

Programación Orientada a Objetos (POO)

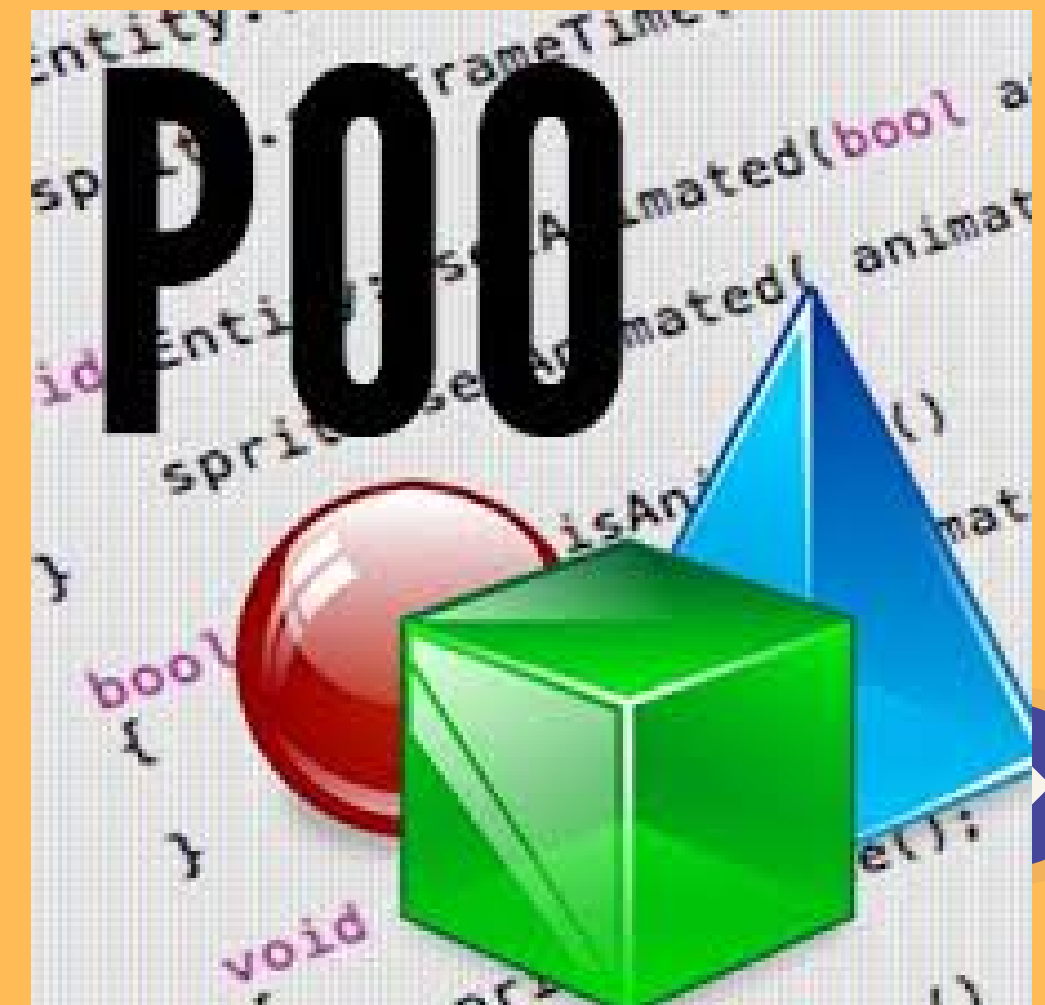
Pontificia Universidad Javeriana
Gabriel Gómez Corredor

Características

- Objetos
- Representa las entidades como objetos
- Instancias

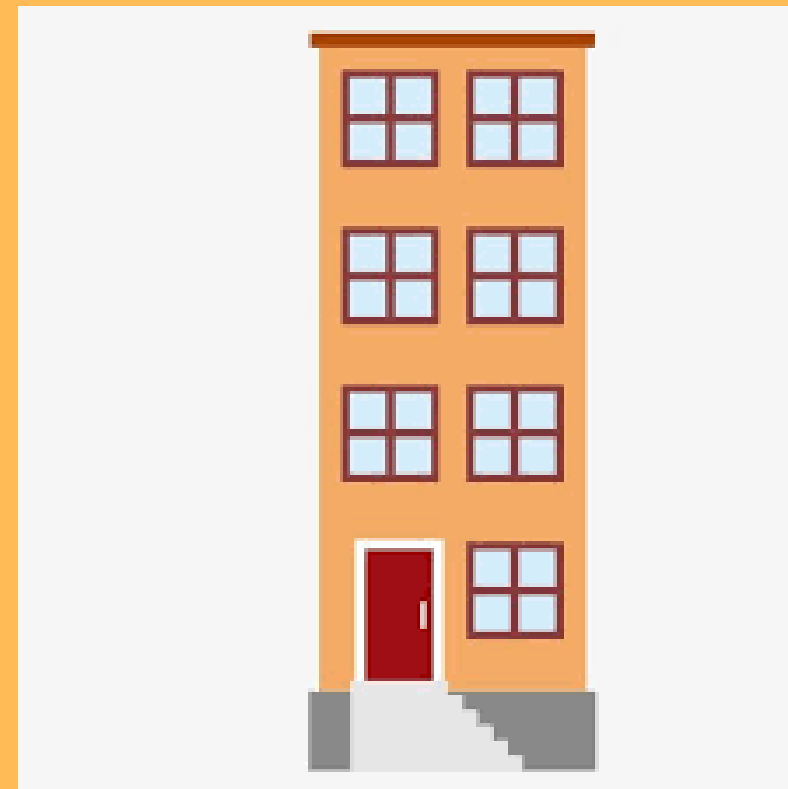
QUE ES ?

Es un paradigma de la programación que facilita la estructuración de los componentes en un software



Objetos

- En POO un objeto va a representar una entidad
- Tiene como función promover la comprensión del mundo real y será la base de la implementación

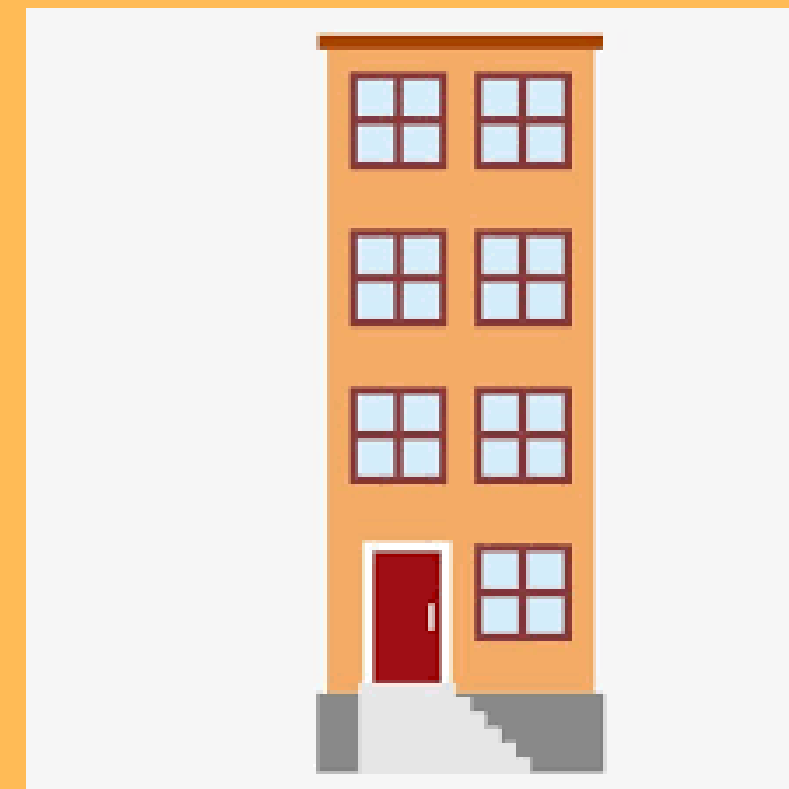


Atributos

- Propiedades del objeto
- Son los datos que tiene el objeto



- placa
- marca
- color
- modelo



- altura
- num_pisos
- cant_residentes



Comportamiento del objeto

El comportamiento del objeto se refiere a los servicios que este presta



Propiedades:

- digito1
- digito2

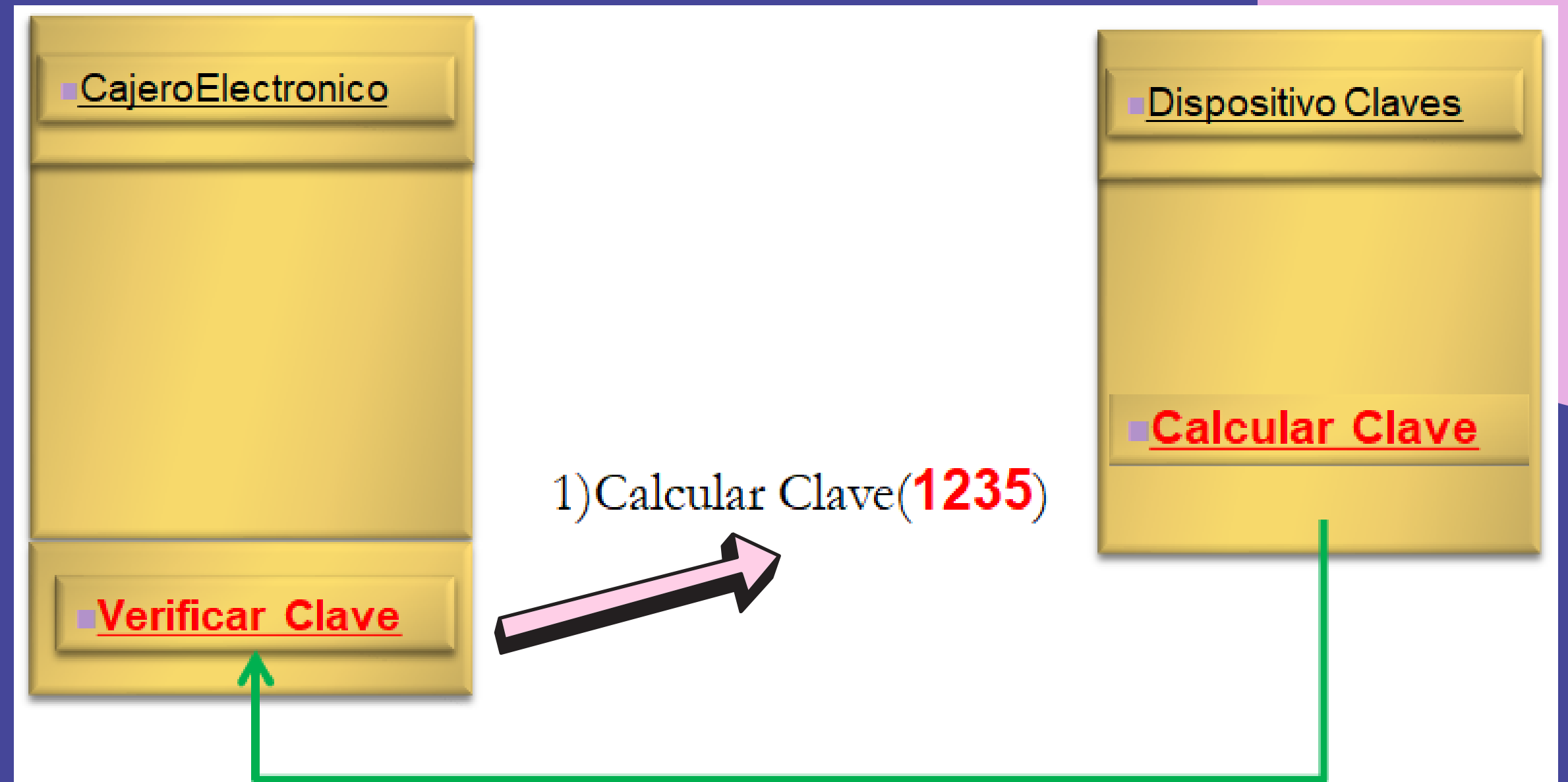
Metodos:

- Sumar()
- Restar()
- Multiplicar()
- Dividir()

Interacción entre objetos

Al enfrentarnos a un software real, nuestros objetos tendrán que interactuar entre ellos

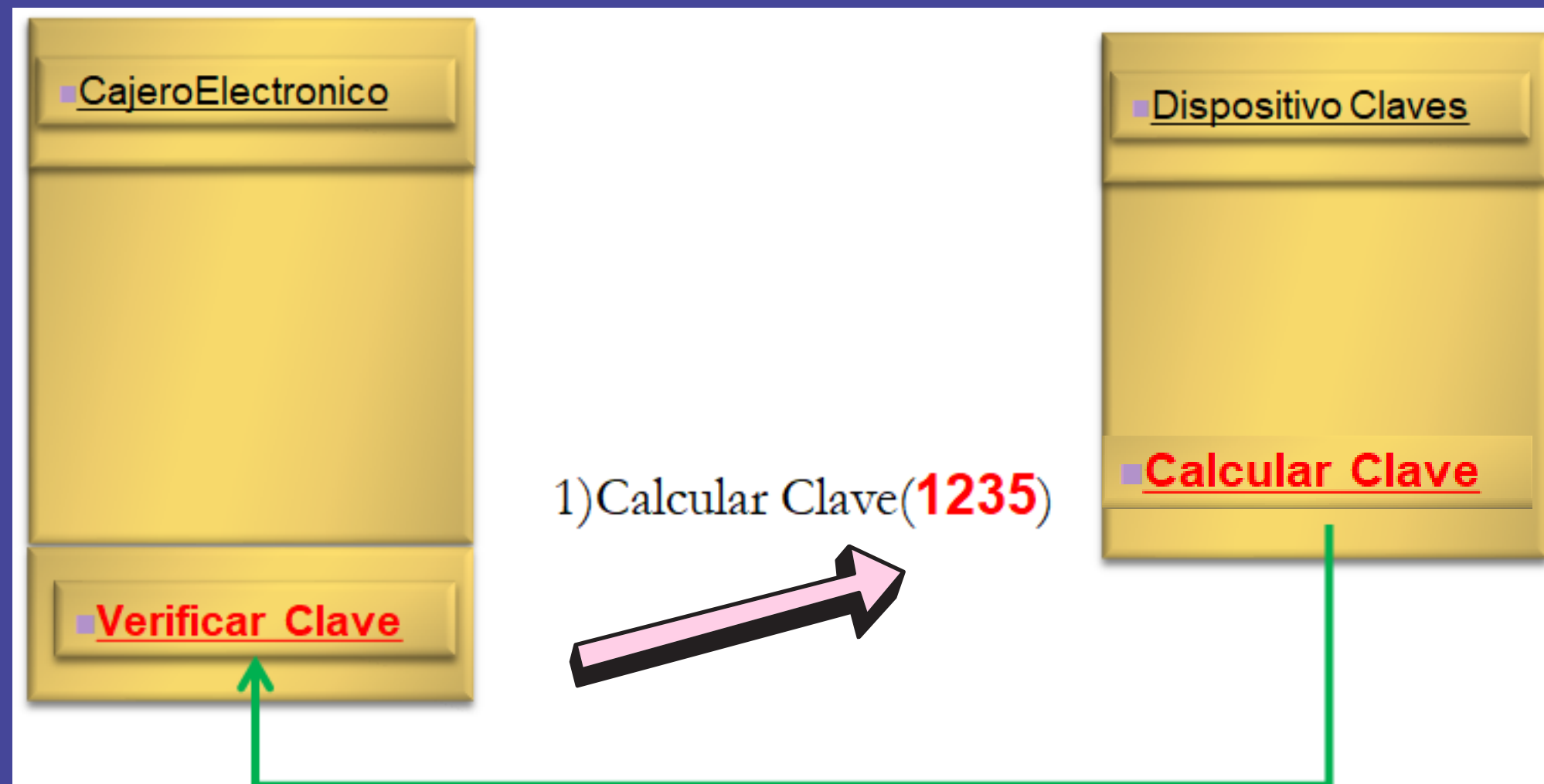
El objeto cajeroElectronico necesitará verificar su clave, para ello necesita del objeto Dispositivo Claves para que haga su labor



Mensajes

Para que un objeto realice una tarea se le debe enviar un mensaje

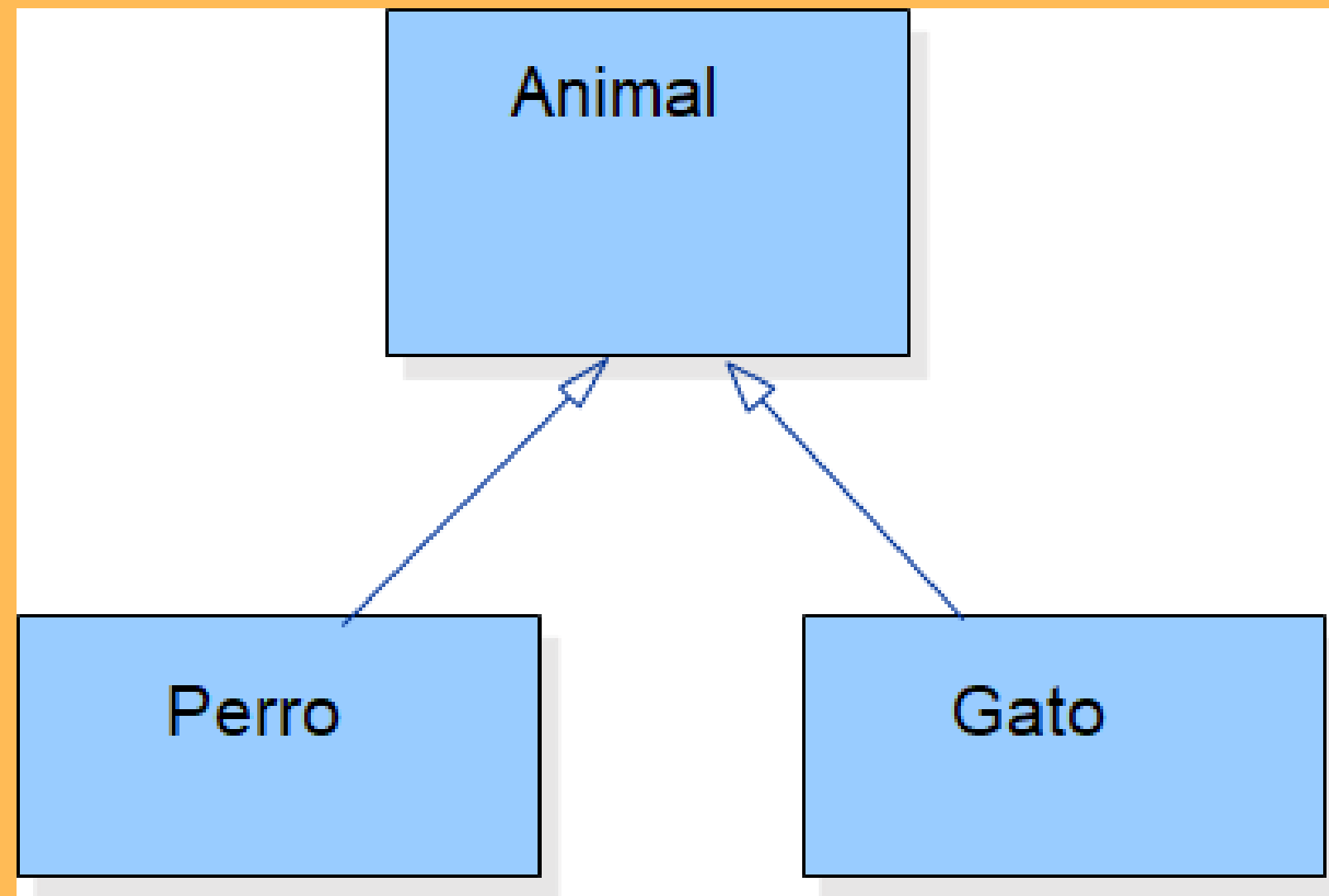
- Solo se le pueden enviar mensajes a objetos que entiendan el mensaje
- El objeto que recibe el mensaje deber tener un método para que maneje el mensaje recibido
- Los valores que viajan en el mensaje se llaman argumentos/parámetros



Herencia

una clase adquiere las:

- propiedades (atributos)
- comportamiento (métodos) de otra.

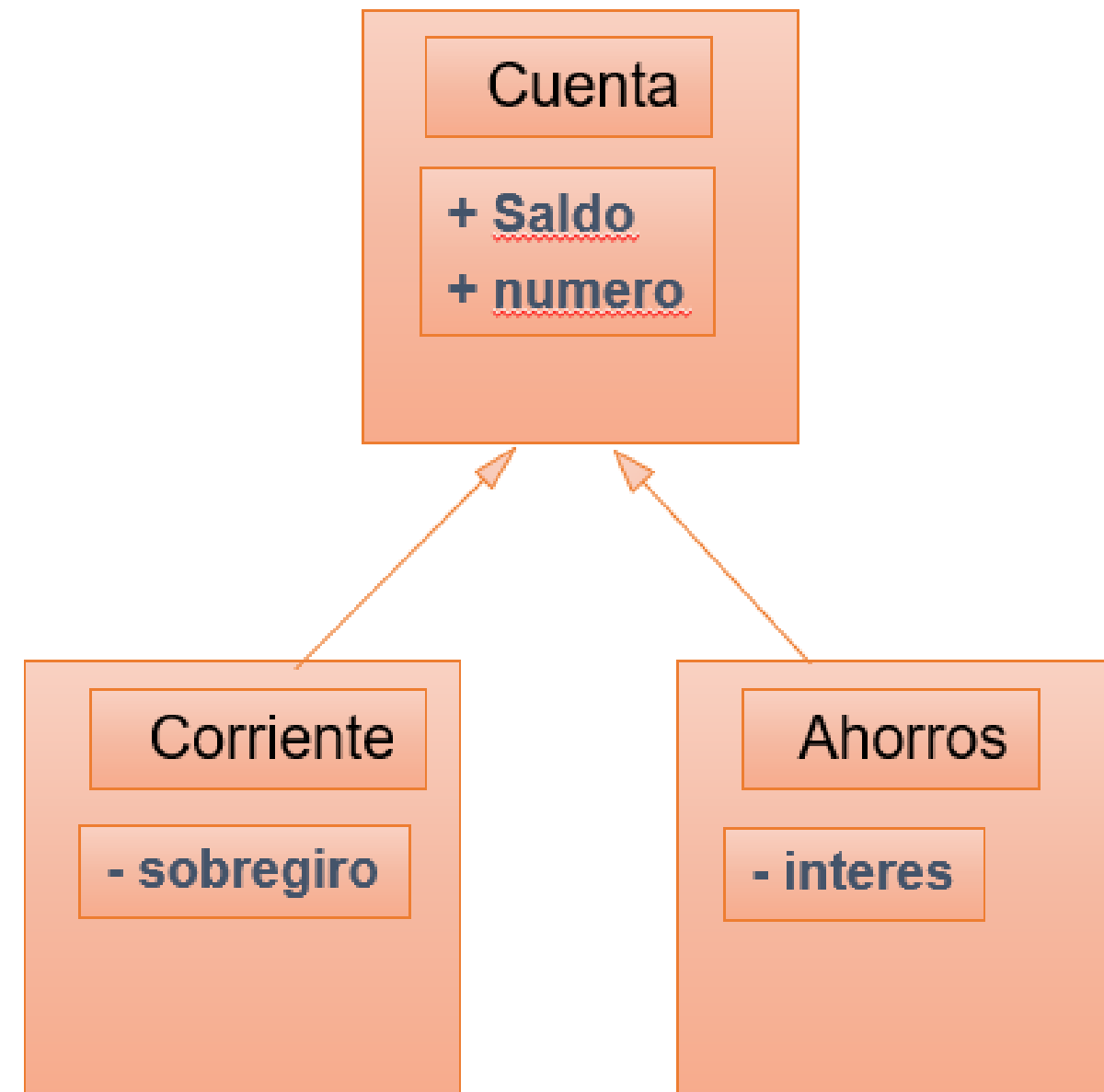


Ejemplo



La clase Corriente hereda
los atributos
de Cuenta

* En total Corriente tiene 3
atributos



Concepto de Herencia

Se puede definir una clase a partir de otra ya existente
Heredando sus atributos y métodos,

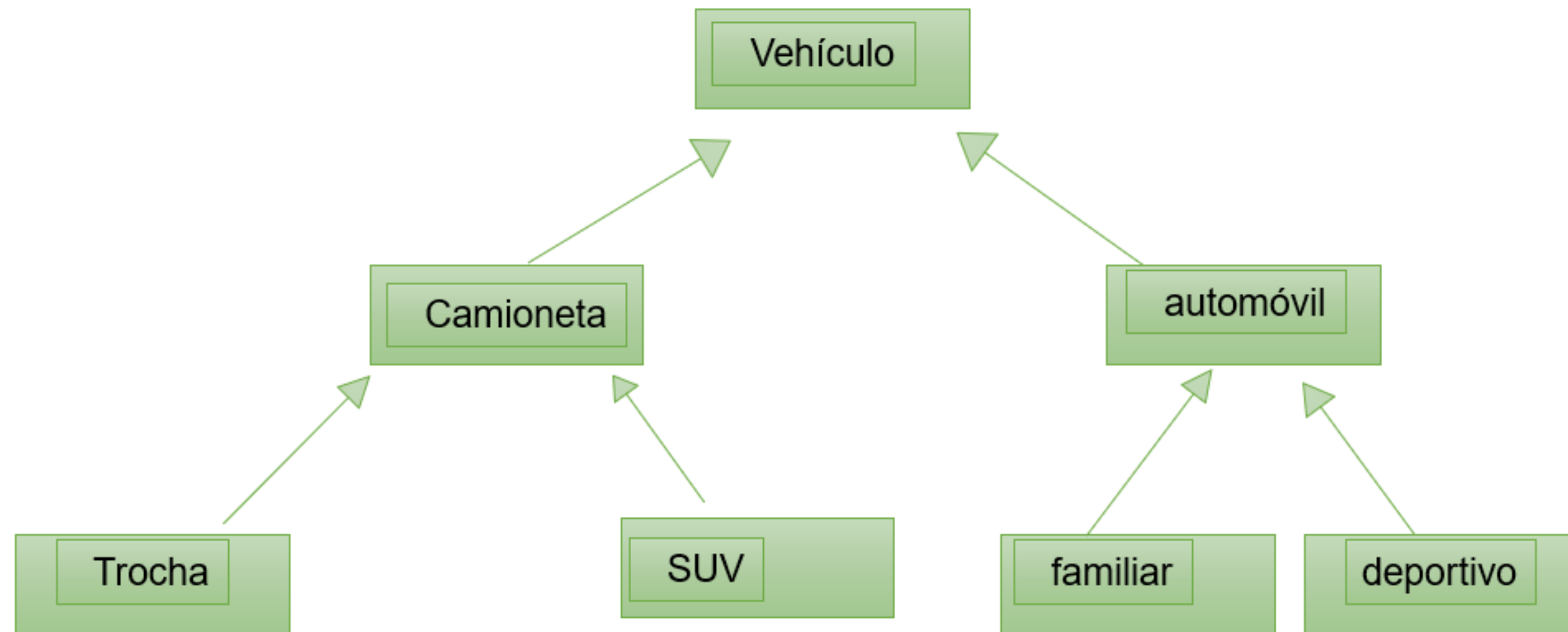
Y siendo posible:

- añadir nuevos elementos (atributos o métodos)
- Redefinir métodos



Jerarquía

Especialización: iniciando arriba y se descompone



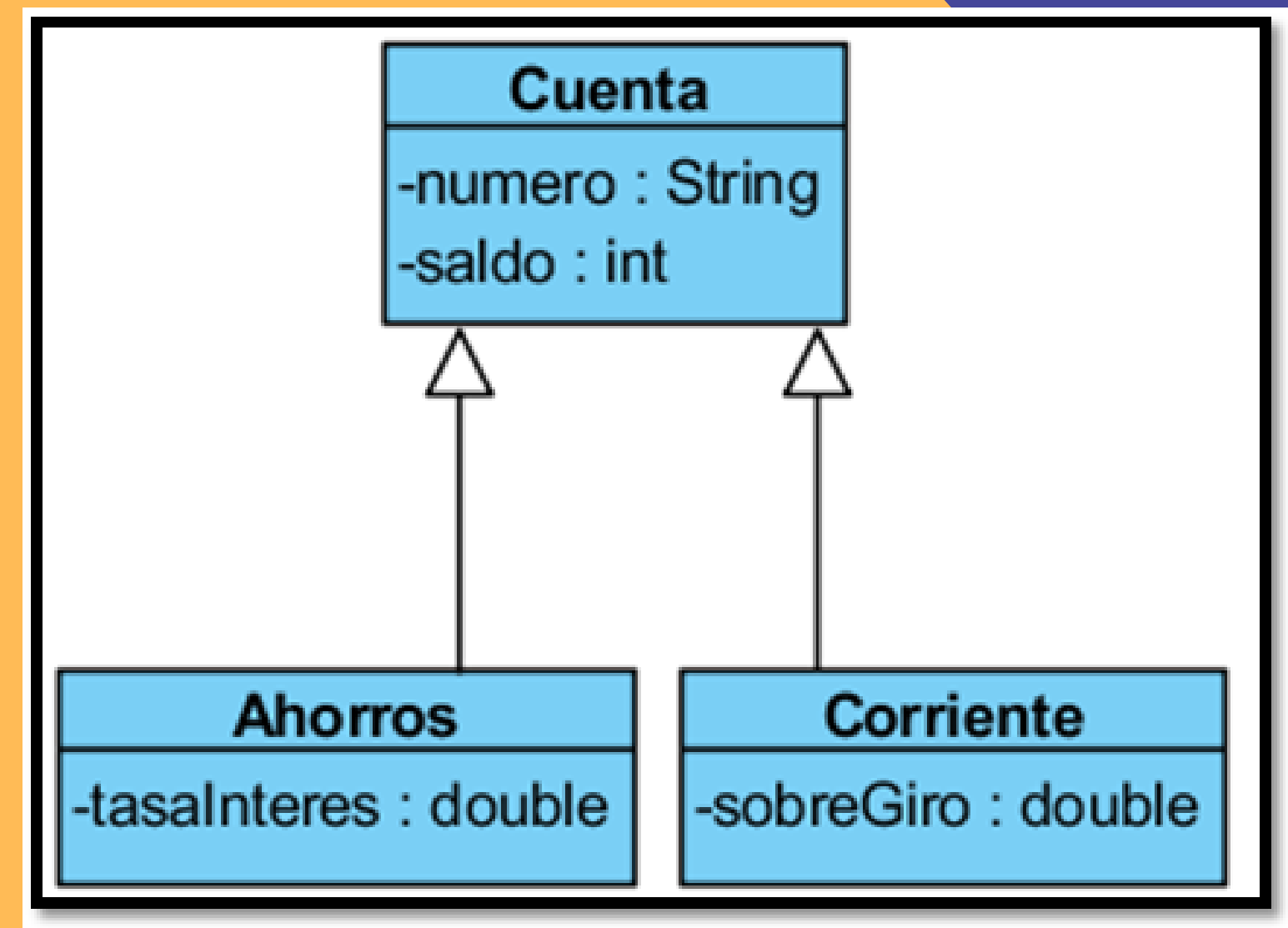
Generalización: iniciando abajo y se abstrae hacia arriba

Relación de Generalización

Características:

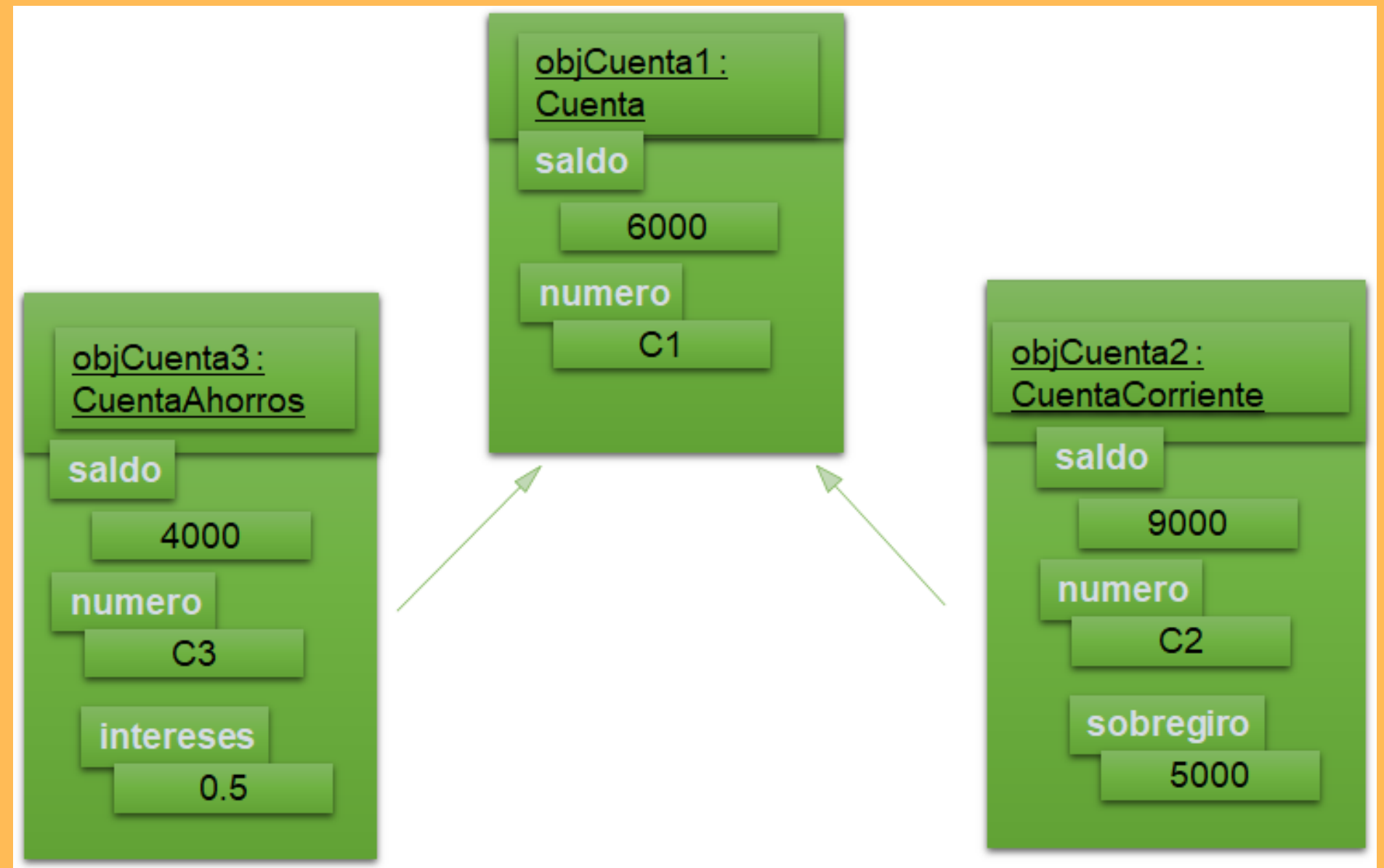
- Clase padre (Superclase)
- Clase hijo (Subclases)

La generalización tiene un triángulo apuntando a la superclase.



No se heredan valores !!

Herencia



Ejercicio

Dibujar un diagrama de herencia donde este represente un caso de la vida real

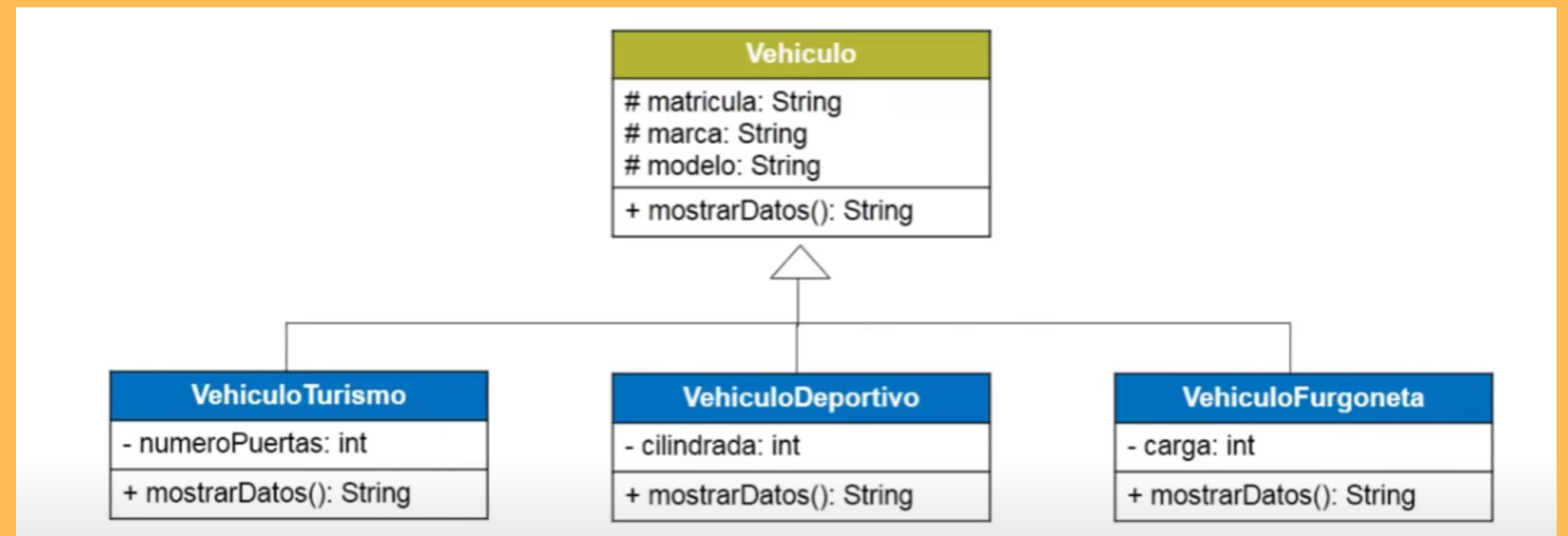
Que contenga:

- una superclase
- más de una subclase
- atributos
- métodos

Polimorfismo

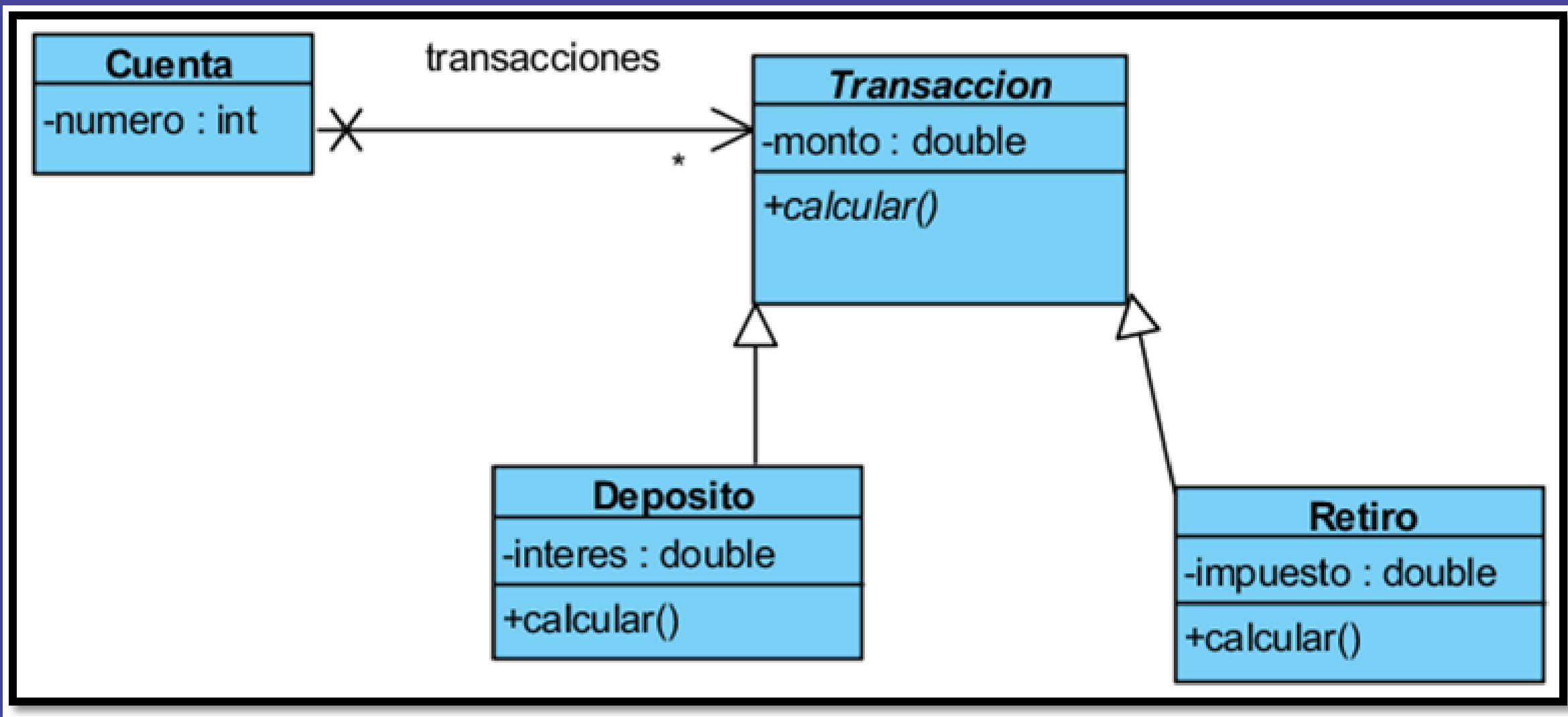
En una relación de herencia, un objeto de la superclase puede almacenar un objeto de cualquiera de sus subclases

Poli: muchas
morfismo: forma



Polimorfismo

El mismo método puede causar que diferentes acciones ocurran, dependiendo del tipo del objeto en el cual el método es invocado.



- Método `calcular()` en la superclase es abstracto
- Las clases **Depósito** y **Retiro** deben implementar sus métodos `calcular()`

Encapsulamiento

Nos permite ocultar o restringir el acceso a los atributos y/o métodos de una clase

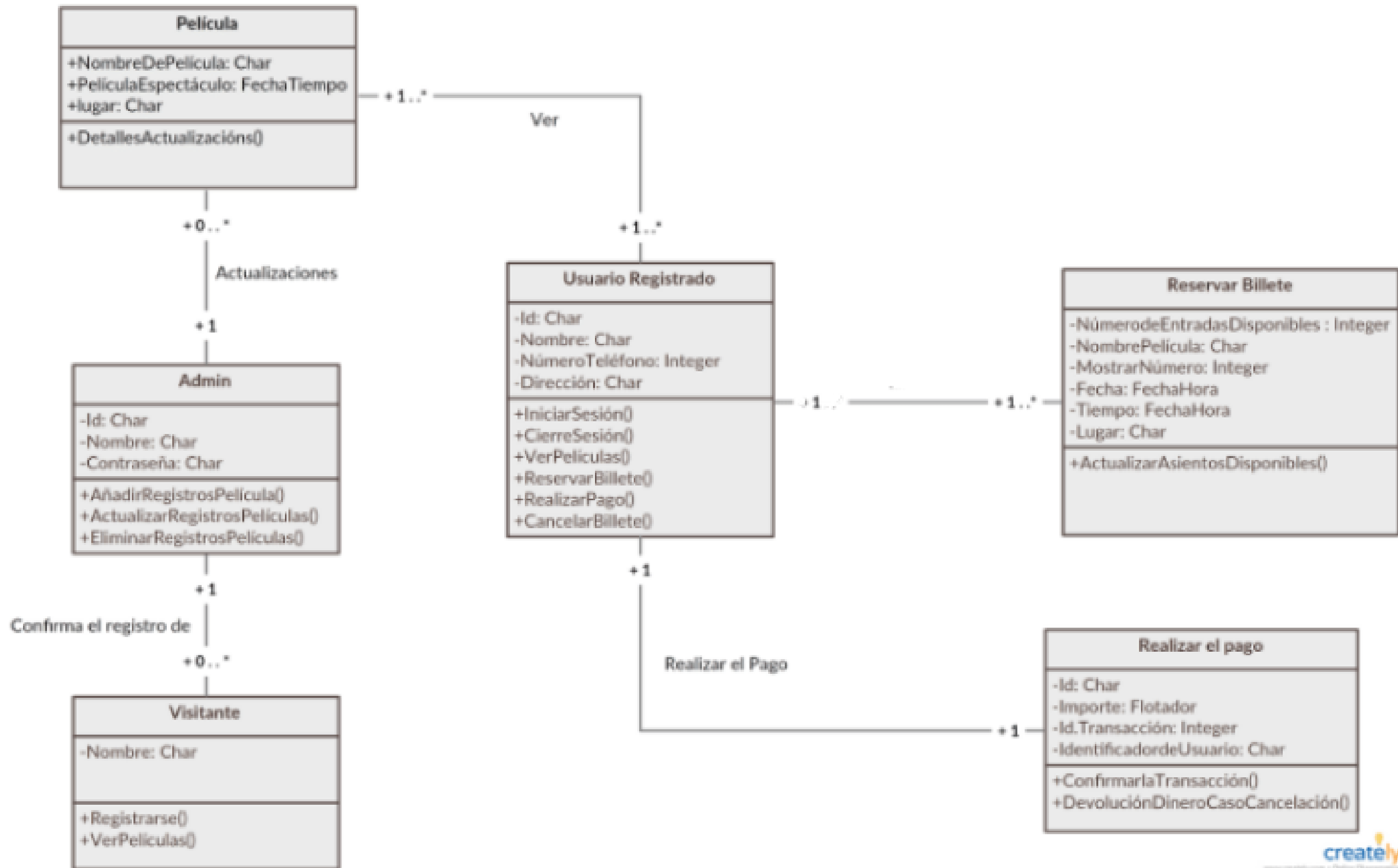
Existen 3 niveles de acceso

- **Public:** Todos pueden acceder a los datos o métodos de una clase (+)
- **Protegido:** No son publicas, solo son accesibles dentro de su clase y por sus subclases (#)
- **Privado:** Solo accesible dentro de la propia clase (-)

Diagrama de clases

Características

- Entidades
- Relaciones
- Cardinalidad
- Navegación



Ejercicio

Realizar un diagrama de clases, que represente un caso de la vida real

Que contenga:

- Todas las características de POO