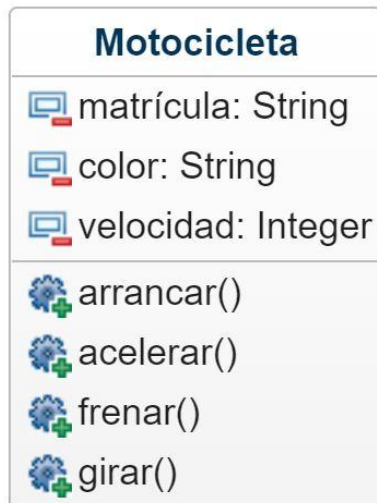
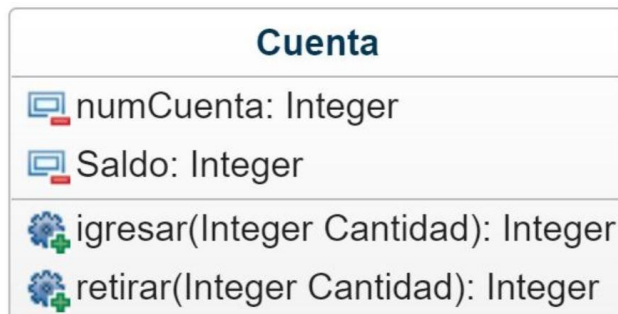


# Pas de UML a Java



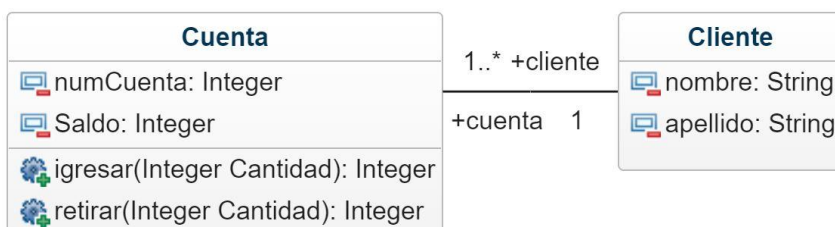
```
public class Motocicleta {
    private String matricula;
    private String color;
    private Integer velocidad;

    public void arrancar(){}
    public void acelerar(){}
    public void frenar(){}
    public void girar(){}
}
```



```
public class Cuenta {
    private Integer balance;
    private Integer limite;

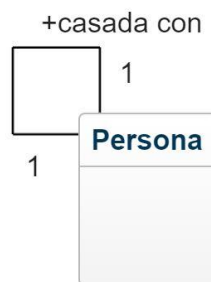
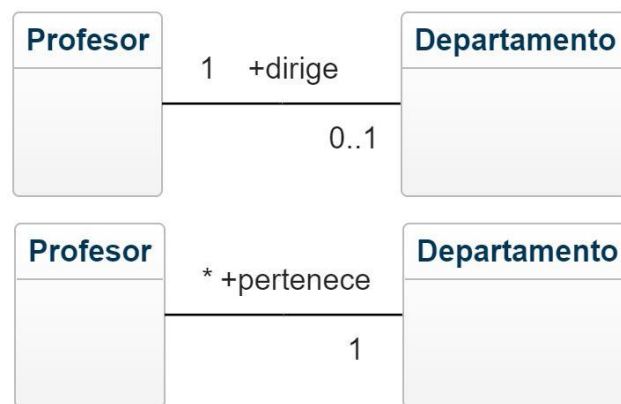
    public void ingresar(Integer cantidad){
        balance = balance + cantidad;
    }
    public void retirar(double cantidad){
        balance = balance - cantidad;
    }
}
```



```
public class Cuenta{
    ...
    private Cliente titular;
    ...
}
public class Cliente{
    ...
    private Cuenta cuentas[];
    ...
}
```

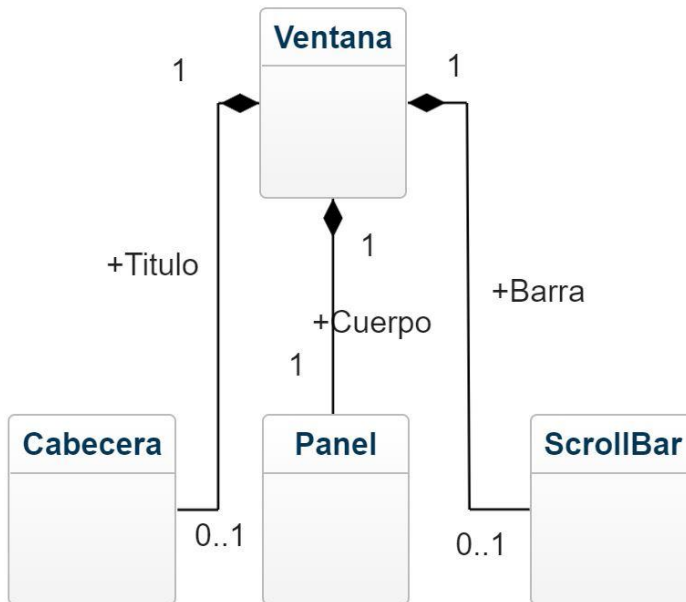
## Multiplicidad de las asociaciones

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)



## Composición

Las partes sólo existen asociadas al compuesto



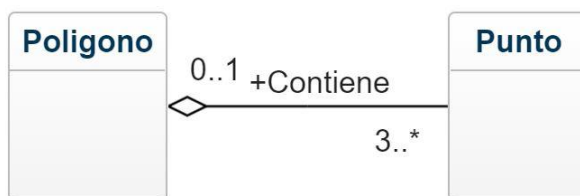
```

Public class Ventana{
    Private Cabecera cabecera;
    Private Panel panel;
    Private ScrollBar scrollBar;

    Public Ventana(){
        this.cabecera = new Cabecera();
        this.panel    = new Panel();
        this.scrollBar = new ScrollBar();
    }
}
    
```

## Agregación

Las partes pueden formar parte de distintos agregados.

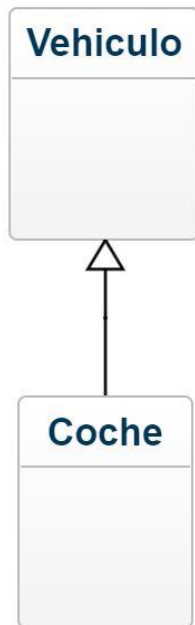


```

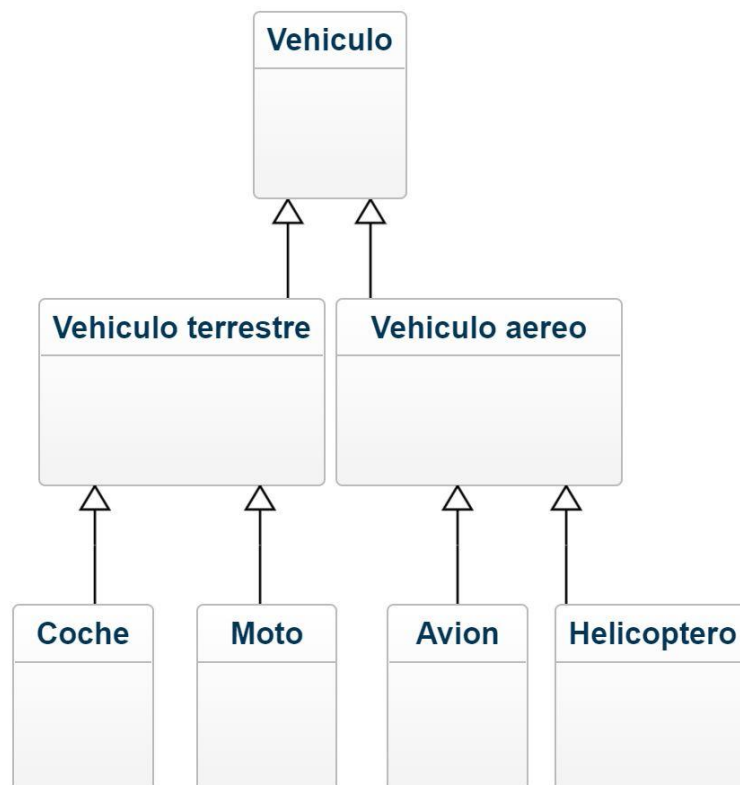
Public class Poligono{
    Private Punto puntos[];

    Public Poligono(){
    }
    Public void dibujar(Array puntos[]){
        this.puntos = puntos;
    }
}
    
```

## Herencia (Generalización / Especialización)

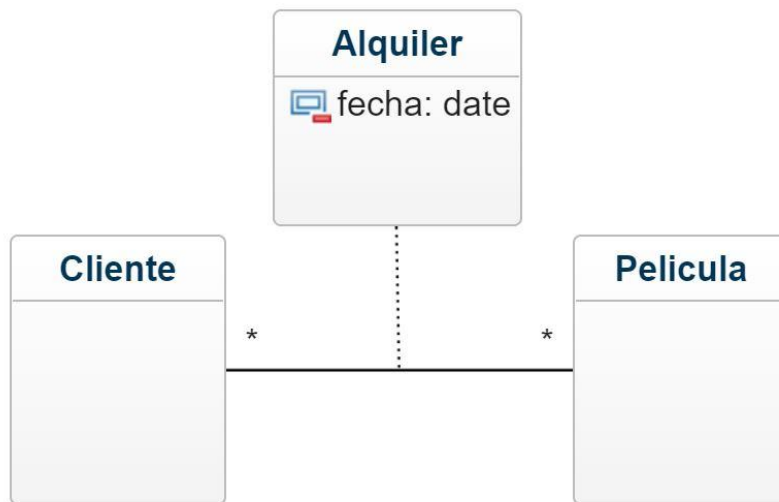


```
Public class Vehiculo(){
}
Public class Coche extends Vehiculo(){
}
```



### Clase asociación

La clase asociación se usa para indicar que la asociación existente entre dos clases tiene atributos propios.



```
Public class Cliente(){
...
}
Public class Pelicula(){
...
}
Public class Alquiler(){
    Private Cliente cliente;
    Private Pelicula pelicula;
    Private Date fecha;

    Public Alquiler(Cliente cliente, Pelicula pelicula, Date fecha){
        this.cliente = cliente;
        this.pelicula = pelicula;
        this.fecha = fecha;
    }
...
}
```