

Tema 5: Introducción al modelo conceptual

Introducción a la Ingeniería del Software y los Sistemas de Información I
Ingeniería Informática – Tecnologías Informáticas
Departamento de Lenguajes y Sistemas Informáticos



1.Introducción

2.Notación UML

1.Diagramas de clases

2.Diagramas de objetos

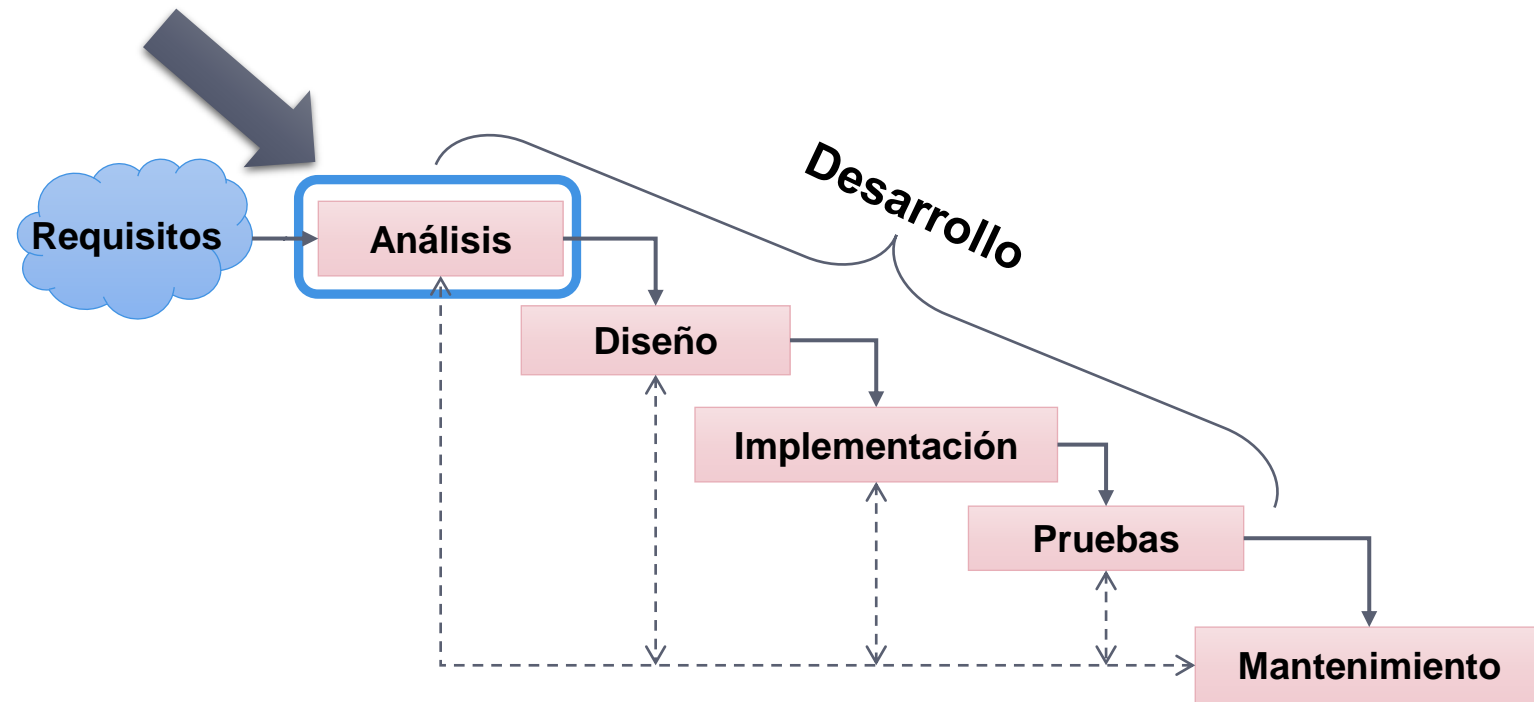
3.Creación de modelos conceptuales

4.Matrices de trazabilidad

5.Herramientas para crear modelos UML

¿Cuándo se usa el modelado conceptual?

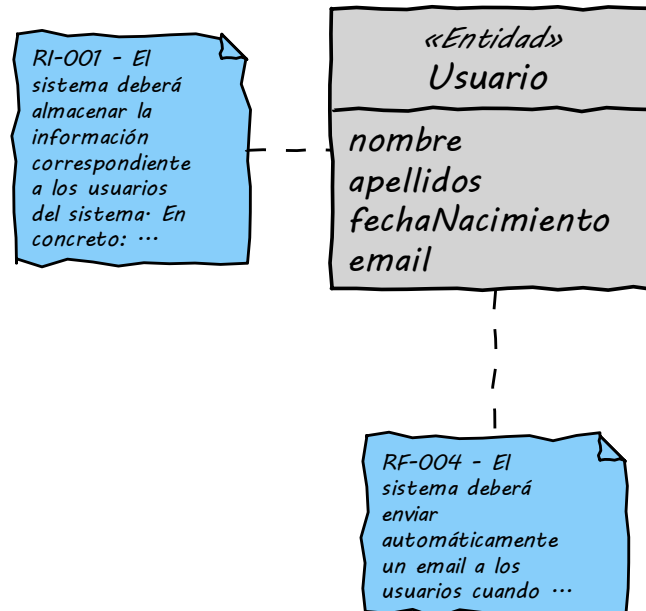
- Independientemente del ciclo de vida, se utiliza durante el último estadio del análisis.



Trazabilidad hacia requisitos

Todo elemento de un modelo conceptual debe estar **trazado** hacia aquellos requisitos que lo justifican, normalmente **requisitos de información y reglas de negocio**.

Ejemplo trazabilidad



Estándar para modelado conceptual: **UML** (*Unified Modeling Language*).

- Resultado de la fusión de varias propuestas previas.
- Gestionado por la **OMG** (*Object Management Group*).
- Ampliamente usando en la industria del software.
- Múltiples herramientas disponibles.
- Define 14 tipos de diagramas para modelar sistemas software (versión 2.5.1, diciembre 2017).
- 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

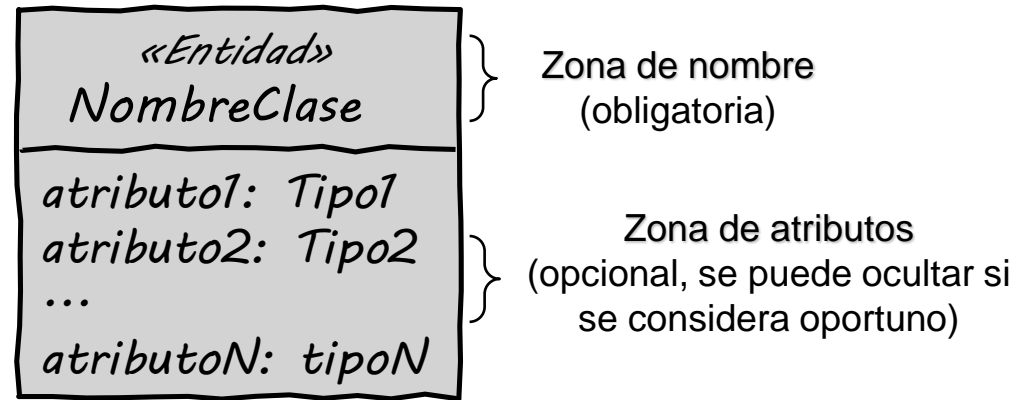
Clase Entidad

- 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

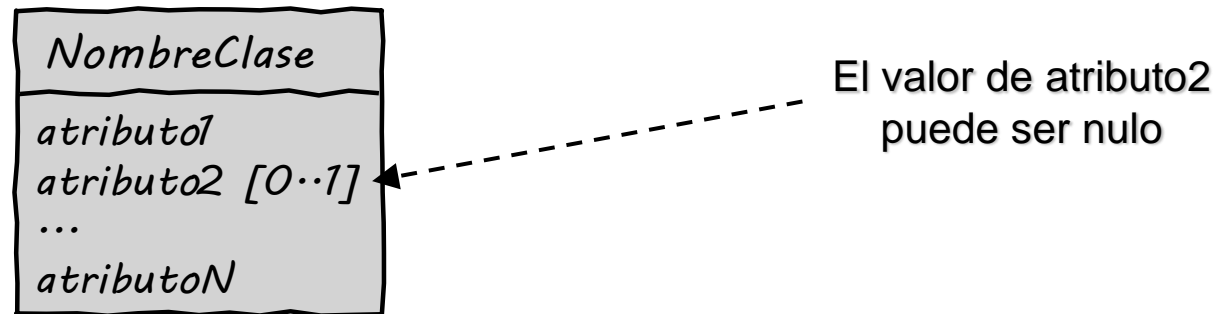
- Son **propiedades** asociadas a un concepto relevante del dominio del problema que el sistema debe almacenar porque así se ha especificado (o se deduce) en uno o más requisitos.
- Se nombran mediante un **sustantivo en singular**.
- Los **valores** de los atributos deben ser **atómicos**.

Notación para clases UML



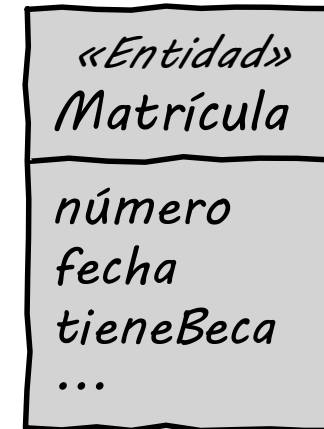
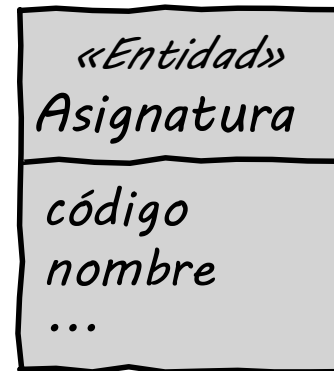
- Con el estereotipo «Entidad» se indica que la clase representa una entidad y no una clase de un lenguaje de programación orientado a objetos.
- En modelado conceptual se asumirá el estereotipo «Entidad» por defecto.
- El tipo de los atributos es opcional y se puede considerar una solución para la fase de diseño.

Notación para clases UML II



- 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**.

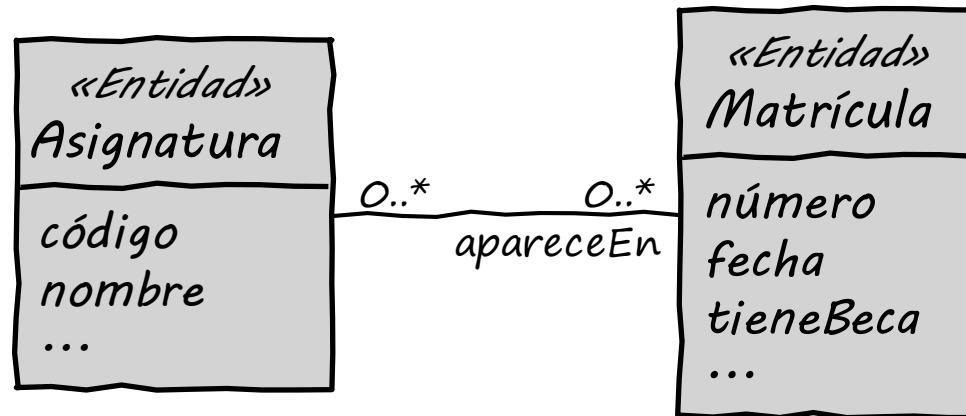
Ejemplo de clases



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.

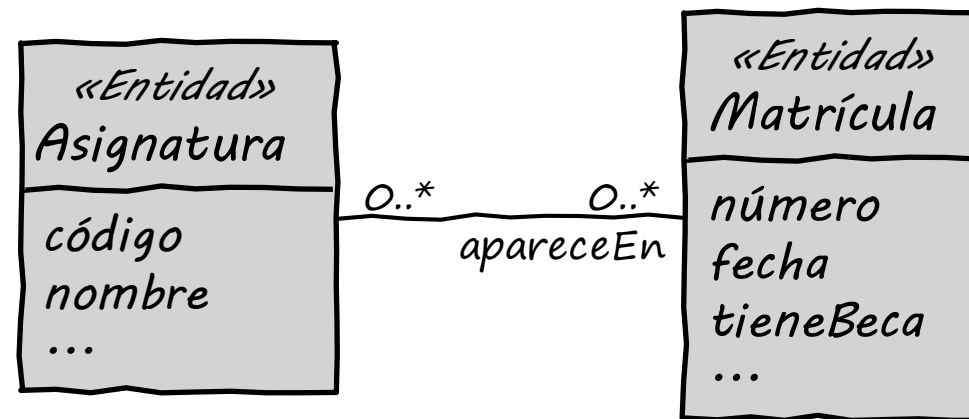
Ejemplo de asociación



Asociación entre clases entidades

- 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.

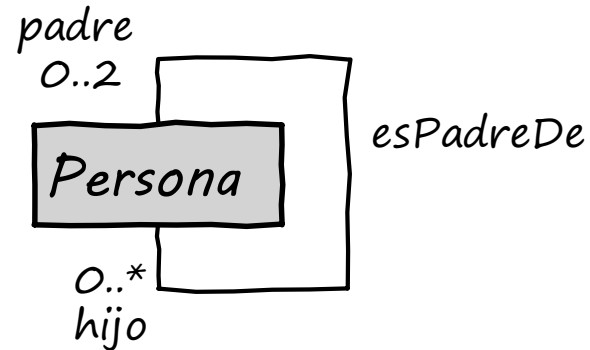
Ejemplo de asociación



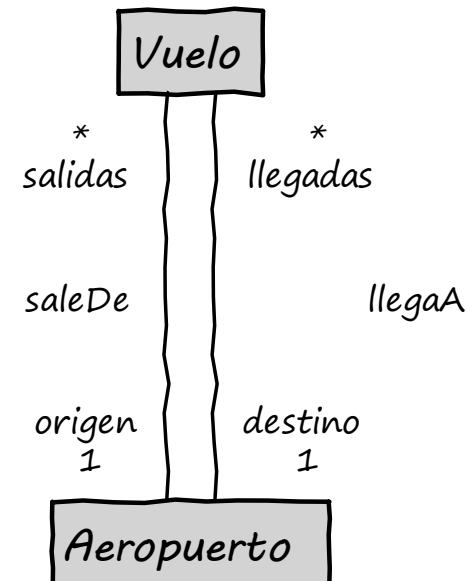
Rol de un extremo de una asociación

- **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



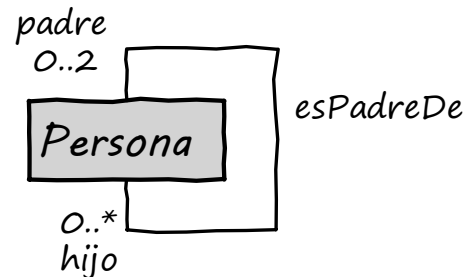
Ejemplos de roles II



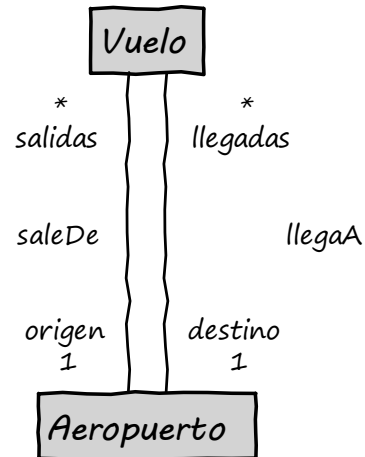
Multiplicidad

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.

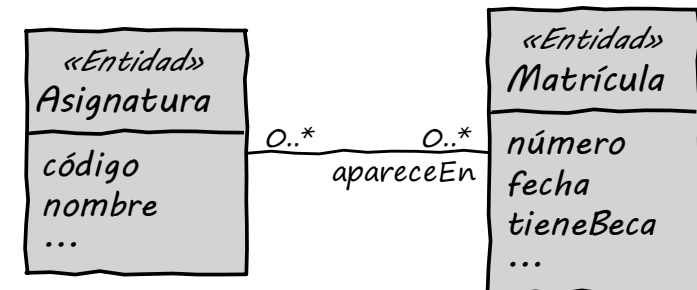
Ejemplos de roles I



Ejemplos de roles II



Ejemplo de asociación

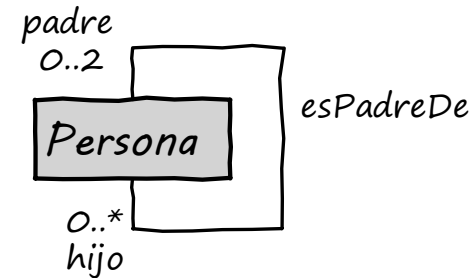


Multiplicidad

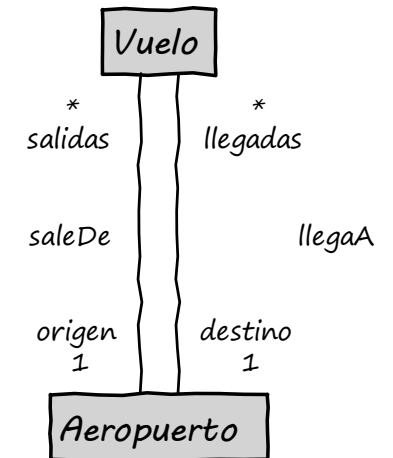
Valores habituales de multiplicidades

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

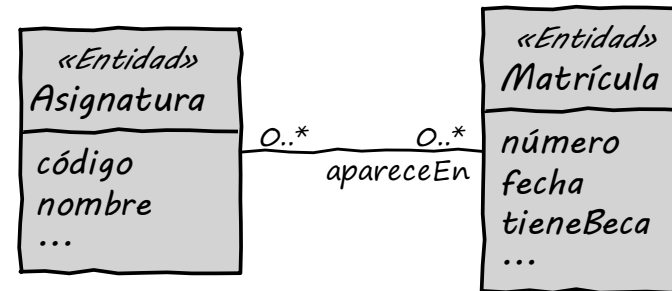
Ejemplos de roles I



Ejemplos de roles II



Ejemplo de asociación



Notación UML: Diagramas de clases

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.

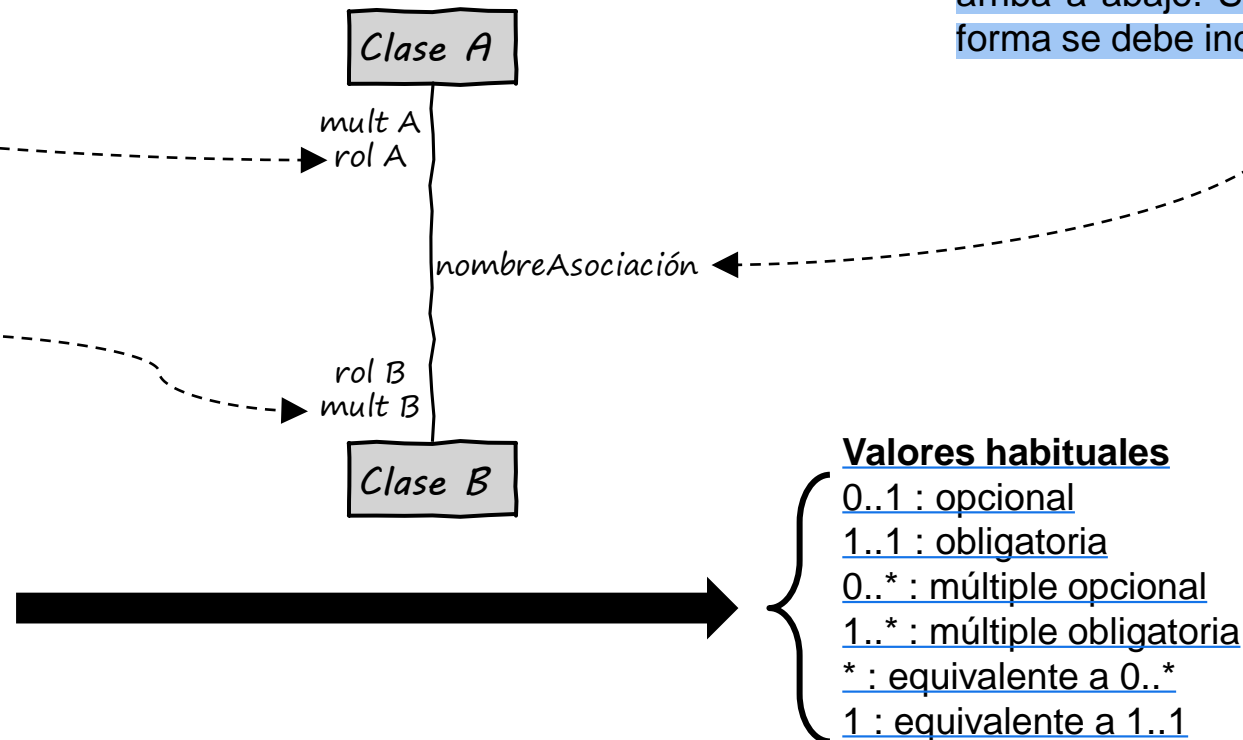
Nombre de la asociación

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

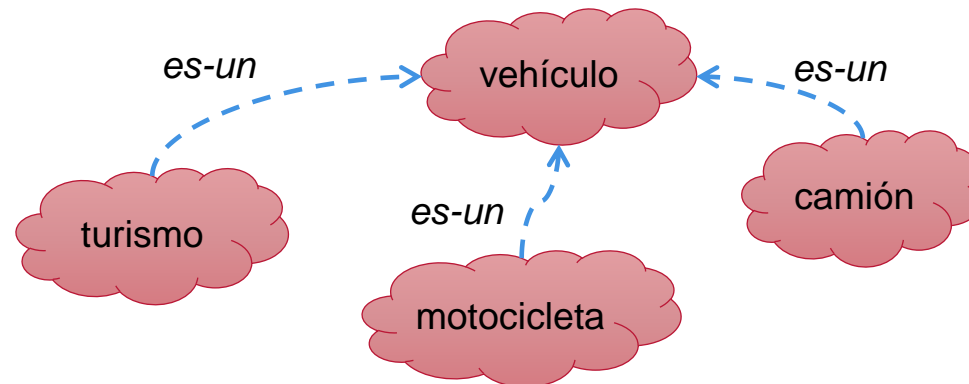
Multiplicidad

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}**.



Generalización/especialización

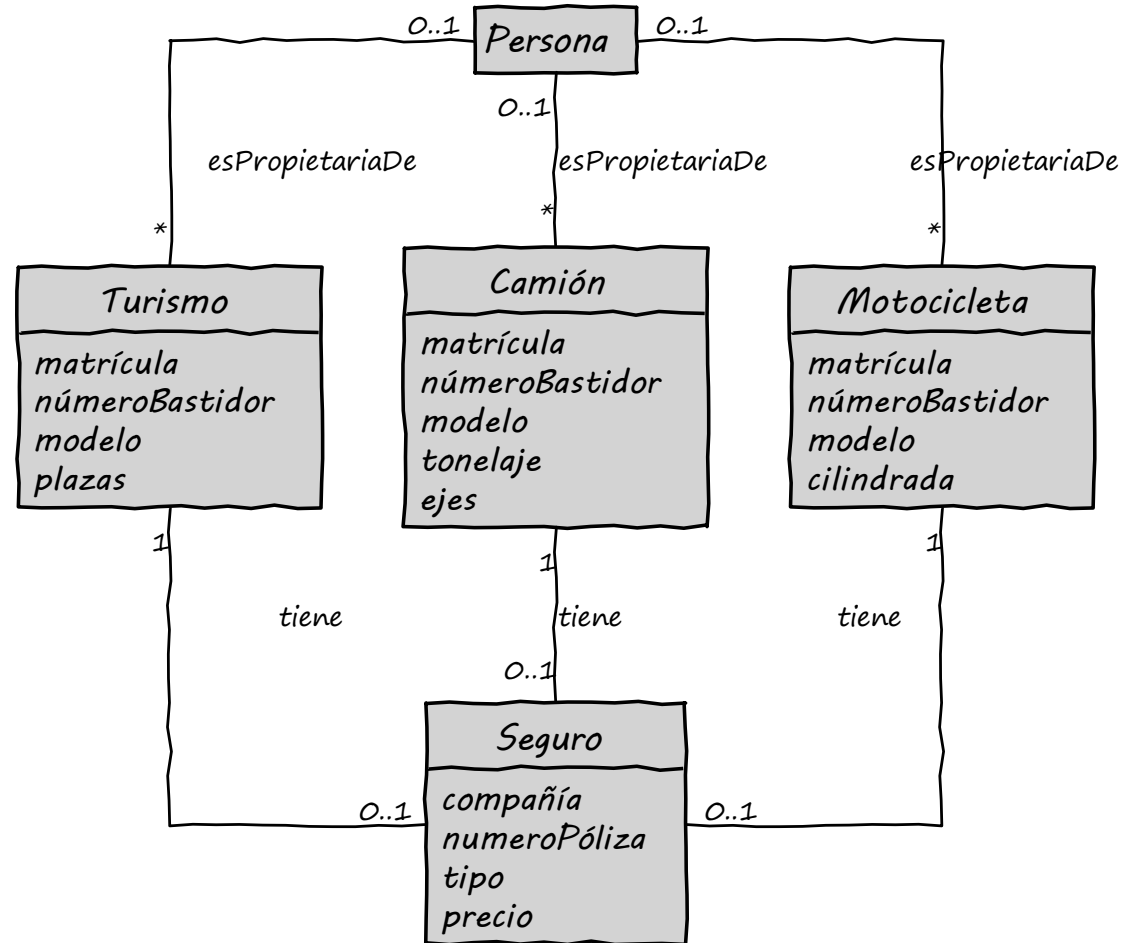
A veces, algunos de los conceptos del dominio del problema presentan entre ellos relaciones del tipo *es-un*, por ejemplo:



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

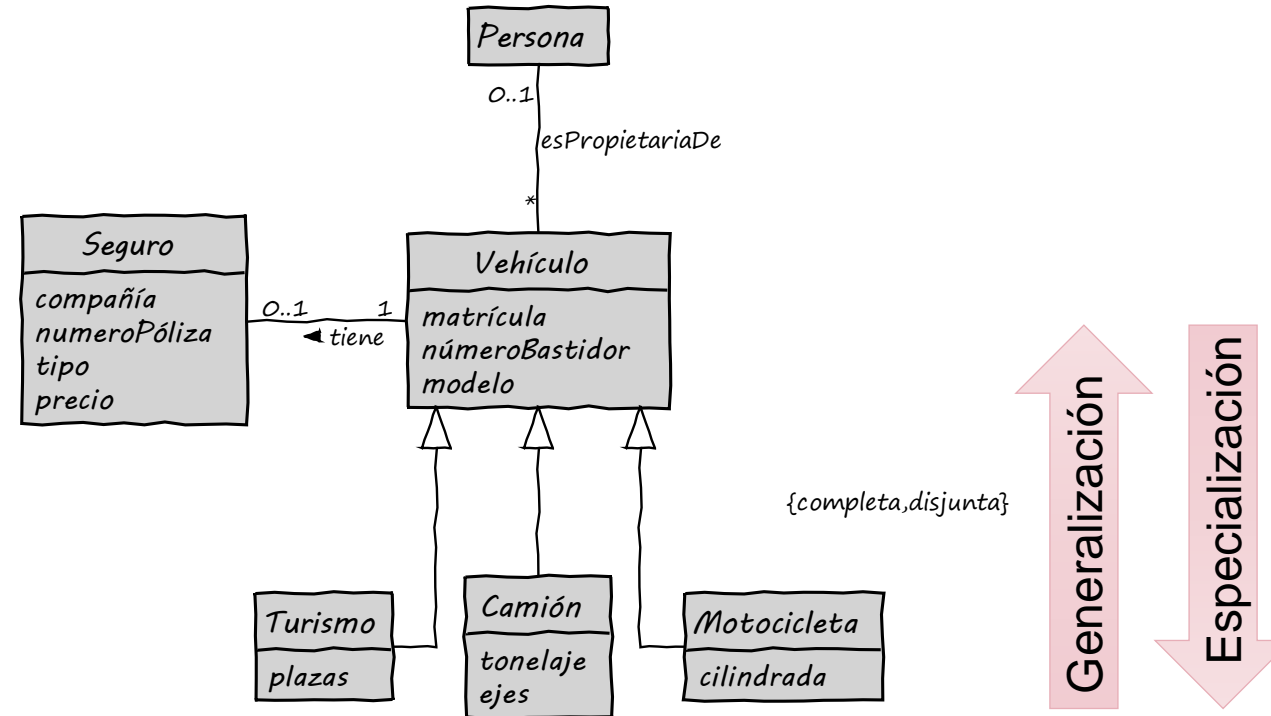
Notación UML: Diagramas de clases

Ejemplo de generalización I



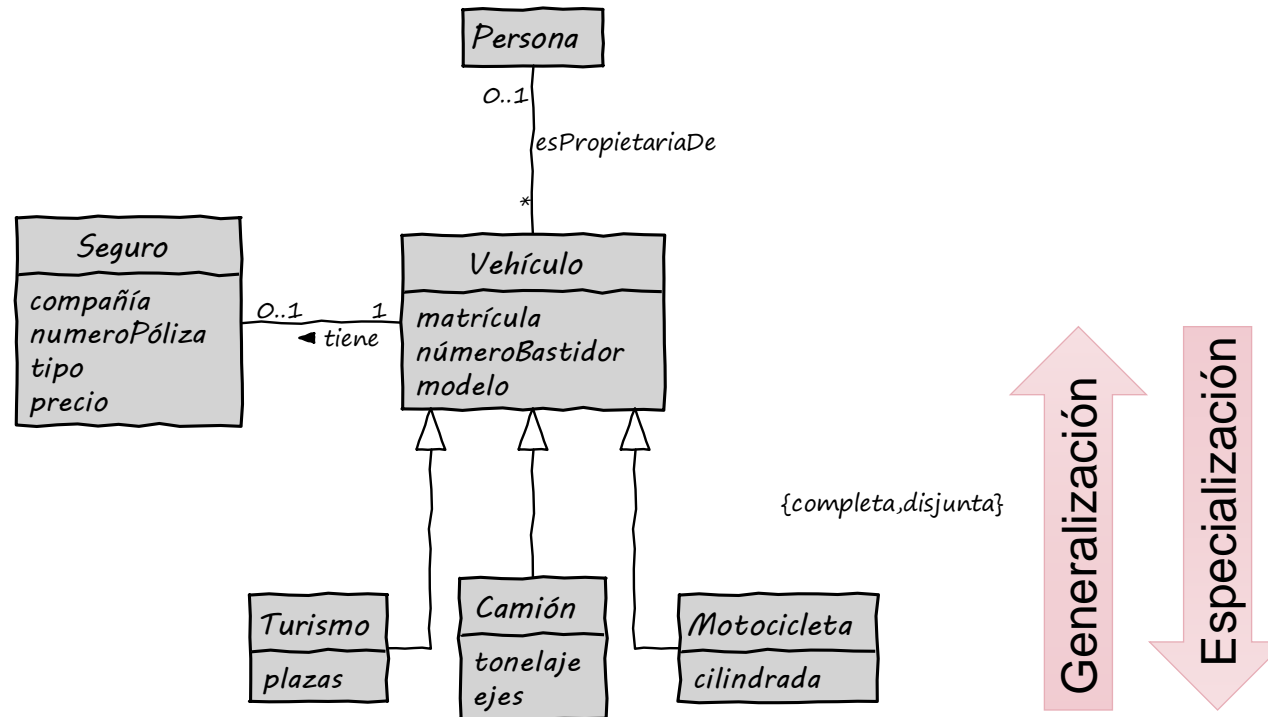
Notación UML: Diagramas de clases

Ejemplo de generalización II



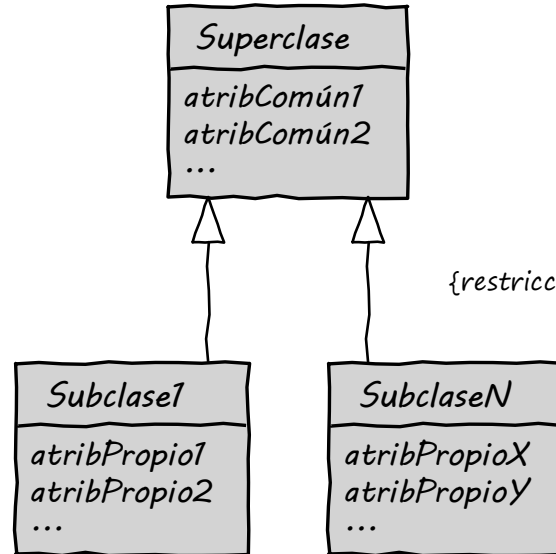
La clase más general (la **superclase**), contiene todas las propiedades (atributos y asociaciones) comunes, que son **heredados** por las clases más específicas (las **subclases**).

Ejemplo de generalización II



- Todas las instancias de las subclases se consideran también instancias de la superclase.
- La generalización es una relación transitiva y antisimétrica.

Notación para clasificación UML I



Restricciones

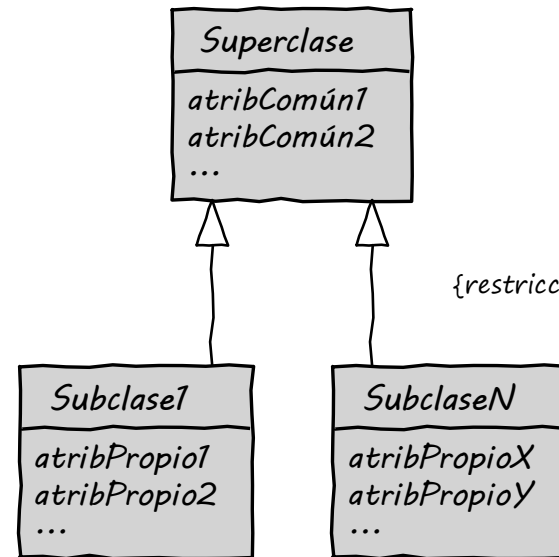
Indican si la clasificación es completa/incompleta y disjunta/solapada.

{restricciones}

- Clasificación completa/incompleta

- {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.

Notación para clasificación UML 1



Restricciones

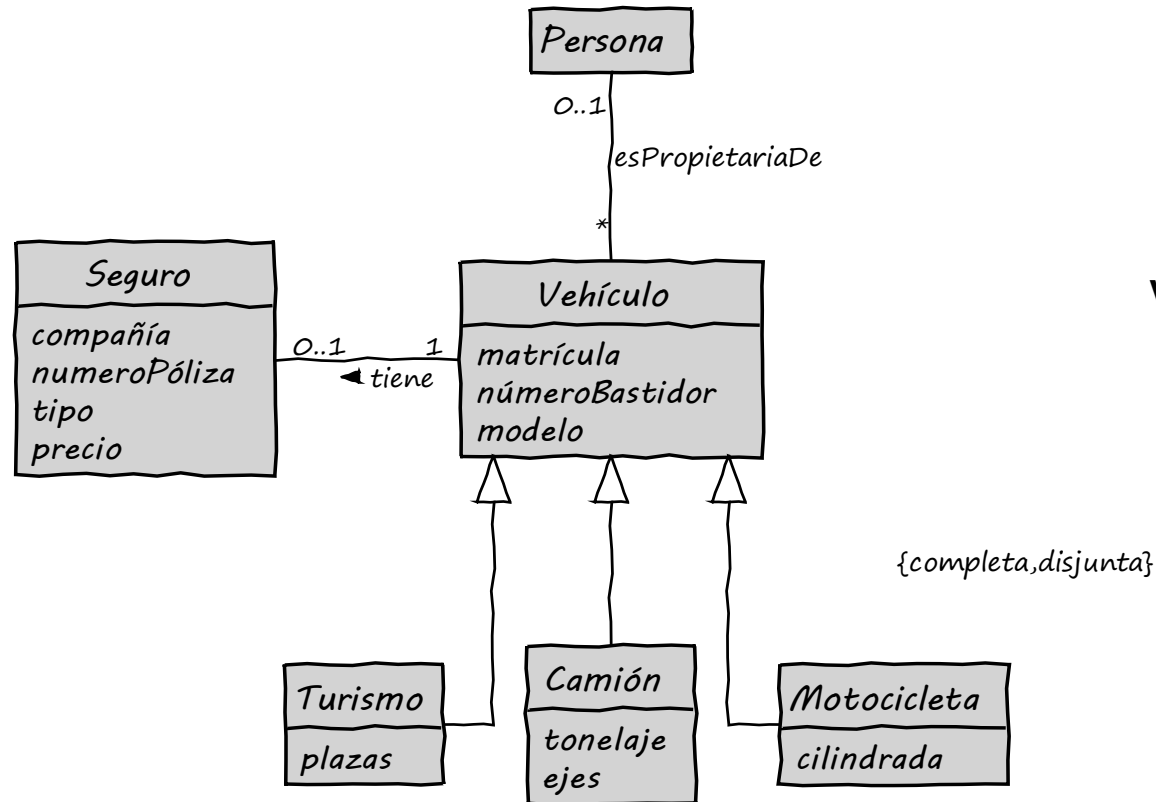
Indican si la clasificación es completa/incompleta y disjunta/solapada.

{restricciones}

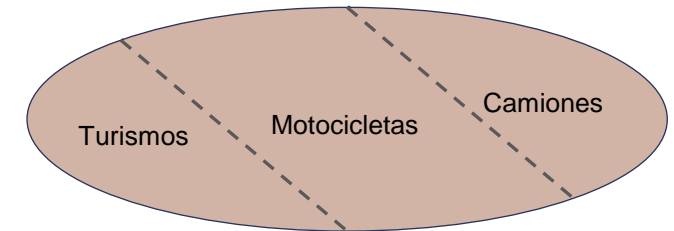
- **Clasificación disjunta/solapada**
 - {disjunta}: las instancias de la superclase pueden ser instancias de **una sola** subclase.
 - {solapada}: las instancias de la superclase pueden ser instancias de **una o más** subclases.

Notación UML: Diagramas de clases

Ejemplo de generalización II

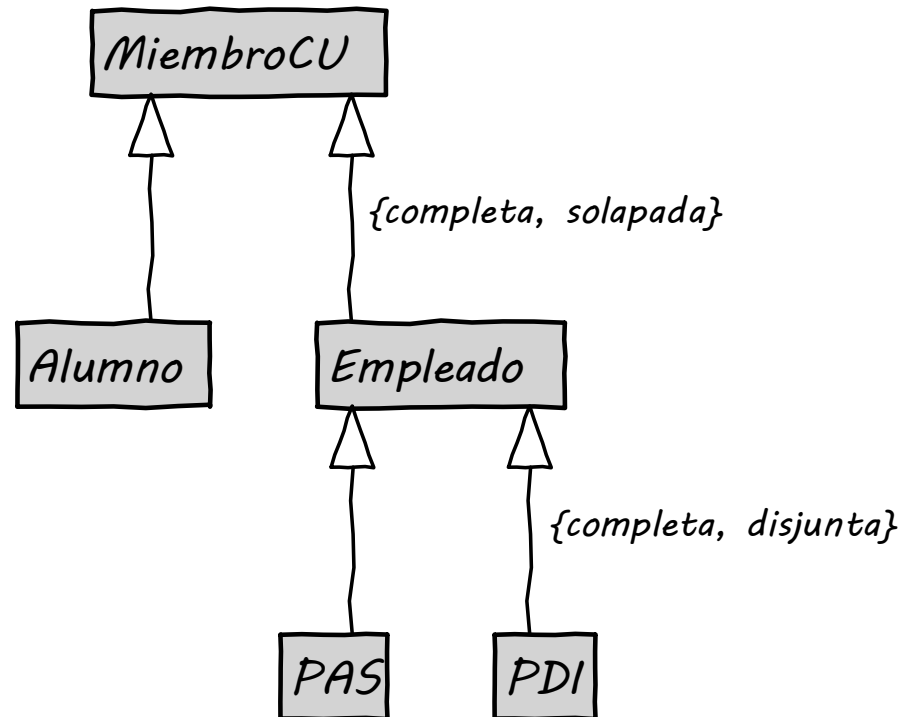


Vehículos



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

Notación para clasificación UML 2

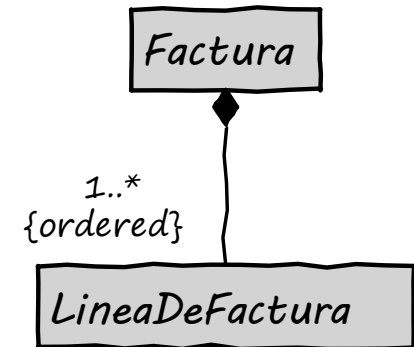


Composición

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.
- La eliminación del todo implica la eliminación de todas sus partes.
- Es una relación transitiva y antisimétrica.
- Puede ser recursiva.

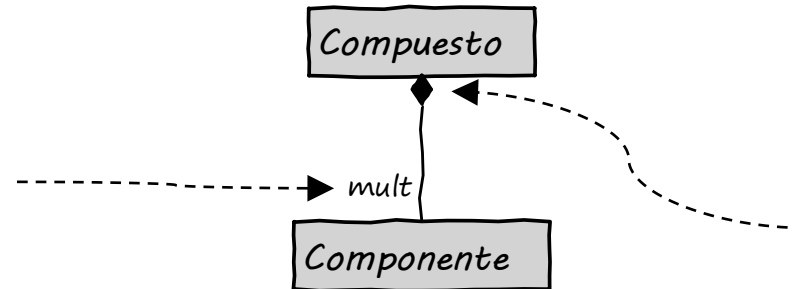
Ejemplo de composición



Componente

La **multiplicidad** del componente puede ser variable, por tanto, hay que indicarla.

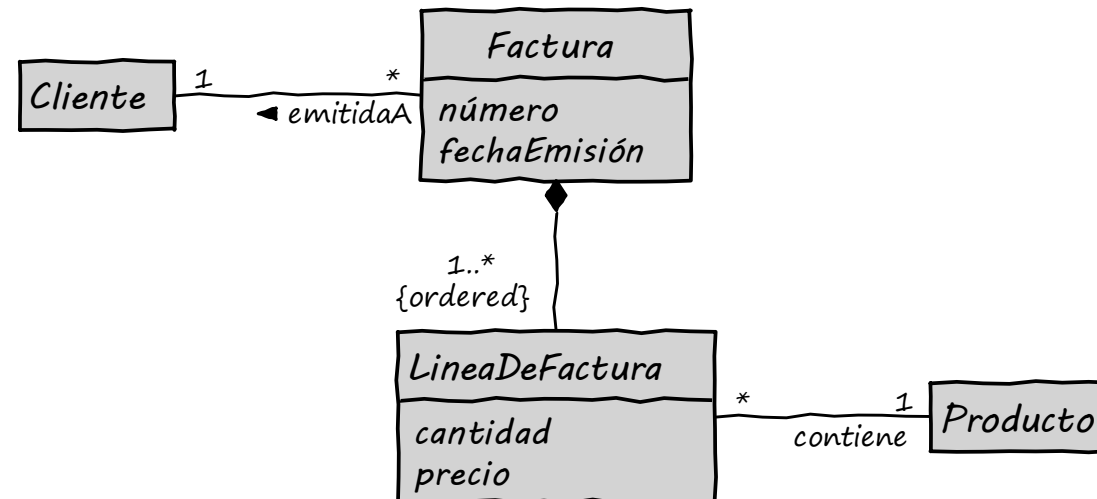
Notación composición



Compuesto

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

Ejemplo Factura

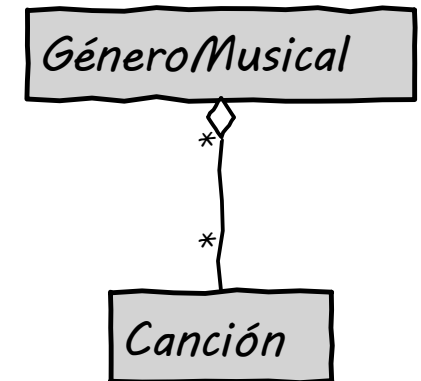


Agregación

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.

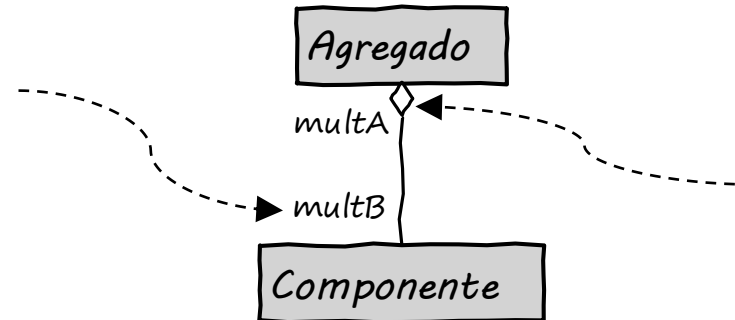
Ejemplo de agregación



Componente

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

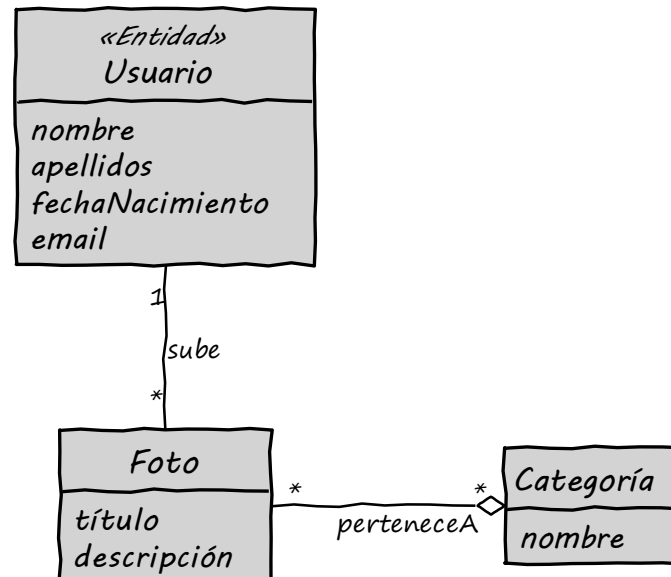
Notación agregación



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).

Ejemplo fotos



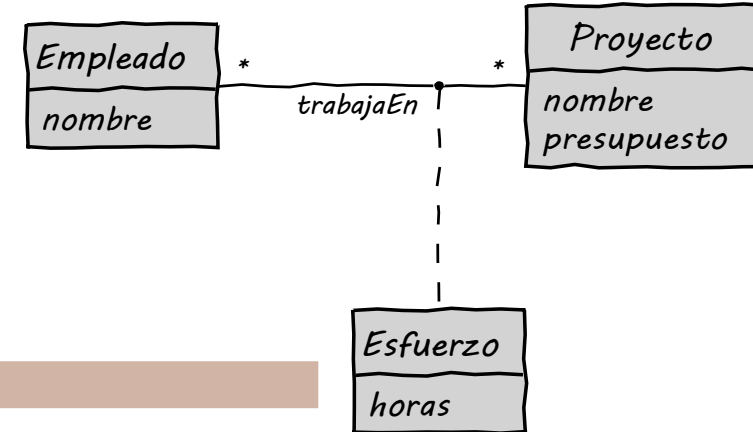
Clases Asociación

A veces es necesario añadir cierta información a las asociaciones, convirtiéndolas en clases.

Ejemplo clase asociación 1



Ejemplo clase asociación 2

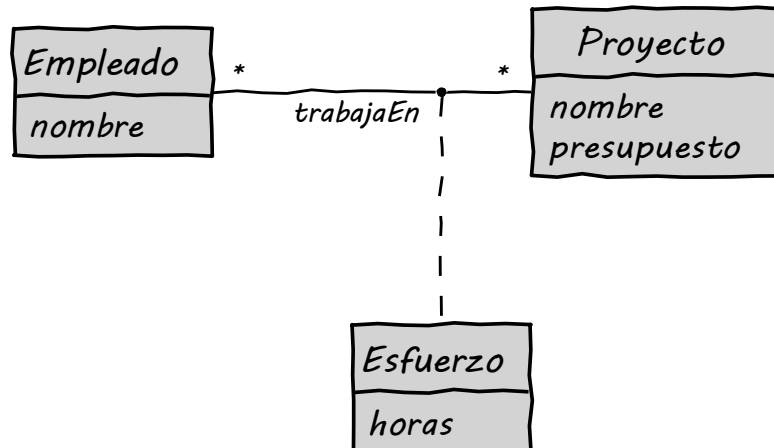


¿Cuántas horas trabaja cada empleado en cada proyecto?

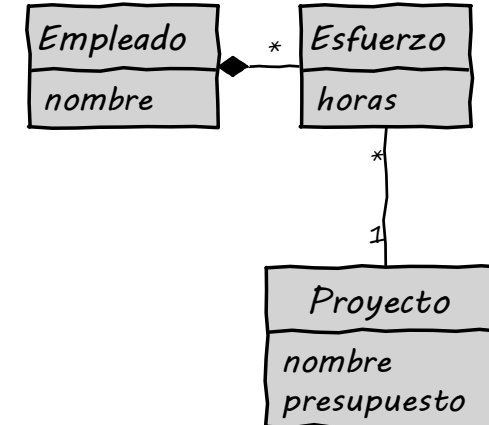
Clases Asociación

- También es posible modelarlas como una clase componente* de las clases participantes.
- Ambos modelos son equivalentes.

Ejemplo clase asociación 2



Ejemplo clase asociación con composición



*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.

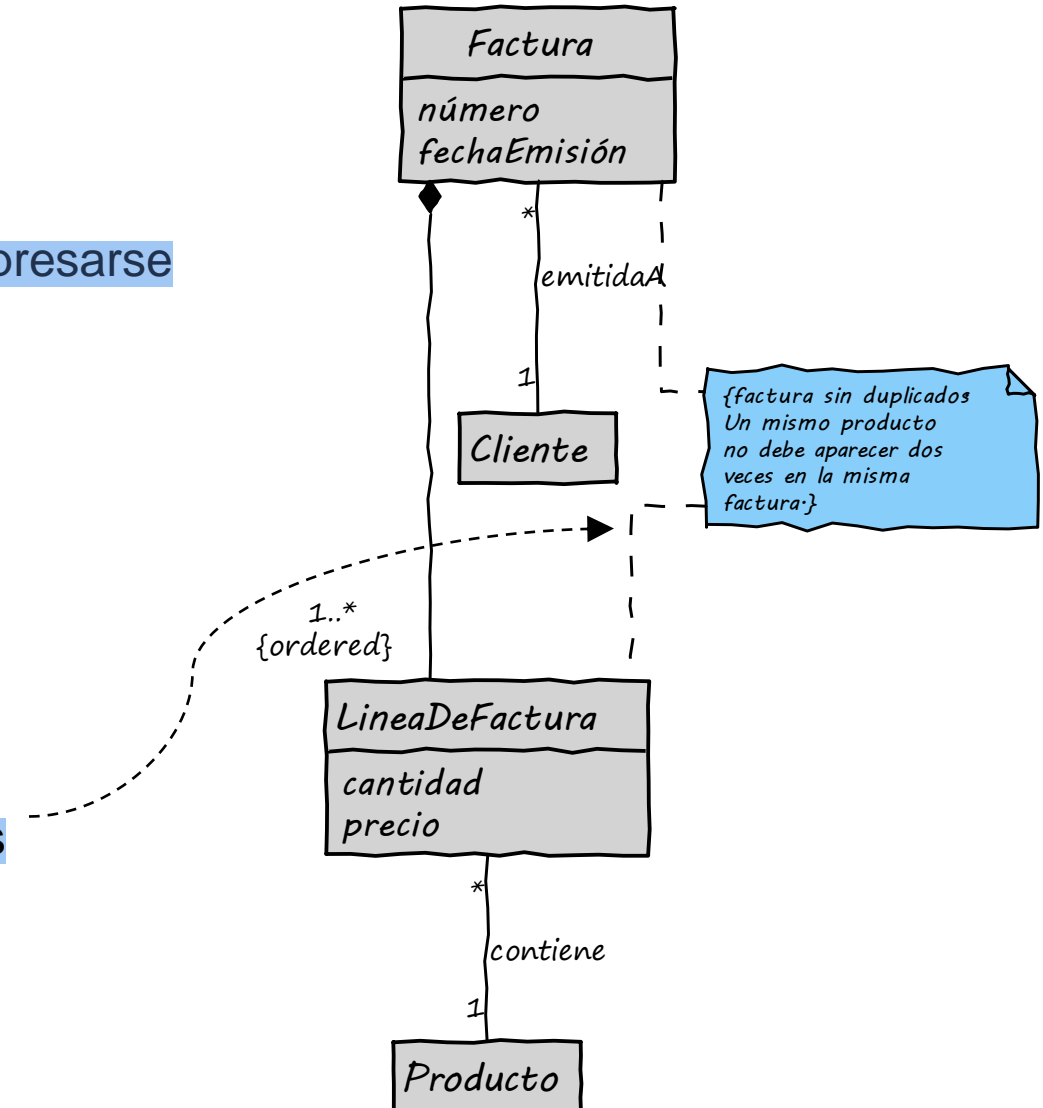
Restricciones

Permiten añadir información al modelo que no puede expresarse de otra forma.

Notación

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

Ejemplo de restricción



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



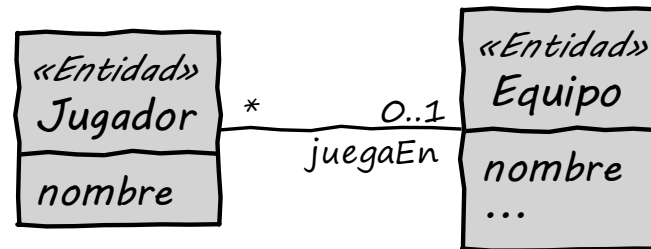
Objeto

- Cada ocurrencia o **instancia** de una clase.

Enlaces

- Cada ocurrencia o **instancia** de una asociación.

Equipos de fútbol· Modelo



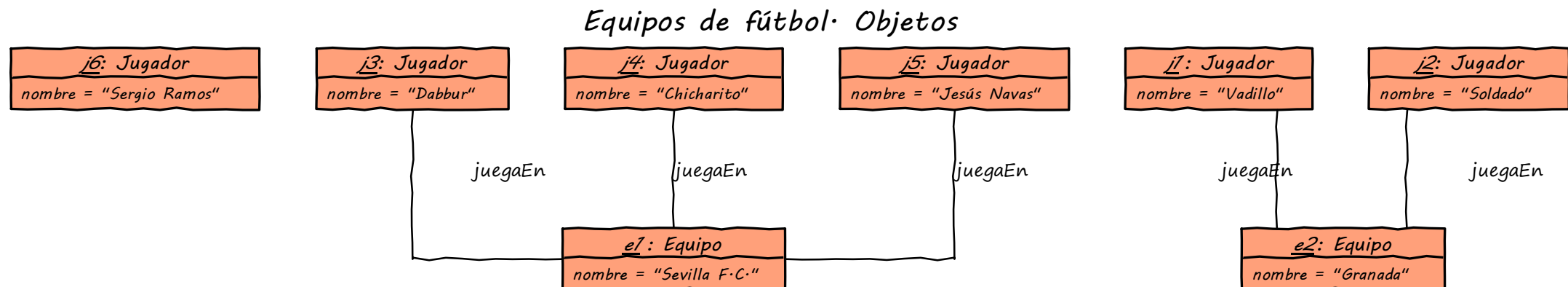
Notación UML: Diagramas de objetos

Objeto

- Cada ocurrencia o **instancia** de una clase.

Enlaces

- Cada ocurrencia o **instancia** de una asociación.

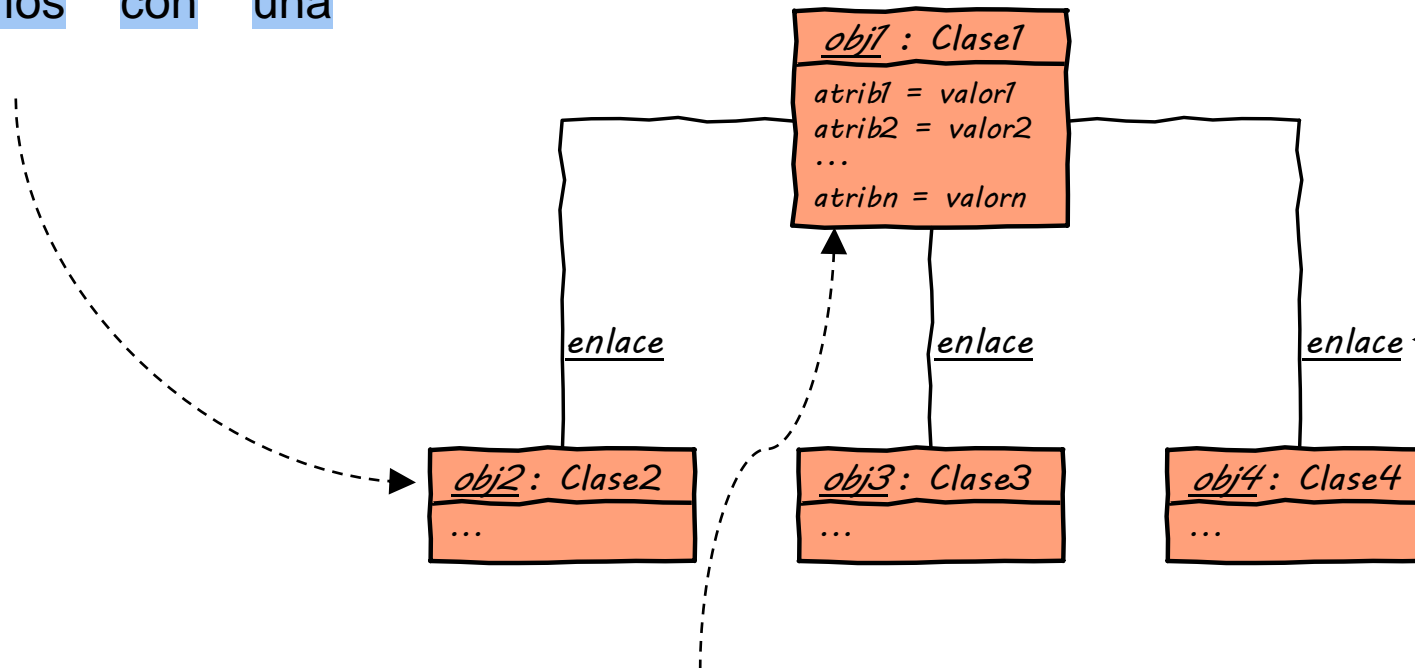


Notación UML: Diagramas de objetos

Nombre y clase del objeto

Deben estar subrayados para no confundirlos con una clase.

Notación para objetos y enlaces UML



Enlaces

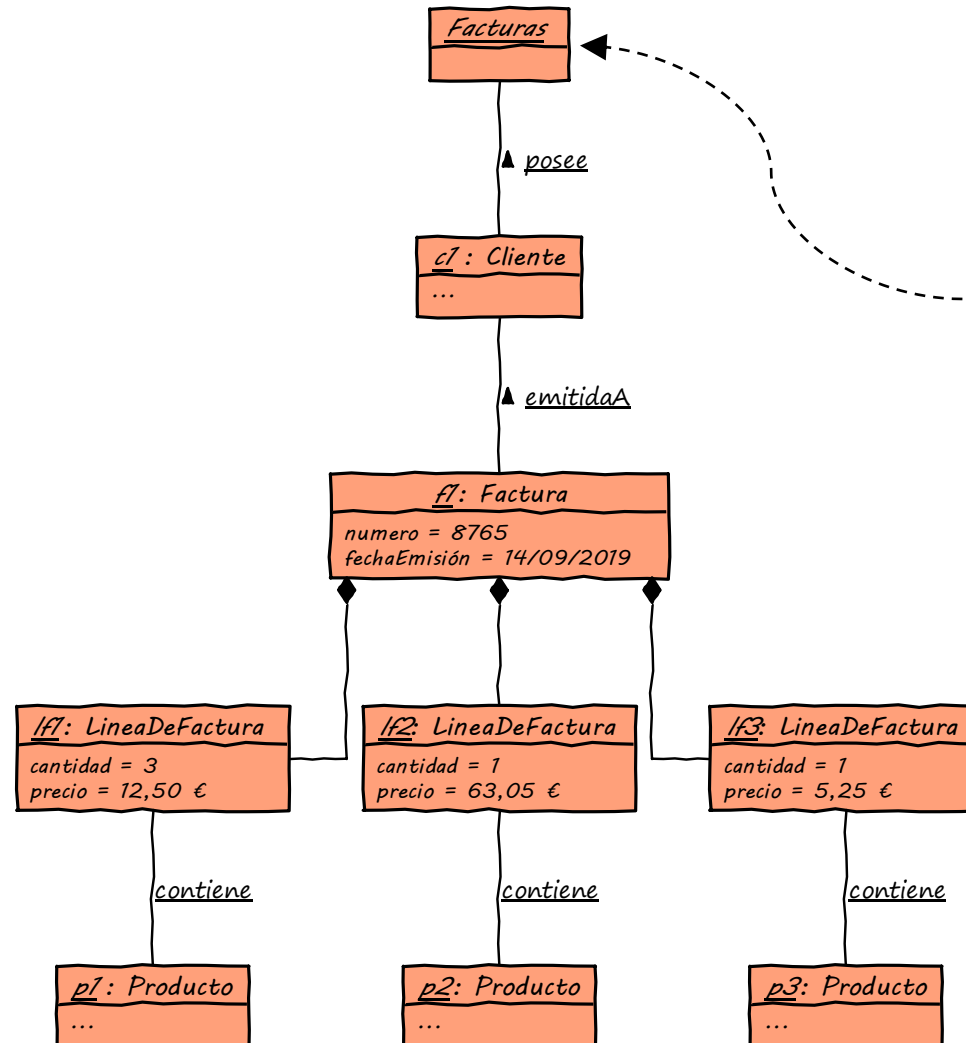
Se identifican mediante el nombre de la asociación subrayado.

Nombre y valores de atributos

Opcionales, muestran los valores de los atributos del objeto.

Notación UML: Diagramas de objetos

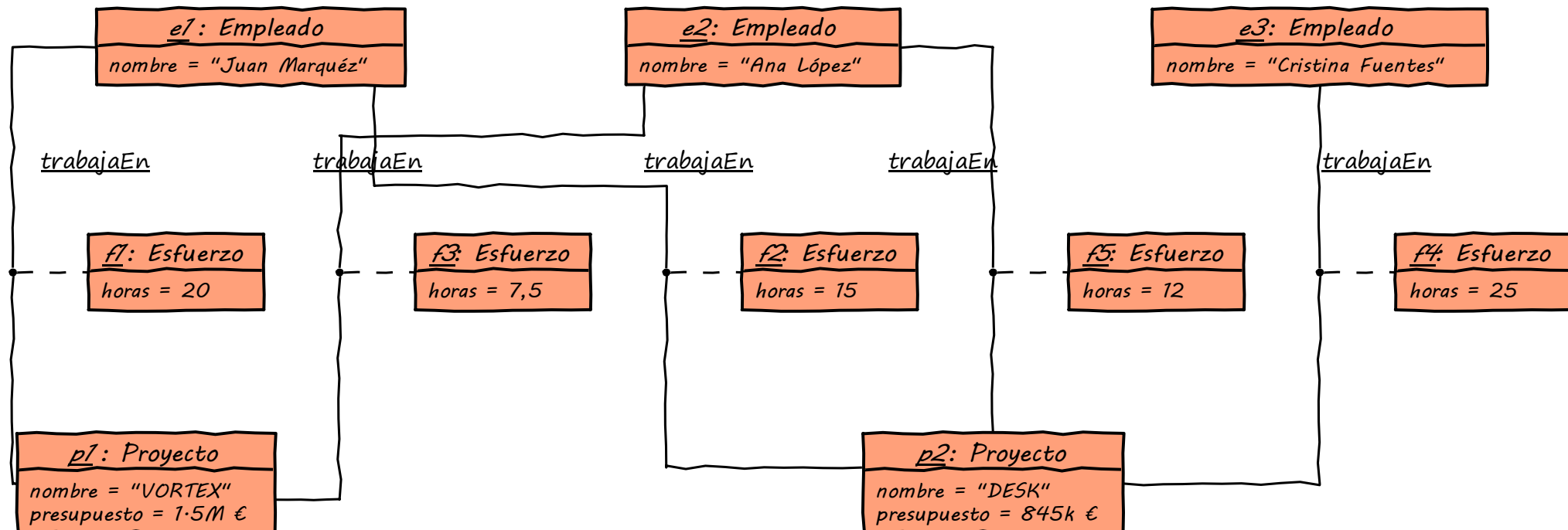
Notación para objetos y enlaces UML II



Notación multiobjeto
Indica múltiples objetos sin identificarlos individualmente.

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.

Diagrama de objetos con clases asociación



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.

La siguiente matriz de trazabilidad permite comprobar fácilmente si todos los requisitos de información y las reglas de negocio están expresados en el modelo conceptual resultante.

| | Persona | Turismo | esPropietario | Camión | ... | Seguro |
|------|---------|---------|---------------|--------|-----|--------|
| RI-1 | | X | X | | | |
| RI-2 | X | | | | | |
| RI-3 | | | | X | | |
| ... | | | | | | |
| RI-n | | X | | | | X |
| RN-1 | X | | | | | |
| ... | | | | | | |

Filas: requisitos de información y reglas de negocio. Columnas: clases, asociaciones y restricciones.

Existen muchas herramientas para crear modelos UML. Para esta presentación:

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

Tema 5: Introducción al modelo conceptual

Introducción a la Ingeniería del Software y los Sistemas de Información I
Ingeniería Informática – Tecnologías Informáticas
Departamento de Lenguajes y Sistemas Informáticos

