

---

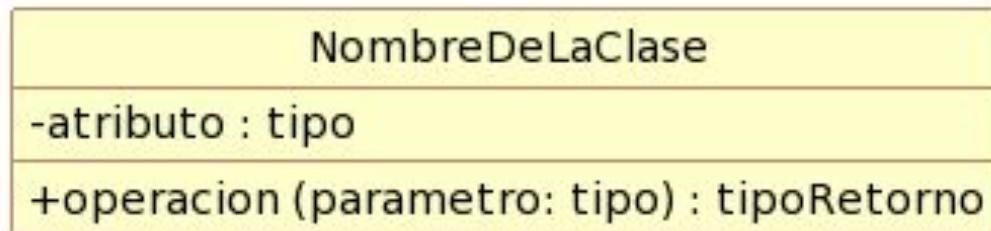
UNIFIED  
MODELING  
LANGUAGE



# Diagrama de Clases

Una clase es una descripción de conjunto de objetos que comparten los mismos atributos, operaciones, métodos, relaciones y semántica.

Una clase es representada gráficamente por cajas con tres compartimientos



# Diagrama de Clases

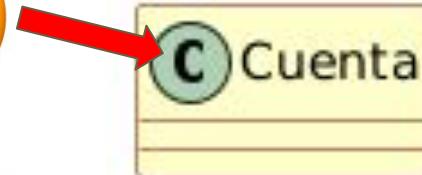
- Nombre de la clase

- singular

- debe comenzar con Mayúscula

- estilo CamelCase

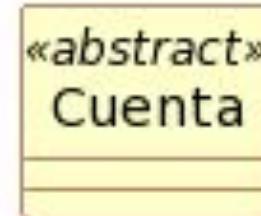
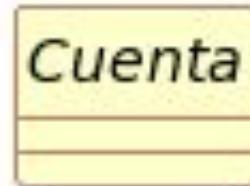
NombreDeLaClase
-atributo : tipo
+operacion (parametro: tipo) : tipoRetorno



- Si la clase es abstracta

- cursiva / Itálica

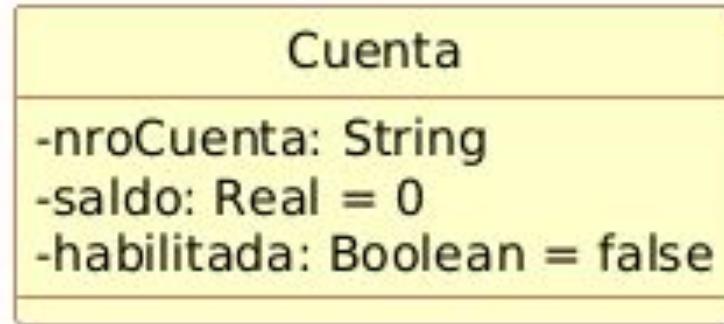
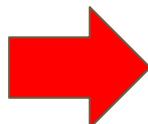
- estereotipo <<abstract>>



# Diagrama de Clases - Atributos

visibilidad nombre: tipo = valor por defecto

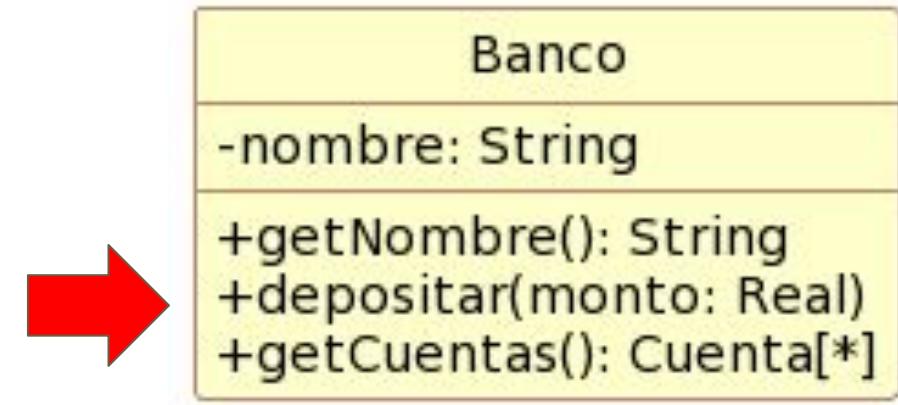
- visibilidad
  - privada (-)
  - protegida (#)
  - pública (+)
  - paquete (~)
- nombre
  - estilo camelCase
  - comienza con minúscula
- tipo
  - tipos UML: Integer, Real, Boolean, String
- valor por defecto



# Diagrama de Clases - Operaciones

**visibilidad nombre (parámetro: tipo): tipo retorno**

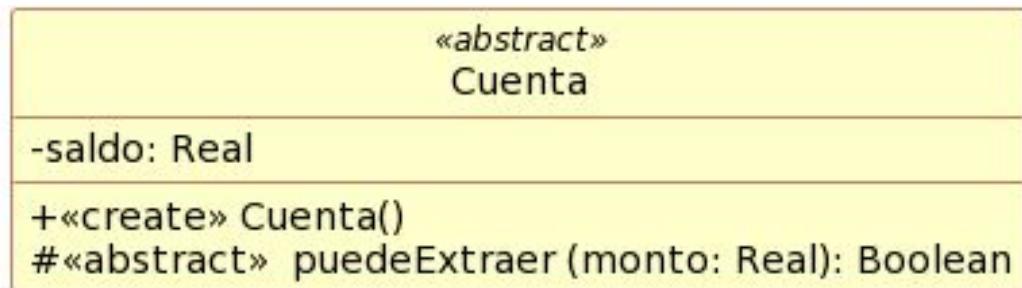
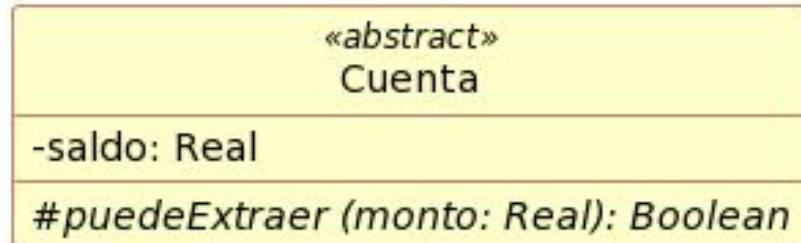
- visibilidad
  - privada (-)
  - protegida (#)
  - pública (+)
  - paquete (~)
- nombre
  - estilo camelCase
  - comienza con minúscula
- Parámetros
  - nombre: estilo camelCase
- tipo de retorno
  - Si no retorna nada, no se especifica
  - Si retorna un objeto, se indica de qué clase
  - Si retorna una colección, se indica el nombre de la clase [\*]



# Diagrama de Clases - Operaciones

visibilidad nombre (parámetro: tipo): tipo retorno

- si el método es **abstracto**
  - *cursiva*
  - con estereotipo <>abstract>>
- si el método es un **constructor**
  - con estereotipo <>create>>



# Diagrama de Clases - Relaciones

Asociaciones 

Herencia (Generalización) 

Implementación 

Dependencia 

# Diagrama de Clases - Asociación

- **navegabilidad**
- multiplicidad
- nombre de rol
- tipo:
  - simple
  - agregación
  - composición

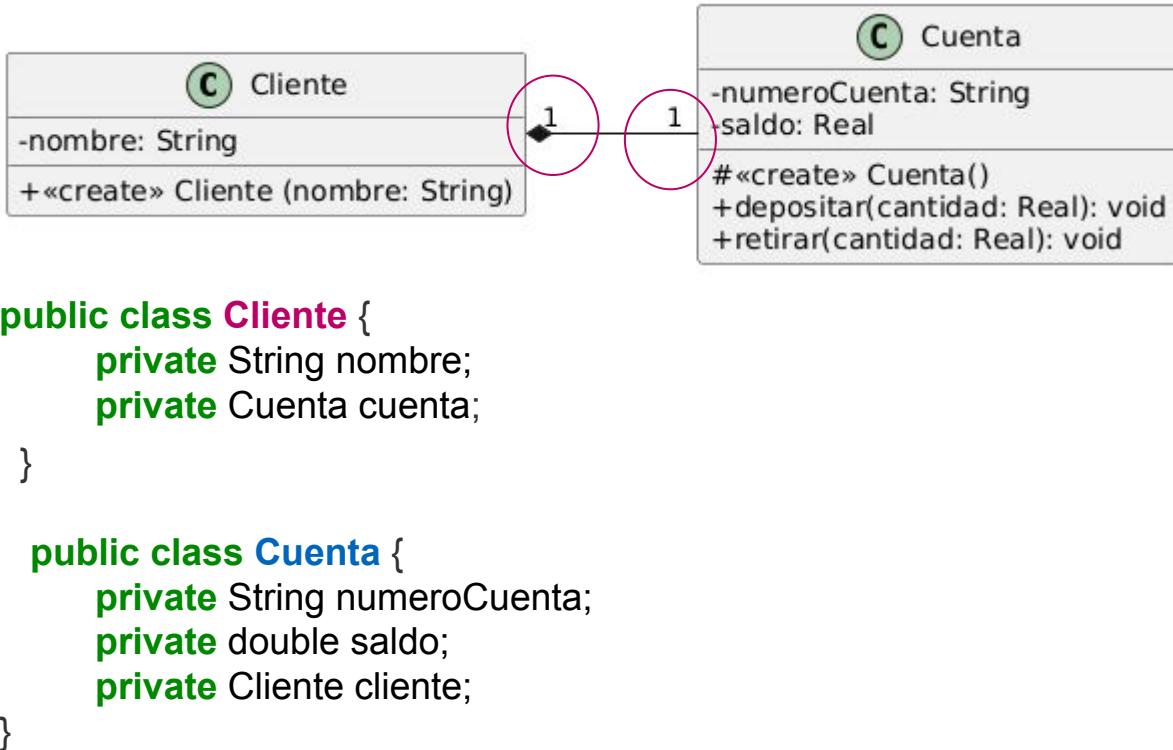


```
public class Cliente {
    private String nombre;
    private Cuenta cuenta;
}

public class Cuenta {
    private String numeroCuenta;
    private double saldo;
}
```

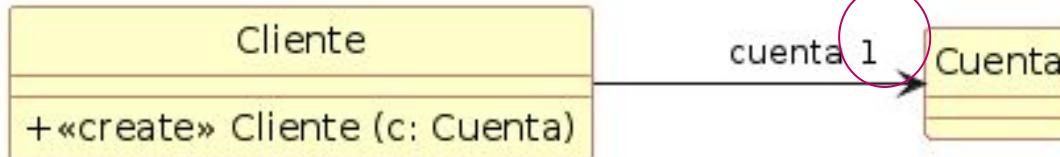
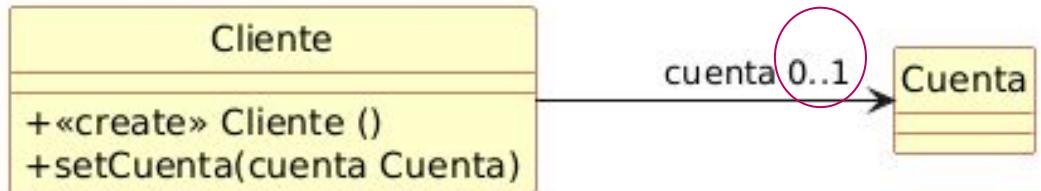
# Diagrama de Clases - Asociación

- **navegabilidad**
- multiplicidad
- nombre de rol
- tipo:
  - simple
  - agregación
  - composición



# Diagrama de Clases - Asociación

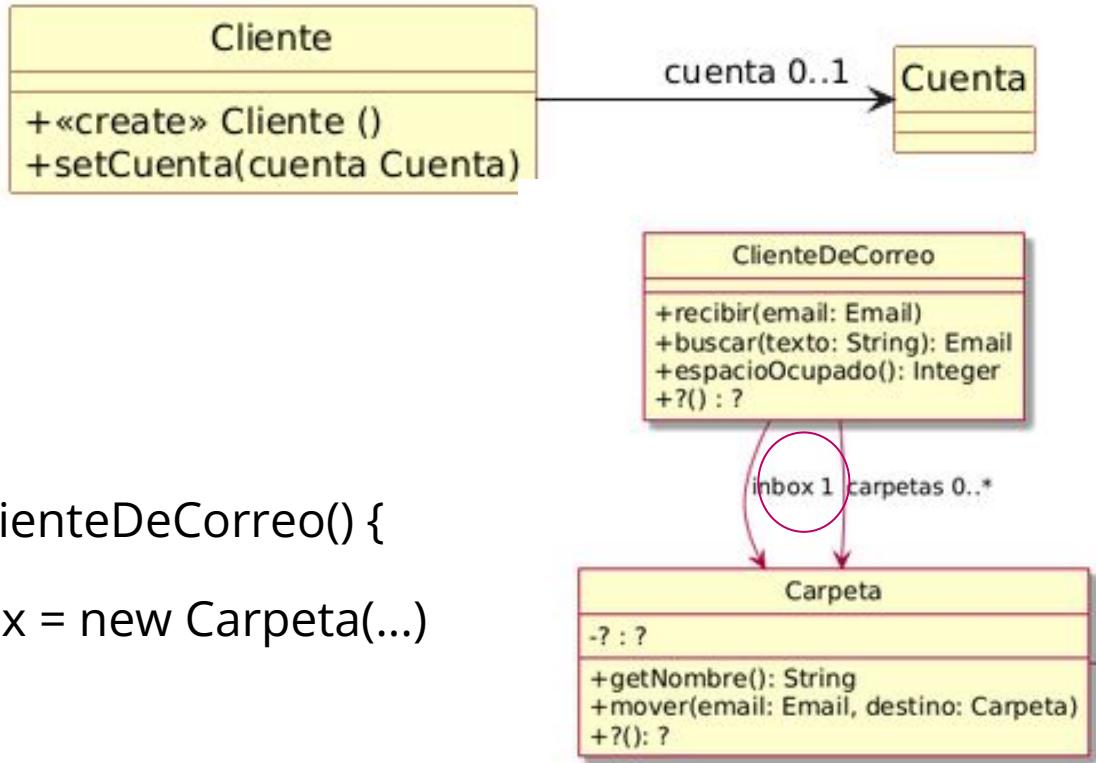
- navegabilidad
- **multiplicidad**
- nombre de rol
- tipo:
  - simple
  - agregación
  - composición



# Diagrama de Clases - Asociación

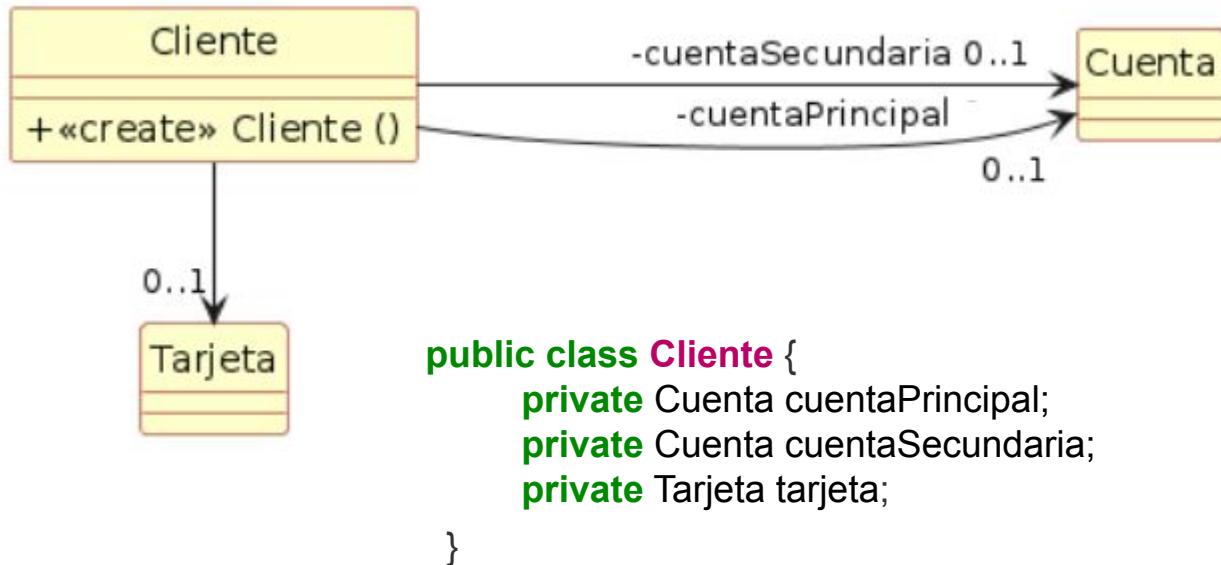
- navegabilidad
- **multiplicidad**
- nombre de rol
- tipo:
  - simple
  - agregación
  - composición

```
public ClienteDeCorreo() {  
    this.inbox = new Carpeta(...)  
}
```



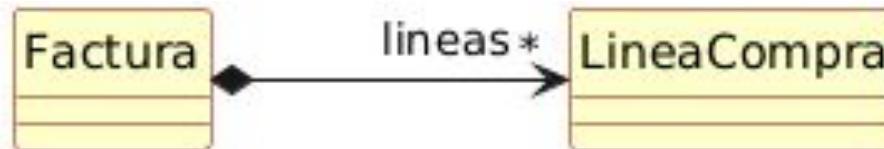
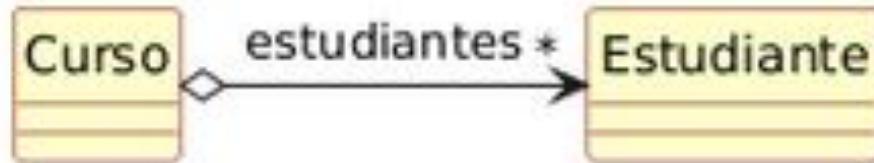
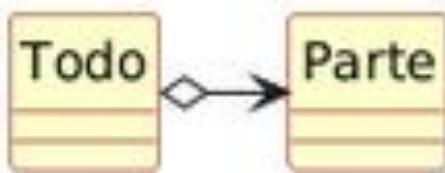
# Diagrama de Clases - Asociación

- navegabilidad
- multiplicidad
- **nombre de rol**
- tipo:
  - simple
  - agregación
  - composición



# Diagrama de Clases - Asociación

- navegabilidad
- multiplicidad
- nombre de rol
- **tipo:**
  - **simple**
  - **agregación**
  - **composición**



# Diagrama de Clases - Interfaces

Una interfaz define un conjunto de operaciones que una clase o componente debe implementar.

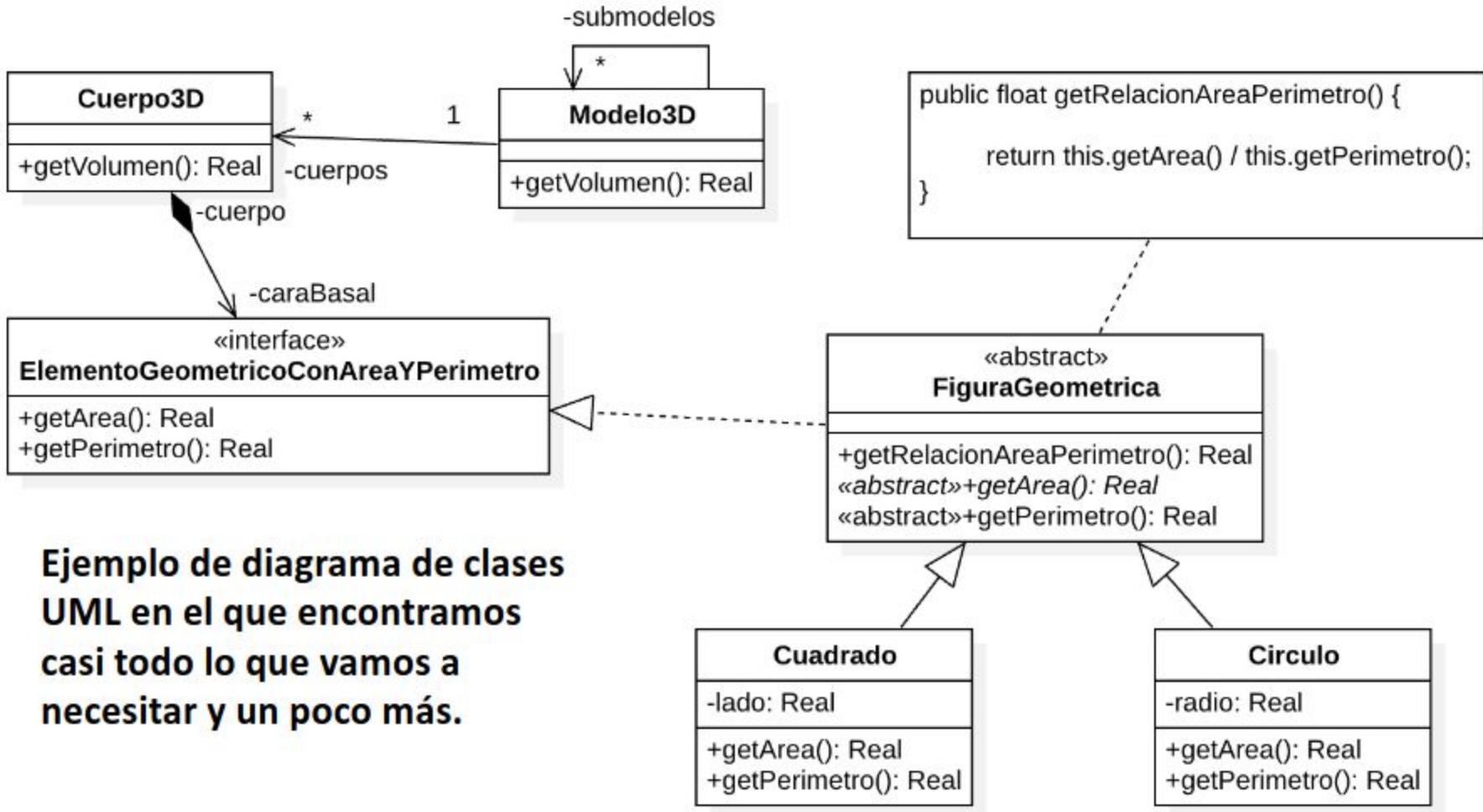
Actúa como un contrato que establece qué métodos deben ser implementados



En UML, las interfaces se representan mediante un rectángulo con el nombre de la interfaz precedido por el estereotipo <<interface>>.

Dentro del rectángulo, se listan las operaciones o métodos que la interfaz define.

La implementación de los métodos definidos en la interfaz es responsabilidad de las clases que la implementan.



## Diagrama de secuencia

Un diagrama de secuencia es un tipo de diagrama de interacción porque describe cómo —y en qué orden— colabora un grupo de objetos.

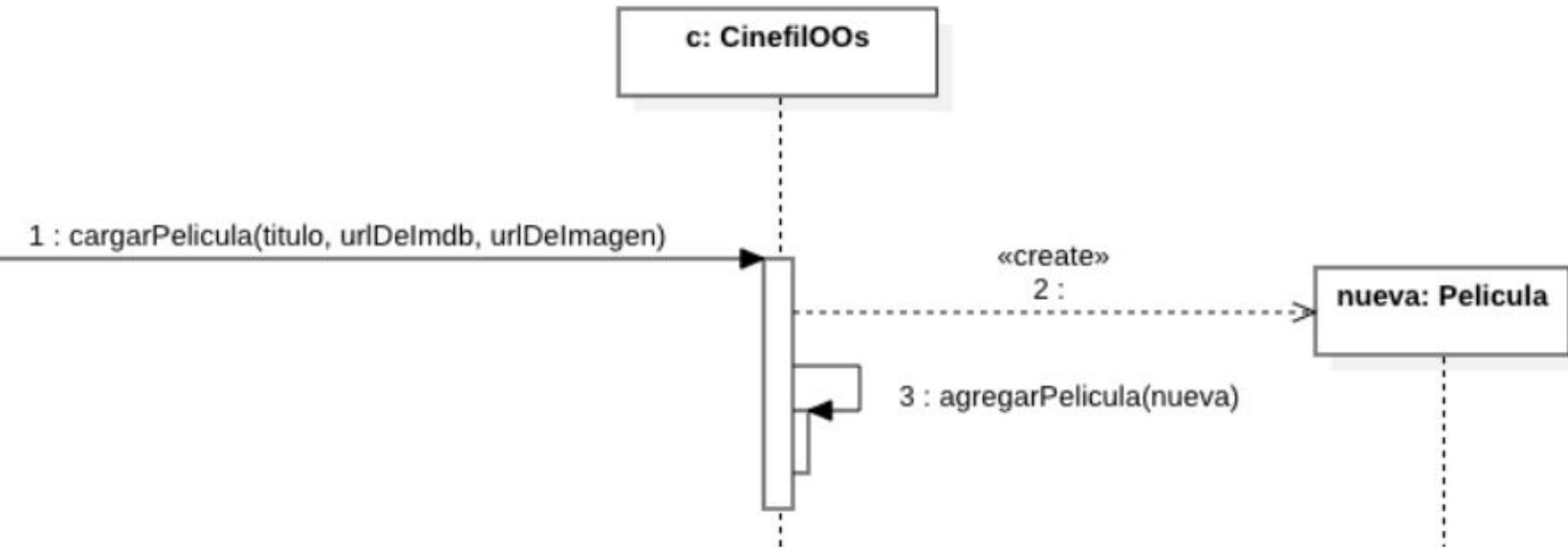
- Muestra claramente cómo interactúan distintos objetos en un sistema a lo largo del tiempo.

En un diagrama de secuencia

- los objetos se representan en la parte superior del diagrama
- el tiempo avanza de arriba hacia abajo.
-

# Diagrama de secuencia

Diagrama de secuencia del mensaje cargarPelicula de CinefilOOs

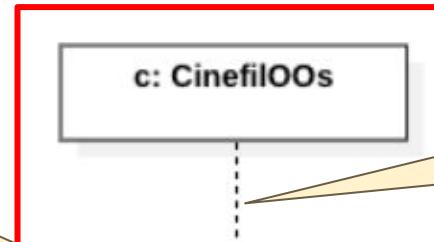


# Diagrama de secuencia

- Las flechas horizontales muestran las interacciones entre los objetos, indicando quién envía un mensaje a quién y en qué orden.

las flechas indican el envío de mensajes sincrónicos

1 : cargarPelícula(título, urlDelmdb, urlDelImagen)



Líneas de vida (punteada) representan la existencia temporal de un objeto

«create»  
2 :

nueva: Película

3 : agregarPelícula(nueva)



Activación: se representa mediante una barra vertical sobre la línea de vida

# Diagrama de secuencia

- Las flechas horizontales muestran las interacciones entre los objetos, indicando quién envía un mensaje a quién y en qué orden.

