

Diagramas de Clases

Universidad Privada de Santa Cruz de la Sierra

José Antonio Benavente

08-10-2023

Introducción

- Son los diagramas más **comunes** en el modelado de sistemas orientados a objetos.
- Muestra un conjunto de clases, interfaces, colaboraciones y sus relaciones.
- Se usan para modelar la vista de diseño estático de un sistema.
- Son importantes no sólo para visualización, especificación y documentación de modelos estructurales, sino también para **construir sistemas**.

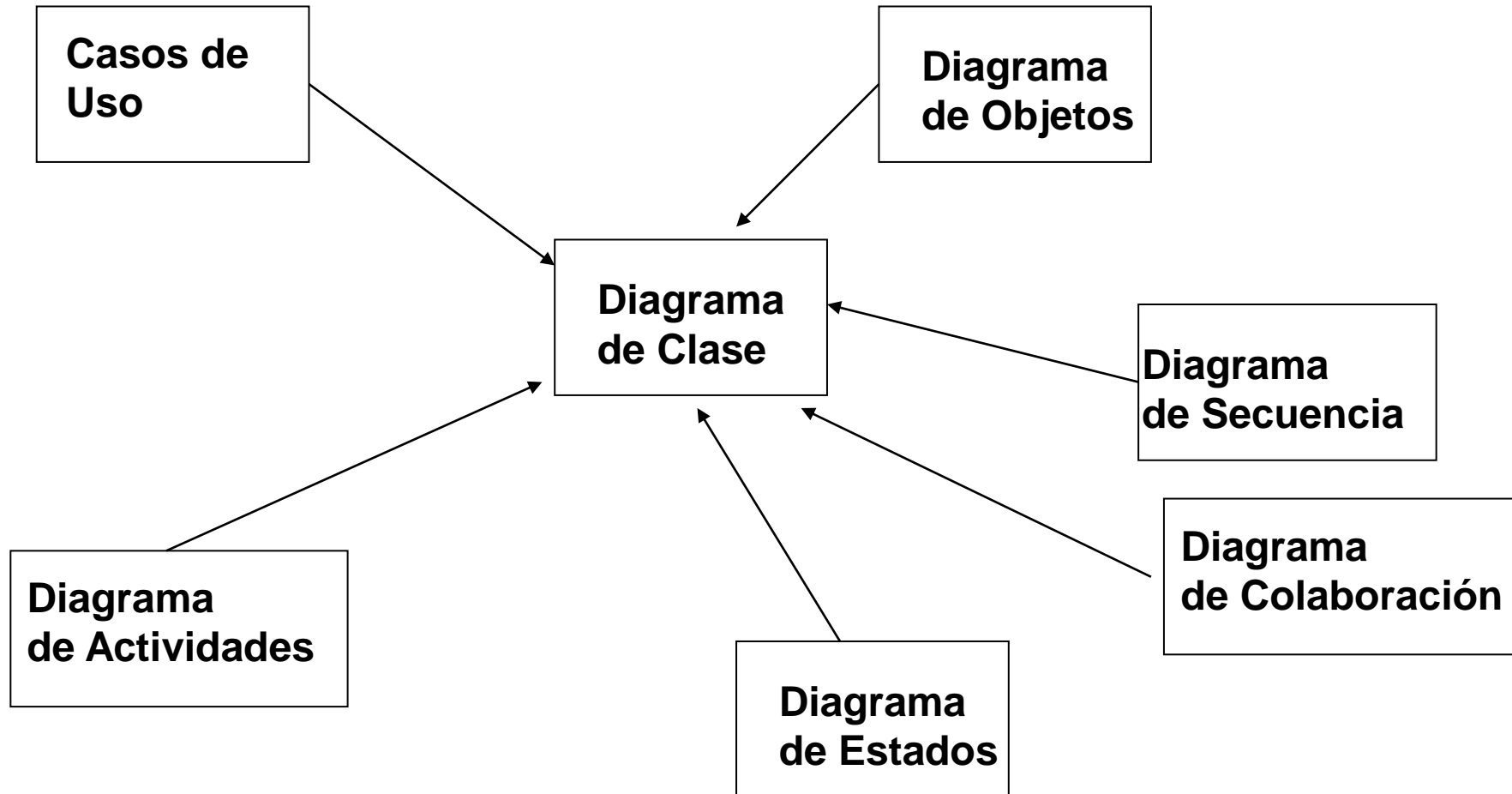
Introducción

- Un diagrama de clase comparte las mismas propiedades comunes como lo hacen todos los otros diagramas – un nombre y contenido gráfico.
- Los **diagramas de clase** contienen lo siguiente: clases, interfaces, colaboraciones, y relaciones de dependencia, generalización y asociación.
- El **diseño estático** de la vista de un sistema soporta principalmente los requerimientos funcionales de un sistema y los servicios que el sistema debe proporcionar a sus usuarios finales.

Introducción

- **Modelar el vocabulario de un sistema.** Involucra decidir sobre las abstracciones (**clases**) que son parte del sistema y las que están fuera de sus fronteras.
- **Modelar colaboraciones simples.** Una colaboración es una sociedad de clases, interfaces, y otros elementos que trabajan juntos para proporcionar algún comportamiento cooperativo que es mayor que la suma de todos los elementos.
- **Modelar un esquema de base de datos lógico.** Es como el anteproyecto para un diseño conceptual de una base de datos.

Los diagramas están vinculados al Diagrama de Clases



Modelo conceptual

El **Modelo conceptual** es una representación visual, mediante un determinado lenguaje, de los conceptos u objetos del mundo real de un dominio de interés.

El modelo conceptual, en general, **muestra:**

- Un conjunto de conceptos. (**clases**)
- Relaciones entre conceptos.
- Atributos de los conceptos.

El modelo conceptual **no muestra:**

- Componentes u objetos software.
- Responsabilidades de estos.

El **Modelo conceptual** se va representar usando como herramienta los **diagramas de clases de UML**.

Diagrama de clases de UML

Los **diagrama de clases de UML** permiten representar los, modelos concretamente:

- Modelo del análisis: **Modelo conceptual**.
- Modelo del diseño: **Diagrama de clases del diseño**.

Un Diagrama de clases de UML **puede** contener lo siguiente:

- Elementos estructurales: **clases e interfaces**.
- Relaciones entre esos elementos: **dependencia, generalización, asociación y realización**.
- Notas y estereotipos.

Diagrama de clases de UML: La Clase

- Una **clase** es una descripción de un conjunto de objetos que comparten: atributos, operaciones, relaciones y semántica”.
- Una clase define los **conceptos** que forman parte del dominio del problema y/o de la solución.

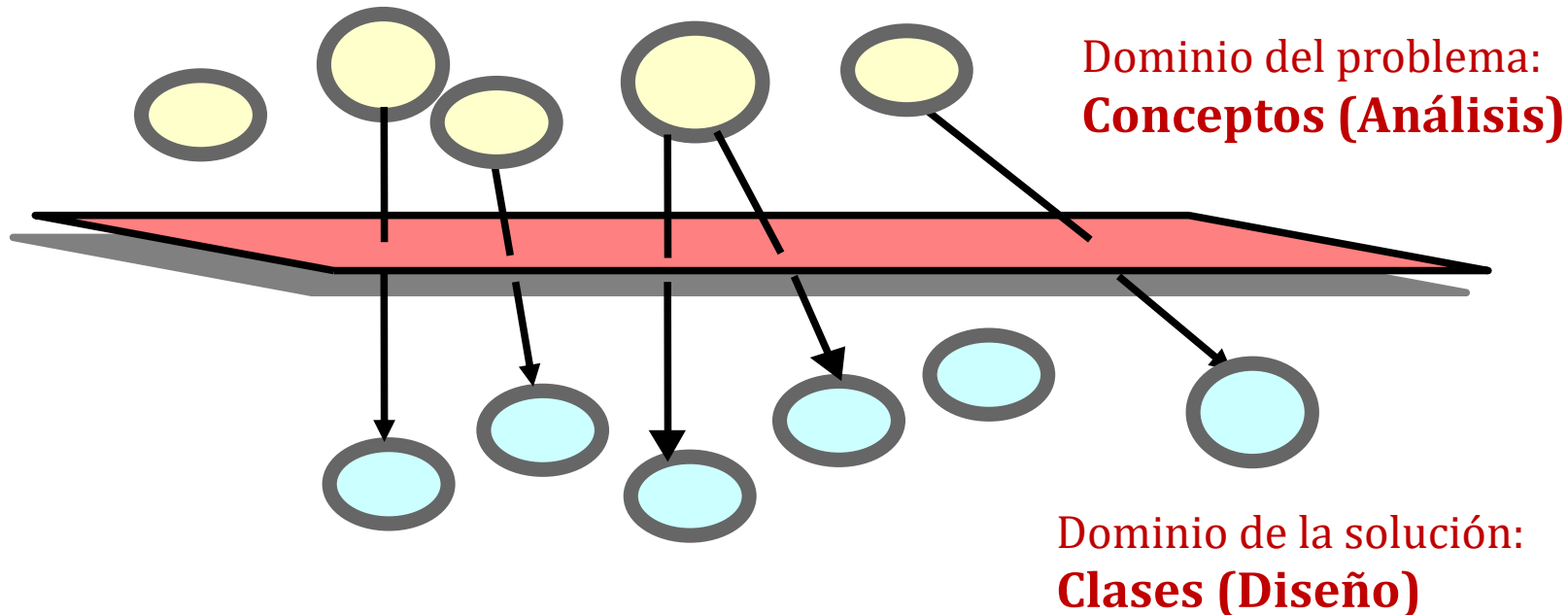
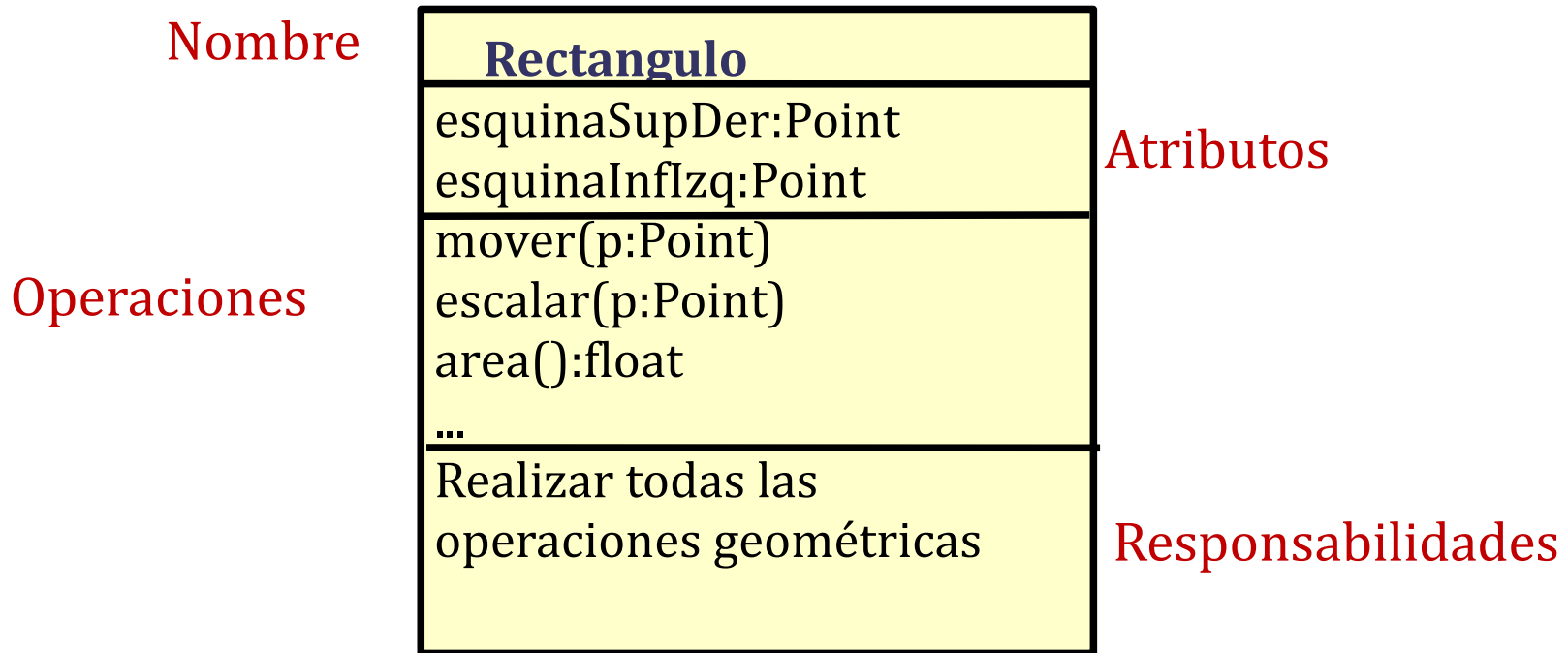


Diagrama de clases de UML: La Clase



Nombre: Expresión nominal extraída del vocabulario del **problema** a modelar. Debe comenzar con mayúscula.

Atributo: Propiedad del elemento que se está modelando.

[visibilidad] **nombre** [multiplicidad] [:tipo] [=valor inicial]

Operación: Es la abstracción de un servicio que puede prestar ese objeto

[visibilidad] **nombre** [(lista parámetros)] [:tipo retorno]

Responsabilidades: Obligación de esa clase con las demás clases.

Nombres

- Cada clase debe tener un nombre que la distinga de otras clases.
- Un nombre es una cadena de texto.
- Un nombre solo se conoce como nombre simple; una ruta nombre es el nombre de clase precedido del nombre del paquete en que vive la clase.

Sensor de temperatura	Reglas de negocio::Agente
Nombre simple	Ruta nombre

Atributos

- Un atributo es una propiedad de una clase que describe un rango de valores que la propiedad puede conservar.
- Una clase puede tener cualquier número de atributos o carecer de ellos.
- Un atributo representa alguna propiedad de la cosa que se esta modelando.
- Por ejemplo cada pared tiene: altura, ancho y espesor.
- Al atributo se le puede especificar su tipo y posiblemente un valor por omisión.

Pared
altura: Float
ancho:Float
estaPintada:Boolean=true

Atributos

- Son descripciones de características, se usan para modelar información asociada con una entidad, sintaxis:

Nombre_atributo[multiplicidad]:Tipo = Valor_inicial

- La multiplicidad es opcional e indica el número de atributos por instancia de la clase.

Operaciones

- Una operación es la implementación de un servicio que puede ser solicitado desde cualquier objeto de la clase para afectar el comportamiento.
- Una clase puede tener cualquier número de operaciones o carecer de ellas.
- Un nombre de operación es un verbo o frase verbal que representa algún comportamiento (acción).
- En una operación se puede establecer su firma, abarcando el nombre, tipo y valor de omisión de todos los parámetros y tipo de retorno.

SensorTemperatura
reiniciar()
ponerAlarma(t:Temperatura)
valor():Temperatura

Operaciones

- Son descripciones del comportamiento, se usan para modelar los servicios u operaciones asociados con una entidad, esto es, lo que una entidad puede hacer, sintaxis:

Nombre_operación[parámetros:tipo]:Valor_retorno:tipo

Interfaces

- Son clases que definen un **juego de operaciones** externas accesibles pero sin métodos.
- Se usan para modelar una serie de operaciones que definen un servicio que puede ser ofrecido por diferentes clases.
- Se representan como clases pero con el estereotipo <<interface>>.
- Solo contienen **operaciones** públicas
- No especifican estructura (no tiene atributos)

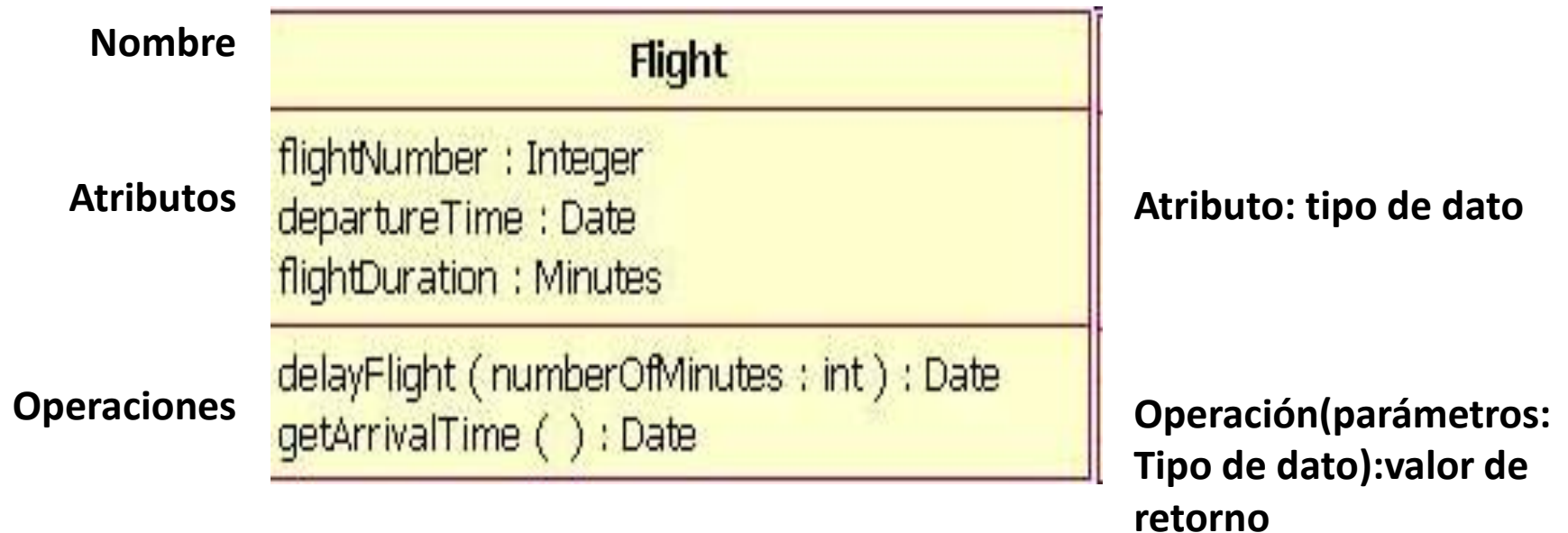
Modelando Clases

La representación de una clase es un rectángulo con 3 divisiones:

- El del **nombre** define la clase, (un tipo de objeto).
- El de los **atributos** contiene la definición de los datos.
- El de las **operaciones** contiene la definición de cada comportamiento soportado por este tipo de objeto.

Modelado de Clases

- La siguiente figura muestra un vuelo de una aerolínea modelado como una clase UML.



Modelado de Atributos

- Un atributo describe una pieza de información que un objeto tiene o conoce de sí mismo.
- Para poder usar esta información se debe asignar un nombre y especificar el tipo de dato.
- El tipo de dato puede ser primitivo o tipo de dato abstracto (definido)
- Cada atributo puede tener reglas que limiten los valores asignados a éste. Se puede usar un valor de default para protegerlo.

Visibilidad de un Atributo

- La definición de un atributo debe especificar que otros objetos los pueden ver.
- La visibilidad puede ser:
 - **Public (+)** permite el acceso a objetos de las otras clases.
 - **Private (-)** limita el acceso a la clase, solo operaciones de la clase tienen acceso.
 - **Protected (#)** permite el acceso a subclases. En el caso de generalización (herencia), las subclases deben tener acceso a los atributos y operaciones de la superclase, sino no pueden heredar.
 - **Package (~)** permite el acceso a los otros objetos en el mismo paquete.

Especificación de un atributo

Elemento	Ejemplo
Nombre del atributo	compañía
Tipo de dato	compañía: character
Valor de default (si hay)	compañía:character = espacios
Restricciones	compañía:character = espacios {1 a 30}
Caracteres	compañía:character = espacios{1 a 30 alfabéticos, espacios, puntuación, no especiales}
Visibilidad	Publico (#)

Modelando una Operación

- Los objetos tienen **comportamientos**, cosas que puedan hacer y que se les puedan dar a éstos.
- Las operaciones requieren un nombre, argumentos y a veces un valor de retorno.
- Las reglas de privacidad se aplican en la misma forma que para los atributos: Private, Public, Protected y Package.

Especificación de una Operación

Elemento	Ejemplo
Nombre	totalOrderAmount
Definir argumentos/ Parámetros, corresponden a una instancia de Order	totalOrderAmount (order: integer)
Definir el tipo de dato de retorno	totalOrderAmount (order: integer) : Dollar
Identificar y describir restricciones	totalOrderAmount (order: integer) : { El total es la suma de cada item (p.u. x cantidad) }
Visibilidad	+ totalOrderAmount (order: integer) : {El total es la suma

Diagramas de Clases

Relaciones entre clases

- Conexión semántica entre elementos del modelo.
- Tipos de Relaciones entre clases:
 - Asociación.
 - Agregación.
 - Composición.
 - Generalización / Especialización.
 - Dependencia.

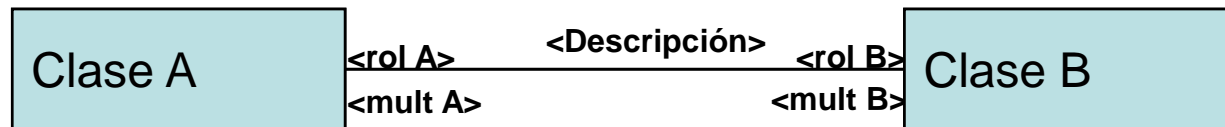
Diagrama de Clases: Asociaciones

- El propósito de la asociación puede expresarse en un nombre, verbo o frase que describa como los objetos de un tipo (clase) se relacionan con objetos de otro tipo (clase).
- Por ejemplo:
 - Una persona **tiene** un coche (1 -> 1..*)
 - Una persona **maneja** un coche (1 -> 1)
- Multiplicidad: cuantos objetos van a participar en la relación

Diagrama de Clases: Asociaciones

- **Asociación:**

- Relación o invocación significativa entre dos o más clases.



- Según notación UML, la asociación comprende:
 - *Descripción*, o nombre de la relación.
 - *Rol*: Responsabilidad de la clase en la relación.
 - *Multiplicidad*: Indica cuantos objetos pueden participar en la relación.
 - 0 ó más: *
 - 1 o más: **1..***
 - De 2 a 4: **2..4**
 - Sólo 7: **7**

Diagrama de Clases: Asociaciones

- Una **Asociación** es una relación estructural fuerte, que indica que los objetos de una clase forman parte del estado del objeto del otro extremo.

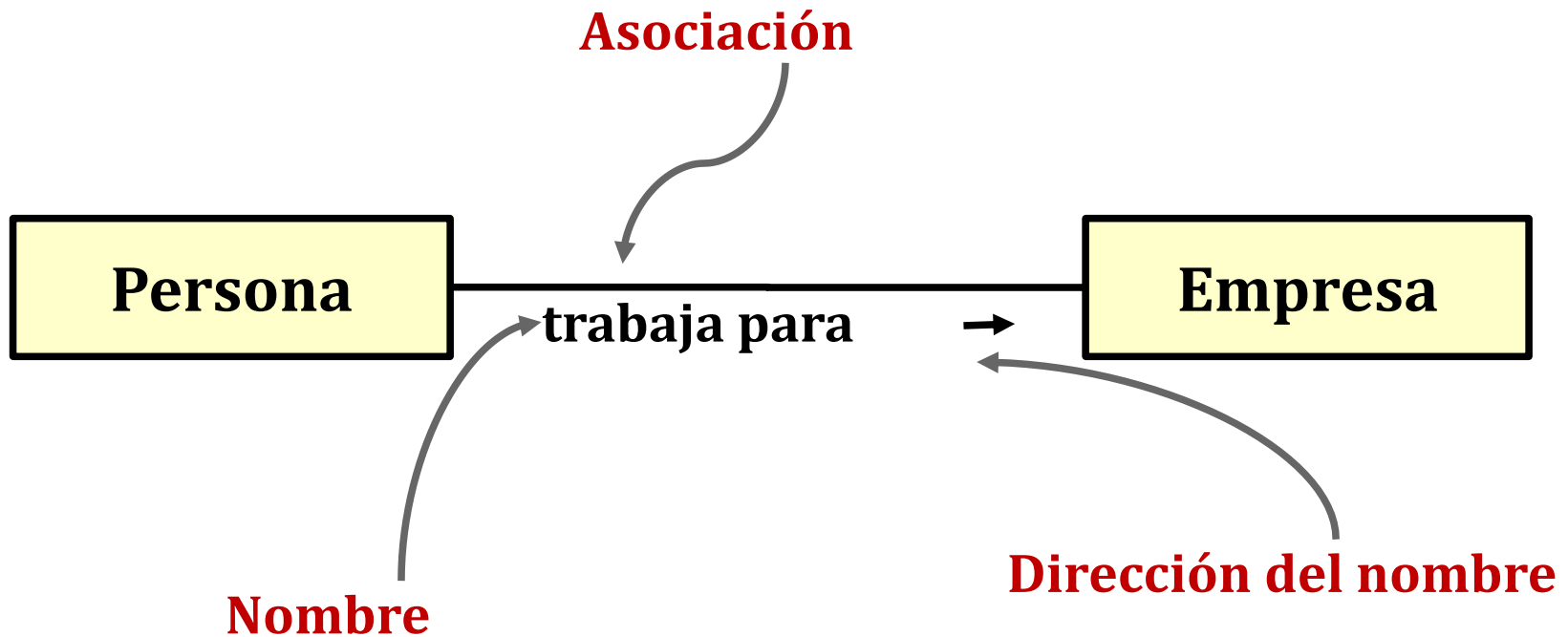


Diagrama de Clases: Asociaciones

La **Multiplicidad** indica **cuantos objetos** de un extremo de la asociación pueden conectarse **con un objeto** del otro extremo.

Sintaxis: **valorMinimo..valorMaximo** (por defecto es 1)

Ejemplos: **1** = Uno, **0..1** = Cero o Uno, **0..*** = Cero o muchos,

1..* = Uno o muchos,

0..1,3..4,6..* = Cualquier número excepto 2 y 5

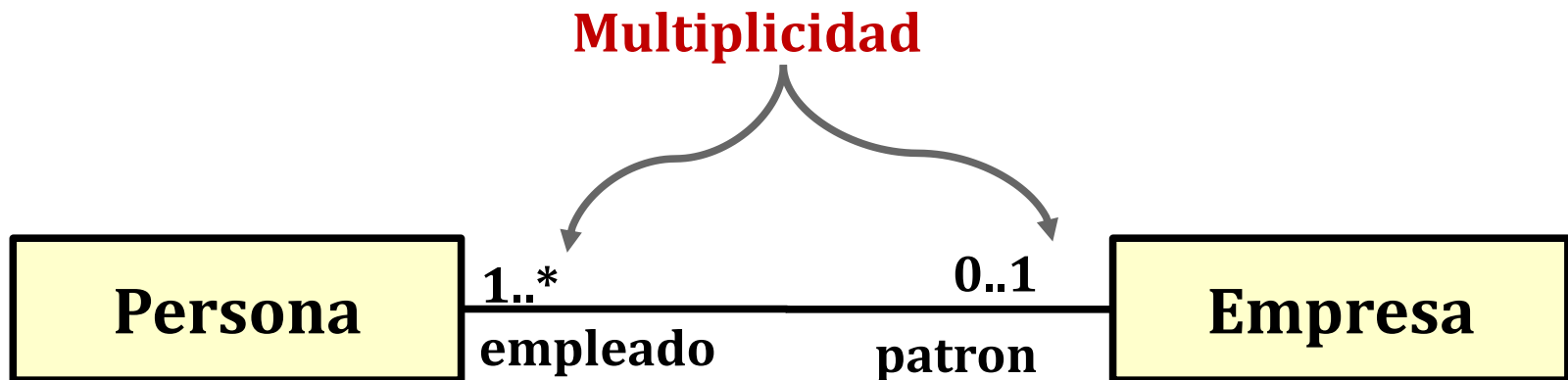


Diagrama de Clases: Asociaciones

- En una asociación se puede **navegar** desde el objeto de una clase hasta el objeto de la otra clase, y viceversa, si no se indica el sentido de navegación.
- Un **rol** es la cara que la clase de un extremo de la asociación presenta a la clase del otro extremo

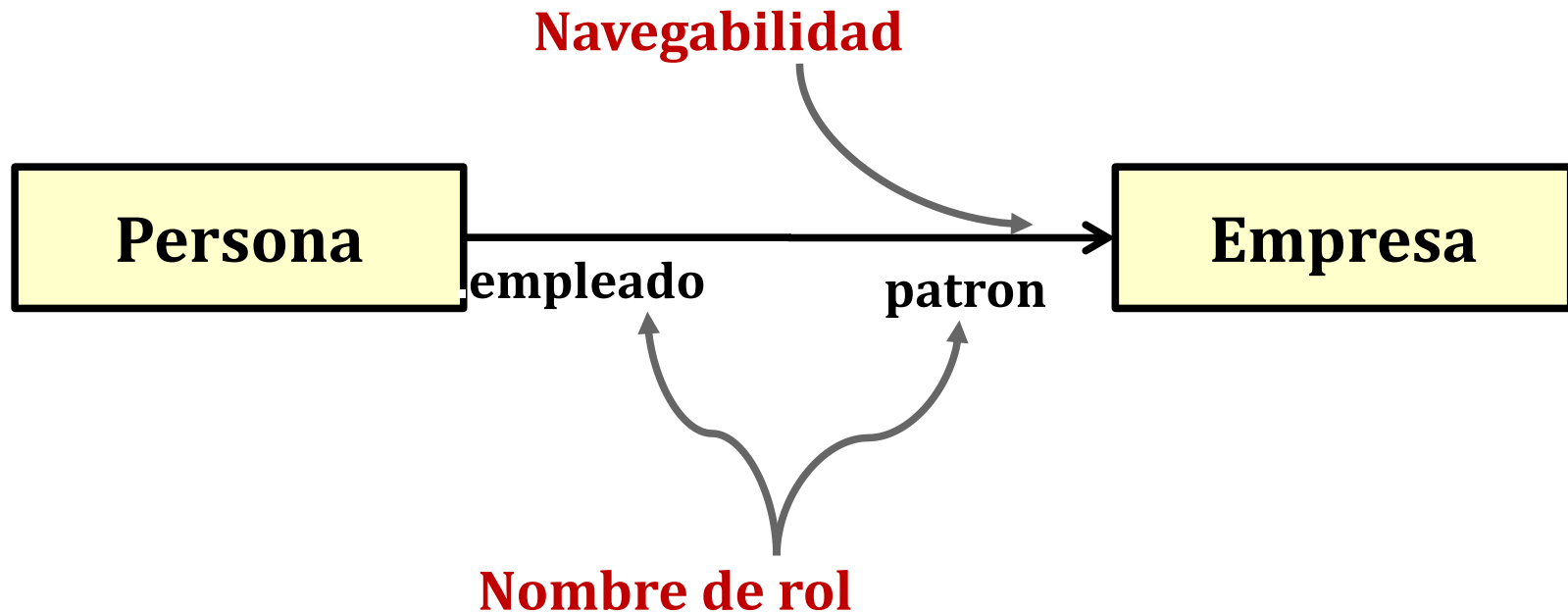
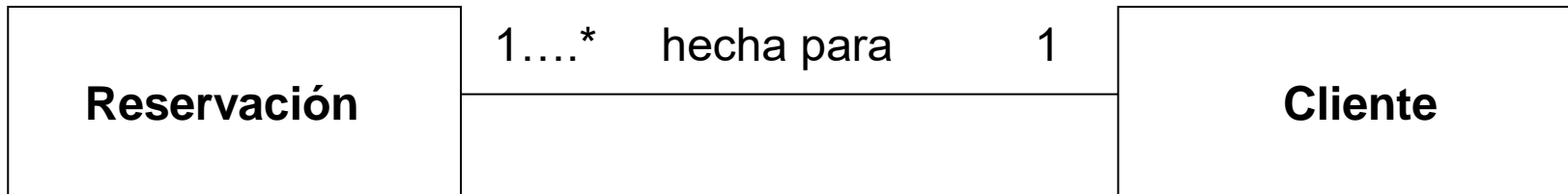


Diagrama de Clases: Asociaciones

- La **dirección** en las flechas de la asociación determinan en que dirección puede recorrerse una asociación en el momento de la ejecución.
- Una asociación **sin flechas** significa que se puede ir de un objeto a otro y viceversa.
- En el ejemplo: el tipo de flecha en la asociación implica que desde el objeto **Reservación** se puede recuperar (dirigirte hacia) el objeto **Cliente**.
- También implica que del objeto Cliente puedes recuperar el juego de reservaciones para ese cliente.

Diagrama de Clases: Asociaciones



Supongamos que los requerimientos para el sistema de reservaciones requieren que “desde una reservación, el sistema pueda recuperar el cuarto

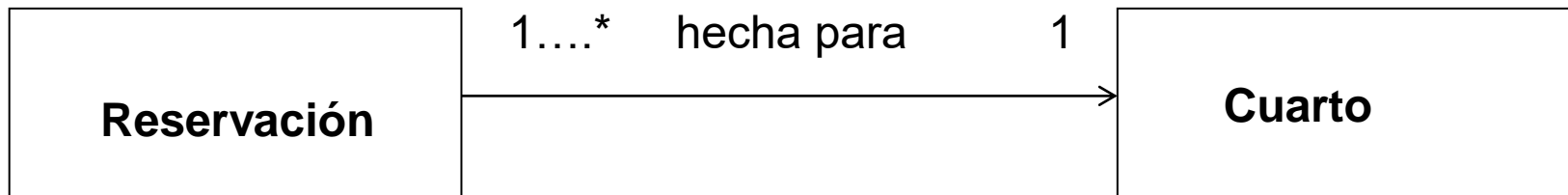


Diagrama de Clases: Asociaciones

- Cuando se modela una asociación entre clases, a veces es necesario **incluir otra clase que contiene información valiosa acerca de la relación.**
- Se representa como una clase normal solo que la línea que la une con la línea que conecta las asociaciones primarias es punteada.
- La figura muestra una clase asociación para el ejemplo de una empresa y persona.

Diagrama de Clases: Asociaciones

- Una **Clase asociación** se da cuando una asociación presenta propiedades, esas propiedades hay que modelarlas como una clase.

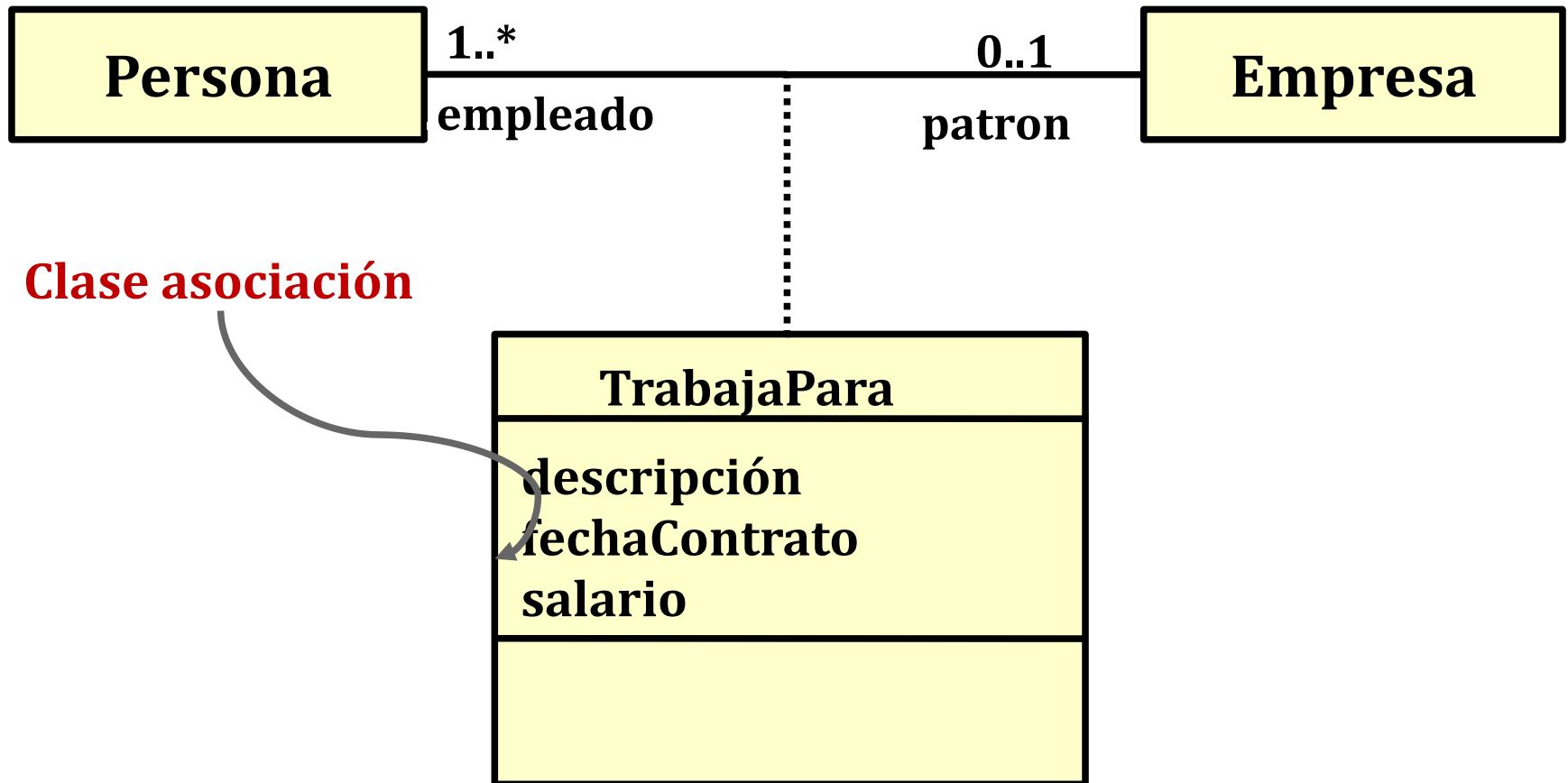


Diagrama de Clases: Asociaciones

Asociación:

- *Grado de la Asociación:* Se determina por el número de clases conectadas por la misma asociación.
- Las asociaciones pueden ser binarias ternarias o de mayor grado.
- Ejemplo de una *asociación ternaria*:

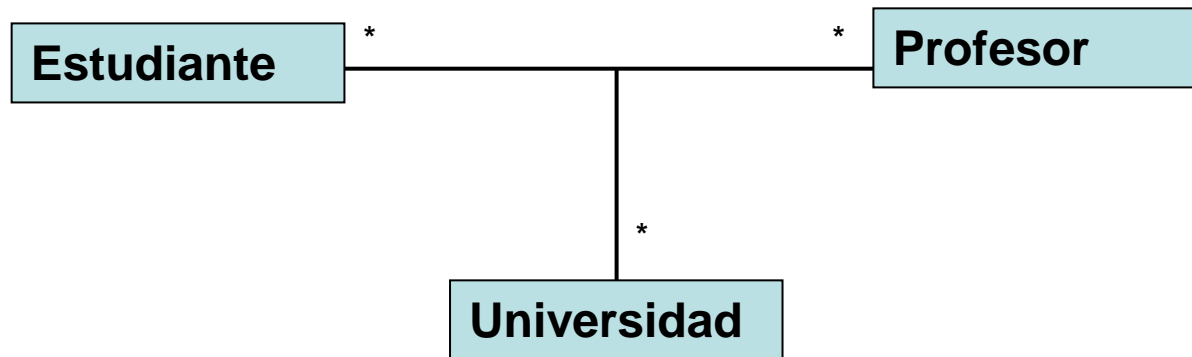


Diagrama de Clases: Asociaciones

Asociación:

- Las *asociaciones* pueden ser *reflexivas*, es decir pueden relacionar distintos objetos de una misma clase.

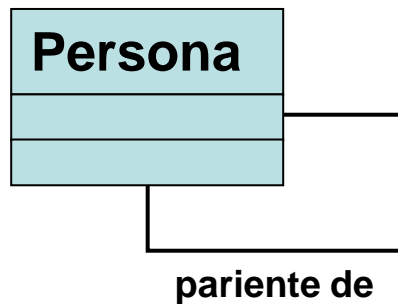


Diagrama de Clases: Asociaciones

Asociación Reflexiva

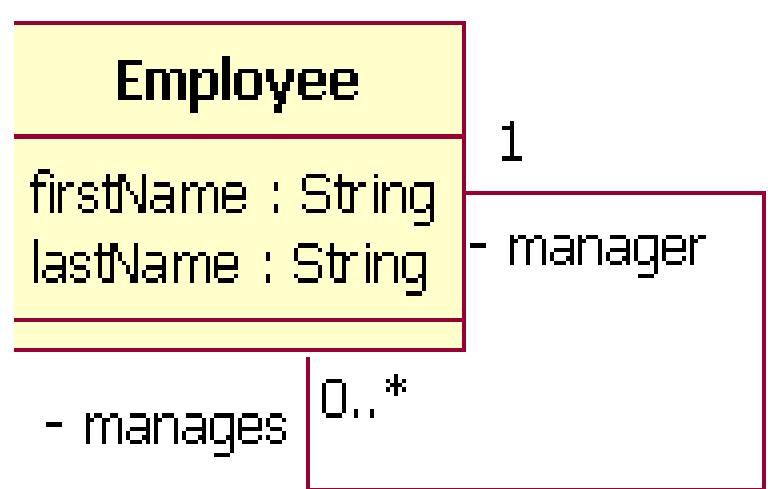
- Una clase puede asociarse con sí misma.
- Una clase Empleado puede relacionarse con sí misma a través del rol gerente/dirige.
- No significa que una instancia está relacionada consigo misma, sino que una instancia de la clase está relacionada con otra instancia de la misma clase.

Diagrama de Clases: Asociaciones

Asociación Reflexiva

- Supongamos que las personas que trabajan en una empresa se tienen registradas sus habilidades, esto significa que cualquier empleado puede tener cualesquiera habilidades.
- ¿Es necesario crear una clase asociación que contenga la información de ambas clases?
- Dibujar las entidades y su asociación.

Diagrama de Clases: Asociaciones



Una instancia de Employee puede ser el gerente de otras instancias de Employee.

Como el rol `manages` tiene una multiplicidad de 0..*, significa que puede no tener otros empleados a quien dirigir.

Una instancia de Employee tiene 1 sólo gerente ó un solo director.

Diagrama de Clases: Asociaciones

- Se puede incluir el **rol de las clases**, el siguiente ejemplo de los roles jugados por la clase Employee (de la asociación reflexiva), mostramos que la relación es entre un Employee jugando el papel de gerente y un Employee jugando el rol de miembro del equipo.

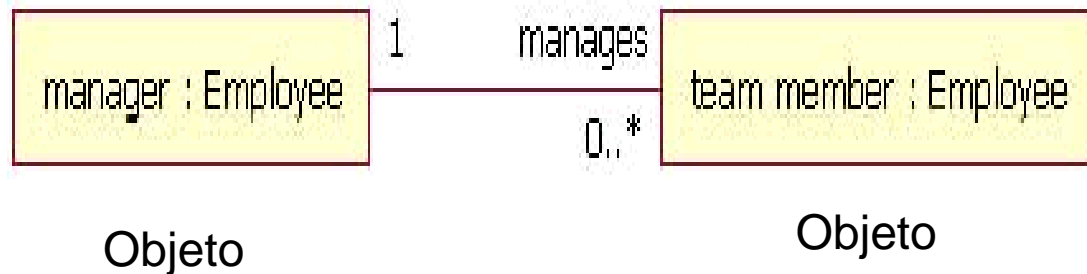


Diagrama de Clases: Asociaciones

- Un **cualificador** es un atributo de la clase en el lado opuesto de la asociación, que permite hacer una búsqueda en función a su valor.
- Ejemplo: “cliente usa **numOrden** para buscar una orden”.
- Un tipo de objeto usa el cualificador para **accesar** el otro tipo de objeto.

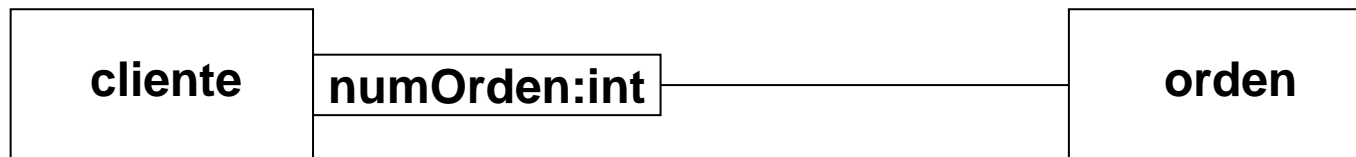


Diagrama de Clase: Agregación y Composición

- Cada agregación es un tipo de asociación.
- Cada composición es una forma de agregación.

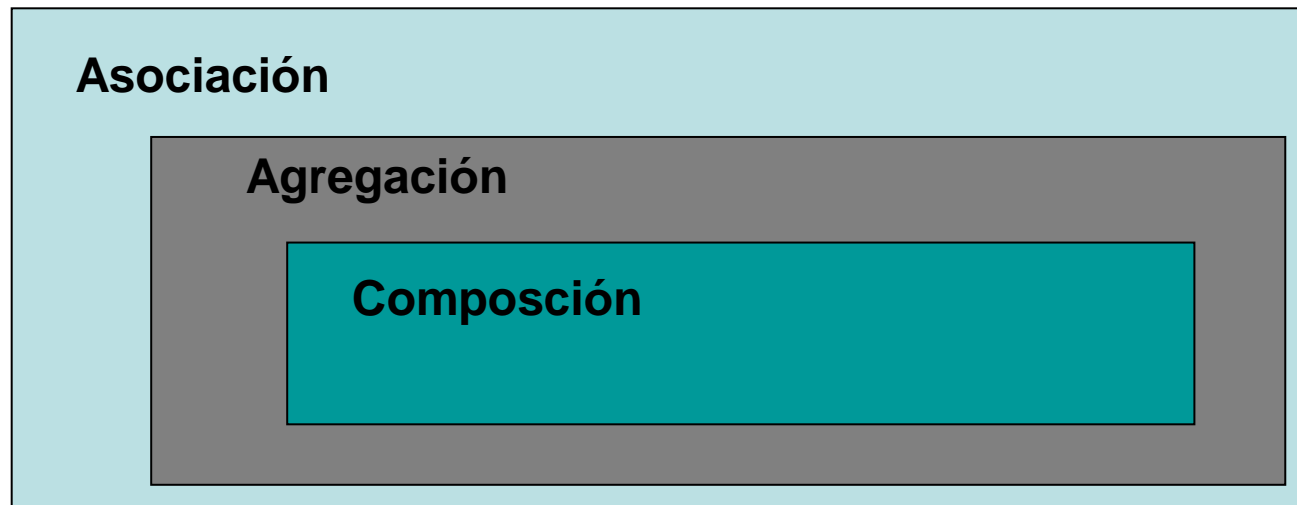


Diagrama de Clases: Agregación

- Es un tipo especial de asociación utilizado para modelar una relación “whole to its parts”.
- Por ejemplo, Coche es una entidad “**todo**” y Llanta es una **parte** del Coche.
- Una asociación con una agregación indica que una clase es parte de otra clase.
- En este tipo de asociación, **la clase hijo puede sobrevivir sin su clase padre.**

Diagrama de Clases: Agregación

Para representar una relación de agregación, se dibuja una línea sólida de la clase padre (total) a la clase hijo (parte), y con un diamante (rombo) en el lado de la clase padre.

Una llanta puede existir sin automóvil

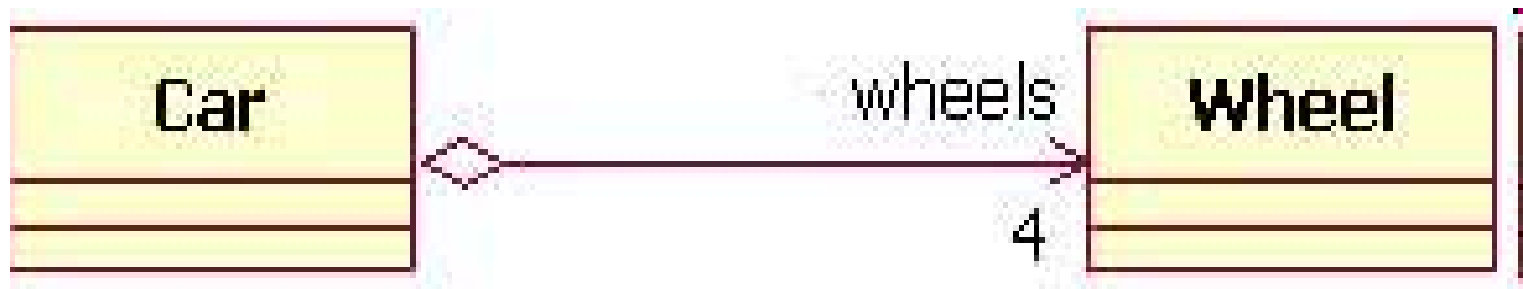


Diagrama de Clases: Agregación

Agregación: (“es parte de”, “contiene”)

- Asociación que especifica relación *Parte de* entre el agregado (Todo) y el componente (Parte).

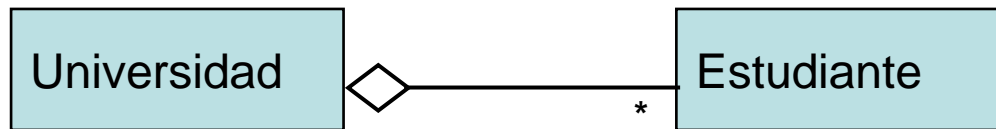


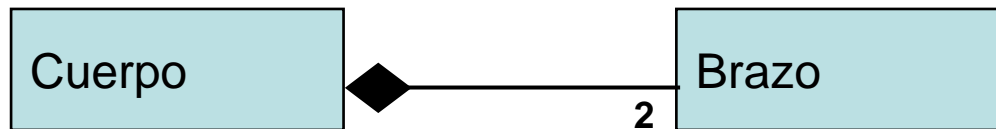
Diagrama de Clases: Composición

- En este caso el ciclo de vida de una instancia de la clase hijo **depende** del ciclo de vida de una instancia de la clase padre.
- A diferencia de la agregación básica, para representarla el diamante no es hueco.
- Una instancia de la clase Compañía debe tener al menos una en la clase Departamento.
- En este tipo de relaciones, si una la instancia Compañía se elimina, automáticamente la instancia Departamento también se elimina.
- Otra característica importante es que la clase hijo solo puede relacionarse con **una instancia** de la clase padre.

Diagrama de Clases: Composición

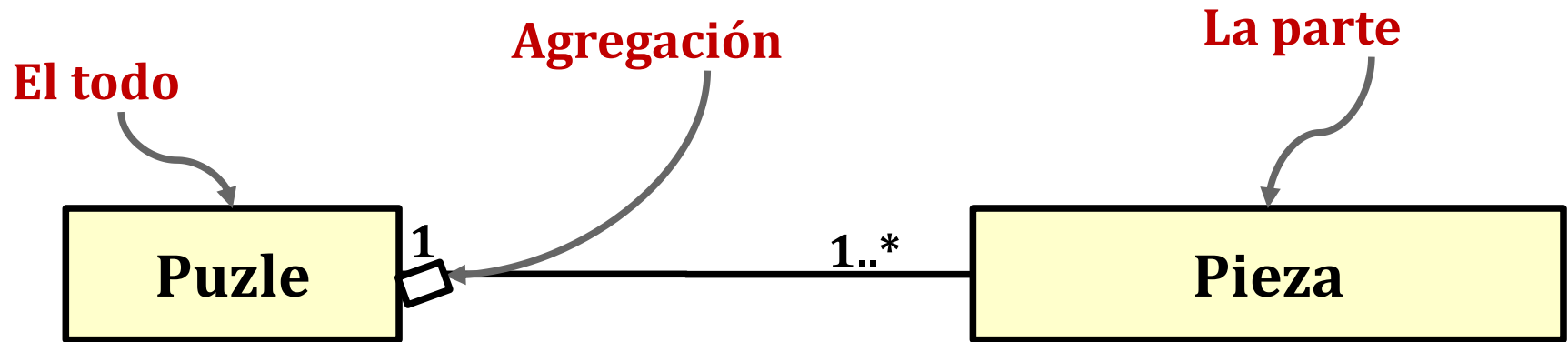
Composición: (“compuesto por”)

- Relación de agregación especial donde las partes no pueden existir sin que exista el objeto todo.



Diagramas de clases: Asociación/Composición

- Una **Agregación** (relación “parte-de”) es una asociación en la que una de las clases representa el “todo” y la/s otra/s la/s parte/s.



- Una **Composición** es una forma de agregación con una fuerte relación de pertenencia y vida de las partes con el todo.



Ejercicios Agregación y Composición

- Identificar en los diagramas de asociación, si existe agregación / composición.
- Anotar la multiplicidad.

1)Jugadores basketball y equipo basketball

2)Libro y capítulos del libro

3)Motor y automóvil

4)Líneas de un pedido (artículos) y el pedido

5)En una empresa se llevan a cabo proyectos, estos proyectos están formados por una ó más actividades y a su vez cada actividad tiene 1 ó más tareas específicas. Cada tarea es asignada a un empleado y los empleados pueden o no tener asignadas tareas.

Ejercicios Agregación y Composición

1) Jugadores basketball y equipo basketball

Tipo: Agregación

Multiplicidad: $* \rightarrow 1$

2) Libro y capítulos del libro

Tipo: Composición

Multiplicidad: $* \rightarrow 1$

3) Motor y automóvil

Tipo: Agregación

Multiplicidad: $1 \rightarrow 1$

Ejercicios Agregación y Composición

4) Líneas de un pedido (artículos) y el pedido

Tipo: Agregación

Multiplicidad: $* \rightarrow 1$

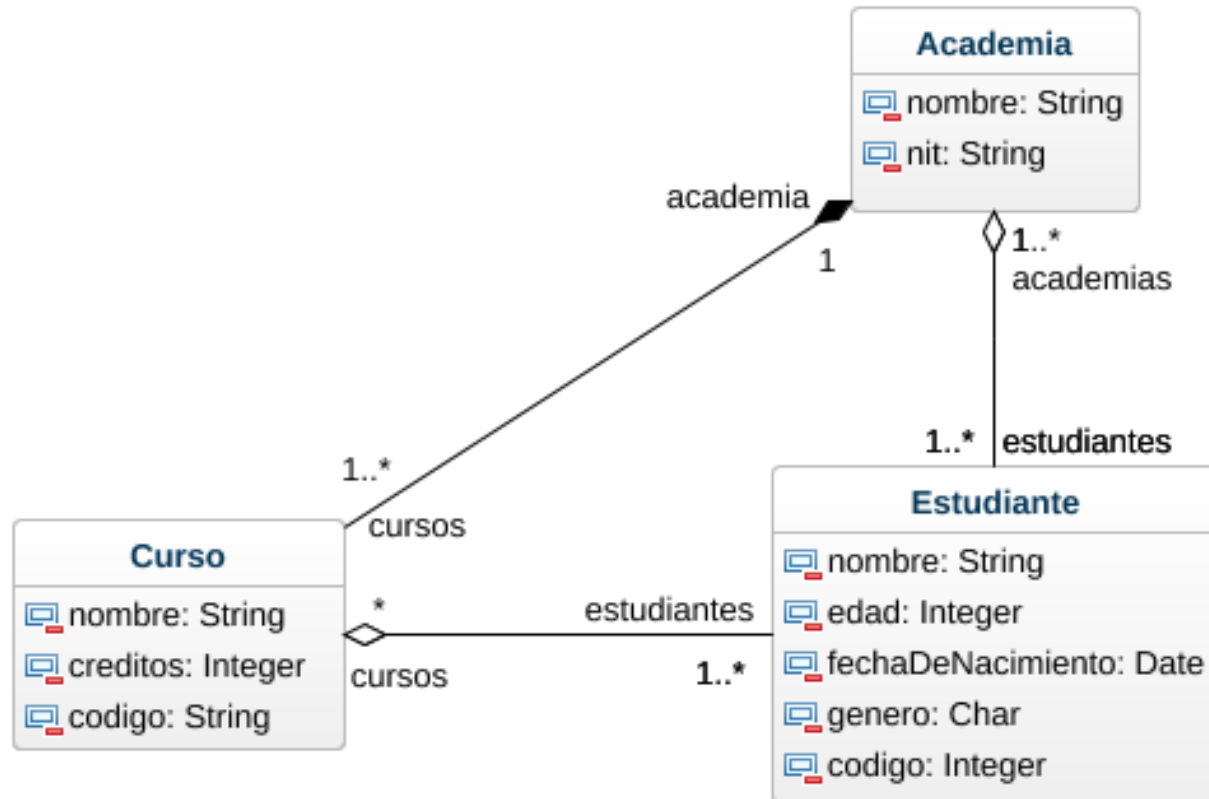
5) En una empresa se llevan a cabo proyectos, estos proyectos están formados por una ó más actividades y a su vez cada actividad tiene 1 ó más tareas específicas. Cada tarea es asignada a un empleado y los empleados pueden o no tener asignadas tareas.

Proyectos \rightarrow Actividad ($1..* \rightarrow 1..*$) Agregación

Actividad \rightarrow Tareas ($1 \rightarrow 1..*$) Agregación

Tareas \rightarrow Empleados ($1 \rightarrow 0..1$) Asociación

Ejercicios Agregación y Composición



Ejercicios Agregación y Composición

- Basado en la lectura y única interpretación del diagrama de clases, selecciona todas las afirmaciones verdaderas sobre la asociación compuesta Academia y Curso:
 - Si se elimina una instancia de Academia, las instancias de Curso pueden seguir existiendo.
 - Una academia debe tener al menos un curso.
 - Un curso puede ser dado por varias academias.
 - Si se elimina una instancia de Academia, se eliminan las instancias de Estudiante respectivas de la clase.

Diagrama de Clases: Generalización

- Son asociaciones entre elementos generales y elementos más específicos, en los cuales éstos últimos son consistentes totalmente con los primeros, por lo que heredan las características proporcionadas por lo elementos generales y además pueden aumentar información.
- Este tipo de relación también se conoce como **herencia**.
- En una generalización no hay multiplicidad ni roles.
- **Una Asociación define las reglas de cómo los objetos se pueden relacionar entre ellos.**
- La visibilidad “protected” permite que solo objetos de la misma clase ó subclase vean el elemento.

Diagrama de Clases: Generalización

- La Generalización consiste en **factorizar** las propiedades comunes de un conjunto de clases en una clase más general.
- Los nombres usados: clase padre - clase hija.
- Otros nombres: superclase - subclase, clase base - clase derivada.
- Las subclases **heredan propiedades** de sus clases padre, es decir, atributos y operaciones

Diagrama de Clases: Generalización

- Para dibujar, hay que definir:
 - **Superclase:** es una clase que contiene alguna combinación de atributos, operaciones y asociaciones que son comunes a dos o más tipos de objetos que comparten el mismo propósito.
 - **Subclase:** es una clase que contiene una combinación de atributos, operaciones y asociaciones que son únicas a un tipo de objeto definido por una superclase.
 - La superclase es reutilizada por la subclase.

Diagrama de Clases: Generalización

Generalización / Especialización: (“es un”)

- *Generalización*: Se crea una clase (superclase), que generaliza las propiedades comunes de varias clases.
- *Especialización*: Dada una clase, se crea(n) otra(s) clase(s) (subclase) que especializa(n) la clase dada, agregando las diferencias.

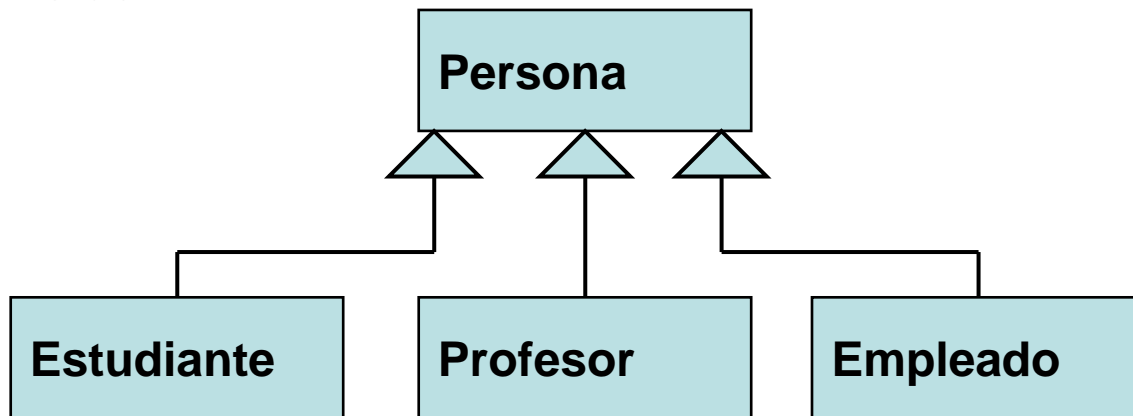
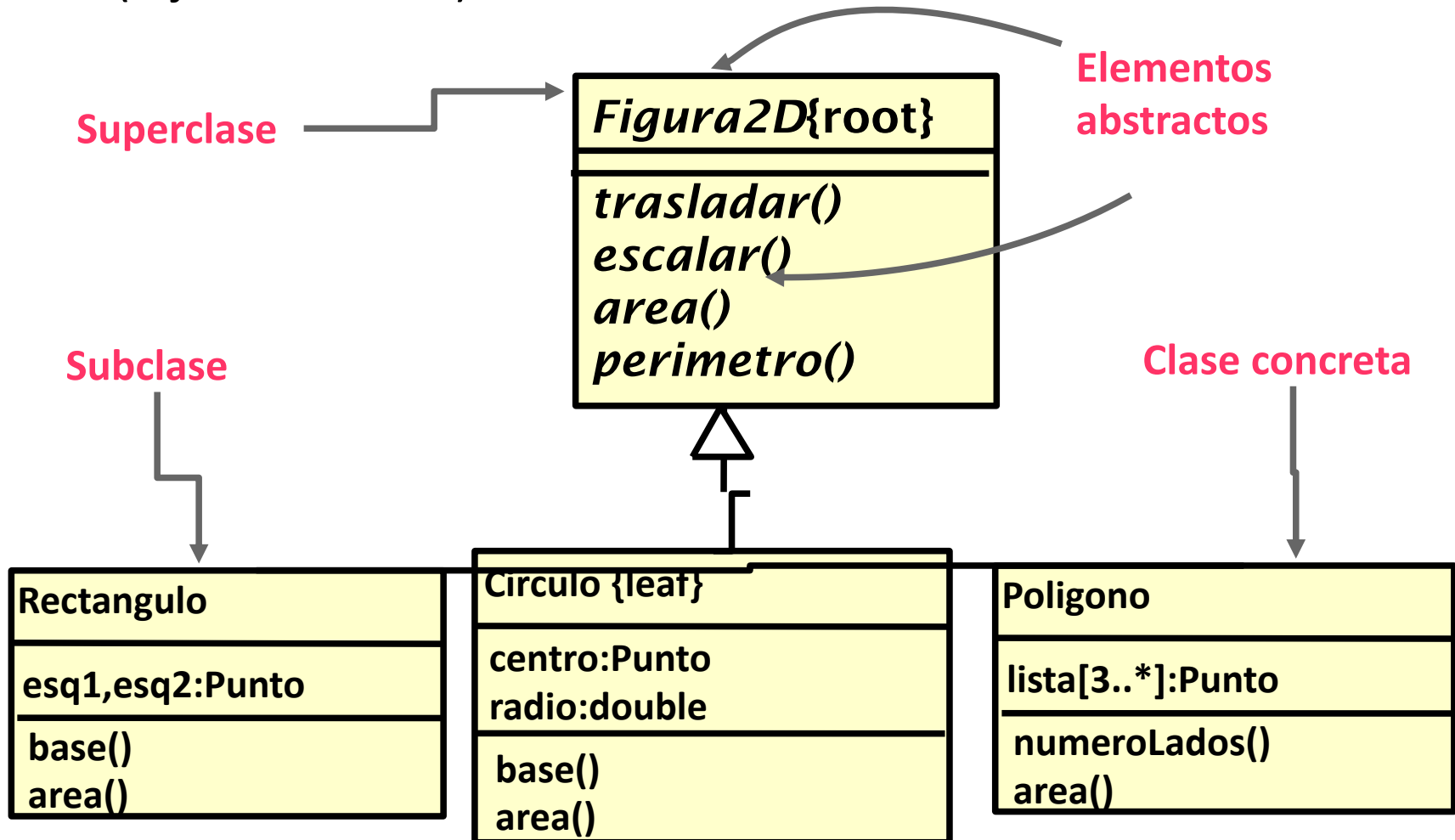


Diagrama de Clases: Generalización

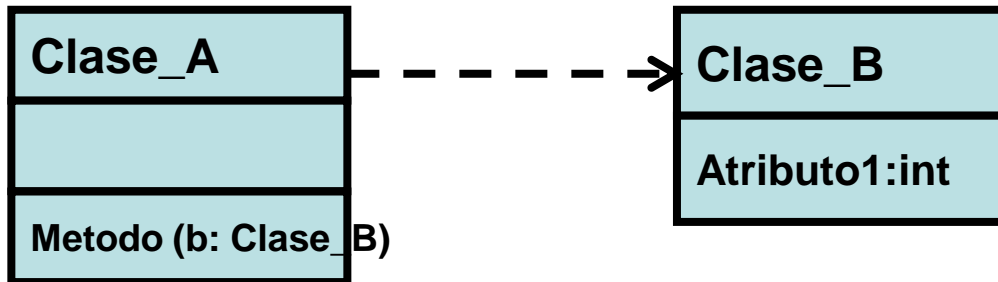
- La **Generalización** (relación “es-un”) es una relación entre un elemento general (padre o superclase) y un caso específico de éste (hijo o subclase).



Diagramas de Clases: Dependencia

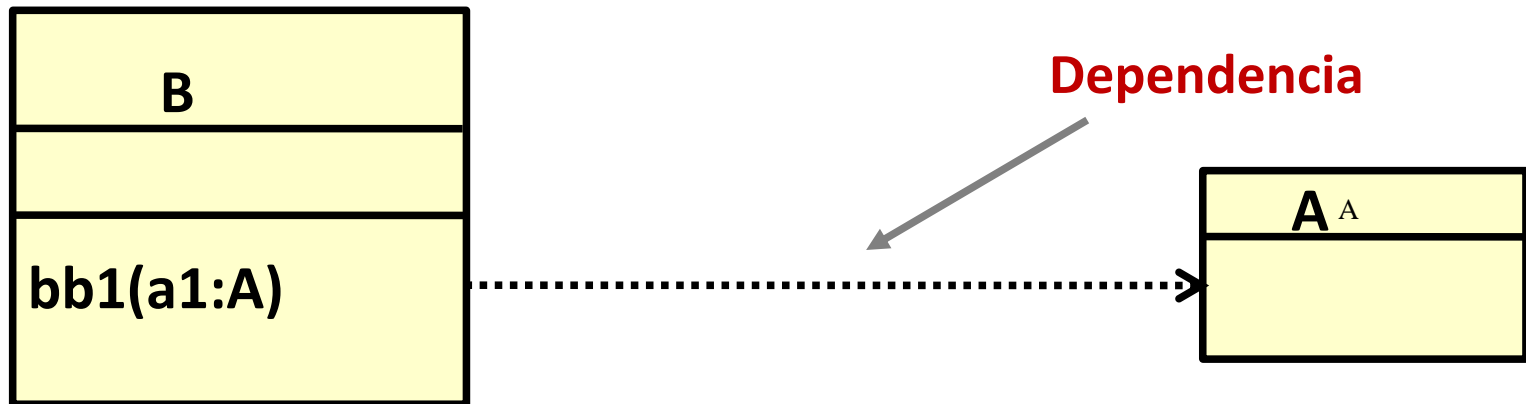
Dependencia:

- Es una conexión entre clases que indica que un cambio en una clase B puede afectar a otra clase A que la *usa*.



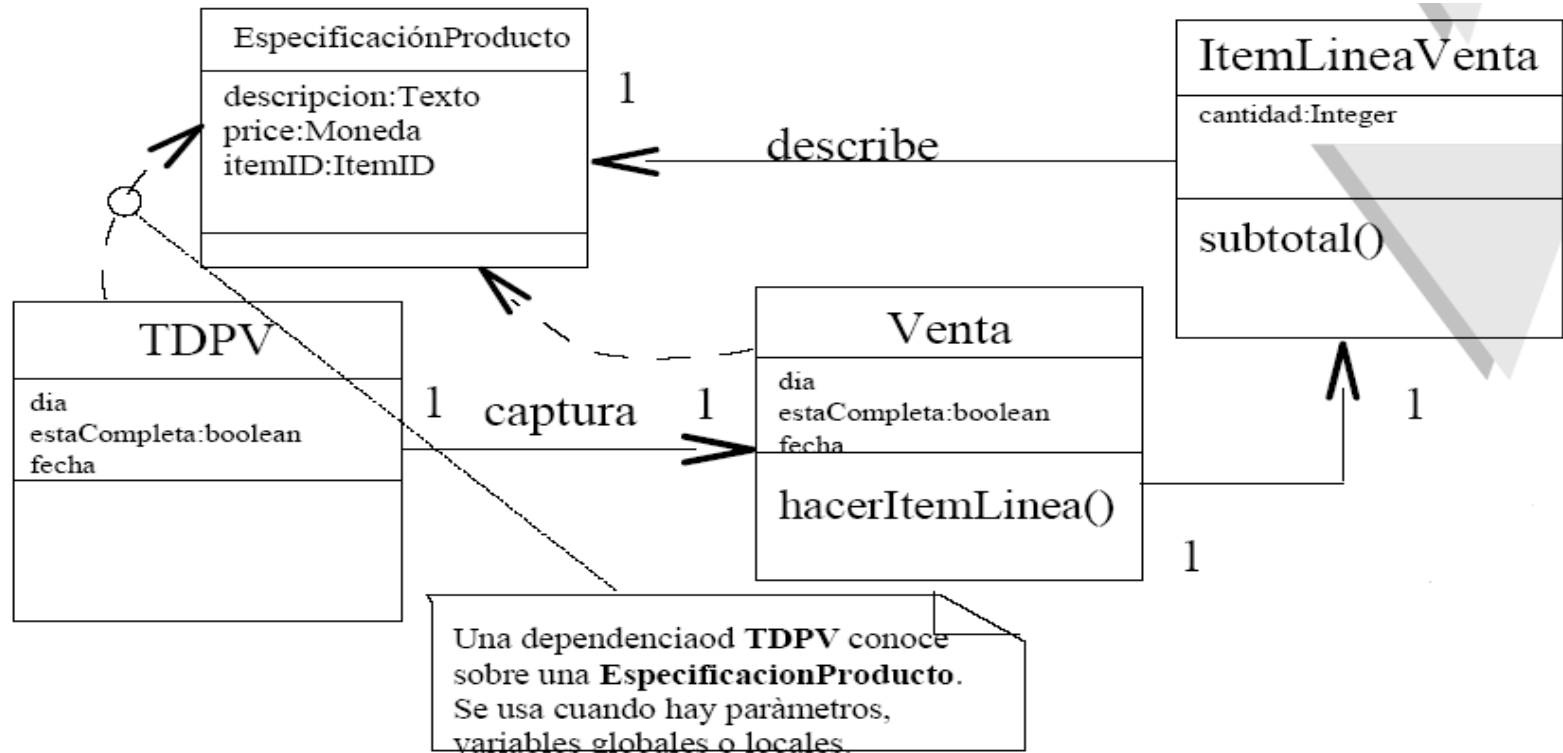
Diagramas de Clases: Dependencia

- Una **Dependencia** modela una relación de uso.
- En una dependencia no es necesario especificar un nombre.



Diagramas de Clases: Dependencia

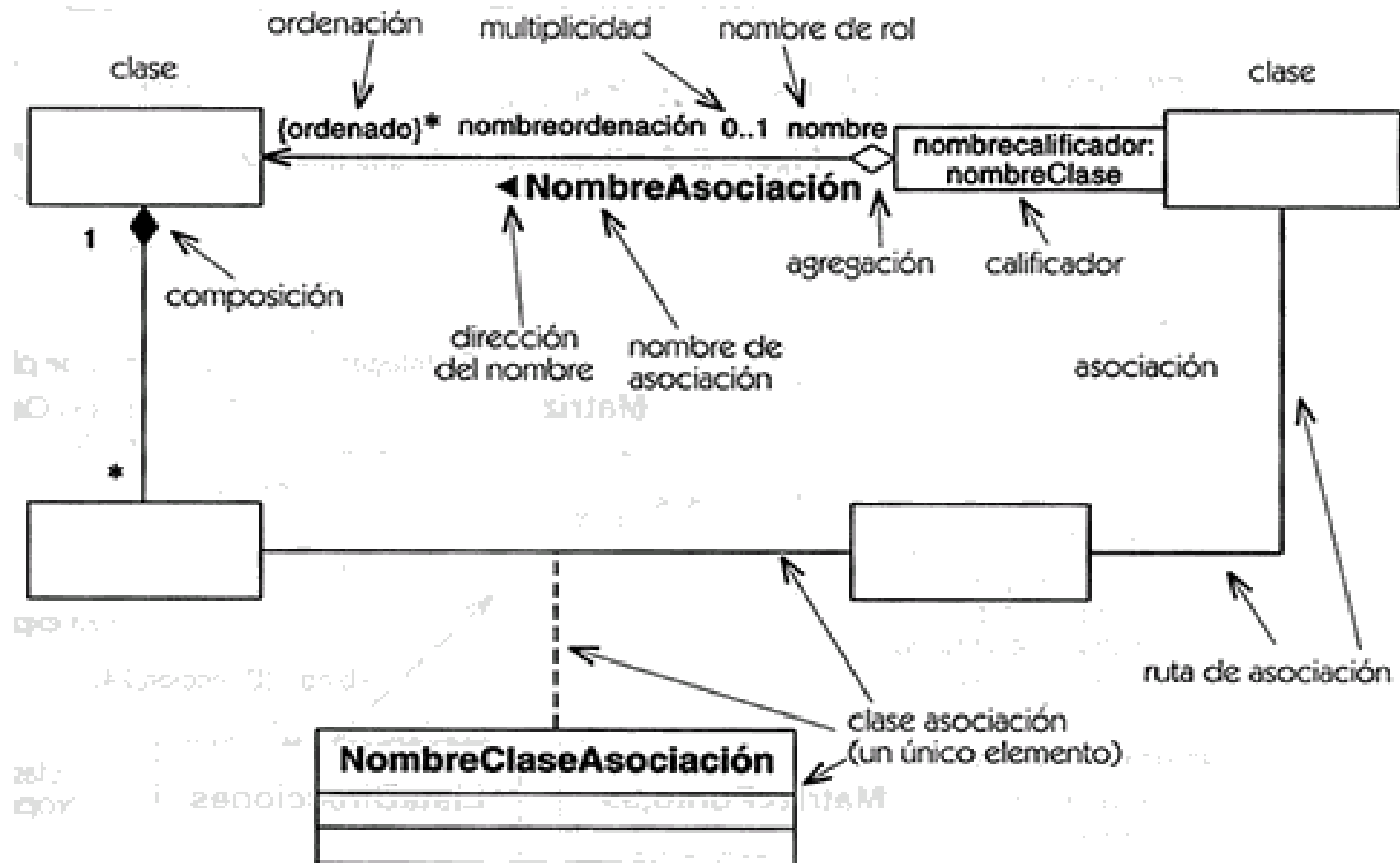
- En UML una *relación de dependencia* indica que un elemento tiene conocimiento de otro elemento.



Construcción del diagrama de clases

1. Identificar las clases, nombrarlas y definirlas con lo que sabes que son parte del modelo.
2. Mostrar los atributos y operaciones (posteriormente)
3. Identificar, nombrar y definir las asociaciones entre pares de clases. Tener cuidado con clases reflexivas
4. Evaluar cada asociación para determinar si debe ser una agregación y cada agregación para ver si debe ser una composición
5. Evaluar las clases para posible generalización (herencia).
6. Etiquetar asociaciones y en caso necesario los roles
7. Indicar multiplicidad
8. Dibujar flechas de dirección

Construcción del diagrama de clases



Construcción del diagrama de clases

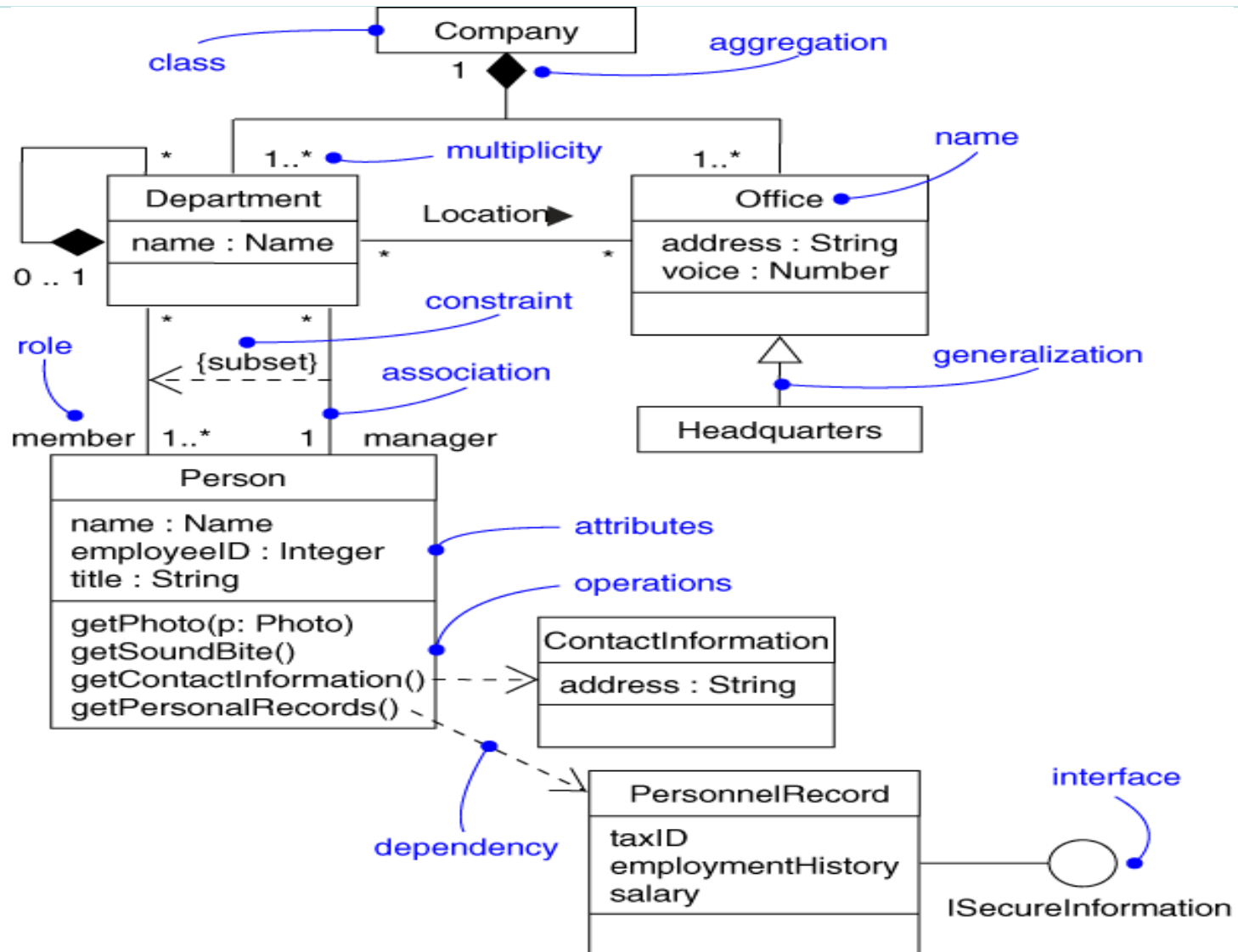
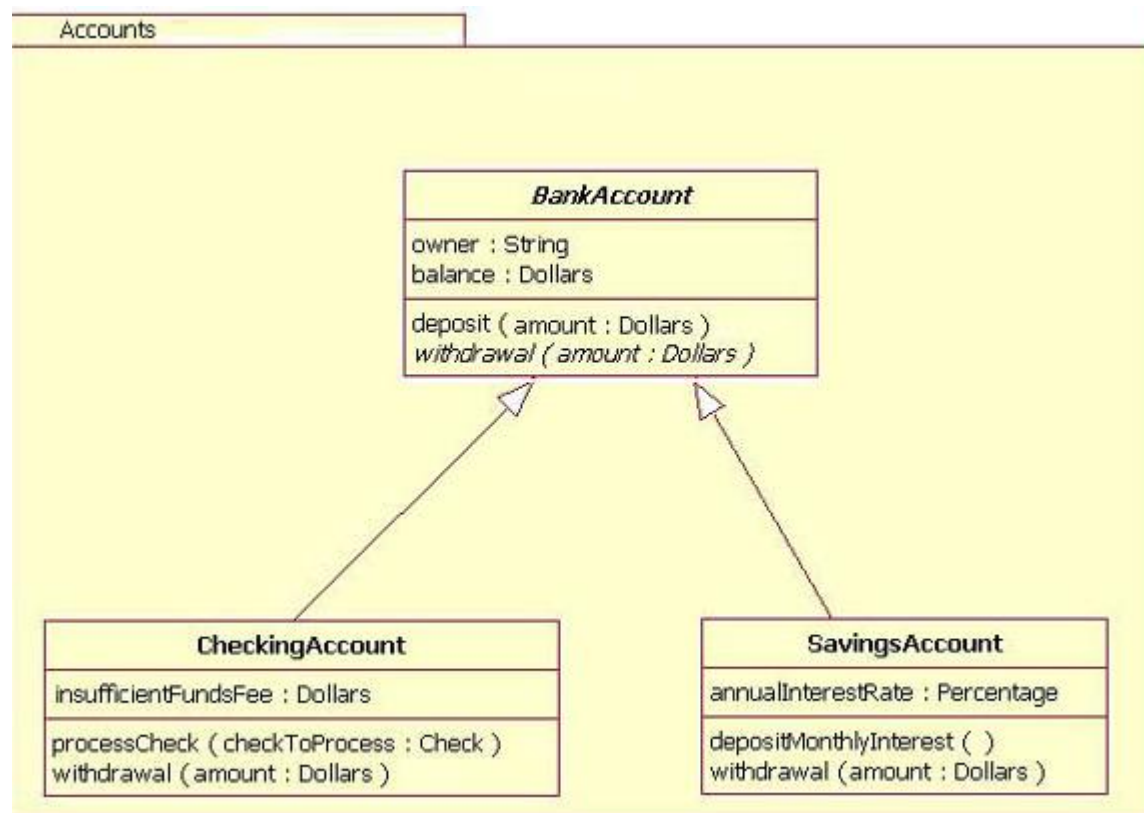


Diagrama de Paquetes

- Es un elemento organizador que proporciona UML al dividir el sistema en paquetes lo hace más fácil de entender.



Descripción del problema

Problema: Asignación de proyectos Informáticos

- En un plan de estudios de una universidad, hay una asignatura denominada "proyectos".
- Para aprobar dicha asignatura el alumno tiene que desarrollar un trabajo práctico, en el que resuelva un determinado problema aplicando los conocimientos adquiridos durante su formación.
- Durante la realización del proyecto (trabajo), el alumno recibe la dirección tutelada de un profesor. Para ello los profesores definen una serie de proyectos(trabajos) a realizar, y a los alumnos se les adjudica un proyecto, de entre sus elegidos, en función de un determinado baremo (Conjunto de normas establecidas).

Descripción del problema

Proceso a seguir :

1. Los alumnos se matriculan de la asignatura “proyectos informáticos”.
2. Los profesores definen los contenidos de los proyectos propuestos, dando el título del proyecto, las asignaturas recomendadas y una descripción general del mismo.
3. Cada alumno elige entre 1 y 10 trabajos de los ofertados.
4. A cada una de sus elecciones le asigna una prioridad.
5. Una vez terminada la elección se asigna los trabajos a los alumnos, teniendo en cuenta el siguiente baremo: *suma de la nota media del expediente y la nota media de las asignaturas recomendadas en el proyecto* (que hayan sido cursadas por el alumno).

Descripción del problema

Restricciones:

- Un proyecto puede ser asignado como máximo a tres alumnos.
- No puede quedar ningún alumno sin proyecto.
- Puede haber proyectos sin alumnos.
- Un profesor puede definir más de un proyecto.
- Un alumno sólo puede ser asignado a un proyecto.

Desarrollo: Incorporar conceptos

Pasos a seguir:

1. **Identificar** los conceptos (entidades).
2. **Seleccionar** los conceptos relevantes del problema.
3. **Representar** esos conceptos en el modelo conceptual, como clases de un Diagrama de clases de UML.

Selección de conceptos

Problema: Asignación de Proyectos

Lista preliminar de conceptos:

- Baremo, Alumno, Profesor, Dirección Tutelada, Plan de Estudios, Asignatura, Nota Media, Titulación Universitaria, Expediente, Proyecto, Trabajo Práctico, Título, Descripción, Problema, Asignatura Recomendadas, elecciones y prioridad.
- Para cada concepto encontrado seleccionar aquellos que sean relevantes para nuestro problema. Tratando de detectar:
 - Términos **sinónimos**: (**Plan de Estudios**, Titulación Universitaria), (**Proyecto**, Trabajo, Problema, Trabajo Práctico)
 - **Relaciones**: Dirección Tutelada, Asignaturas Recomendadas y elecciones.
 - **Atributos** de conceptos o de relaciones: Baremo, Nota Media, Título, Descripción, prioridad

Representar los conceptos

Problema: Asignación de Proyectos

Lista Definitiva de conceptos:

Alumno, Profesor, Plan de Estudios, Asignatura, Expediente y Proyecto.

Representar estos conceptos en el modelo conceptual usando las clases de UML.

Alumno

Profesor

Plan Estudios

Asignatura

Expediente

Proyecto

Incorporar Asociaciones

Una **asociación** es una conexión significativa y relevante entre dos conceptos.

Pasos a seguir:

1. **Identificar** posible asociaciones.
2. **Representarlas en el Modelo de Clases y Seleccionar** asociaciones válidas.
3. **Nombrar** las asociaciones seleccionadas.
4. **Identificar multiplicidad.**

Identificar Asociaciones

Problema: Asignación de Proyectos

1.En un plan de estudios de una titulación universitaria, hay una asignatura denominada "proyectos".

Plan Estudios ----- Asignaturas (1)

2.Para aprobar dicha asignatura el alumno tiene que desarrollar un trabajo práctico, en el que resuelva un determinado problema aplicando los conocimientos adquiridos durante su formación.

Alumno ----- Proyecto (2)

3.el alumno recibe la dirección tutelada de un profesor.

Alumno ----- Profesor (3)

Identificar Asociaciones

Problema: Asignación de Proyectos

4. Los profesores definen una serie de proyectos(trabajos)

Profesor ----- Proyecto (8)

5. a los alumnos se les adjudica un proyecto

Alumno ----- Proyecto (2)

6. Entre sus elegidos,

Alumno ----- Proyecto (2)

7. Los alumnos se matriculan de dicha asignatura “proyecto”.

Alumno ----- Asignatura (2)

Identificar Asociaciones

Problema: Asignación de Proyectos

8. Expediente del Alumno

Expediente ----- Alumno (8)

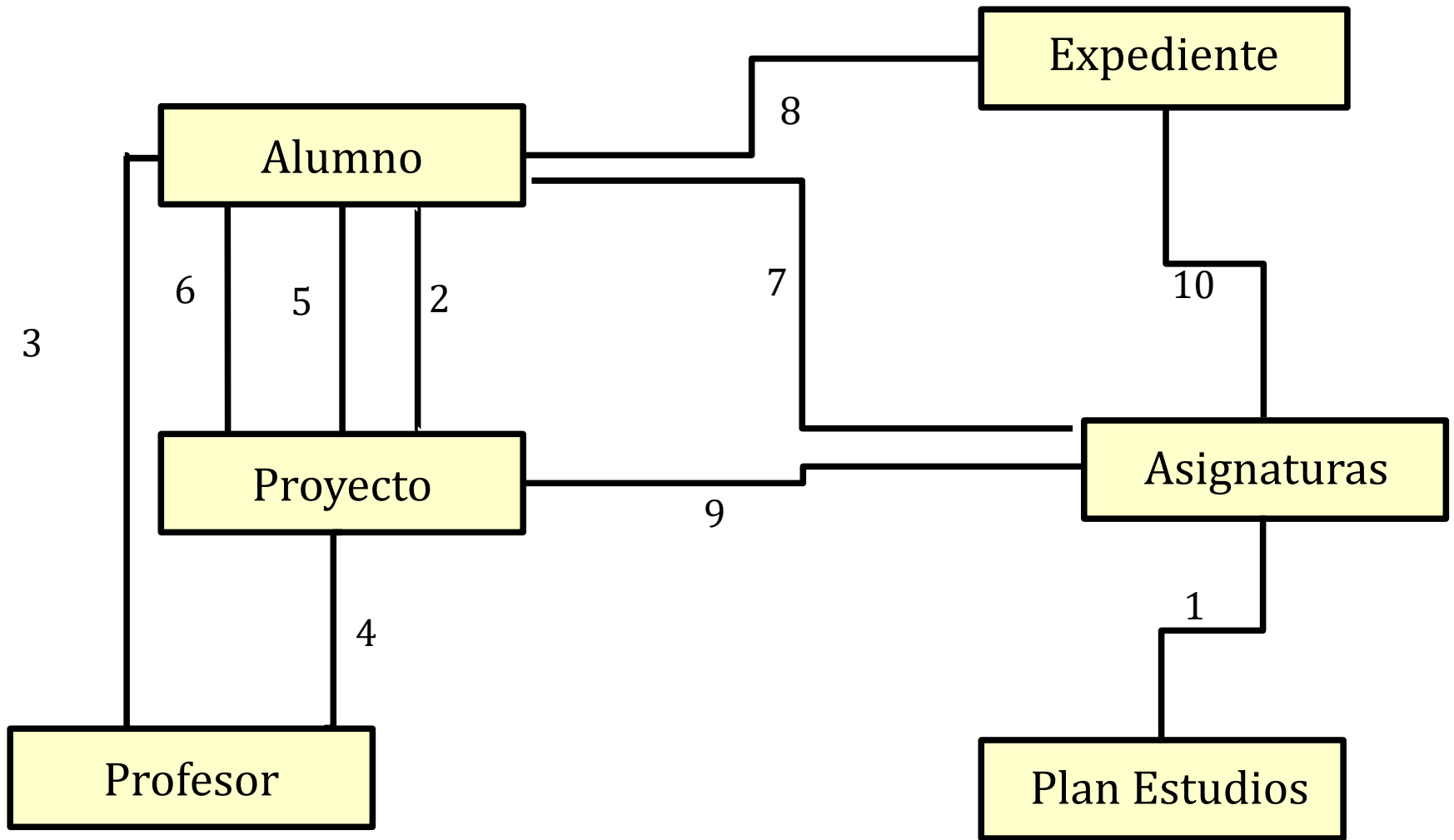
9. Asignaturas recomendadas en el proyecto

Asignatura ----- Proyecto (9)

10. Del enunciado del problema (punto 4º del proceso a seguir) se deduce que: El Expediente está formado por Asignaturas y sus notas.

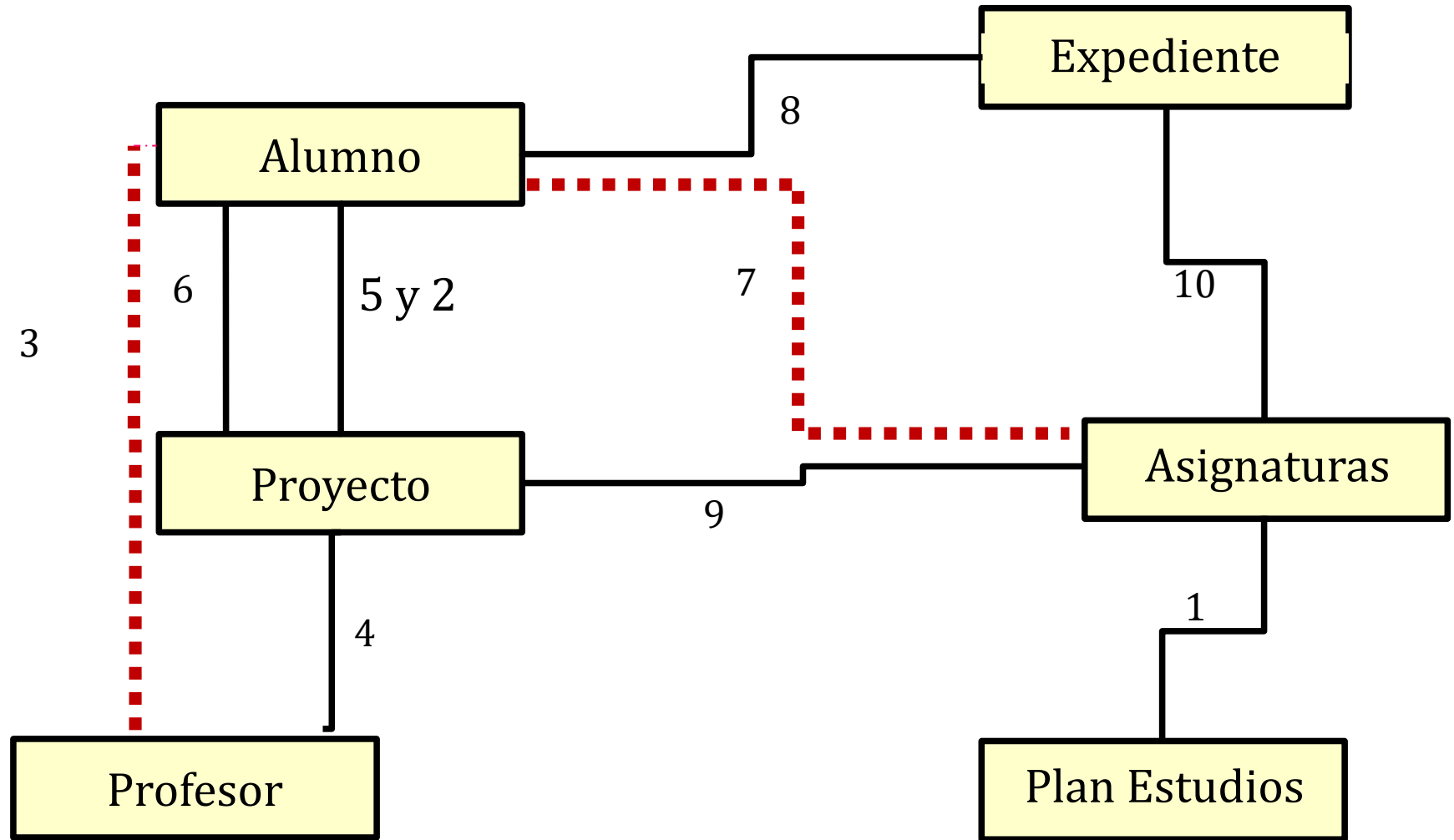
Asignatura ----- Expediente (1)

Representar y seleccionar asociaciones



Problema: Asignación de Proyectos

Representar y seleccionar asociaciones



..... Asociación redundante o derivada

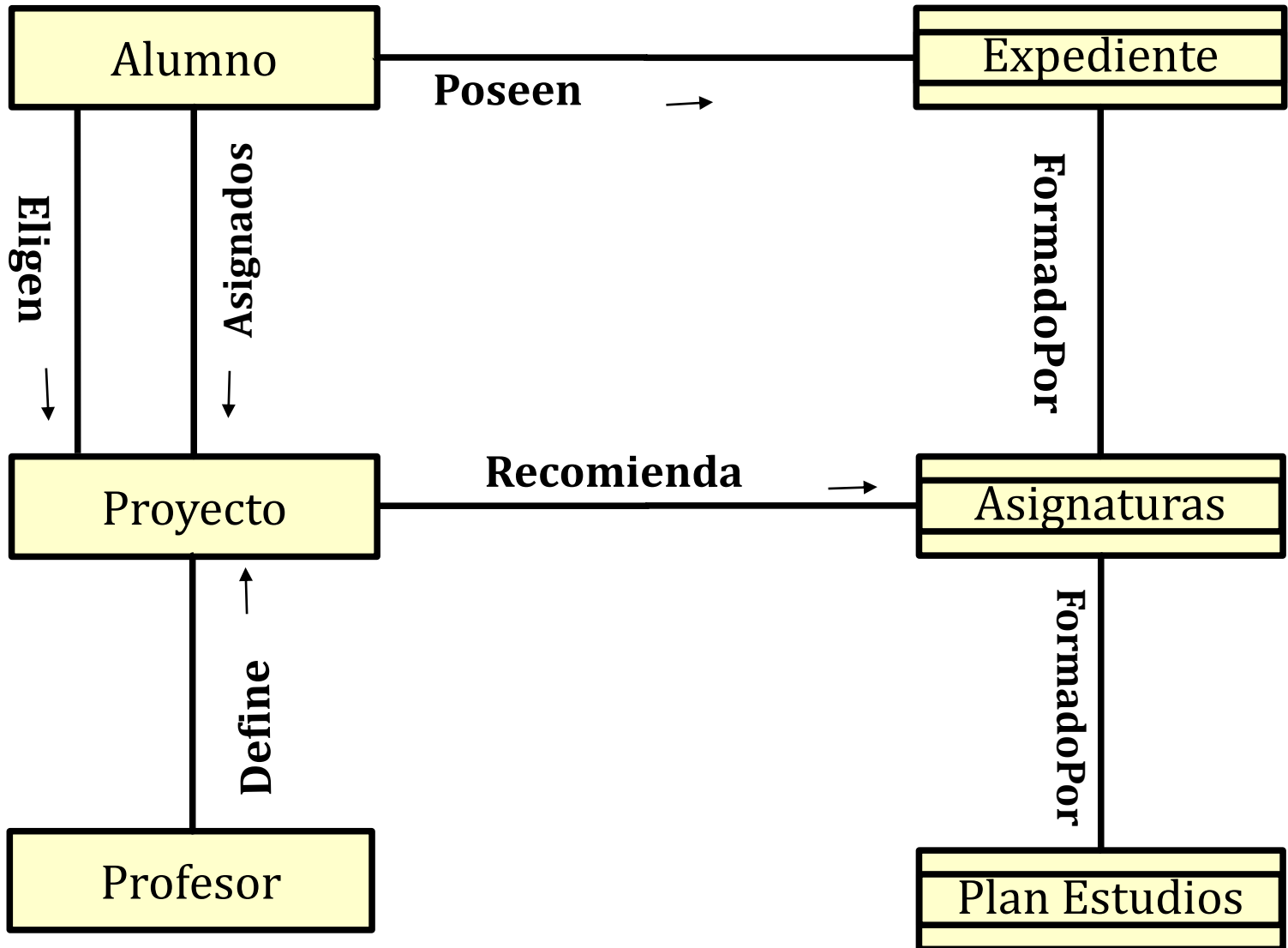
Problema: Asignación de Proyectos

Nombrar Asociaciones

Problema: Asignación de Proyectos

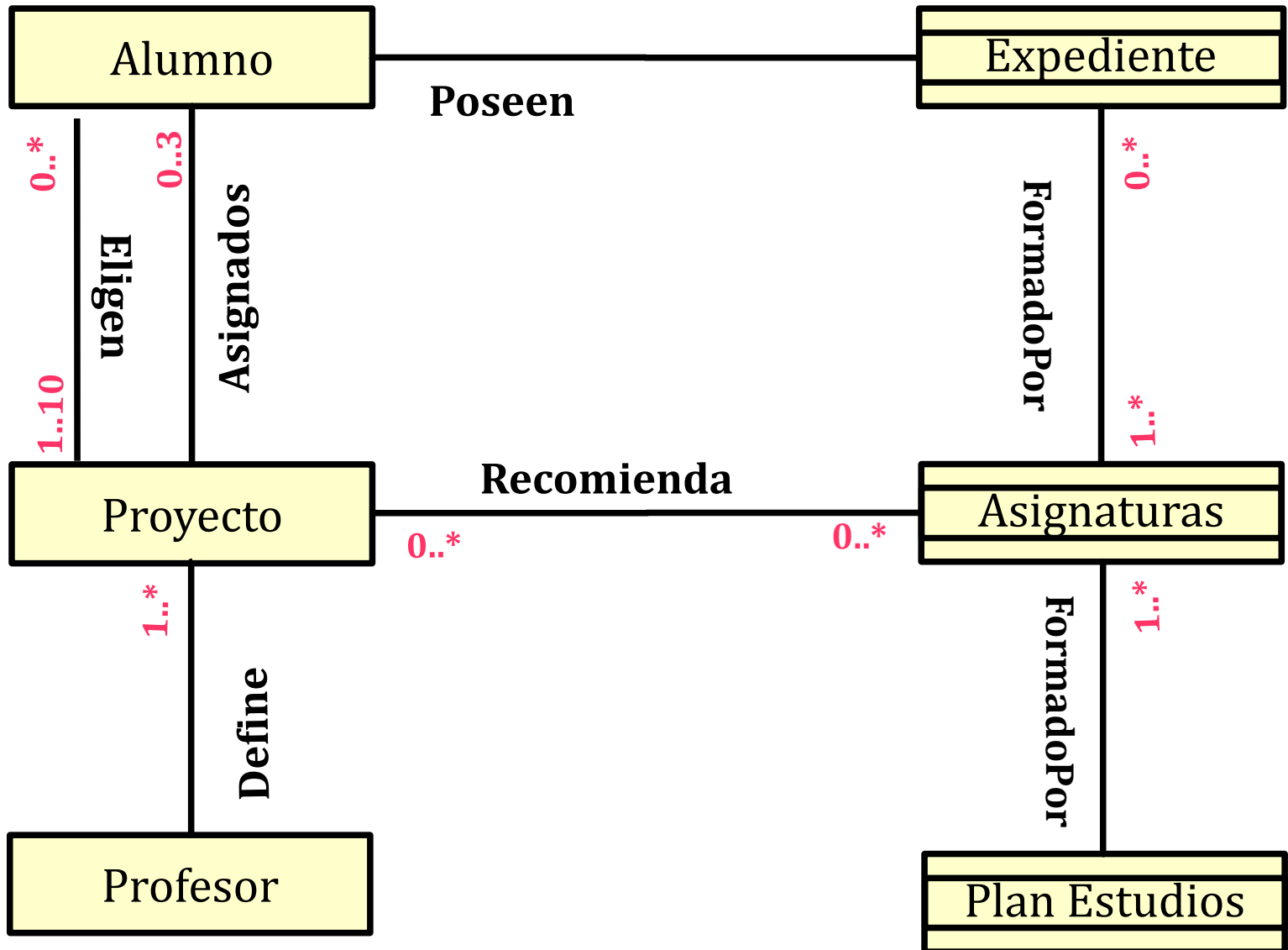
1. Un Plan de Estudios **está formado** por Asignaturas.
6. Los alumnos **eligen** proyectos.
- 2 y 5. Los Proyectos son **asignados/realizados** a/por Alumnos.
4. Los profesores **definen** Proyectos.
8. Los alumnos **tienen** expedientes.
10. Los expedientes **están formados** por Asignaturas y su nota.
9. Los proyecto **recomiendan** asignaturas.

Nombrar Asociaciones



Problema: Asignación de Proyectos

Identificar Multiplicidad



Problema: Asignación de Proyectos

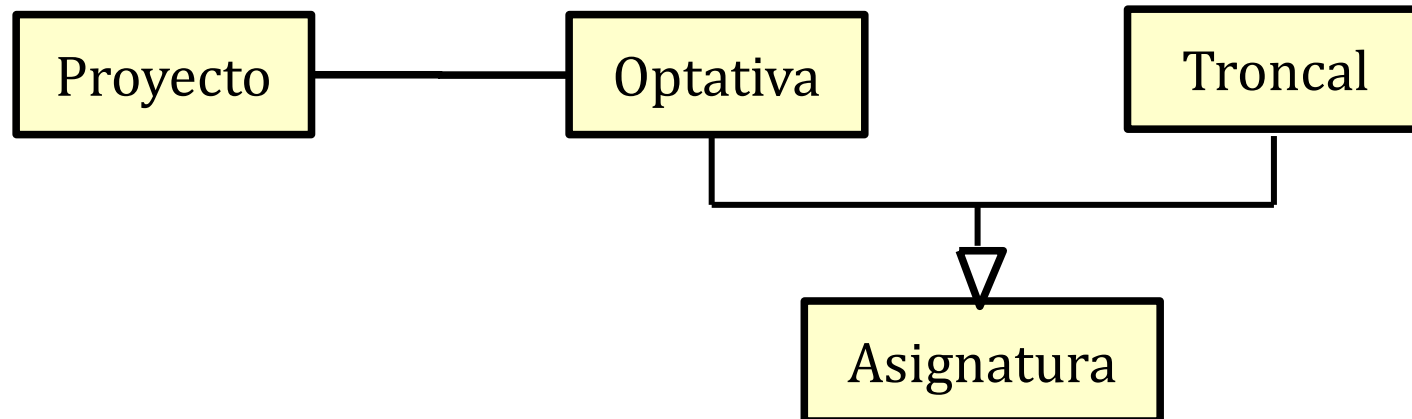
Incorporar generalizaciones

Problema: Asignación de Proyectos

Supongamos que:

2º *Los profesores definen los contenidos de sus proyectos, ..., las **asignaturas optativas** recomendadas, ...*

- Hay dos tipos de asignaturas: **Optativas y Troncales.**
- Está justificada la subclasificación: Si, ya que asignatura optativa tiene una asociación relevante con Proyecto.
- Representación en el modelo conceptual



Agregar Atributos

Pasos a seguir:

1º Identificar Atributos desde:

- La descripción del problema ,o
- de otras fuentes de información.

2º Representarlos en el Modelo Conceptual en los conceptos o en las asociaciones que correspondan.

Tipos de atributos válidos:

- **Primitivos** o valores puros de datos:(Entero, Real, Carácter, Boolean, Cadena).
- **No primitivos**: (Nombre de persona, Número de teléfono, Hora, Fecha, Dirección, Punto,...)

Agregar Atributos

Problema: Asignación de Proyectos

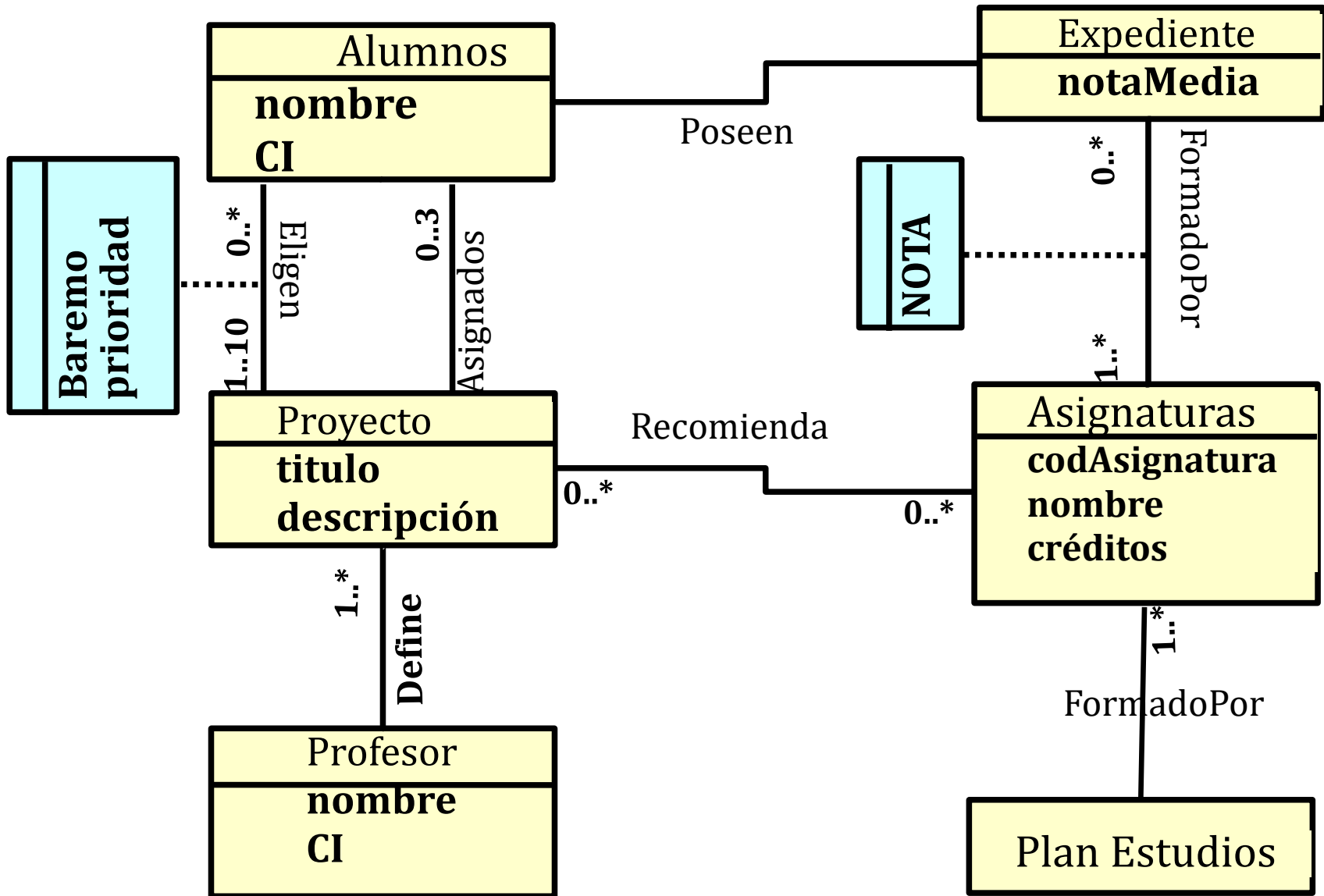
Atributos extraídos de la descripción del problema:

- **Nombre** de la Asignatura. (de Asignatura)
- **Título** del Proyecto. (de Proyecto)
- **Descripción** del Proyecto. (de Proyecto)
- **Nota media** del Expediente. (de Expediente)
- **Baremo**. (de la asociación Eligen).
- **Prioridad** (de la asociación Eligen).
- **Nota** de un Alumno en una Asignatura. (de la asociación Formado Por entre Expediente y Asignatura).

Atributos extraídos de otras fuentes:

- Nombre y CI del Profesor y del Alumno.
- Código y Créditos de la Asignatura.

Agregar los Atributos



Problema: **Asignación de Proyectos**

Diagrama de Clases: Ejercicio 1

Crea un diagrama de clases para representar la información siguiente:

- Existen tres tipos de entidades geográficas: puntos, líneas y áreas.
- Cada entidad geográfica tiene un nombre y un código.
- Un punto esta definido por sus coordenadas.
- Una línea esta definida por dos puntos por lo menos.
- Un área esta definida por tres puntos por lo menos.

Diagrama de Clases: Ejercicio 1

