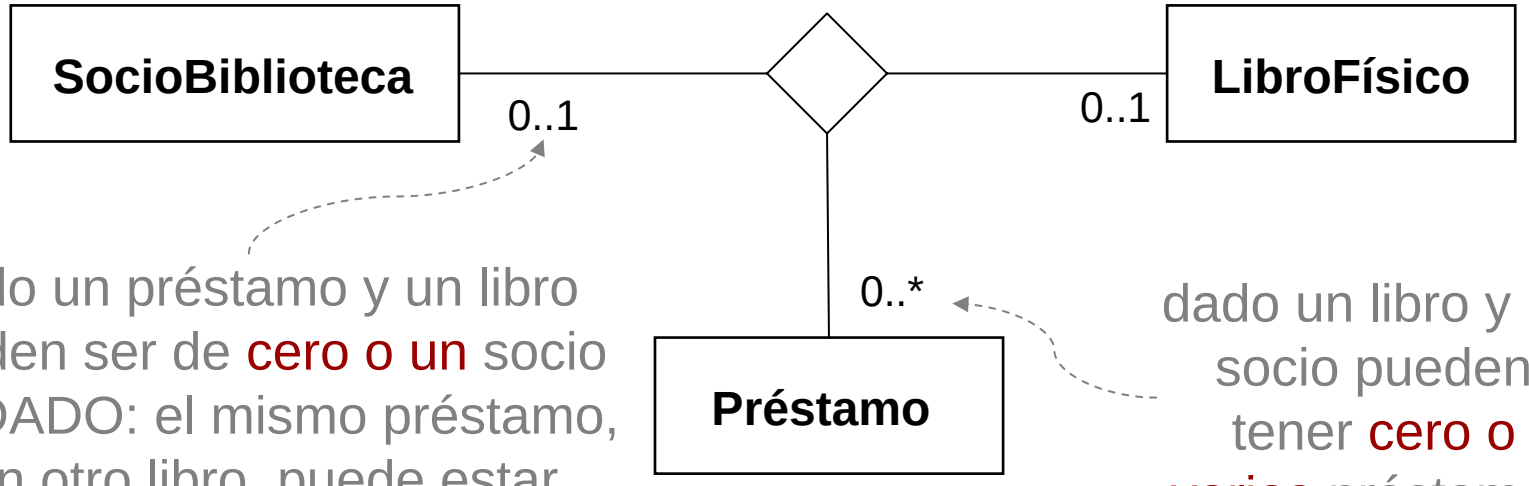
Ejemplo de préstamo con UML 1.5



- un préstamo tiene asociado siempre un único libro y un único socio
- un socio puede haber solicitado varios préstamos (de un libro cada vez)
- un libro ha podido ser prestado varias veces (a un socio cada vez).
- un socio puede haber cogido prestado un mismo libro varias veces.

las restricciones son las mismas

Ejemplo de préstamo con asociación ternaria



- un préstamo puede estar asociado varias veces con libros y socios
- un socio puede haber solicitado varios préstamos (de un libro cada vez)
- un libro ha podido ser prestado varias veces (a un socio cada vez).
- un socio puede haber cogido prestado un mismo libro varias veces.

¡¡¡¡las restricciones son distintas!!!!

Asociaciones ternarias en UML

En una asociación ternaria es conveniente poder dar dos tipos de información:

- por cada pareja de instancias de dos clases, con cuántas instancias de la otra clase está relacionada.

  Por ejemplo, indicar que cada pareja (socio, libro) se relaciona con cero o varios préstamos

  esta información SÍ se indica en UML a través de la multiplicidad

- en cuántas asociaciones participa una instancia de una clase.

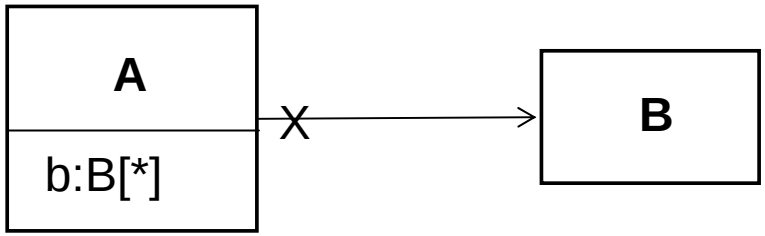
  Por ejemplo, indicar que dado un préstamo siempre participa en una sola instancia de la asociación

  ¡¡¡¡esta información NO se indica en UML a través de la multiplicidad!!!!

  ⇓

  es necesario imponer una restricción y especificarla usando OCL

Navegación vs atributo

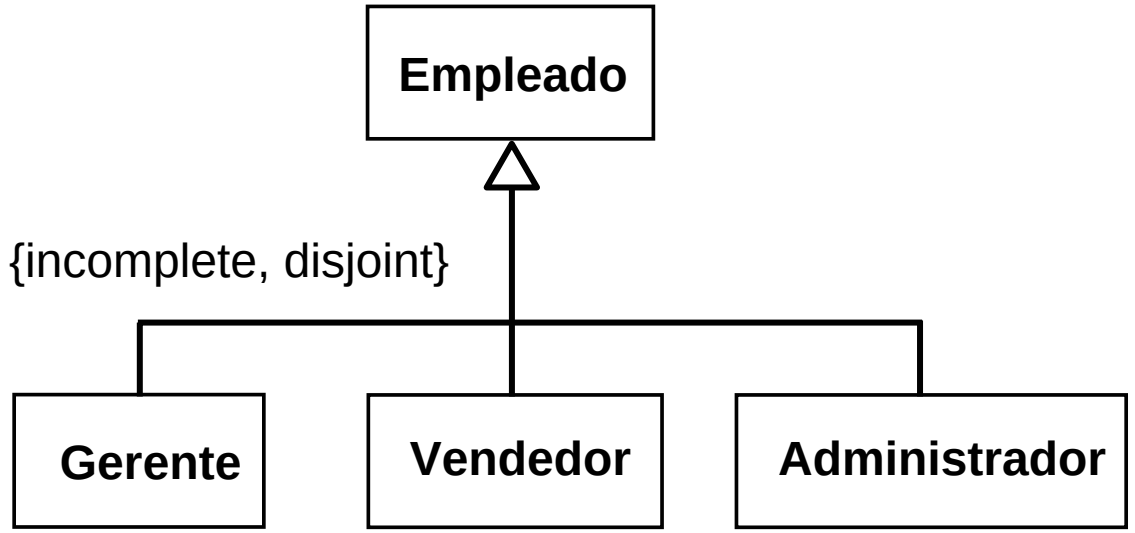


Un extremo de asociación navegable es también un atributo.

Se puede utilizar un atributo para representar el extremo navegable de una asociación.

Esta notación se puede usar en conjunción con la representación gráfica de la asociación con objeto de que quede claro que el atributo representa el extremo de una asociación

Especialización dinámica (1/2)

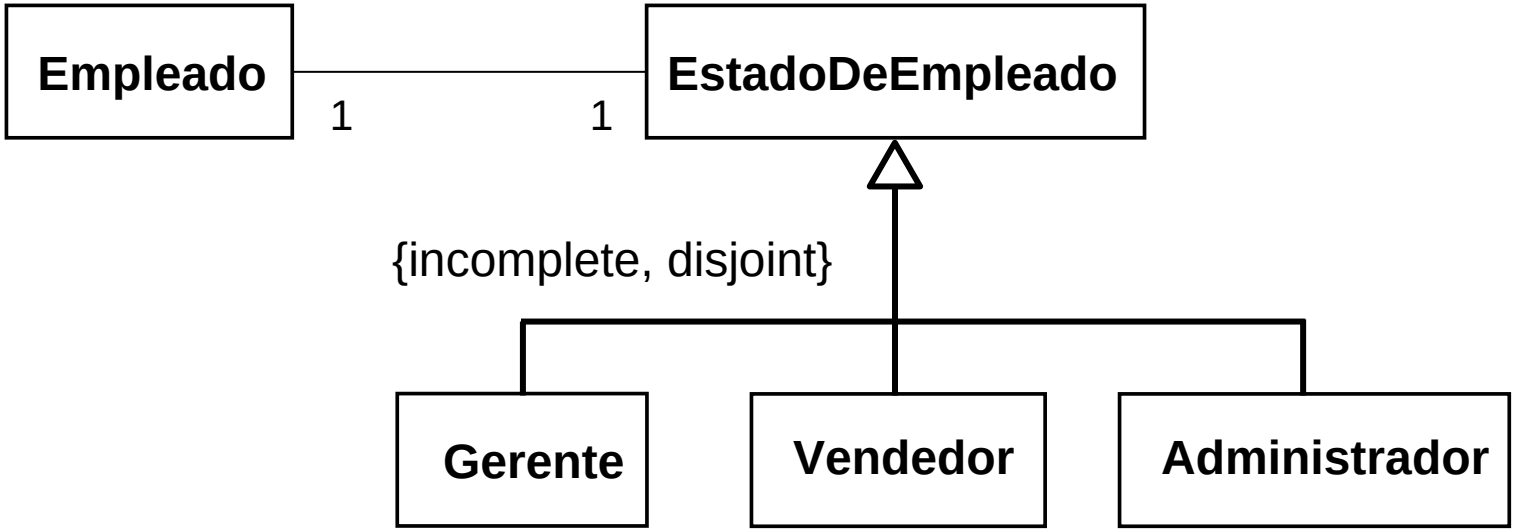


Un mismo empleado puede ir desempeñando distintos puestos en la empresa.

Por lo tanto un mismo objeto puede ir cambiando de clase a lo largo del tiempo.

El problema es que la mayoría de los lenguajes de programación no ofrecen la posibilidad de que un objeto cambie de clase.

Especialización dinámica (1/2)



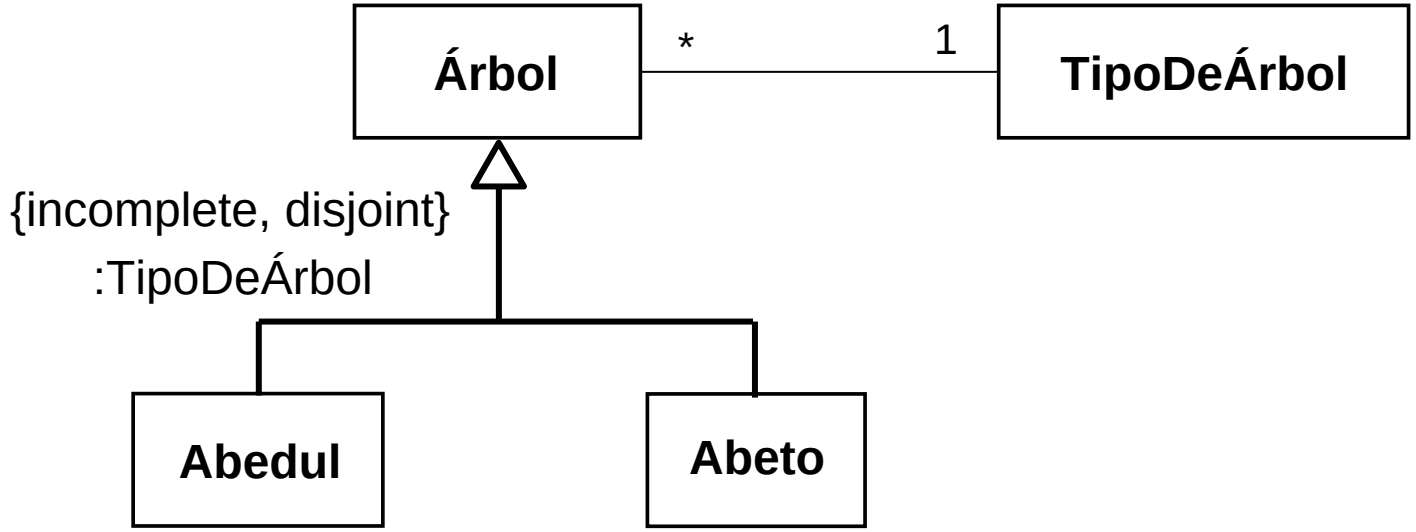
El empleado es siempre el mismo y lo que cambia es el estado en el que se encuentra.

Cuando un empleado cambia de estado se destruye el objeto que representa ese estado y se crea un nuevo objeto representando el nuevo estado.

Con este diseño ya no se utiliza especialización dinámica.

Se puede representar muy fácilmente la posibilidad de que un mismo empleado desempeñe varios papeles en la misma empresa o el hecho de guardar la historia de los puestos que ha ido ocupando un empleado.

PowerType (supra tipo)



Power type: clase cuyas instancias son subclases de otra clase.

Por ejemplo, las instancias de la clase TipoDeÁrbol son abedul y abeto, que a su vez son subclases de Árbol.

Aunque parece redundante porque se representa dos veces lo mismo, no es así puesto que las instancias del 'power type' y las subclases *son* los mismos objetos.