

Diagramas de clases

OMG Unified Modeling Language Specification



March 2003
Version 1.5
Formal/03-03-01



Diagrama de clases

- Diagrama que contiene “elementos clasificadores” conectados por relaciones estáticas.
- Elementos clasificadores:
 - clases
 - Tipos de datos
 - Interfaces
- los distintos diagramas de clases no se corresponden con particiones del sistema

Clase:

- descriptor de un conjunto de objetos que comparten estructura, comportamiento y relaciones similares
- concepto del sistema a modelizar
- se representa mediante un rectángulo con hasta tres compartimentos:
 - nombre de la clase
 - lista de atributos (opcional)
 - lista de operaciones (opcional)

Ventana

Ventana

tamano:Area
visible:Boolean

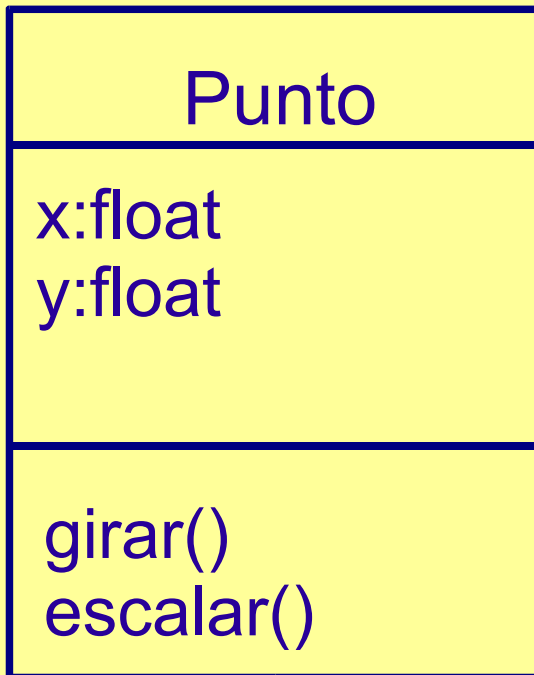
mostrar()
ocultar()

Diagramas de objetos

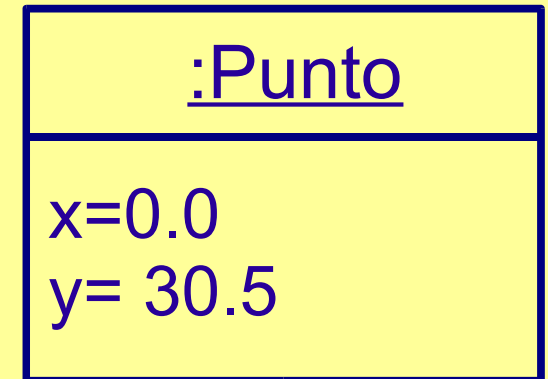
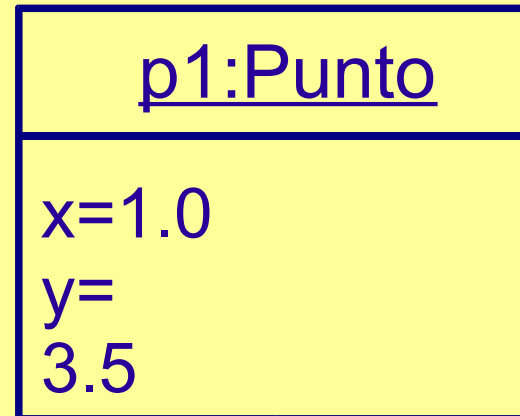
- Un diagrama de clases también podría contener instancias:
 - objetos
 - Enlaces
- Un diagrama que sólo contuviera objetos sería un diagrama de objetos.

Objetos

- son instancias particulares de una clase.
- Tienen identidad y atributos
- Representación:



Clase



Objetos

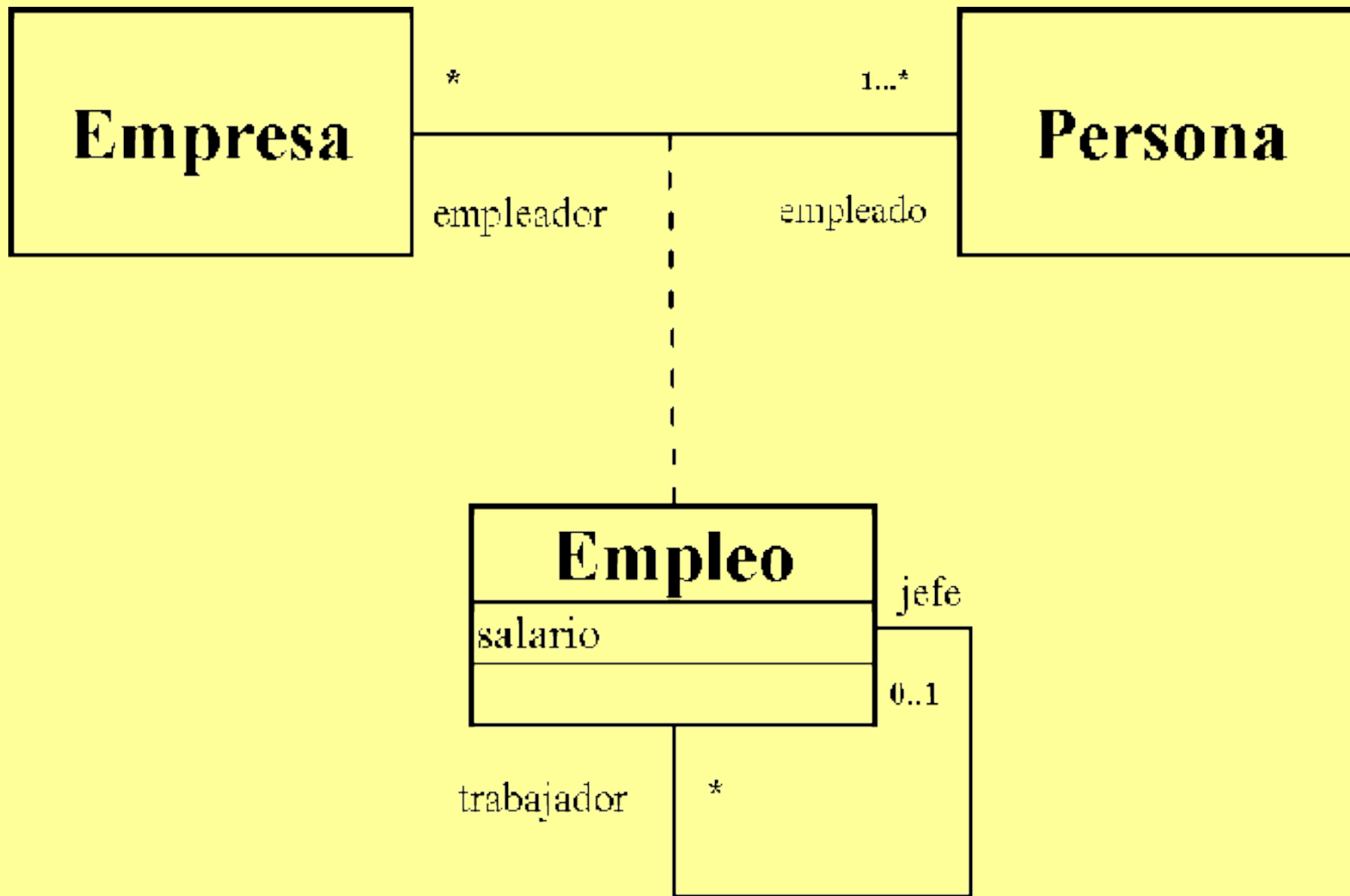
Asociación binaria

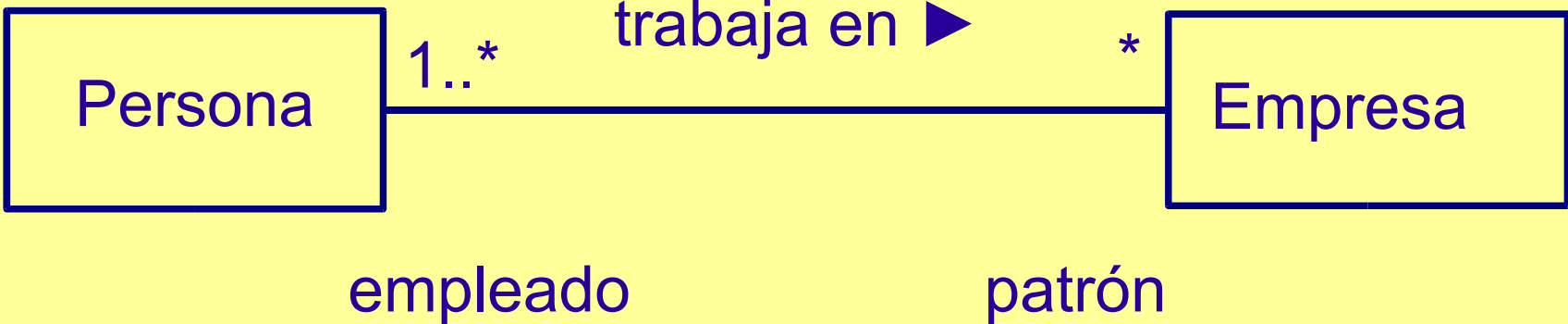
- asociación entre dos clasificadores
- un clasificador puede asociarse consigo mismo
- se denotan mediante un camino continuo entre los dos clasificadores
- la mayoría de la información relevante de una asociación están ligadas a sus extremos (associations ends)



el camino de una asociación puede tener las indicaciones siguientes:

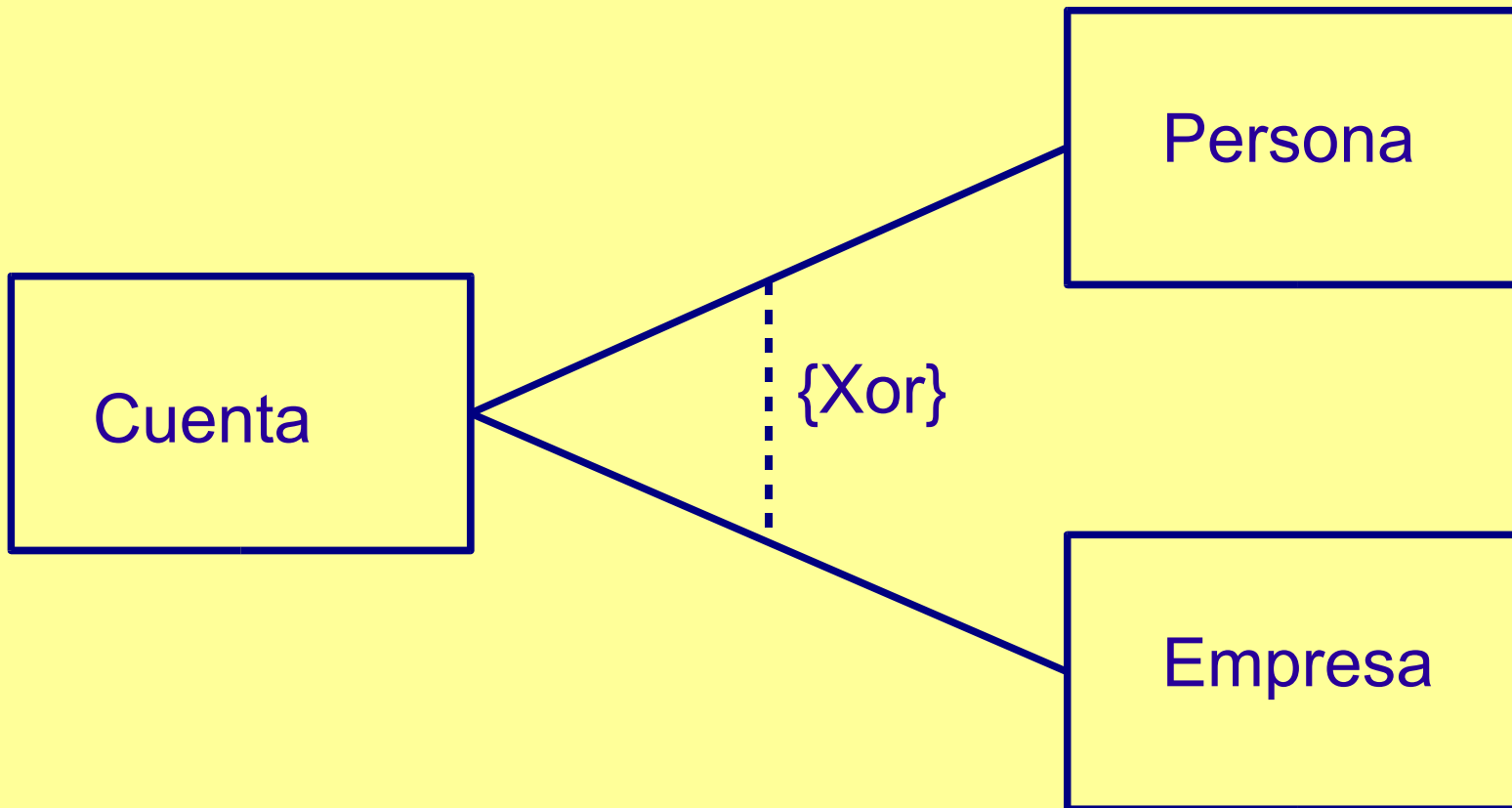
- nombre (se le puede asociar el símbolo ► para indicar el sentido en el que debe leerse la asociación)
- símbolo de clase asociación:
 - para cuando la asociación tenga propiedades de clase (atributos, operaciones, ..)
 - esta clase se une al camino de la asociación mediante una línea discontinua.





asociación Xor :

para una instancia sólo una de entre varias asociaciones posibles podrá instanciarse



Propiedades de la asociación

relativas a sus extremos:

- multiplicidad
- orden (multiplicidad > 1): {unordered} (defecto), {ordered}
- cualificador
- navegabilidad
- indicador de agregación
- role
- modificabilidad: nada, {frozen}, {addOnly}
- Visibilidad: '+', '#', '-' , explícitamente {public} p.e.

Multiplicidad:

- Limite_inferior..limite_superior
- * En el límite superior --> sin límite

- Ejemplos:

0..1

1

0..*

*

1..*

1..6

1..3,7..10,15,19..*

- Es preferible

1..3,7,10

a

7,10,1..3

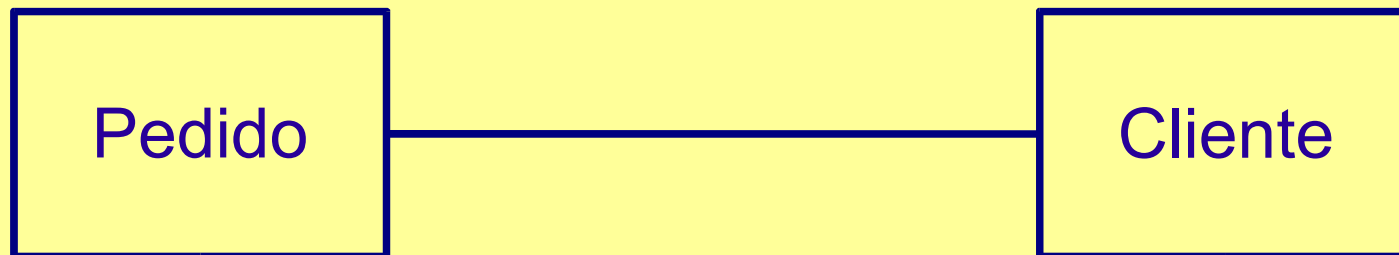
0..1

a

0,1

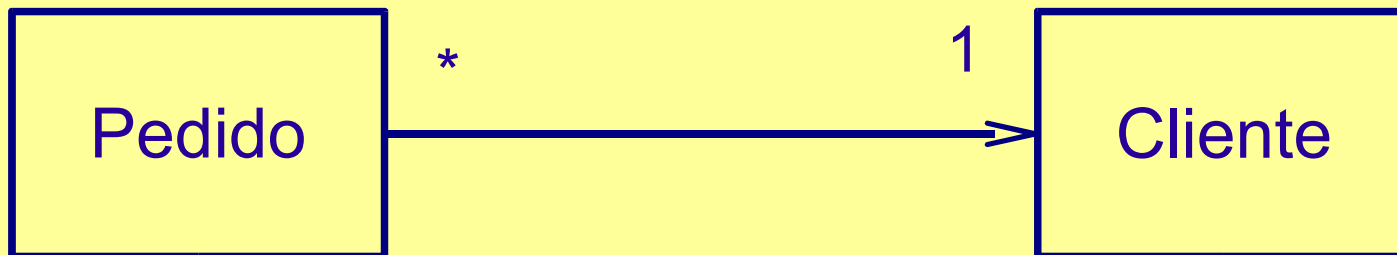
Navegabilidad

- Una punta de flecha del lado del extremo hacia el que hay navegabilidad
- Puede estar a un lado, a ambos o a ninguno



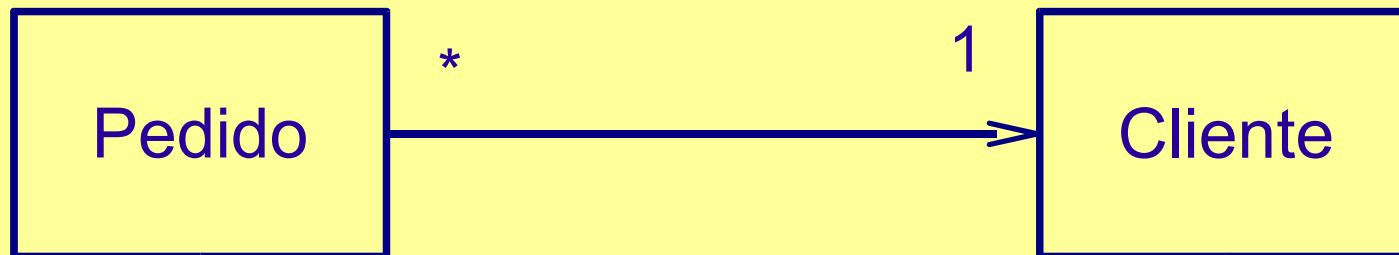
Navegabilidad

- Un Pedido *tendrá la responsabilidad de saber* cuál es su Cliente
- Un Cliente *no tiene porqué saber* cuales son sus pedidos



Navegabilidad

- Un Pedido podría poseer un puntero que permita acceder a su Cliente
- Un Cliente no tendrá punteros que permitan acceder a los Pedido en que aparezca

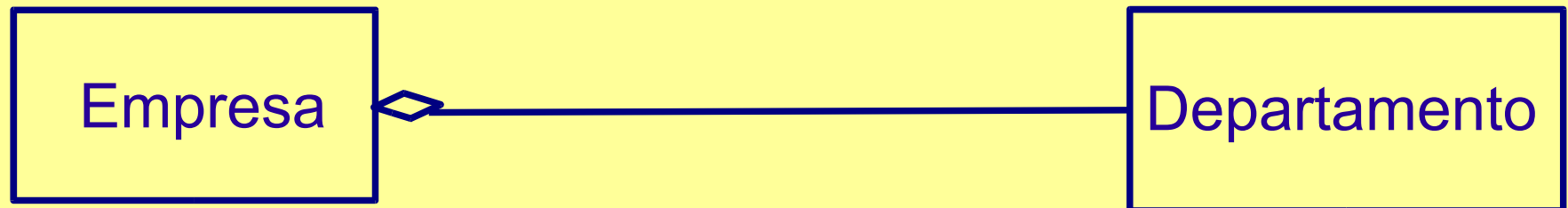


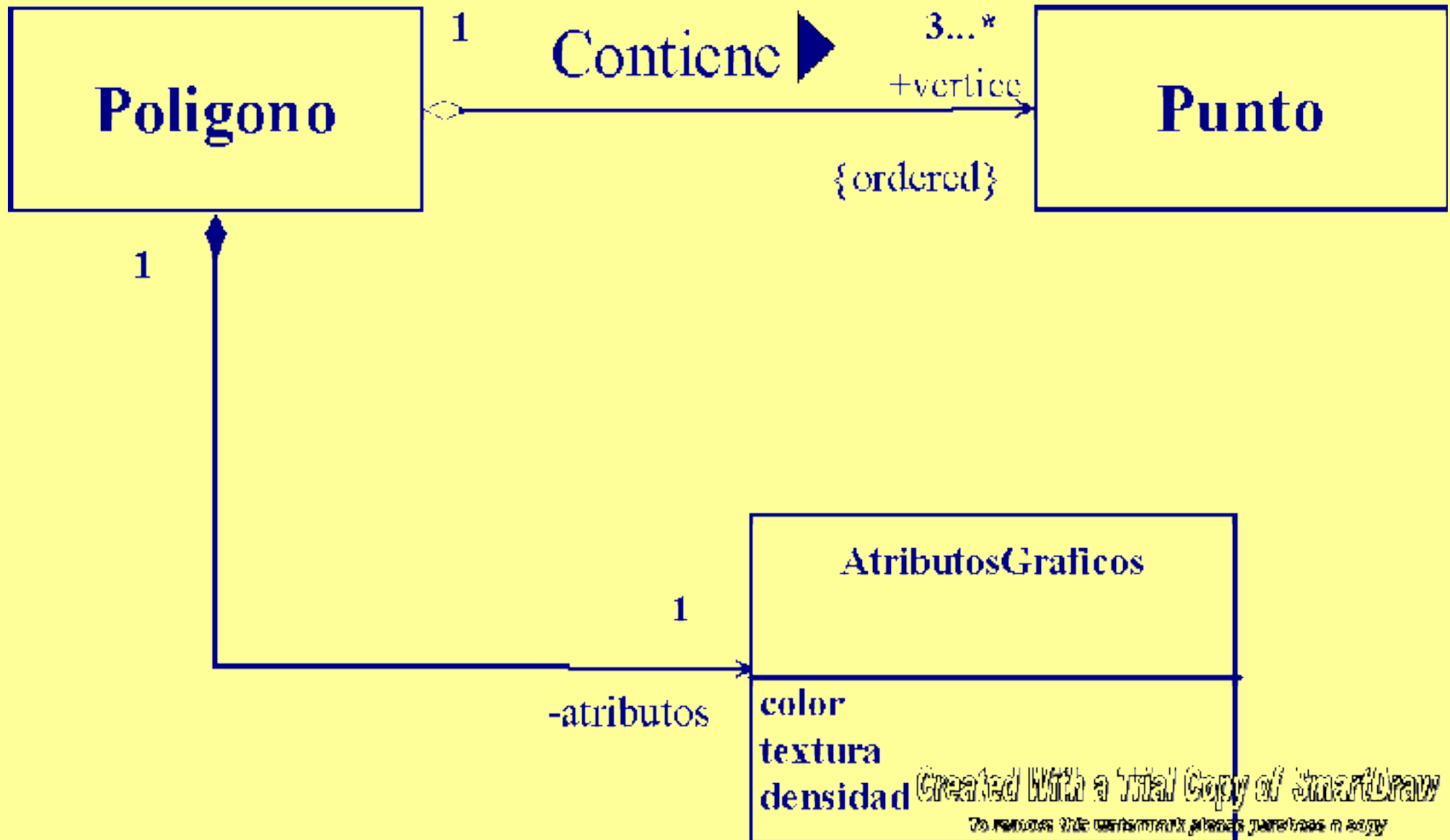
Indicador de agregación:

- Es opcional pero no suprimible
- nunca a ambos extremos
- un diamante vacío del lado del agregado
- si el diamante aparece lleno éste indica composición

Agregación (cont):

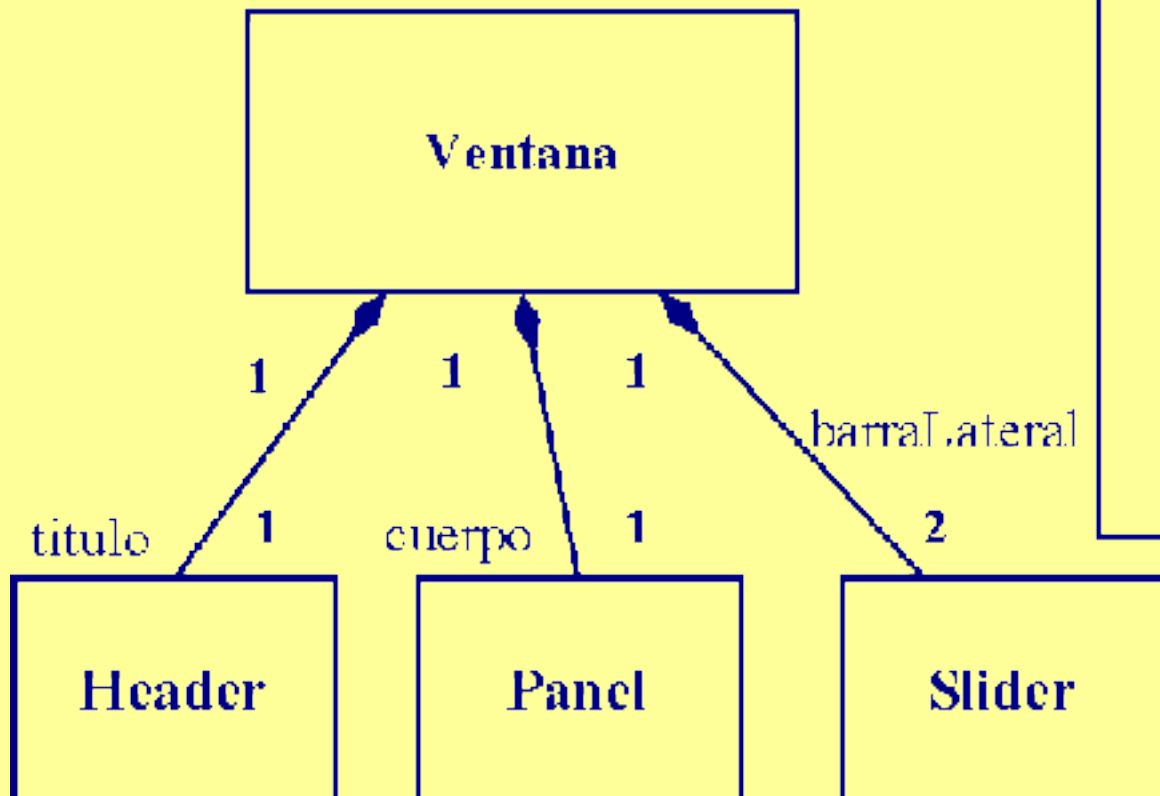
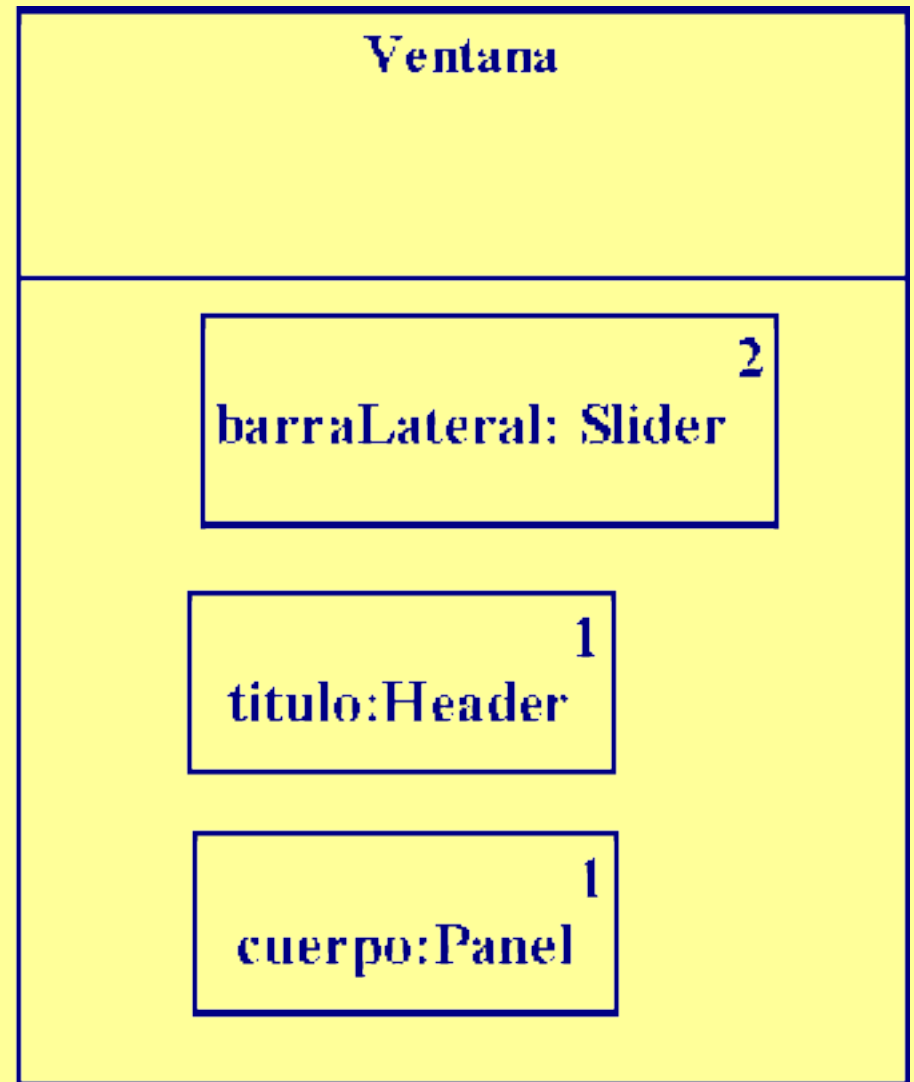
- Representa una relación del todo a la parte, una relación *tiene un*, o al revés (de la parte al todo) *está contenido en*
- Algunos piensan que se puede prescindir de ellas en los diagramas de clases (que basta con representarlas como asociaciones)





Composición

- es una versión fuerte de la agregación en la que:
 - el agregado (contenedor) sólo puede pertenecer a una instancia de la asociación.
 - El objeto compuesto es responsable del conjunto (*las partes viven y mueren con el todo*)
 - el agregado no puede tener multiplicidad superior a 1.
- se indica mediante un diamante negro
- para que una asociación forme parte de un compuesto el camino entre sus extremos no debe traspasar los límites del compuesto
- hay varias formas alternativas de representar la composición

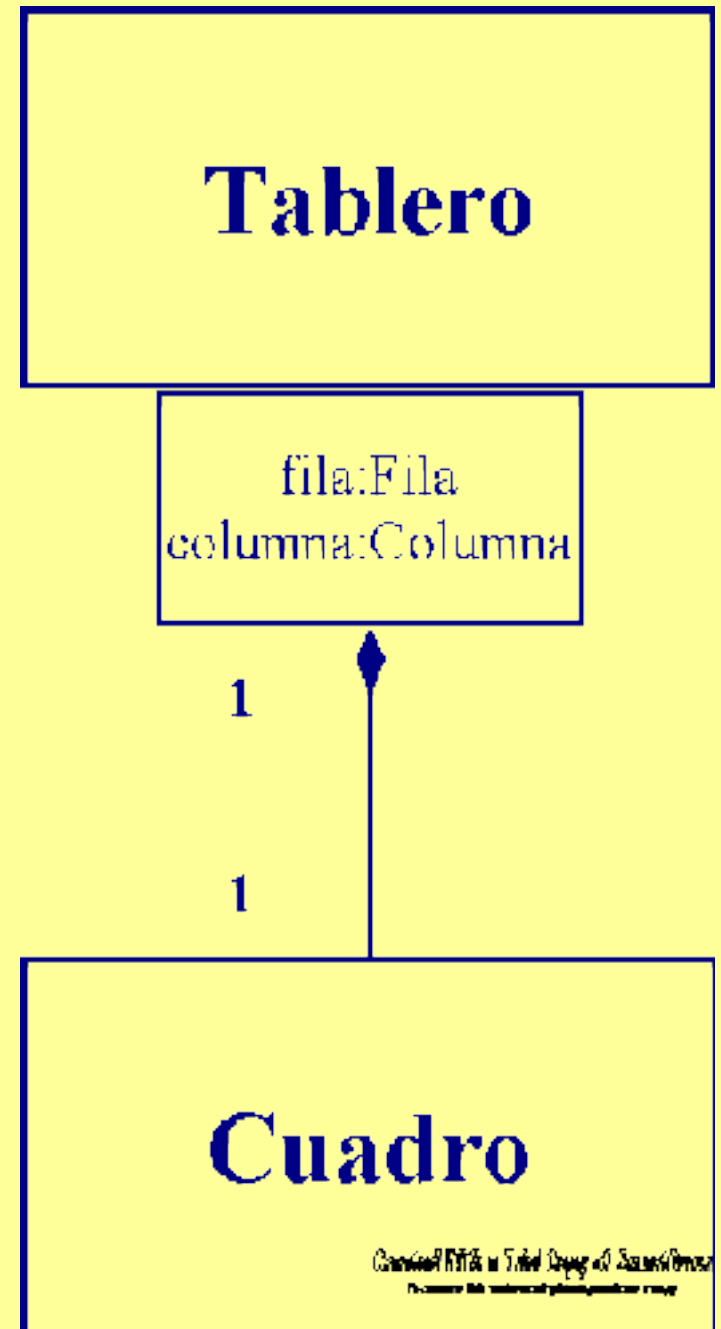
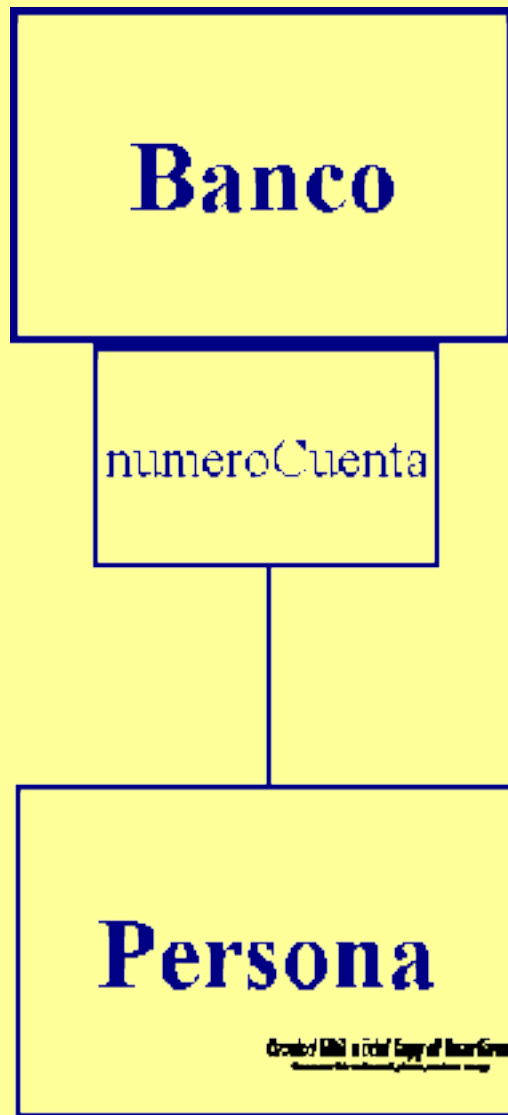


Cualificador:

- Atributo o lista de atributos cuyos valores sirven para distinguir las distintas instancias que participan en una asociación
- es una propiedad de la asociación
- se indica mediante un rectángulo adosado al extremo (origen) de la asociación
- una instancia del clasificador origen, junto con un valor del cualificador denota una partición del conjunto de instancias del clasificador destino

Cualificador(cont):

- las posibles multiplicidades del extremo destino indican:
- 0..1 se puede seleccionar un solo valor , pero no todo valor del cualificador selecciona instancias destino
- 1 cada valor selecciona una instancia (el conjunto de valores finito)
- * el cualificador es un índice que divide en subconjuntos al conjunto de instancias destino



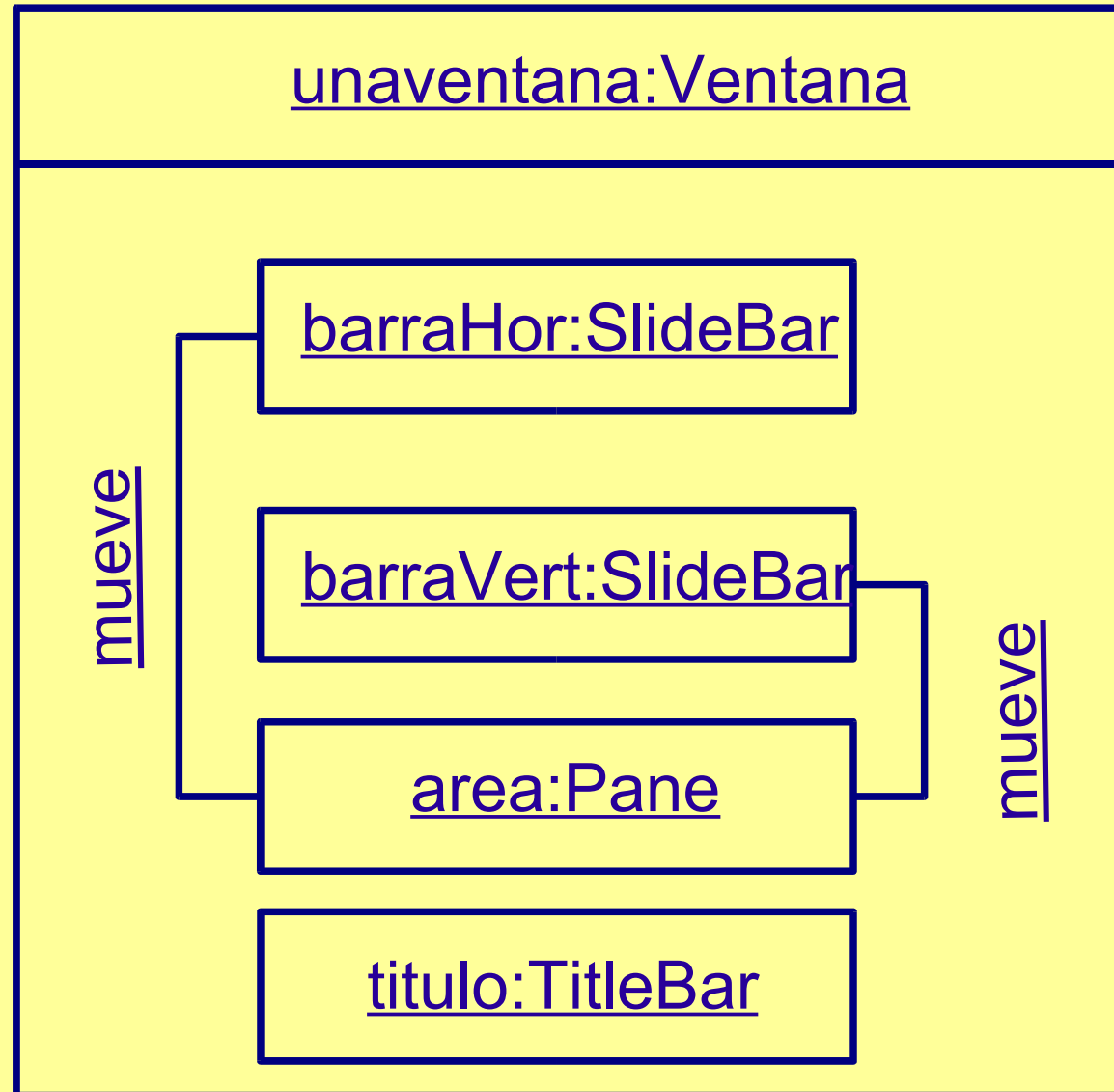
asociaciones N-arias:

- una asociación entre tres clasificadores o más
- una instancia de la asociación sería una tupla de instancias de los clasificadores implicados
- un clasificador puede aparecer más de una vez
- el significado de la multiplicidad sería el posible número de instancias de la asociación una vez fijadas las instancias de los otros $N-1$ clasificadores.
- Ninguno de los participantes en la relación N-aria puede tener el indicador de agregado.



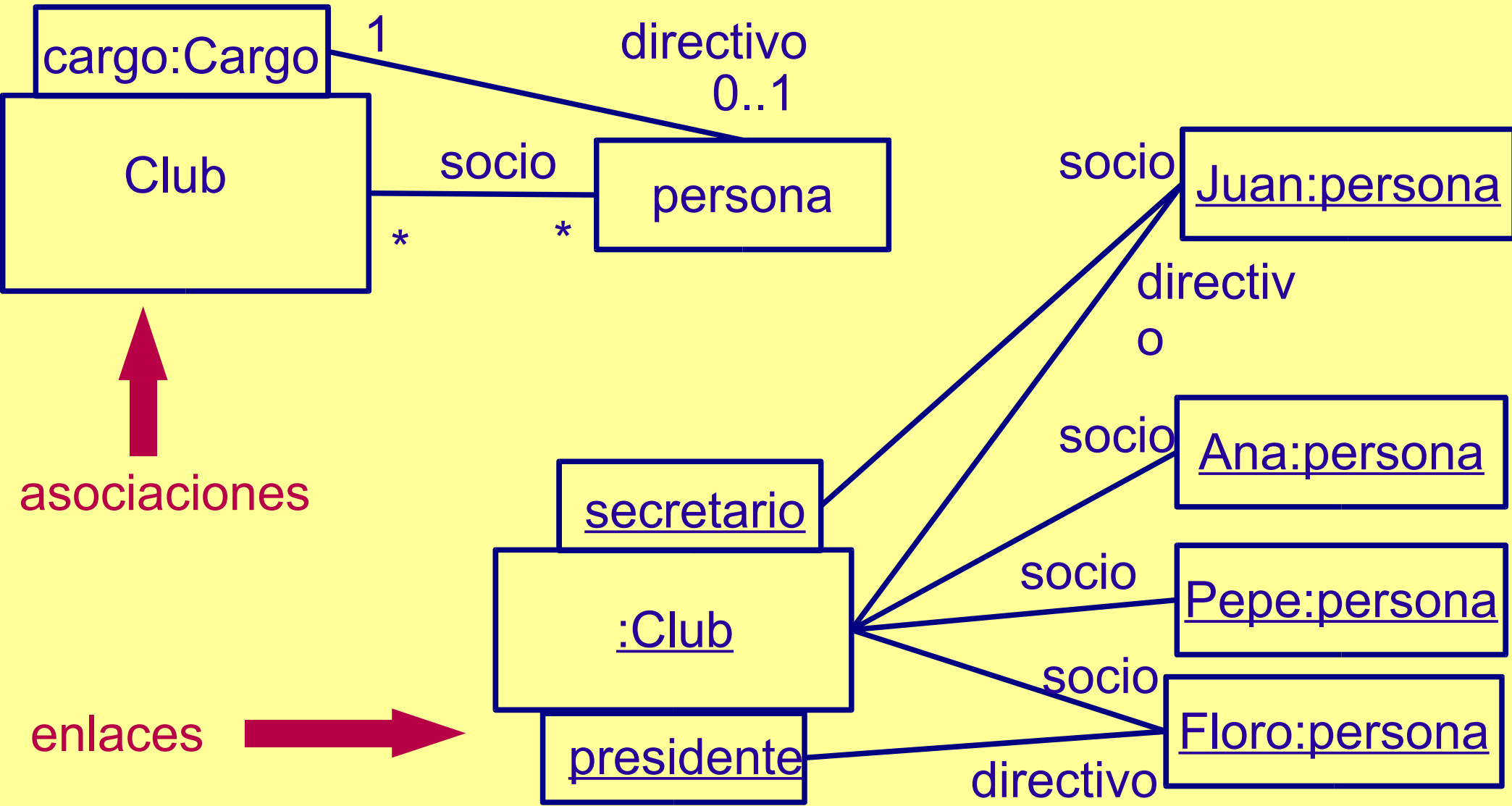
Objetos compuestos

Los objetos de clases compuestas (agregaciones) pueden representarse indicando una lista de atributos



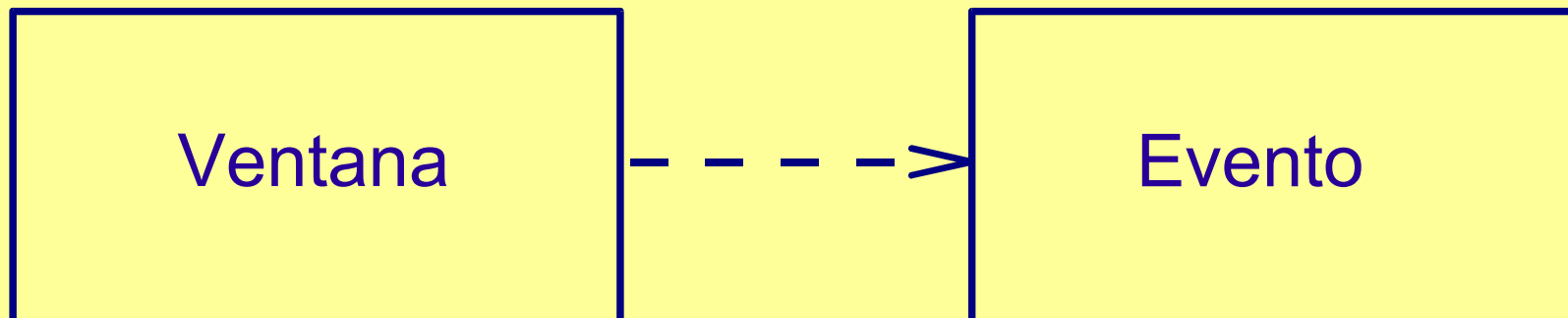
Enlace (Link)

- instancia de una asociación
- es una tupla (lista) de referencias a objetos
- Su representación no debe indicar el nombre de la asociación, o si lo hace debe estar subrayado (indicando instanciación)



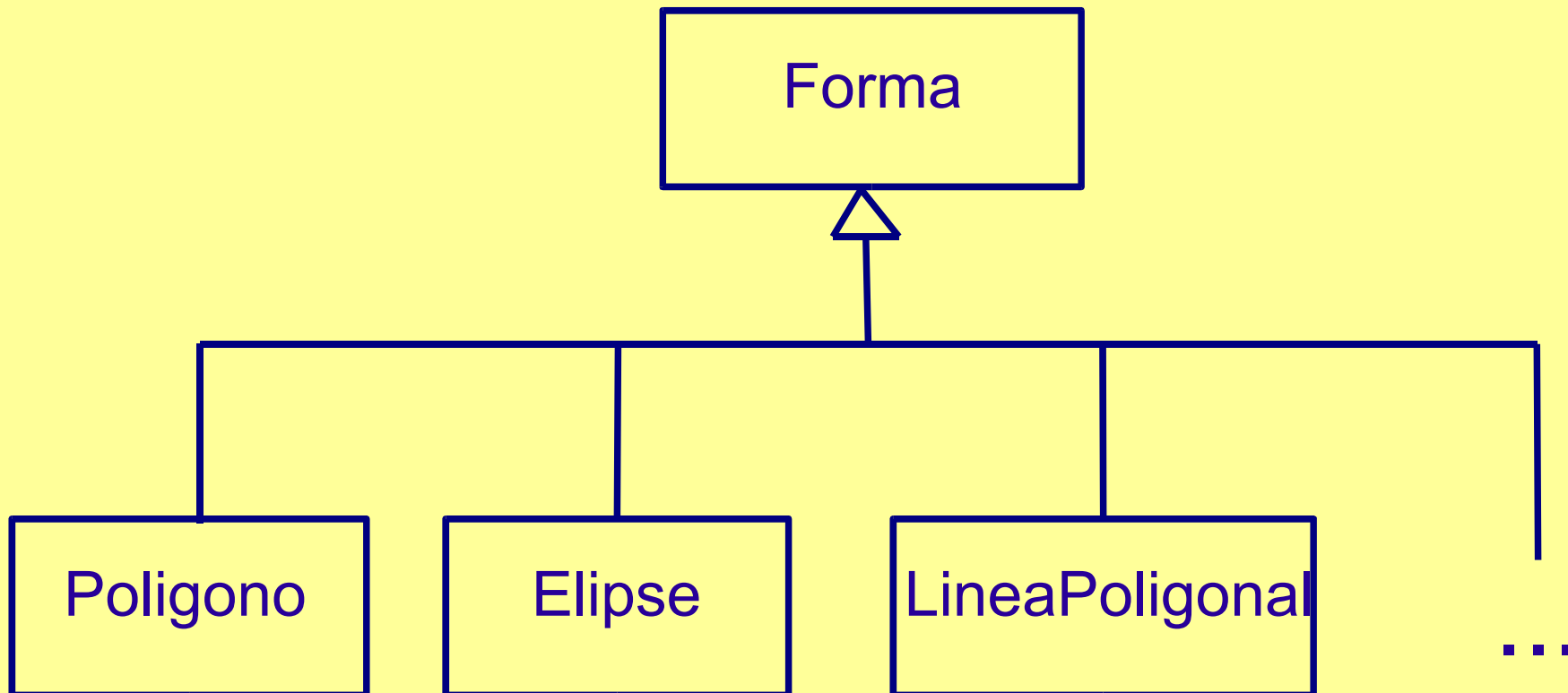
Dependencia

- Una *dependencia* es una relación de uso, que declara que un cambio en la especificación de un elemento (por ejemplo la clase Evento) puede afectar a otro elemento que la utiliza (por ejemplo la clase Ventana), pero no necesariamente a la inversa
- Una dependencia quiere decir que un elemento utiliza a otro. puede usarse para clases, paquetes, casos de uso y asociaciones.
- Se denota mediante un camino discontinuo.
- la flecha va dirigida hacia el elemento del cual se depende



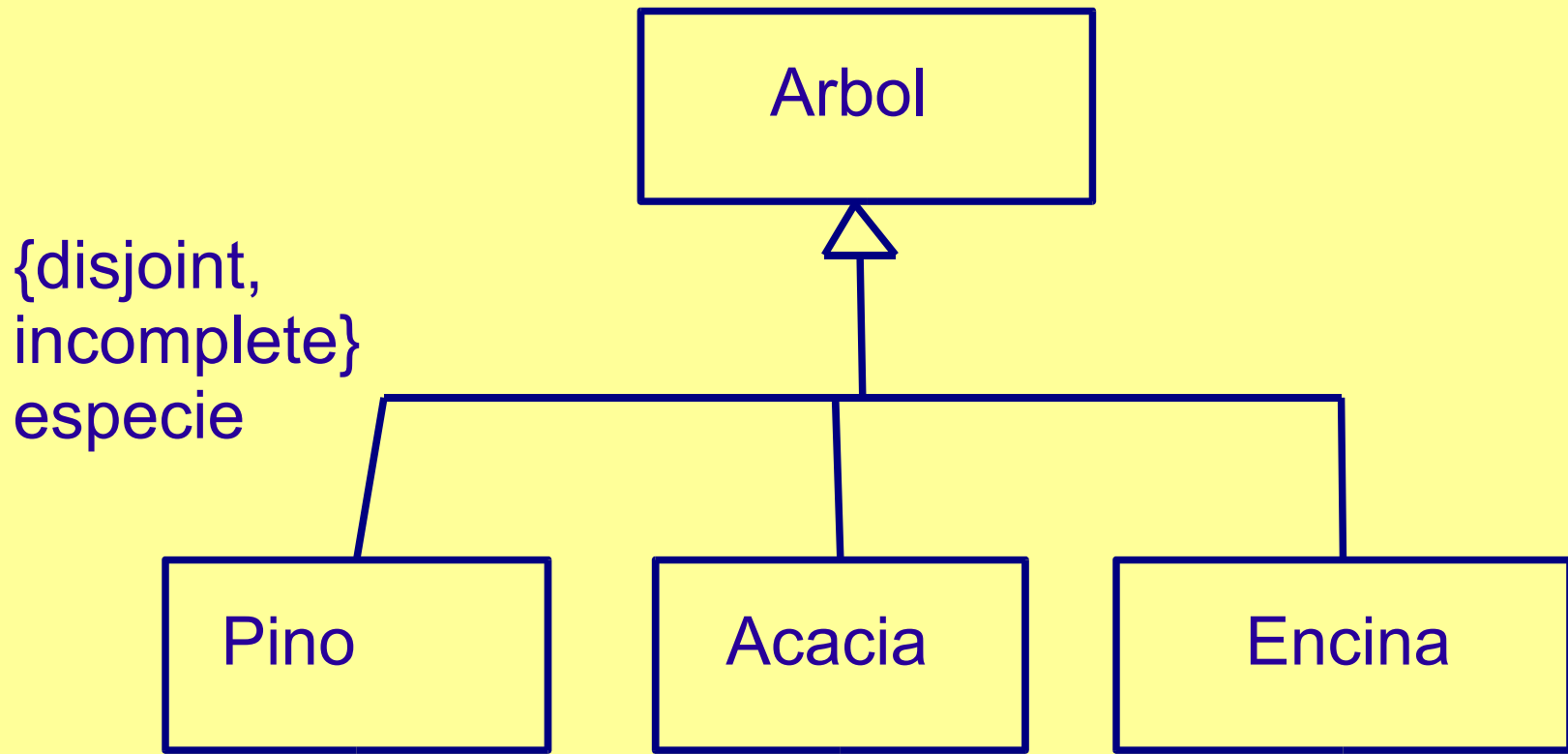
Generalización

- puede usarse para clases, paquetes, casos de uso y asociaciones.
- Se denota mediante un camino continuo que tiene una punta de flecha triangular vacía junto al elemento más general (ancestro).



Generalización

- la existencia de “hijos” no presentes en un diagrama se puede indicar mediante “...” en su lugar.
- Pueden indicarse las siguientes restricciones sobre los hijos
 - **overlapping:** un elemento puede tener dos o más hijos como ancestros . Una instancia puede ser una instancia directa o indirecta de dos o más hijos.
 - **Disjoint:** un elemento no puede tener dos o más hijos como ancestros . Una instancia no puede ser una instancia directa o indirecta de dos o más hijos.
 - **Complete:** toda instancia debe pertenecer a una de las subclases.
 - **Incomplete:** puede haber instancias que no pertenezcan a alguna de las subclases.



Generalización

- El camino puede ir acompañado de un discriminante que indica una partición de los hijos.
- El discriminante debe ser único entre los atributos y papeles del padre.
- El mismo discriminante puede aparecer más de una vez (indican que los hijos pertenecen a una misma partición).
- Es indiferente usar una punta de flecha con caminos divergentes o un camino independiente para cada hijo.

