

# Relaciones entre clases de análisis

# ¿Qué es una relación?

## ■ Definición

- ♦ Una conexión con significado entre elementos de modelado
- ♦ Forma de conectar unos elementos con otros

## ■ Ejemplos de relaciones vistas

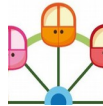
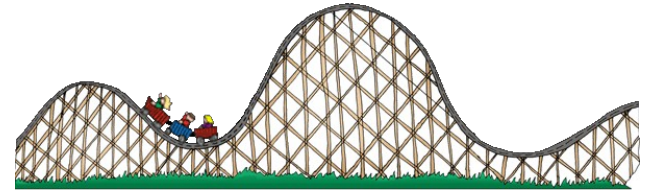
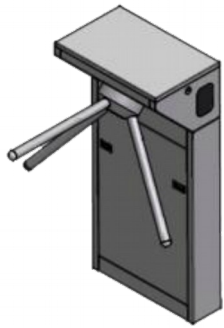
- ♦ Entre actores y casos de uso
- ♦ Entre casos de uso y casos de uso
- ♦ Entre actores y actores

## ■ Veremos

- ♦ Relaciones entre objetos (vínculo)
- ♦ Relaciones entre clases (asociaciones)

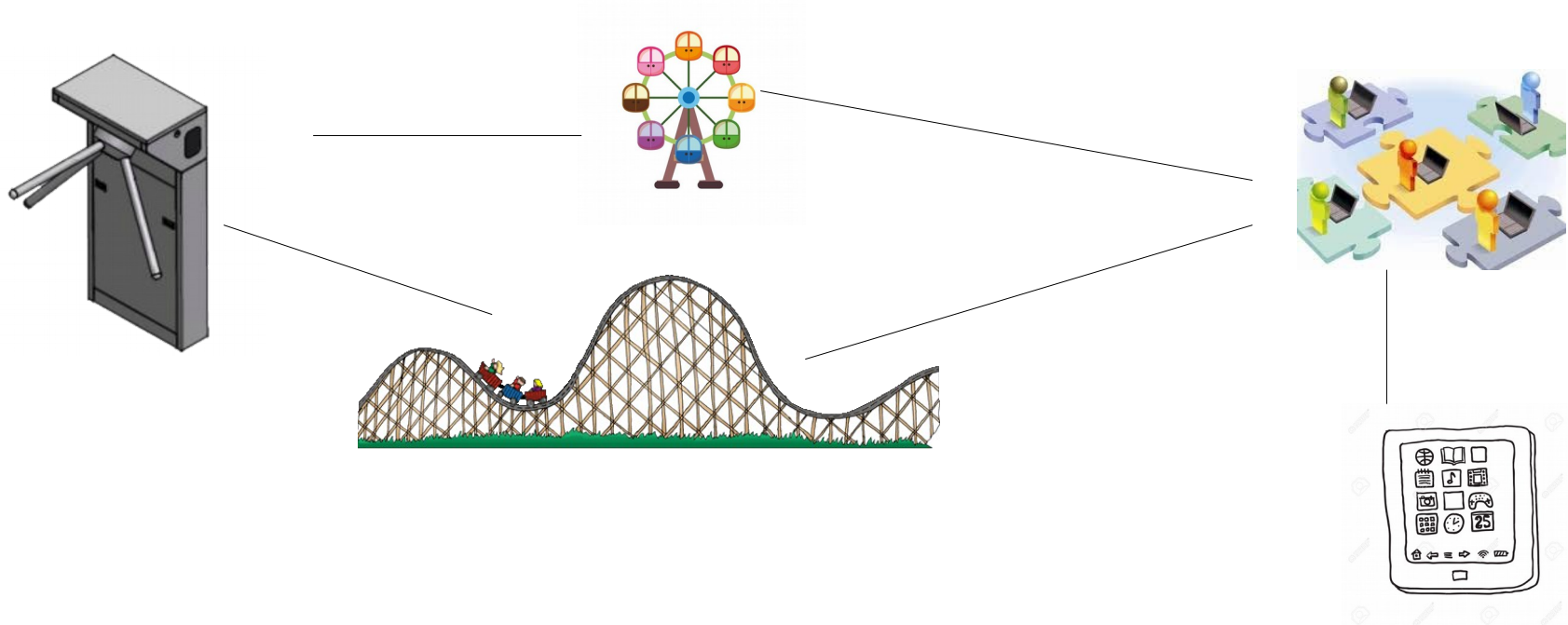
# Vínculos

- Crear un sistema orientado a objetos funcional
  - ◆ Los objetos no pueden existir aisladamente



# Vínculos

- Crear un sistema orientado a objetos funcional
  - ◆ Los objetos deben colaborar
  - ◆ Los objetos deben conectarse unos con otros
    - Proporcionar beneficio a los usuarios del sistema





# Vínculos

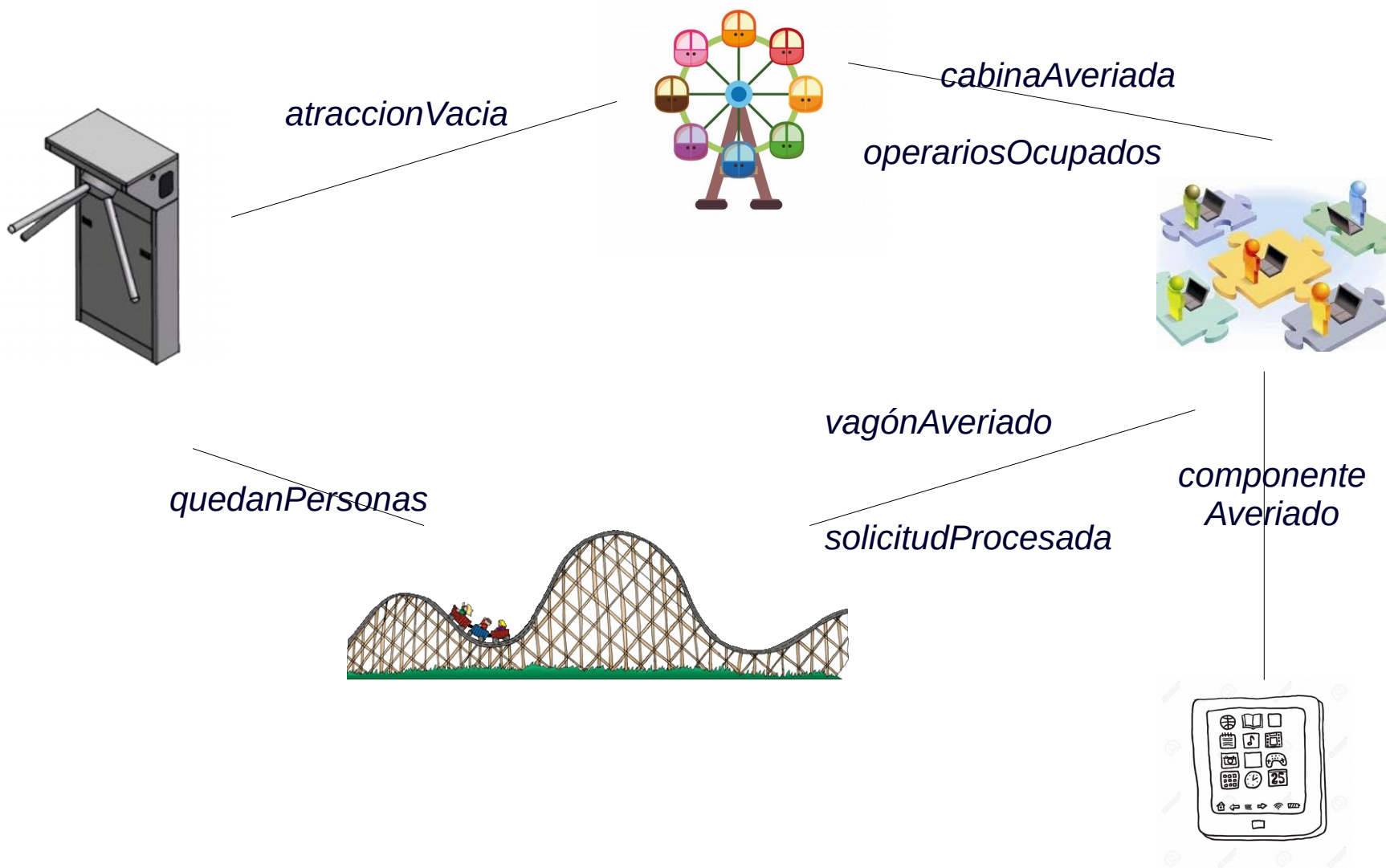
## ■ Vínculo (Link)

- ◆ Conexión entre dos objetos
- ◆ Forma que tienen los objetos de comunicarse



- ◆ Permiten mandar mensajes entre objetos
  - Al recibir un mensaje, un objeto invoca la operación correspondiente
- ◆ Se implementan mediante
  - Punteros o referencias

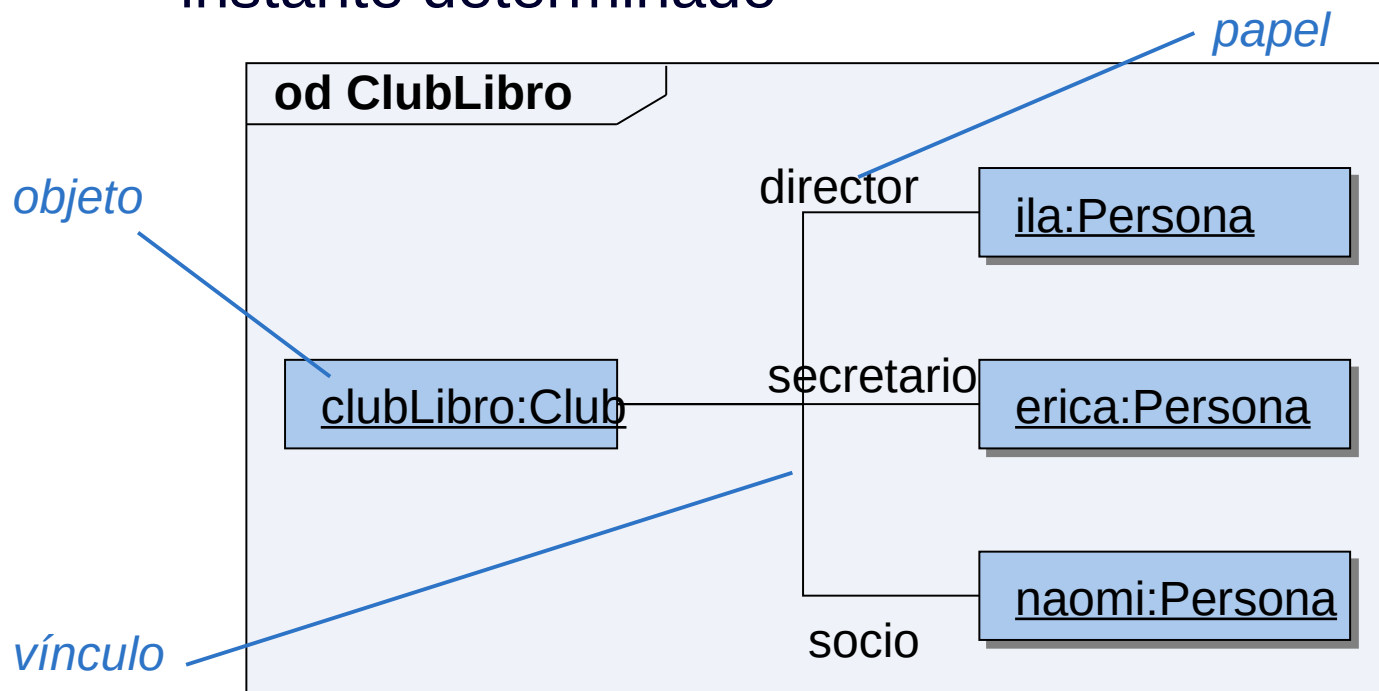
# Vínculos



# Diagrama de objetos

## ■ Qué muestra ...

- ◆ Los objetos y sus relaciones en un momento determinado
  - Instantánea de una parte de un sistema OO en un instante determinado



# Diagrama de objetos

## ■ Consideraciones

- ◆ El *rol* que desempeña un objeto en un diagrama de objetos es algo dinámico, puede cambiar con el tiempo
  - La secretaria puede pasar de erica a ila
- ◆ Los vínculos suelen conectar exactamente dos objetos
- ◆ Podemos especificar en que sentido se envían los mensajes usando la navegabilidad

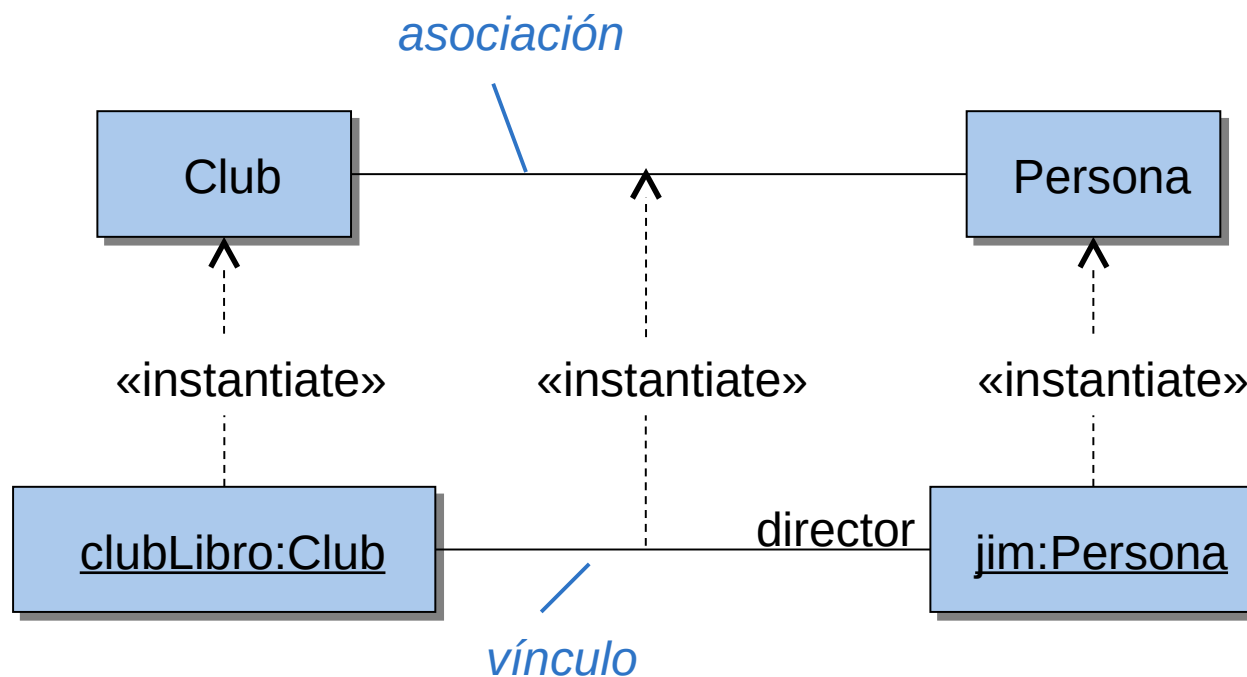




# Asociaciones

## ■ ¿Qué es?

- ◆ Relación entre clases
- ◆ Indican que existe un vínculo entre los objetos de esas clases

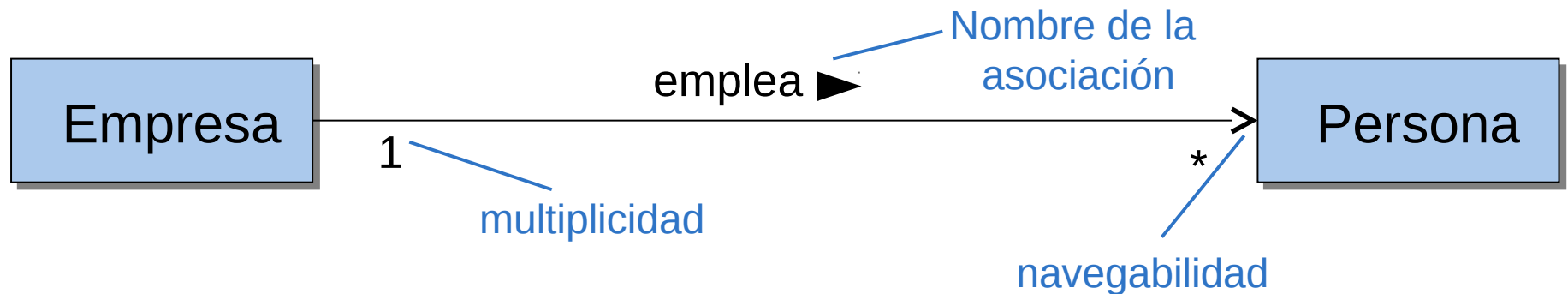


# Asociaciones

## ■ Sintaxis

### ◆ Nombre asociación

- Sintagmas verbales → representan acciones
- Punta de flecha para indicar en qué dirección leer la asociación
- lowerCamelCase

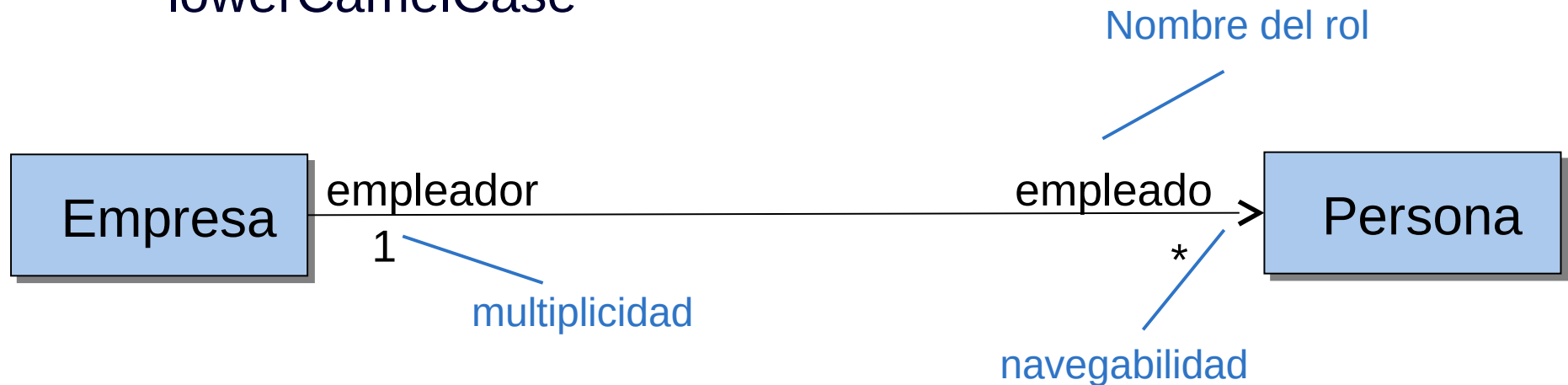


# Asociaciones

## ■ Sintaxis

### ◆ Nombre de los roles

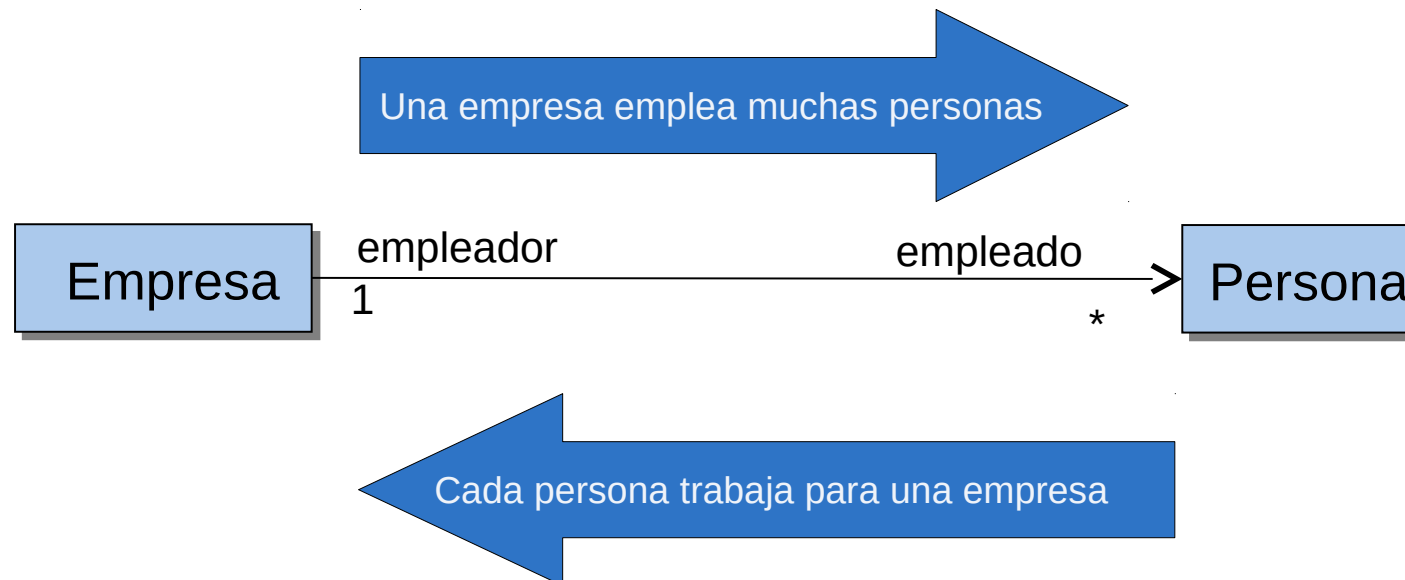
- En uno u ambos lados de la asociación
- Sintagmas nominales
- Representan los papeles que desempeñan lo objetos vinculados a la asociación
- lowerCamelCase



# Asociaciones

## ■ Multiplicidad

- ◆ Restricción que especifica el número de objetos de una clase que participan en una relación en *cualquier momento en el tiempo*



# Asociaciones

## ■ Multiplicidad

- ◆ Se expresa como un intervalos
  - mínimo..máximo
  - Números enteros o expresión entera
- ◆ Si no se indica explícitamente
  - Pendiente de determinar
  - No hay una multiplicidad por defecto

Multiplicidad mínimo..máximo	
0..1	Cero ó 1
1	Exactamente 1
0..*	Cero ó más
*	Cero ó más
1..*	1 ó más
1..6	1 a 6
2..7	2 a 7

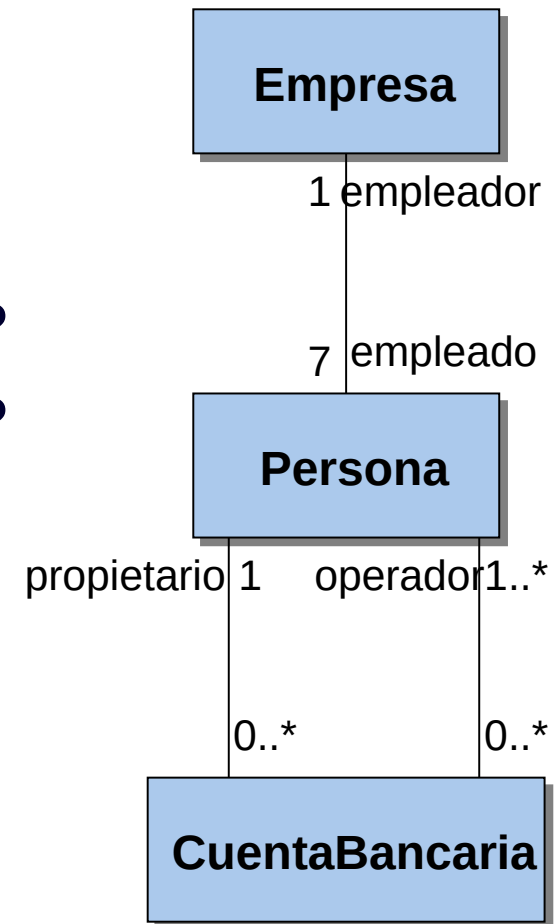


# Asociaciones

## ■ Ejercicio (I)

### ◆ ¿Cuántos

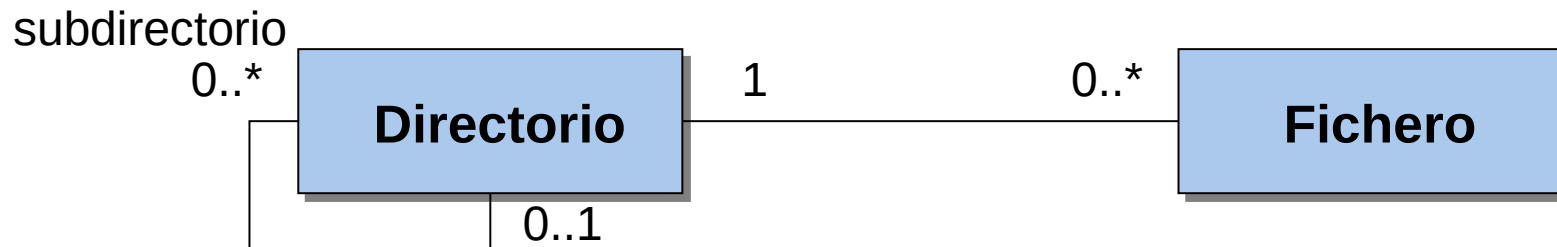
- empleados puede tener una empresa?
- empleadores puede tener una persona?
- propietarios puede tener una cuenta?
- operadores puede tener una cuenta?
- cuentas puede tener una persona?
- cuentas puede tener un operador?



# Asociaciones

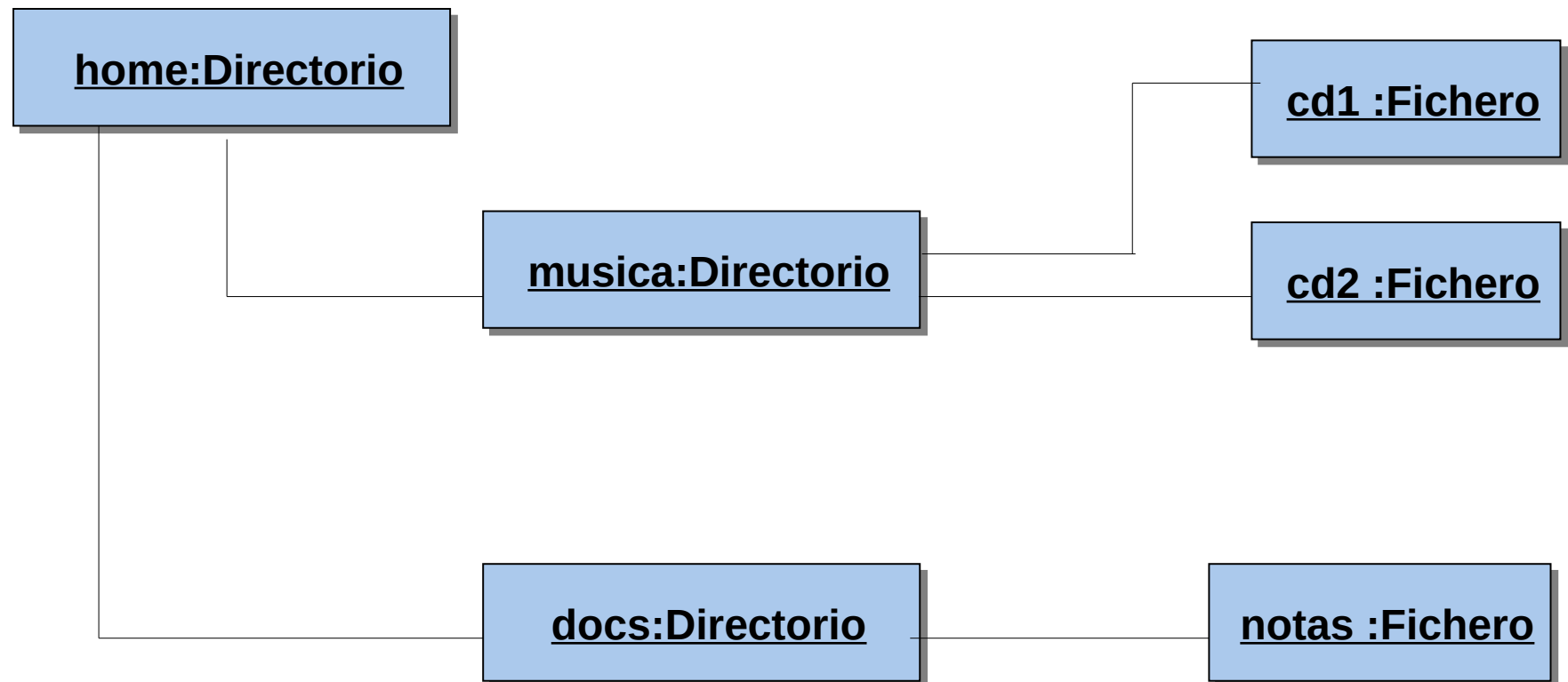
## ■ Asociación reflexiva

- ♦ Una clase tiene una relación consigo misma
- ♦ Los objetos de esa clase tienen vínculos a otros objetos de la misma clase



# Asociaciones

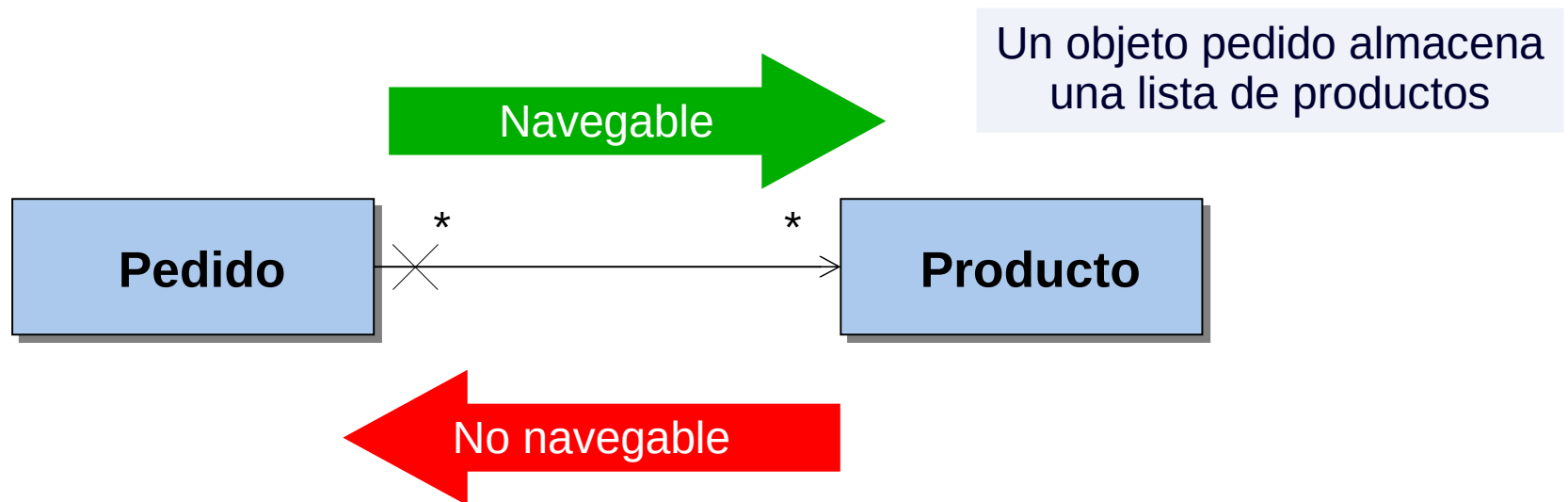
- Asociación reflexiva
  - ◆ Diagramas de objetos (ejemplos)



# Asociaciones

## ■ Navegabilidad

- ◆ Posibilidad de enviar mensajes solo en la dirección de la flecha
  - Minimizar el acoplamiento entre clases



Un objeto producto no almacena una lista de pedidos

# Asociaciones

## ■ Navegabilidad



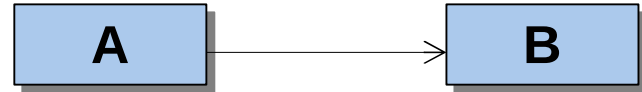


### ◆ Opciones representación

- Navegabilidad estricta
  - Indicar la navegabilidad de forma explícita
  - Se muestran todas las cruces y flecha
- Navegabilidad invisible
  - Se omiten todas las flechas y cruces
- Práctica estándar
  - Omitir todas las cruces (no navegabilidad)
  - Las asociaciones bidireccionales se indican sin flechas
  - Las asociaciones unidireccionales se indican con una flecha



# Asociaciones

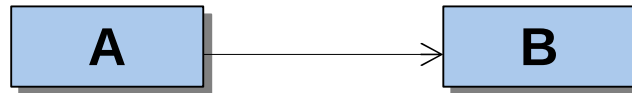
## ■ Navegabilidad

	Navegabilidad estricta	Navegabilidad invisible	Práctica estándar
	A a B → <input checked="" type="checkbox"/> B a A → <input checked="" type="checkbox"/>		
	A a B → <input checked="" type="checkbox"/> B a A → <input checked="" type="checkbox"/>		
	A a B → <input checked="" type="checkbox"/> B a A → ?		A a B → <input checked="" type="checkbox"/> B a A → <input checked="" type="checkbox"/>
	A a B → ? B a A → ?	A a B → ? B a A → ?	A a B → <input checked="" type="checkbox"/> B a A → <input checked="" type="checkbox"/>
	A a B → <input checked="" type="checkbox"/> B a A → <input checked="" type="checkbox"/>		

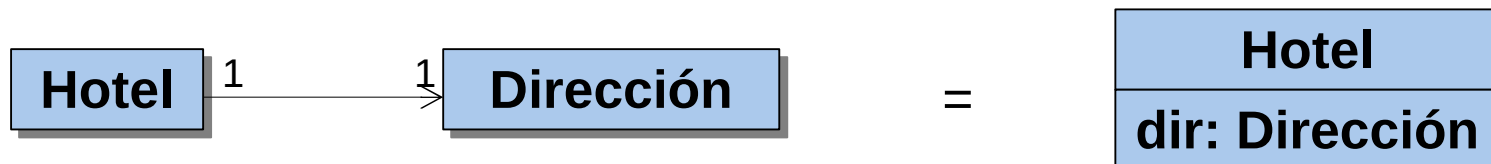
# Asociaciones

## ■ Interpretación

- ♦ Los objetos de A tienen un pseudo atributo de la clase B.



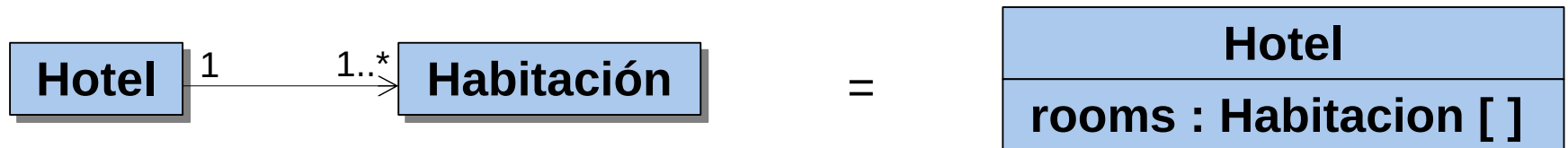
- ♦ Los lenguaje de programación no tienen soporte para asociaciones
  - Al generarse código (a partir de un diagrama), las asociaciones se convierten en atributos de la clase origen



# Asociaciones

## ■ Interpretación

- ◆ Si el destino tiene multiplicidad mayor que 1
  - Atributo tipo array
  - Atributo de algún tipo colección



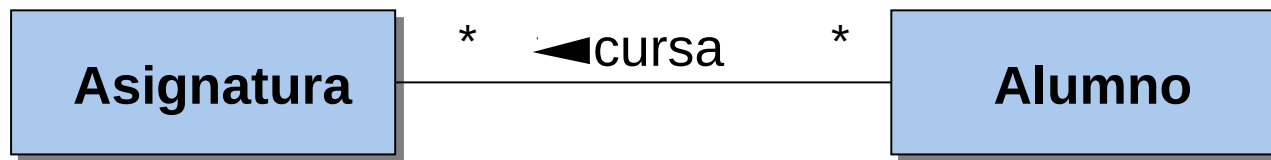
# Asociaciones

## ■ Interpretación

- ◆ ¿Cómo modelamos? ¿Asociación o atributo?
  - Atributo
    - La clase destino no es una parte importante del modelo
    - La clase destino es algún tipo de detalle de implementación
  - Asociación
    - La clase destino es una parte importante del modelo
    - La clase destino es una clase diseñada por nosotros y debe mostrarse en el modelo

# Asociaciones

## ■ Clase de asociación



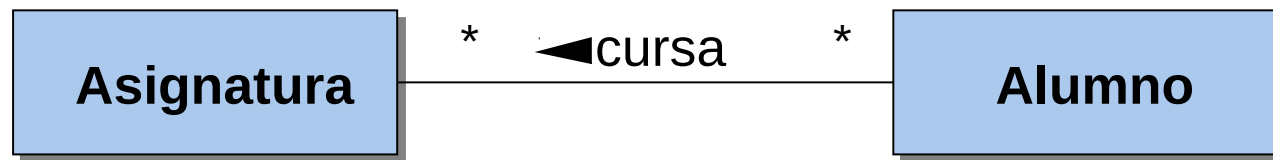
- ◆ Cada objeto ***Alumno*** puede cursar varios objetos ***Asignatura***
- ◆ Cada objeto ***Asignatura*** puede tener matriculados varios objetos ***Alumno***
- ◆ Cuando un objeto ***Alumno*** cursa un objeto ***Asignatura***, el ***Alumno*** tiene una calificación

¿Dónde almacenamos la calificación del alumno?



# Asociaciones

## ■ Clase de asociación

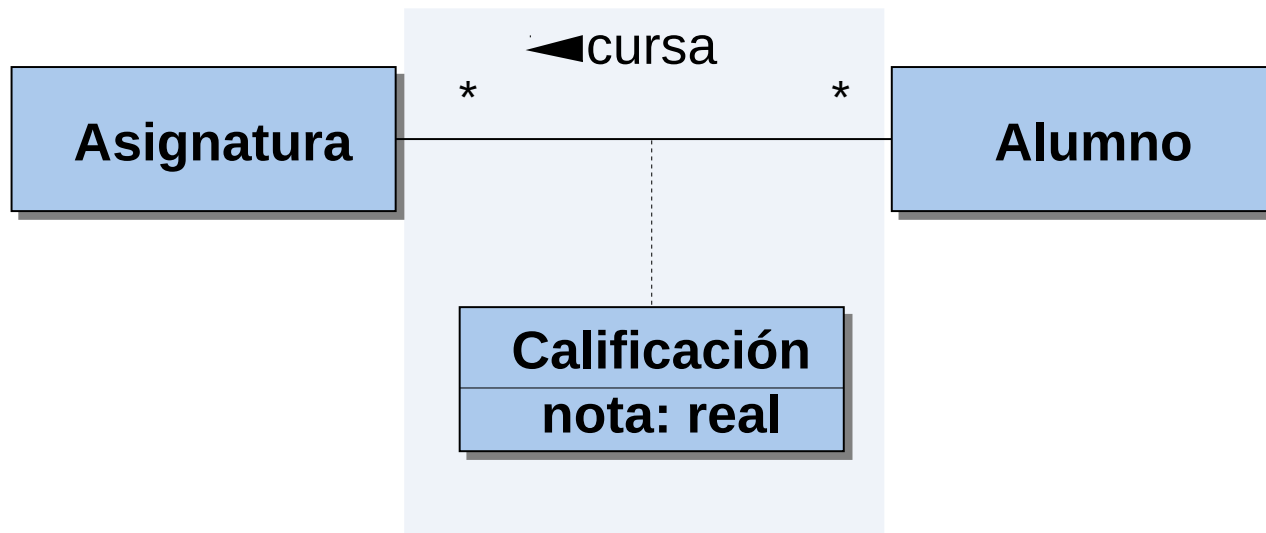


- ♦ En la clase Alumno no porque hay una calificación diferente para cada asignatura cursada
- ♦ En la clase Asignatura no porque diferentes objetos Alumno tienen diferentes calificaciones
- ♦ La calificación es una propiedad de la relación *curso*

¿Dónde almacenamos la calificación del alumno?

# Asociaciones

## ■ Clase de asociación

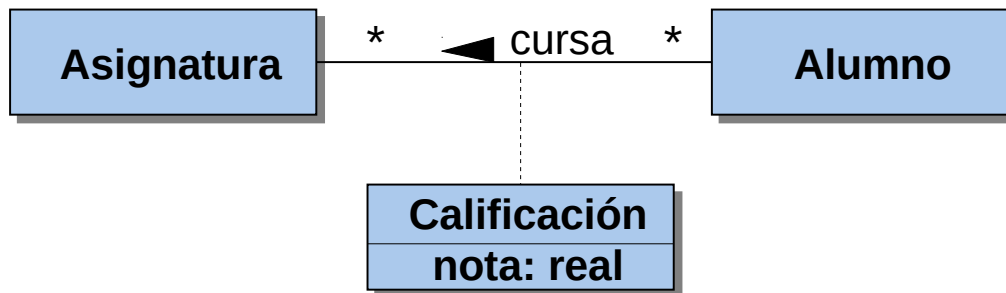


- ◆ Formada por la clase, la asociación y la línea discontinua
- ◆ Restricción
  - Sólo puede existir un vínculo entre dos objetos específicos

# Asociaciones

## ■ Clase de asociación

- ◆ Un Alumno concreto puede tener solo una calificación con una Asignatura concreta



## ■ Asociación cosificada (reified association)

- ◆ Un Alumno concreto puede tener varias calificaciones con una asignatura concreta

