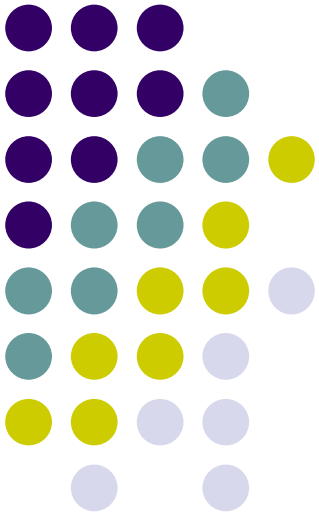
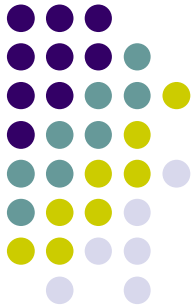
Metodologías de Diseño y Programación

Análisis

Modelado del Dominio

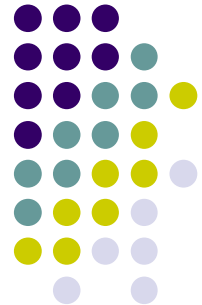


Contenido



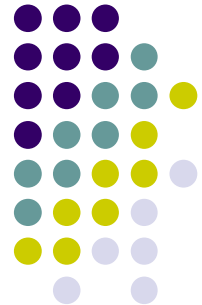
- Introducción
- Modelo de Dominio
- Conceptos
- Asociaciones
- Atributos
- Generalizaciones
- Otros elementos
- Restricciones

Introducción



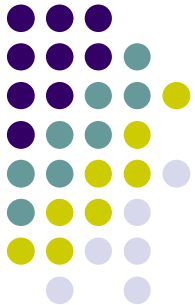
- La esencia del análisis orientado a objetos es la descomposición del problema en conceptos individuales
- Un Modelo de Dominio contiene principalmente los conceptos y sus relaciones que sean significativos en el dominio del problema
  - Significativos para el modelador
  - El problema y los requerimientos determinan qué es significativo

Modelo de Dominio



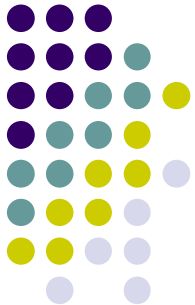
- Está enfocado en conceptos del dominio y no en entidades de software
- Contenido
  - **Introducción:** Breve descripción que sirve como introducción al modelo
  - **Conceptos:** Clases que representan conceptos significativos presentes en el dominio
  - **Tipos:** Data types que describen propiedades de las clases que representan conceptos

Modelo de Dominio (2)



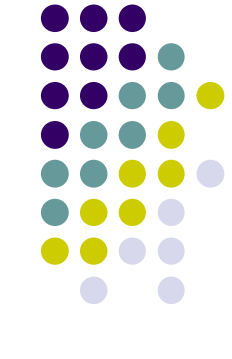
- Contenido (cont.)
  - **Relaciones:** Relaciones de asociación o generalización entre las clases que representan conceptos
  - **Restricciones:** Expresiones que restringen las posibles instancias de los conceptos del modelo
  - **Diagramas:** Representaciones (usualmente uno solo) de conceptos, tipos y relaciones presentes en el modelo

Conceptos



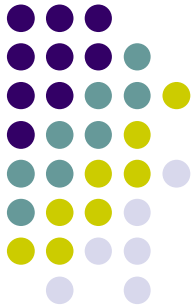
- Un concepto es una idea, cosa u objeto. Puede ser considerado en términos de
  - **Símbolo:** Imagen que representa al concepto
  - **Intención:** La definición del concepto
  - **Extensión:** El conjunto de ejemplos al que el concepto aplica

Conceptos (2)



Símbolo	Intención	Extensión
«domain class**»** **Venta** fecha hora importe	Acto de intercambiar dinero por un producto	v1 : Venta v2 : Venta ▪ ▪ ▪

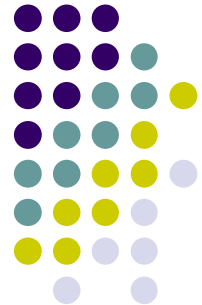
Conceptos (3)



- En el Modelo de Dominio todo concepto es una clase tal que
  - Tiene aplicado el estereotipo «domain class»: significa que sus únicas propiedades son
    - Nombre
    - Atributos
  - El valor del tagged value “documentation” es usualmente la *intención* del concepto

Conceptos

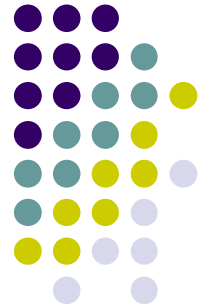
Identificación de Conceptos



- Es muy común omitir conceptos en esta fase (identificación) que pueden ser descubiertos en una fase o etapa posterior
  - Al descubrirlos se los agrega al Modelo de Dominio
- Es posible encontrar conceptos interesantes que no tengan atributos (que tengan un rol de comportamiento más que de información)
- Comenzar la construcción de un Modelo de Dominio haciendo una lista de conceptos candidatos
- Existen dos técnicas para ello
  - Lista de categorías de conceptos
  - Identificación de sustantivos

Conceptos

Identificación de Conceptos (2)

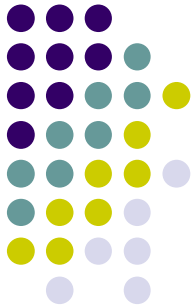


- Lista de categorías de conceptos
  - Consiste en repasar la lista de categorías de conceptos buscando los conceptos del dominio del problema que apliquen a cada categoría

Categoría	Ejemplo
Objetos físicos o tangibles	Avión
Descripciones de cosas	DescripcionVuelo
Lugares	Aeropuerto
Transacciones	Reserva
Roles	Piloto

Conceptos

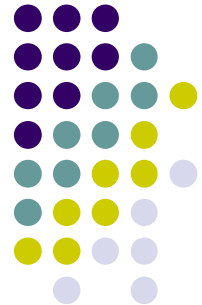
Identificación de Conceptos (3)



- Lista de categorías de conceptos (cont.)
  - La lista se puede continuar con
    - contenedores de cosas
    - cosas contenidas en contenedores
    - sistemas externos
    - sustantivos abstractos
    - organizaciones
    - eventos
    - reglas y políticas
    - catálogos
    - registro de asuntos financieros o legales
    - servicios e instrumentos financieros

Conceptos

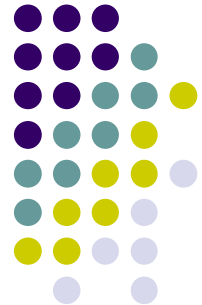
Identificación de Conceptos (4)



- Identificación de sustantivos
  - Se identifican los sustantivos de una descripción textual del problema (visión del problema y/o casos de uso) y se los considera como conceptos o atributos candidatos
  - No es posible realizar esta actividad en forma totalmente automática
    - El lenguaje natural es ambiguo
    - No todo sustantivo refiere a un concepto significativo

Conceptos

Identificación de Conceptos (5)



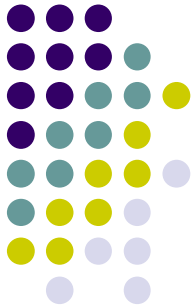
- Identificación de sustantivos (cont.)
  - Ejemplo:

    *… Un **cliente** llega a un **puesto de venta** para reservar un **pasaje** de **avión**… El **empleado** hace la **reserva** en el sistema de **aerolínea**.*

    Consideramos estos sustantivos como los primeros candidatos para ser conceptos

Conceptos

Sugerencias

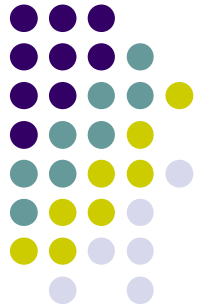


- Cómo crear un Modelo de Dominio
  1. Listar los conceptos candidatos usando cualquiera de las dos técnicas presentadas (o una combinación de ambas)
  2. Incluirlos en el Modelo de Dominio
  3. Agregar las asociaciones necesarias para registrar relaciones que necesiten ser preservadas
  4. Agregar los atributos necesarios para satisfacer los requerimientos de información

Sugerencia: Generar y mantener el diagrama en paralelo

Conceptos

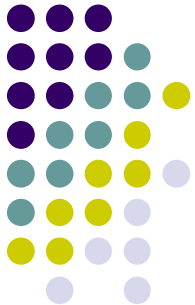
Sugerencias (2)



- Nombres y modelado
  - La estrategia del cartógrafo se aplica tanto a la construcción de mapas y a la de Modelos de Dominio
    - Usar nombres que existan en el territorio
    - Excluir características irrelevantes
    - No incluir cosas inexistentes

Conceptos

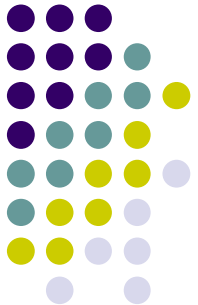
Sugerencias (3)



- Granularidad de la especificación
  - Es mejor sobreespecificar con muchos conceptos de granularidad fina, que subespecificar
  - Siempre es posible agregar o eliminar conceptos
  - El costo de eliminar un concepto que resultó innecesario es menor que el de agregar uno que fue omitido

Conceptos

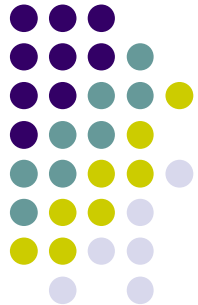
Sugerencias (4)



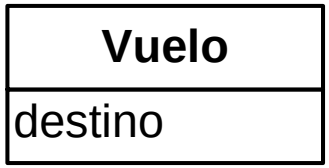
- Error común al identificar conceptos
  - El error más común al crear un Modelo de Dominio es representar algo como un atributo cuando debió ser un concepto
  - Si no se piensa en un concepto X básicamente como un número o un texto, entonces X probablemente sea un concepto
  - En caso de duda, representarlo como un concepto

Conceptos

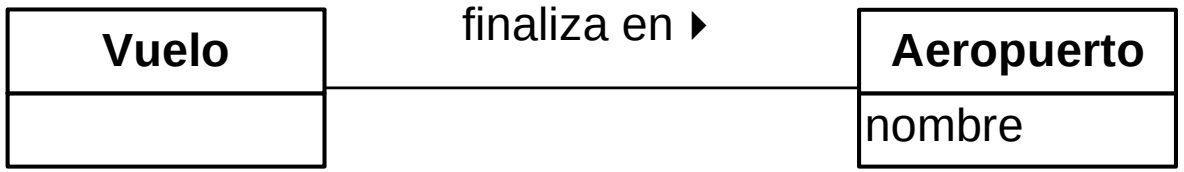
Sugerencias (5)



- Error común al identificar conceptos (cont.)

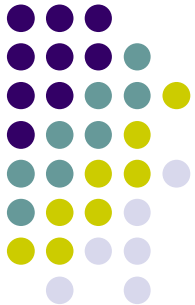


vs.



Conceptos

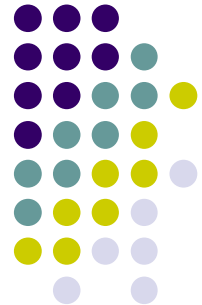
Sugerencias (6)



- Supóngase la siguiente situación:
  - Una instancia de “Producto” representa a un producto físico en una tienda
  - Un producto tiene un número de serie, una descripción, un precio y un código, que no aparecen en ninguna otra parte
  - Los que trabajan en la tienda no tienen “memoria”
  - Cada vez que un producto físico es vendido, la correspondiente instancia de “Producto” es eliminada del sistema

Conceptos

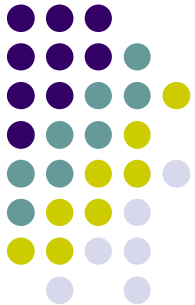
Sugerencias (7)



- En el caso de que un producto se agote nadie podrá saber cuál era el precio de ese producto
- Ese dato estaba incluido solamente en las instancias que conformaban el inventario
- Notar también que existe información repetida

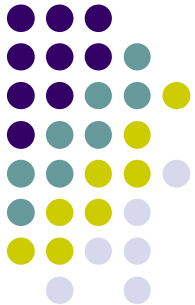
Conceptos

Sugerencias (8)

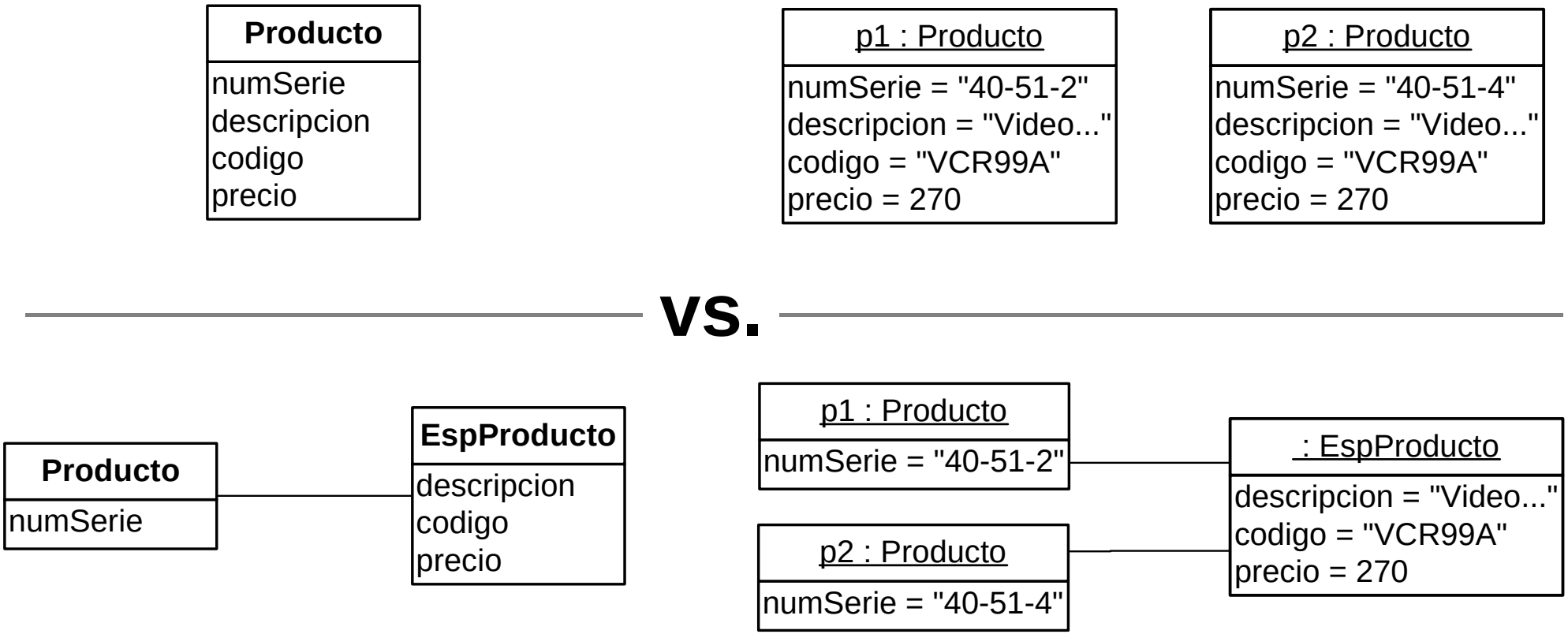


- Especificaciones y descripciones
  - Se necesitan conceptos que sean descripciones de otros conceptos
  - En el caso del producto necesitamos una “DescripcionProducto” que registre la información de los productos
  - Estos conceptos no representan los productos, sino información acerca de ellos
  - Si todas las instancias de “Producto” son eliminadas, la “DescripcionProducto” permanece

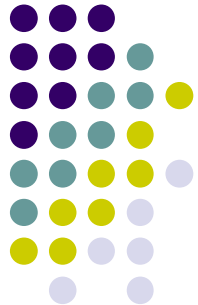
Conceptos
Sugerencias (9)



- Especificaciones y descripciones (cont.)
  - Ejemplo

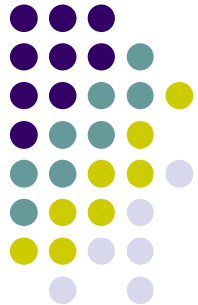


Asociaciones



- Es necesario identificar aquellas asociaciones entre conceptos que:
  - Sean necesarias para satisfacer los requerimientos de información
  - Ayuden a la comprensión del Modelo de Dominio
- Una asociación es una relación entre conceptos que indica alguna conexión interesante o significativa entre ellos
- En general surgen del conocimiento de una relación que debe ser preservada por una cierta duración (desde segundos, hasta años)

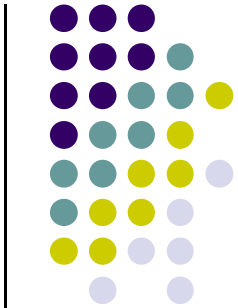
Asociaciones (2)



- Se distinguen dos categorías de asociaciones:
  - **De comprensión:** Permiten comprender mejor el problema
  - **Need-to-know:** Permiten satisfacer las requerimientos de información

Asociaciones

Detección de Asociaciones

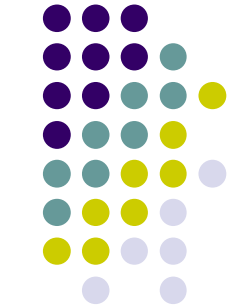


- Lista de categorías de asociaciones comunes que pueden resultar de utilidad al momento de realizar un Modelo de Dominio

Categoría	Ejemplo
A es una parte física de B	Ala - Avión
A es una parte lógica de B	Tramo - Ruta
A está contenido físicamente en B	Pasajero - Avion
A está contenido lógicamente en B	Vuelo - PlanVuelo
A es un miembro de B	Piloto - Aerolinea

Asociaciones

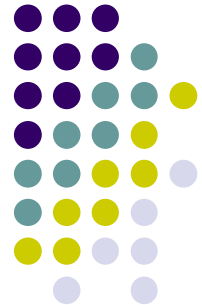
Detección de Asociaciones (2)



- Lista de categorías de asociaciones (cont.)
  - La lista se puede continuar con
    - A es una descripción de B
    - A es un ítem de una transacción B
    - A es conocido/registrado/capturado en B
    - A es una subunidad organizacional de B
    - A usa o maneja B
    - A se comunica con B
    - A esta relacionado con la transacción B
    - A es una transacción relacionada con la transacción B
    - A está cerca de B
    - A es propiedad de B

Asociaciones

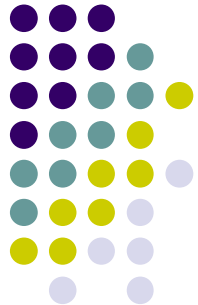
Asociaciones a Considerar



- Considerar la inclusión de las siguientes asociaciones
  - Asociaciones para las que el conocimiento de la relación debe ser preservado por una cierta duración (need-to-know)
  - Asociaciones derivadas de la “Lista de Asociaciones”
  - De ser necesario incluir asociaciones de comprensión

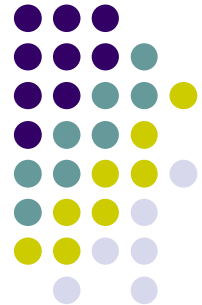
Asociaciones

Asociaciones a Considerar (2)

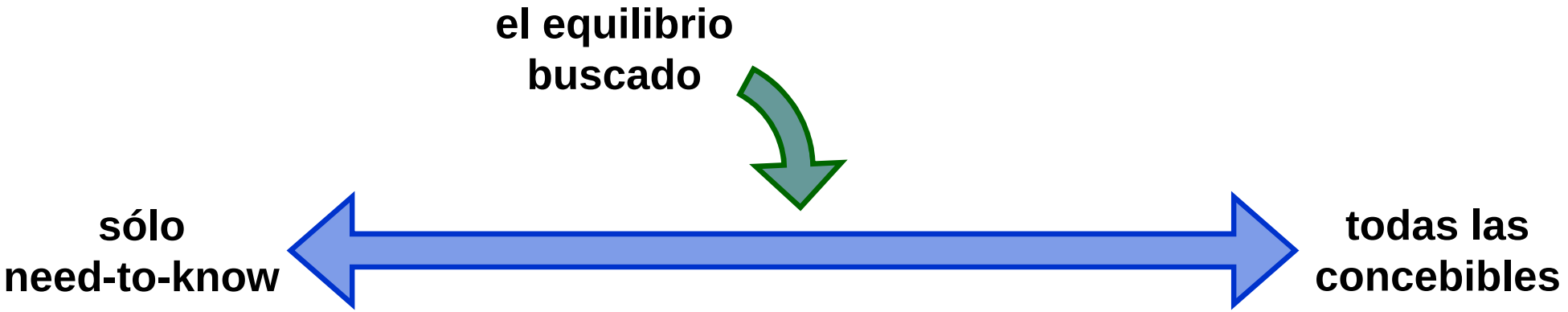


- Sugerencias
  - Concentrarse en identificar conceptos más que asociaciones
  - Evitar mostrar asociaciones derivables o redundantes
  - Demasiadas asociaciones tienden a confundir más que a ilustrar

Asociaciones a Considerar (3)

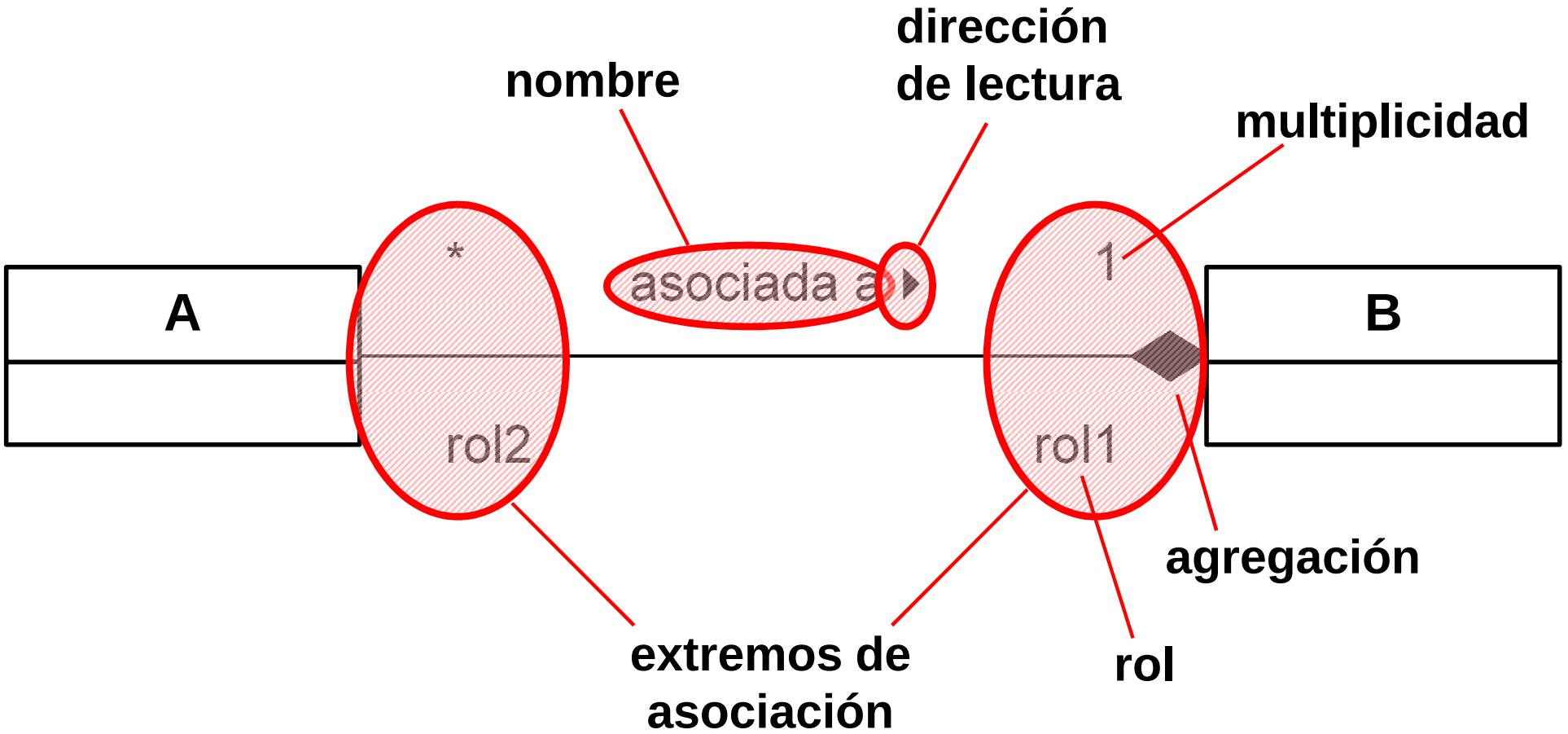


- **Ideal:** punto intermedio entre un modelo minimal sólo asociaciones need-to-know y otro con todas las asociaciones concebibles
- **Criterio:** que satisfaga todos los requerimientos de información y además permita una comprensión de los conceptos en el problema



Asociaciones
Notación

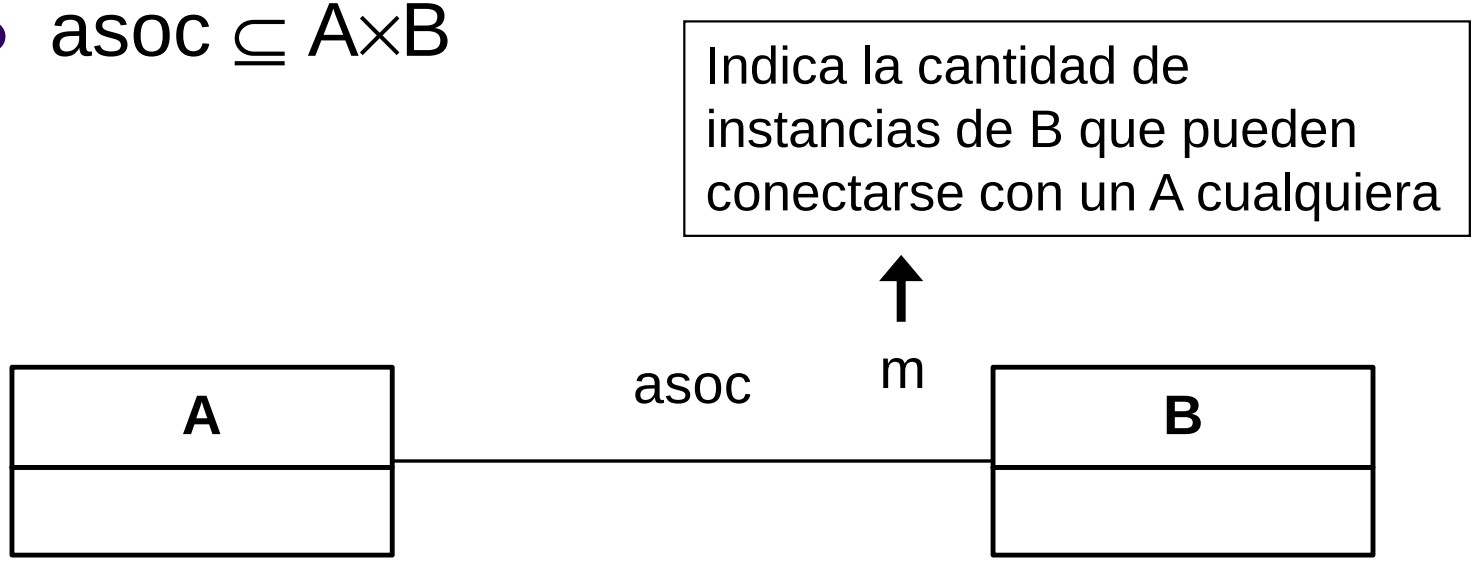
- La asociación se lee: “A asociada a B”



Asociaciones

Notación - Multiplicidades

- La multiplicidad limita la cantidad de veces que un objeto determinado está conectado a otros a través de una asociación
- Eso se indica en el extremo de asociación opuesto a la clase del objeto
- asoc $\subseteq$ A$\times$B

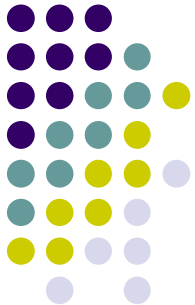


asoc

a1	b1
a1	b2
a2	b2
a3	b1
a1	b3
a2	b3

Asociaciones

Notación – Multiplicidades (2)

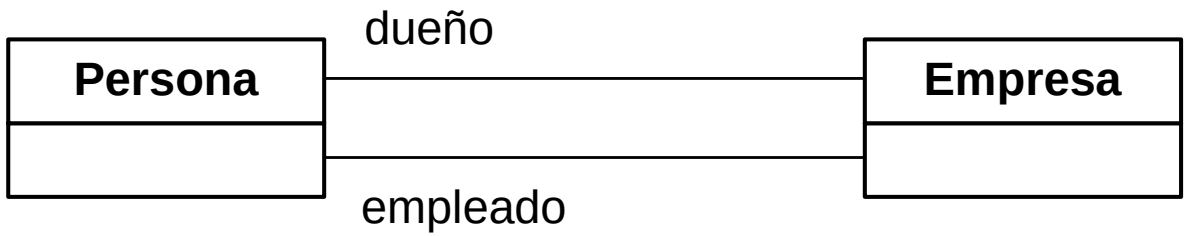


- Se expresa como un subconjunto de los naturales (subrango o enumerado)
  $m \subseteq \mathbb{N}$ tal que max(m)>0
- Ejemplos
  - * Cualquier cantidad (cero o más)
  - 1..* Al menos uno (uno o más)
  - 0..1 Opcionalmente uno (cero o uno)
  - 5 Exactamente cinco
  - 3,5,8 Exactamente tres, cinco u ocho

Asociaciones

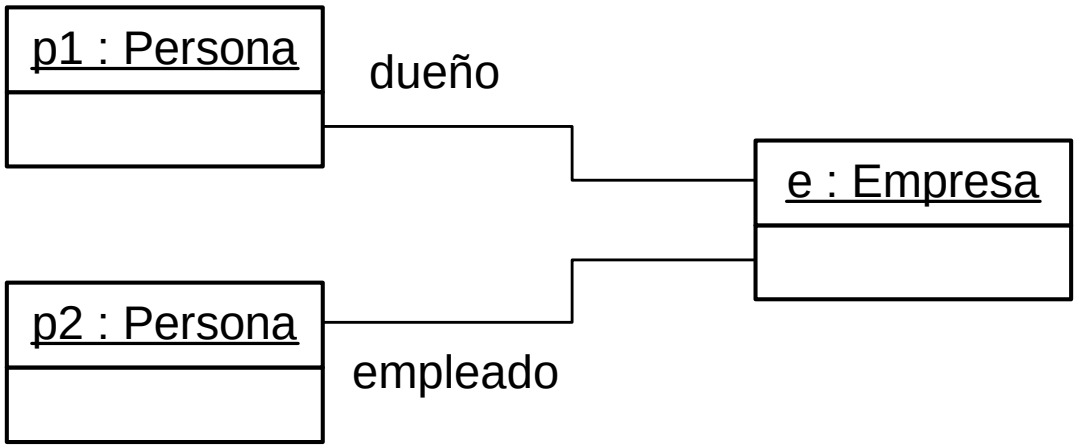
Notación - Roles

- Especifican el papel que juegan las clases en una asociación
- Pueden ser necesarios para eliminar ambigüedades



¿p1 es dueño o empleado de e?

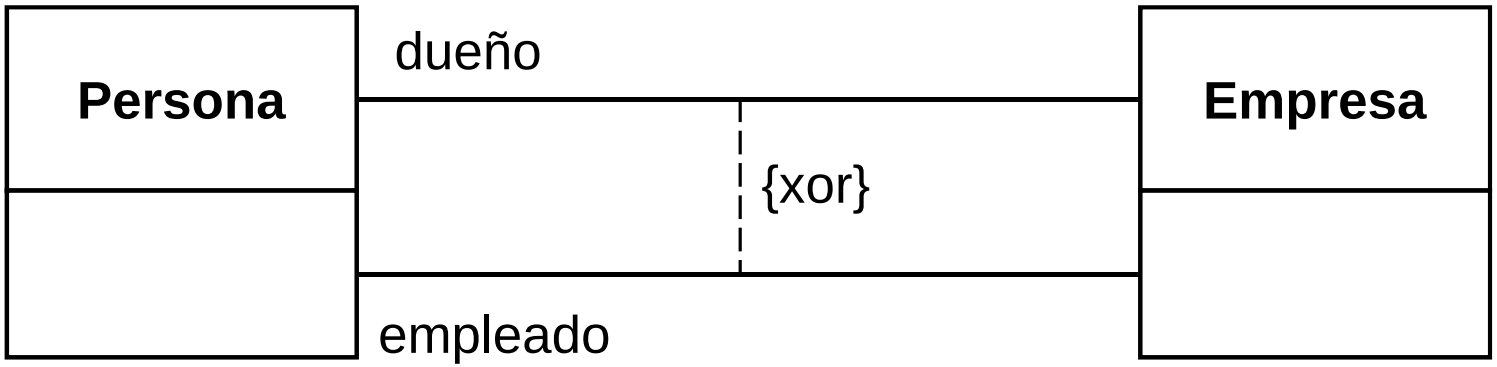
¿y p2?



Asociaciones

Notación - Restricciones

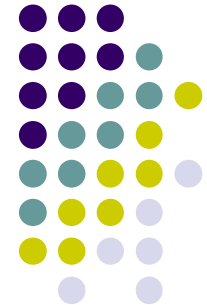
- En ocasiones es necesario especificar que si un objeto está conectado con otro no puede estar conectado a un tercero



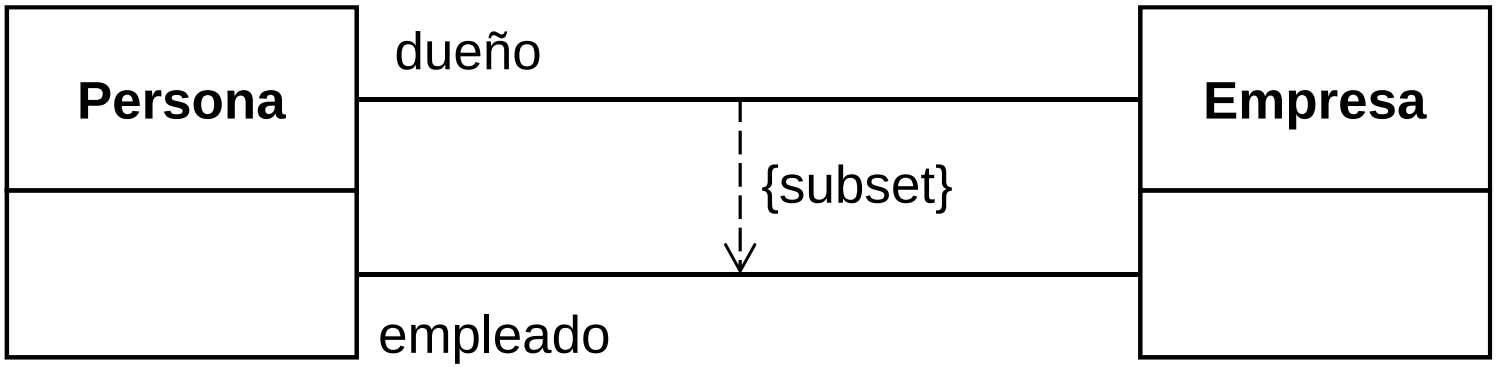
- De esta forma una persona no puede ser dueño y empleado de la misma empresa

Asociaciones

Notación – Restricciones (2)



- En otras puede ser necesario especificar que si un objeto está conectado con otro tiene que estar conectado a un tercero

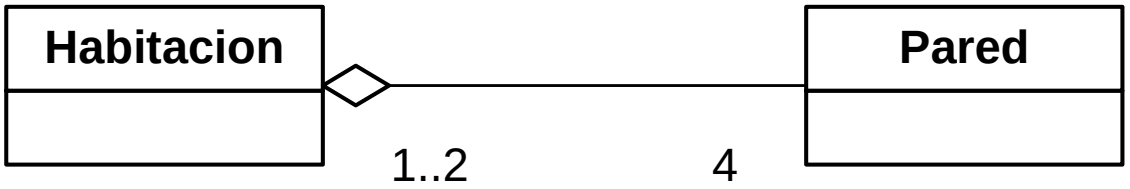


- De esta forma una persona que sea dueña de la empresa tiene que ser empleado

Asociaciones

Notación - Agregación

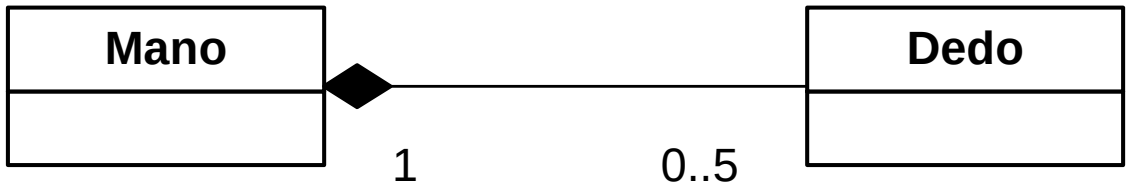
- Es una forma más fuerte de asociación
- Significa que un elemento es parte de otro
- Existen dos variantes
  - Agregación compartida (agregación)
  - Agregación compuesta (composición)
- Agregación compartida



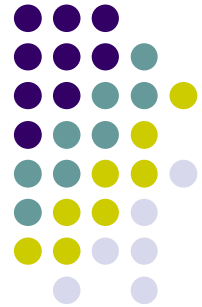
Asociaciones

Notación - Agregación

- Agregación compuesta
  - Un elemento es exclusivo del compuesto (máximo de la multiplicidad es 1)
  - Generalmente una acción sobre el compuesto se propaga a las partes (típicamente en la destrucción)



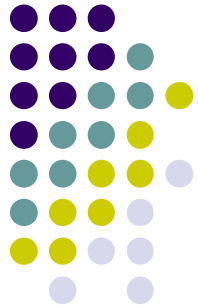
Atributos



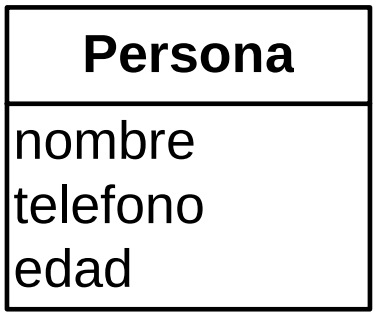
- Es necesario identificar aquellos atributos que permitan satisfacer los requerimientos de información
- Un atributo de un objeto se entiende como un *data value*
- El tipo de un atributo es un *data type*

Atributos

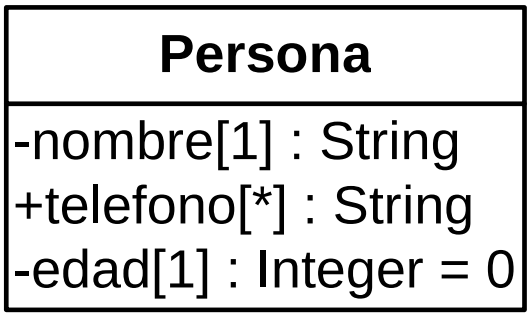
Notación



- Al mostrar un atributo es necesario especificar al menos su nombre
- Propiedades opcionales
  - Tipo, multiplicidad, valor inicial, visibilidad, etc.



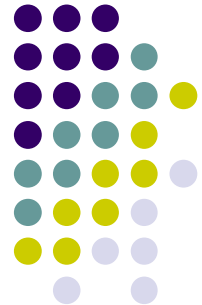
Representación mínima



Representación completa

Atributos

Notación (2)



- Alcance de atributos

Empleado
sueldo

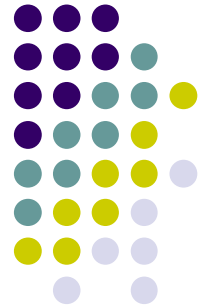
De instancia

Empleado
sueldo

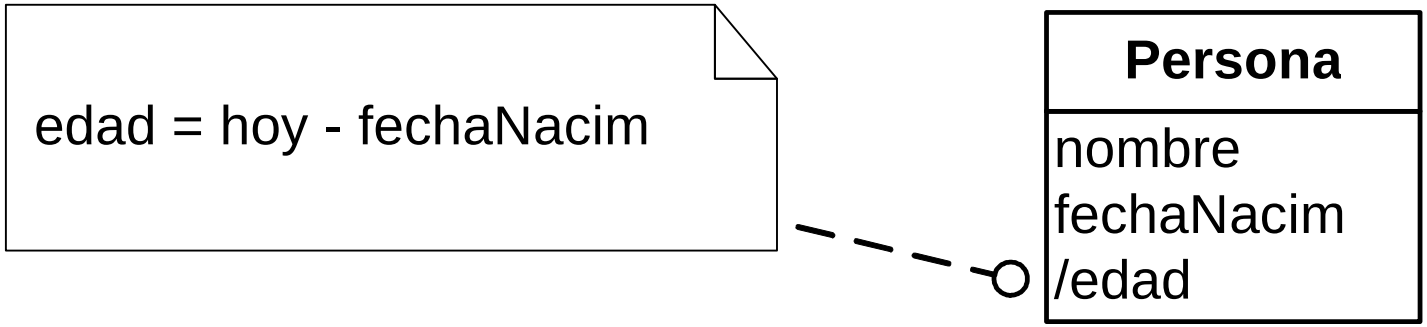
De clase

Atributos

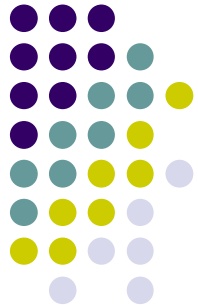
Notación (3)



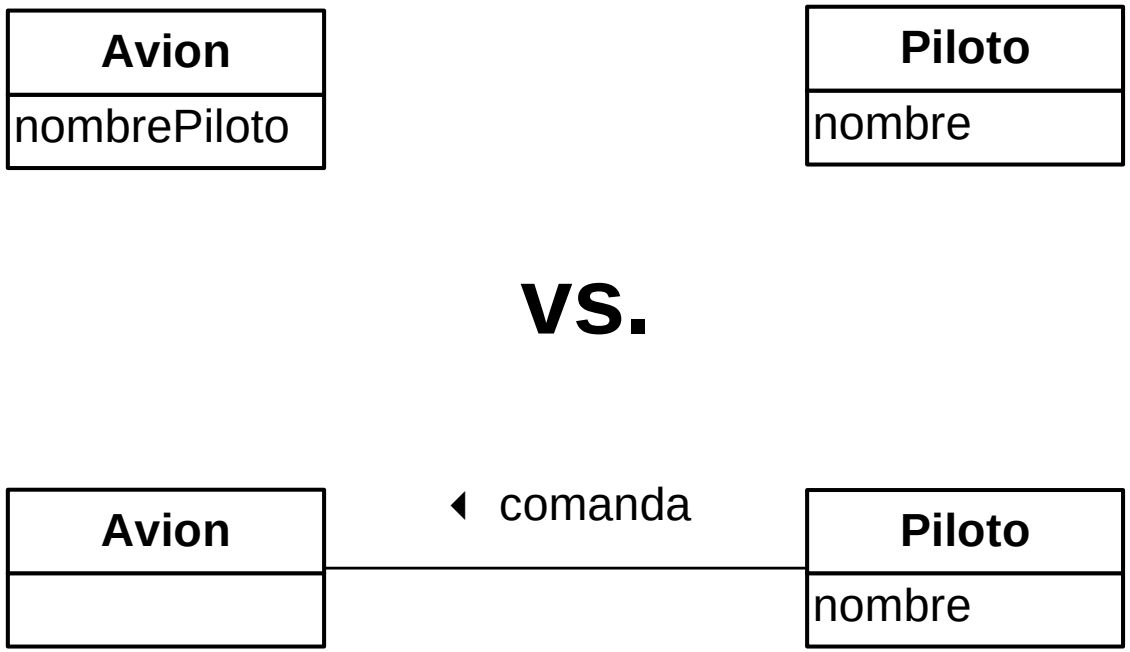
- Un atributo (o cualquier elemento) que sea derivable se marca con un ‘/’
- Lo usual es adjuntarle una nota especificando la forma en que se calcula



Atributos
Sugerencias

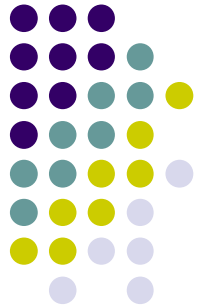


- No utilizar atributos como clave foránea
  - Los atributos **no** deben ser utilizados para relacionar elementos del modelo

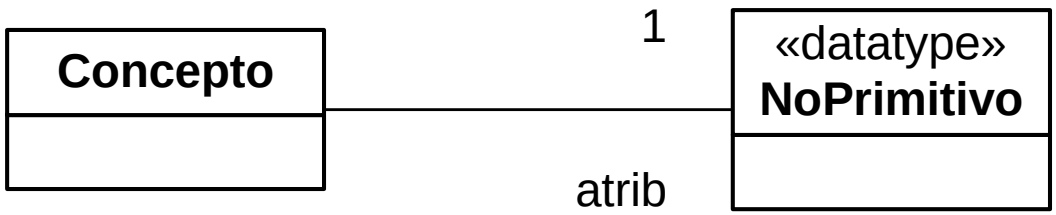


Atributos

Sugerencias (2)

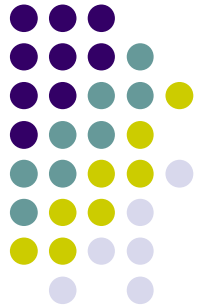


- Tipos primitivos y no-primitivos
  - Los tipos de los atributos son en general tipos primitivos (Integer, String, Real, etc.)
  - De ser necesario es posible definir tipos no-primitivos para un problema

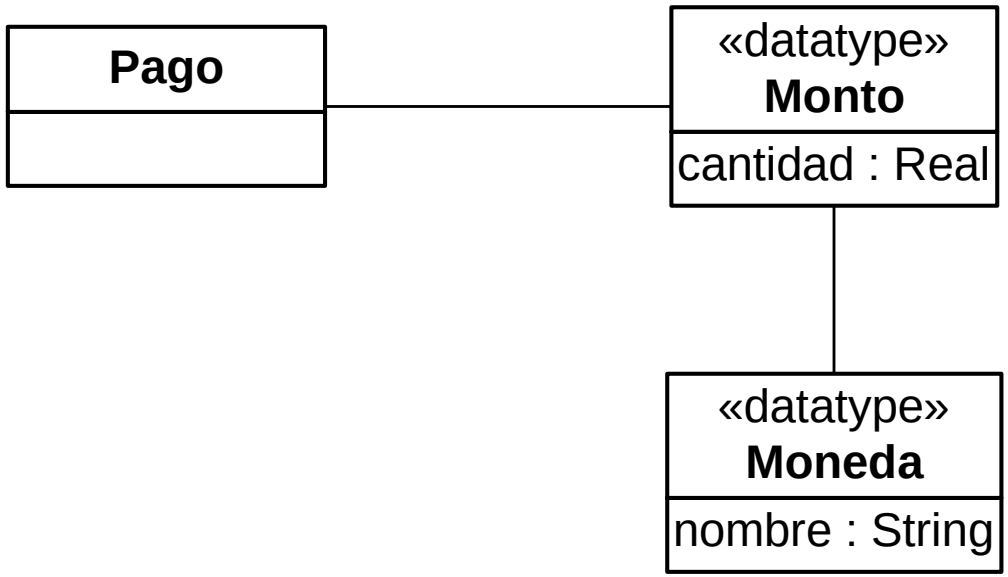


Atributos

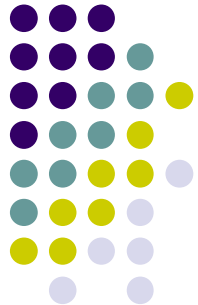
Sugerencias (3)



- Técnica de modelado
  - El monto de un pago puede ser un número
  - En ciertos casos esto no es flexible si se necesita conocer además otra información (como la moneda)

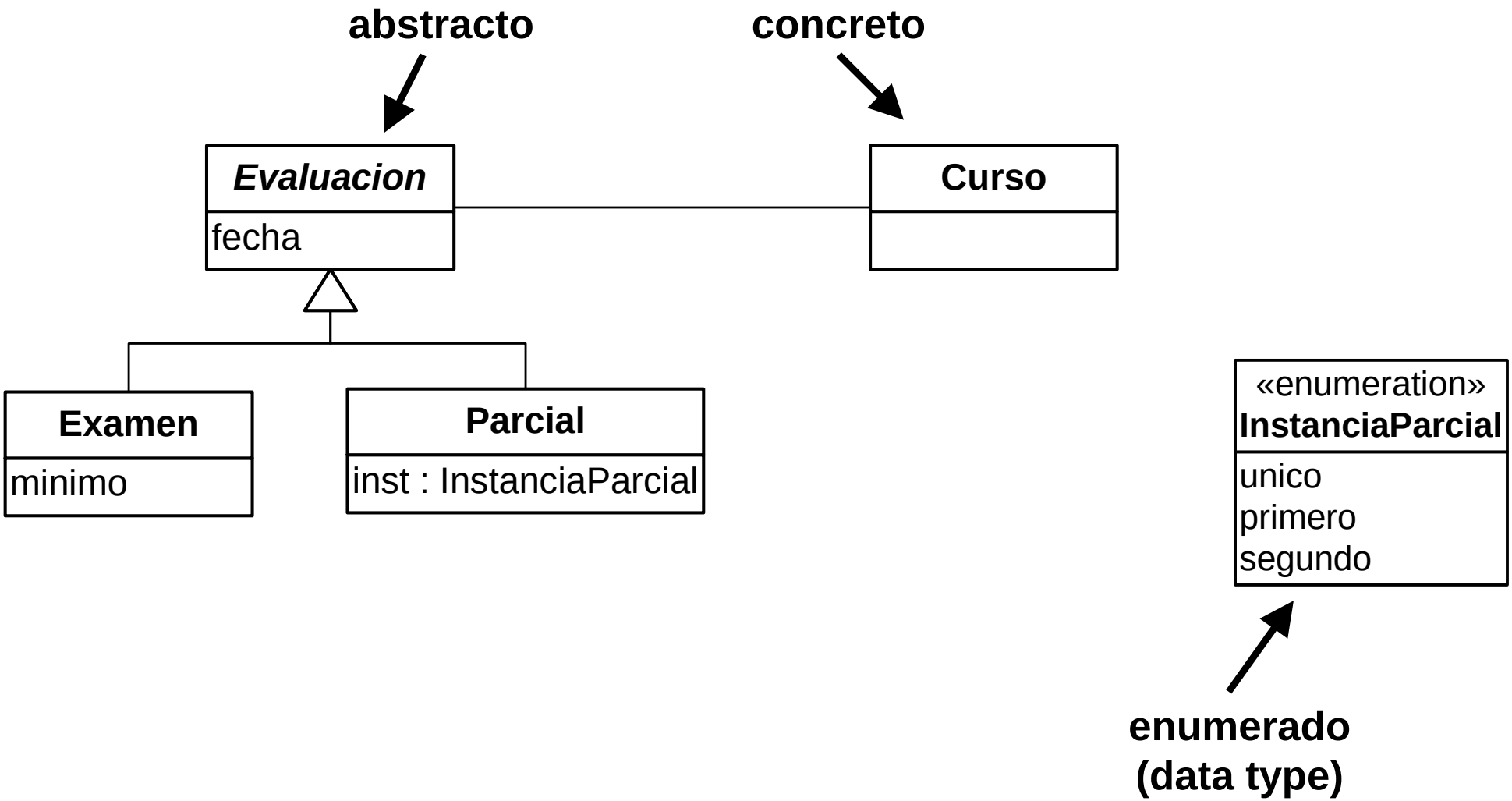


Generalizaciones



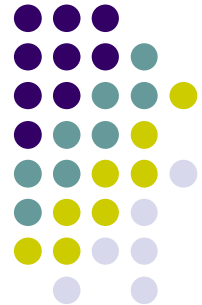
- Es posible especificar variantes de un concepto cuando
  - Los subtipos potenciales representan variantes interesantes de un cierto concepto
  - Un subtipo es consistente con su supertipo (se aplica subsumption)
  - Todos los subtipos tienen atributos comunes que pueden ser factorizados en el supertipo
  - Todos los subtipos tienen asociaciones comunes que pueden ser factorizadas en el supertipo

Generalizaciones
Notación



Generalizaciones

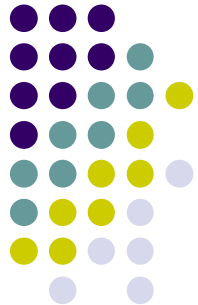
Sugerencias



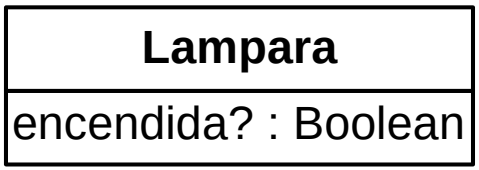
- Modelado de estados
  - Modelar los estados de un concepto solamente cuando resulte imprescindible para comprender el problema
  - No modelar los estados de un concepto X como subtipos de X
  - De ser necesario, utilizar
    - Atributos
    - Delegación (más frecuentemente en diseño)

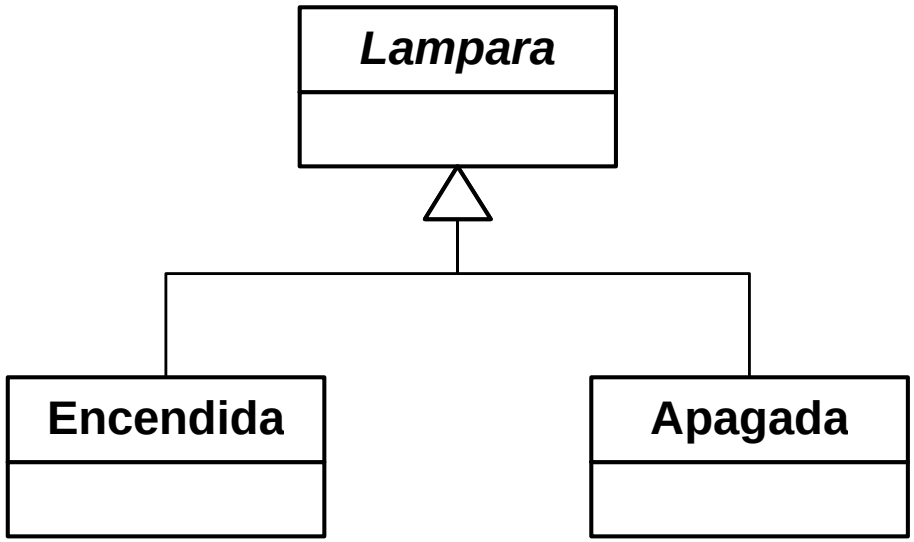
Generalizaciones

Sugerencias (2)



- Modelado de estados (cont.)

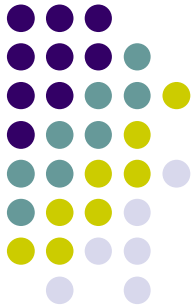




Correcto

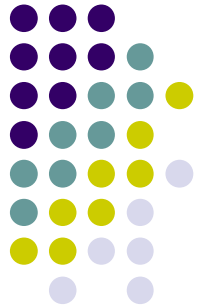
Incorrecto

Tipos Asociativos



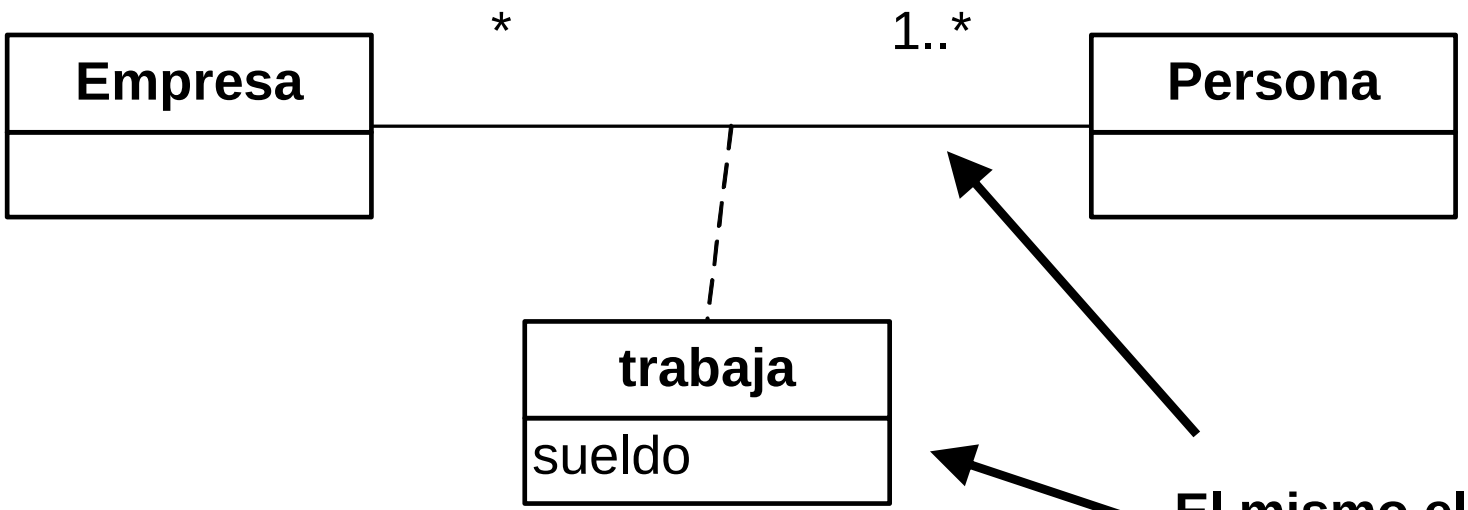
- Un tipo asociativo es un elemento que es tanto *clase* como *asociación*
- Motivación para usar tipos asociativos
  - Una empresa contrata a diferentes personas para trabajar y a cada una le asigna un sueldo particular
  - Una persona puede ser contratada por diferentes empresas y puede recibir un sueldo diferente por cada trabajo
  - Interesa saber cuánto cobra una persona en cada trabajo

Tipos Asociativos (2)



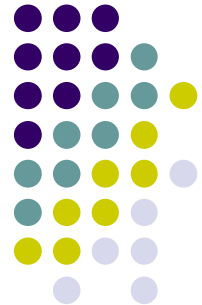
- Incluir el *sueldo* en la *Persona* no es correcto ya que una *Persona* puede tener más de un *sueldo* y éste depende del trabajo
- Incluir el *sueldo* en la *Empresa* tampoco es correcto ya que la *Empresa* paga sueldos distintos a cada empleado
- Esto conduce a la noción de tipos asociativos, los cuales permiten agregar propiedades a las asociaciones

Tipos Asociativos
Notación



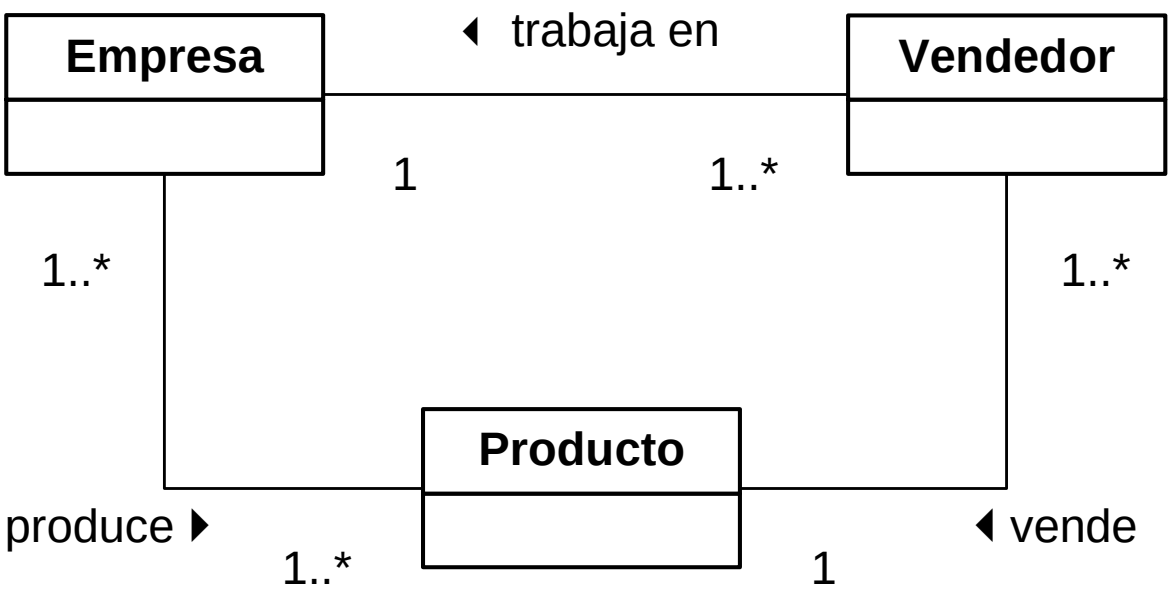
El mismo elemento representado de dos formas diferentes:
Para multiplicidades, etc., uno
Para atributos el otro

Restricciones



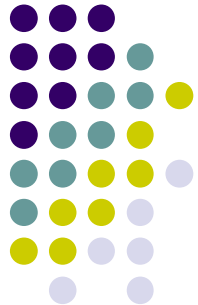
- Es muy común el hecho de que un Modelo de Dominio no alcance a representar exactamente la realidad planteada
- Existen casos donde un modelo representa fielmente la mayoría de los aspectos de la realidad sin embargo permite otros que no son deseables

Restricciones
Motivación

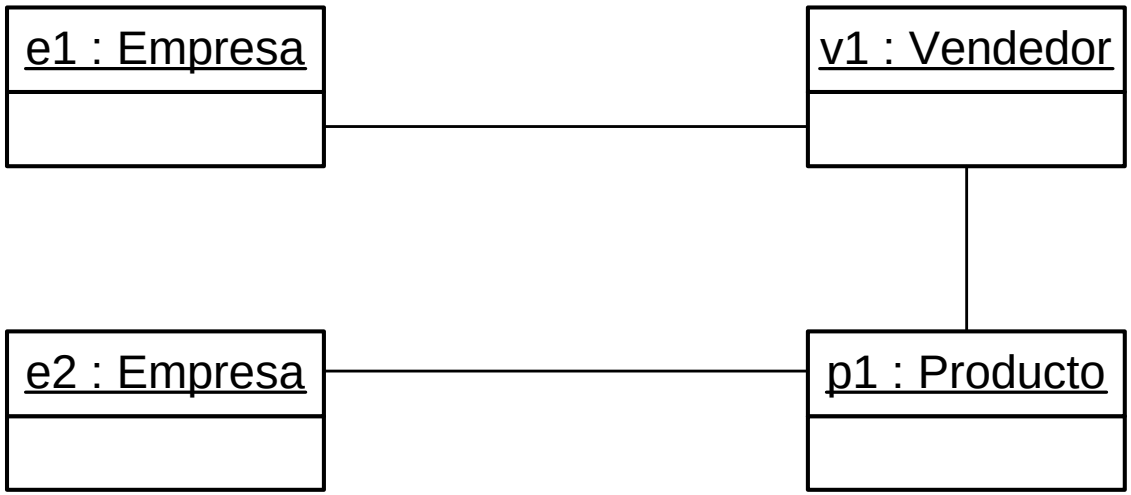


El modelo representado por este diagrama: ¿Refleja fielmente la realidad?

Restricciones
Motivación (2)



Permite o considera como válidos casos como:
"Un *vendedor* vende un *producto* producido por una *empresa* para la cual él no trabaja"

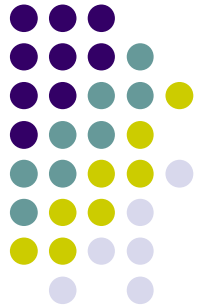


¡Todas las multiplicidades están satisfechas!
(esta configuración de objetos es válida respecto al Modelo de Dominio)

La empresa de v1 (o sea e1) debería producir el producto que él vende (o sea p1), o v1 debería trabajar en la empresa e2

Restricciones

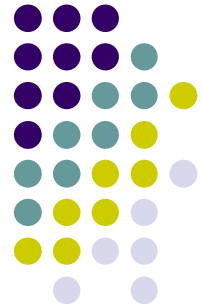
Motivación (3)



- Es muy común enfrentarse ante este tipo de situaciones
- Existen dos alternativas para solucionar el problema
  - Modificar el Modelo de Dominio para evitar que configuraciones no deseadas puedan ser válidas
  - Adjuntar restricciones al modelo tales que invaliden aquellas configuraciones no deseadas

Restricciones

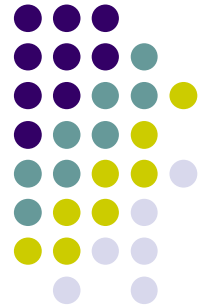
Modificar del Modelo



- En muchos casos es posible eliminar situaciones como la descrita mediante una modificación al modelo
- Es común que esta modificación no sea menor por lo que es posible que
  - Insuma demasiado tiempo
  - La versión modificada sea muy complicada
  - La versión modificada restrinja los casos no deseados pero introduzca otros nuevos

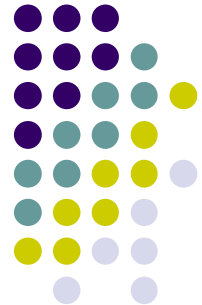
Restricciones

Adjuntar Restricciones



- Otra alternativa al problema es la imposición de restricciones (en particular invariantes)
- Un invariante es un predicado que expresa una condición sobre los elementos del Modelo de Dominio y que siempre debe ser verdadero
  - Cuando es evaluado contra una cierta configuración de objetos dando un resultado de *falso* significa que la configuración de objetos no es válida
- UML no especifica el modo en que un invariante deba ser expresado
  - Puede utilizarse notación informal o formal

Restricciones
Invariantes - Informal



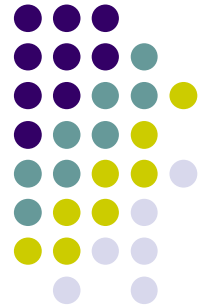
- Los invariantes pueden ser expresados informalmente en lenguaje natural
- Un ejemplo de esto puede ser

Invariante:

"Todo *vendedor* debe vender un *producto* que sea producido por la *empresa* para la cual trabaja"

Restricciones

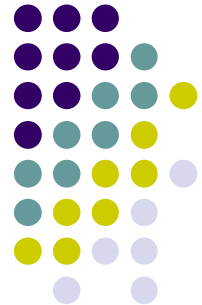
Invariantes - Informal (2)



- Ventajas
  - Es “entendido” por todos
- Desventajas
  - Es ambiguo: una restricción compleja puede
    - Ser difícil de escribir y/o leer
    - Fácilmente dar lugar a confusiones
  - No puede ser procesado en forma automática

Restricciones

Invariantes - Formal

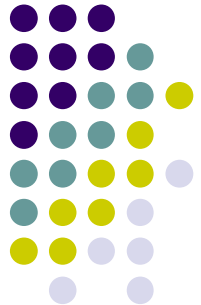


- UML contiene un lenguaje que fue diseñado específicamente para la especificación de este tipo de restricciones
- Es relativamente simple e intuitivo
- Este lenguaje es el Object Constraint Language
- Ejemplo

```
context Vendedor inv:
  self.producto.empresa->includes(self.empresa)
```

Restricciones

Invariantes - Formal (2)



- Ventajas
  - Una restricción tiene un significado único y preciso
  - Puede ser procesada en forma automática
- Desventajas
  - El lenguaje a utilizar puede resultar extremadamente complejo
  - Requiere el aprendizaje de las construcciones del lenguaje

Representación de Diagramas