

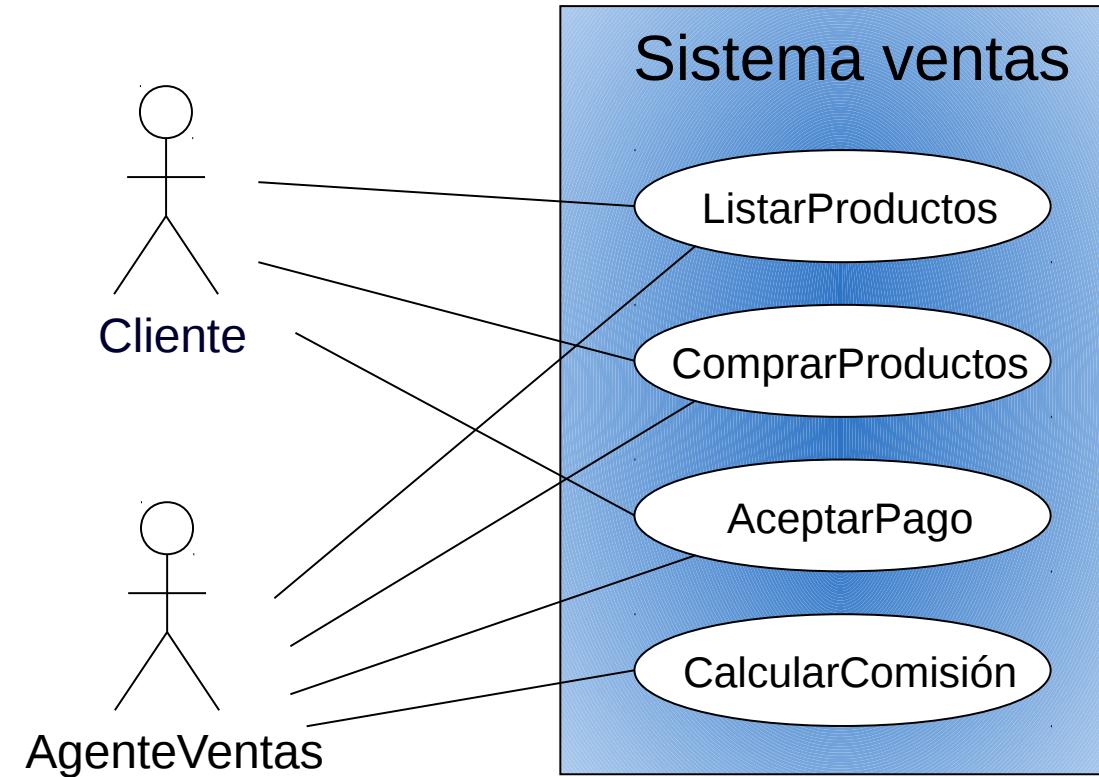
# Modelado Avanzado de Casos de Usos

# Aspectos avanzados

- Generalización actores
  - ◆ Relación de generalización entre un actor más general y un actor más específico
- Generalización casos de uso
  - ◆ Relación de generalización entre un caso de uso más general y otro más específico
- <<include>>
  - ◆ Relación entre casos de uso que permite que un caso de uso incluya comportamiento de otro
- <<extend>>
  - ◆ Relación entre casos de uso que permite que un caso de uso extienda su comportamiento con uno o más fragmentos de comportamiento de otro

# Generalización actores

- ¿Por qué?



# Generalización actores

## ■ ¿Por qué?

- ◆ Similitudes entre el cliente y el AgenteVentas
  - Ambos disparan los casos de uso ListarProductos, ComprarProductos y AceptarPago
  - CalcularComision sólo es disparado por AgenteVentas
- ◆ La similitud no está bien expresada
  - Es un poco lío con tantas líneas cruzadas

## ■ Solución

- ◆ Usar la generalización para expresar la similitud y simplificar el diagrama

# Generalización actores

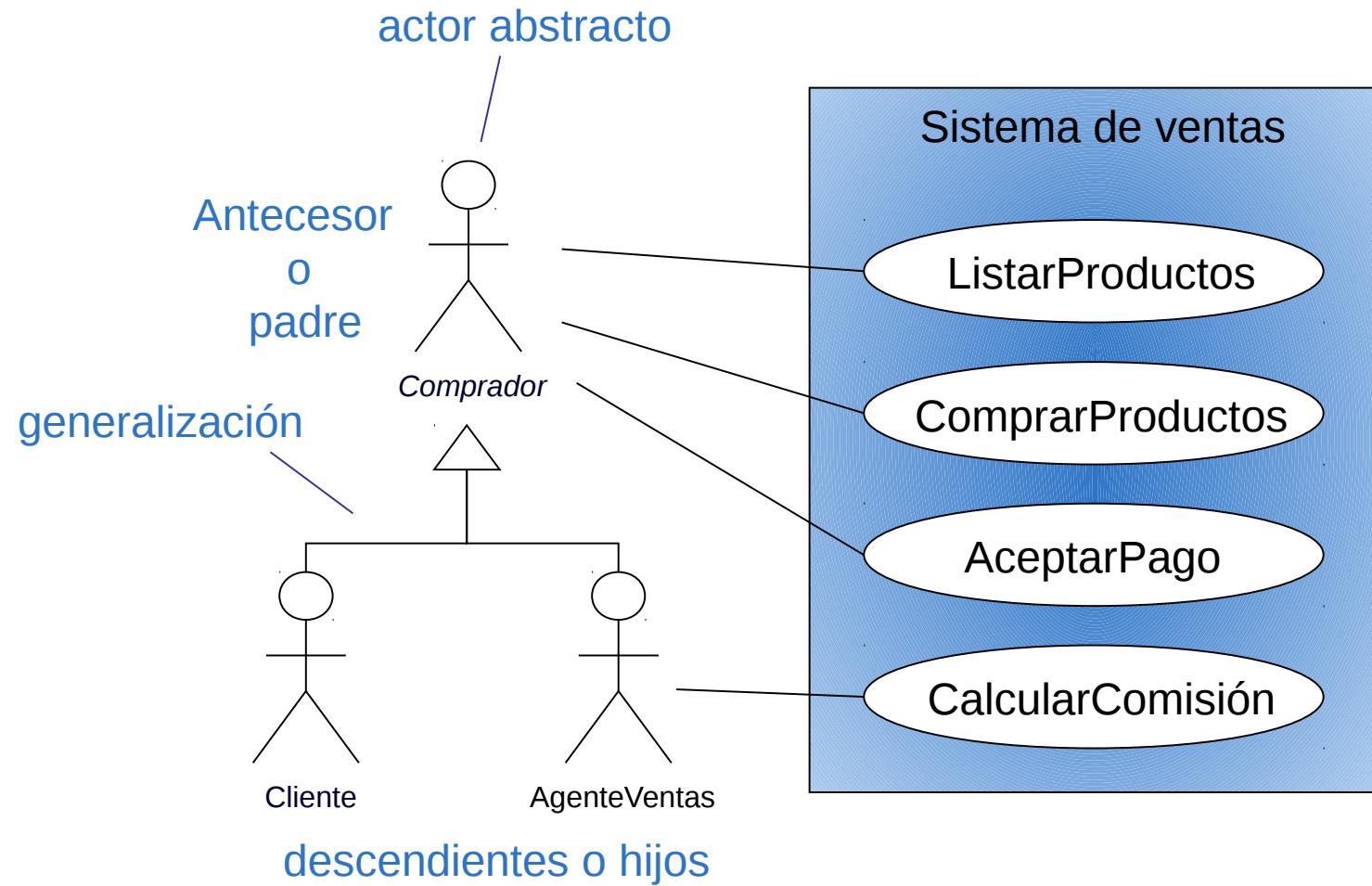
## ■ ¿Cómo?

- ◆ Introducir un actor abstracto
  - Captura el comportamiento común entre actores que se comunican con el mismo conjunto de casos de uso de la misma manera
  - Comprador
- ◆ Los descendientes heredan todos los roles y relaciones de los casos de uso de su padre abstracto

## ■ Ventajas

- ◆ Simplifica el diagrama de casos de uso
- ◆ Simplifica la semántica del modelo

# Generalización actores



# Generalización actores

## ■ Principio de sustitución

- ◆ Podemos usar un descendiente en cualquier lugar donde se pudiera usar un antecesor
  - En cualquiera de los casos de uso ListarProductos, ComprarProductos y Aceptarpago
  - Usar Cliente o AgenteVentas en vez de Comprador

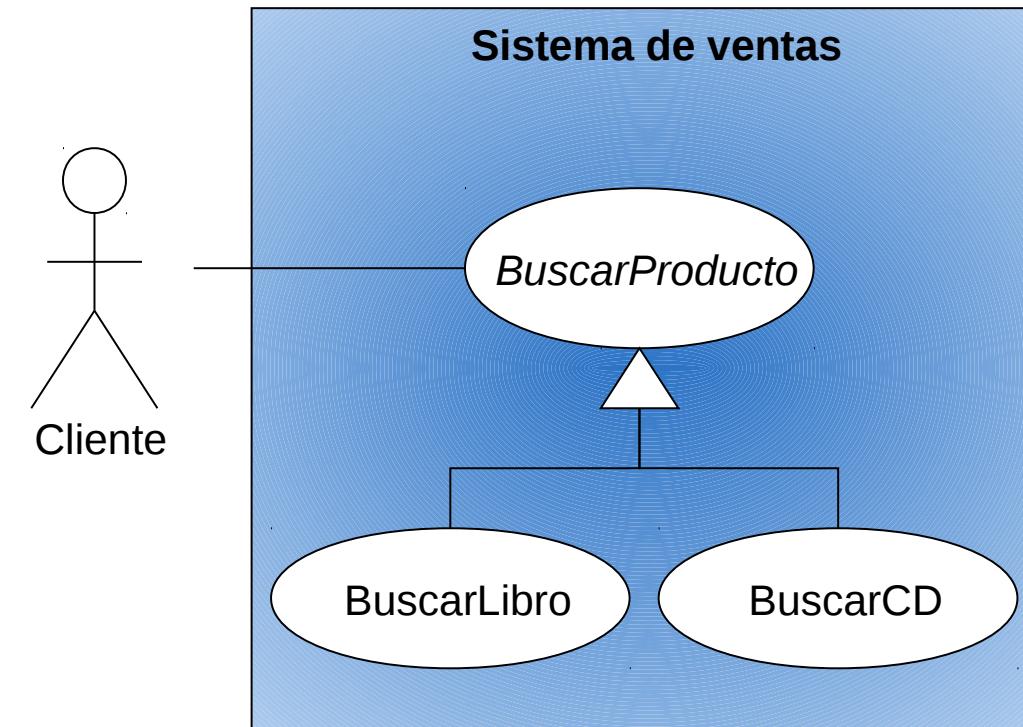
## ■ Actor abstracto

- ◆ Representa un papel abstracto dentro del dominio
- ◆ No puede ser instanciado por una persona o cosa

# Generalización de caso de uso

## ■ ¿Cuándo?

- ◆ Tenemos varios casos de uso que son una especialización (hijos) de un caso más general (padre)



# Generalización de caso de uso

## ■ Semántica

- ◆ Los hijos más específicos que el padre. Pueden:
  - Heredar características
  - Añadir nuevas características
  - Sobrescribir (cambiar) características heredadas

Característica	Heredar	Añadir	Cambiar
Relación	Si	Si	No
Extension point	Si	Si	No
Precondiciones	Si	Si	Si
Postcondiciones	Si	Si	Si
Paso flujo principal	Si	Si	Si
Flujo alternativo	Si	Si	Si

# Generalización de caso de uso

## ■ Documentación

Caso de uso: <i>BuscarProducto</i>
ID:6
<b>Descripción:</b> El cliente busca un producto
<b>Actores principales:</b> Cliente
<b>Actores secundarios:</b> Ninguno
<b>Precondiciones:</b> Ninguna
<b>Flujo principal:</b> 1. El Cliente selecciona “buscar producto” 2. El sistema le pide al Cliente los criterios de búsqueda 3. El Cliente introduce los criterios pedidos 4. El sistema busca los productos que coinciden con los criterios del Cliente 5. Si el sistema encuentra productos que coinciden 5.1 El sistema muestra la lista de productos 6. Sino 6.1 El sistema le indica al cliente que no se han encontrado productos
<b>Postcondiciones:</b> Ninguna
<b>Flujo alternativo:</b> Ninguno

# Generalización de caso de uso

## ■ Documentación

- ◆ Cada paso en el hijo está seguido por el paso equivalente en el padre, si existe
  - 3. (3.) El cliente introduce los criterios de búsqueda
  - 6.2 (6.1) El sistema indica al Cliente que no se han encontrado productos coincidentes
  - 6.3 El sistema vuelve a mostrar la página de búsqueda de libros
- ◆ Si el paso en el hijo cambia el paso del padre, se le antepone “o” número del paso del padre
  - 1. (o1.) El Cliente secciona “buscar libro”
  - 5.2 (o5.1) El sistema muestra una página mostrando los detalles de cinco libros

# Generalización de caso de uso

cambiado  
cambiado  
heredado sin cambio  
cambiado  
cambiado  
añadido  
cambiado y reenumerado  
añadido  
añadido  
heredado sin cambio  
añadido  
reenumerado

Caso de uso: <i>BuscarLibro</i>	
ID:7	
ID del padre: 6	
Descripción: El cliente busca un libro	
Actores principales: Cliente	
Actores secundarios: Ninguno	
Precondiciones: Ninguna	
Flujo principal:	<ol style="list-style-type: none"> <li>1. (o1.) El Cliente selecciona “buscar libro”</li> <li>2. (o2.) El sistema le pide al Cliente los criterios de búsqueda de un libro: autor, título</li> <li>3. (3.) El Cliente introduce los criterios pedidos</li> <li>4. (o4.) El sistema busca los libros que coinciden con los criterios del Cliente</li> <li>5. (o5.) Si el sistema encuentra libros que coinciden           <ol style="list-style-type: none"> <li>5.1 El sistema muestra el más vendido</li> <li>5.2 (o5.1) El sistema muestra los detalles de 5 libros</li> <li>5.3 Para cada libro en la página el sistema muestra el título, autor y precio</li> <li>5.4 Mientras haya más libros, el sistema da la opción de mostrar la página siguiente</li> </ol> </li> <li>6. (6.) Sino           <ol style="list-style-type: none"> <li>6.1 El sistema muestra el libro más vendido</li> <li>6.2 (6.1) El sistema le indica al cliente que no se han encontrado productos</li> </ol> </li> </ol>
Postcondiciones: Ninguna	
Flujo alternativo: Ninguno	

# Generalización de caso de uso

## ■ ¡¡¡CUIDADO!!!!

- ◆ Puede llegar a ser inmanejable
- ◆ Tarea laboriosa y propensa a errores
  - Mantener manualmente la consistencia entre los padres y los hijos si se producen cambios

## ■ Solución

- ◆ Casos de uso sin flujo principal, sólo descripción de la semántica.
- ◆ No hay que preocuparse de la herencia o de los cambios



# Generalización de caso de uso

## ■ Caso de uso abstracto

- ◆ Caso de uso padre que no tiene asociado un flujo de eventos o éste está incompleto
  - No puede ser ejecutados por el sistema
- ◆ Permite capturar comportamiento a un alto nivel de abstracción
- ◆ Tiene asociada una descripción del comportamiento que se espera implementen sus hijos.

# Generalización de caso de uso

## ■ Documentación

Caso de uso: <i>BuscarProducto</i>
ID:6
<b>Descripción:</b> Los clientes seleccionaran la opción que quieran buscar e introducirán los criterios de búsqueda que le requiera el sistema. El sistema realizará la búsqueda y mostrará por pantalla los productos coincidentes con los criterios introducidos o un mensaje de error.
<b>Actores principales:</b> Cliente
<b>Actores secundarios:</b> Ninguno
<b>Precondiciones:</b> Ninguna
<b>Flujo principal:</b> Ninguno
<b>Postcondiciones:</b> Ninguna
<b>Flujo alternativo:</b> Ninguno

# Generalización de caso de uso

## Caso de uso: *BuscarLibro*

ID:7

ID del parente: 6

Descripción:  
El cliente busca un libro

Actores principales:  
Cliente

Actores secundarios:  
Ninguno

Precondiciones:  
Ninguna

Flujo principal:

1. El Cliente selecciona “buscar libro”
2. El sistema le pide al Cliente los criterios de búsqueda de un libro: autor, título
3. El Cliente introduce los criterios pedidos
4. El sistema busca los libros que coinciden con los criterios del Cliente
5. Si el sistema encuentra libros que coinciden
  - 5.1 El sistema muestra el más vendido
  - 5.2 El sistema muestra los detalles de 5 libros
  - 5.3 Para cada libro en la página el sistema muestra el título, autor y precio
  - 5.4 Mientras haya más libros, el sistema da la opción de mostrar la página siguiente
6. Sino
  - 6.1 El sistema muestra el libro más vendido
  - 6.2 El sistema le indica al cliente que no se han encontrado productos

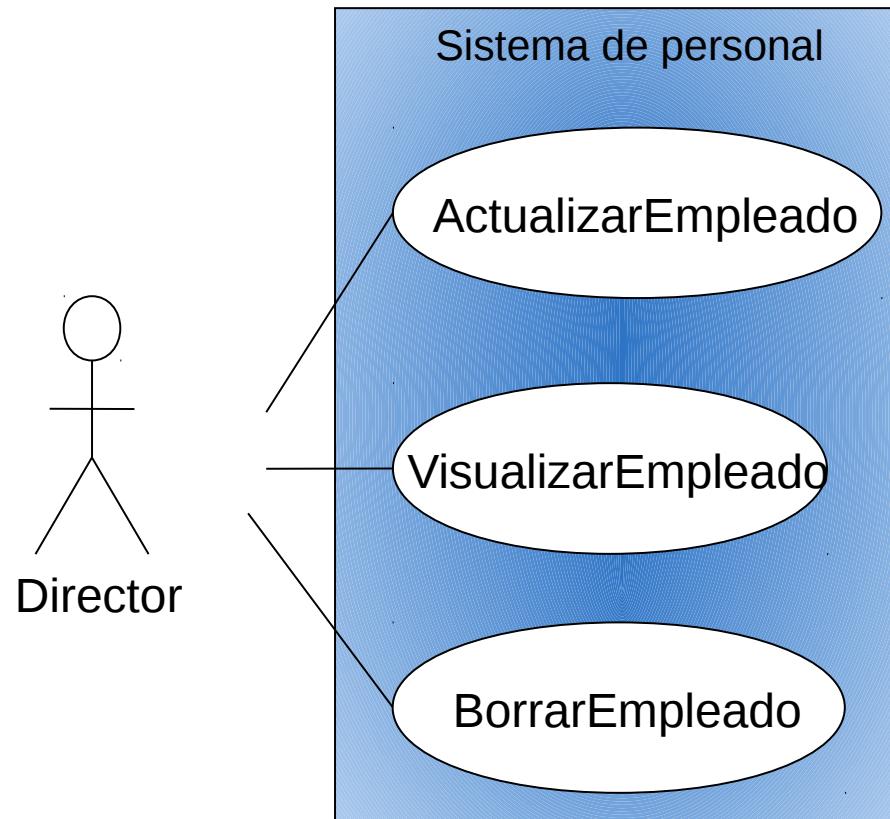
Postcondiciones:  
Ninguna

Flujo alternativo:  
Ninguno

# Relación <<include>>

## ■ Problema

- ◆ Escribir casos de uso puede ser muy repetitivo



**Todos los casos de uso necesitan buscar al empleado  
Pasos comunes en el flujo de eventos principal**

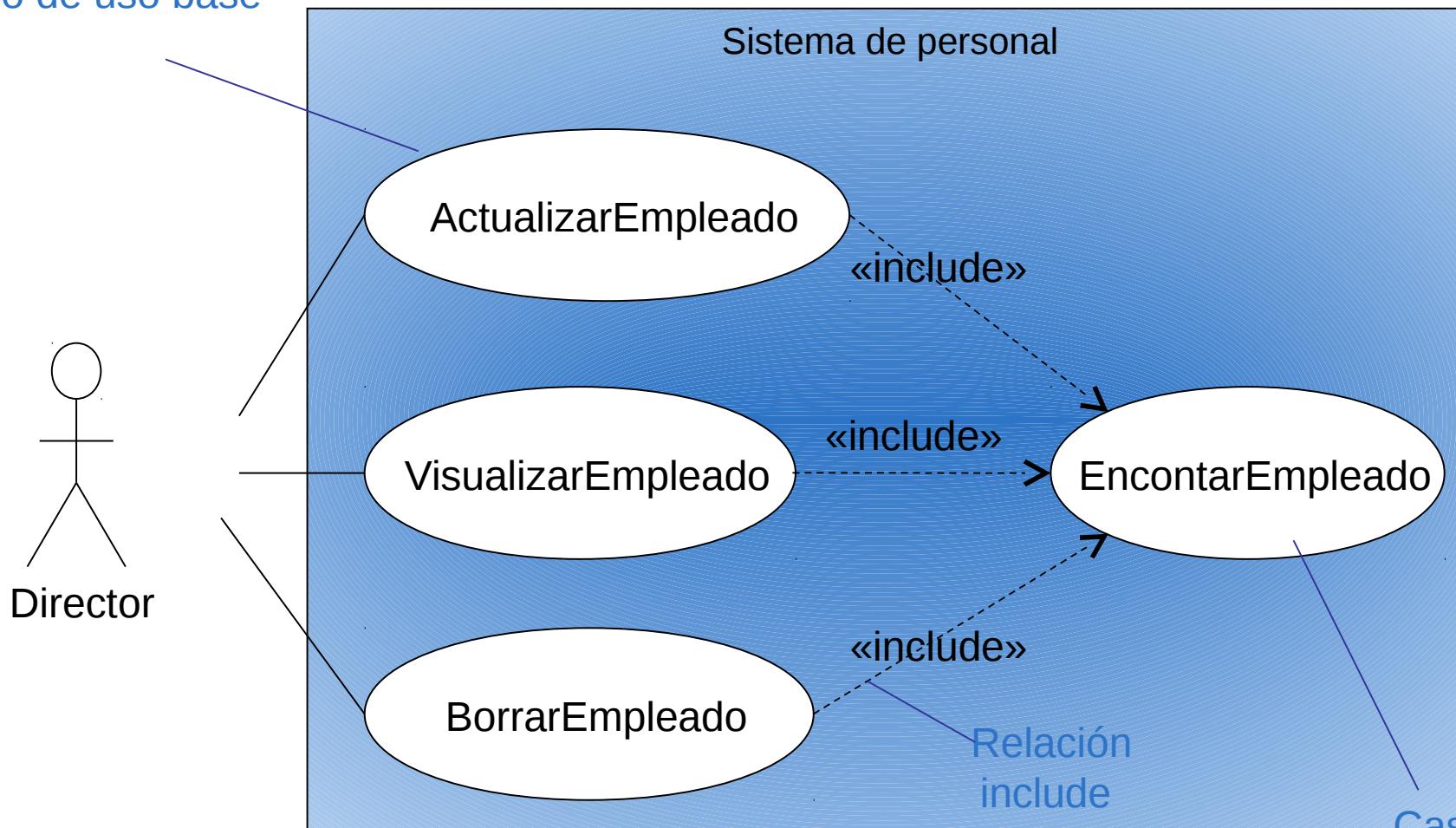
# Relación <<include>>

## ■ Solución

- ◆ Extraer el comportamiento repetido a un caso de uso separado
  - EncontrarEmpleado
- ◆ Relación <<include>>
  - Relación entre casos de uso que permite incluir el comportamiento de uno (**caso de uso de inclusión**) en el flujo de otro (**caso de uso base**)
  - ◆ Especificar el punto exacto, en el caso base donde es necesario el comportamiento del caso de inclusión

# Relación <<include>>

caso de uso base



Caso de uso  
de inclusión

# Relación <<include>>

## ■ Funcionamiento

- ◆ El caso base se ejecuta hasta el punto de inclusión
  - `include(casoUsoInclusion)`
- ◆ El control pasa al caso de inclusión, que se ejecuta de manera normal
- ◆ El control vuelve al siguiente paso del caso base

## ■ ¡¡Cuidado!!

- ◆ El caso base no está completo sin todos los casos de inclusión
  - Los casos de inclusión pueden ser completos o un fragmento de comportamiento

# Relación <<include>>

## ■ Ejemplo

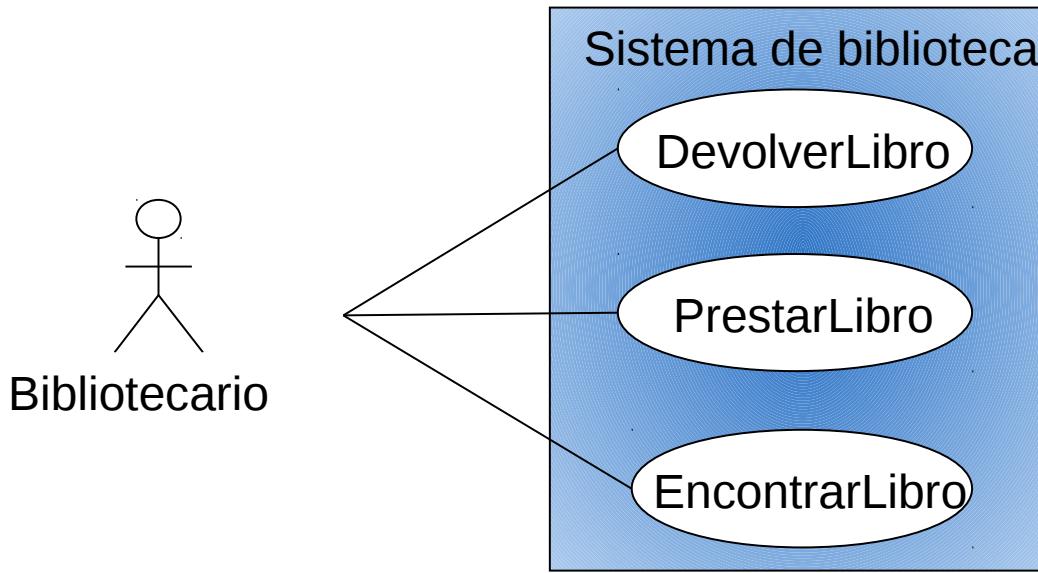
<p><b>Use case:</b> ChangeEmployeeDetails</p> <p><b>Brief description:</b> The Manager changes the employee details.</p> <p><b>Actors:</b> Manager</p> <p><b>Preconditions:</b> 1. The Manager is logged on to the system.</p> <p><b>Main flow:</b> 1. <code>include( FindEmployeeDetails ).</code> → 2. The system displays the employee details. ← 3. The Manager changes the employee details. ...</p> <p><b>Postconditions:</b> 1. The employee details have been changed.</p> <p><b>Alternative flows:</b> None</p>	<p><b>Use case:</b> FindEmployeeDetails</p> <p><b>Brief description:</b> The Manager finds the employee details.</p> <p><b>Actors:</b> Manager</p> <p><b>Preconditions:</b> 1. The Manager is logged on to the system.</p> <p><b>Main flow:</b> 1. The Manager enters the employee's ID. 2. The system finds the employee details. ...</p> <p><b>Postconditions:</b> 1. The system has found the employee details.</p> <p><b>Alternative flows:</b> None</p>
--	--

- ◆ En este ejemplo, el caso de uso es completo y puede ser activado por un actor

# Relación <<extend>>

## ■ Objetivo

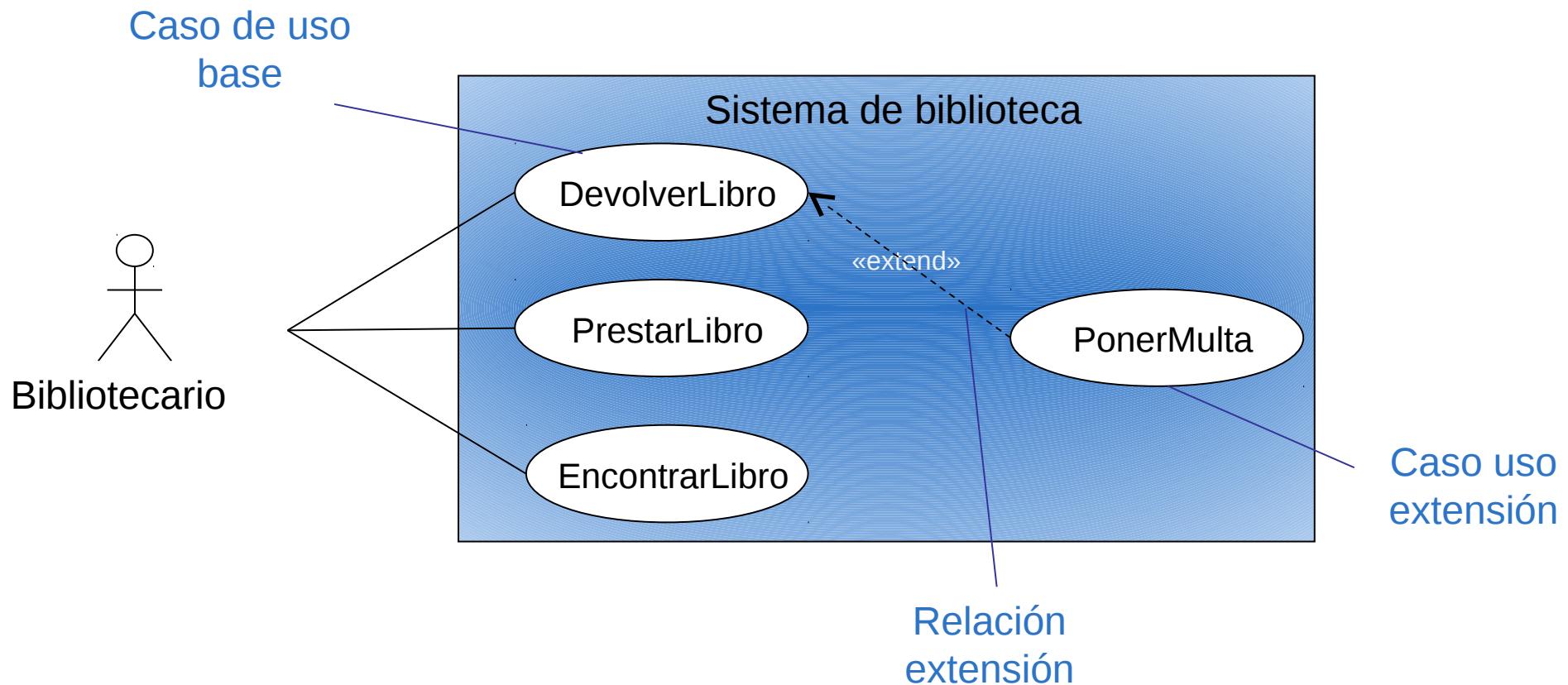
- ◆ Proporcionar un medio para insertar un nuevo comportamiento en un caso de uso existente



**¿Qué ocurre si el libro se devuelve tarde?**

# Relación <<extend>>

## ■ Objetivo



# Relación <<extend>>

## ■ ¿Cómo?

- ◆ Caso de uso base

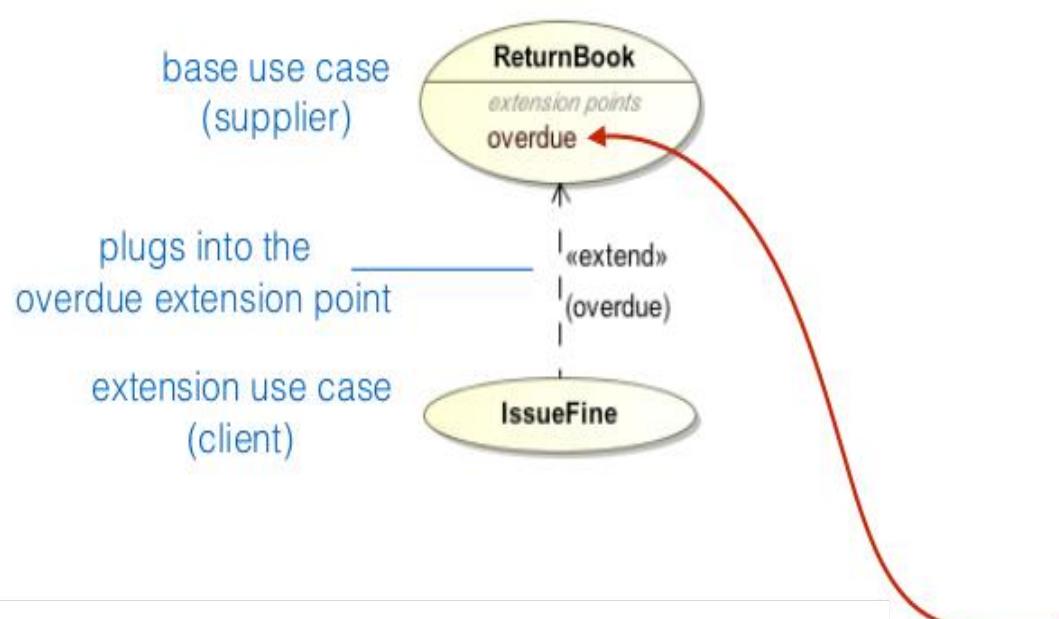
- Proporciona un conjunto de “anclajes” (puntos de extensión) donde se añadirá el nuevo comportamiento.
    - No sabe nada de los casos de uso de extensión

- ◆ Caso de uso extensión

- Proporciona un conjunto de segmentos de inserción que pueden ser insertados en los “anclajes”



# Relación <<extend>>



<b>Use case:</b> ReturnBook
<b>Brief description:</b> The Librarian returns a borrowed book.
<b>Actors:</b> Librarian
<b>Preconditions:</b> 1. The Librarian is logged on to the system.
<b>Main flow:</b> 1. The Librarian enters the borrower's ID number. 2. The system displays the borrower's details including the list of borrowed books. 3. The Librarian finds the book to be returned in the list of books. extension point: overdue 4. The Librarian returns the book.
<b>Postconditions:</b> 1. The book has been returned.
<b>Alternative flows:</b> None

# Relación <<extend>>

## ■ Características

- ◆ El caso base está completo sin las extensiones
  - A diferencia de los <<include>>
- ◆ Los puntos de extensión no son parte del flujo
  - No se numeran
- ◆ Puede haber más de un punto de extensión
  - La relación <<extend>> puede especificar a cual de los puntos de extensión está conectado el caso de uso de extensión

# Relación <<extend>

## ■ Casos de uso de extensión

- ◆ Mismo número de segmentos que de puntos de extensión especificados en la relación
  - ◆ Sólo un trozo de comportamiento
    - No tienen sentido por sí solos
  - ◆ Precondición
    - Debe satisfacerse para que el segmento se ejecute
  - ◆ Postcondición
    - Estado del sistema después de la ejecución del segmento

**Use case:** IssueFine

**Brief description:** The system calculates and prints out a fine.

**Actors:** None

**Preconditions:**  
1. The returned book is overdue.

**Main flow:**  
1. The system calculates the fine.  
2. The system prints out the fine.

**Postconditions:**  
1. The system has printed out the fine.

**Alternative flows:**  
None

<b>Use case:</b> IssueFine	
<b>Brief description:</b>	The system calculates and prints out a fine.
<b>Actors:</b>	None
<b>Preconditions:</b>	<ul style="list-style-type: none"><li>► 1. The returned book is overdue.</li></ul>
<b>Main flow:</b>	<ul style="list-style-type: none"><li>1. The system calculates the fine.</li><li>2. The system prints out the fine.</li></ul>
<b>Postconditions:</b>	<ul style="list-style-type: none"><li>1. The system has printed out the fine.</li></ul>
<b>Alternative flows:</b>	None

# Relación <<extend>>

## ■ Funcionamiento

- ◆ El caso de uso base hasta el primer punto de extensión
  - extension point: overdue
- ◆ El control pasa al caso de uso de extensión, si hay alguno conectado en ese punto, que se ejecuta con normalidad
- ◆ El control vuelve al siguiente paso en el caso de uso base

## ■ El caso de uso base es independiente de las extensiones que puedan estar o no conectadas

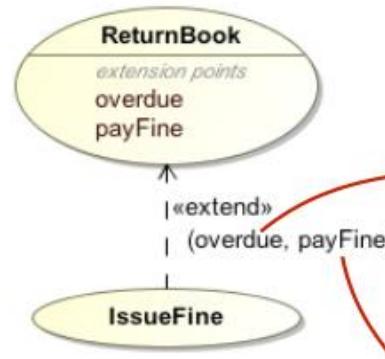
# Relación <<extend>>

<b>Use case:</b> ReturnBook
<b>Brief description:</b> The Librarian returns a borrowed book.
<b>Actors:</b> Librarian
<b>Preconditions:</b> 1. The Librarian is logged on to the system.
<b>Main flow:</b> 1. The Librarian enters the borrower's ID number. 2. The system displays the borrower's details including the list of borrowed books. 3. The Librarian finds the book to be returned in the list of books. extension point: overdue 4. The Librarian returns the book.
<b>Postconditions:</b> 1. The book has been returned.
<b>Alternative flows:</b> None

<b>Use case:</b> IssueFine
<b>Brief description:</b> The system calculates and prints out a fine.
<b>Actors:</b> None
<b>Preconditions:</b> 1. The returned book is overdue.
<b>Main flow:</b> 1. The system calculates the fine. 2. The system prints out the fine.
<b>Postconditions:</b> 1. The system has printed out the fine.
<b>Alternative flows:</b> None

# Relación <<extend>>

## ■ Múltiples segmentos de inserción



Si la relación <<extend>> especifica más de un punto de extensión , el caso de uso de extensión debe tener el mismo número de segmentos y estos se insertarán en orden

<b>Use case:</b> IssueFine
<b>Brief description:</b> Flow 1: The system calculates and prints out a fine. Flow 2: The Librarian accepts payment for a fine.
<b>Actors:</b> Librarian
<b>Segment 1</b>
<b>Preconditions 1:</b> 1. The returned book is overdue.
<b>Main flow 1:</b> 1. The system calculates the fine. 2. The system prints out the fine.
<b>Postconditions 1:</b> 1. The system has printed out the fine.
<b>Segment 2</b>
<b>Preconditions 2:</b> 1. A fine is due from the borrower.
<b>Main flow 2:</b> 1. The Librarian accepts payment for the fine from the borrower. 2. The Librarian enters the paid fine in the system. 3. The system prints out a receipt for the paid fine.
<b>Postconditions 2:</b> 1. The fine is recorded as paid. 2. The system has printed a receipt for the fine.

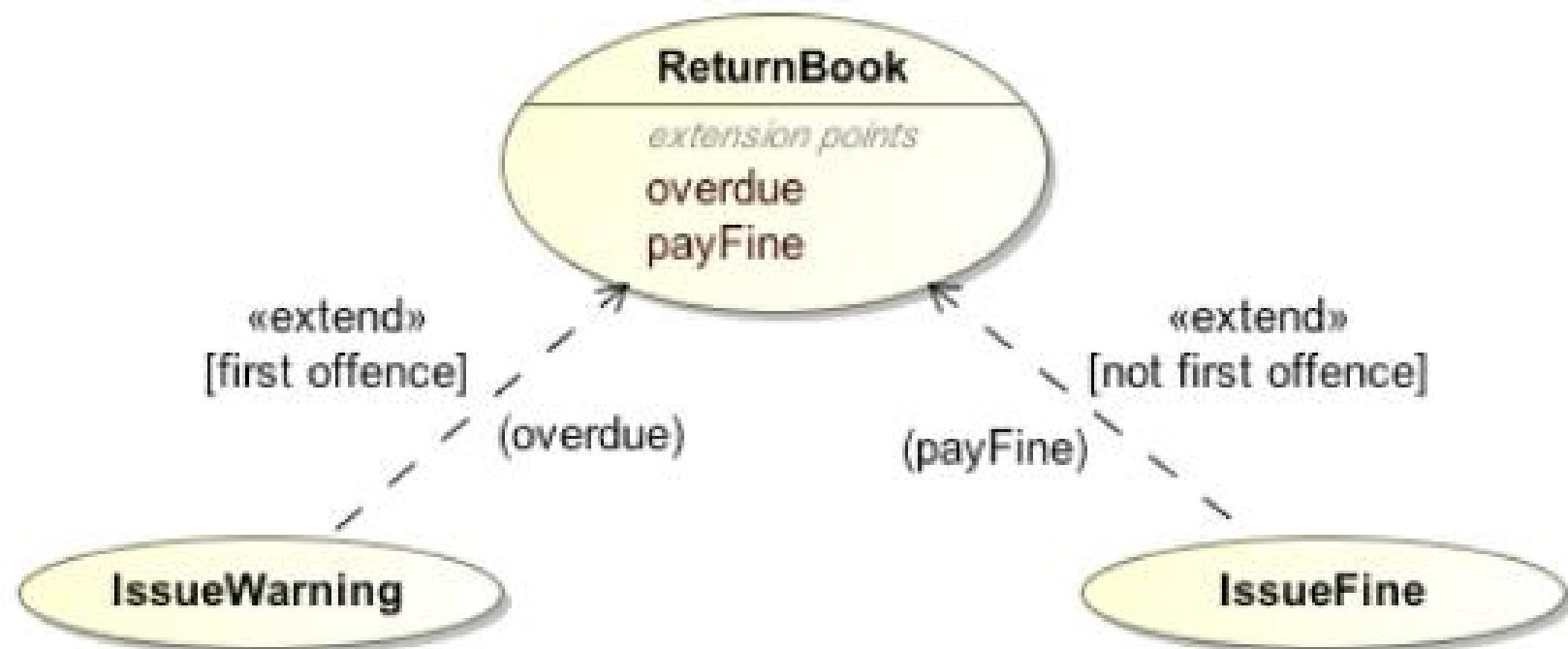
# Relación <<extend>>

## ■ Extensiones condicionales

- ◆ Podemos especificar condiciones sobre las relaciones <<extend>>
- ◆ Las condiciones son expresiones booleanas y la inserción sólo se produce si la condición se evalúa a true
- ◆ Se puede conseguir el mismo efecto con las precondiciones y postcondiciones en el caso de uso de extensión, pero las condiciones en las relaciones son más explícitas

# Relación <<extend>>

## ■ Extensiones condicionales



# Aspectos avanzados

## ■ ¿Cuándo?

- ◆ Cuando simplifiquen el modelo de casos de uso
  - Tiene que entenderlos todo los implicados
    - Capturan requisitos
  - Mejor cuantas menos características avanzadas
- ◆ Mejoren la claridad global del modelo de casos de uso
  - Mantener los modelos los más sencillos posibles
  - No emocionarse con los <<extend>> e <<include>>

# Sugerencias

- Casos de uso breves y sencillos
  - ◆ Sólo el detalle necesario para capturar los requisitos
  - ◆ El flujo no debe ocupar más de una hoja de papel
    - Simplificar el texto
    - ¿Flujos alternativos?
    - ¿Más de un caso de uso?
- Centrarse en el qué, no en el cómo
  - ◆ 4 El sistema pide al Cliente que confirme el pedido
  - ◆ 5 El cliente pulsa el botón de Aceptar 
  - ◆ 5 El cliente acepta el pedido 

# Sugerencias

- Evitar la descomposición funcional
  - ◆ Casos de uso de alto nivel , bajo nivel y primitivos

