
UNIFIED
MODELING
LANGUAGE
TM

Temas de la semana: UML (Lenguaje Unificado de Modelado).

Diagramas de clases.

Diagramas de secuencia.

Diagrama de Casos de uso.

Correspondencia entre elementos UML y elementos de lenguajes de programación.

Editores gráficos vs. Editores UML

****

Cómo está definido UML?

UML

- Es un lenguaje de modelado visual que nos permite
  - especificar
  - visualizar
  - construir
  - documentar

...artefactos de un sistema de software.

- Permite capturar decisiones y conocimientos.

Historia de UML

Historia de UML

Ivar Jacobson (Objectory)

Harel (state charts)

Odell

Meyer

Grady Booch (Booch)

James Rumbaugh (OMT)

Historia de UML

Ivar Jacobson (Objectory)

Harel (state charts)

Odell

Meyer

Grady Booch (Booch)

James Rumbaugh (OMT)

UNIFIED MODELING LANGUAGE

UML - Lenguaje de Modelado UNIFICADO

Grady Booch

- Método Booch (de Grady Booch)

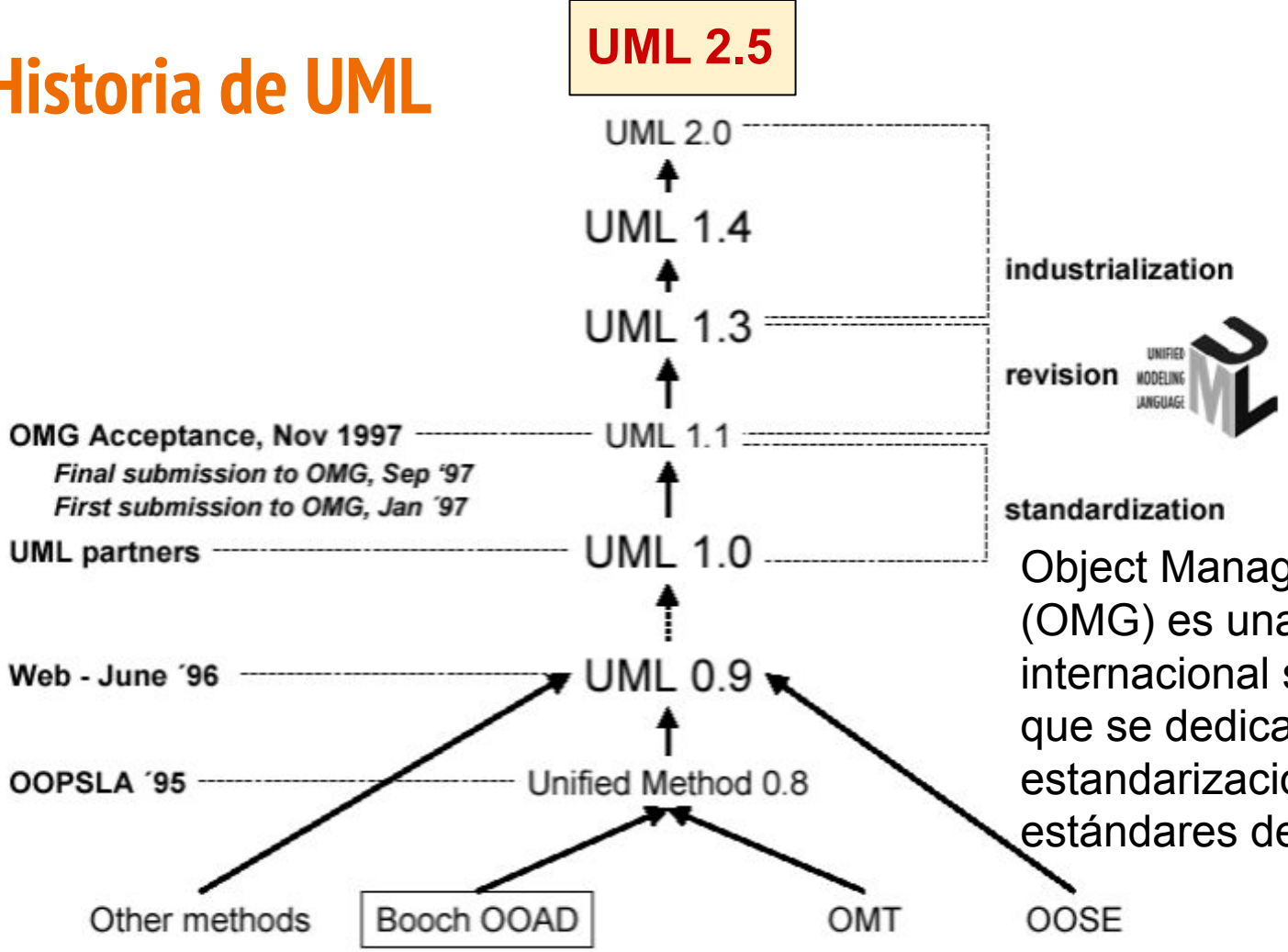
Jim Rumbaugh

- Object-Modeling Technique (OMT) de Rumbaugh

Ivar Jacobson

- OOSE (Object-oriented Software Engineering)

Historia de UML



Object Management Group (OMG) es una organización internacional sin fines de lucro que se dedica a la estandarización y promoción de estándares de tecnología.

Lenguaje UML

Lenguaje UML

- Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software.
-
- Permite capturar decisiones y conocimientos.

Diagramas de estructura

- Diagrama de Clases
- Diagrama de Paquetes
- Diagrama de Componentes
- Diagrama de Objetos
- Diagrama de Despliegue

Diagramas de estructura

- **Diagrama de Clases**
- Diagrama de Paquetes
- Diagrama de Componentes
- **Diagrama de Objetos**
- Diagrama de Despliegue

Diagramas de Comportamiento

- Diagrama de Casos de Uso
- Diagramas de Interacción
  - Diagrama de Secuencia
  - Diagrama de Colaboración
- Diagrama de Máquinas de estado
- Diagrama de Actividades

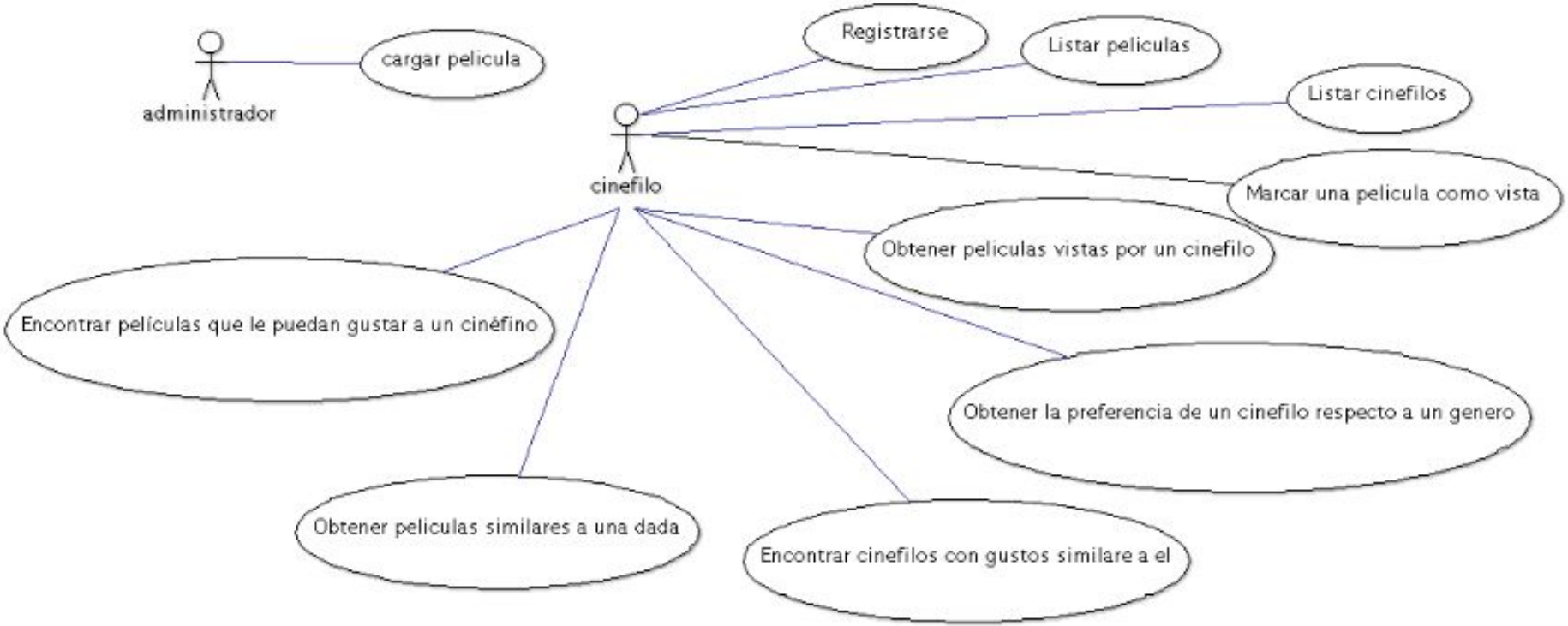
Diagramas de Comportamiento

- **Diagrama de Casos de Uso**
- Diagramas de Interacción
  - **Diagrama de Secuencia**
  - Diagrama de Colaboración
- Diagrama de Máquinas de estado
- Diagrama de Actividades

Diagramas de Casos de Uso

- Un caso de uso es una representación del comportamiento de un sistema tal como se percibe por un usuario externo.
- Describe una interacción específica entre los actores y el sistema, proporcionando un proceso completo de cómo se utiliza el sistema en situaciones reales.
- El término 'actor' engloba a personas, así como a otros sistemas que interactúan con el sistema
- Los elementos de un modelo de casos de uso son:
  - Actores
  - Casos de uso
  - Relaciones

Diagramas de Casos de Uso



Diagramas de Casos de Uso

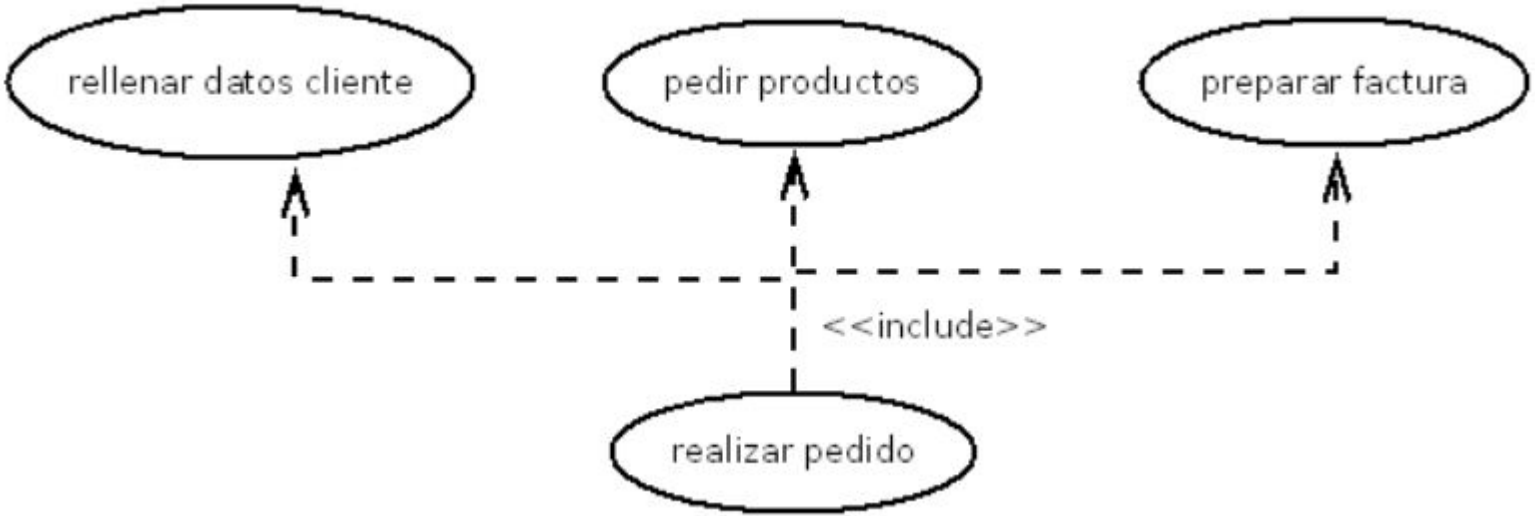
Actor: representa a un usuario externo, un sistema o una entidad que interactúa con el sistema que se está modelando.

Un caso de uso es una representación de una funcionalidad específica

Entre casos de uso pueden darse relaciones:
extensión <<extends>>
inclusión <<includes>>

Diagramas de Casos de Uso - Relaciones

una relación A include a B significa que una instancia de A también incorporará el comportamiento especificado en B.



Diagramas de Casos de Uso - conversación

Curso Normal (Basico)			
1	Actor 1: Acción realizada por el actor		
2	Actor 2: Acción realizada por el actor	**3**	Acción realizada por el sistema
		N	Cuando se realiza la inclusión de otro caso de uso lo representaremos de la forma. Incluir (CU_identificador. CU_Nombre)
	<< Se incluyen la secuencia de acciones realizadas por los actores que intervienen en el CU , se usaran, frases cortas, que describan el dialogo entre los actores y el sistema>> *<< Se pueden añadir referencias a elementos de un boceto del Interfaz del Usuario >>*		*<< Se incluyen la secuencia de acciones que realiza el sistema ante las acciones de los actores >>*

Cursos Alternos	
1a	Descripción de la secuencia de acciones alternas a la acción 1 del Curso Normal
1b	
	<< Secuencia de los cursos alternos del CU >>

acciones del actor

respuesta del sistema

Diagramas de Casos de Uso - conversación

Comprar producto			
1-	Se indica el producto que se quiere comprar		
		2-	Se agrega el producto al carrito de compras
3.	Se procede a terminar la compra		
		4-	Se calcula el total
		5-	Se muestran las opciones de pago
6	Se elige una opción de pago		
		7-	Se crea una orden de compra, registrando el pago

acciones del actor

respuesta del sistema

Diagrama de clases

Diagrama de Clases

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

Una clase es representada gráficamente por cajas con tres compartimientos

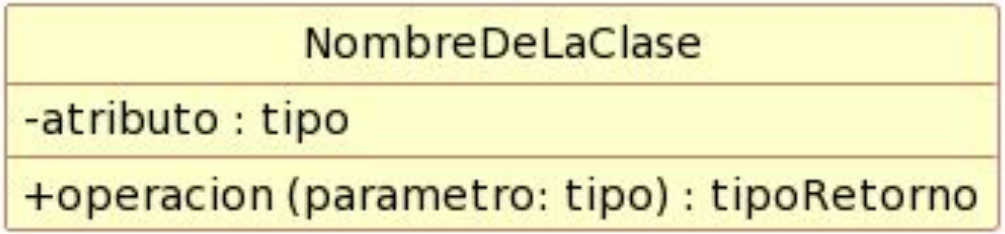
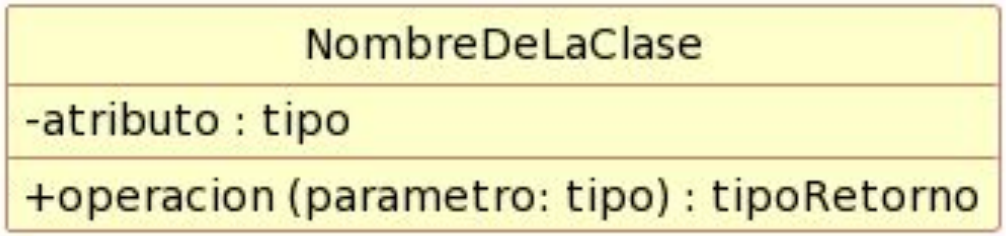
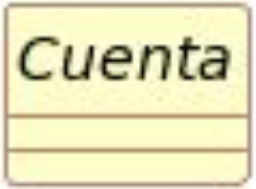


Diagrama de Clases



- Nombre de la clase
  - singular
  - debe comenzar con Mayúscula,
  - estilo CamelCase

- Si la clase es abstracta
  - cursiva
  - estereotipo <<abstract>>



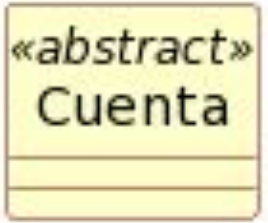


Diagrama de Clases - Atributos

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

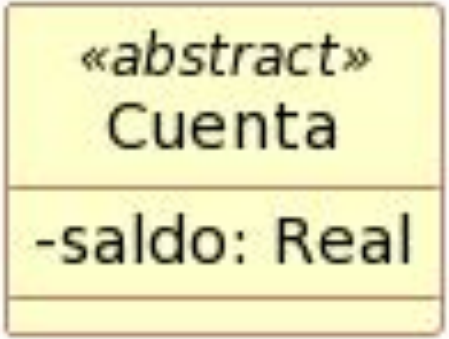


Diagrama de Clases - Operaciones

visibilidad nombre (parámetro: tipo): tipo retorno

- visibilidad
  - privada (-)
  - protegida (#)
  - pública (+)
  - paquete (~)
- nombre
  - estilo camelCase
  - comienza 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 [*]

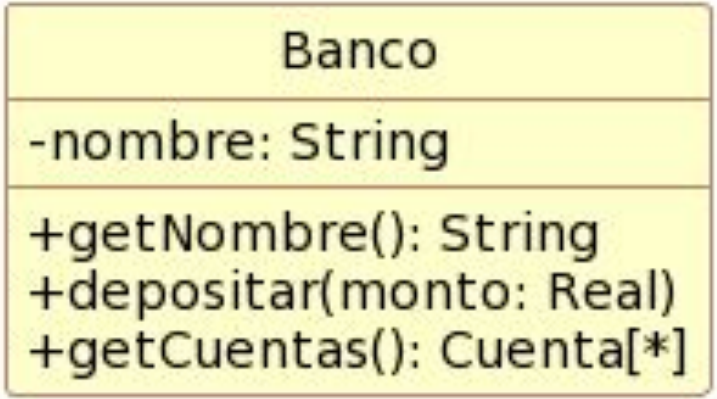
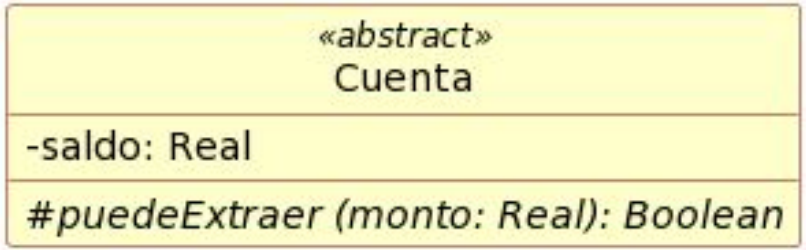


Diagrama de Clases - Operaciones

visibilidad nombre (parámetro: tipo): tipo retorno

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



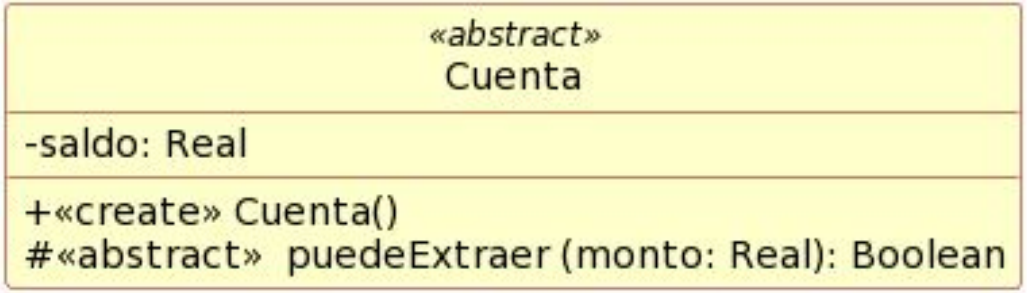
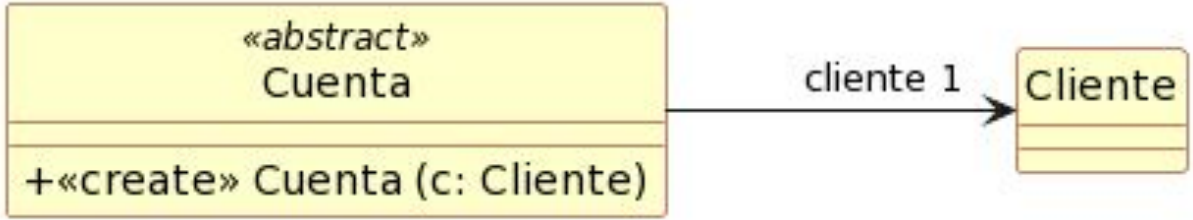
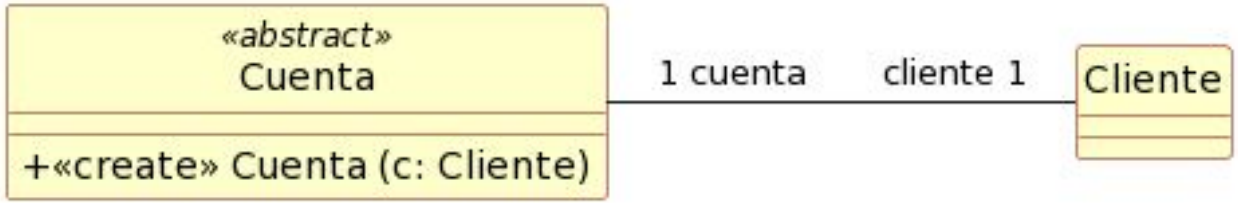


Diagrama de Clases - Asociación

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




```
public class Cuenta {
    private Cliente cliente
```



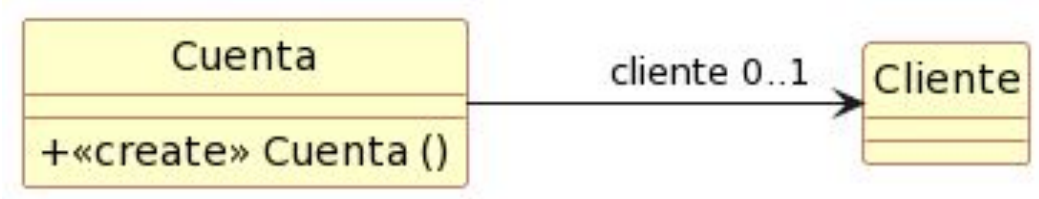

```
public class Cuenta {
    private Cliente cliente
```

```
public class Cliente{
    private Cuenta cuenta
```

```
this.cliente = c;
c.setCuenta(this)
```

Diagrama de Clases - Asociación

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



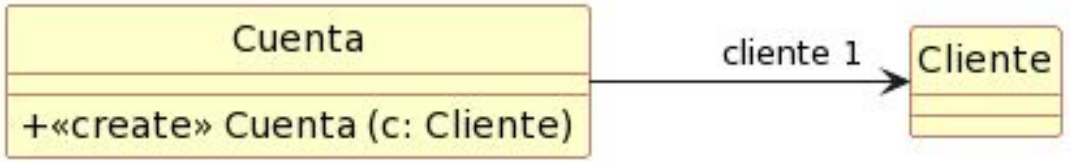


Diagrama de Clases - Asociación

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

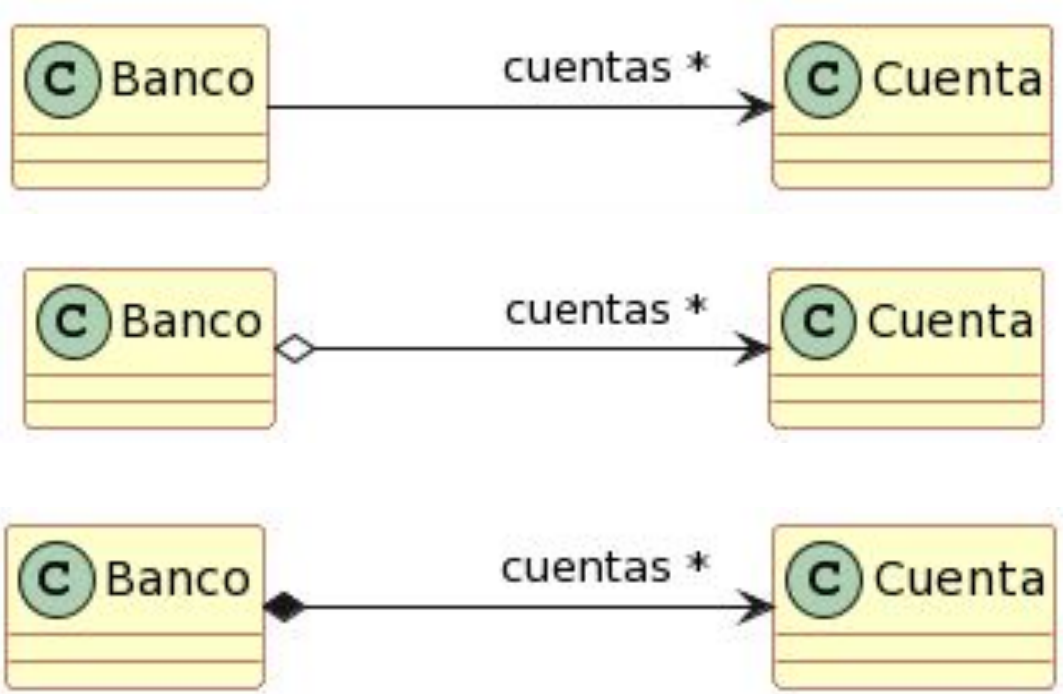


Diagrama de Clases - Interfaces

es un elemento que define un conjunto de operaciones que una clase o componente debe implementar.

Las interfaces en UML se representan mediante una caja con el nombre de la interfaz y una lista de operaciones o métodos que deben ser implementados por las clases que la utilicen. No se especifica la implementación de las operaciones en la interfaz; simplemente se define qué operaciones deben estar disponibles en las clases que la implementen.

Diagrama de Clases - Interfaces y herencia

Una clase concreta implementa una interface. Esto significa que la clase proporciona una implementación concreta para todos los métodos (operaciones) definidos en esa interfaz.

Si una clase concreta extiende a una clase abstracta, significa que debe proveer las implementación de los métodos abstractos.

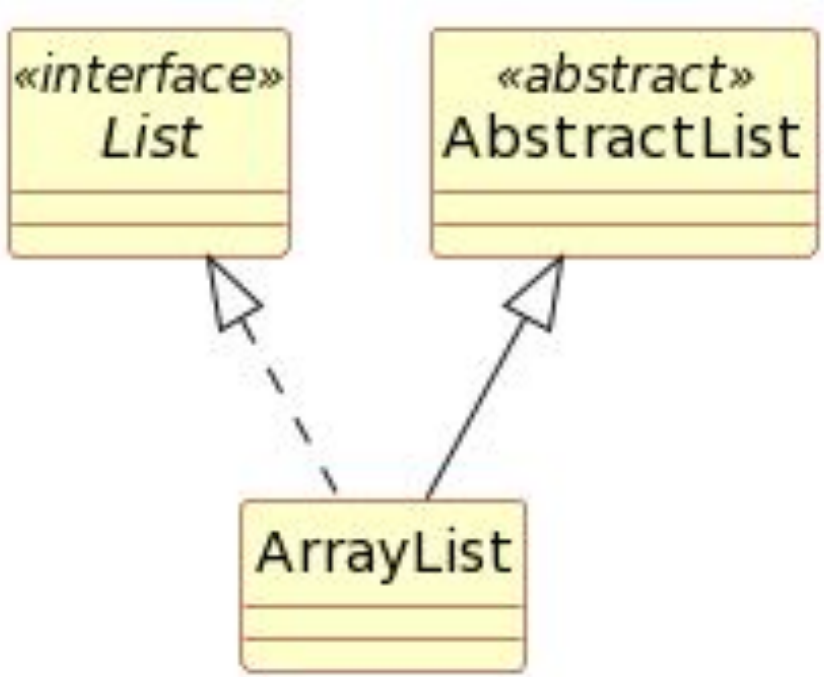


Diagrama de Clases

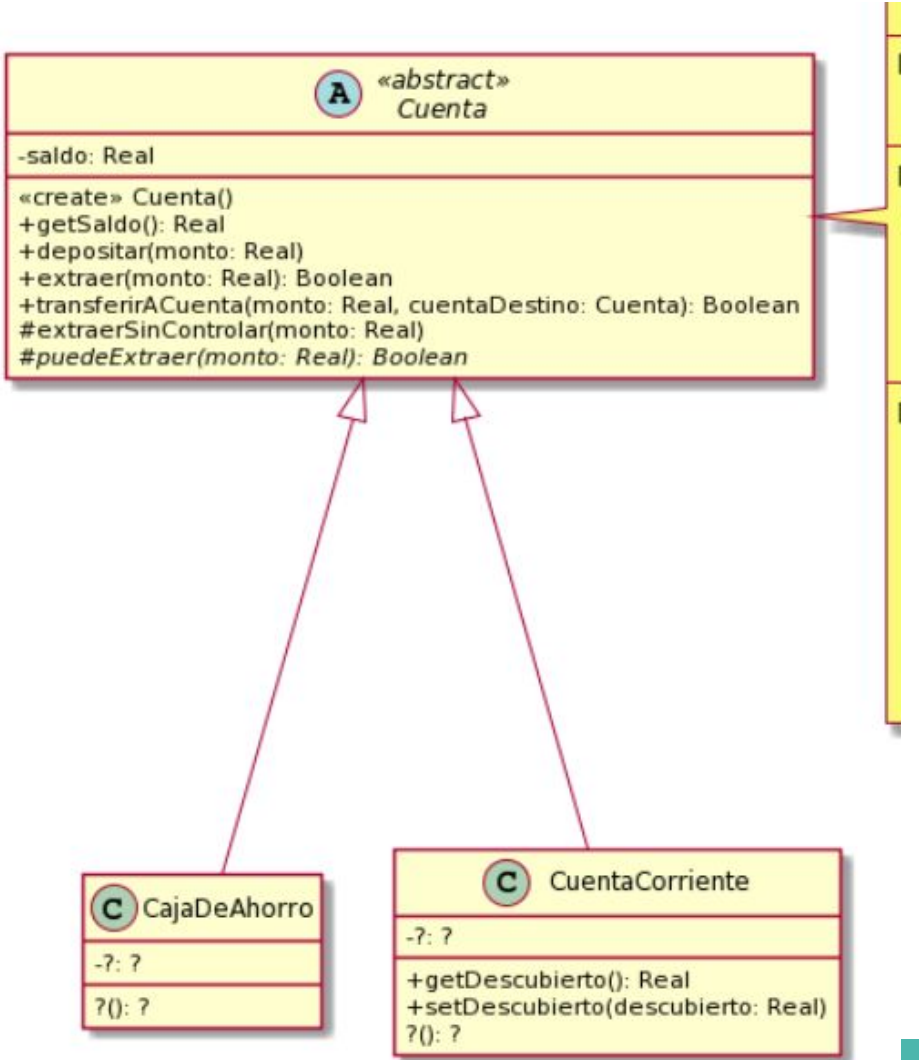


Diagrama de objetos

Diagrama de objetos

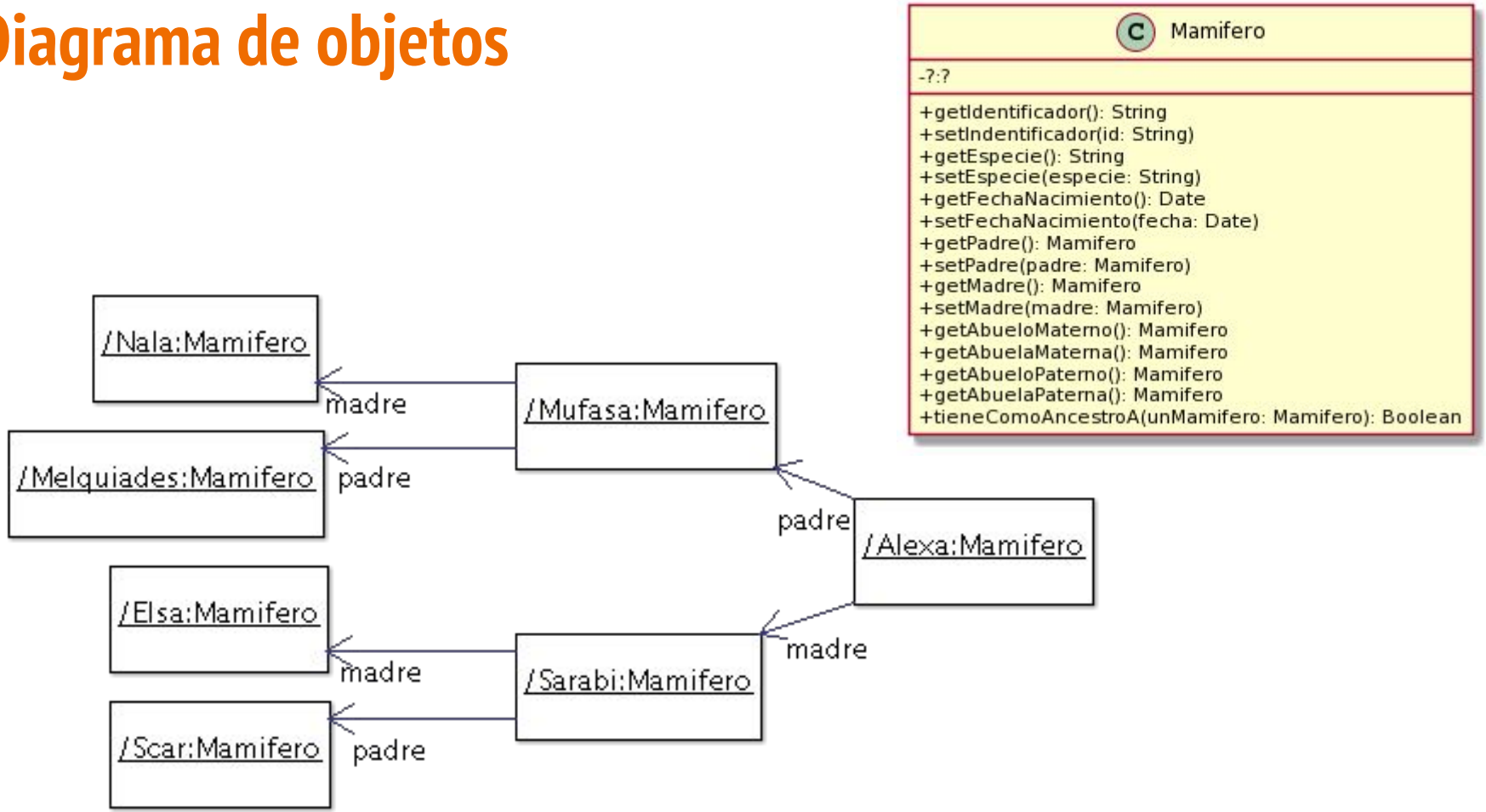
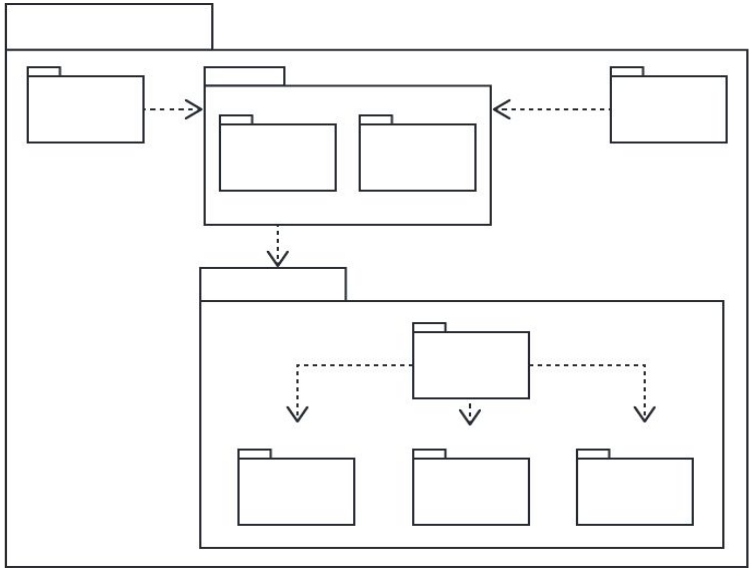


Diagrama de paquetes

Diagrama de paquetes



Permiten la agrupación de clases

Se utiliza para organizar los elementos.

Son útiles para mostrar la organización de un sistema y cómo los elementos se agrupan y relacionan entre sí.

Se quiere

- una alta cohesión dentro de un paquete. Los elementos dentro de un paquete están relacionados
- poco acoplamiento entre ellos (exportando sólo aquellos elementos necesarios e importando solo lo necesario)

Diagrama de secuencia

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

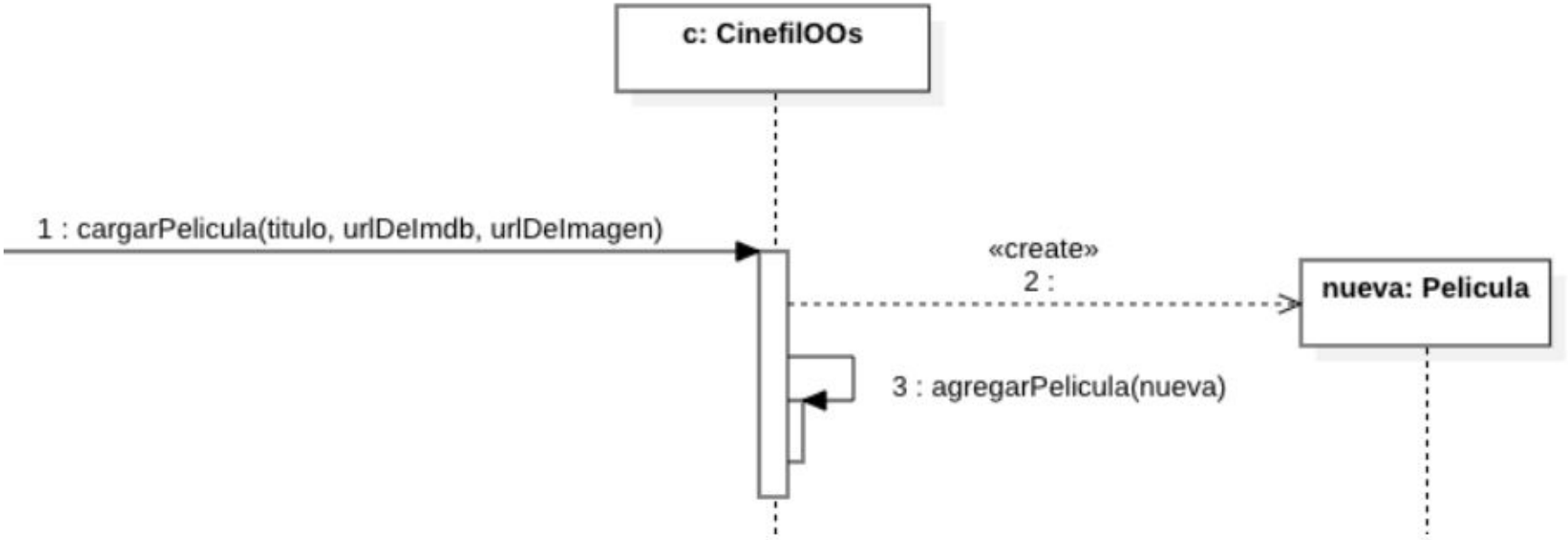


Diagrama de secuencia

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

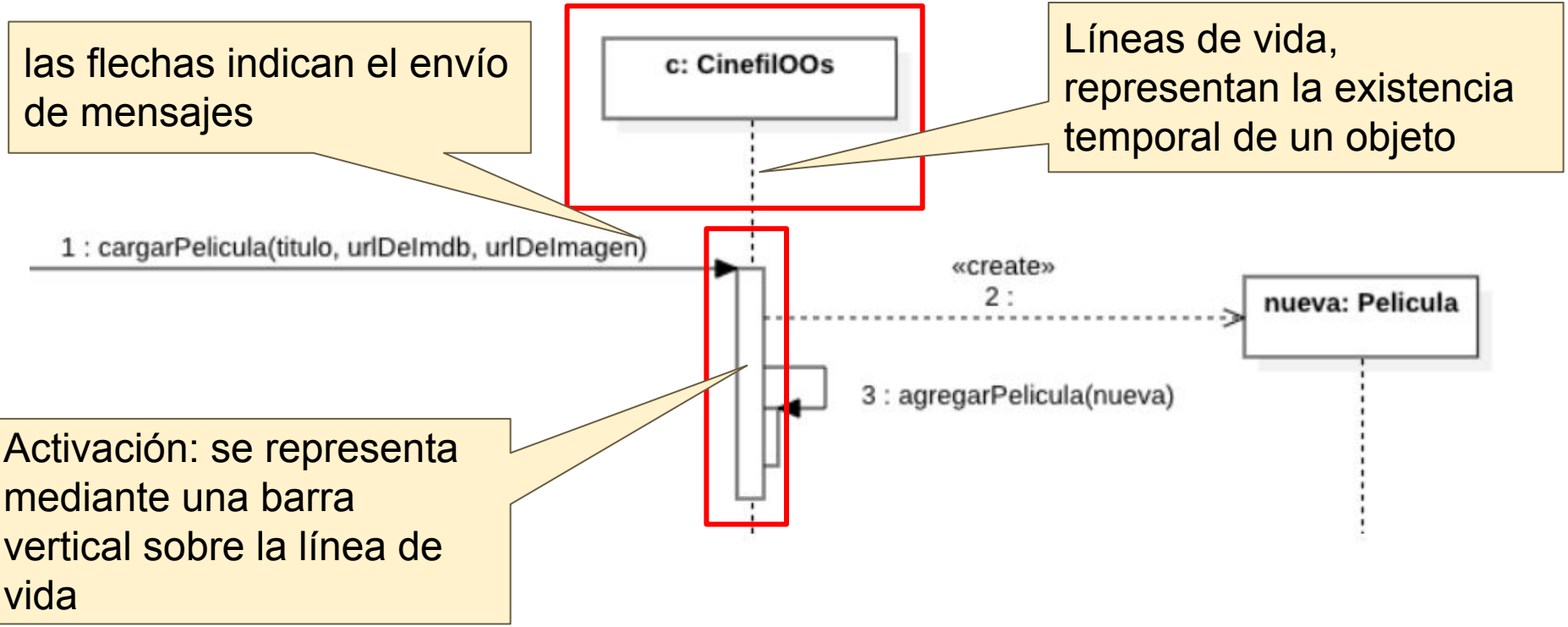


Diagrama de secuencia

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

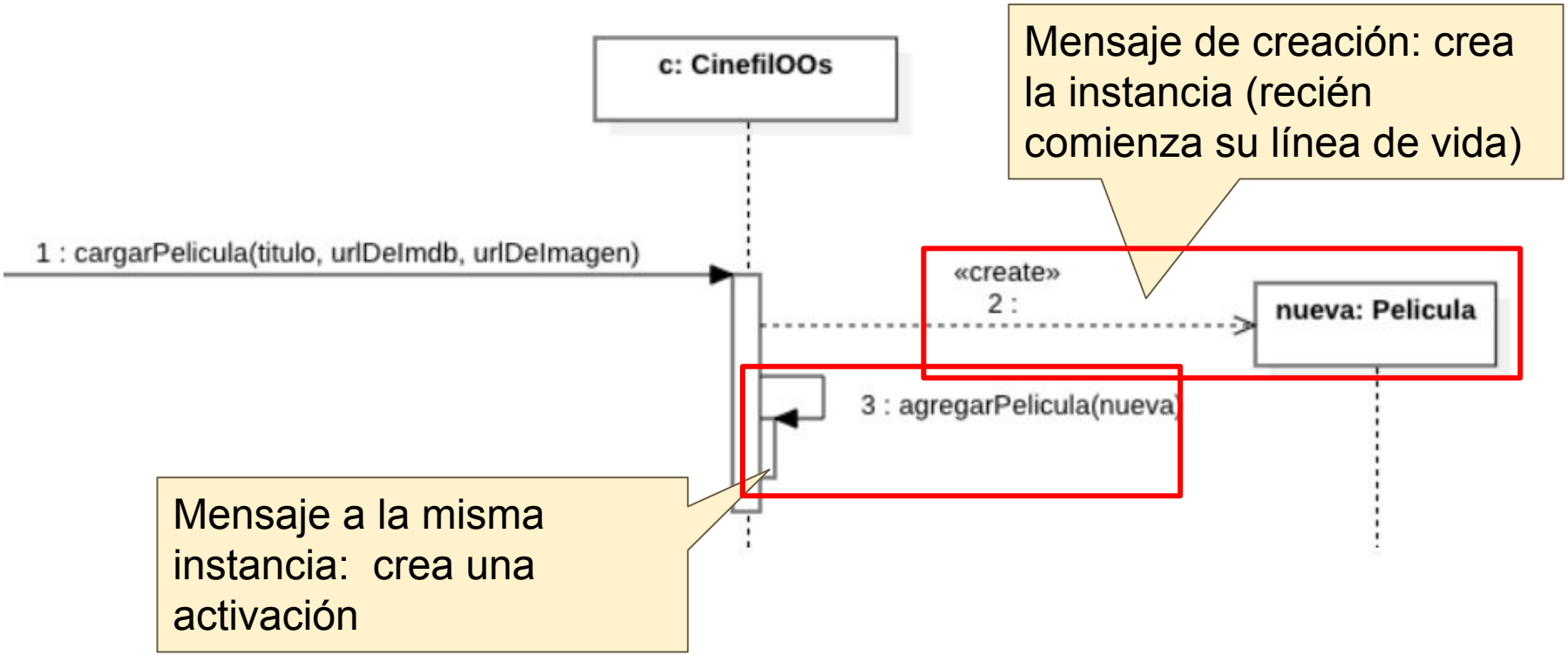


Diagrama de secuencia

Los corchetes indican que es opcional

Sintaxis del mensaje:

```
[valor de retorno := ] nombre del mensaje (parámetros)
```

Ejemplos:

```
agregarPelicula (nueva)

hayMonto:= hayMontoSuficienteParaExtraer(monto)
```

Diagrama de secuencia - CombinedFragment

Un fragmento combinado es un elemento que se utiliza para representar la lógica de control y las estructuras condicionales en una secuencia de interacción entre objetos. A través de ellos se pueden especificar bloques repetitivos, opcionales y alternativos, entre otros.

Fragmentos más utilizados:

- opt
- alt
- loop

Diagrama de secuencia - CombinedFragment

- **alt (Alternativa)**: Este tipo de fragmento se utiliza para modelar una elección entre diferentes opciones de interacción. En cada opción se evalúa una condición booleana para determinar cuál de las opciones se ejecutará.

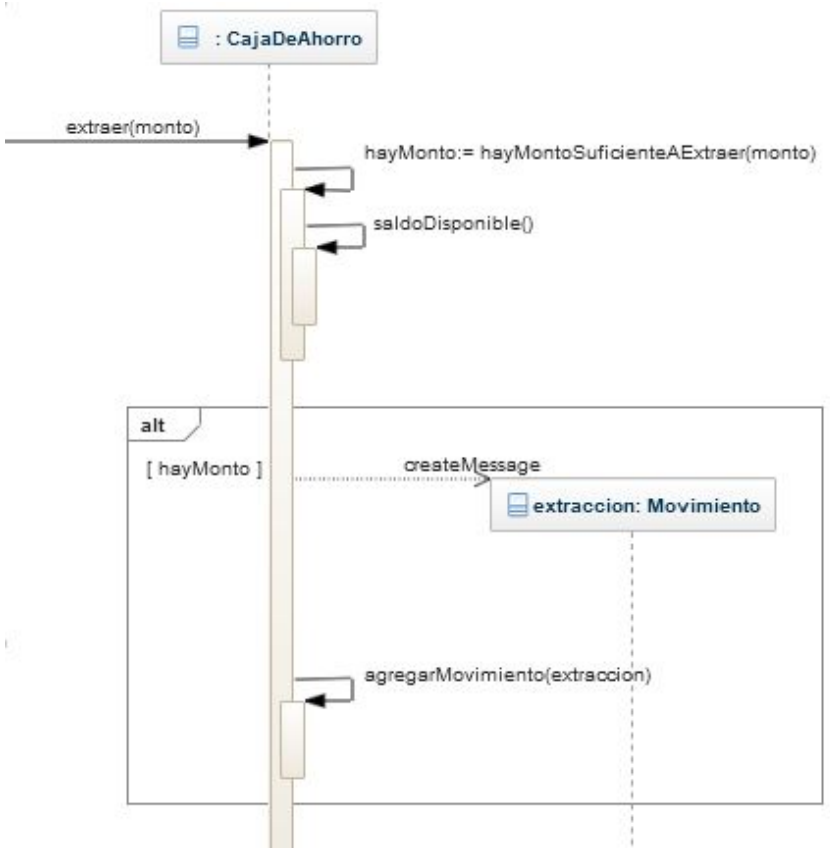
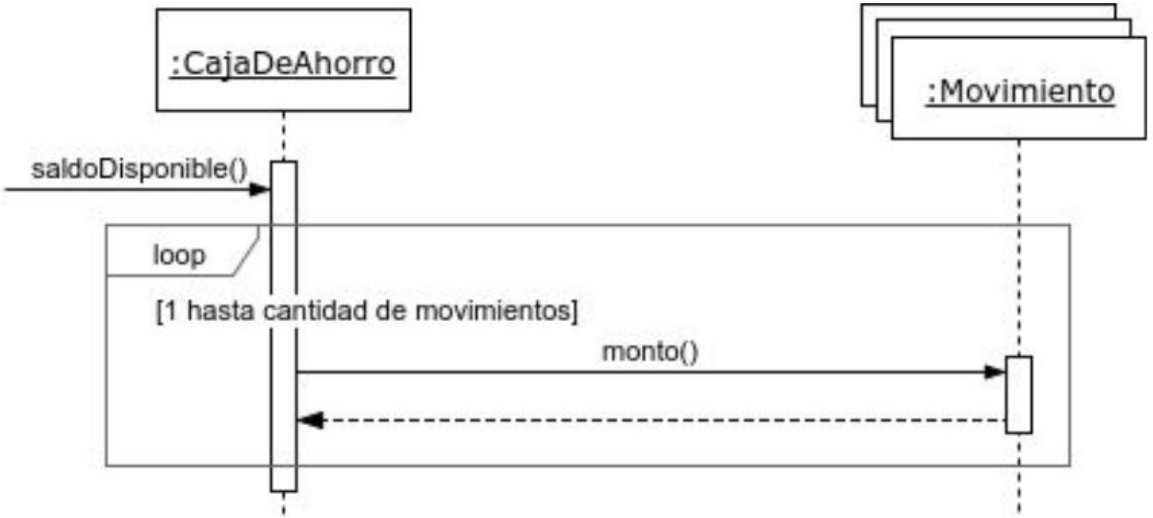


Diagrama de secuencia - CombinedFragment

- **opt (Opcional)**: Representa una parte de la secuencia de interacción que puede o no ejecutarse, dependiendo de una condición booleana. Si la condición es verdadera, se ejecuta la parte opcional; de lo contrario, se omite.

Diagrama de secuencia - CombinedFragment

loop (Bucle): Se utiliza para modelar repeticiones de una secuencia de interacción. Puede especificar el número de repeticiones o utilizar una condición para controlar la terminación del bucle.



Herramientas UML

Cómo se define UML?

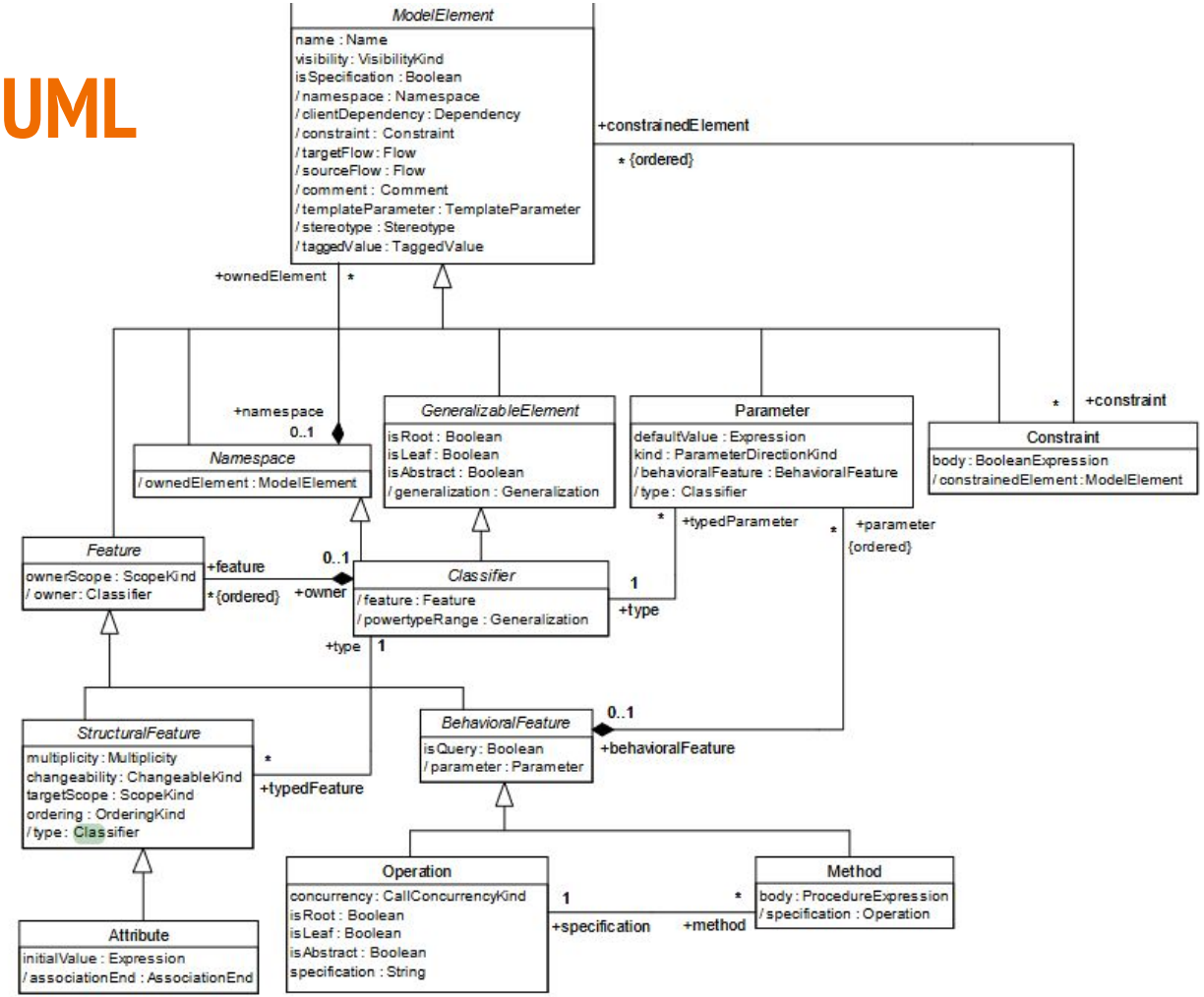
Cómo se define UML?

Documento de especificación de UML

Metamodelo de UML

- es una especificación que define las construcciones y elementos básicos que pueden utilizarse para crear diagramas en UML.
- es esencialmente una descripción formal de cómo se estructuran y relacionan los elementos

Metamodelo de UML



Herramientas de modelado

Las herramientas de modelado UML pueden ser tanto gráficas como basadas en el metamodelo UML

Herramienta gráfica: permite dibujar elementos con la misma imagen que UML

Herramientas basadas en el metamodelo UML: Estas herramientas se centran más en la manipulación directa de los elementos del metamodelo UML.

PlantUML

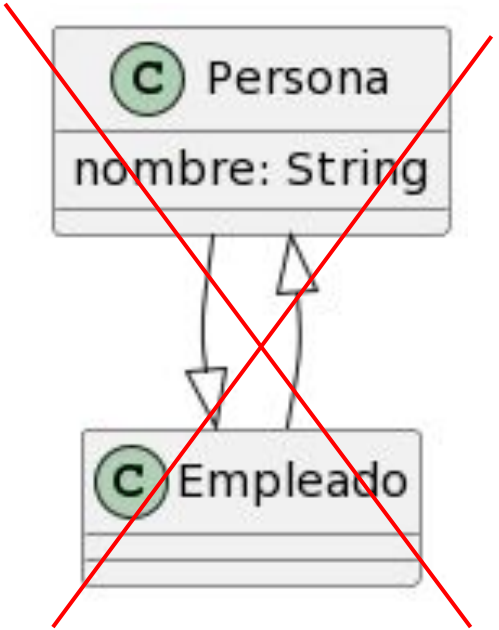
Herramienta que pasa de texto a imagen

no tiene chequeo del metamodelo.

Permite dibujar cosas ajenas a UML, o con errores semánticos

Versión online

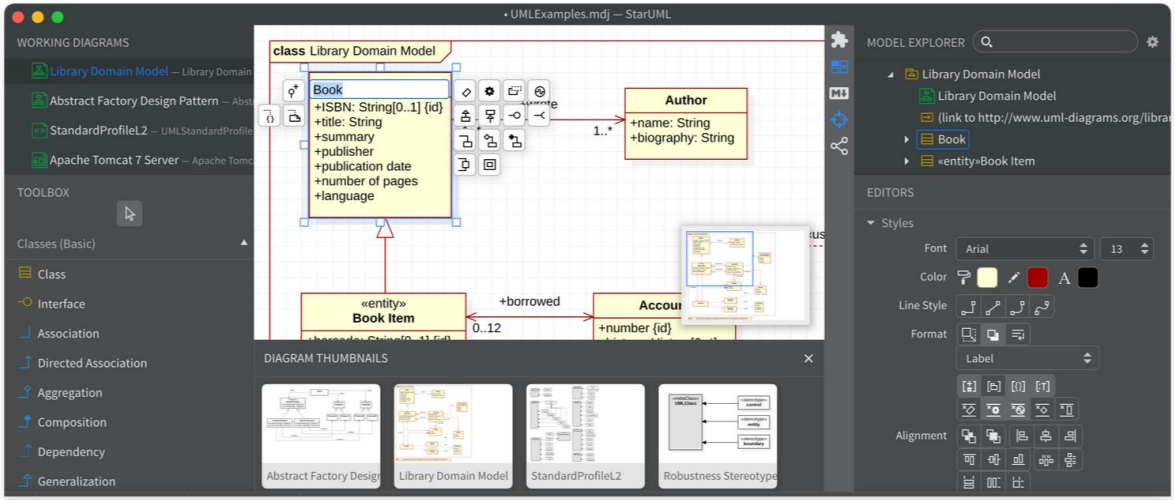
En UML no es posible una jerarquía en ciclo.



starUML

Realiza el chequeo de algunas reglas definidas en el metamodelo

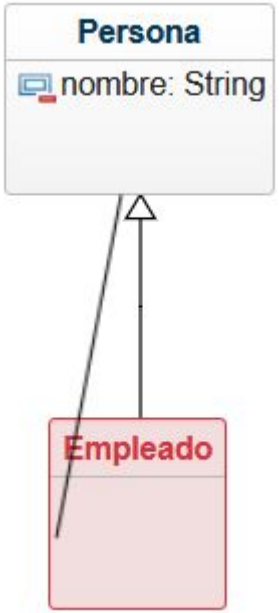
Versión instalable

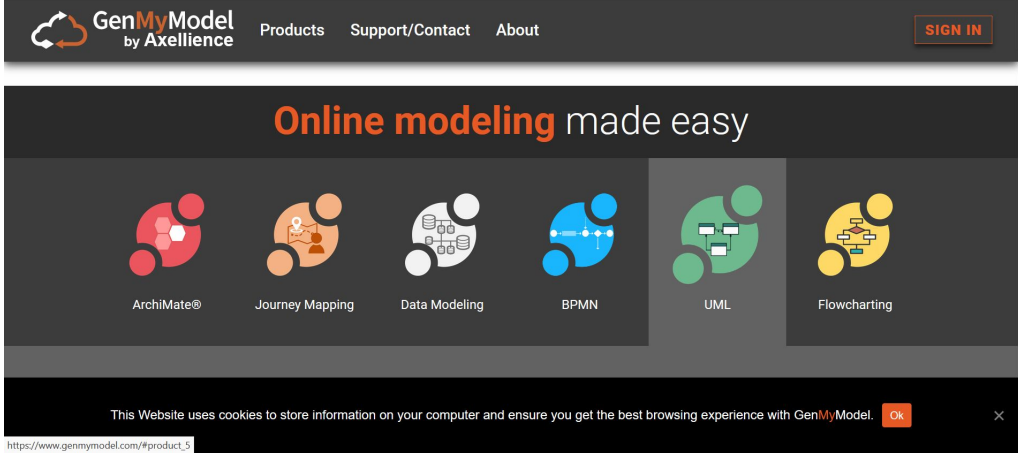


GenMyModel

Realiza el chequeo de algunas reglas definidas en el metamodelo

Versión online





En UML no es posible una jerarquía en ciclo.

No es posible dibujarlo.

Para repasar

- Sitio oficial UML - https://www.uml.org/
- Documento de especificación - Metamodelo (https://www.omg.org/spec/UML/2.5/PDF)

- Grady Booch Reflects on UML 1.1 20th Anniversary (LINK)
- PlantUML (https://plantuml.com/es/)
- GenMyModel (https://app.genmymodel.com)

¿ Preguntas ?