

---

## **Tema 2.**

# **Análisis y Diseño Orientado a Objetos mediante UML**

### Índice

1. Introducción a la OO
2. Introducción a UML
3. Planificación (Casos de Uso)
4. Construcción: Análisis y Diseño Orientado a Objetos con UML
5. Proceso Unificado de Desarrollo de Software
6. Bibliografía

# 1. Introducción a la OO

---

Cuanto más complejo es un sistema a desarrollar,



**más ventajas del desarrollo O.O.**

*Reutilización*

*Resistencia a los cambios*

*Flexibilidad*

*Objetos más complejos*

*Menos código*

*Confianza*

*Mantenimiento más fácil*

*Mejor comunicación U -- I.Sw*

*Modularidad*

*Reduce la complejidad*

***Necesidad de usar una metodología***

# 1. Introducción a la OO

---

- Primeras recomendaciones de diseño (**Booch**, 1986)
- Modelo orientado a las caract. de los objetos (**Shlaer y Mellor**, 1988)
- Modelo de Análisis O.O. De Booch, 1989
- Surgen otros métodos de varios autores (**Coad & Yourdon, Martin & Odell, Rumbaugh, Embley, etc.**, 1990 y ss.)
- Métodos orientados al comportamiento de los O. (**Wirfs-Brock, Jacobson, Rubin & Goldberg**, 1994)
- Comienza a gestarse UML (1994)

***Proceso Unificado (**Rational Unified Process**)***

## Ciclo de Vida Orientado a Objetos

1. Planificación
2. Análisis Orientado a Objetos (AOO)
3. Diseño Orientado a Objetos (DOO)
4. Construcción o Implementación
5. Prueba y Mantenimiento

*Nos ayudaremos de UML (**Unified Modeling Language**)*

## 2. Introducción a UML

---

- UML es un lenguaje estándar (OMG, [www.omg.org](http://www.omg.org)) para representar “planos” (footprints) software
- Los planos software se representan mediante distintas vistas de un mismo modelo
- Una vista es una proyección de la organización y estructura de un sistema, centrada en un aspecto particular del mismo
- UML define una notación: material gráfico para representar cada una de las vistas del modelo; es la sintaxis del lenguaje UML
- UML es por lo tanto un lenguaje para modelar un sistema real (análisis) o software (diseño)

## 2. Introducción a UML

---

- UML es un lenguaje de modelado de software:
  - Proporciona un vocabulario y reglas para crear modelos software
  - Suficientemente expresivo para cubrir distintas vistas de la arquitectura del software a lo largo del ciclo de vida
  - Mayor nivel de abstracción que un lenguaje de programación
- UML es un lenguaje para visualizar los elementos de un gran sistema software, facilitando:
  - la comunicación entre los participantes (incluidas herramientas) en el desarrollo
  - la comprensión de las soluciones (notación gráfica)
  - el mantenimiento de las soluciones conceptuales a lo largo del tiempo (documentación)

## 2. Introducción a UML

---

- UML es un lenguaje para especificar software:
  - Se pueden construir modelos precisos, no ambiguos y completos
  - Cubre las decisiones de análisis, diseño e implementación
- UML es un lenguaje para construir software:
  - No es un lenguaje de programación visual, pero sus modelos se pueden conectar de forma directa a una gran variedad de ellos
  - Correspondencias entre UML y lenguajes: Java, C++, etc.
  - Ingeniería directa: generación de código
  - Ingeniería inversa: reconstrucción de modelos
- UML es un lenguaje para documentar:
  - requisitos, arquitectura, diseño, código fuente, pruebas, ...

## 2. Introducción a UML

---

- UML **NO** es un método: “proceso disciplinado para generar un conjunto de modelos que describen varios aspectos de un sistema software en desarrollo, utilizando una notación bien definida” [Booch99]
- UML **NO** es una metodología: “colección de métodos a lo largo del ciclo de vida del desarrollo de software unificados por alguna aproximación general” [Booch99]
- El lenguaje UML **NO** es un proceso de desarrollo software: “conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software” [Jacobson99]



## 2. Introducción a UML

---

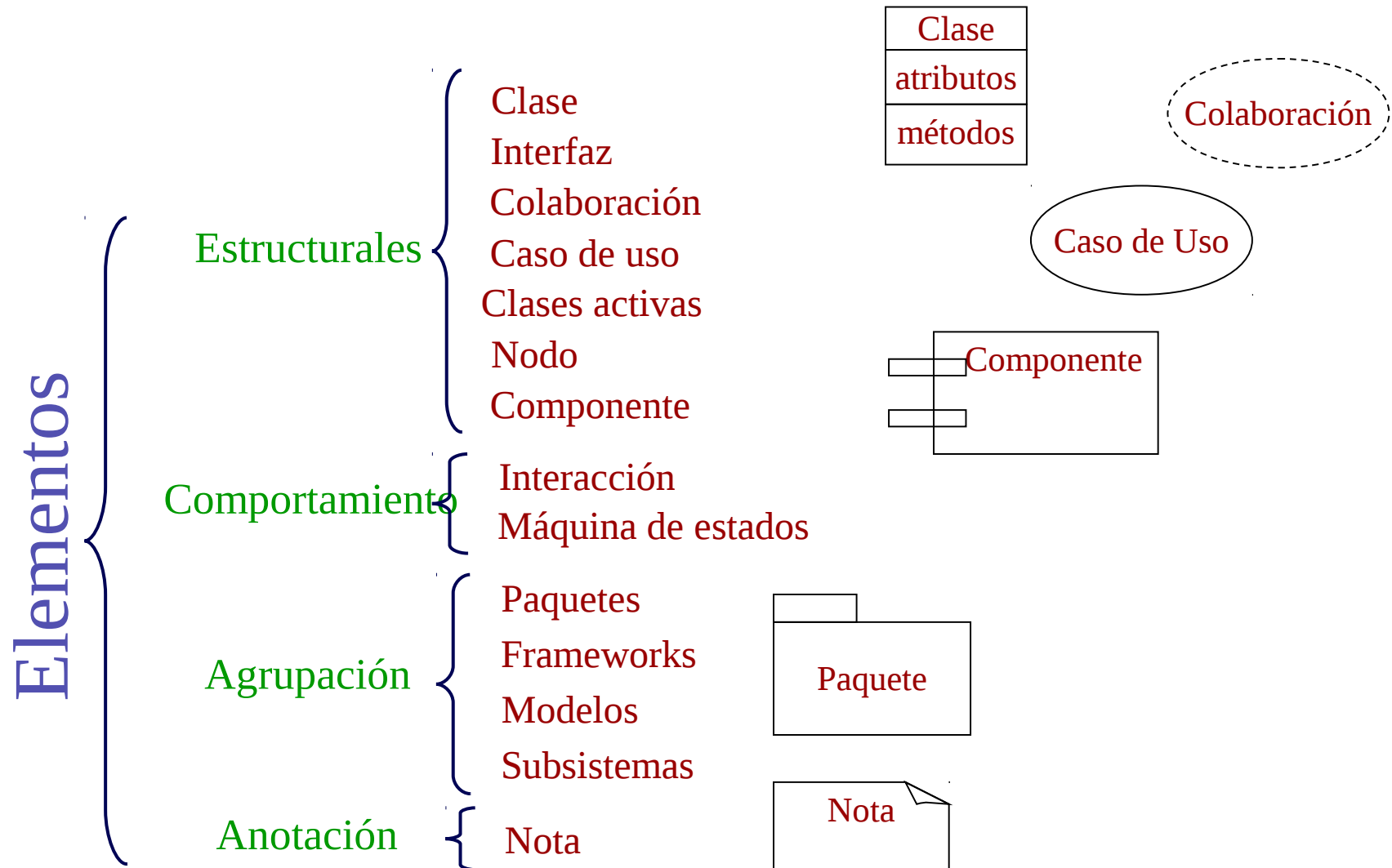
- Por tanto UML nos visualiza, describe distintas vistas de un mismo sistema, donde cada vista nos muestra un aspecto distinto del sistema.
- Los modelos de UML que se tratan en esta tema son los siguientes:
  - Diagrama de Casos de Uso
  - Diagrama de Estructura Estática
  - Diagrama de Secuencia
  - Diagrama de Colaboración
  - Diagrama de Estados

## 2. Introducción a UML

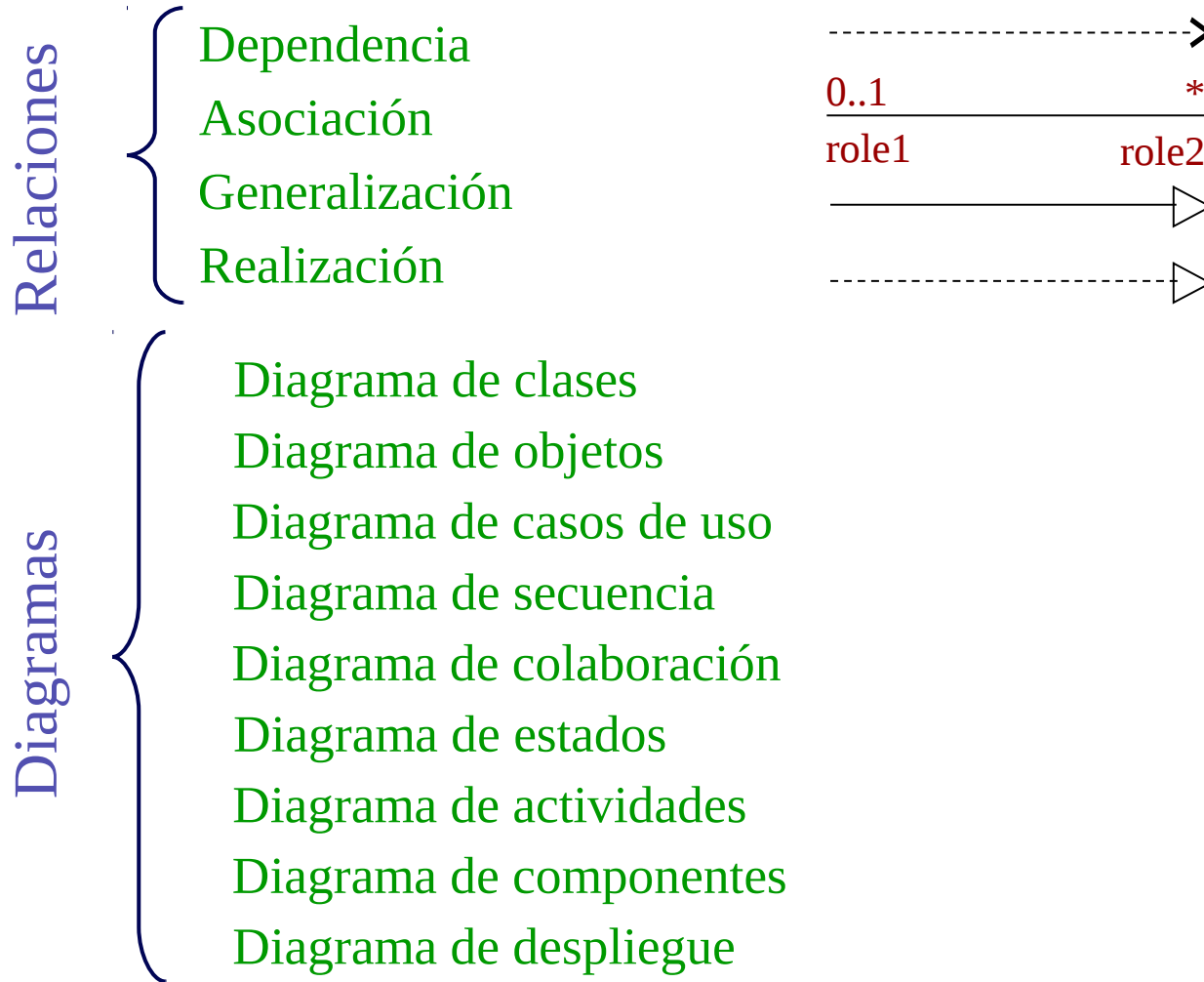
---

- UML esta compuesto:
  - **Elementos:** abstracciones básicas a partir de las que se construyen los modelos
    - Elementos estructurales
    - Elementos de comportamiento
    - Elementos de agrupación
    - Elementos de anotación
  - **Relaciones** entre los elementos
    - Asociación
    - Dependencia
    - Generalización
    - Realización
  - **Diagramas:** Grupo consistente de elementos y sus relaciones
    - Clases, objetos, casos de uso, secuencia, colaboración, estados, componentes, despliegue.

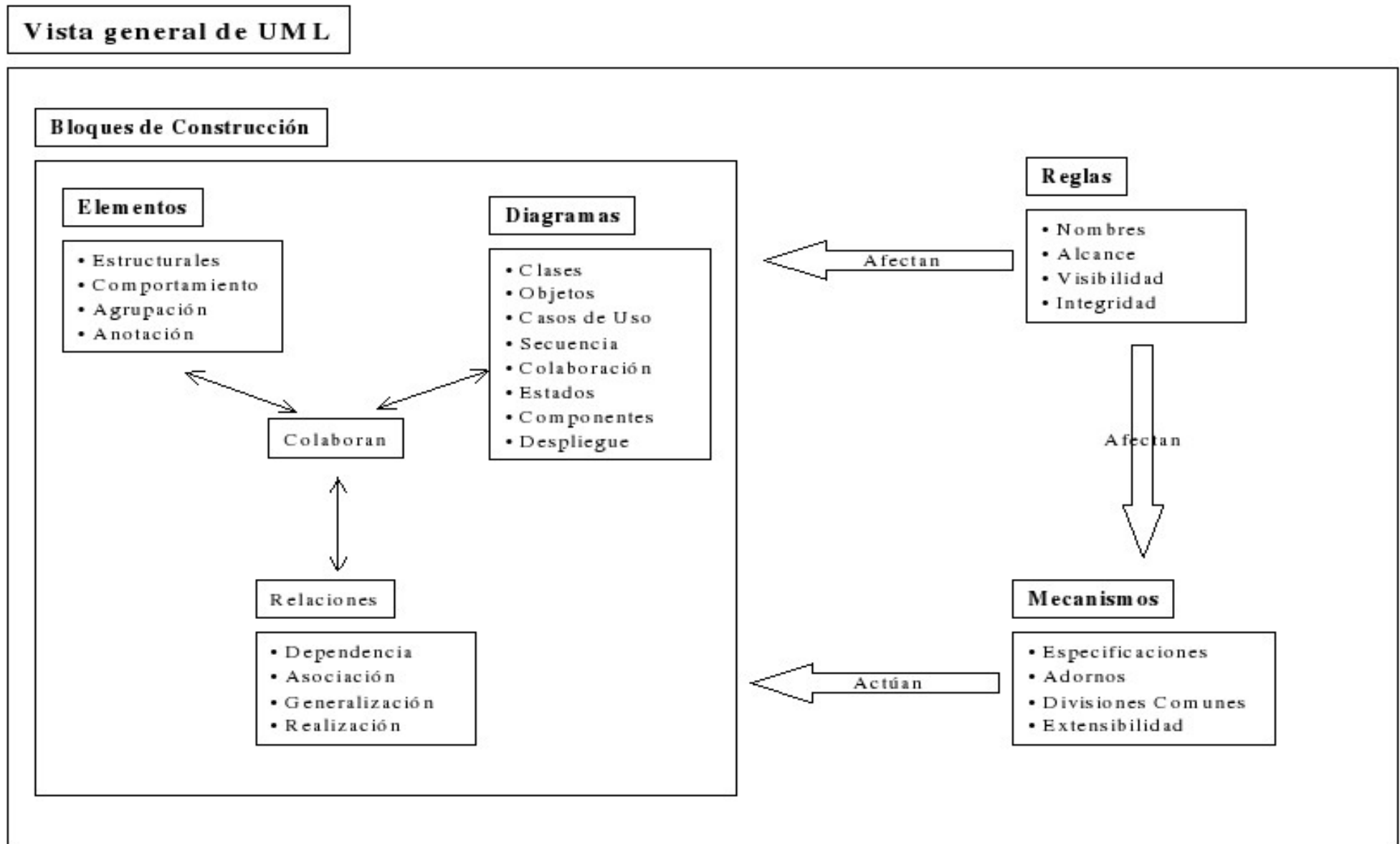
## 2. Introducción a UML



## 2. Introducción a UML



## 2. Introducción a UML



## 2. Introducción a UML

---

### ➤ Herramientas de UML

- ❖ de Requisitos

  - Casos de Uso

- ❖ de Modelado estático de la estructura

  - Diagramas de Clases

- ❖ de Administración de sistemas complejos

  - Diagramas de Paquetes

- ❖ de Modelado de dinámica del comportamiento

  - Diagramas de Estados, de Actividad y de Perspectiva

- ❖ de Interacción

  - Diagramas de Colaboración y de Secuencia (varios tipos)

- ❖ de Implementación

  - Diagramas de Componentes y de Despliegue

- Necesidad de utilizar una metodología de desarrollo (proceso software)
- Llevado a cabo mediante 5 etapas, donde abordaremos las dos primeras: Análisis y Diseño
  - Análisis: ¿qué debe hacer nuestro sistema?
  - Diseño: ¿Cómo lo hace?
- Para modelar, se usará el lenguaje UML, formado por un conjunto de diagramas compuestos por elementos y sus relaciones. Importante:
  - La misma notación puede utilizarse para análisis y para diseño
  - Depende de qué tipo de entidades representemos (problema o solución)

## ➤ Planificación

- ✓ Identificación. Título, descripción, versión, fecha,..
- ✓ Documento de análisis
- ✓ **Especificación de requisitos o requerimientos**
- ✓ **Diagramas de casos de uso**
- ✓ **Escenarios y sub-escenarios**

## ➤ Análisis y Diseño OO

- ❖ Clases, Objetos y mecanismos de colaboración
  - ✓ Diagramas de interacción (**secuencia**, colaboración)
  - ✓ **Diagramas de Clases** y patrones de diseño.
  - ✓ Diagramas de objetos
- ❖ Modelado del comportamiento de clases y objetos
  - ✓ Diagramas de actividades
  - ✓ Diagramas de estados
- ❖ Construcción del modelo físico
  - ✓ Diagramas de componentes
  - ✓ Diagramas de despliegue

## ➤ Implementación

## ➤ Prueba y Mantenimiento: Informes de errores y Nueva especificación



## 3. Planificación

---

3.1 Identificación.

3.2 Documento de análisis

3.3 Especificación de requisitos o requerimientos

3.4 Diagramas de casos de uso

3.5 Escenarios y sub-escenarios

3.6 Prototipos

## 3. Planificación. 3.1 Identificación y 3.2 Análisis del documento

---

- 3.1 Identificación y Objetivos
  - Es necesario identificar todos los elementos del proceso de desarrollo de software de una forma unívoca
  - Todos los documentos deben estar identificados
    - Título, Descripción, Autores, Versión, Revisión. Autores, Fecha, Código de cada documento o diagrama
- 3.2 Análisis del documento para la obtención de requisitos
  - ❖ Contiene
    - ✓ la documentación que aporta el cliente que encarga la aplicación
    - ✓ Entrevistas realizadas
    - ✓ Software previo
    - ✓ Auditorias de trabajos
    - ✓ ....

## 3. Planificación. 3.3 Especificación de Requisitos

### ➤ Especificación de requisitos:

- ❖ La captura de requisitos es complicada
  - ✓ Los usuarios habitualmente no saben expresar exactamente lo que quieren
  - ✓ Es difícil tener una visión global del problema a resolver
- ❖ La especificación de requisitos es un documento más técnico y elaborado de los documentos de análisis
- ❖ Es importante codificar los requisitos para poder seguirlos a lo largo del proceso de desarrollo de software

### ➤ Tipos de requisitos

- ❖ **Requisitos funcionales:** Se hace con casos de uso.
- ❖ **Requisitos no funcionales.** Por ejemplo:
  - ✓ Requisito de interfaz: Especifica como se relaciona con un elemento externo.
  - ✓ Requisito físico: Características como forma, tamaño, peso, etc.
  - ✓ Restricción de diseño: Extensibilidad, mantenibilidad o re-utilización.
  - ✓ Restricción de implementación: Normas de codificación, lenguaje, etc.

## 3. Planificación. 3.3 Especificación de Requisitos

---

- **Ejemplo:**

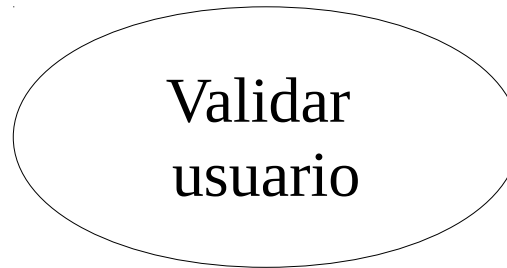
- R.0 Requisitos generales
  - R.0.1 Fechas que codifiquen el año con cuatro cifras y con separador “/”.
  - R.0.2 Las unidades monetarias serán con cifras decimales (euro)
  - R.0.3 La aplicación se ejecutará
    - en TPV táctiles (pantallas inferiores a 12”)
    - También debe existir una aplicación para equipos de sobremesa
- R.1 Gestión de clientes
  - R.1.0 Requisitos generales de los clientes
    - R.1.0.1 Clientes pueden ser fijos o eventuales
    - R.1.0.2 Clientes tienen un n° identificativo
    - R.1.0.3 Los clientes se definen por D.N.I., nombre, dirección, teléfono

## 3. Planificación. 3.4 Casos de usos

---

### ➤ Caso de uso

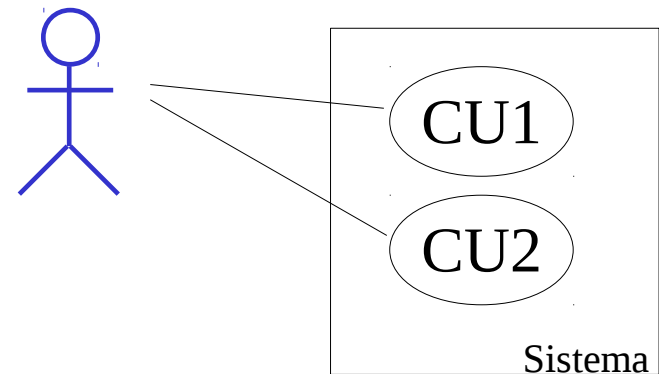
- ❖ **Descripción de un conjunto de secuencias de acciones que ejecuta un sistema para producir un resultado observable de valor para un actor**



- ### ➤ Los objetivos de los casos de uso son los siguientes:
- ❖ Capturar los requisitos funcionales del sistema y expresarlos desde el punto de vista del usuario
  - ❖ Guiar todo el proceso de desarrollo del sistema de información

### 3. Planificación. 3. 4 Casos de usos.

- Los casos de uso proporcionan, por tanto, un modo claro y preciso de comunicación entre cliente y desarrollador
  - ❖ Desde el punto de vista del cliente proporcionan una visión de “caja negra” del sistema, esto es, visión del sistema desde el exterior sin necesidad de entrar en los detalles de su construcción
  - ❖ Para los desarrolladores, suponen el punto de partida y el eje sobre el que se apoya todo el desarrollo en sus procesos de análisis y diseño
- Especificación de requisitos de los casos de uso



## 3. Planificación. 3.4 Diagrama de Casos de Uso

### ➤ Formado por

#### ❖ Casos de uso

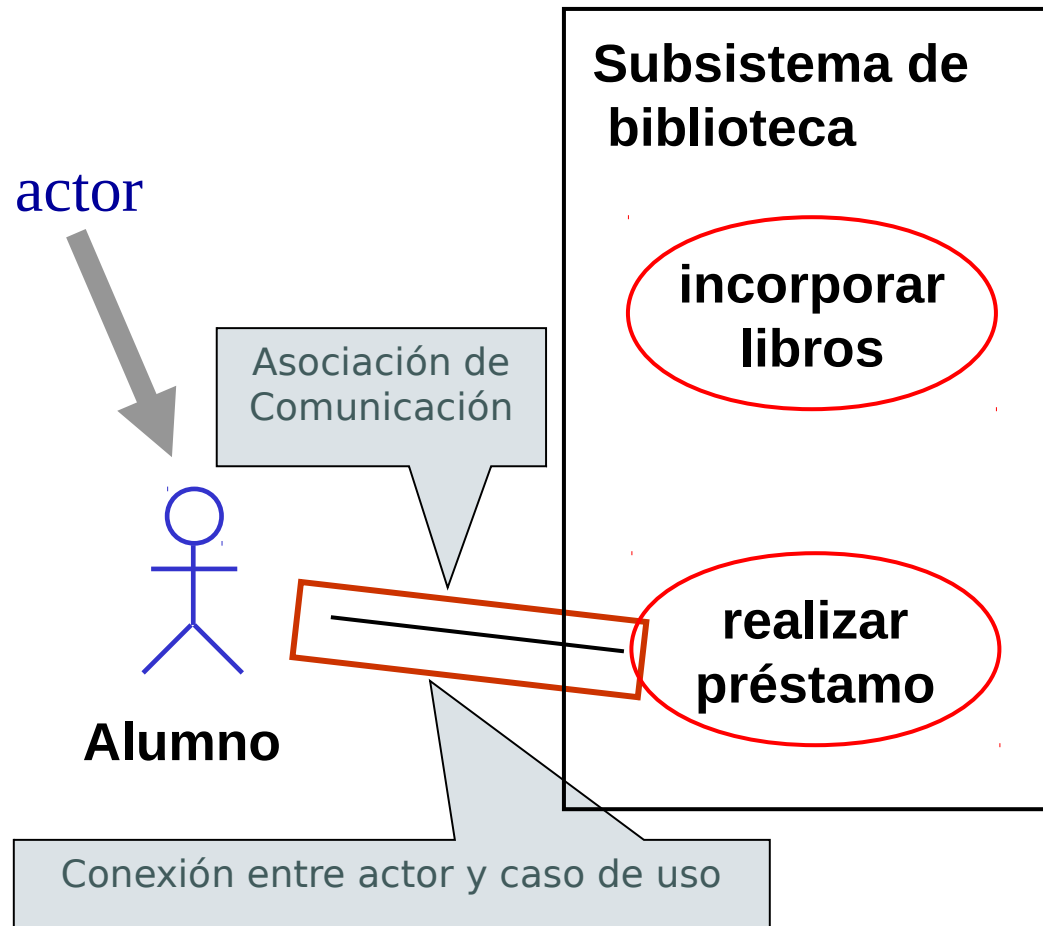
❖ Actores. Un actor es algo o alguien que se encuentra fuera del sistema y que interactúa con él. En general, los actores serán los usuarios del sistema y los sistemas externos.

#### ❖ Relaciones:

- ✓ Asociación: Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple.
- ✓ Relación obligatoria entre dos casos de uso. Describe que para la ejecución de un caso de uso es necesaria la ejecución de otro. Se representa con una flecha discontinua que los une, nominada con la secuencia de caracteres <<include>> o <<uses>>.
- ✓ Generalización: Este tipo de relación es uno de los más utilizados, cumple una doble función dependiendo de su estereotipo <<extends>> .
  - extends: Se recomienda utilizar cuando un caso de uso es similar a otro (característica).

### 3. Planificación. 3.4 Diagrama de Casos de Uso

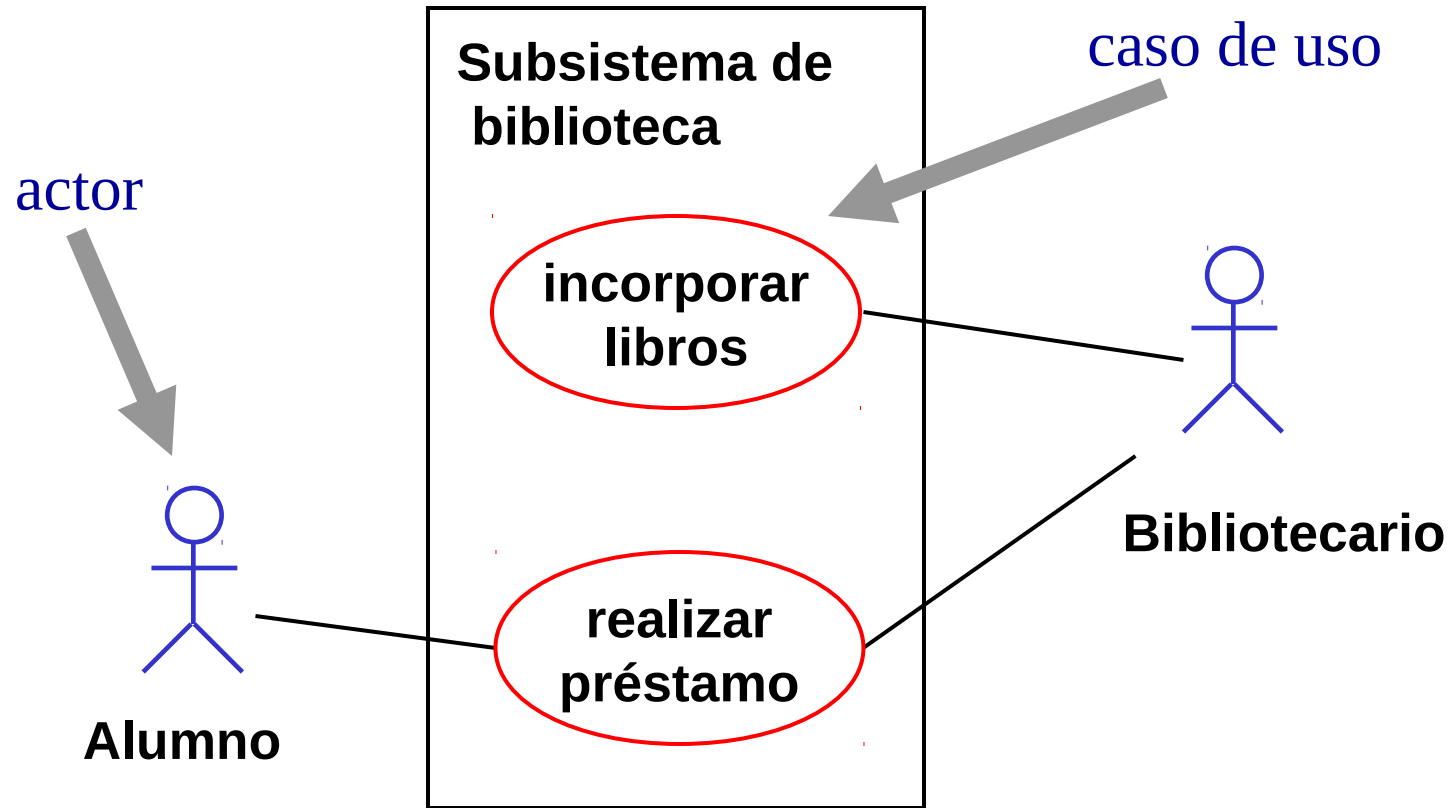
#### ➤ Diagrama de caso de uso





### 3. Planificación. 3.4 Diagrama de Casos de Uso

#### ➤ Diagrama de caso de uso



### 3. Planificación. 3.4 Diagrama de Casos de Uso

---

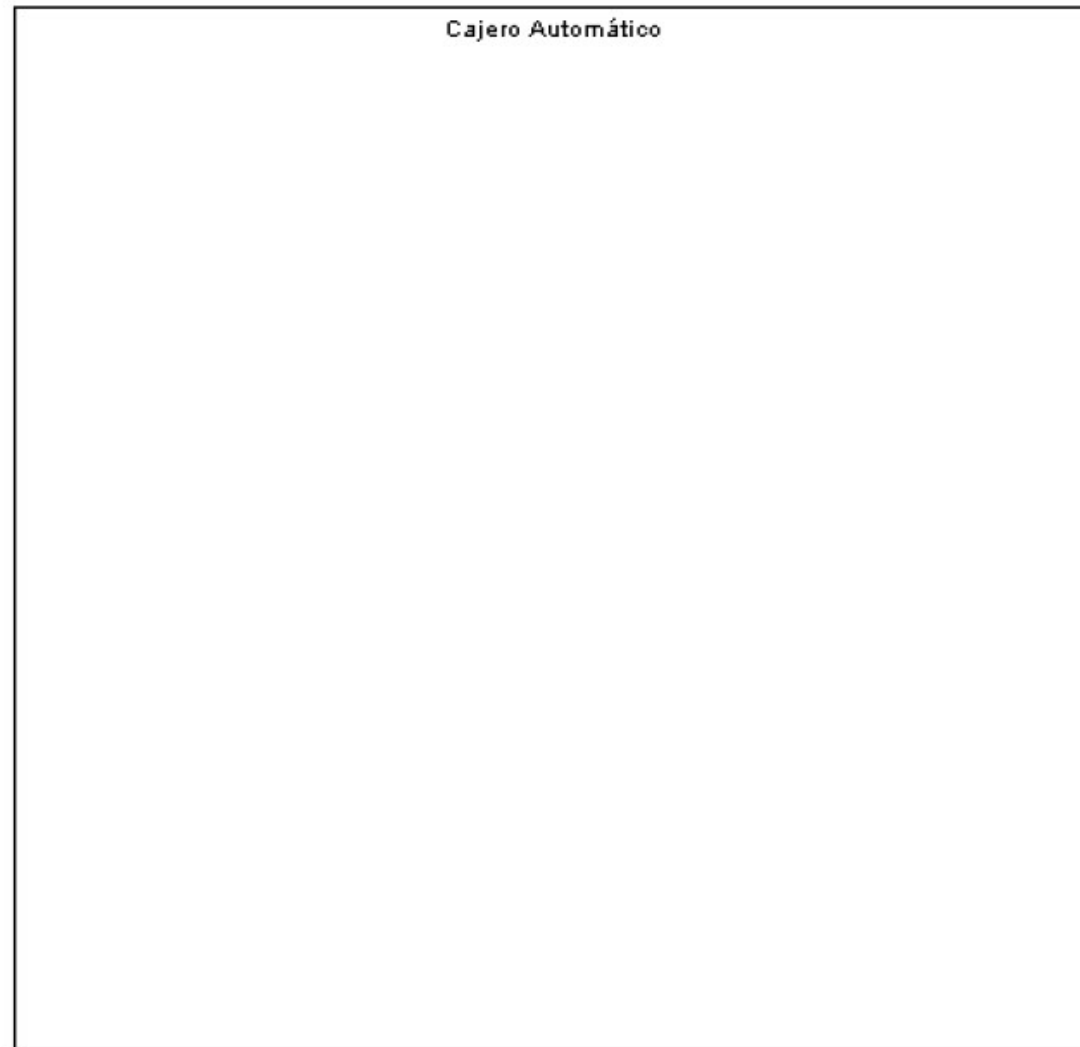
r

- Sistema simplificado de cajero automático
  - ❖ Distribución de dinero a cada poseedor de tarjeta de crédito
  - ❖ Consulta del saldo de cuentas, depósito de efectivo y cheques para los clientes del banco con cartilla
  
- Pasos
  - ❖ Identificar actores
  - ❖ Identificar casos de uso
  - ❖ Identificar relaciones
  - ❖ Construir diagrama

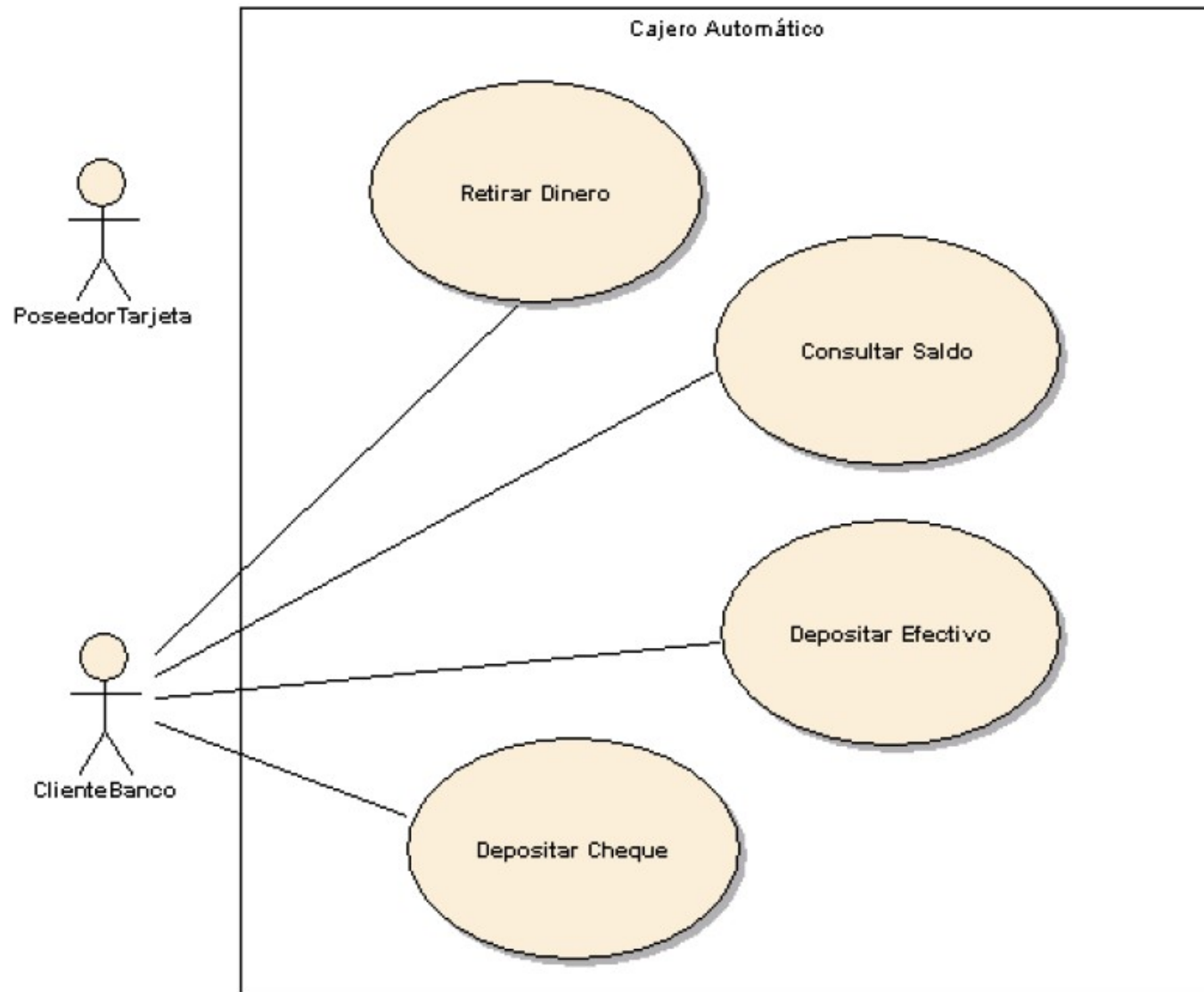
## 3.4 Ejemplo de Diagrama de Casos de Uso: Identificar actores

---

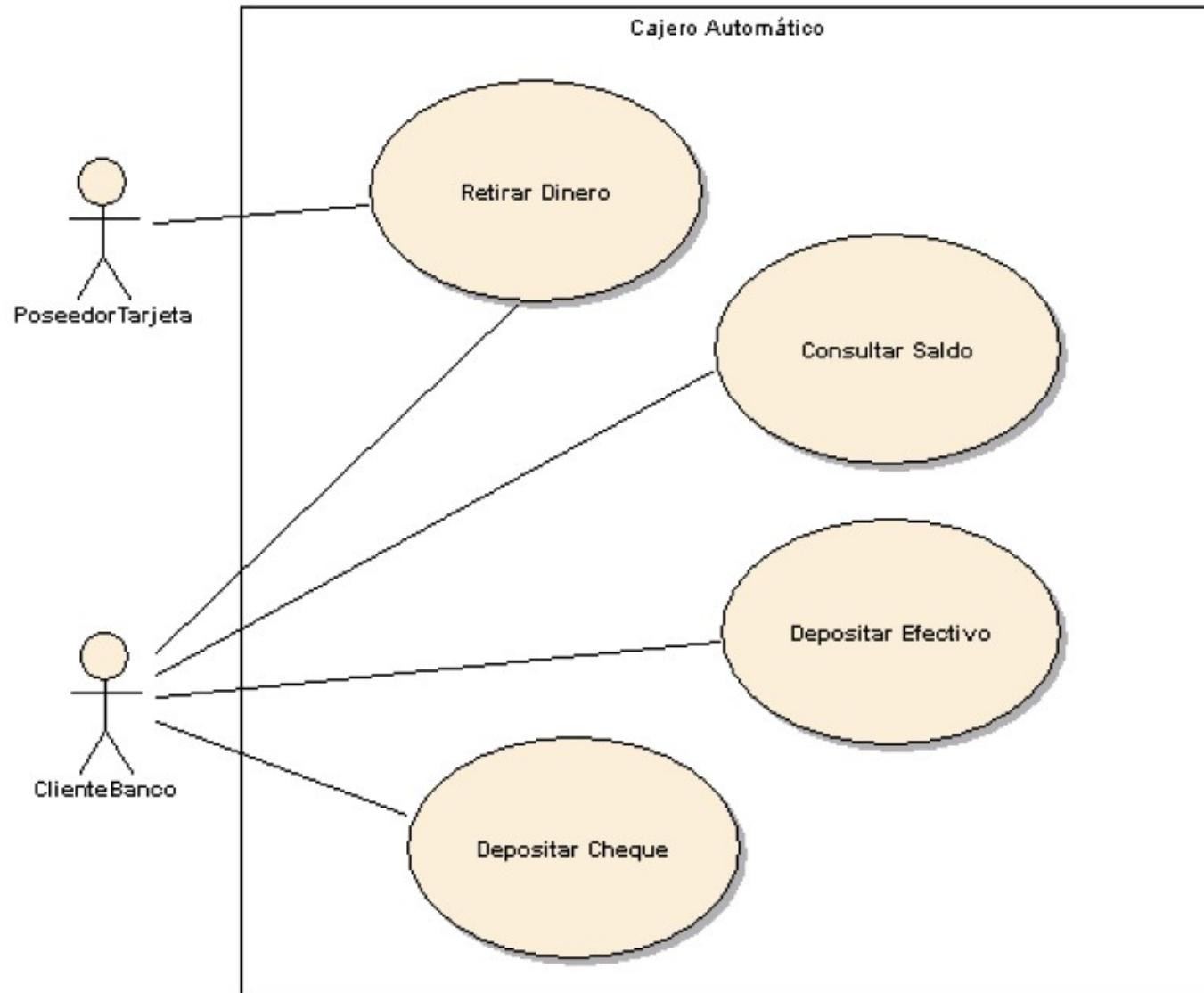
Identificar



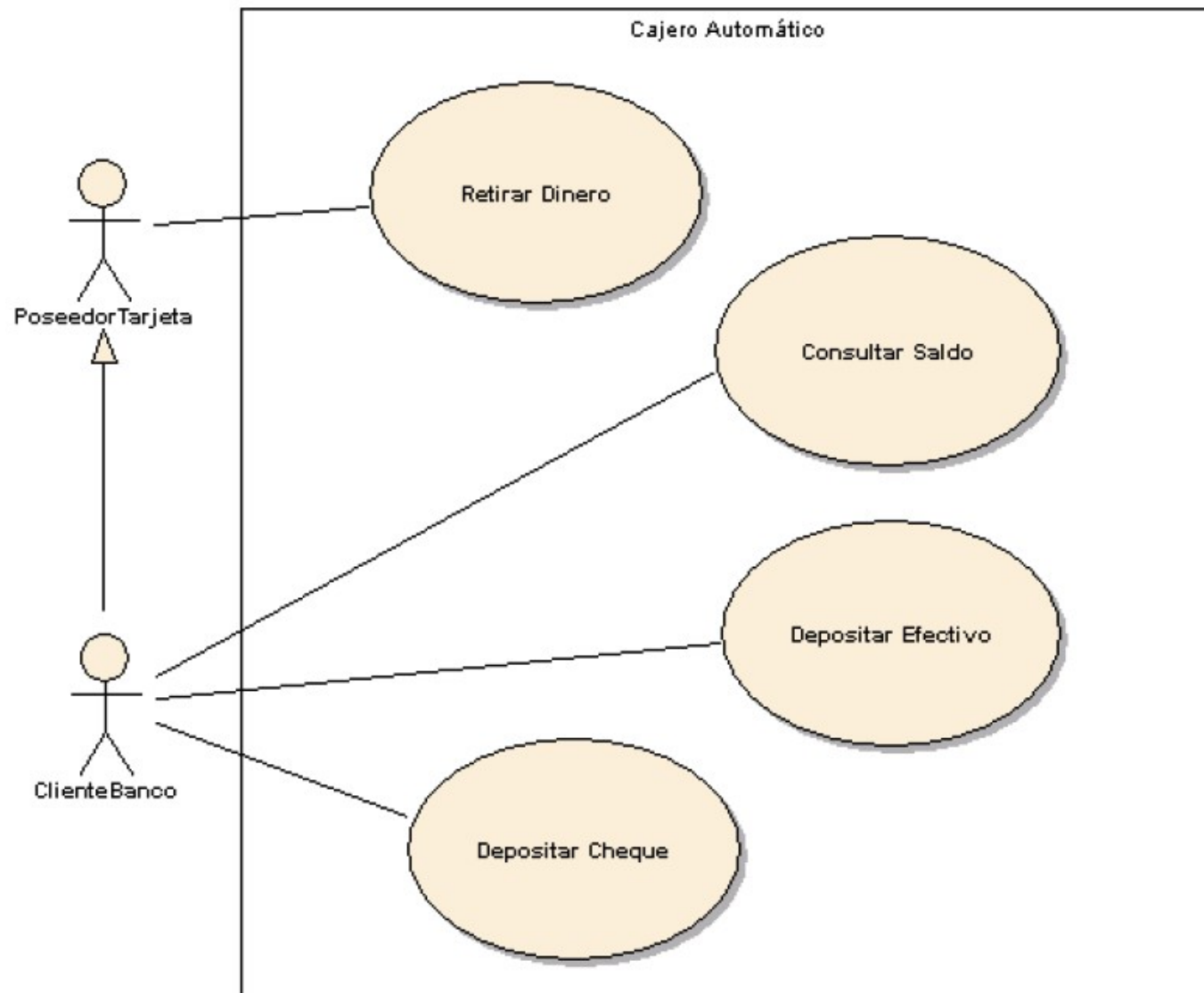
## 3.4 Ejemplo de Diagrama de Casos de Uso: Identificar casos de uso



## 3.4 Ejemplo de Diagrama de Casos de Uso: Identificar casos de uso

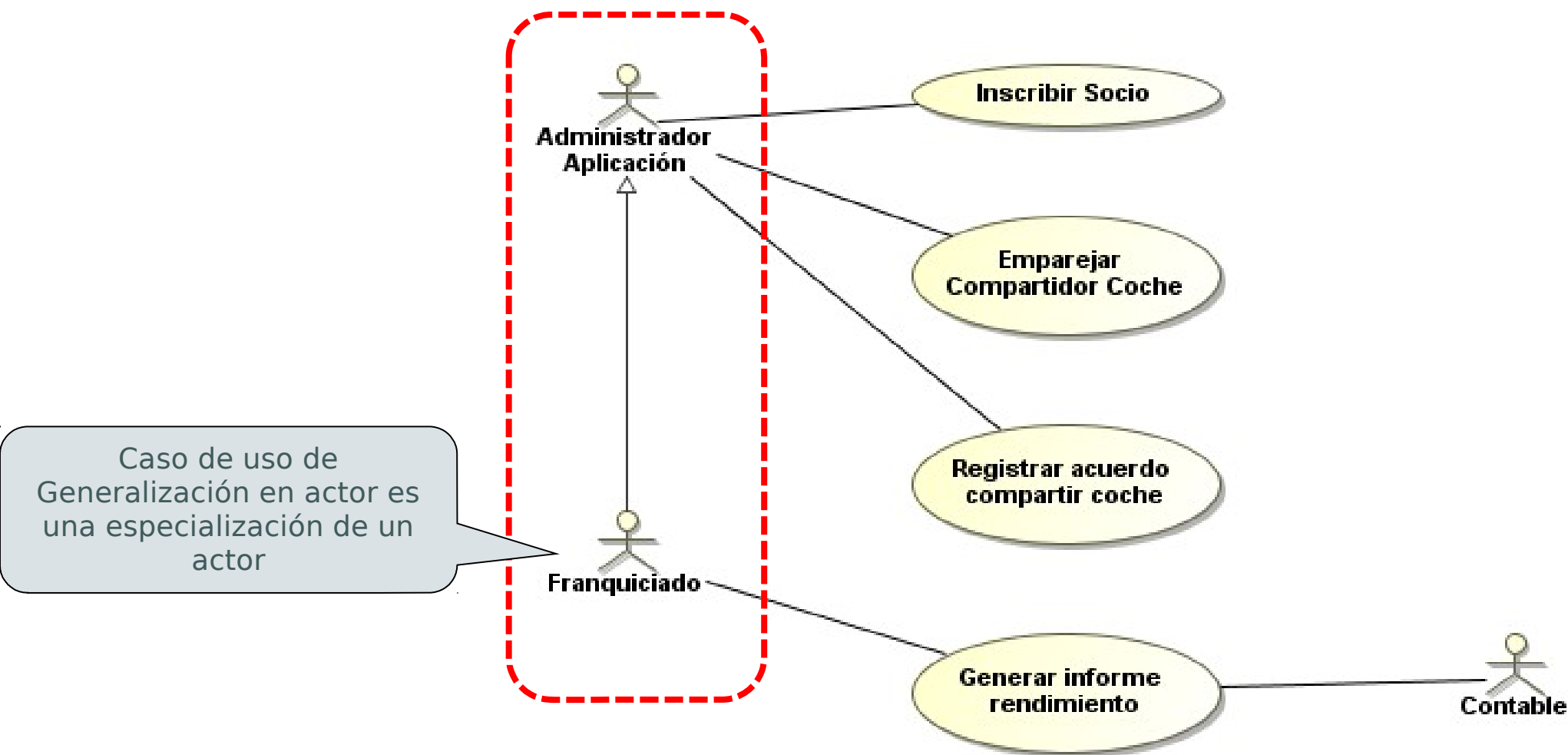


### 3.4 Ejemplo de Diagrama de Casos de Uso: Identificar relaciones



## 3.4 Ejemplo de Diagrama de Casos de Uso: más detalles

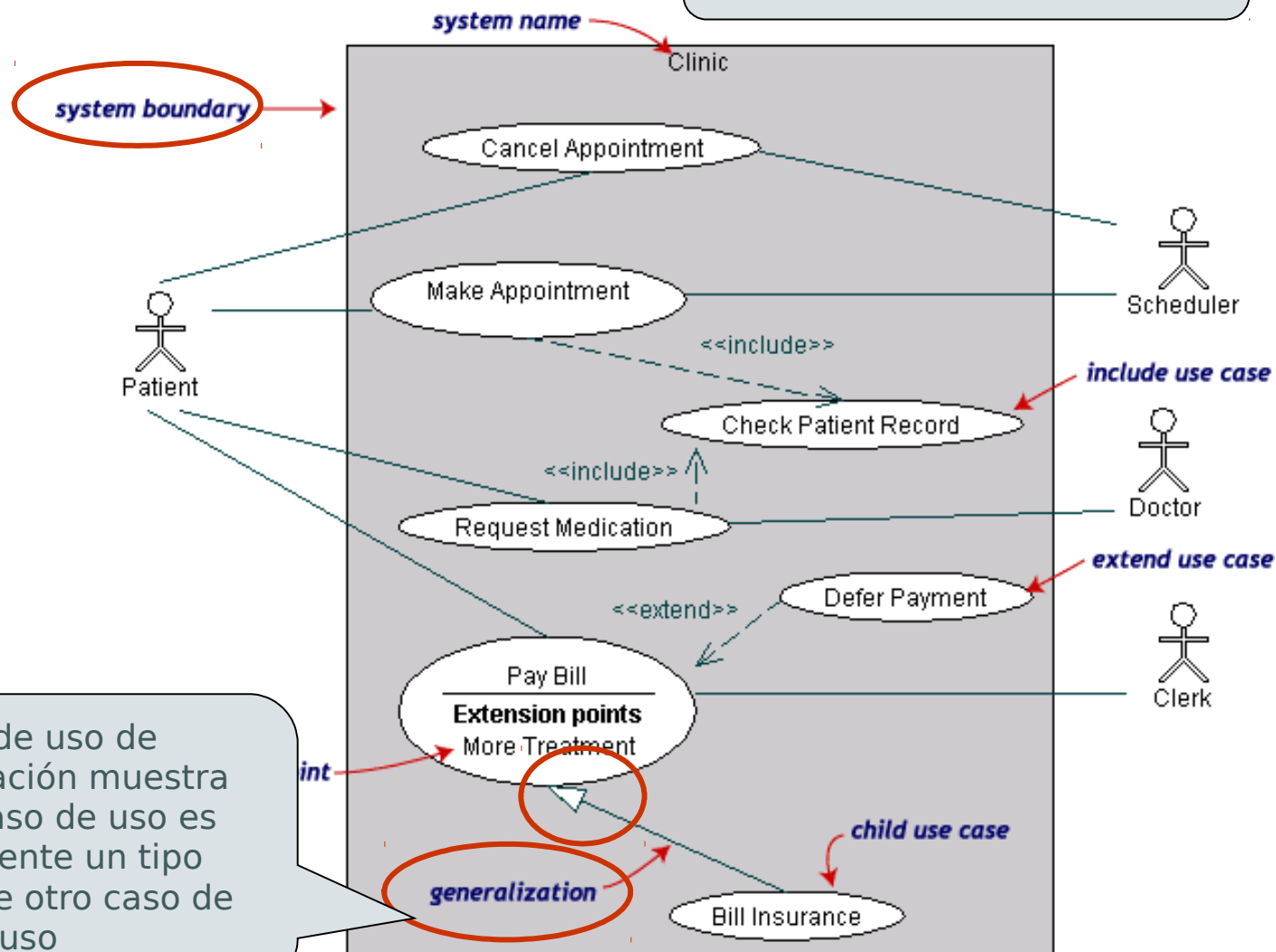
### ➤ Generalización en Actores



## 3.4 Ejemplo de Diagrama de Casos de Uso: más detalles

### ➤ Generalización en Caso de Uso

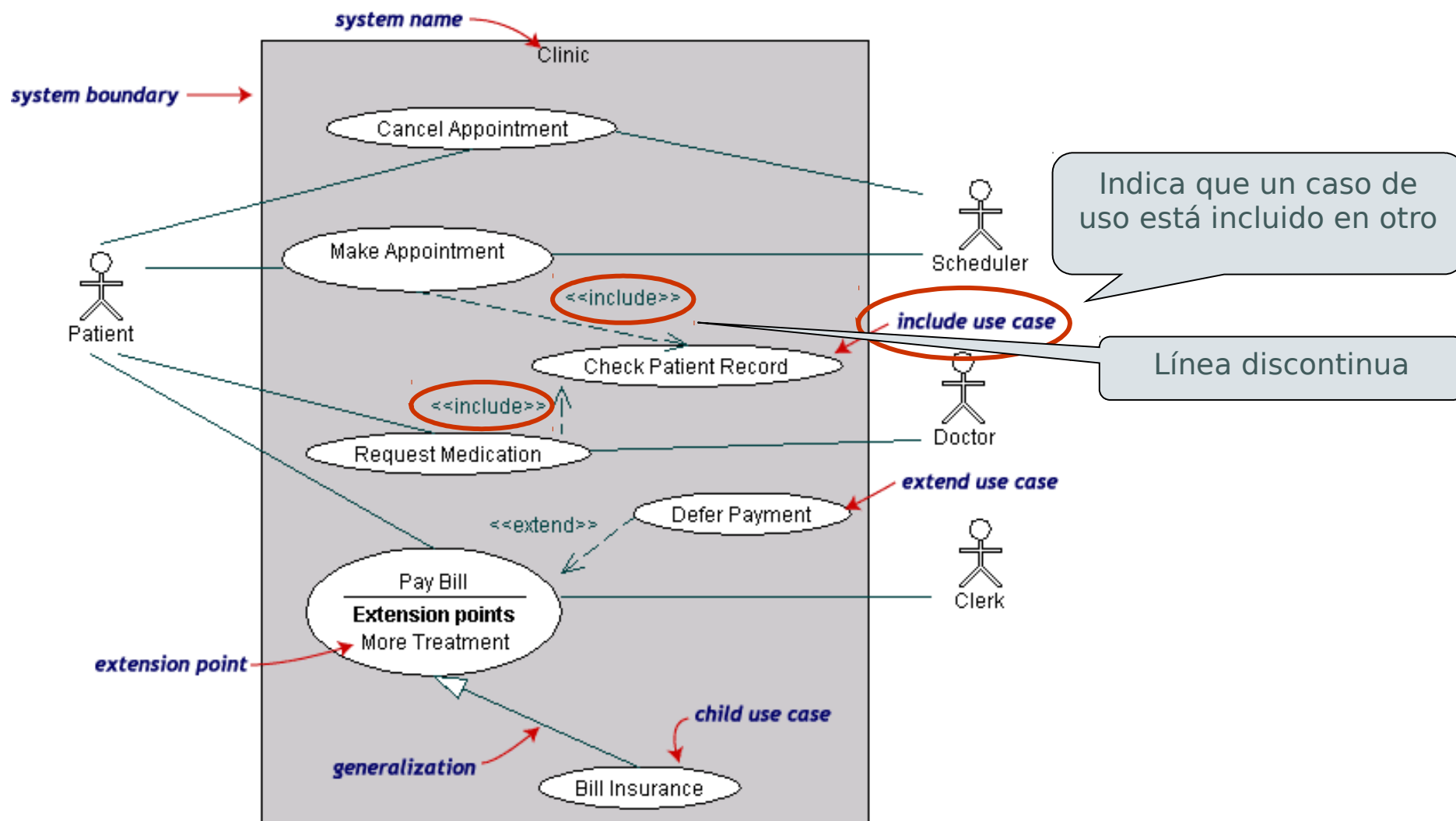
Un rectángulo de Límites del sistema separa el sistema de los actores externos



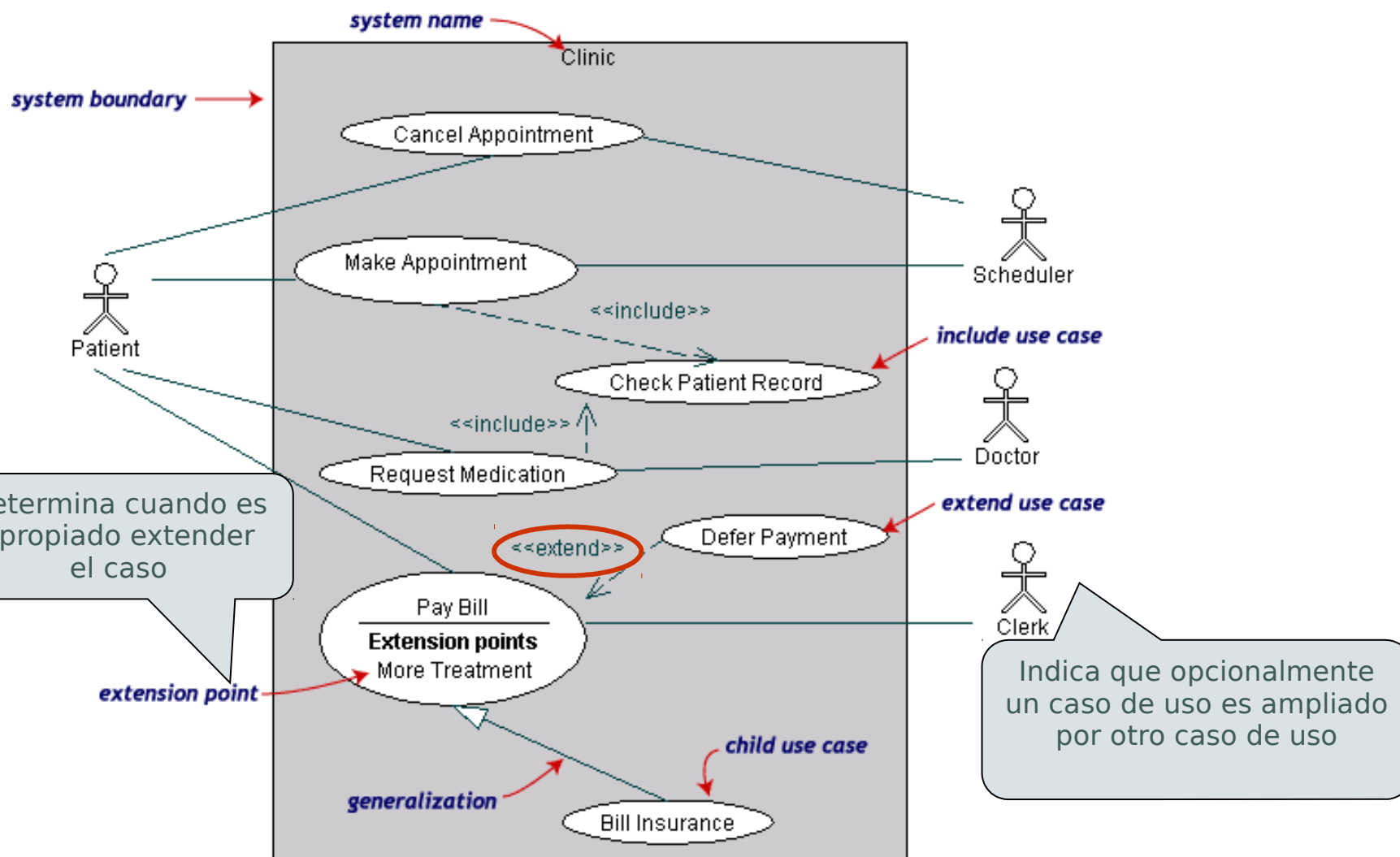
Caso de uso de Generalización muestra que un caso de uso es simplemente un tipo especial de otro caso de uso



### 3.4 Ejemplo de Diagrama de Casos de Uso: más detalles



## 3.4 Ejemplo de Diagrama de Casos de Uso: más detalles



## 3.4 Ejemplo de Diagrama de Casos de Uso: más detalles

### ➤ Include

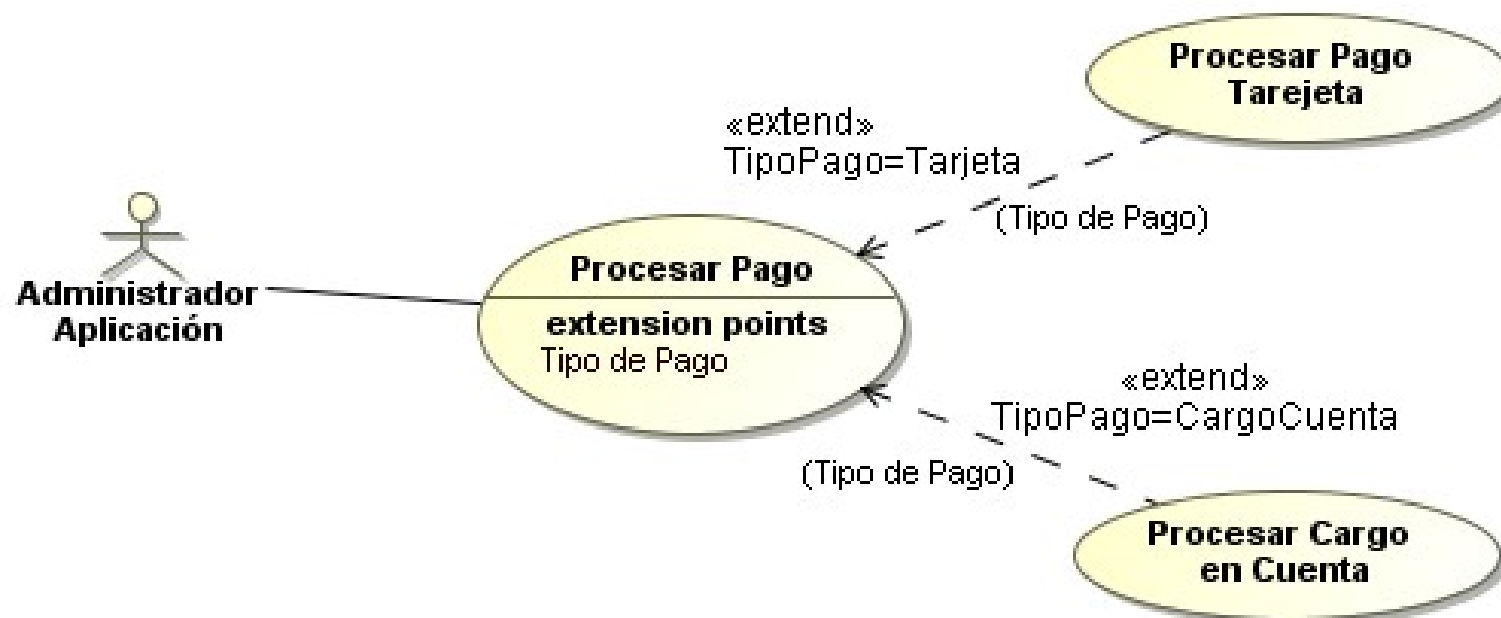
- ❖ Una relación de inclusión (Include) denota que un caso de uso está incluido en otro.
- ❖ Se da cuando un caso de uso se utiliza por sí mismo y, además, otros casos de uso incluyen siempre esa funcionalidad
- ❖ También aparecen cuando dos o más casos de uso comparten una funcionalidad
- ❖ Denota que el caso de uso **SIEMPRE** está incluido en el otro caso de uso

### ➤ Extended

- ❖ Indica que OPCIONALMENTE un caso de uso es ampliado (extendido) por otro caso de uso
- ❖ Se puede indicar mediante un “punto de extensión” (condición) cuando se amplía el caso de uso

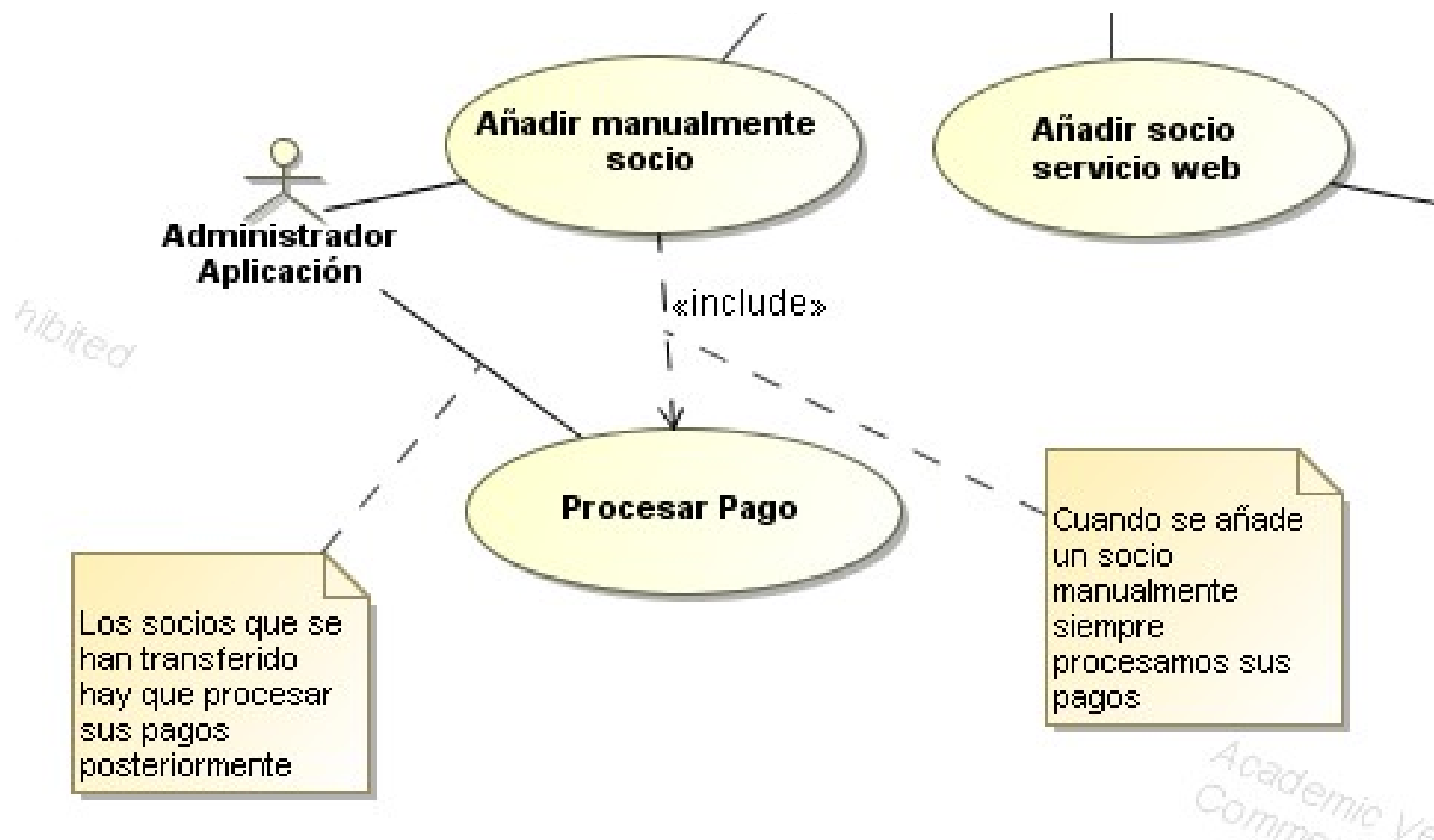
## 3.4 Ejemplo de Diagrama de Casos de Uso: más detalles

### ➤ Relación Extended



## 3.4 Ejemplo de Diagrama de Casos de Uso: más detalles

### ➤ Relación Include



## 3. Planificación. 3. 5. Especificación de un caso de uso

---

### ➤ Especificación de un caso de uso

- ❖ Un caso de uso recoge, en un primer momento, una descripción general
- ❖ Se puede completar la descripción definiendo
  - ✓ Las precondiciones y postcondiciones del caso de uso: qué condiciones deben cumplirse para que se realice un caso de uso y cuáles son aquellas condiciones que se cumplen posteriormente al caso de uso
  - ✓ El flujo de sucesos: descripción textual de la secuencia de acciones del caso de uso
  - ✓ Excepciones: casos concretos en los que el caso de uso fallaría

### 3. Planificación

#### ➤ Modelo de caso de uso

RF-<id>	<nombre descriptivo>	
Versión	<nº de la versión actual> (<fecha de la versión actual>)	
Autores	• <autor de la versión actual> (<organización del autor>) ...	
Fuentes	• <fuente de la versión actual> (<organización de la fuente>) ...	
Objetivos asociados	• OBJ-x <nombre del objetivo> ...	
Requisitos asociados	• Rx-y <nombre del requisito> ...	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso { durante la realización de los casos de uso <lista de casos de uso>, cuando <evento de activación> }	
Precondición	<precondición del caso de uso>	
Secuencia normal	Paso	Acción
	p <sub>1</sub>	{El actor <actor>, El sistema} <acción/es realizada/s por actor/sistema>
	p <sub>2</sub>	Se realiza el caso de uso <caso de uso (RF-x)>
	p <sub>3</sub>	Si <condición>, {el actor <actor>, el sistema} <acción/es realizada/s por actor/sistema>
	p <sub>4</sub>	Si <condición>, se realiza el caso de uso <caso de uso (RF-x)>
	...	...
Postcondición	<postcondición del caso de uso>	
Excepciones	Paso	Acción
	p <sub>i</sub>	Si <condición de excepción>, {el actor <actor>, el sistema} <acción/es realizada/s por actor/sistema>, a continuación este caso de uso {continúa, termina}

### 3. Planificación. 3. 5. Especificación de un caso de uso

#### ➤ Tabla de ejemplo

Caso de uso	R1. Reservar un libro del fondo bibliográfico
Objetivo:	Reservar un libro del fondo bibliográfico
Actores:	Alumno
Precondiciones:	El Libro debe existir El libro no debe ser para préstamo en sala Debe existir suficiente stock El alumno no puede estar amonestado
Postcondiciones:	---
Secuencia normal	1. El alumno llega con un libro 2. Se identifica al alumno en el sistema 3. Se registra el préstamo
Excepciones	---
Observaciones	El ordenador no tendrá una pantalla superior a 14". Resolución máxima: 800x600



### 3. Planificación

---

## ¿Cómo construir un diagrama de Caso de Uso?

Tres etapas:

- Identificar los actores
  - Delimitar quien/quienes van a usar el sistema
- Buscar los casos de uso
  - Se sugiere un CU por cada rol de cada actor (por cada actor, intentar describir su roles)
  - Elegir nombres descriptivos para los CU
- Describir los casos de uso
  - Por cada caso de uso o aquellos complejos se tiene que describir su funcionalidad

### 3. Planificación. Ejemplo de caso de Uso

---

Modelar un sistema informático para la biblioteca de la Universidad.

- Formado por usuarios (alumnos) que pueden realizar búsquedas y reservas
- Por el profesorado que además pueden reservar tantos libros como revistas
- Existe un proceso de actualización de los fondos bibliográficos.
  - Añadir libros
  - Dar de baja libros
  - Etc.

### 3. Planificación. Ejemplo de caso de Uso

---

#### Ejemplo de la Biblioteca: Pasos 1/6

- **Objetivos:** Modelar un Sist. Infor. para una biblioteca
- **Requisitos**
  - ***Libros y Revistas***
    - La biblioteca contiene libros y revistas, es posible que de cada libro haya varias copias.
    - Algunos de los libros se prestan por pequeños periodos de tiempo, el resto pueden ser prestados por periodos de hasta tres semanas
    - Solo el personal pueden tomar prestadas las revistas.
    - Es posible tomar prestados hasta seis elementos a la vez, el personal hasta doce.
    - Regularmente llegan nuevos ejemplares de libros y revistas, algunas veces se eliminan los antiguos.
    - Al final del año se envían las revistas para ser encuadernadas.

### 3. Planificación. Ejemplo de caso de Uso

---

#### Ejemplo de la Biblioteca: Pasos 2/6

##### – Requisitos

- ***Prestamos:***

- El sistema debe mantener información sobre los libros prestados y devueltos. El sistema actual ya lo hace.
- Deben emitirse recordatorios cuando un libro se excede del tiempo de préstamo.
- Es posible que en un futuro se permita a los usuarios extender el periodo de préstamo si un libro no esta reservado.

- ***Búsquedas:***

- El sistema debe permitir a los usuarios la búsqueda de libros sobre un tópico, un autor, etc...
- El sistema debe permitir comprobar si un libro está disponible.
- El sistema debe permitir la reserva de libros.
- Cualquiera puede realizar búsquedas en la biblioteca.

### 3. Planificación. Ejemplo de caso de Uso

#### Ejemplo de la Biblioteca: Pasos 3/6

- **Actores**

- **Bibliotecario, Personal (alumno, profesor)**

- **Requisitos**

Requisitos	
<b>Libros</b>	
R1. Reservas	Reserva un fondo
R2. Préstamo	Presta un fondo
R3. Devolución	Devolución de un fondo
R4. Ampliación	Amplia el plazo de reserva
<b>Revista</b>	
R5. Préstamo	Préstamo de una revista
R6. Devolución	Devolución de una revista
<b>Buscador</b>	
R7. Buscar	Búsqueda
<b>Fondo</b>	
R8. Añadir Título	....
R9. Borrar Título	....
R10. Añadir Ejemplar	....
R11. Borrar Ejemplar.	....

### 3. Planificación. Ejemplo de caso de Uso

#### Ejemplo de la Biblioteca: Pasos 4/6

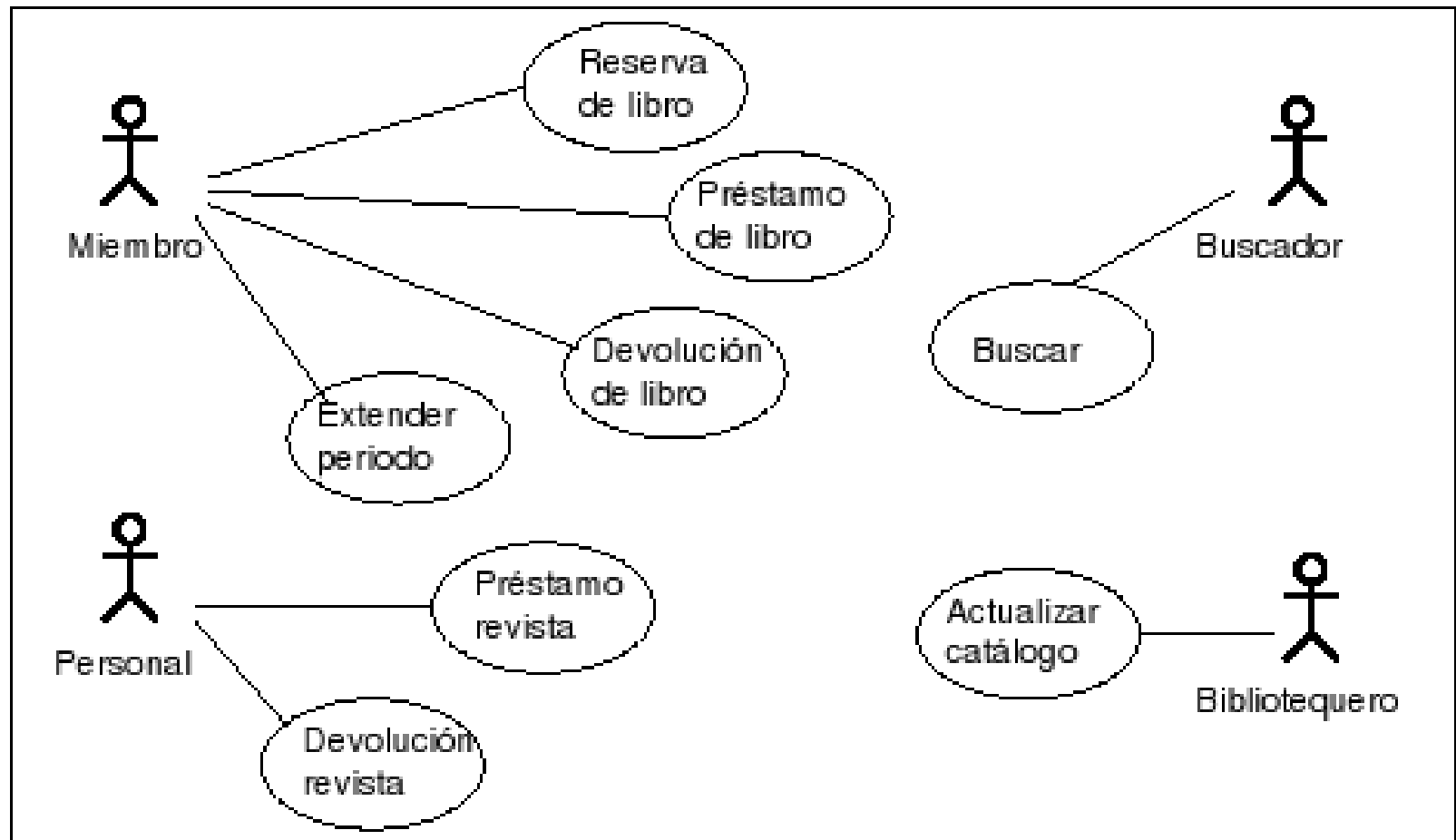
##### – Especificación de requisitos

- **Por cada caso de uso, debe formalizarse su especificación.**

Caso de Uso	R1. Reserva un libro del fondo bibliográfico
Objetivo:	Reserva un libro del fondo bibliográfico
Actores:	Alumno
Precondiciones	El Libro debe existir Debe existir suficiente stock El alumno no puede estar amonestado
Postcondiciones	-
Cuestiones	-

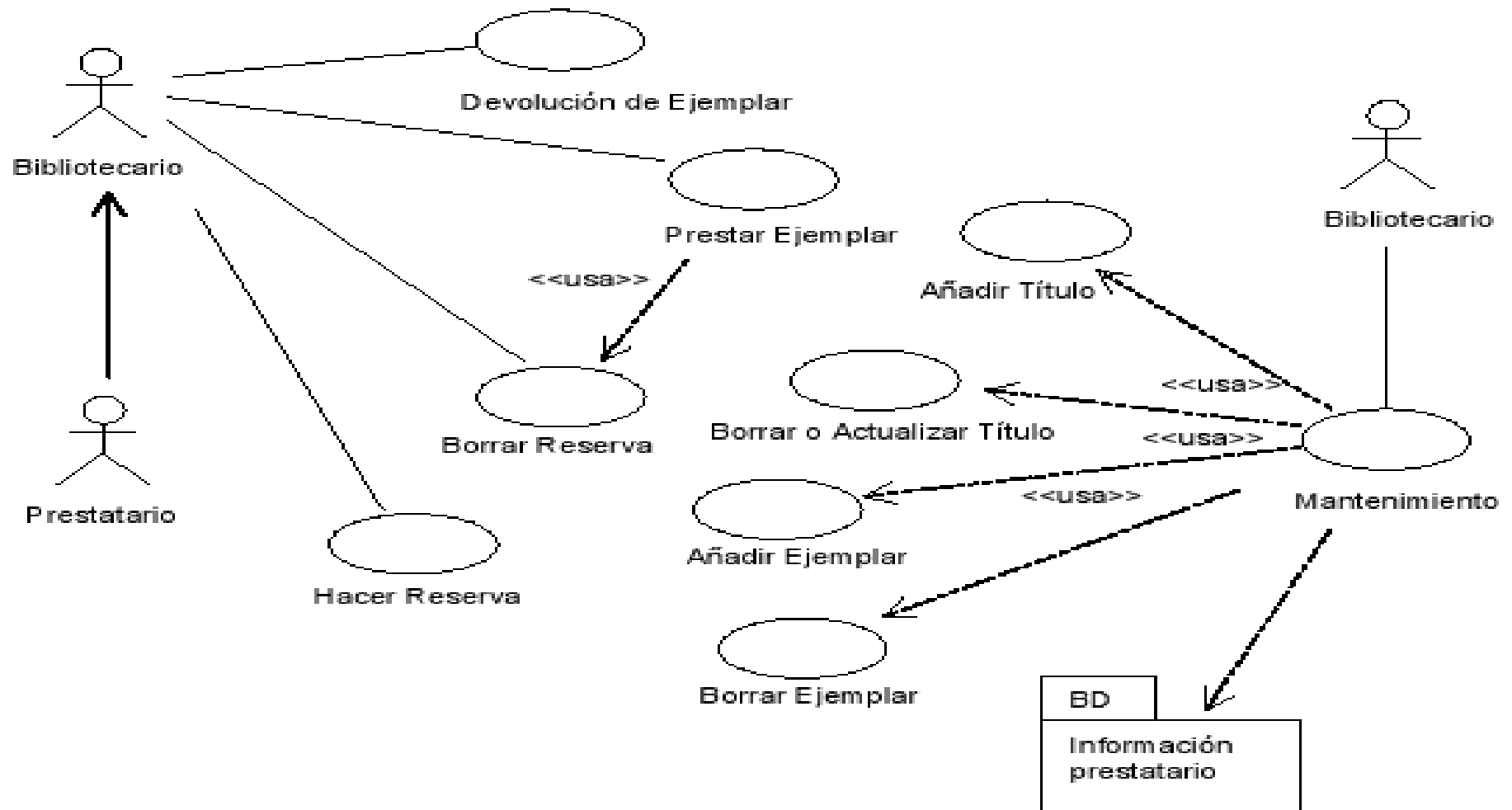
### 3. Planificación. Ejemplo de caso de Uso

#### Ejemplo de la Biblioteca: Pasos 5/6



### 3. Planificación. Ejemplo de caso de Uso

#### Ejemplo de la Biblioteca: Pasos 5b/6





### 3. Planificación. Ejemplo de caso de Uso

---

#### Ejemplo de la Biblioteca: Pasos 6/6

##### – Escenarios

- **R1-Reserva de Libro**

- R1.1 El portal de reserva pide al alumno que se autentifique

- R1.1.1 Mediante Usuario

- R1.1.1.2 Mediante contraseña

- R1.2 El cliente web envía el usuario y contraseña al servidor de contraseña (LDAP)

- R1.3 El sistema notifica la aceptación

- R1.4 El portal de reserva pide al alumno el identificador del libro

- R1.5 El sistema comprueba si el libro existe

- R1.6 Si el libro esta disponible, se reserva

### 3. Planificación. Ejemplo de caso de Uso

---

#### Ejemplo de la Biblioteca: Pasos 6b/6

- **Escenarios alternativos**

- **R1-Reserva de Libro**

R.1.3\_a: El usuario no se autentifica, ERROR

R.1.5\_a: El libro no existe en el fondo bibliográfico

R.1.6\_a: El libro no esta disponible, se queda pendiente

### 3. Planificación. Ejemplo de caso de Uso

---

Más ejemplos en documentos adicionales