Relaciones entre clases: Diagramas de clases UML

Las relaciones existentes entre las distintas clases
nos indican cómo se comunican los objetos de esas clases entre sí:

Los mensajes “navegan”
por las relaciones existentes entre las distintas clases.

Existen distintos tipos de relaciones:

- **Asociación** (conexión entre clases)
- **Dependencia** (relación de uso)
- **Generalización/especialización** (relaciones de herencia)

Asociación

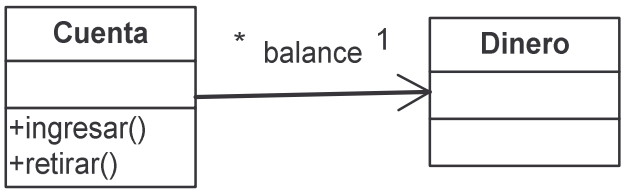
Una asociación es una relación estructural que describe una conexión entre objetos.

Gráficamente, se muestra como una línea continua que une las clases relacionadas entre sí.

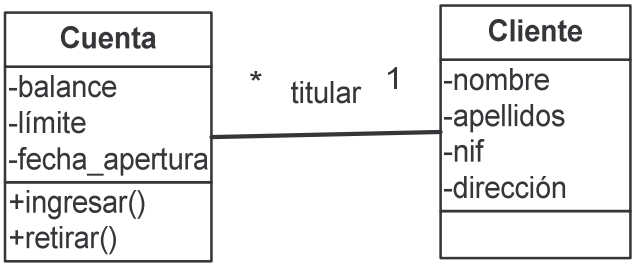
Navegación de las asociaciones

Aunque las asociaciones suelen ser bidireccionales (se pueden recorrer en ambos sentidos), en ocasiones es deseable hacerlas unidireccionales (restringir su navegación en un único sentido).

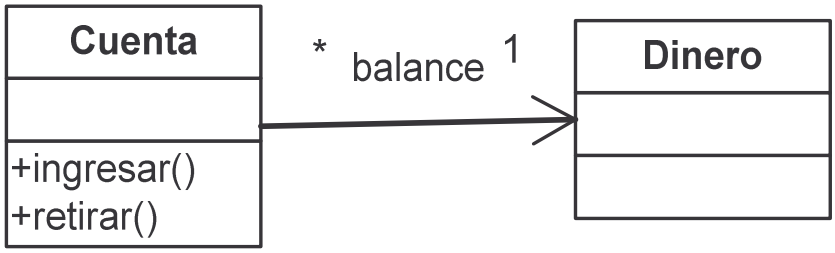
Gráficamente, cuando la asociación es unidireccional, la línea termina en una punta de flecha que indica el sentido de la asociación:



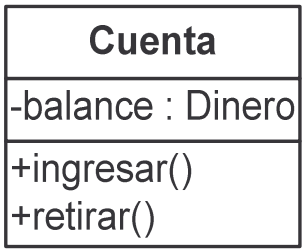
Asociación unidireccional



Asociación bidireccional

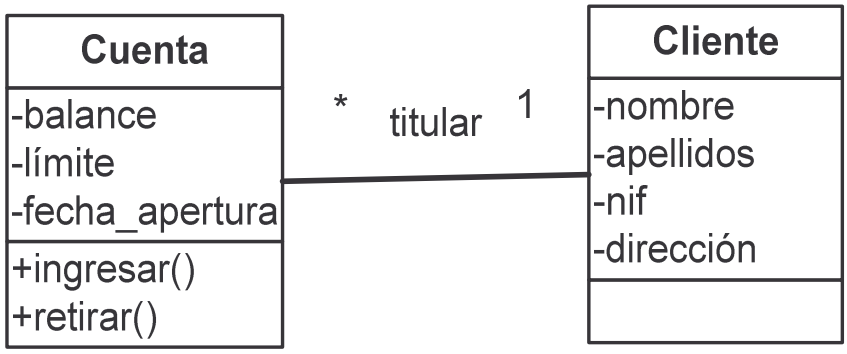


equivale a

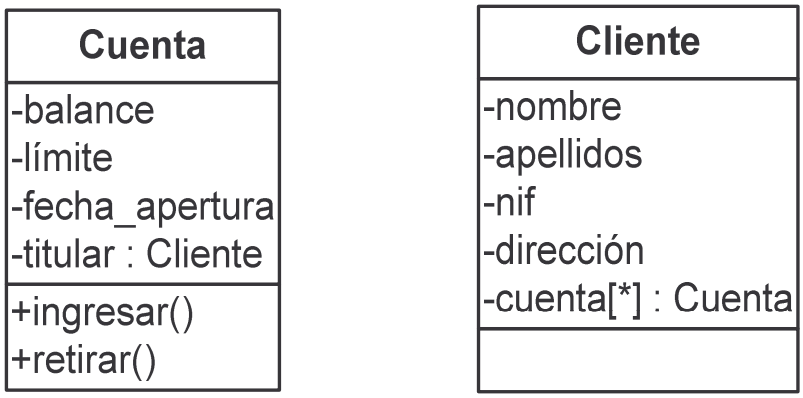



```
class Cuenta
{
  private Dinero balance;

  public void ingresar (Dinero cantidad)
  {
    balance += cantidad;
  }

  public void retirar (Dinero cantidad)
  {
    balance -= cantidad;
  }

  public Dinero getSaldo ()
  {
    return balance;
  }
}
```

Hemos supuesto que `Dinero` es un tipo de dato con el que se pueden hacer operaciones aritméticas y hemos añadido un método adicional que nos permite comprobar el saldo de una cuenta.



viene a ser lo mismo que



con la salvedad de
que el enlace bidireccional hemos de mantenerlo nosotros

```
public class Cuenta
{
  …
  private Cliente titular;
  …
}

public class Cliente
{
  …
  private Cuenta cuenta[];
  …
}
```

Un cliente puede tener varias cuentas, por lo que en la clase cliente hemos de mantener un conjunto de cuentas (un vector en este caso).

Multiplicidad de las asociaciones

La multiplicidad de una asociación determina
cuántos objetos de cada tipo intervienen en la relación:

> El número de instancias de una clase que se relacionan
> con UNA instancia de la otra clase.

- Cada asociación tiene dos multiplicidades
  (una para cada extremo de la relación).

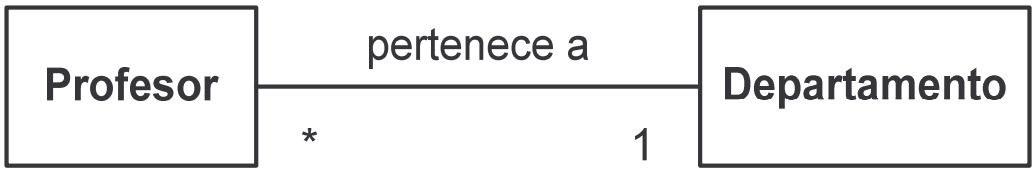
- Para especificar la multiplicidad de una asociación hay que indicar la multiplicidad mínima y la multiplicidad máxima (mínima..máxima)

Multiplicidad	Significado
`1`	Uno y sólo uno
`0..1`	Cero o uno
`N..M`	Desde N hasta M
`*`	Cero o varios
`0..*`	Cero o varios
`1..*`	Uno o varios (al menos uno)

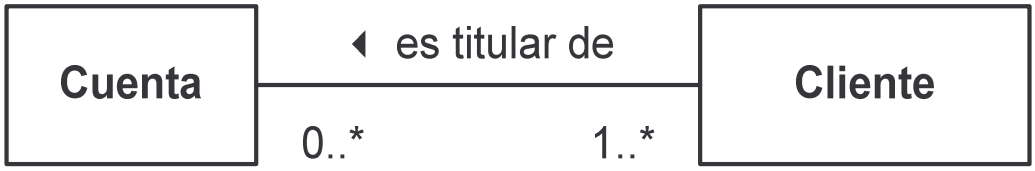
- Cuando la multiplicidad mínima es 0, la relación es opcional.

- Una multiplicidad mínima mayor o igual que 1 establece una relación obligatoria.

Todo departamento tiene un director.
Un profesor puede dirigir un departamento.



Todo profesor pertenece a un departamento.
A un departamento pueden pertenecer varios profesores.



Relación opcional
Un cliente puede o no ser titular de una cuenta

Relación obligatoria
Una cuenta ha de tener un titular como mínimo

Relaciones involutivas

Cuando la misma clase aparece en los dos extremos de la asociación.

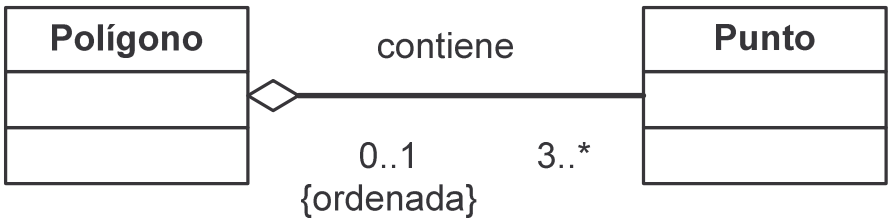
Agregación y composición

Casos particulares de asociaciones:
Relación entre un todo y sus partes

Gráficamente,
se muestran como asociaciones con un rombo en uno de los extremos.

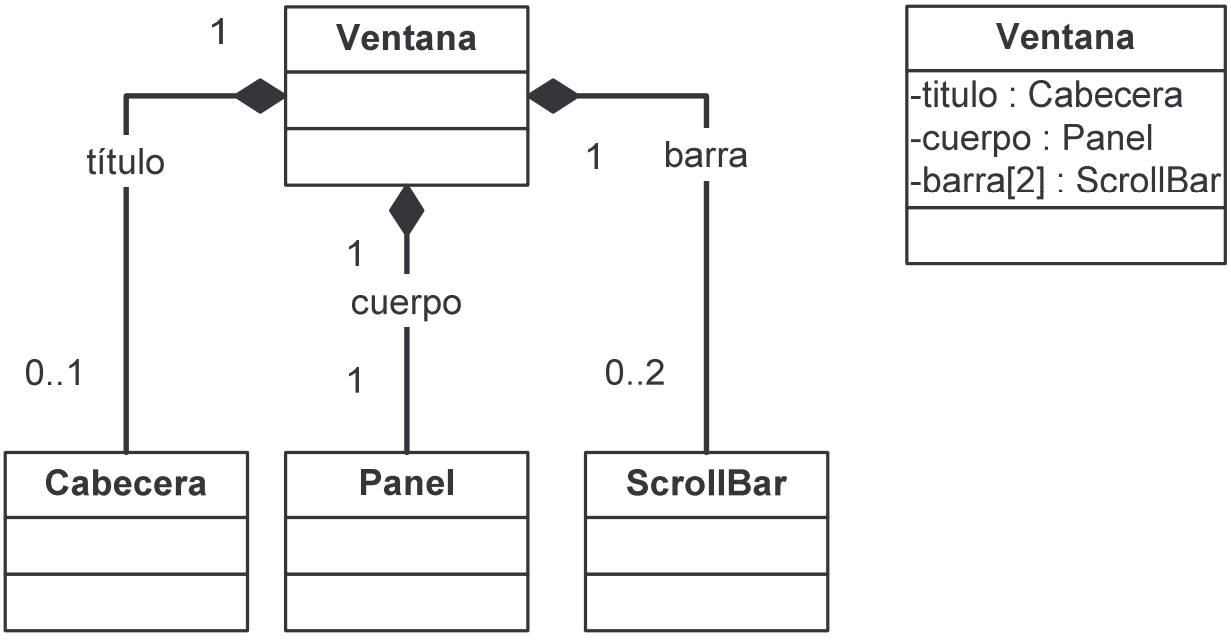
Agregación

Las partes pueden formar parte de distintos agregados.



Composición

Agregación disjunta y estricta:
Las partes sólo existen asociadas al compuesto
(sólo se accede a ellas a través del compuesto)



Dependencia

Relación (más débil que una asociación) que muestra la relación entre un cliente y el proveedor de un servicio usado por el cliente.

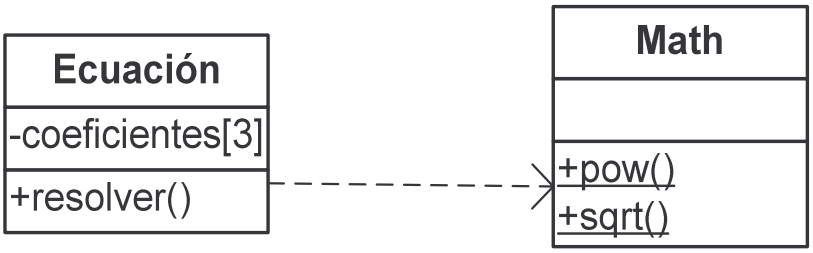
- Cliente es el objeto que solicita un servicio.

- Servidor es el objeto que provee el servicio solicitado.

Gráficamente, la dependencia se muestra como una línea discontinua con una punta de flecha que apunta del cliente al proveedor.

Ejemplo

Resolución de una ecuación de segundo grado



$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Para resolver una ecuación de segundo grado hemos de recurrir a la función `sqrt` de la clase `Math` para calcular una raíz cuadrada.

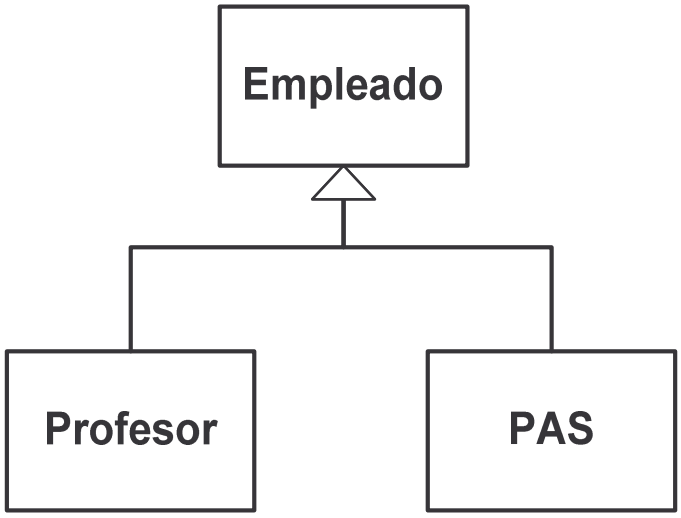
NOTA:

La clase `Math` es una clase "degenerada" que no tiene estado. Es, simplemente, una colección de funciones de cálculo matemático.

Herencia (generalización y especialización)

La relación entre una superclase y sus subclases

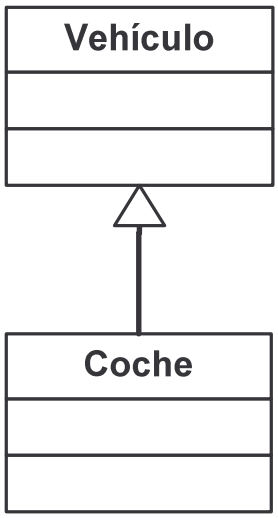
Objetos de distintas clases pueden tener atributos similares y exhibir comportamientos parecidos (p.ej. animales, mamíferos…).




```
public class Empleado
{
   ...
}

public class Profesor extends Empleado
{
   ...
}

public class PAS extends Empleado
{
   ...
}
```

La noción de clase está próxima a la de conjunto:

Generalización y especialización
expresan relaciones de inclusión entre conjuntos.



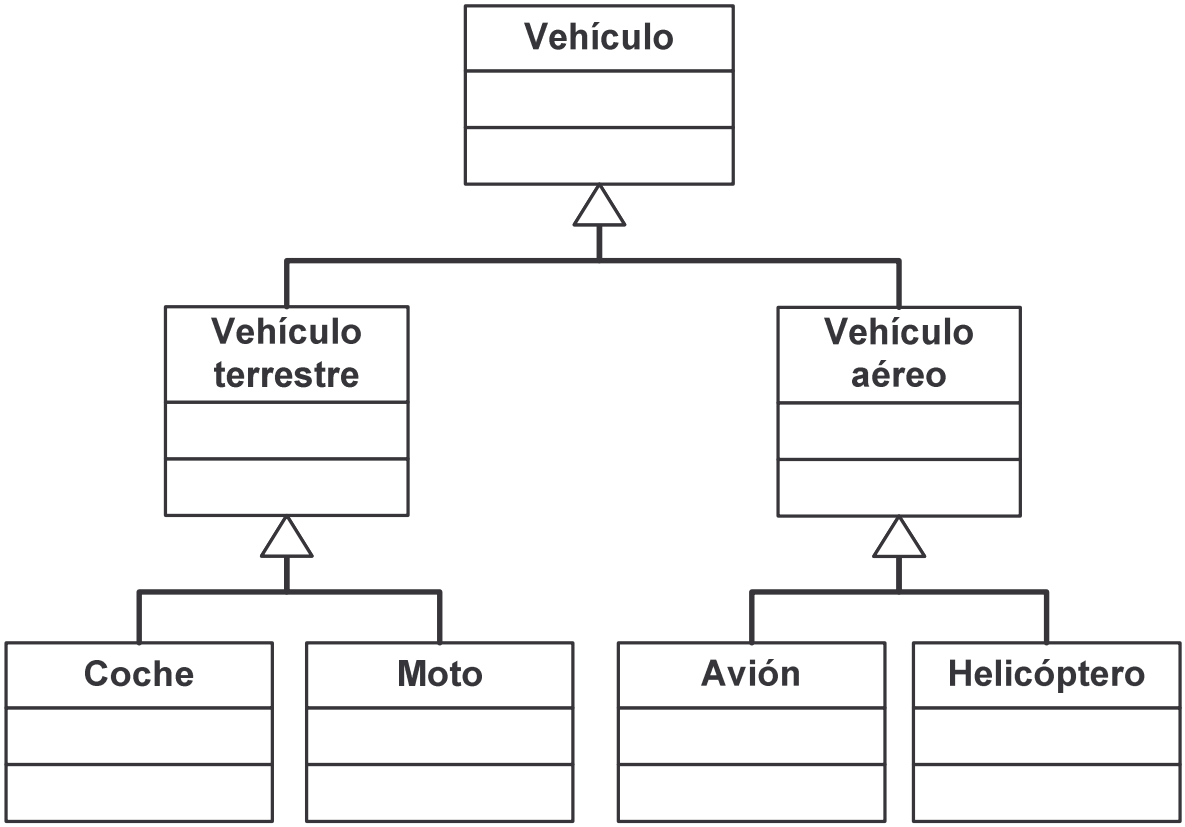
Instancias: **coches ⊂ vehículos**

- Todo coche es un vehículo.
- Algunos vehículos son coches.

Propiedades: **propiedades(coches) ⊃ propiedades(vehículos)**

- Un coche tiene todas las propiedades de un vehículo.
- Algunas propiedades del coche no las tienen todos los vehículos.

Jerarquías de clases



Las clases se organizan
en una estructura jerárquica formando una taxonomía.

- El comportamiento de una categoría más general es aplicable a una categoría particular.

- Las subclases heredan características de las clases de las que se derivan y añaden características específicas que las diferencian.

En el diagrama de clases,
los atributos, métodos y relaciones de una clase se muestran en el nivel más alto de la jerarquía en el que son aplicables.