

# Tema 2.3: Modelo del Dominio



A. Goñi, J. Ibáñez, J. Iturrioz, J.A. Vadillo



OCW  
2013



# Artefactos a conseguir en la captura de requisitos

- Un artefacto es cualquier clase de descripción o de información (modelo, elemento de un modelo o documento) del sistema

Actores ✓

Casos de Uso ✓

Prototipos de Interfaces de Usuario ✓

Glosario

Diagramas de clase  
(Modelo del Dominio)

Descripción de la arquitectura

# Artefacto: Glosario

**GLOSARIO:** Documento donde se definen los términos más comunes e importantes utilizados en el sistema

■ **ASIGNATURA:** ...

**-- EJEMPLO DE GLOSARIO --**

■ **ESTUDIANTE:** es una persona que está estudiando una carrera en la universidad UnivX. Necesariamente debe estar matriculado en por lo menos una ASIGNATURA.

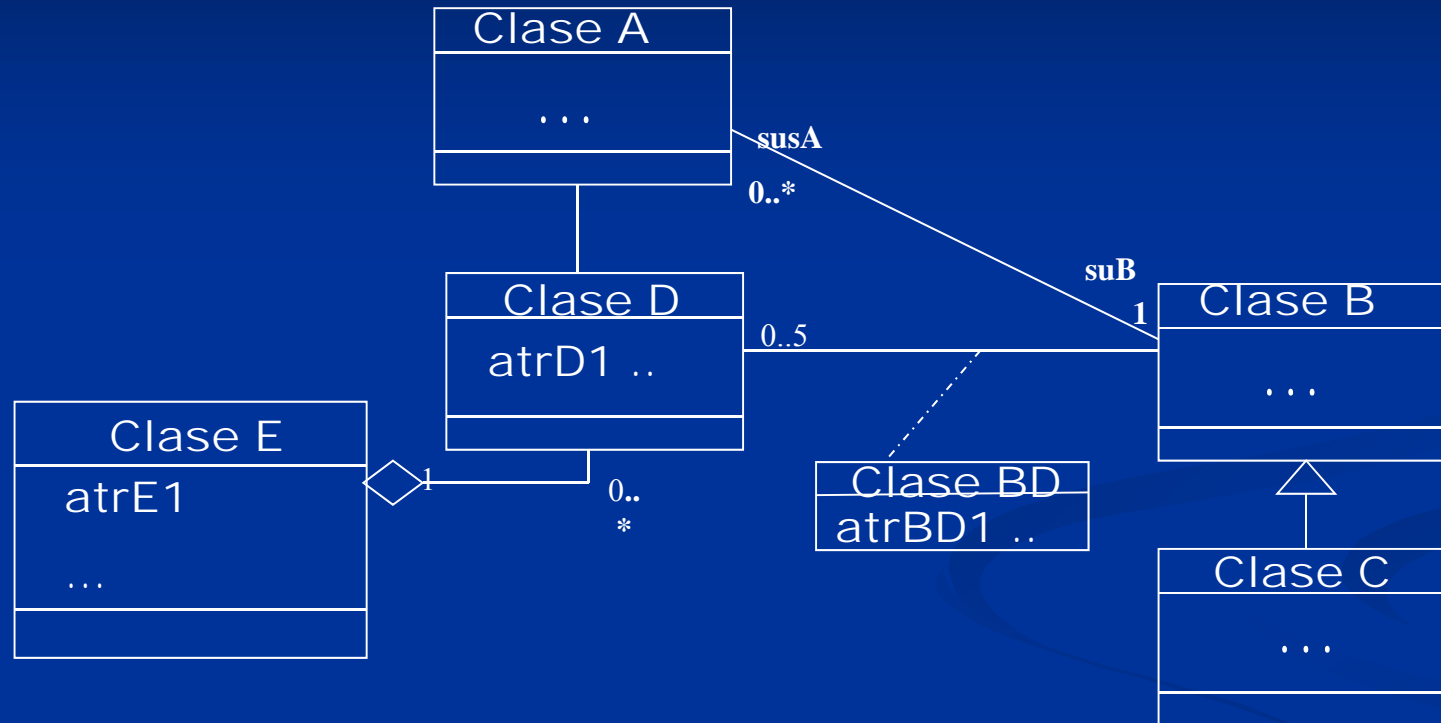
■ **MATRÍCULA:** es el resultado de un proceso administrativo por el cual un ESTUDIANTE adquiere el derecho a ser evaluado en dos convocatorias de una ASIGNATURA. Se le asocia a un GRUPO. Tiene derecho a asistir a las clases del PROFESOR responsable de dicha ASIGNATURA en el GRUPO asignado.

■ **PROFESOR:** es una persona que trabaja en UnivX y que imparte al menos una asignatura de una determinada TITULACIÓN. Se encarga de evaluar a todos los estudiantes matriculados en la asignatura y asignados a sus grupos. El profesor no puede ser estudiante en la misma carrera en la que imparte clases, pero sí en otras.

# Artefacto: Modelo del Dominio

- Captura los tipos de objetos en el contexto del sistema y sus relaciones.
  - “cosas” que existen
  - eventos que suceden
- Seguramente aparecerán en el glosario
- Definido con diagrama de clases UML

# Diagrama de clases en UML



Durante la captura de requisitos se utiliza para representar el MODELO DEL DOMINIO. De momento, sólo interesan los ATRIBUTOS de las clases y NO SUS MÉTODOS

# Clase UML

## VISIBILIDAD:

+ = público

- = privado

# = visible para  
subclases

El símbolo / indica que un atributo es calculado o derivado a partir de otros atributos

### NOMBRE DE LA CLASE

atributo1

atributo2

...

método1 (parámetros): resultado

método2 (parámetros) : void

...

-- responsabilidades de la clase

-- texto que indica qué hace,  
restricciones especiales de uso, etc.

Los objetos de dicha clase pueden almacenar valores en los atributos y ejecutar sus métodos

# Ejemplo: Clase UML

Cliente
- nombre, dirección, tfno: String - fechaNac: Date -Aficiones: Vector(String) ... / númeroDeAficiones: int
+ getNombre(): String,... + setNombre(n: String): void + addAficion(a:String): void ...
- - almacena datos de clientes

Los tipos (String, Date, void,..) NO son definidos por UML. Se suelen usar los del LP que se escoja

# La clase CLIENTE permite definir los siguientes objetos

@C1 

P. Pérez	Nagusia 13	943112232	3/3/89	Fútbol, Leer
----------	------------	-----------	--------	--------------

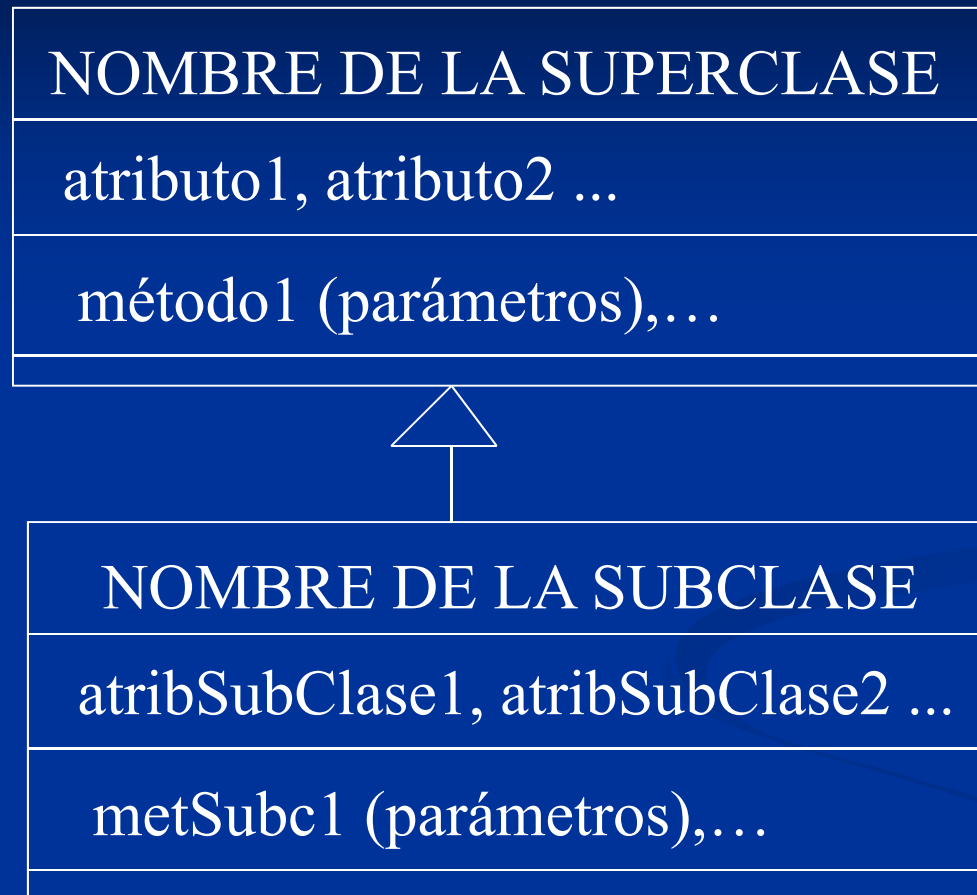
@C2 

K. Sola	Easo 3	943222232	4/3/89	Mus, Monte
---------	--------	-----------	--------	------------

Cada objeto representa a  
un cliente real

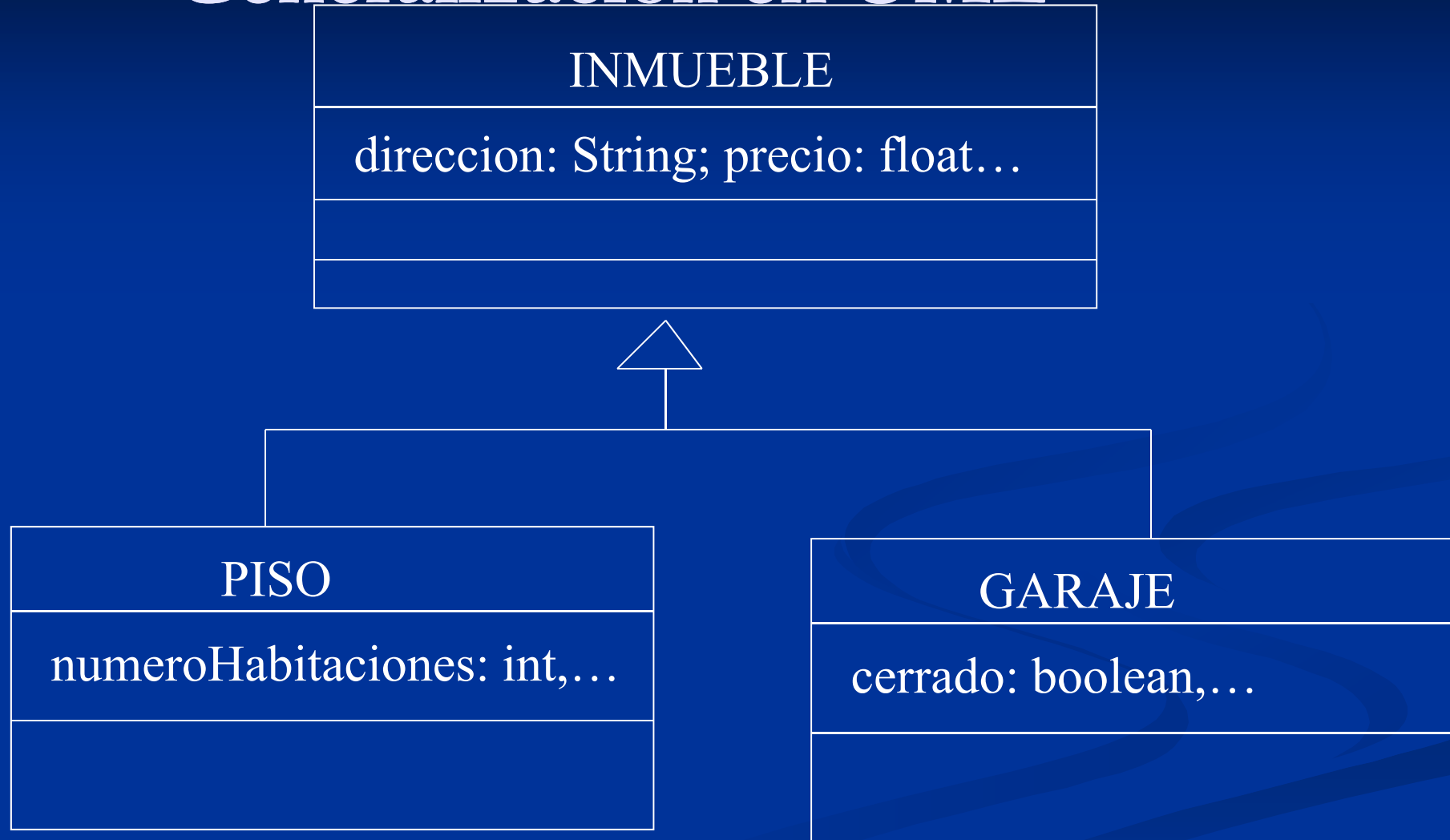


# Especialización y Generalización en UML

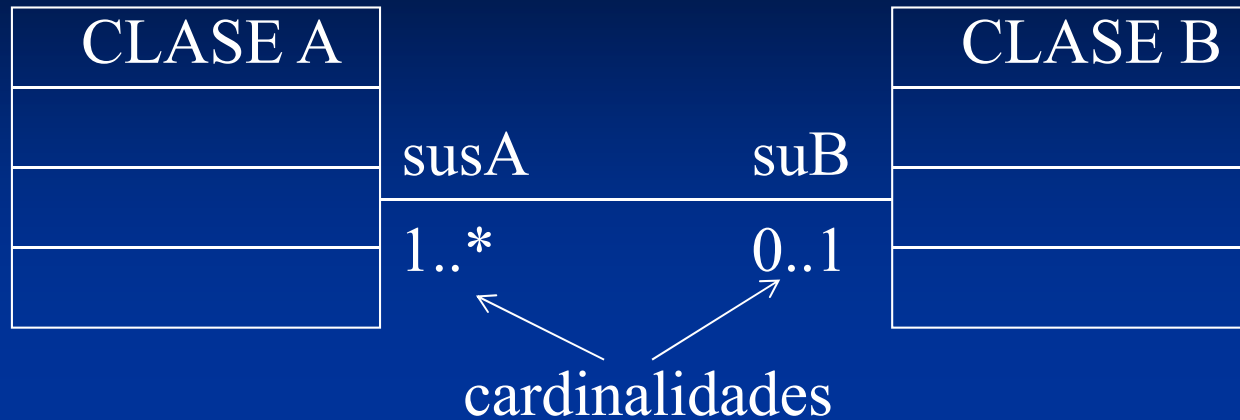


**La SUBCLASE hereda todos los ATRIBUTOS y los MÉTODOS de la SUPERCLASE.**

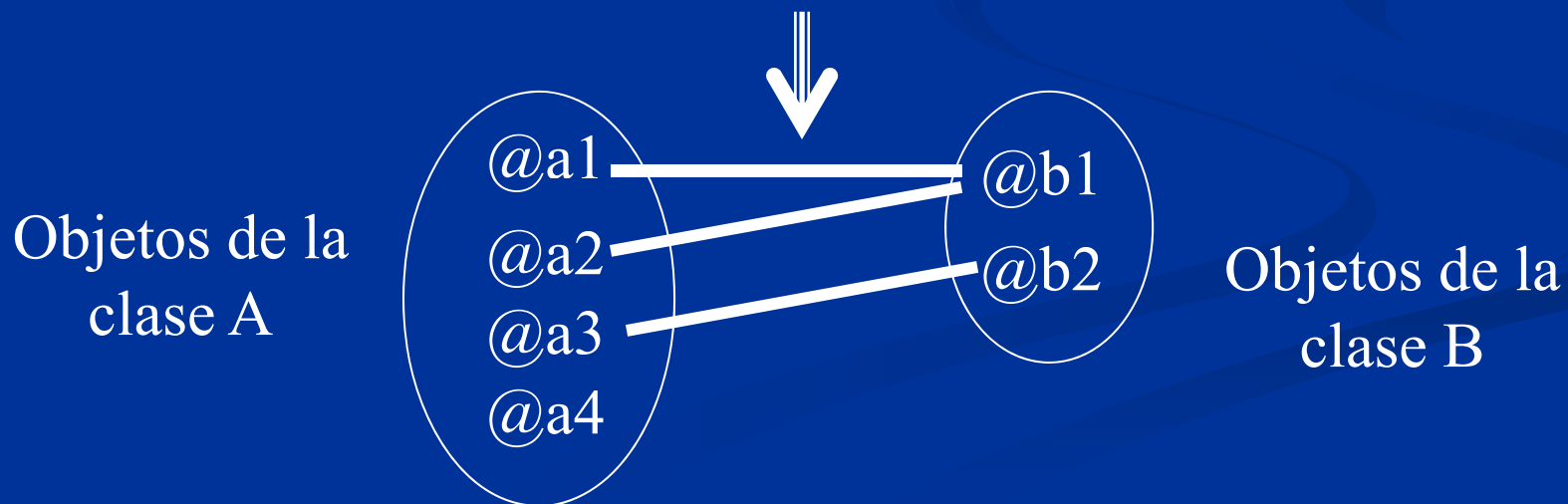
# Ejemplo de Especialización y Generalización en UML



# Asociación en UML



Almacenar pares <objeto de A, objeto de B>  
Indicando CARDINALIDAD



# Cardinalidades en UML

1 → con uno

0..1 → con uno o ninguno

\* → con cero, uno o varios

0..\* → con cero, uno o varios

1..\* → con uno o varios

n → con n exactamente

□ n..m → mínimo con n y máximo con m

Nota: n y m son números naturales

Ejemplo: 8 , 17 , 7..9 , ...

# Ej. de Asociación en UML



Otra opción es dar un único nombre a la asociación e indicar “cómo se lee”



# Ej. de Asociación en UML

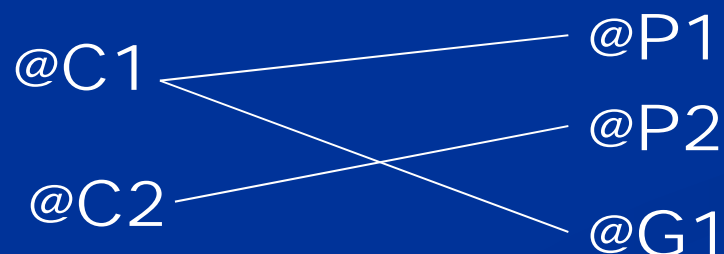
@C1	P. Pérez	Mayor 13	943112232	3/3/89	Leer, Fútbol
-----	----------	----------	-----------	--------	--------------

@C2	K. Sola	Mayor 3	943222232	4/3/89	Mus, Monte
-----	---------	---------	-----------	--------	------------

@P1	Matia 12, 1A	90000.00	3
-----	--------------	----------	---

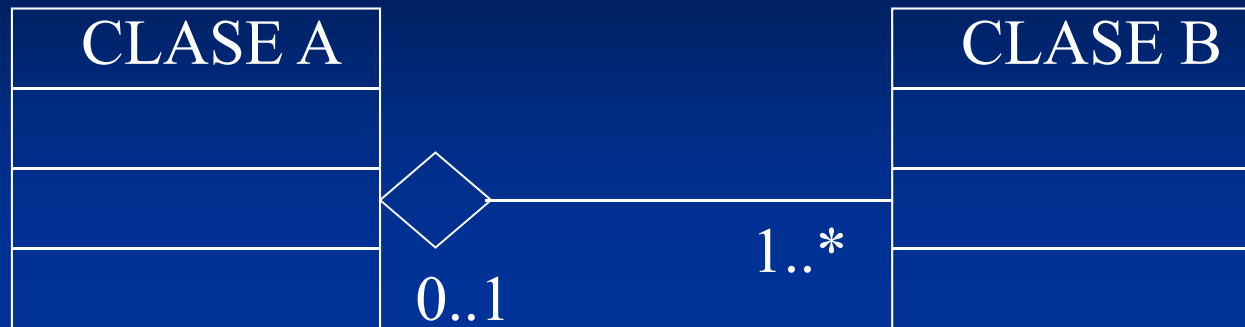
@P2	Hériz 1, 2A	85000.50	2
-----	-------------	----------	---

@G1	Hériz 5	15000.50	true
-----	---------	----------	------

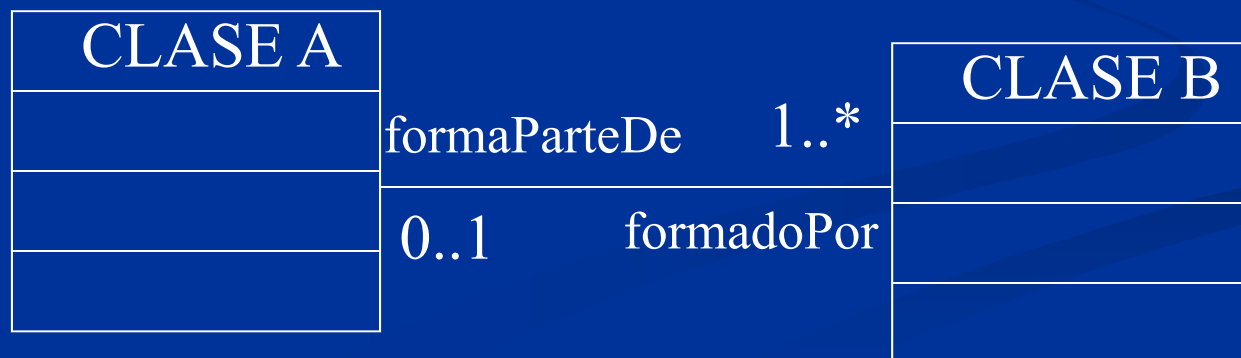


← ASOCIACIÓN

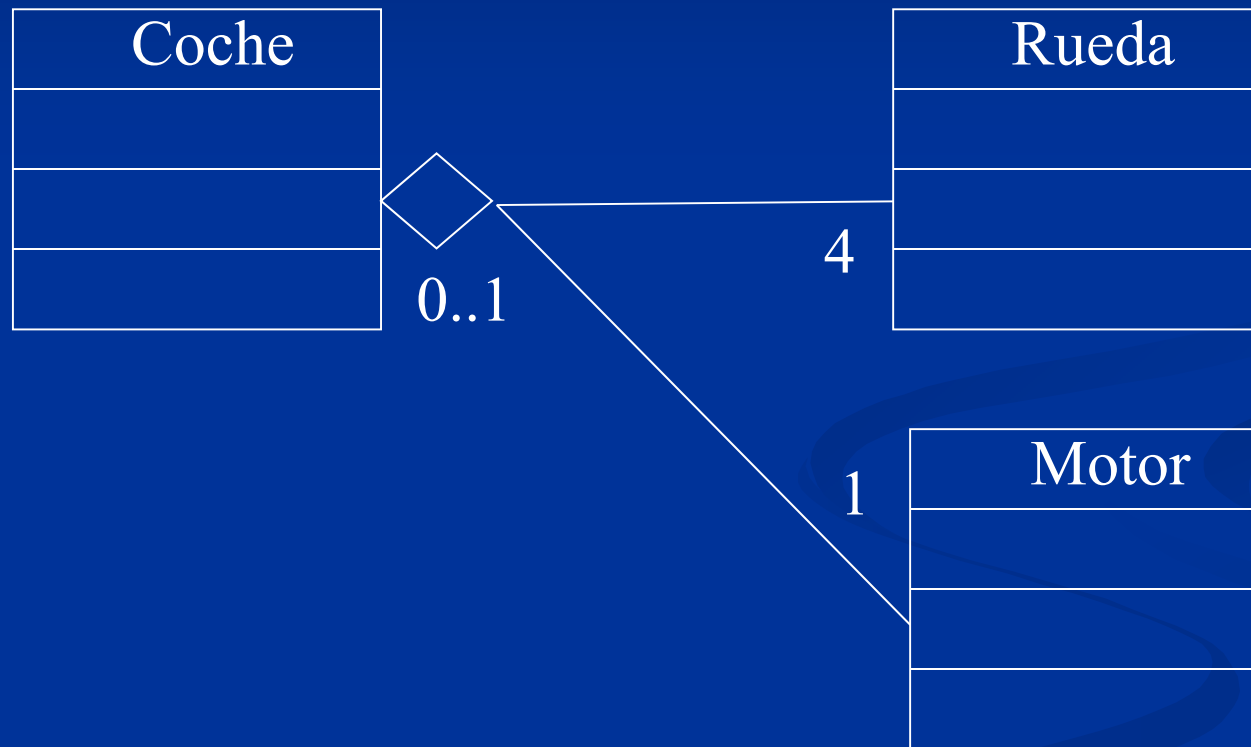
# Agregación en UML



**ES UNA ASOCIACIÓN CON LA SEMÁNTICA  
“FORMADO POR/FORMA PARTE DE”**



# Ej. de agregación en UML





# Composición en UML

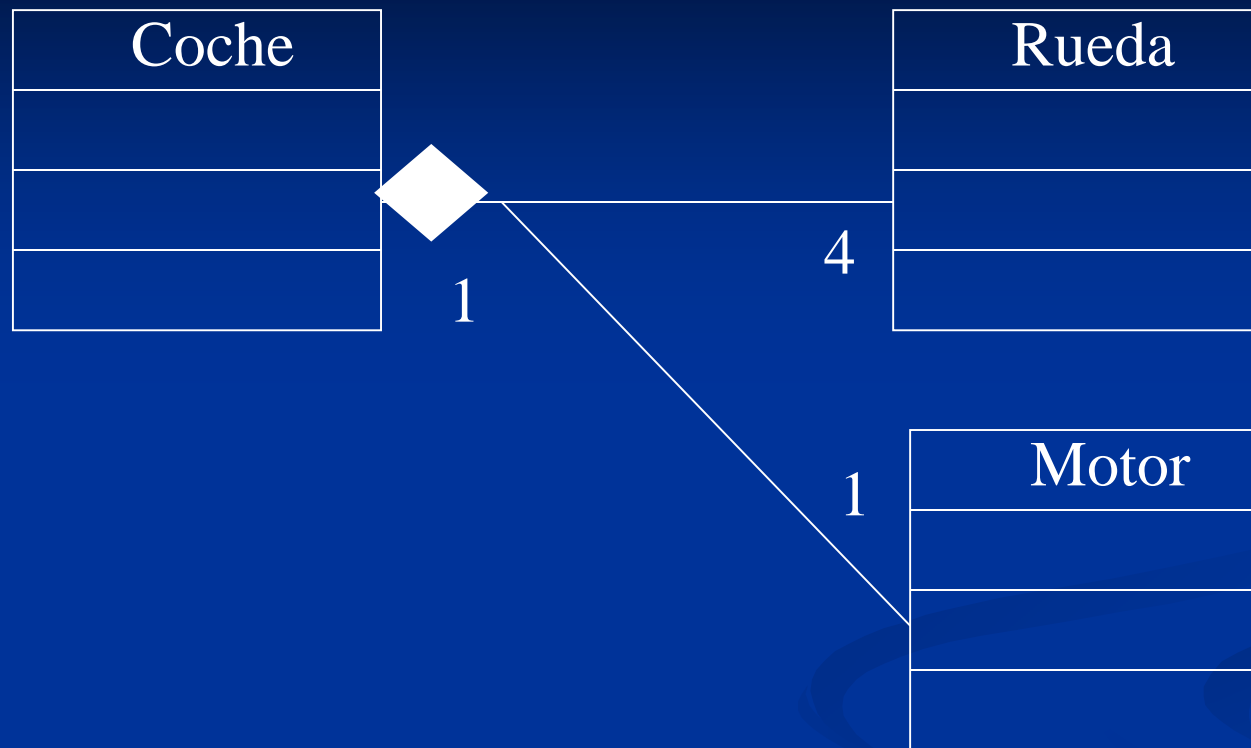


ES UNA ASOCIACIÓN CON LA SEMÁNTICA  
"COMPUESTO POR/ES COMPONENTE DE"

Si se borra el a, se borran los b



## Ej. de composición en UML



En este S.I. no se permite tener "motores" ni "ruedas" sueltos, y al borrar el coche, se borra todo... Seguro que no se trata de un desguace.

# Composición en UML

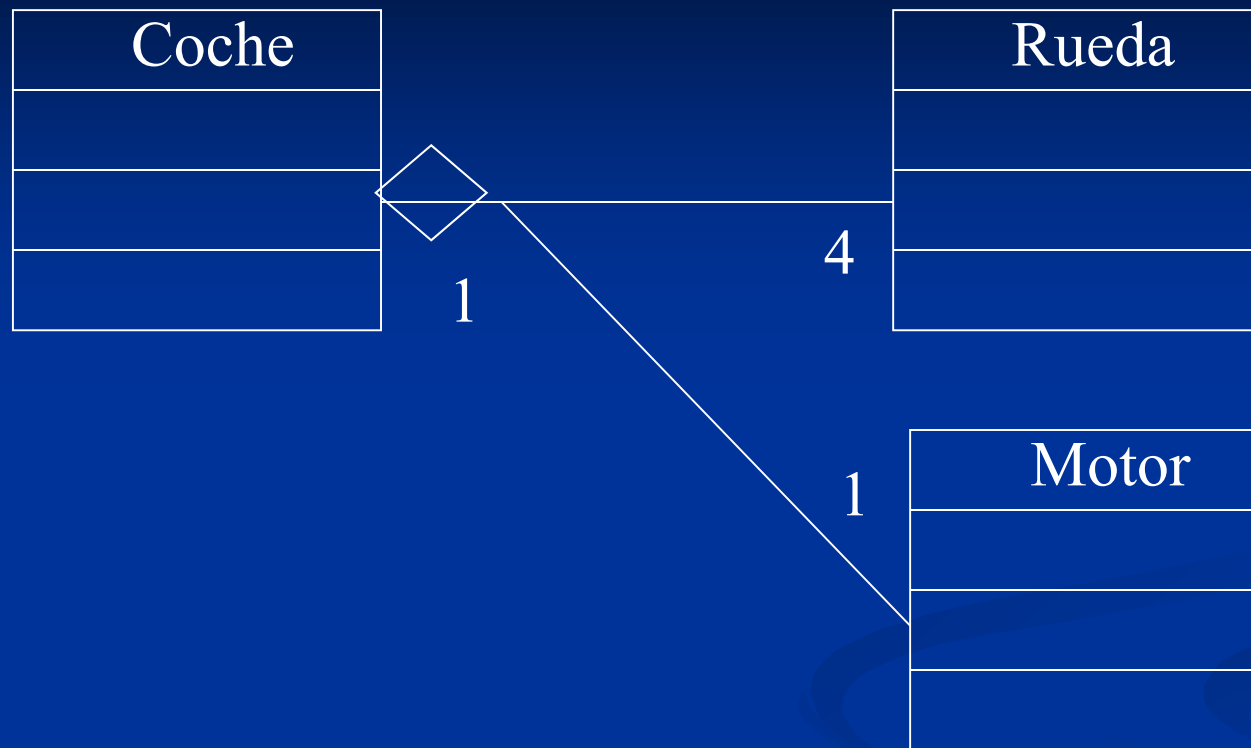


ES UNA ASOCIACIÓN CON LA SEMÁNTICA  
"COMPUESTO POR/ES COMPONENTE DE"

Si se borra el a, se borran los b

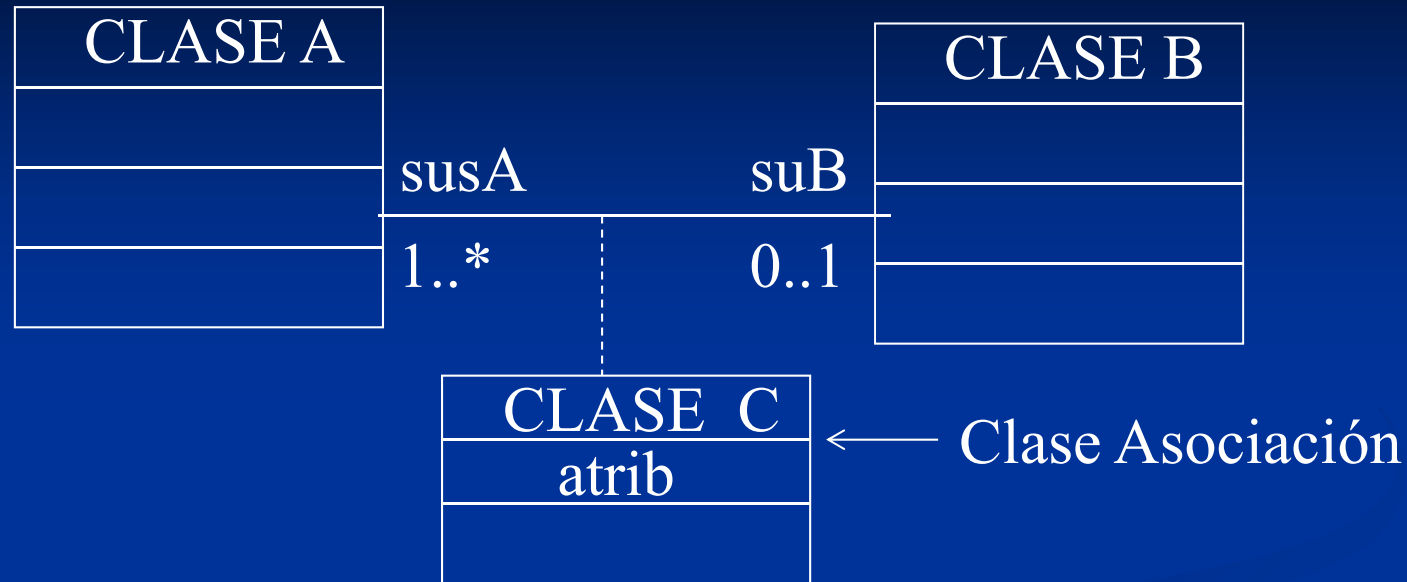


## Ej. de composición en UML

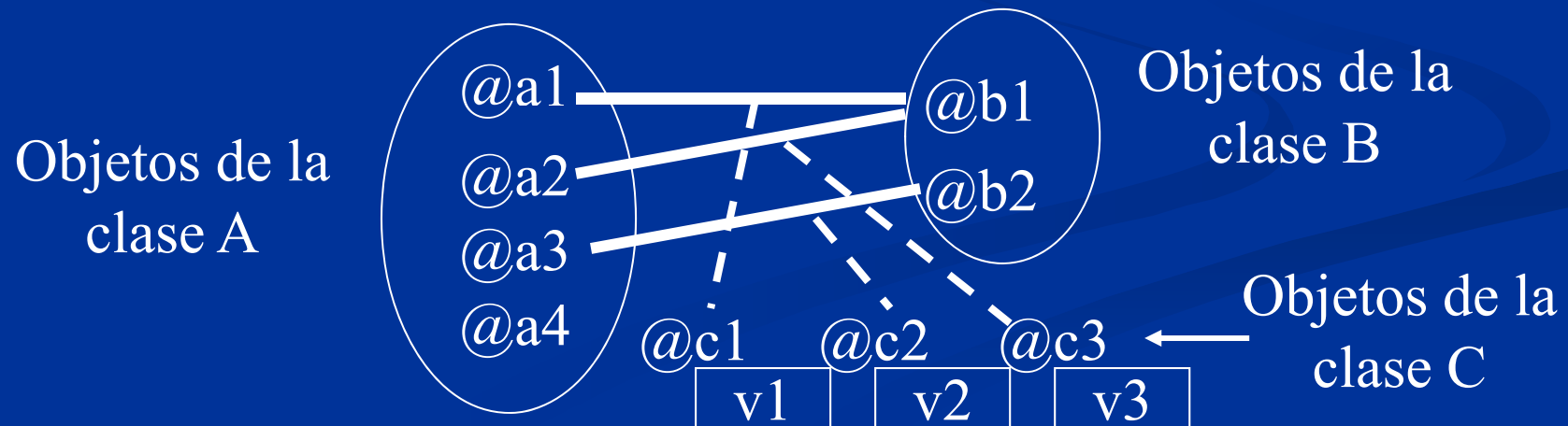


En este S.I. no se permite tener "motores" ni "ruedas" sueltos, y al borrar el coche, se borra todo... Seguro que no se trata de un desguace.

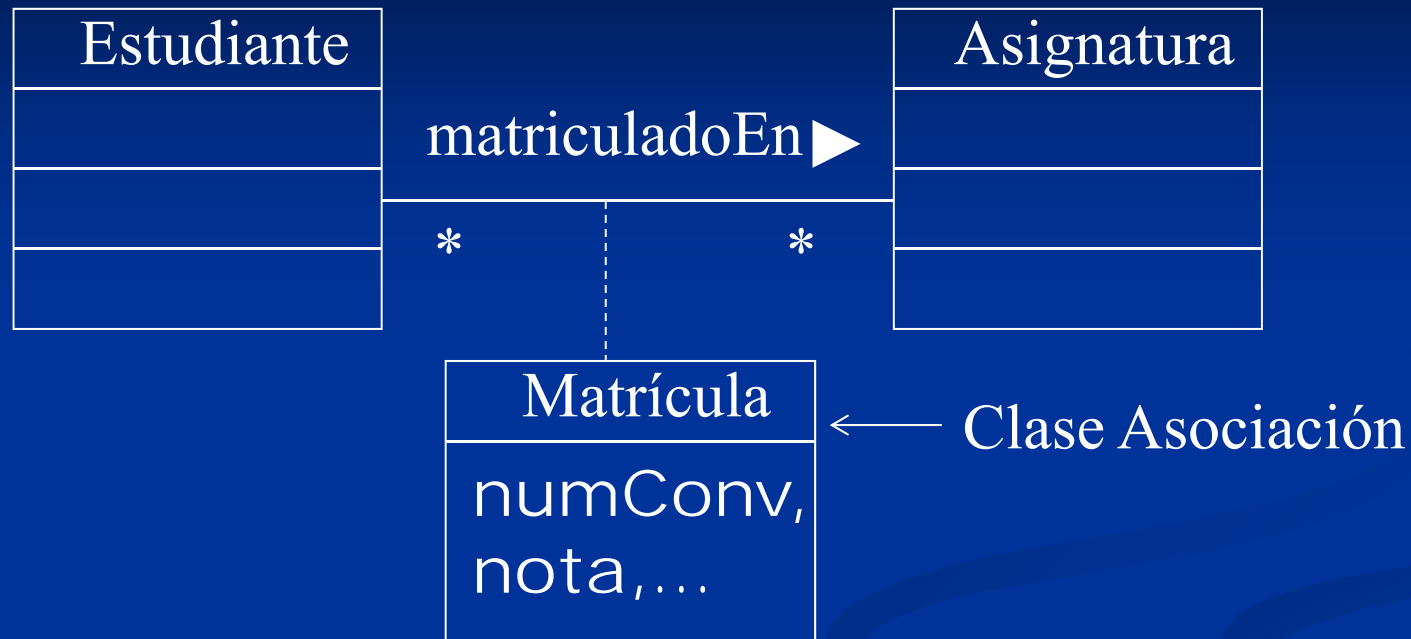
# Clase Asociación en UML



Para almacenar  
<objeto de A, objeto de B, Atrs. PROPIOS>



# Ej. de clase Asociación en UML

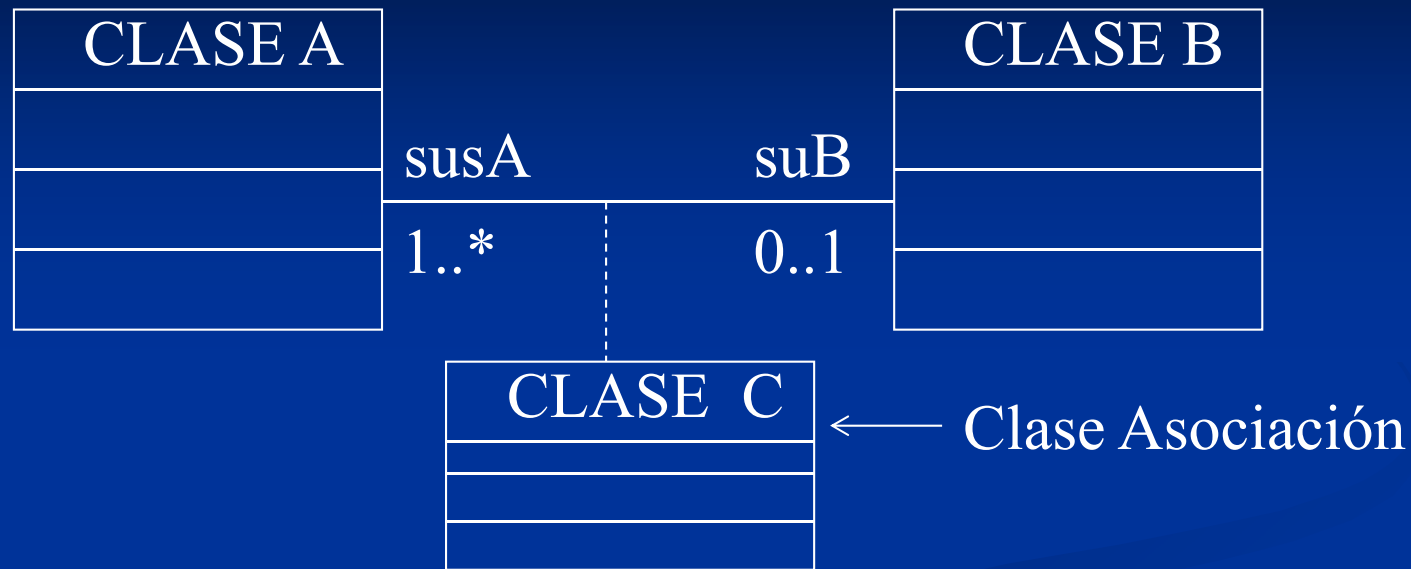


Si se desea poder almacenar el  
nº de convocatoria, nota  
obtenida, curso académico, etc.

# El MCU (Modelo de Casos de Uso) y el MD (Modelo del Dominio) deben ser COHERENTES

- Los DATOS que aparecen en todos los FLUJOS DE EVENTOS del MCU deben estar recogidos en el MD y ser accesibles en cada momento.
- Todas las clases, atributos y asociaciones que aparecen en el MD deben aparecer en alguno de los FLUJOS de EVENTOS del MCU.

# Clase Asociación en UML



Cada objeto de C se refiere  
a un único objeto de A y a  
un único objeto de B

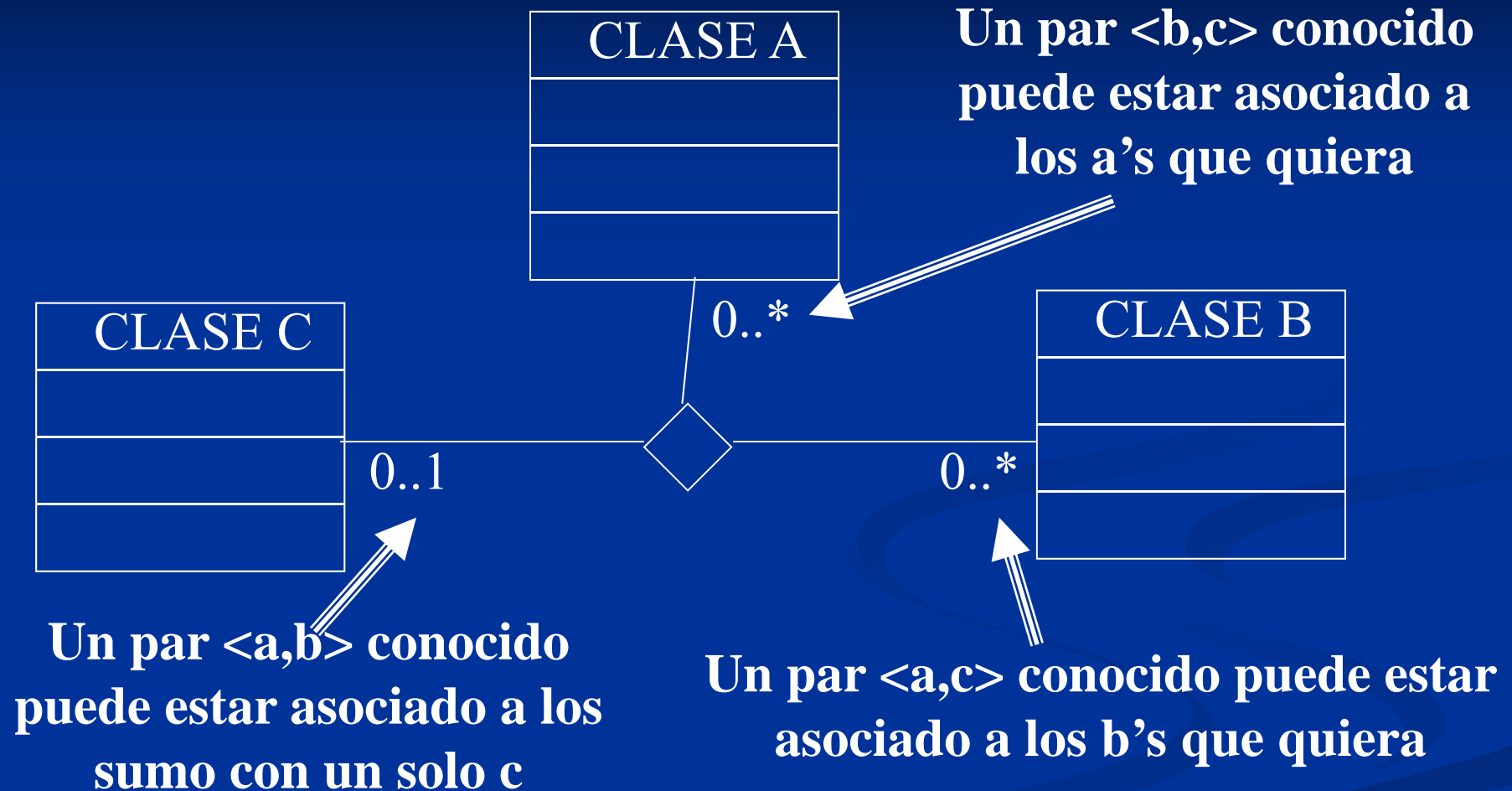


# Clase Asociación en UML



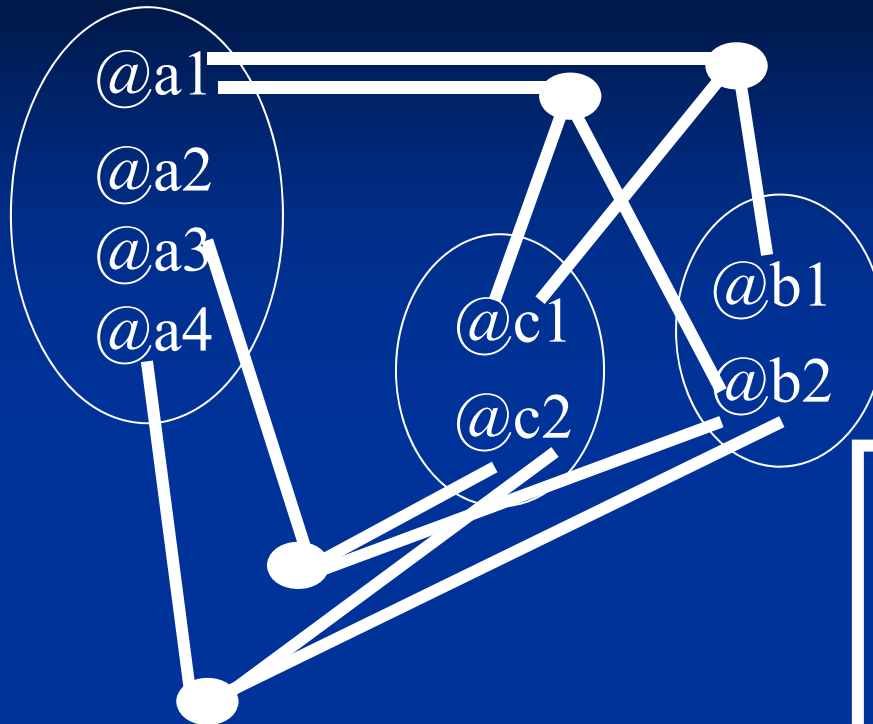
Si se quisiera que uno de C pudiera asociarse con varios de A y con 0 ó 1 de B entonces no se puede usar una clase asociación sino una clase (normal) y 2 asociaciones

# Asociación n-aria en UML



Fijados el resto de objetos que participan en la asociación, ¿con cuántos pueden relacionarse?

# Asociación n-aria en UML



<@a1,@c1,@b1>  
 <@a1,@c1,@b2>  
 <@a3,@c2,@b2>  
 <@a4,@c2,@b2>

cardinalidad 0..1 en el lado de C

<@a1,@b1> → @c1  
 <@a1,@b2> → @c1  
 <@a3,@b2> → @c2  
 <@a4,@b2> → @c2

<@a1,@c1> → @b1 y @b2  
 <@a3,@c2> → @b2  
 <@a4,@c2> → @b2

cardinalidad 0..\* en el lado de B

cardinalidad 0..\* en el lado de A

<@c1,@b1> → @a1  
 <@c1,@b2> → @a1  
 <@c2,@b2> → @a3 y @a4

# Ej. de asociación n-aria en UML

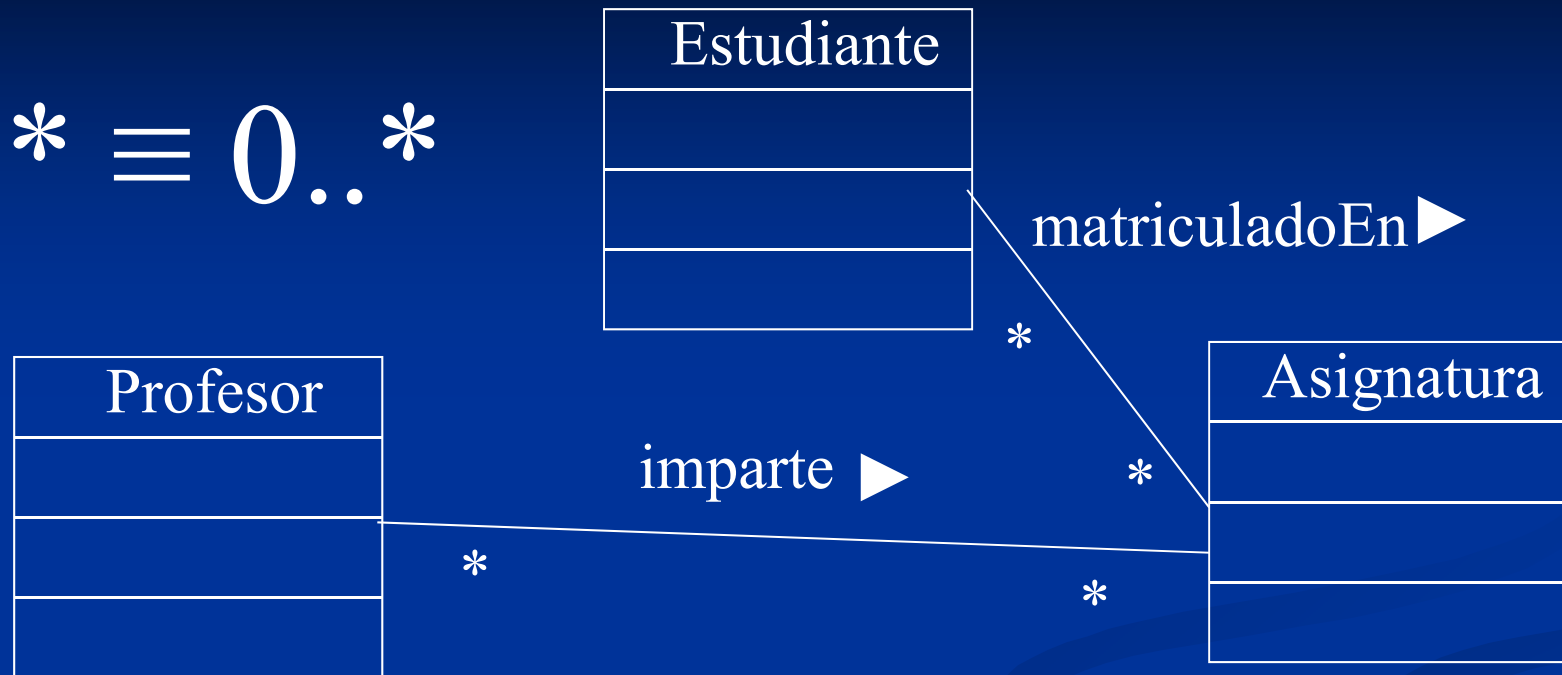


**Información sobre qué profesores imparten  
qué asignaturas a qué estudiantes**

HAY QUE ESTAR SEGUROS DE QUE SE  
NECESITA UNA ASOCIACIÓN TERNARIA

# Ej. de asociación n-aria en UML

$*$   $\equiv$   $0..*$



**Los estudiantes se matriculan en asignaturas**  
**Los profesores imparten asignaturas**

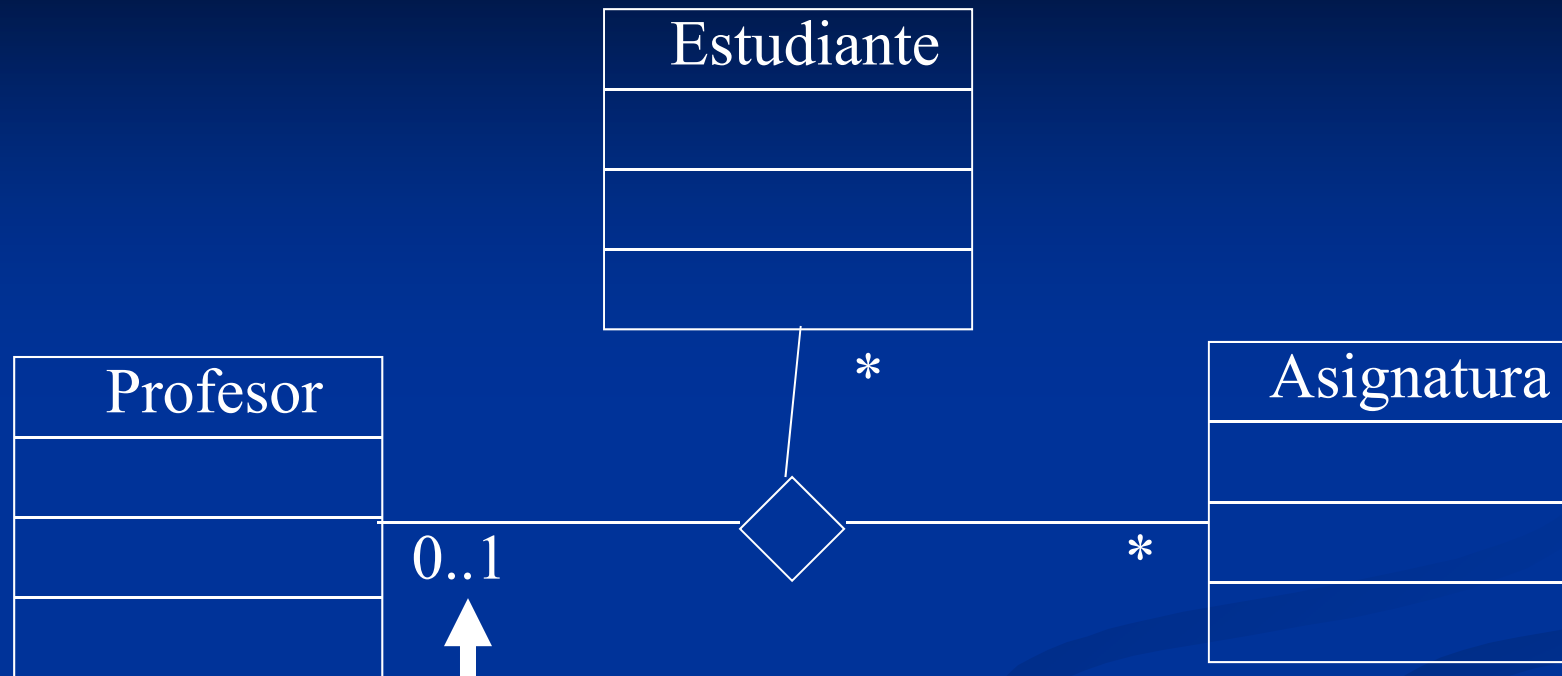
ASOCIACIÓN TERNARIA SÓLO SI HAY  
QUE DISTINGUIR CON QUÉ  
PROFESOR/ES SE HA MATRICULADO

# Ej. de asociación n-aria en UML



**Los estudiantes se matriculan en asignaturas.  
Los profesores imparten asignaturas.  
Cuando un estudiante se matricula en una  
asignatura, NO todos los profesores que la  
imparten son sus profesores.**

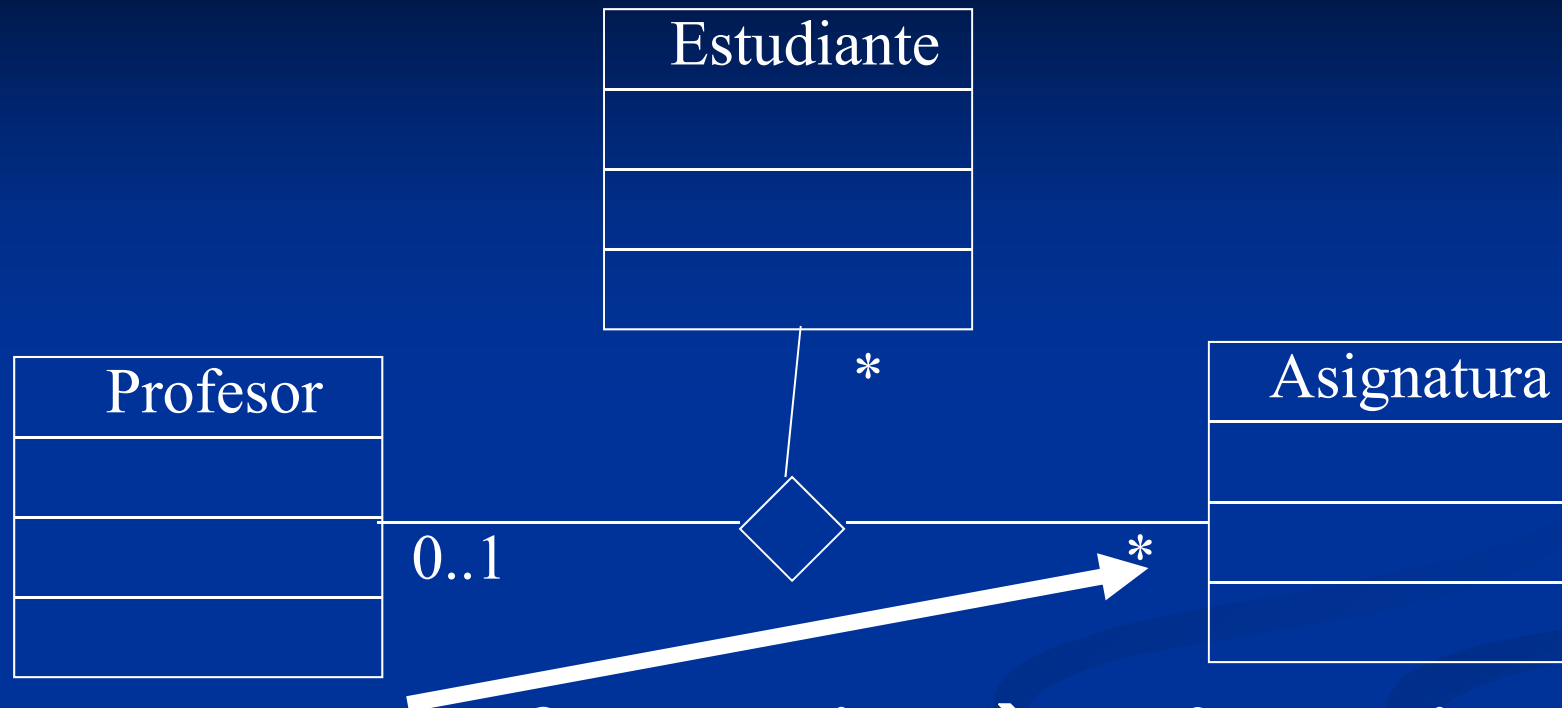
# Ej. de asociación n-aria en UML



par <est,asig> conocido → con 0 ó 1 prof.

**Un estudiante se puede matricular en una asignatura SÓLO CON UNO DE LOS PROFESORES QUE LA IMPARTE, o no matricularse, claro.**

# Ej. de asociación n-aria en UML

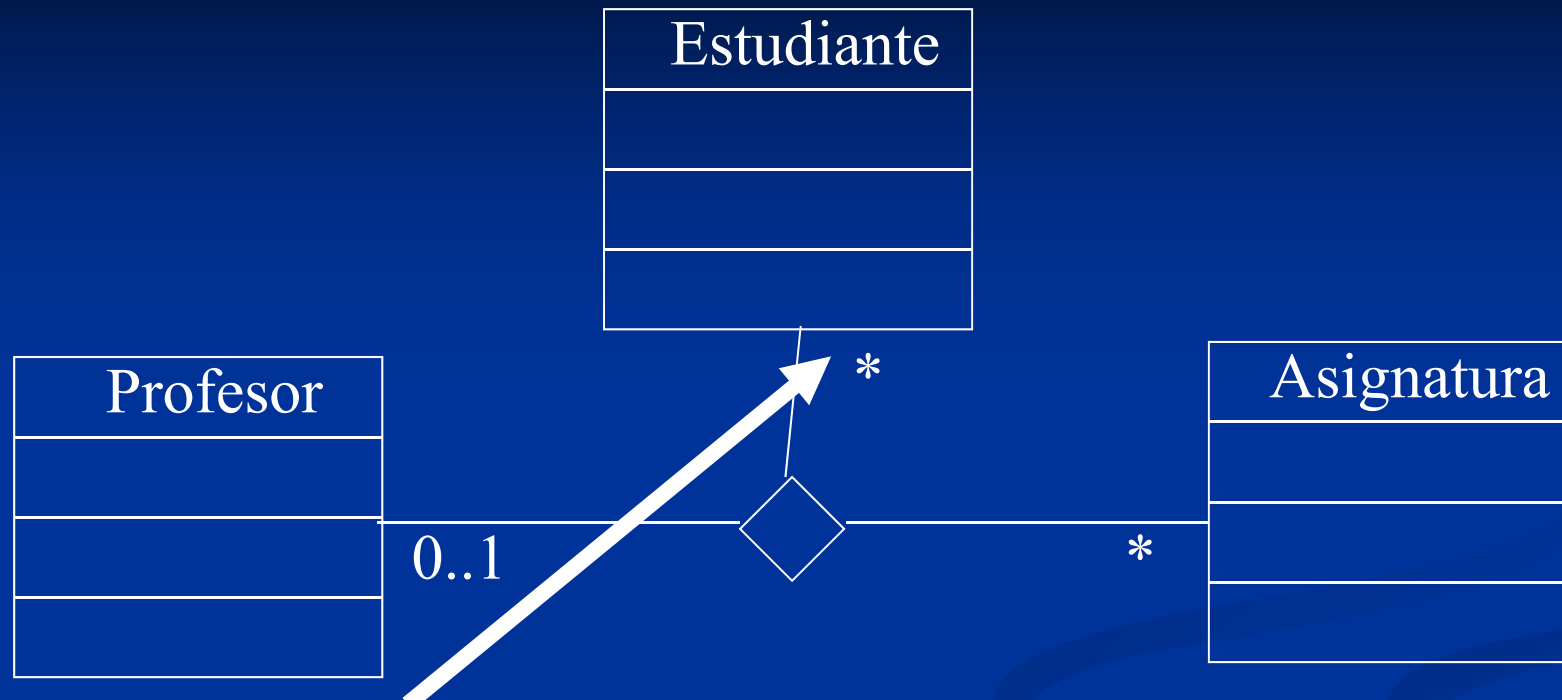


par  $\langle \text{est}, \text{prof} \rangle$  conocido  $\rightarrow$  en 0 o varias asig

**Un estudiante se puede matricular con el mismo profesor en DISTINTAS asignaturas o puede que no le corresponda ese profesor.**



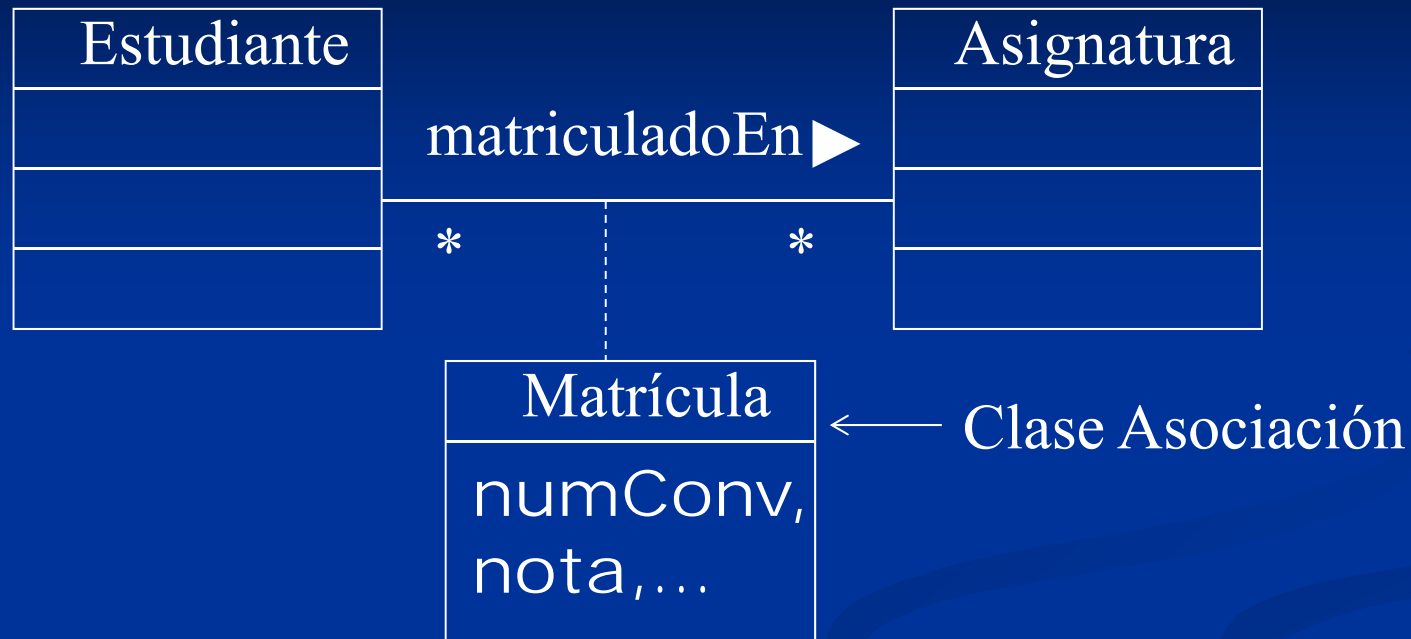
# Ej. de asociación n-aria en UML



par <prof,asig> conocido → con 0 ó varios est

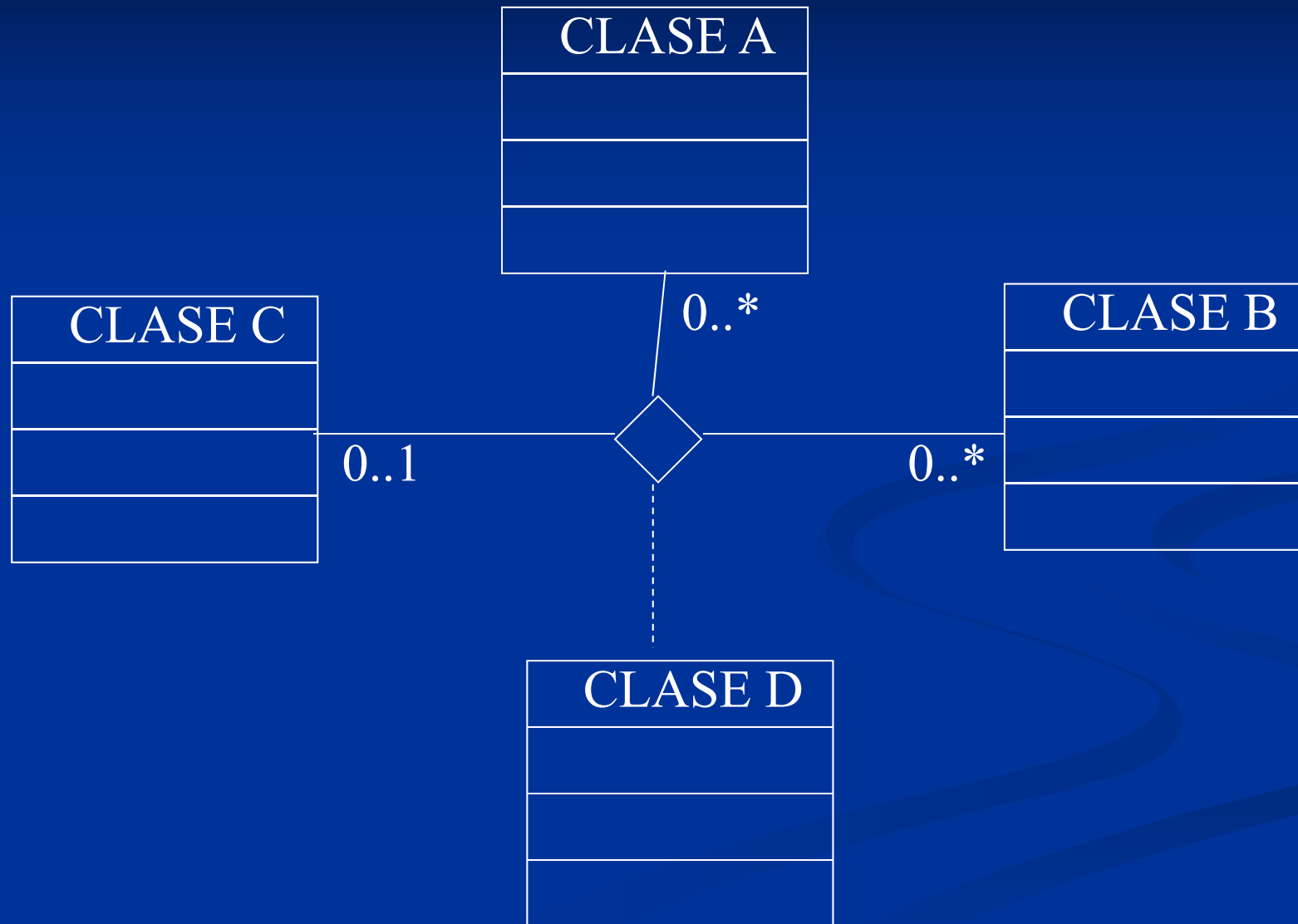
Un profesor puede impartir o no una asignatura, y si la imparte, entonces puede tener **VARIOS** estudiantes

# Ej. de clase Asociación en UML

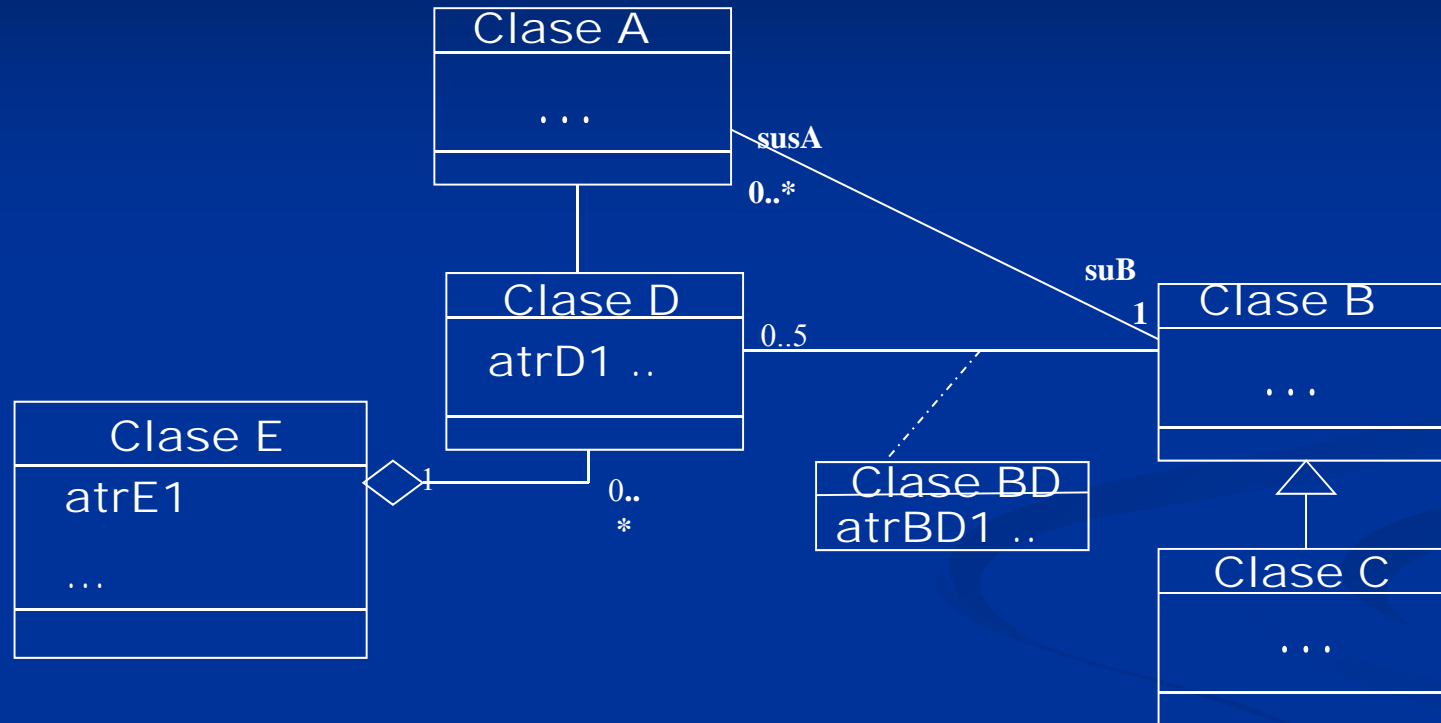


Si se desea poder almacenar el  
nº de convocatoria, nota  
obtenida, curso académico, etc.

# Clase asociación n-aria en UML



# Diagrama de clases en UML



Durante la captura de requisitos se utiliza para representar el MODELO DEL DOMINIO. De momento, sólo interesan los ATRIBUTOS de las clases y NO SUS MÉTODOS

# Artefactos a conseguir en la captura de requisitos

- Un artefacto es cualquier clase de descripción o de información (modelo, elemento de un modelo o documento) del sistema

Actores



Casos de Uso



Prototipos de Interfaces de  
Usuario



Glosario



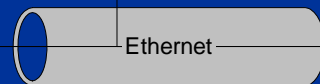
Diagramas de clase  
(Modelo del Dominio)



Descripción de la arquitectura

# Arquitectura Preliminar y Casos de Uso Críticos

Ordenadores de los bibliotecarios con lectores de códigos de barras



Lenguaje de programación visual (para construir interfaces gráficas)



Terminales para consultar el catálogo



Máquina servidora

- usada por el encargado
- incluye un SGBD y la BD

## CASOS DE USO CRÍTICOS:

Consultar Catálogo

Actualizar Catálogo

Tomar Préstamo Copia Libro

Devolver Copia Libro

Reservar Libro