



T6 - Introducción al Modelado Conceptual

Universidad de Sevilla

**Departamento de Lenguajes y Sistemas
Informáticos**

UNIVERSIDAD DE SEVILLA

Objetivos



Contenido

Introducción

Notación UML

Diagramas de clases

Diagramas de objetos

Creación de modelos conceptuales



Contenido

Introducción

Notación UML

Diagramas de clases

Diagramas de objetos

Creación de modelos conceptuales



¿Qué es el modelado conceptual?

El modelado conceptual es una técnica de **análisis** de requisitos y de **diseño** de bases de datos.

Como técnica de análisis de requisitos...

- Ayuda a **identificar problemas** en los requisitos antes de comenzar el desarrollo, evitando gastos innecesarios.

Como técnica de diseño de bases de datos...

- Permite representar de forma abstracta los conceptos y hechos relevantes del dominio del problema y transformarlos posteriormente en un **esquema de una base de datos** concreta.

¿Qué es el dominio del problema?

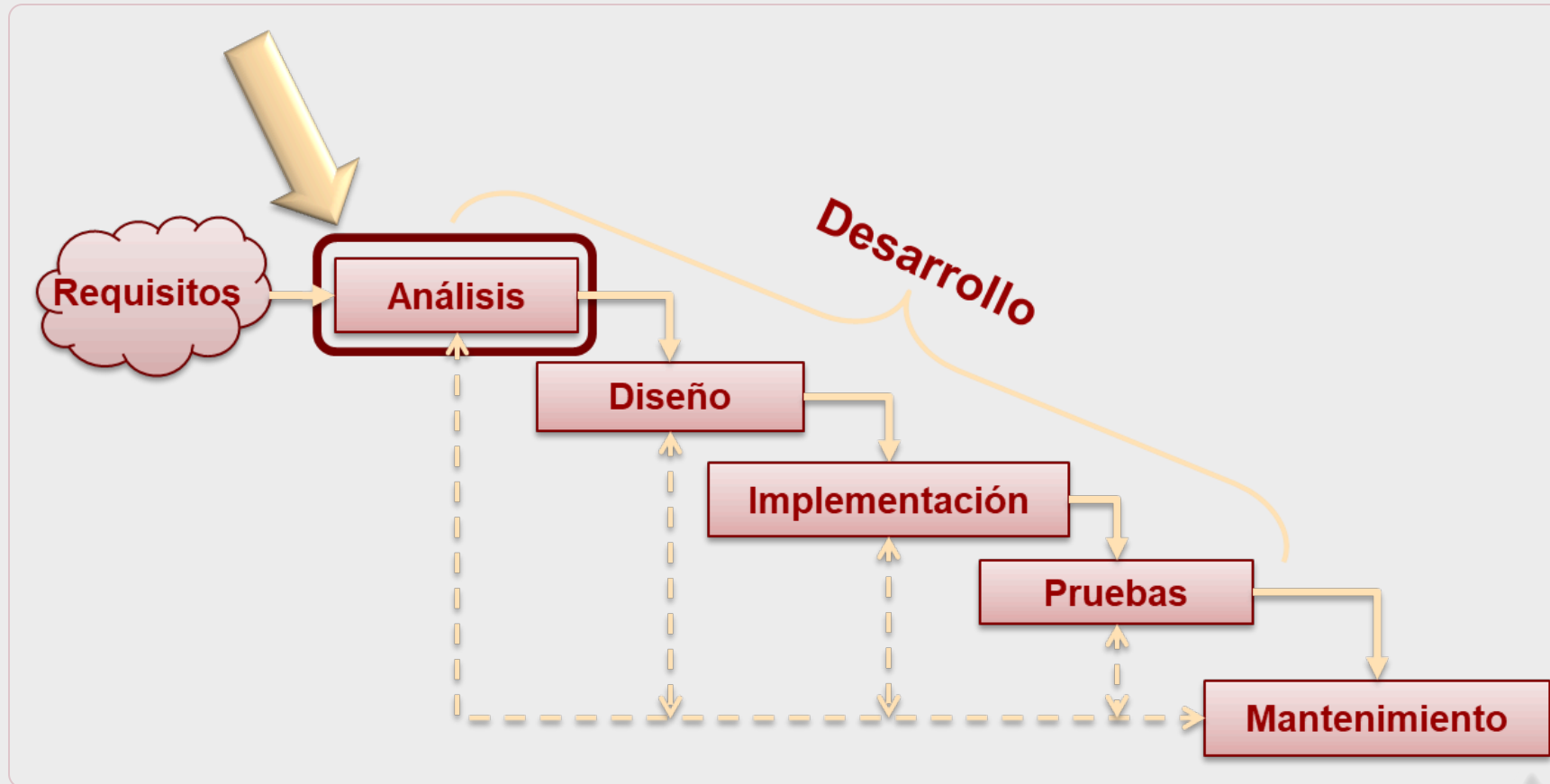
Área de **experiencia** o aplicación que necesita conocerse para resolver un problema.

En el ámbito de los sistemas de información, el dominio del problema es el conjunto de conceptos interrelacionados que es necesario conocer para entender el **negocio del cliente**, y por lo tanto, para poder entender sus necesidades y proponer una solución adecuada.



¿Cuándo se usa el modelado conceptual?

Independientemente del ciclo de vida, se utiliza durante el análisis.

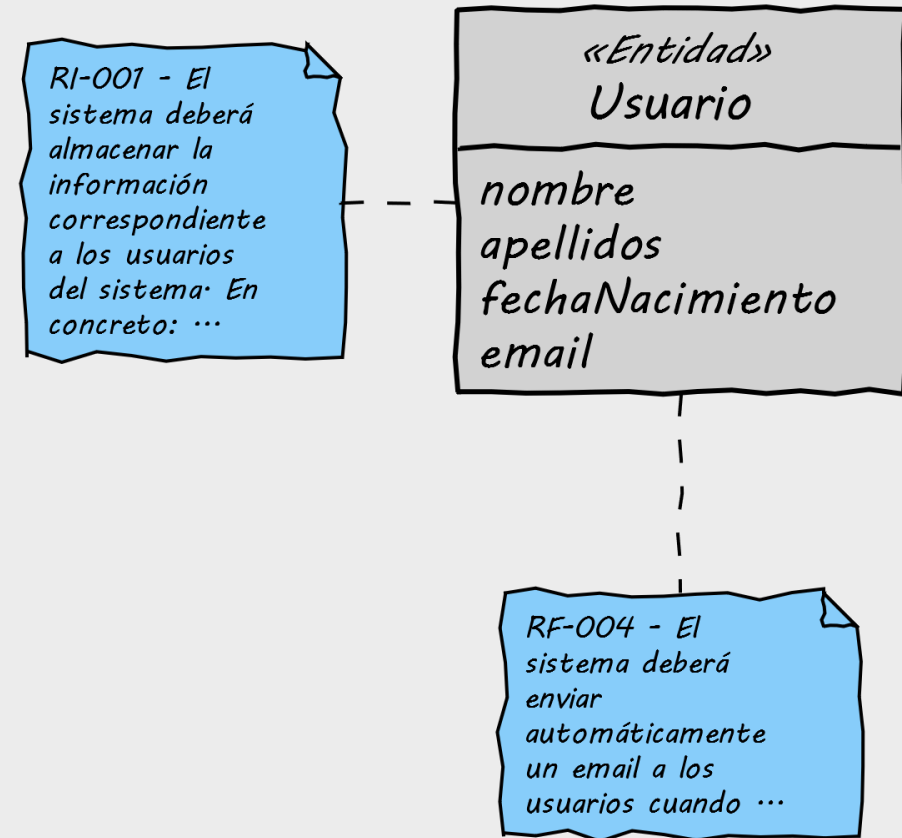


Trazabilidad hacia requisitos

Cada elemento del modelo conceptual debe justificarse mediante trazabilidad explícita con:

- Requisitos de información
- Reglas de negocio

Ejemplo trazabilidad



Estándar para modelado conceptual

UML (**U**nified **M**odeling **L**anguage).

Resultado de la fusión de varias propuestas previas.

Gestionado por la OMG (**O**bject **M**anagement **G**roup).

Ampliamente usando en la industria del software.

Múltiples herramientas disponibles.

Define 14 tipos de diagramas para modelar sistemas software (versión 2.4.1, agosto 2011).

Para modelado conceptual, se utilizan principalmente:

- Diagramas de clases
- Diagramas de objetos



Conceptos básicos del modelado conceptual

Clase entidad

- Atributo

Asociación

- Rol
- Multiplicidad

Objeto (instancia de una clase)

Enlace (instancia de una asociación)

Generalización/especialización

Composición y agregación



Contenido

Introducción

Notación UML

Diagramas de clases

Diagramas de objetos

Creación de modelos conceptuales



Clase entidad

Representa un **concepto relevante** del dominio del problema sobre el que el sistema debe almacenar información porque así se ha especificado (o se deduce) en uno o más requisitos.

Se nombran mediante un **sustantivo en singular**.

Atributo de una clase entidad

Son **propiedades** asociadas a un concepto relevante del dominio del problema que el sistema debe almacenar porque así se ha especificado (o se deduce) en uno o más requisitos.

Se nombran mediante un **sustantivo en singular**.

Los valores de los atributos deben ser **atómicos**.

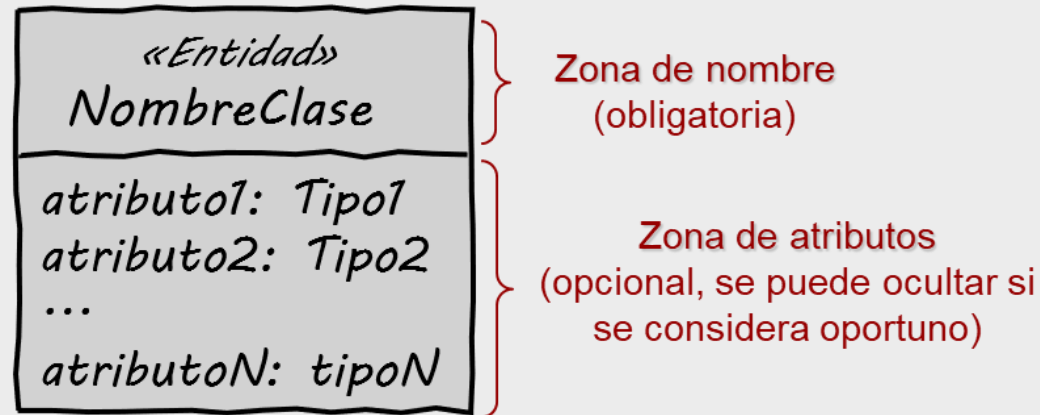


Notación para las clases entidades UML

Con el estereotipo «Entidad» se indica que la clase representa una entidad y no una clase de un lenguaje de programación orientado a objetos.

En modelado conceptual se asumirá el estereotipo «Entidad» por defecto.

Notación para clases UML



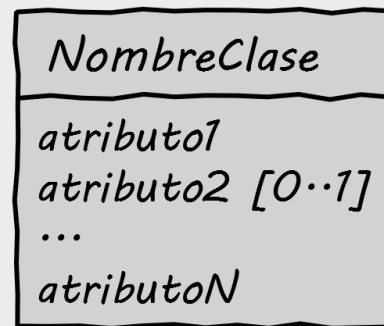
Notación para las clases entidades UML

En modelado conceptual no se suele especificar el tipo de los atributos (salvo los enumerados).

Mediante [0..1] se indica que el atributo es opcional, es decir, que habrá momentos en los que no se conocerá su valor y se representará mediante un valor nulo.

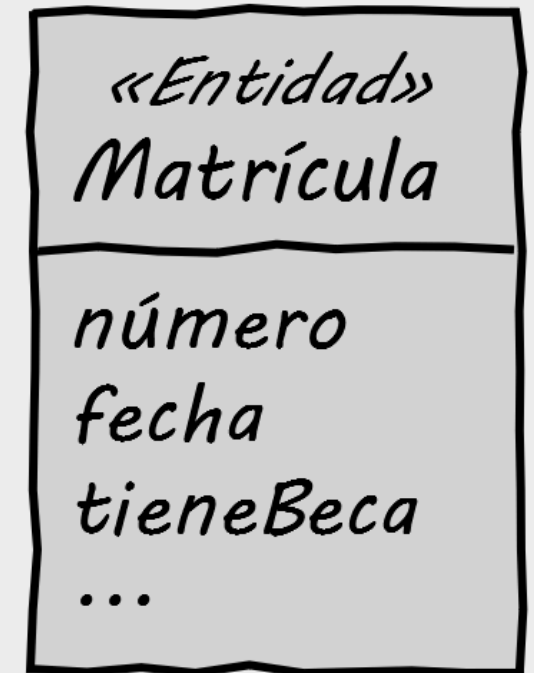
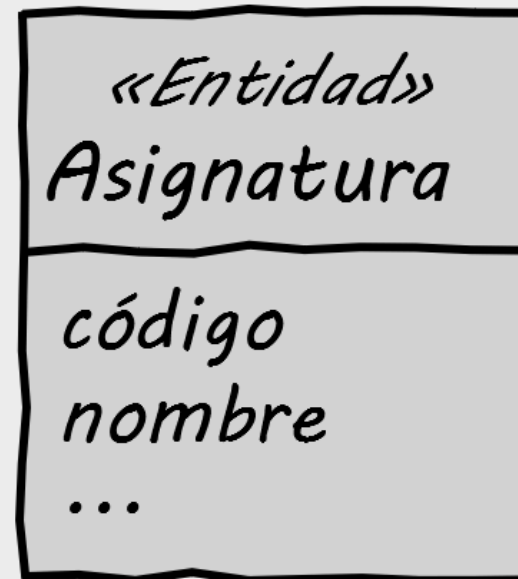
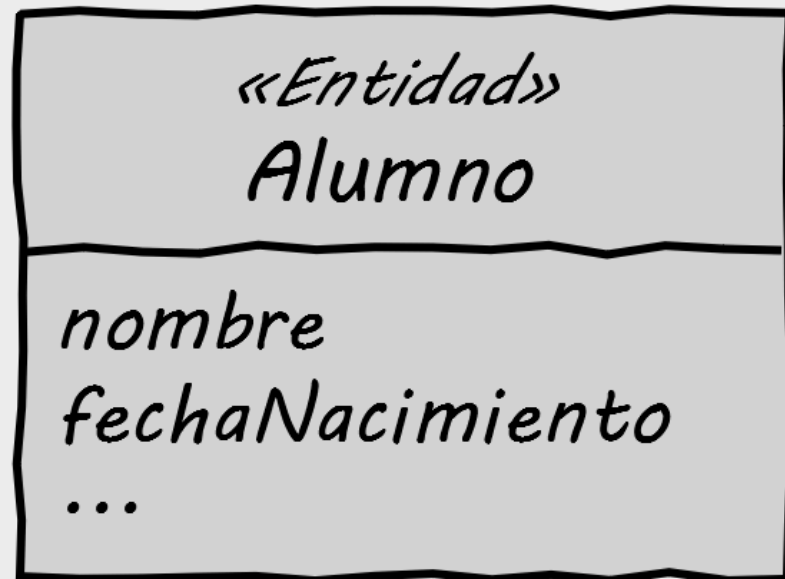
El valor de atributo2 puede ser nulo

Notación para clases UML II



Ejemplo de clases UML

Ejemplo de clases



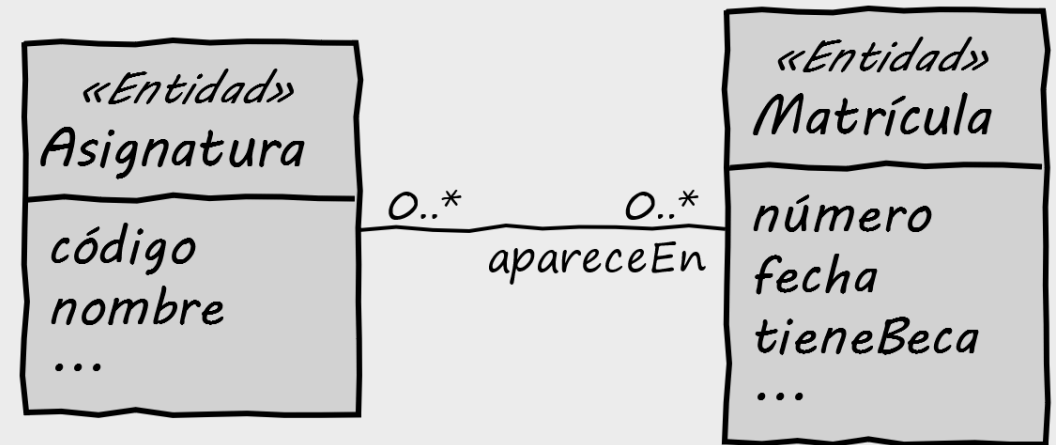
Asociación entre clases entidades

Representa algún tipo de relación entre dos o más conceptos relevantes del dominio del problema que el sistema debe almacenar porque así se ha especificado (o se deduce) en uno o más requisitos.

Se nombra mediante un verbo en tercera persona del singular y las preposiciones que hagan falta.

Debe formar una frase con sentido al leerla con los roles.

Ejemplo de asociación

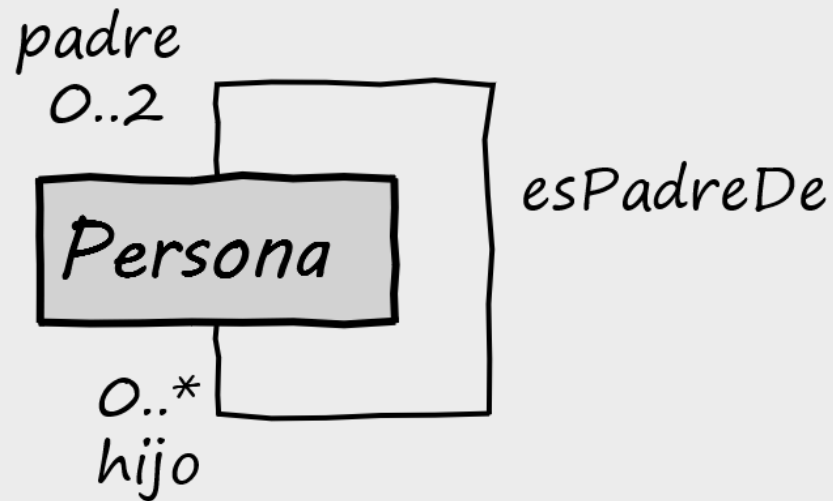


Rol de un extremo de una asociación

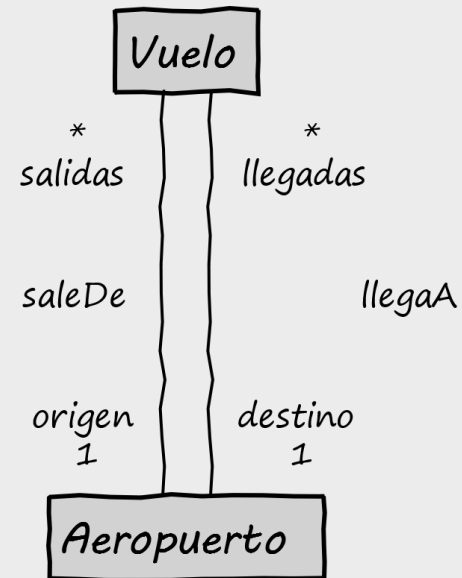
Papel que juega cada una de las clases que participan en una asociación.

Solo es necesario indicarlo en asociaciones de **una clase consigo misma** o cuando existe **más de una asociación entre dos clases**.

Ejemplos de roles I



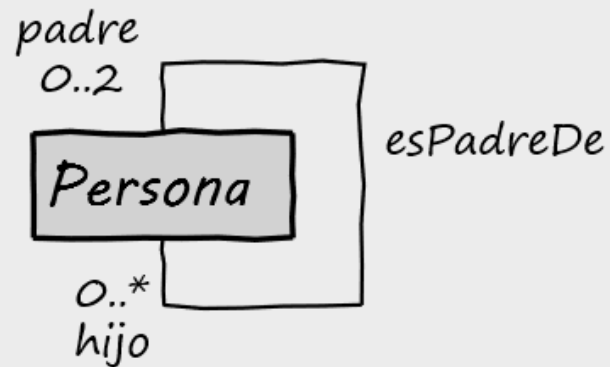
Ejemplos de roles II



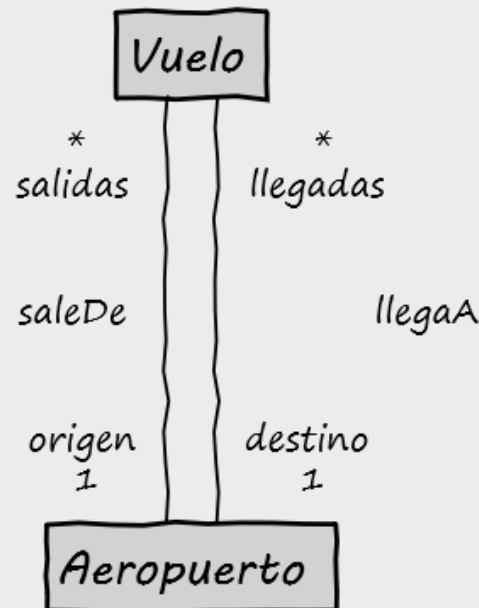
Multiplicidad

Dado un objeto de una clase, indica los números **mínimo** y **máximo** número de **objetos** de la otra clase con los que puede estar asociado.

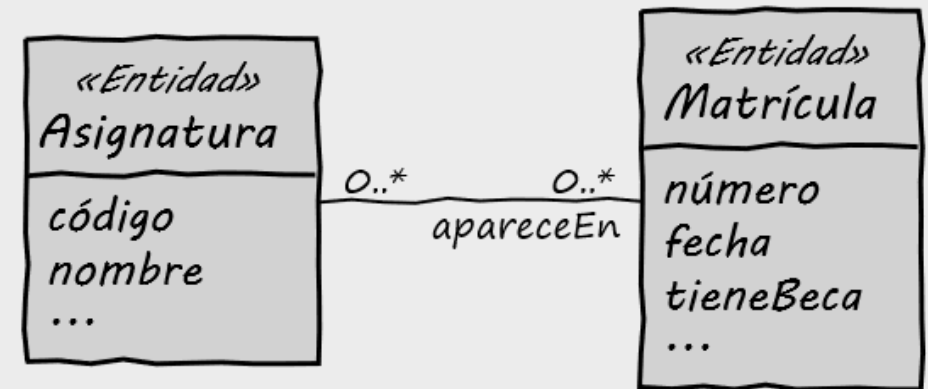
Ejemplos de roles I



Ejemplos de roles II



Ejemplo de asociación



Multiplicidad

0..1: opcional

0..*: opcional múltiple

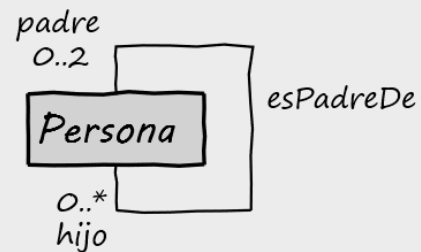
1..1: obligatoria

1..*: obligatoria múltiple

: equivalente a 0..

1: equivalente a 1..1

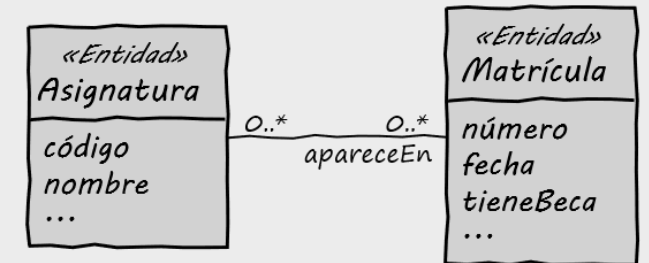
Ejemplos de roles I



Ejemplos de roles II



Ejemplo de asociación



Notación para asociaciones en UML

Nombre de rol

Indica el papel o rol que juega cada clase en la asociación. Solo en asociaciones de una clase consigo misma o si hay más de una asociación entre dos clases.

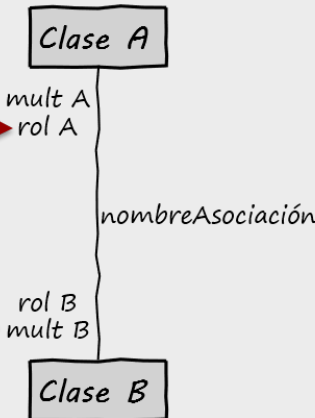
Nombre de la asociación

Es opcional y debe ser una forma verbal que tenga sentido al leerla con los roles. Se lee de izqda. a dcha. y de arriba a abajo. Si se debe leer de otra forma se debe indicar la dirección.

Notación para clases UML III

Multiplicidad

Indica los números mínimo y máximo de instancias de la clase que se interrelacionan con una instancia concreta de la otra clase. En multiplicidades múltiples, se puede indicar orden mediante **{ordered}**.



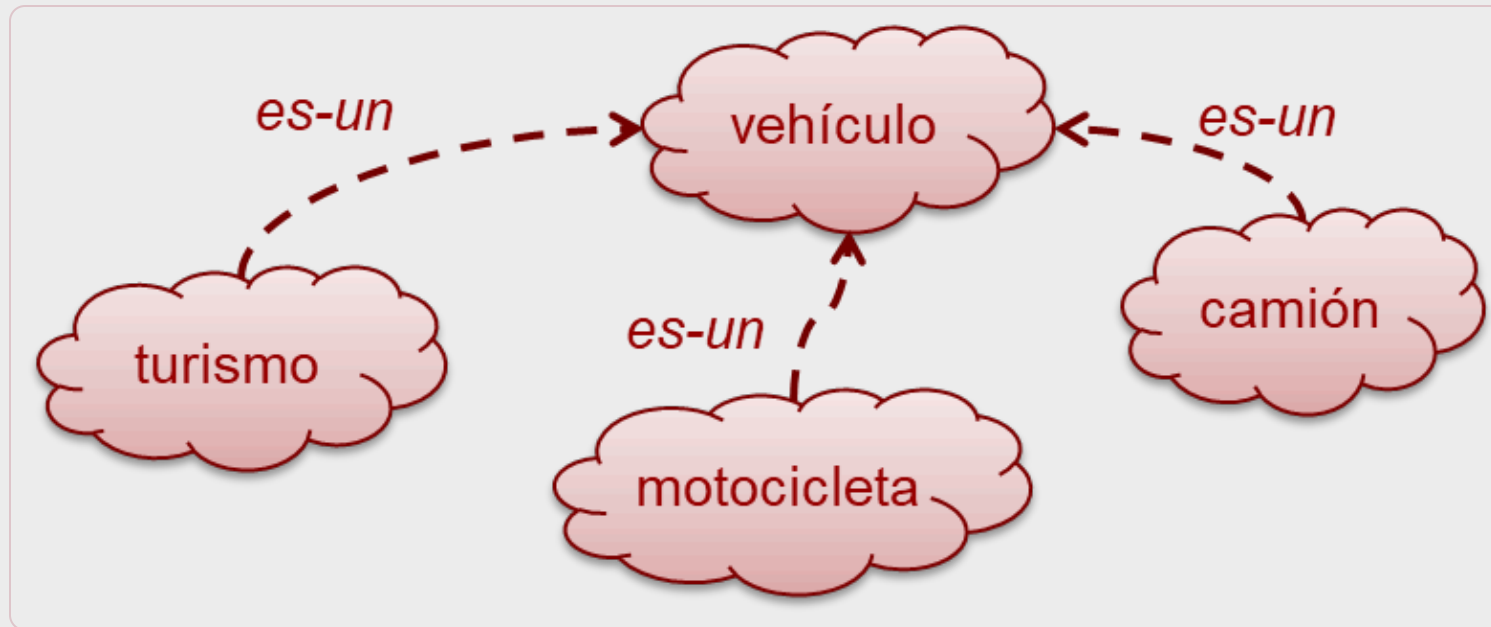
Valores habituales

- 0..1 : opcional
- 1..1 : obligatoria
- 0..* : múltiple opcional
- 1..* : múltiple obligatoria
- * : equivalente a 0..*
- 1 : equivalente a 1..1

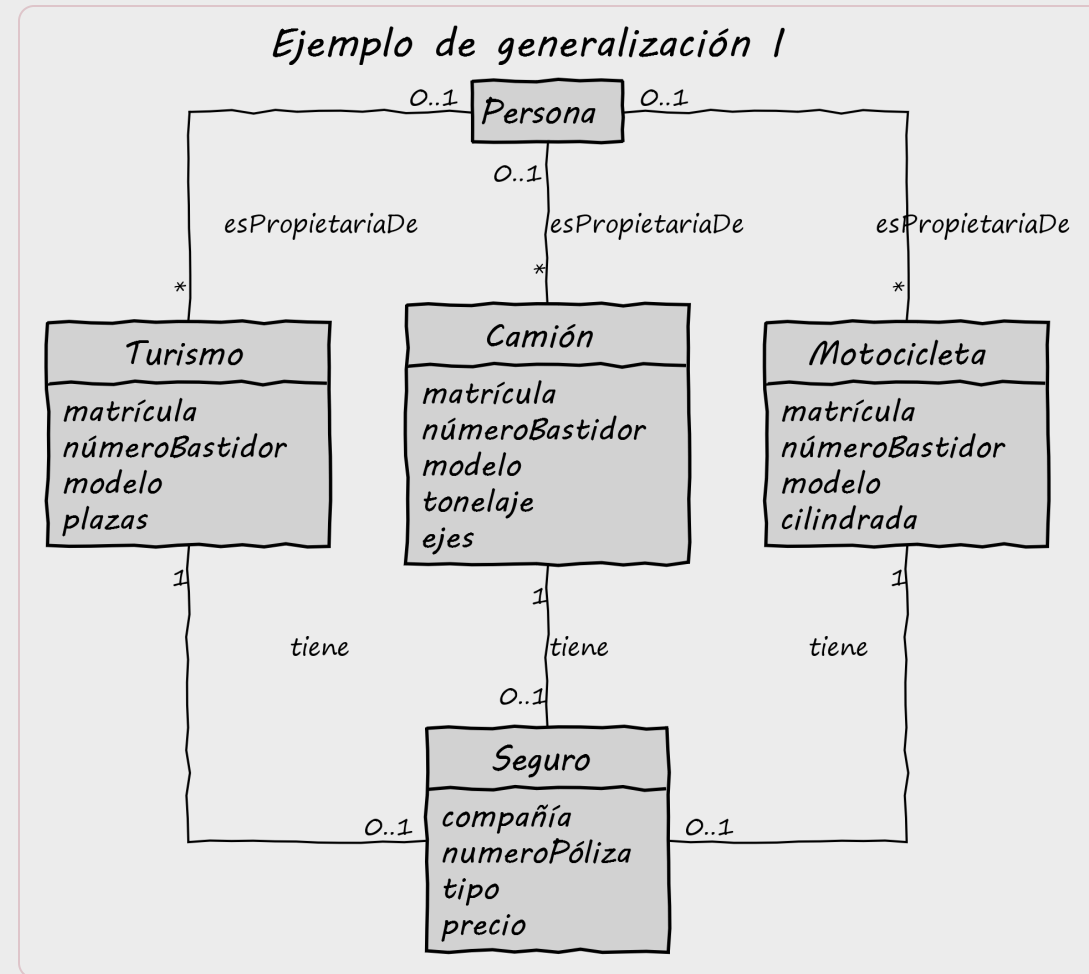
Generalización/especialización

A veces, algunos de los conceptos del dominio del problema presentan entre ellos relaciones del tipo es-un, por ejemplo:

Estos conceptos suelen tener propiedades comunes, que al modelarlos conceptualmente aparecen como atributos o asociaciones comunes.



Introducción al Modelado Conceptual

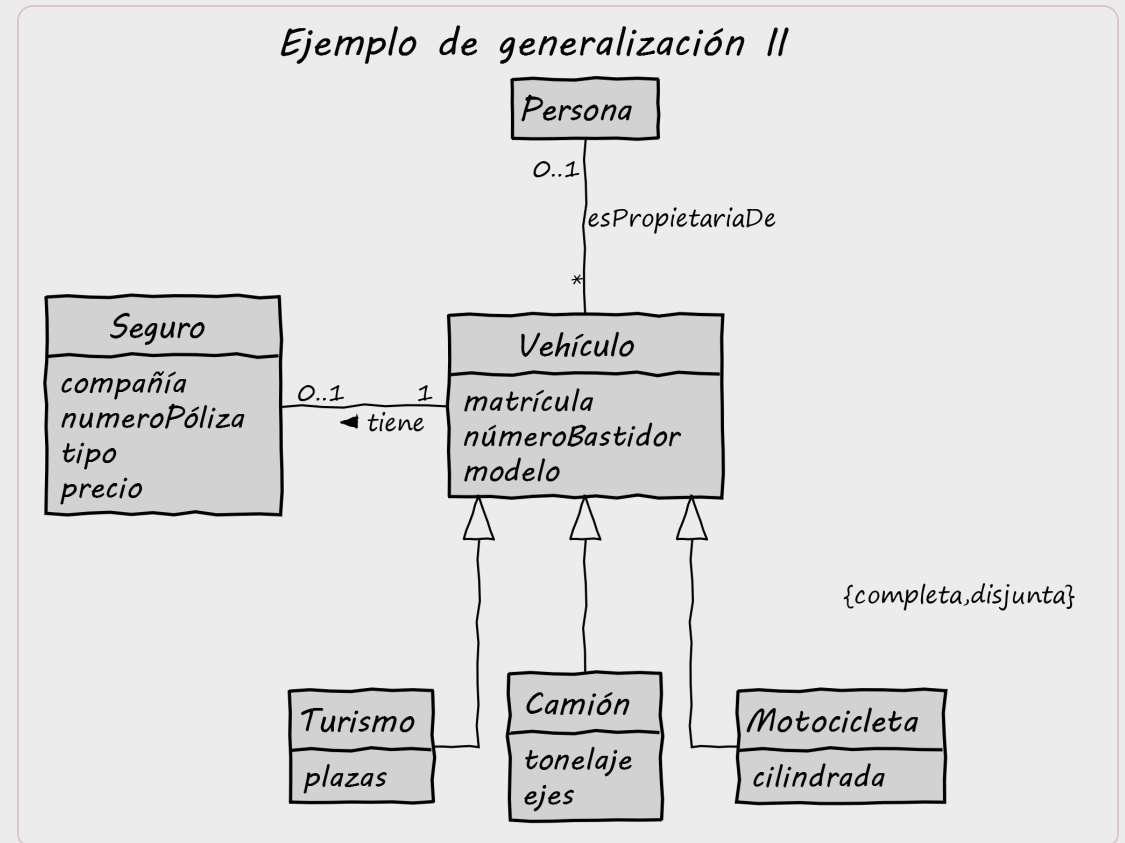


Generalización/especialización

La clase más general (la superclase), contiene todas las propiedades (atributos y asociaciones) comunes, que son heredados por las clases más específicas (las subclases).

Todas las instancias de las subclases se consideran también instancias de la superclase.

La generalización es una relación transitiva y antisimétrica.

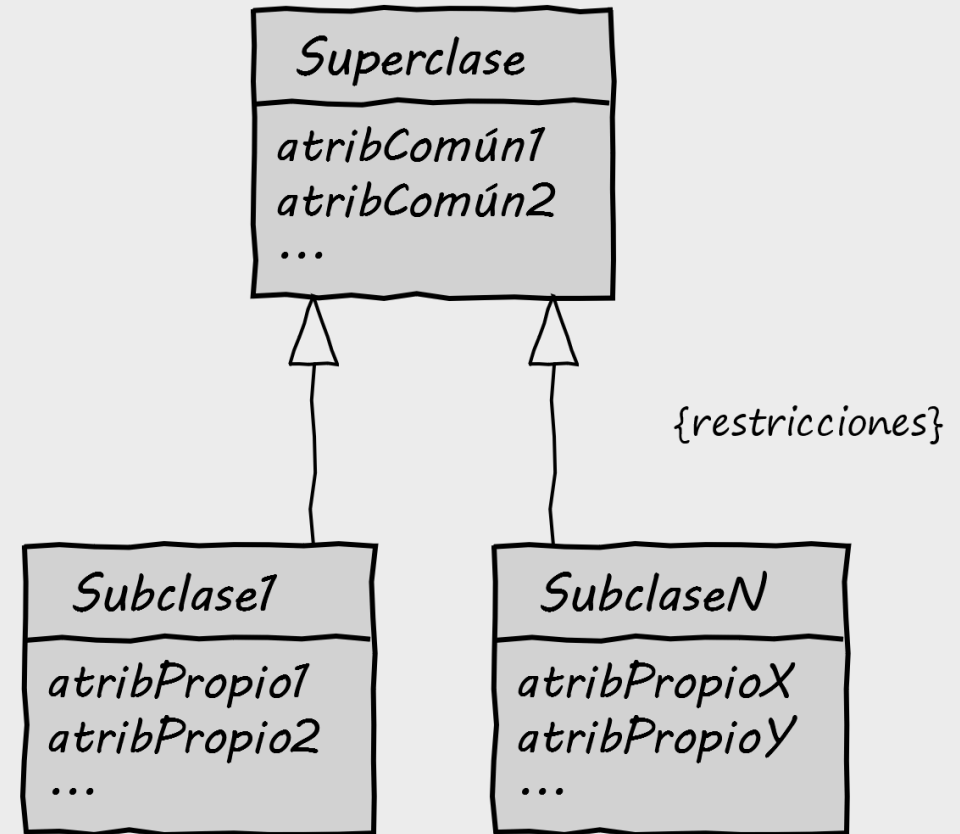


Notación para la generalización en UML

Clasificación completa/incompleta (restricciones)

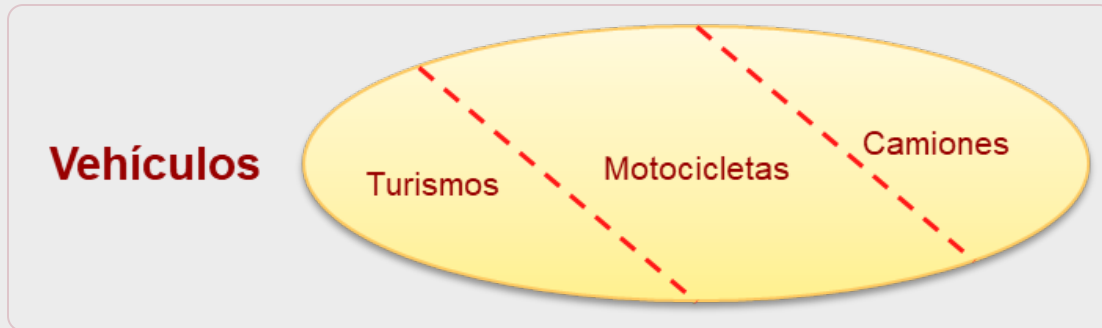
- **{completa}**: las instancias de la superclase deben ser instancias de al menos una subclase, la superclase es abstracta.
- **{incompleta}**: puede haber instancias de la superclase que no lo sean de ninguna subclase.
- **{disjunta}**: las instancias de la superclase pueden ser instancias de una sola subclase.
- **{solapada}**: las instancias de la superclase pueden ser instancias de una o más subclases.

Notación para clasificación UML 1

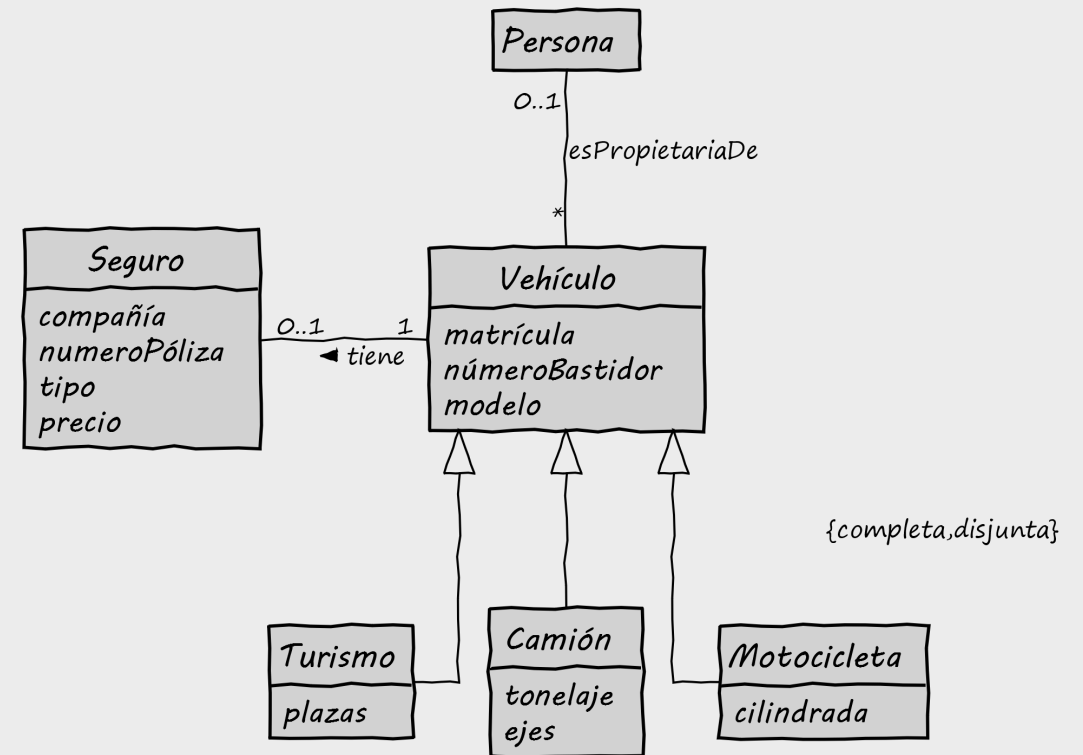


Notación para la generalización en UML

{completa, disjunta} implica una partición del conjunto de instancias de la superclase.

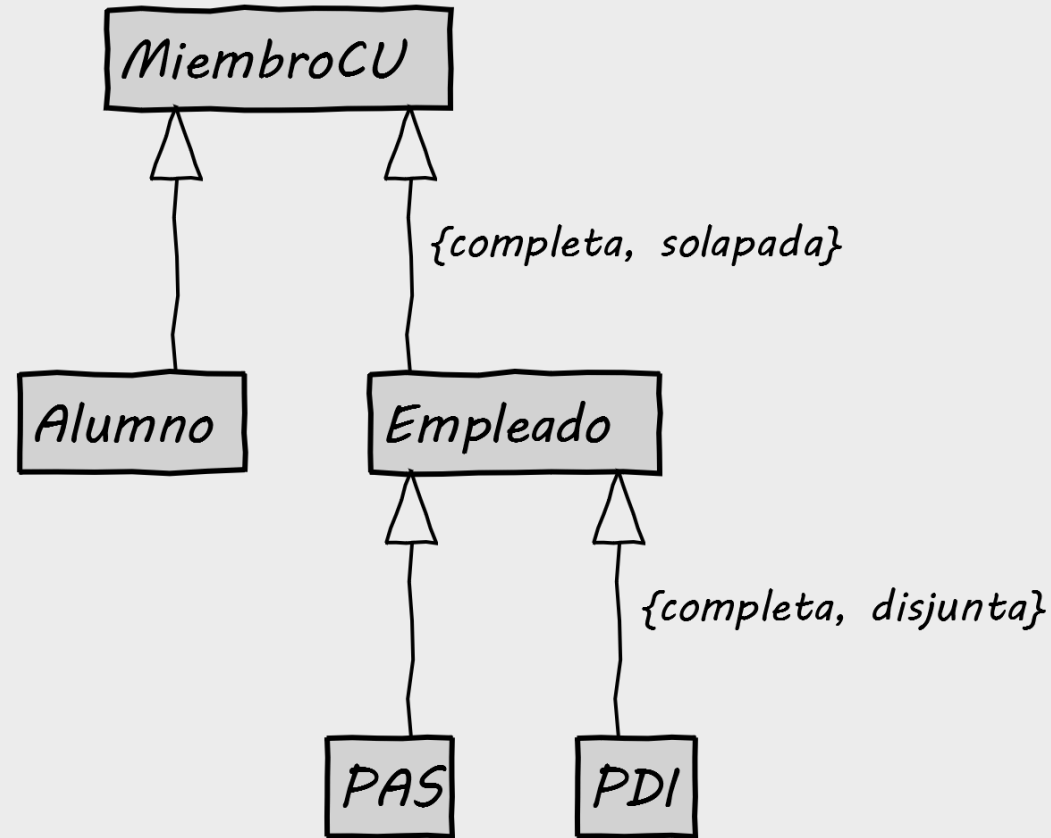


Ejemplo de generalización II



Notación para la generalización en UML

Notación para clasificación UML 2

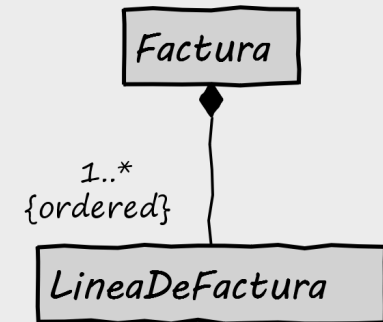


Composición

Asociación especial que representa el concepto de **ser-parte-de** o de **estar-compuesto-por**:

- Una parte sólo puede pertenecer a un todo.
- Una parte no puede existir sin pertenecer a un todo.
- Es una relación transitiva y antisimétrica.
- Puede ser recursiva.
- La eliminación del todo implica la eliminación de todas sus partes.

Ejemplo de composición

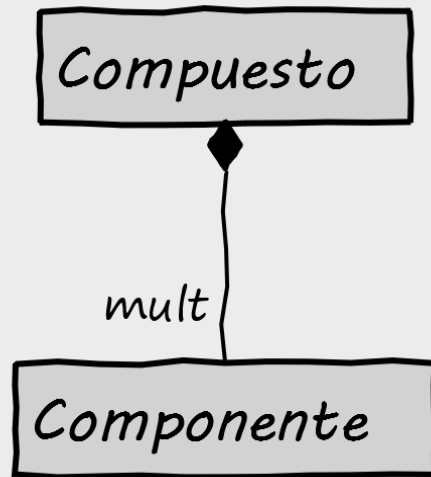


Notación para la composición en UML

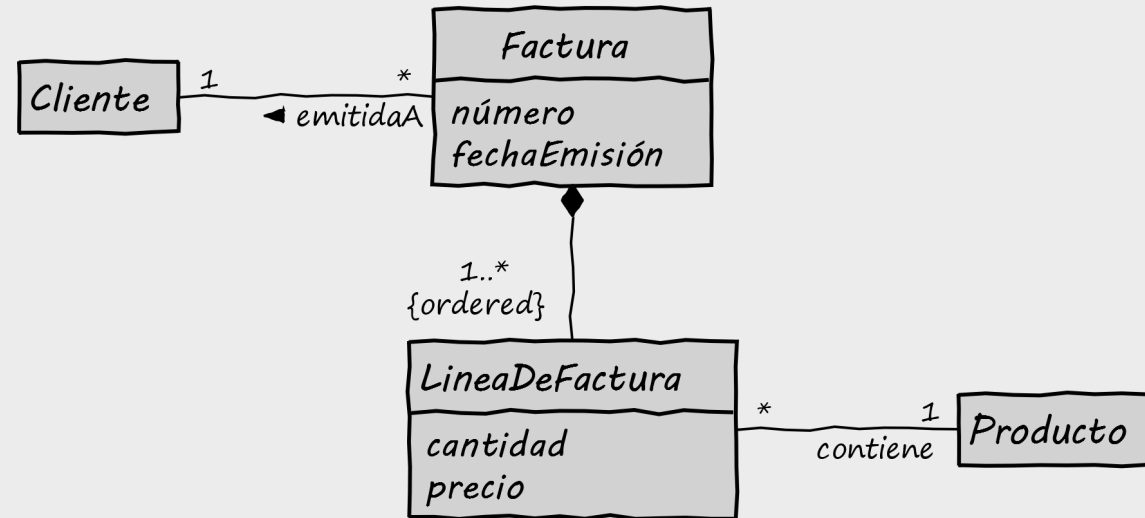
Compuesto: El rombo negro identifica al compuesto. Su multiplicidad es siempre 1, por lo que no es necesario indicarla.

Componente: La multiplicidad del componente puede ser variable, por tanto, hay que indicarla

Notación composición



Ejemplo Factura

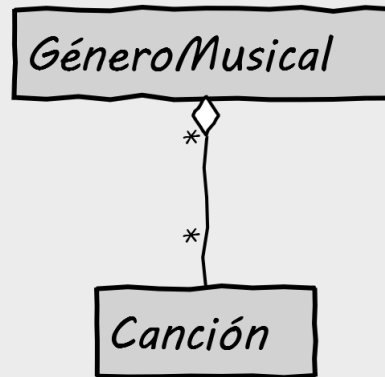


Agregación

Es similar a la composición, pero más débil:

- Una parte puede pertenecer a uno, varios o ningún todo.
- Una parte sí puede existir sin pertenecer a un todo.
- La eliminación del todo no implica la eliminación de todas sus partes.
- Sigue siendo transitiva, antisimétrica y recursiva.

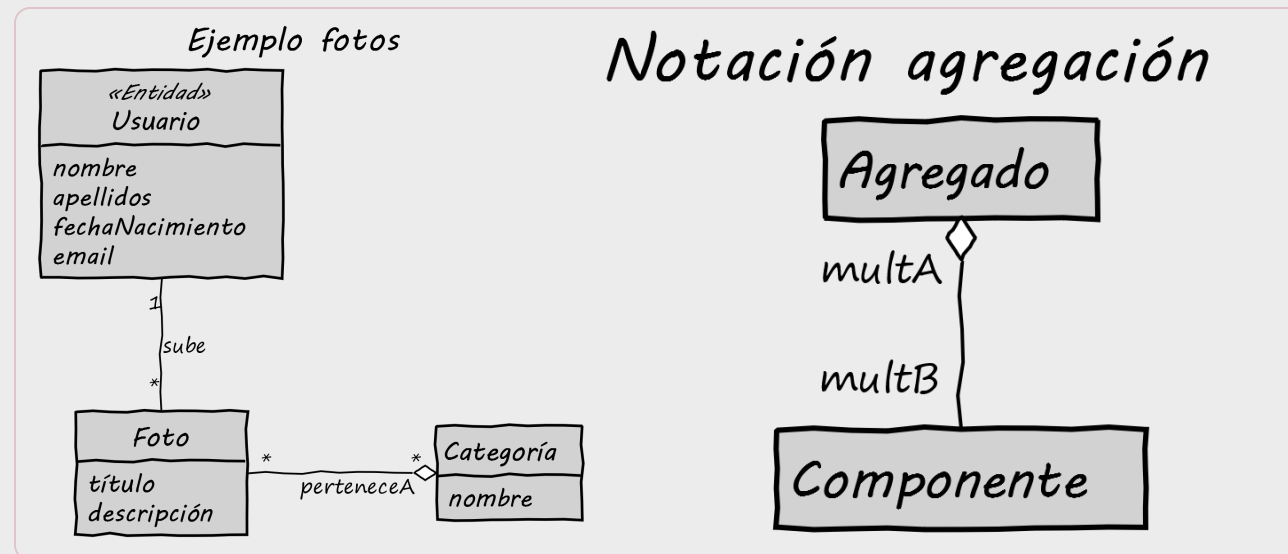
Ejemplo de agregación



Notación para la agregación en UML

Agregado: El rombo blanco identifica al agregado. A diferencia de la composición, puede tener cualquier multiplicidad (un componente puede pertenecer a más de un agregado)

Componente: La multiplicidad del componente puede ser variable, por tanto, hay que indicarla



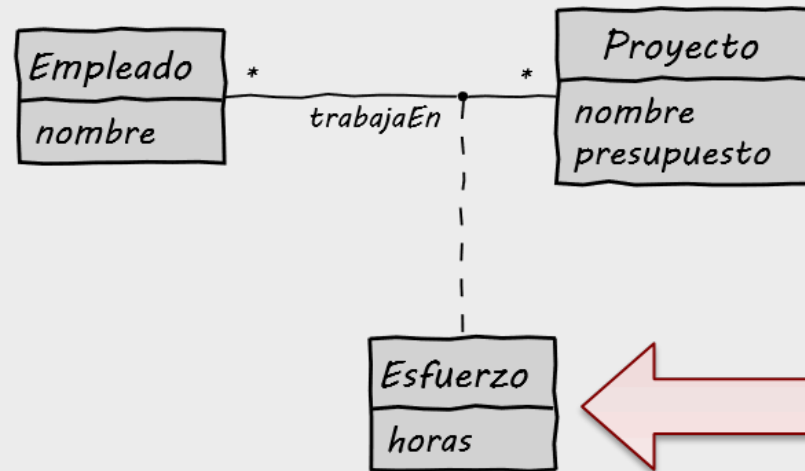
Notación para clases asociación en UML

Ejemplo clase asociación 1



¿cuántas horas
trabaja cada
empleado en
cada proyecto?

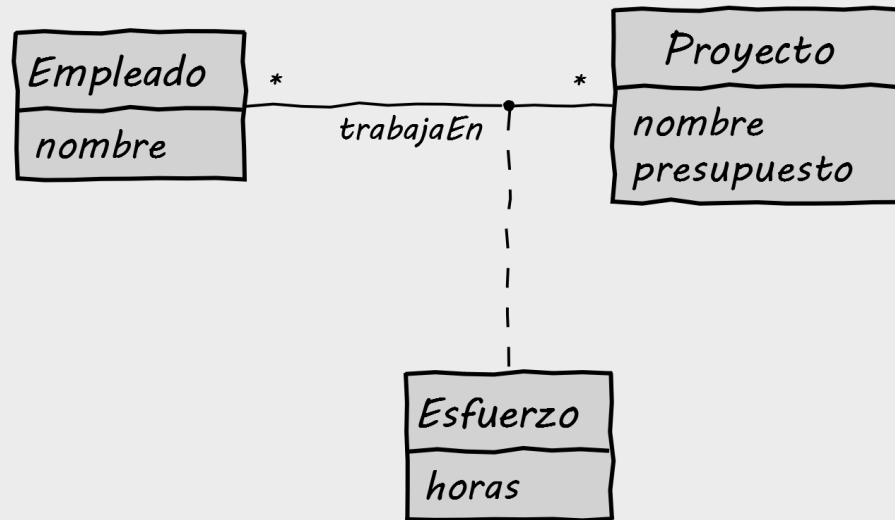
Ejemplo clase asociación 2



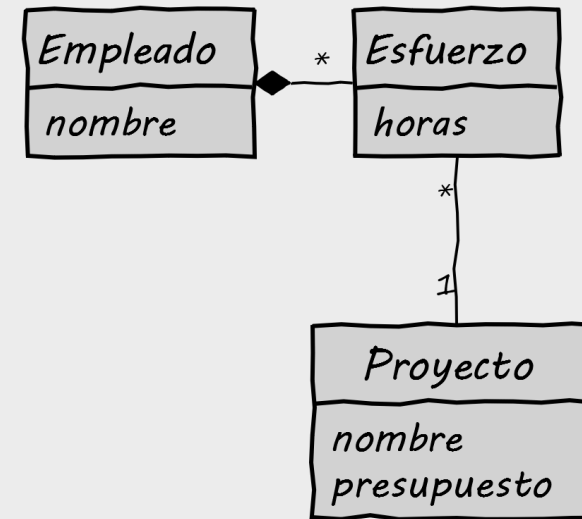
Notación para clases asociación en UML

La clase asociación puede modelarse como componente de cualquiera de las clases que participan en la asociación, pero sólo de una, ya que una parte sólo puede pertenecer a un todo.

Ejemplo clase asociación 2



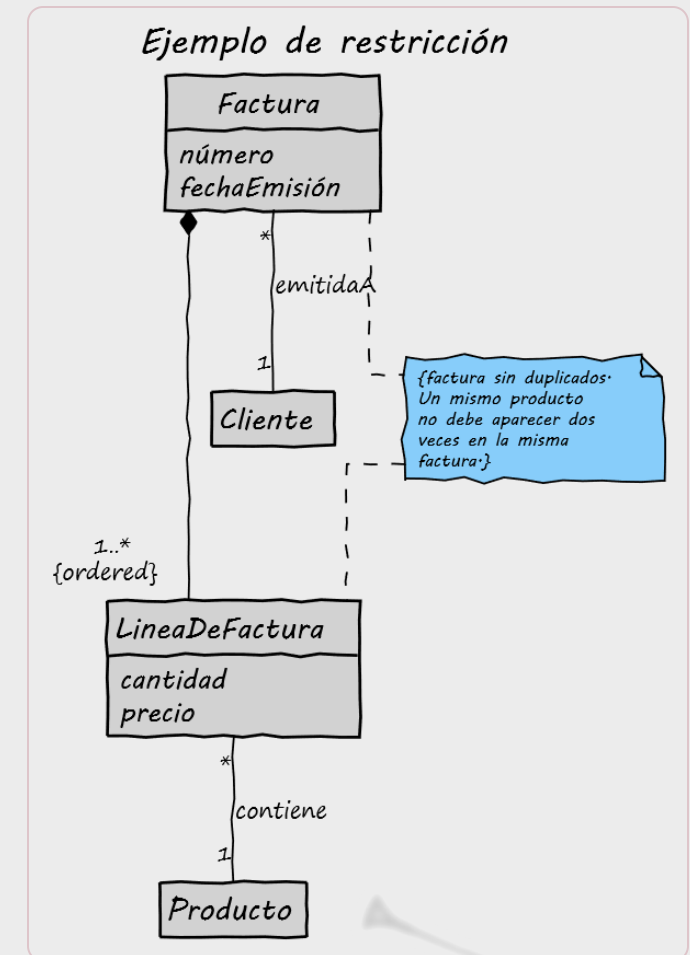
Ejemplo clase asociación con composición



Ambos modelos son equivalentes.

Notación para restricciones

Se representan mediante **notas**. El texto suele ir entre llaves, indicando tanto el nombre de la restricción como su descripción. Opcionalmente, se pueden enlazar a las entidades afectadas.



Notación para tipos enumerados

Definen un tipo que puede ser usado en los atributos de las clases entidades.

Se utiliza el estereotipo

«Enumerado» o «Enum»

Los atributos son los posibles valores.

Notación enumerados

«Enumerado»
Sexo

masculino
femenino

«Enumerado»
VíaPública

calle
plaza
avenida
carretera

«Enumerado»
Categoría

infantil
aventuras
cienciaFicción
drama

Contenido

Introducción

Notación UML

Diagramas de clases

Diagramas de objetos

Creación de modelos conceptuales

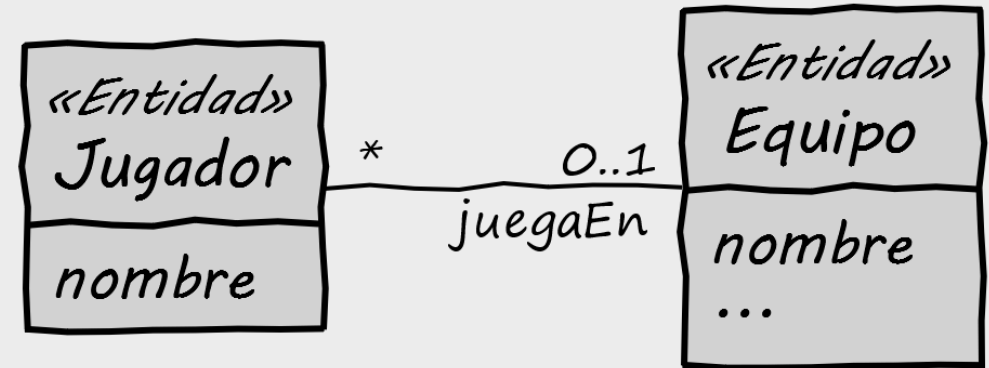


Objetos y enlaces

Objeto: Cada ocurrencia o instancia de una clase.

Enlaces: Cada ocurrencia o instancia de una asociación.

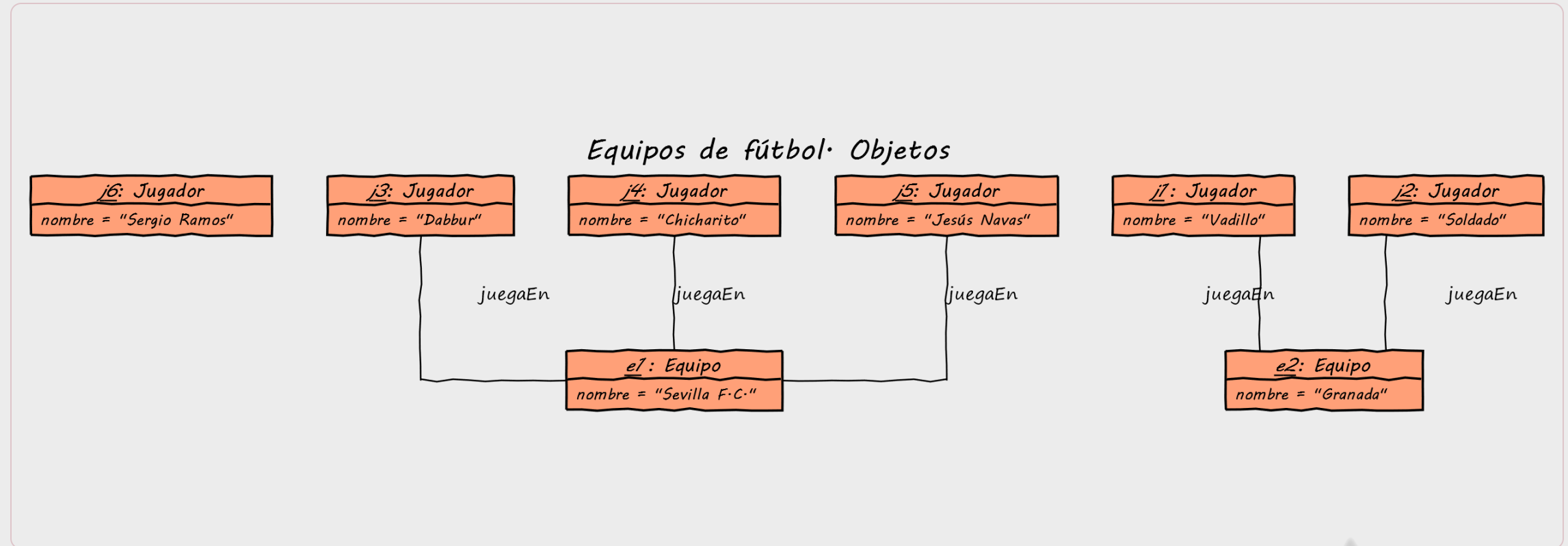
Equipos de fútbol. Modelo



Objetos y enlaces

Objeto: Cada ocurrencia o instancia de una clase.

Enlaces: Cada ocurrencia o instancia de una asociación.



Notación para objetos y enlaces en UML

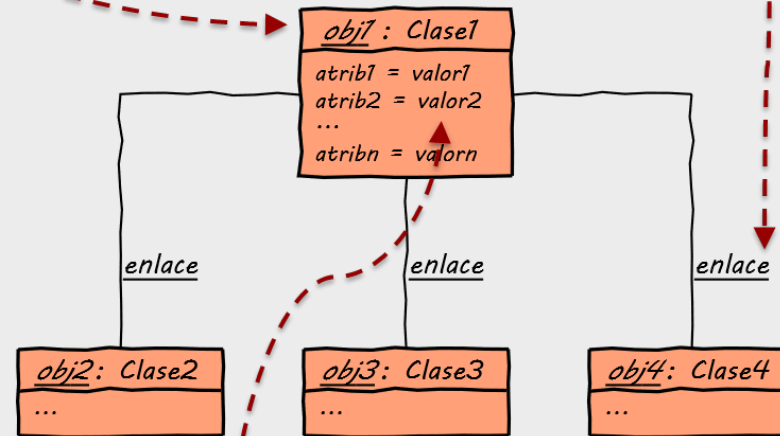
Nombre y clase del objeto

Deben estar subrayados para no confundirlos con una clase.

Enlaces

Se identifican mediante el nombre de la asociación subrayado.

Notación para objetos y enlaces UML

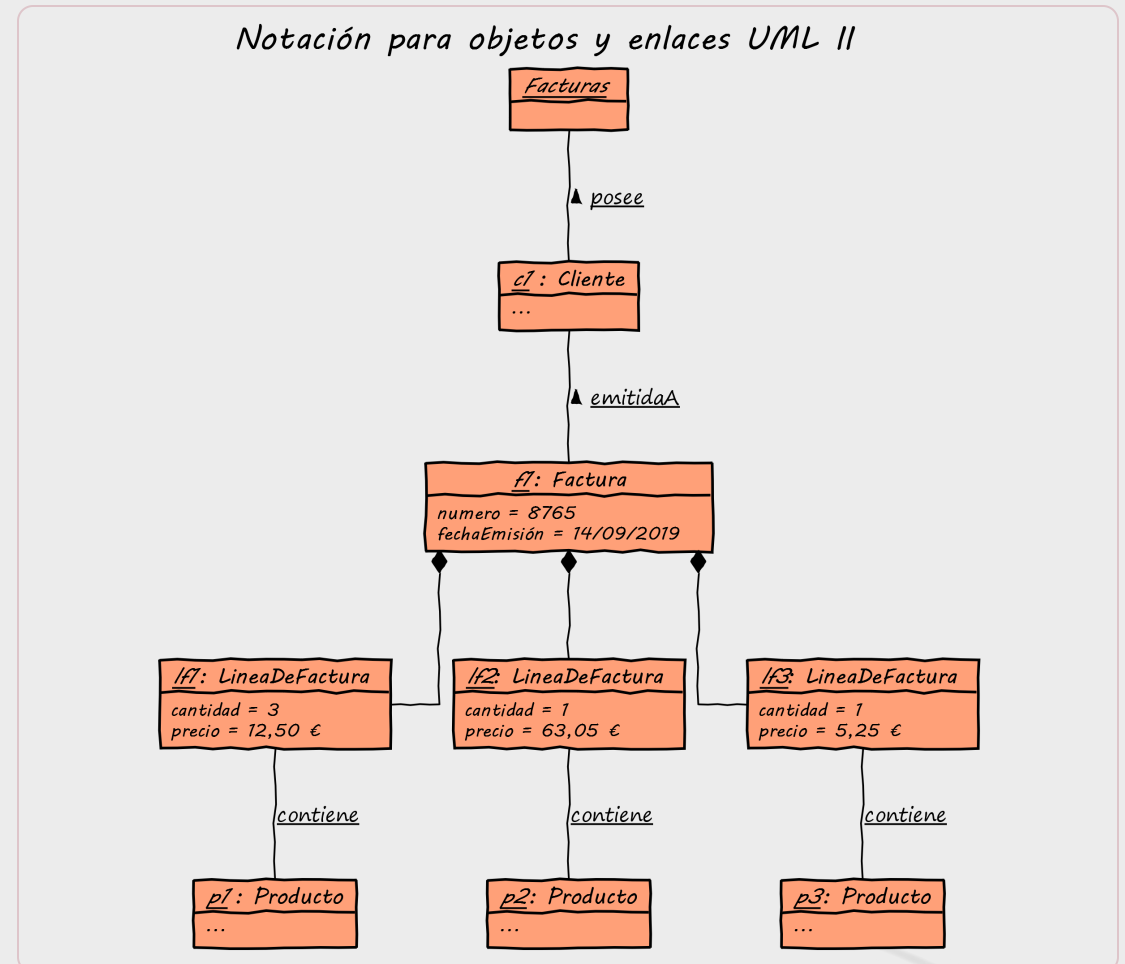


Nombre y valores de atributos

Opcionales, muestran los valores de los atributos del objeto.

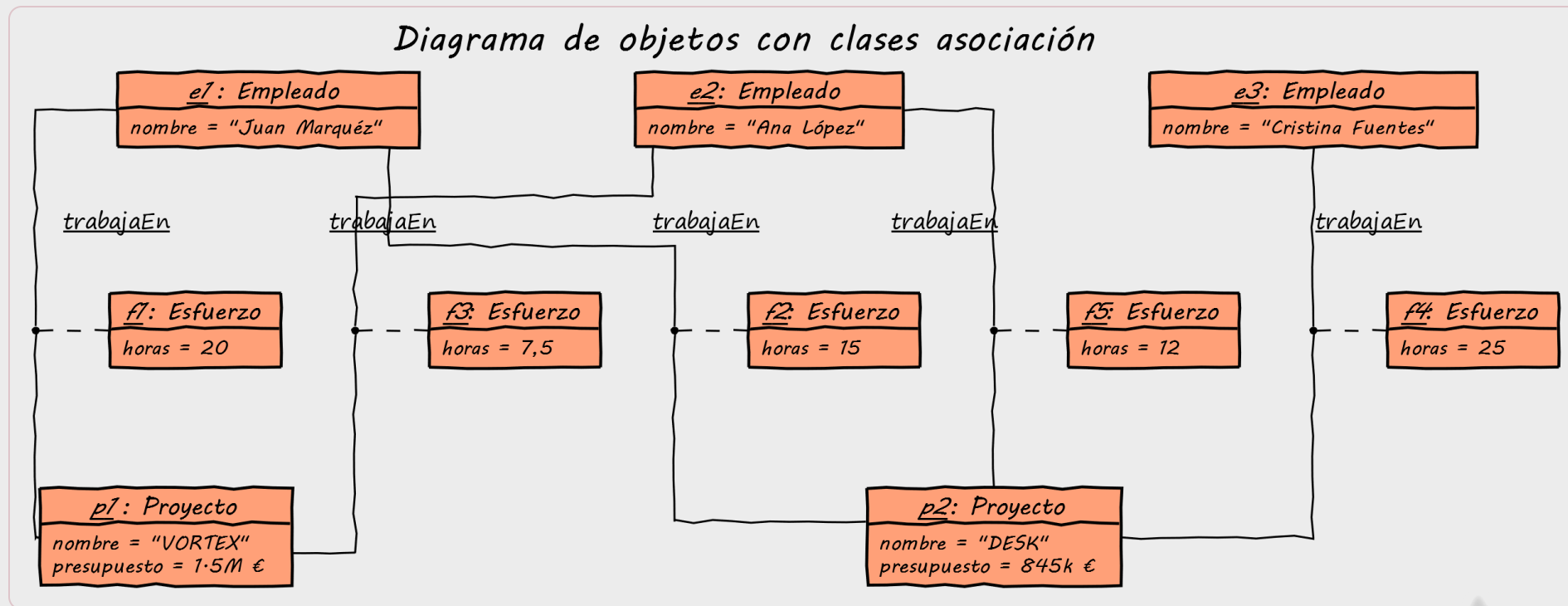
Notación para objetos y enlaces en UML

Facturas: múltiples objetos sin identificarlos individualmente.



Notación para clases asociación en UML

Ejemplo: las horas que trabaja un empleado en un proyecto no son una propiedad ni del empleado ni del proyecto, sino de la asociación entre ambos.



Contenido

Introducción

Notación UML

Diagramas de clases

Diagramas de objetos

Creación de modelos conceptuales



Creación de modelos conceptuales

Pasos recomendados:

1. Analizar la información sobre el **dominio** del problema (glosario) y los requisitos.
2. Identificar posibles **entidades** y **atributos**.
3. Identificar posibles **asociaciones**.
4. Construir incrementalmente el modelo conceptual e identificar las **multiplicidades** de las asociaciones.
5. Identificar **clasificaciones** entre entidades con propiedades (atributos y/o asociaciones) comunes.
6. Identificar **composiciones** entre entidades.
7. Añadir las **restricciones** que no puedan expresarse gráficamente.
8. Validar con posibles **escenarios** mediante diagramas de objetos.
9. Registrar todos aquellos **problemas semánticos** que deban ser aclarados con clientes y usuarios.

Herramientas

Existen muchas herramientas para crear modelos UML.

- PlantUML: Lenguaje para describir modelos UML
- VS Code: Editor para escribir modelos
- Draw.io: Herramienta web con interfaz gráfica para desarrollar modelos

Editor en línea y documentación

Enlace | Documentación

The screenshot displays the PlantUML Web Server interface in a web browser. The browser's address bar shows the URL `plantuml.com/plantuml/uml/SyfFKj2rKt3CoKnELR1Io4ZDoSa70000`. The interface includes a sidebar with sponsored searches such as "create diagram online", "uml diagram tool", "uml online", "online website testing", and "sequence diagram". The main area contains a text editor with the following code:

```
@startuml
Bob -> Alice : hello
@enduml
```

Below the editor, there are buttons for "Submit", "Switch layout", and "Pin it", along with links to "View as PNG", "View as SVG", and "View as ASCII Art". The rendered output at the bottom shows a sequence diagram with two participants, Bob and Alice, and a message labeled "hello" from Bob to Alice.

Instalación VS Code + PlantUML

Tutorial de instalación:

https://videos.us.es/playlist/dedicated/181227451/1_isfiic9l/1_8l51etme

(Disponible en el canal de IISI en videos.us.es)

Conclusiones

El **modelado conceptual** se usa en **análisis** para comprender el **dominio del problema** y reducir errores antes de implementar.

Cada elemento del modelo debe tener **trazabilidad explícita** con **requisitos de información** y **reglas de negocio**.

El estándar principal utilizado es **UML**, especialmente con **diagramas de clases** y **diagramas de objetos**.

Modelar bien mejora la **comunicación** con usuarios y facilita una solución más **coherente**.



Conclusiones

En diagramas de clases, los conceptos clave son **clase entidad, atributo, asociación, rol y multiplicidad**.

La **generalización/especialización** permite factorizar propiedades comunes y expresar jerarquías del dominio.

Composición y agregación representan relaciones todo-parte con distinta **fuerza semántica**.

Las **restricciones** y tipos **enumerados** completan la semántica cuando la notación gráfica no es suficiente.



Conclusiones

Los diagramas de **objetos y enlaces** permiten validar el modelo con **escenarios concretos**.

La creación del modelo debe ser **incremental**: identificar entidades, asociaciones, multiplicidades, clasificaciones y restricciones.

Es clave registrar **problemas semánticos** para validarlos con clientes y usuarios antes de avanzar.

Un modelo conceptual sólido es la base para el posterior diseño de **bases de datos** y del **sistema de información**.





Gracias

Universidad de Sevilla

**Departamento de Lenguajes y Sistemas
Informáticos**

UNIVERSIDAD DE SEVILLA