

# Ingeniería de Requisitos

---

## **Tema 3b. Modelado de Casos de Uso en Rational Unified Process (RUP)**

*Departamento de Lenguajes y Sistemas Informáticos  
Universidad de Alicante*

*Materiales cedidos para su uso docente por el prof. Emilio Insfran Pelozo. Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia*

# Contenido

## Parte I

- RUP y tipos de requisitos
- Tipos de Requisitos y trazabilidad
- Características y Casos de Uso

## Parte II

- Casos de Uso. Definición
- Actores
- Relaciones
- Fallos comunes

# Introducción

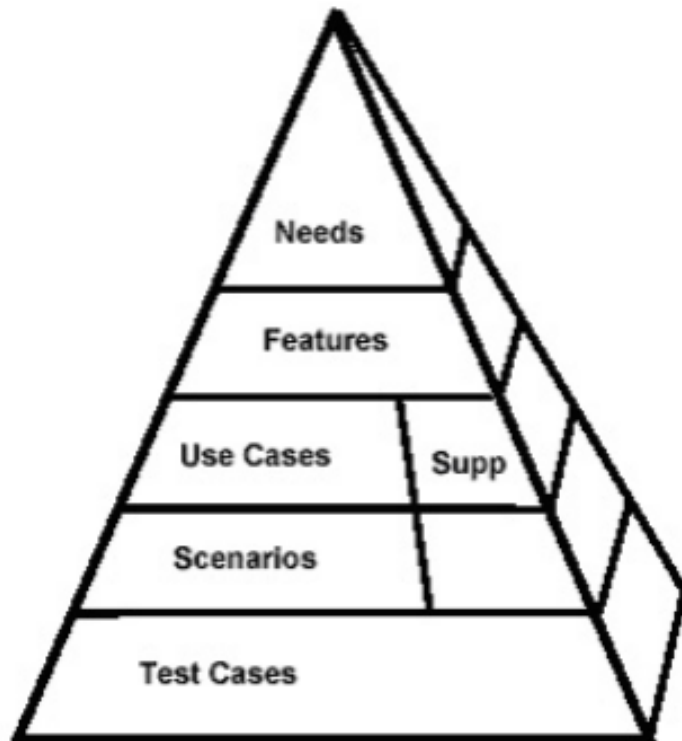
- RUP es el principal proceso de desarrollo de software basado en UML
- RUP es una aproximación *dirigida por casos de uso* (significa que los casos de uso son la base de TODO el proceso de desarrollo)

## Principales tipos de Requisitos:

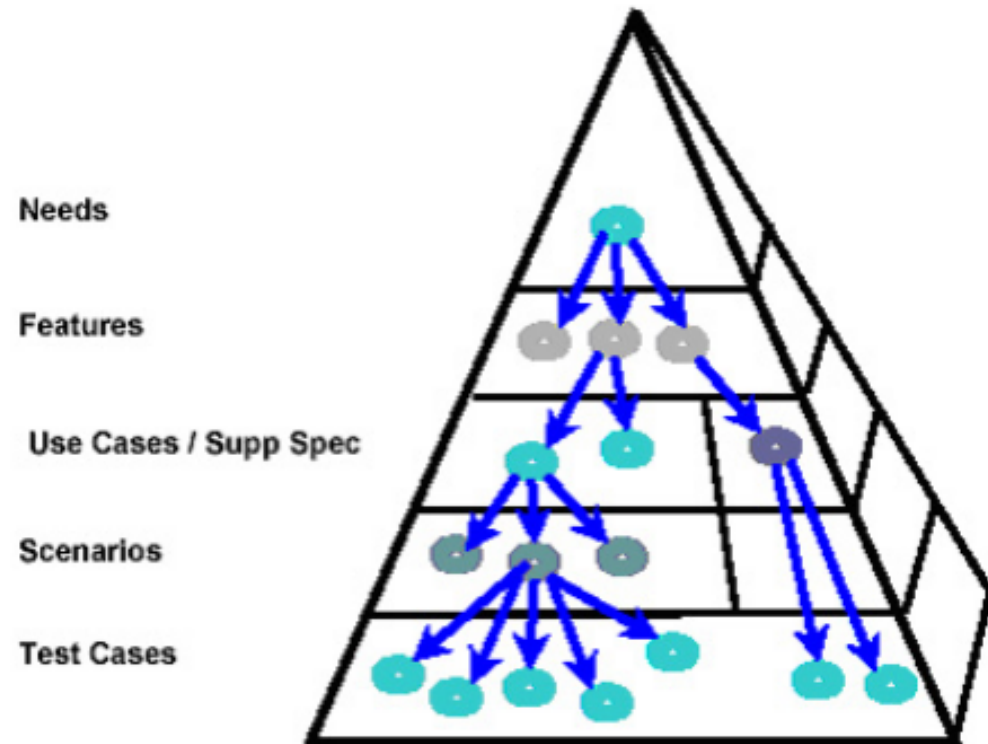
- Features (Características)
  - representan requisitos de alto nivel del sistema software
  - en RUP se consideran como un tipo de requisito debido a que son fáciles de discutir (lenguaje natural), documentar y comunicar
- Requisitos Funcionales
  - los *features* necesitan mayor nivel de detalle que se puedan diseñar, implementar y verificar. En RUP los RF son los Casos de Uso!!!
- Requisitos No Funcionales (también llamados suplementarios)
  - especificaciones textuales *asociadas* a los casos de uso

# Introducción. Tipos de Requisitos y Trazabilidad

## Pirámide de Requisitos



## Trazabilidad en la Pirámide de Requisitos



**Needs:** Necesidades de usuarios o del negocio (objetivos)

**Features** (características). Funcionalidad de alto nivel. Agrupa funcionalidades más específicas

**Use cases:** Casos de Uso, son los Requisitos Funcionales

**Supp:** Requisitos No Funcionales (RNF) o Requisitos suplementarios

**Scenarios:** descripción de la funcionalidad de RF o casos de uso

**Test cases:** Pruebas que se realizan para comprobar los RF y RNF

# Introducción. Tipos de Requisitos y Trazabilidad.

## Ejemplo

### Ejemplo: Sistema de Videoclub

- *Gestión de películas*
  - Alta de película
  - Modificar datos
  - *Administrar copias*
    - Crear copia
    - Borrar copia
- *Gestión de socios*
  - Alta de socio
  - Modificar datos
  - Baja de socio
- *Gestión de alquileres*
  - Alquilar película
  - Devolver película

**Características:** son funcionalidades de alto nivel que agrupan funcionalidades más específicas (casos de uso). Pueden anidarse a varios niveles. *Ejemplo:* Administrar copias.

**Caso de Uso:** en RUP es sinónimo de Requisito Funcional. Representa una funcionalidad del sistema.



## Parte II

# Casos de Uso

- Se utilizan para especificar el comportamiento deseado de un sistema o subsistema
  - Su especificación describe el conjunto de **secuencias de acciones** que se lleva a cabo en el sistema para producir un resultado para un **actor**
  - Capturan el comportamiento deseado del sistema, sin especificar *cómo* se lleva a cabo dicho comportamiento
- Principalmente son un medio de comunicación para que los desarrolladores y los usuarios lleguen a un consenso en la especificación
- Ayudan a **validar y verificar** el sistema durante el desarrollo

# Casos de Uso

- Los casos de uso son principalmente descripciones *textuales* de requisitos (guiones/escenarios)
- La notación gráfica de UML (**Diagrama de Casos de Uso**) sólo muestra los nombres y sus relaciones
- Al ser textuales, son informales y no son buenas para razonar acerca de la estructura del sistema (no para el *cómo* pero sí para el *qué*)
  - Para el *diseño de alto nivel* es mejor utilizar los diagramas de interacción para analizar las interacciones internas de objetos (interacciones necesarias para realizar un caso de uso).



# Casos de Uso

- Formas de especificar Casos de Uso
  - Guión del caso de uso: *narrativa, escenario*
  - Alternativas/Complemento (realización del caso de uso): *diagramas de interacción, diagramas de actividades*
  - **Narrativa:**
    - Nombre de Caso de Uso: xxx
    - Descripción: xxx
  - Guión genérico (**escenario**):
    1. El actor envía al sistema una petición y los datos necesarios para llevarla a cabo
    2. El sistema valida la petición y los datos
    3. El sistema altera su estado interno
    4. El sistema devuelve el resultado al actor

# Casos de Uso. Plantilla

<b>Referencia:</b>	
<b>Nombre:</b>	Caso de uso 1
<b>Descripción</b>	
<b>Actor</b>	
<b>Relaciones</b>	
<b>Precondición</b>	
<b>Comentarios</b>	

# Casos de Uso. Definición

Un caso de uso es...

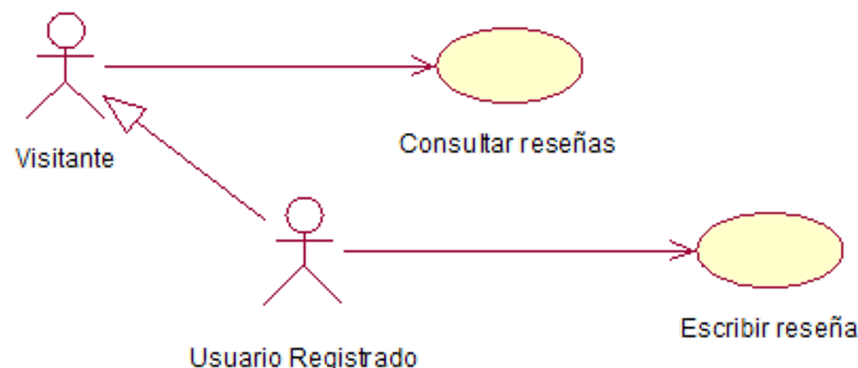
una descripción de la secuencia de interacciones  
con los *actores* externos y dentro del sistema  
para realizar una funcionalidad determinada

Una funcionalidad → Un caso de uso

Un grupo de funcionalidades → Una característica

# Casos de Uso. Actores

- Un **actor** representa un conjunto coherente de roles que los usuarios de los casos de uso representan al interactuar con el sistema
  - Normalmente representan a una persona, un dispositivo hardware u otro sistema al interactuar con el nuestro
- Se pueden definir categorías generales de actores y especializarlos a través de las relaciones de *generalización*
- Los actores se conectan a los casos de uso mediante *asociaciones*



# Casos de Uso. Relaciones

- Los casos de uso se pueden organizar en paquetes
- Se pueden especificar relaciones entre casos de uso:
  - Generalización (no es muy utilizado)
  - Extensión
  - Inclusión
- Estas relaciones se usan para:
  - Factorizar el comportamiento común
    - extrayendo un comportamiento de los casos en que se incluye
  - Factorizar variantes
    - poniendo ese comportamiento en otros casos de uso que lo extienden
- Habitualmente estas relaciones aparecen en fases avanzadas de la especificación!

# Casos de Uso. Relaciones

- Generalización (no es muy utilizado)

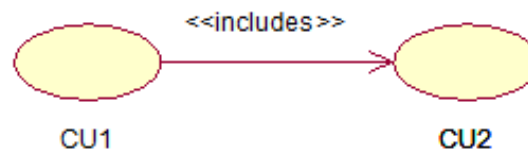
- El caso de uso hijo hereda el comportamiento y el significado del caso de uso padre.
- El hijo puede añadir o redefinir el comportamiento del padre.
- El hijo se puede colocar en cualquier lugar en que aparezca el padre.



Ejemplo: 'CU - Hijo' hereda de 'CU - Padre'.

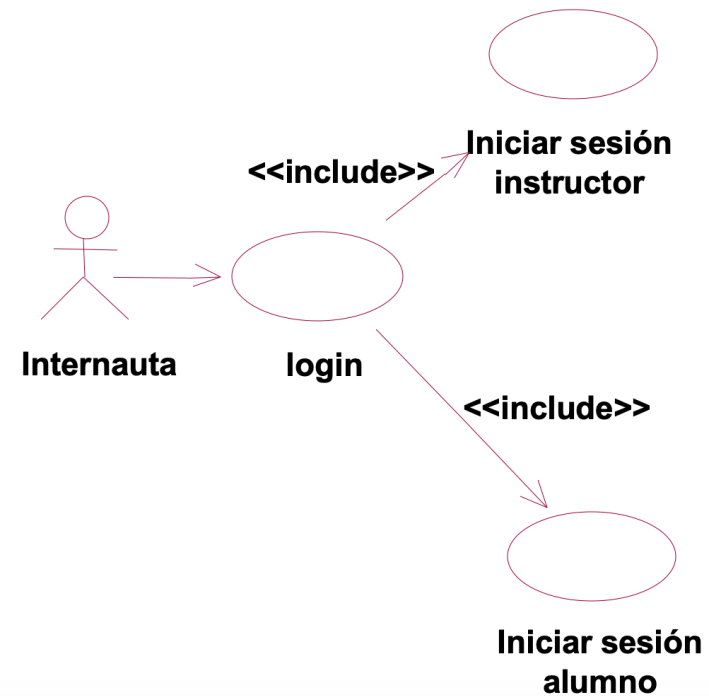
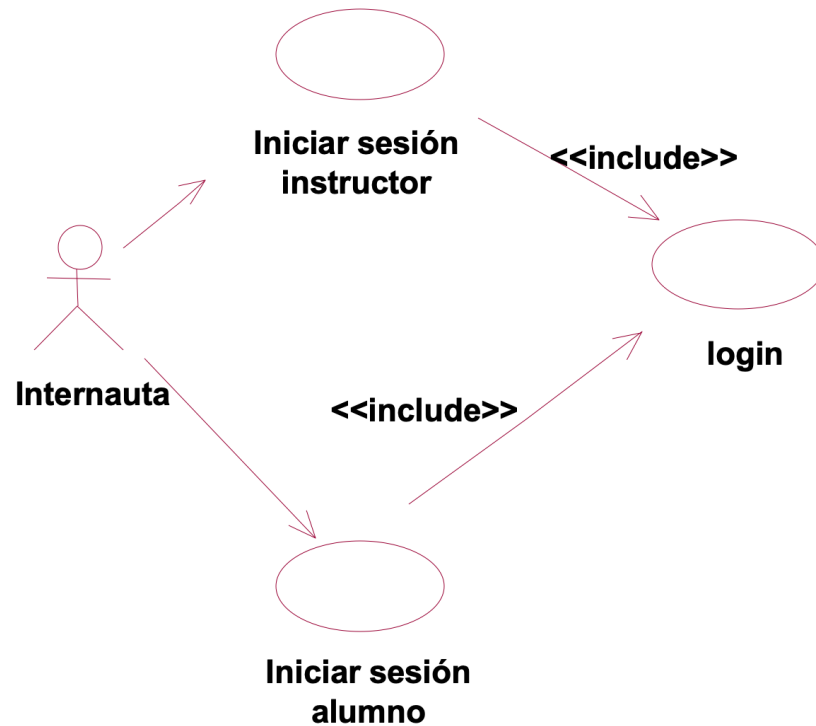
- Inclusión

- El caso de uso base incorpora *explícitamente* el comportamiento del caso de uso incluido.
- El caso de uso incluido forma parte de otro más complejo.
- Se suele utilizar para evitar describir flujos repetidos.
- La inclusión entre casos de usos se nombra (o estereotipa) con **<<includes>>**



Ejemplo: CU1 (CU base) incluye la funcionalidad de CU2 (CU incluido).

# Casos de Uso: Relaciones. Ejemplo <<include>>



# Casos de Uso: Relaciones <<extends>>

- Extensión

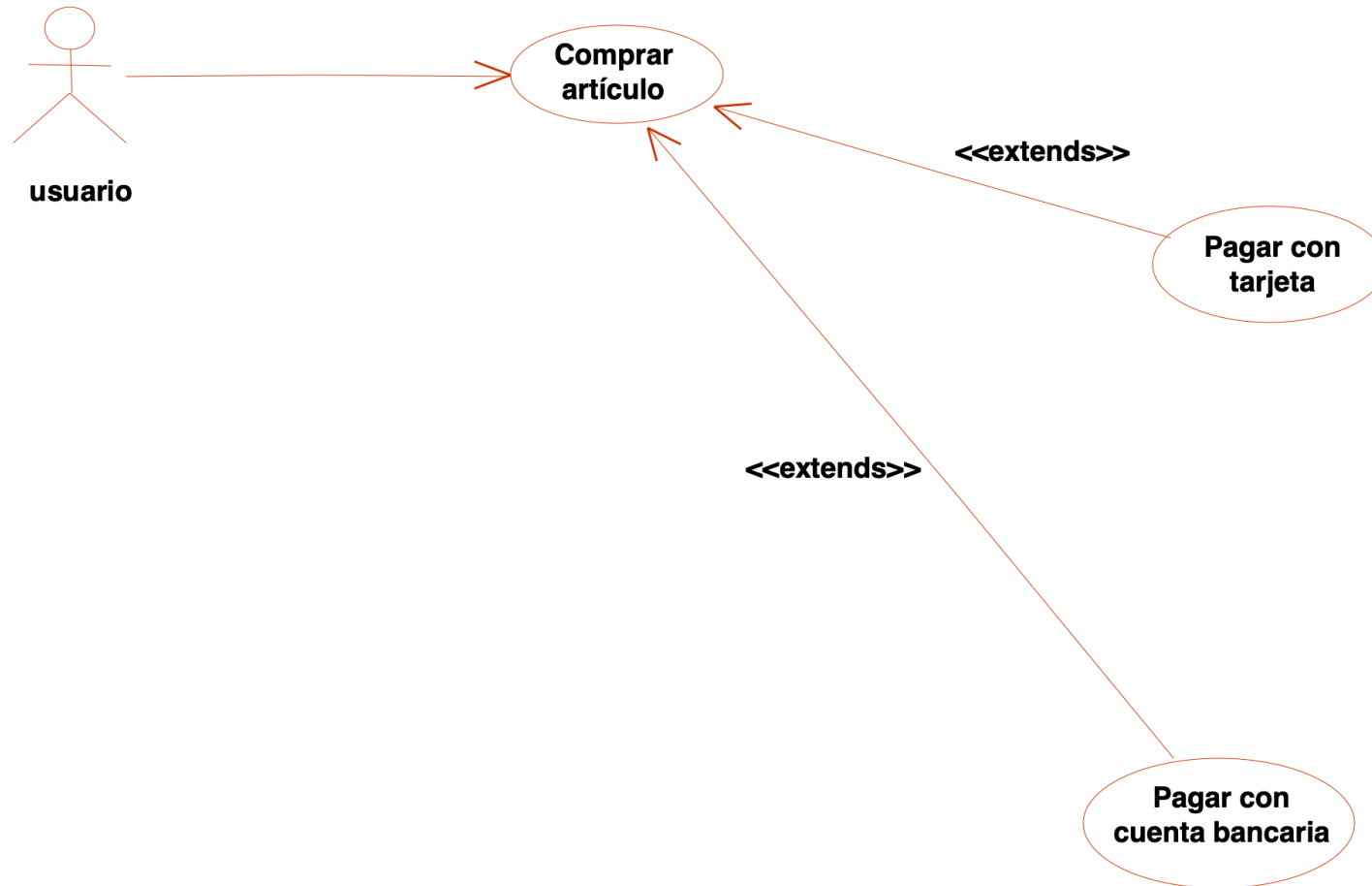
- Se utiliza para modelar la parte de un caso de uso que puede ser vista como un comportamiento opcional
- También se pueden utilizar para modelar un subflujo separado que sólo se ejecuta bajo ciertas condiciones
  - Un ejemplo es el modelado de varios flujos que se puedan dar en un punto dado por la interacción explícita con un actor
- Para añadir comportamiento a un Caso de Uso ya existente
- La extensión entre casos de usos se nombra (o estereotipa) con <<extends>>. (Atención al sentido de la asociación)



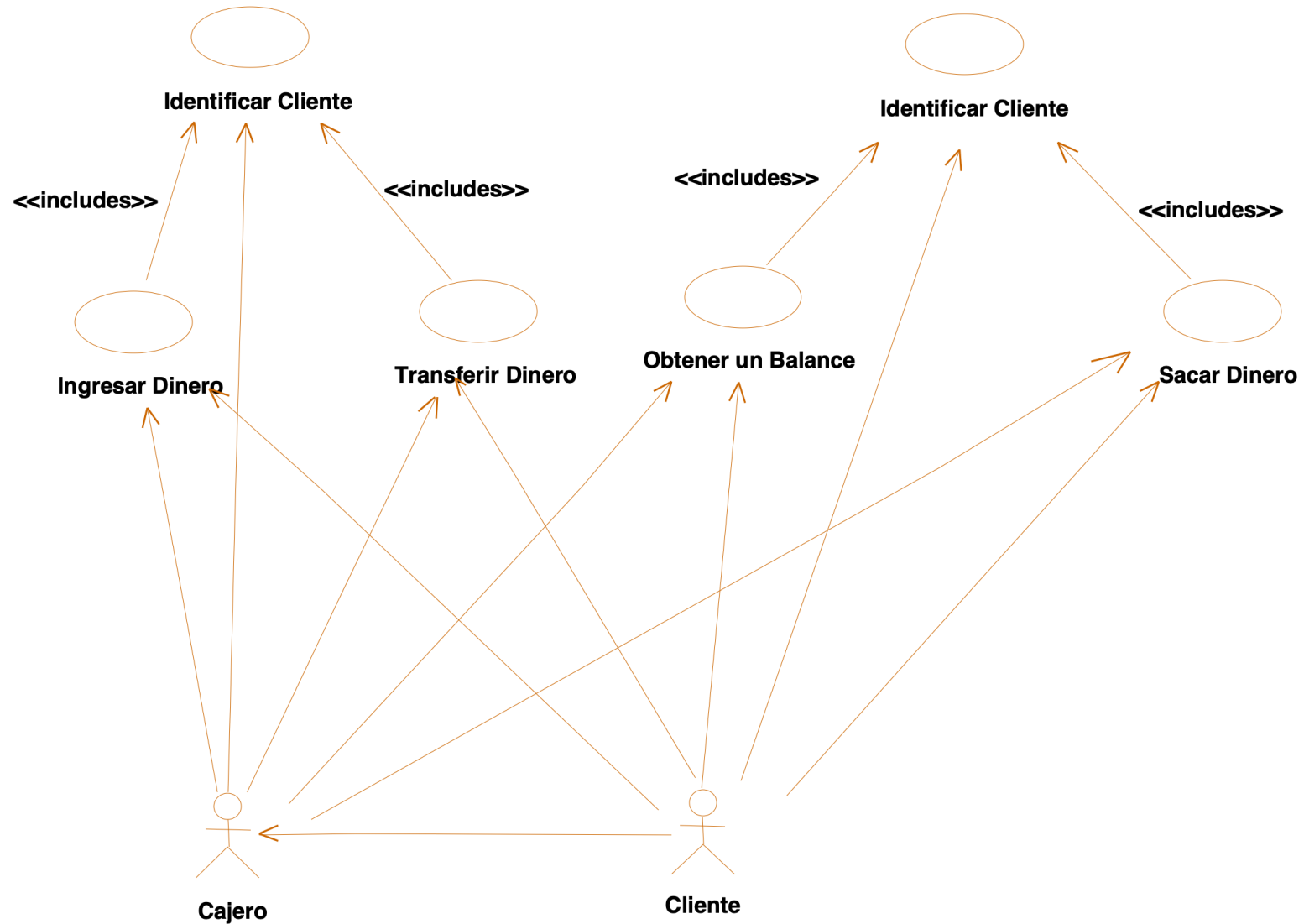
Ejemplo: CU2 extiende la funcionalidad de CU1



# Casos de Uso: Relaciones. Ejemplo <<extends>>



# Casos de Uso: Ejemplo



# Casos de Uso: Ejemplo

- Identificación de niveles del diagrama anterior:
  - Funciones de Usuario (Casos de Uso)
    - Ingresar Dinero
    - Transferir Dinero
    - Obtener un Balance
    - Sacar Dinero
  - Subfunciones (Sub Caso de Uso)
    - Identificar Cliente

# Casos de Uso: Resumen

- Los diagramas de Casos de Uso pueden ser vistos como un *mapa* general de las funcionalidades ofrecidas por el sistema
- Debe incluir TODAS las interacciones externas entre los actores y el sistema
- Deben ser lo suficientemente claros para que alguien externo al sistema los entienda
- Los casos de uso se usan para:
  - **Modelar el contexto del sistema**
    - Especificar las fronteras e identificar los actores
    - Identificar subsistemas
  - **Modelar los requisitos del mismo**
    - Especificar *qué* debe hacer el sistema desde el punto de vista externo (no *cómo*)

# Casos de Uso: Fallos Comunes

1. Límites
2. Punto de vista (analista, diseñador, IU!!)
3. Actores
4. Pocos CU (casos de uso muy grandes)
5. Demasiados CU (casos de uso muy pequeños)
6. Telas de araña
7. Especificaciones demasiado largas
8. Especificaciones confusas
9. Descripción funcional errónea
10. Ininteligibles
11. Inacabados

# Casos de Uso: Glosario

- **Actor**: Algo o alguien que envía o recibe (interacciona) con el sistema.
- **Actor principal**: Actor que inicia una interacción con el sistema para lograr un objetivo.
- **Caso de uso**: Una funcionalidad concreta del sistema.
- **Característica**: Agrupaciones de casos de uso.
- **Límite**: Delimita las funciones a ser soportadas por el sistema.
- **Precondiciones** y **Postcondiciones**: Lo que debe ser cierto antes y después de la ejecución del caso de uso.
- **Escenario principal**: El caso en que todo sale bien.
- **Escenario secundario**: Variantes del escenario principal.
- **Relaciones**: Relación entre casos de uso, en la especificación se utiliza para decir que un caso de uso llama a otro (uses), o que es la extensión de otro (extends).