

## **TEMA 3.2: CASOS DE USO**

# Índice

- Introducción
- Elementos del diagrama
- Actores
- Caso de uso
- Identificación de casos de uso
- Descripción de casos de uso
- Conectores
- Diagrama de casos de uso
- Ejercicios

# Introducción

- Se debe comprender el **punto de vista del usuario** para generar **sistemas útiles y funcionales**
- El **cumplimiento de los requisitos** es la clave para desarrollar un sistema que satisfaga las necesidades de los usuarios
- El **modelado** de un sistema desde el punto de vista del usuario se realiza **mediante los casos de uso**
- El **objetivo** de los casos de uso es describir la manera en que se usará el sistema:
  - Describir sus **funcionalidades esenciales**

# Introducción

- Los **diagramas de casos de uso** fueron concebidos por **I. Jacobson** -Objectory/OOSE (Jacobson et al. 92) y describen una **interacción entre un agente externo (actor) y el sistema**
- Permiten **definir los límites del sistema** y las relaciones entre el sistema y el entorno
- Produce algo de valor para algún actor, como por ejemplo, el cálculo de algún resultado
- Describen **qué hace el sistema** pero no cómo lo hace

# Introducción

- **¿Para qué sirven los casos de uso?**
- Para **expresar el comportamiento deseado** del sistema (especificación de requisitos)
  - Independientemente de cómo se implemente
- Como **medio de comprensión** del sistema para desarrolladores, usuarios finales y expertos del dominio
- Para **establecer las necesidades** según la categoría de los usuarios
- Para la generación de casos de prueba
  - A partir de los escenarios de un caso de uso

# Elementos del diagrama

- Casos de uso

Nombre del Caso de Uso

- Actores



- Conectores



Generalización

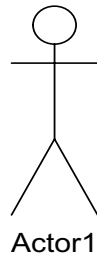


Inclusión o extensión



Asociación

# Actores



- Un actor representa a **alguien o algo** que actúa sobre el sistema que se está desarrollando
  - Personas u otro software que actúa sobre el sistema
- Los actores son **externos al sistema** que vamos a desarrollar
  - Por lo tanto, al identificar actores estamos empezando a delimitar el sistema, y a definir su alcance.
- Definir el alcance del sistema debe ser el primer objetivo de todo analista

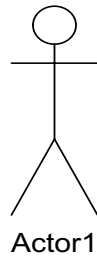
# Actores



- Un usuario no es exactamente lo mismo que un actor:
  - Un actor es un tipo de rol
  - Un usuario es una persona que cuando usa el sistema asume un tipo de rol
- Un usuario puede acceder al sistema como distintos actores (con distintos roles):
  - Diferentes perfiles de usuario para un sistema operativo
- Los perfiles de usuario serían el equivalente a los actores en un diagrama de casos de uso



# Actores



- Otro sistema que interactúa con el que estamos desarrollando también es un actor
- Por ejemplo:
  - Si nuestro sistema debe generar asientos contables para ser procesados por el sistema de contabilidad
  - Entonces el sistema de contabilidad será un actor, que usa los servicios de nuestro sistema

# Caso de uso

Nombre del Caso de Uso

- Un caso de uso es un comportamiento del sistema que produce un resultado de interés para algún actor
- Un **caso de uso** representa un **requisito funcional**
- En UML los casos de uso son siempre iniciados por un actor, y no deben iniciarse espontáneamente desde el interior del sistema
- Un caso de uso describe no sólo una funcionalidad, sino también una interacción entre un actor y el sistema bajo la forma de un flujo de eventos.

# Caso de uso

Nombre del Caso de Uso

- Uno de los beneficios de los casos de uso es que muestran los límites del sistema respecto al mundo exterior
- Los **actores** están **fuera del sistema**
- Los **casos de uso** están **dentro del sistema**
- Se utilizará un rectángulo (con el nombre del sistema dentro de él) para representar los límites del sistema
- El rectángulo envolverá los casos de uso del sistema

# Caso de uso

Nombre del Caso de Uso

- Cada **caso de uso** es una **colección de escenarios**
- Cada **escenario** es una **secuencia de pasos**
- Los pasos de cada escenario no aparecen en el diagrama
- Cada escenario tendrá su propia página donde se listará en modo texto:
  - Actor que inicia el caso de uso
  - Condiciones previas para el caso de uso
  - Pasos en el escenario
  - Condiciones posteriores cuando se finaliza el caso de uso
  - El actor que se beneficia del caso de uso

# Caso de uso

Nombre del Caso de Uso

**Identificador:** Realiza\_pedido

**Actores que lo inician:** Cliente y Agente

**Precondiciones:** Un cliente registrado en el sistema ha accedido correctamente al sistema.

**Secuencia de eventos de flujo:**

1. El cliente introduce su nombre y dirección.
2. Si el cliente introduce el ZIP, el sistema introduce la ciudad y región.
3. El cliente introduce los códigos de los productos que desea incluir en el pedido.
4. El sistema aporta la descripción y el precio del producto.
5. El sistema almacena temporalmente la lista de productos incluidos en el pedido.
6. El cliente introduce la información de la tarjeta de pago.
7. El cliente pulsa el control Ejecuta.
8. El sistema verifica la información, almacena el pedido temporalmente y requiere confirmación del banco. Si la información es incorrecta, el sistema requiere su corrección al cliente.
9. Cuando el pago es confirmado, se acepta el pedido, se le asigna un ID que se retorna al cliente.

**Postcondiciones:** Si el pedido no ha sido cancelado, es registrado en el sistema y confirmado al cliente.

# Identificación de casos de uso

- Un error común en la identificación de los casos de uso consiste en representar los pasos, las operaciones o las transacciones individuales como casos
- Por ejemplo:
  - En un terminal de punto de venta (TPV) se puede definir (incorrectamente) un caso de uso “Imprimir el recibo”, cuando esta operación no es más que un paso de un proceso más amplio del caso “Comprar productos”
- Un caso de uso es una descripción de un proceso de principio a fin relativamente amplia y suele abarcar muchos pasos o transacciones

# Identificación de casos de uso

- Un método para identificar los casos de uso se basa en los actores:
  - Se identifican los actores relacionados con un sistema o empresa
  - Para cada actor, se identifican los procesos que inician o en los que participan
- El nombre de un caso de uso:
  - Se identifica con un verbo seguido del objeto o entidad del sistema afectado por el caso (Realizar pedido, Obtener listado clientes, etc)

# Identificación de casos de uso

- Características:
  - Están expresados **desde el punto de vista del actor.**
  - Se documentan con texto informal.
  - Describen tanto lo que hace el actor como lo que hace el sistema cuando interactúa con él.
  - Son iniciados por un único actor.
  - Están acotados al uso de una determinada funcionalidad –claramente diferenciada– del sistema.



# Identificación de casos de uso

- ¿Qué es una “funcionalidad claramente diferenciada”?
- Por ejemplo:
  - ¿Ingresar pedidos es un caso de uso y autorizarlos es otro?
  - ¿Cancelar los pedidos, es otro caso de uso, o es parte del caso de uso referido al ingreso de pedidos?
- En principio la respuesta a todas estas preguntas es que son todos casos de uso distintos.

# Identificación de casos de uso

- La regla general es: *“una función del sistema es un caso de uso si se indica explícitamente al sistema que uno quiere acceder a esa función”*.
- Por ejemplo:
  - Si se quiere dar de alta un pedido, se accederá a la funcionalidad de alta de pedidos del sistema
- Sin embargo:
  - Si se quiere dar de alta un campo del pedido, no hay que indicar al sistema que se quiere acceder a esa función
    - Dar de alta un campo de un pedido es una función que forma parte de un caso de uso mayor: dar de alta un pedido

# Identificación de casos de uso

- Esta regla:
  - No debe seguirse al pie de la letra
- Por ejemplo:
  - Un sistema en el cual los usuarios pueden ver un pedido, y tienen disponibles funciones para ver el siguiente pedido, el anterior, el último y el primero
  - El actor debe indicar al sistema que quiere acceder a cada una de esas funciones
    - Según nuestra regla serían todas ellas casos de uso distintos
  - Sin embargo, es mucho más práctico definir un único caso de uso *Navegando pedidos*

# Descripción de casos de uso

- Se documentan con texto informal. En general, se usa una lista numerada de los pasos que sigue el actor para interactuar con el sistema
- A continuación se muestra una parte simplificada de la descripción del caso de uso “Ingresando Pedido”.

# Descripción de casos de uso

<b>Caso de Uso:</b> Ingresando Pedido.
<b>Actor:</b> Empleado de Ventas.
1) El cliente se comunica con la oficina de ventas, e informa su número de cliente
2) El oficial de ventas ingresa el número de cliente en el sistema
3) El sistema obtiene la información básica sobre el cliente
4) El cliente informa el producto que quiere comprar, indicando la cantidad
5) El sistema obtiene la información sobre el producto solicitado, y confirma su disponibilidad.
6) Se repite el paso 4) hasta que el cliente no informa más productos
7) ...

# Descripción de casos de uso

- Alternativas:

- Son errores o excepciones que aparecen durante la ejecución de un caso de uso
- No tienen sentido por sí mismas, fuera del contexto del caso de uso en el que ocurren
- Por ejemplo, mientras se toma un pedido, el cliente puede solicitar un producto que esté agotado.
  - El sistema deberá en este caso informar esta situación al empleado que toma el pedido

# Descripción de casos de uso

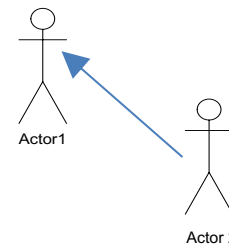
Caso de Uso: Ingresando Pedido	
Actor: Empleado de ventas	
Curso Normal	Alternativas
1) El cliente se comunica con la oficina de ventas, e informa su número de cliente	
2) El oficial de ventas ingresa el número de cliente en el sistema	
3) El sistema obtiene la información básica sobre el cliente	3.1 Si no está registrado, se le informa que debe registrarse en la oficina de clientes
4) El cliente informa el producto que quiere comprar, indicando la cantidad	
5) El sistema obtiene la información sobre el producto solicitado, y confirma su disponibilidad.	5.1 Si no hay disponibilidad del producto, el sistema informa la fecha de reposición
6) Se repite el paso 4) hasta que el cliente no informa más productos	
...	

# Conectores

- Muestran la manera en que los actores y los casos de uso están relacionados
- Hay tres tipos de conectores:

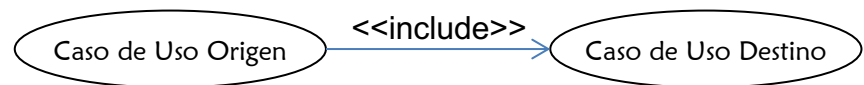
- Generalización

- Entre pares de actores  
o entre pares de casos de uso



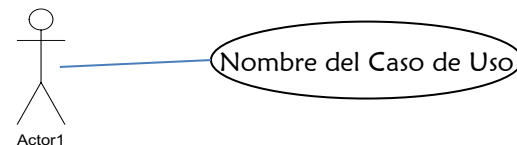
- Inclusión o extensión

- Entre casos de uso



- Asociación

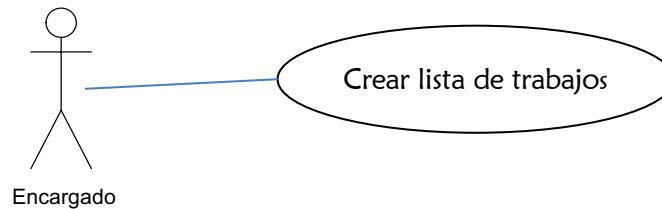
- Entre actores y casos de uso





# Conectores

- Asociación
  - Se utiliza para mostrar qué actores se relacionan con los casos de uso



- Sirve para identificar las funcionalidades asociadas a diferentes tipos de actores
- La relación es bidireccional

# Conectores

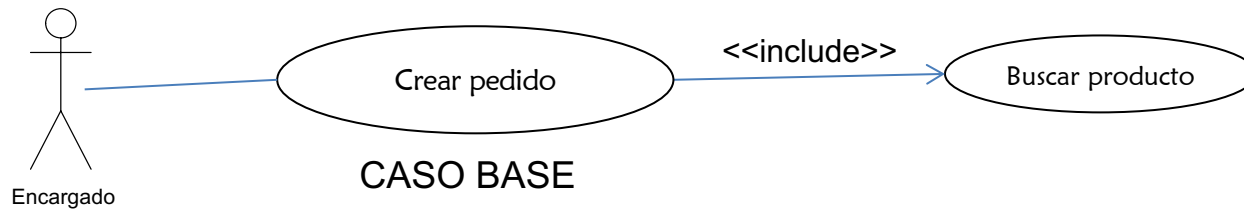
- Asociación
  - Hay un actor que inicia un caso de uso y otro (posiblemente el que lo inició, pero no necesariamente) que recibirá algo de valor
  - El actor que inicia el caso de uso se sitúa a la izquierda del caso de uso
  - El actor que recibe el valor se sitúa a la derecha del caso de uso

# Conectores

- Inclusión y extensión
  - Se utilizan para mostrar una relación de dependencia entre dos casos de uso
  - Se debe añadir el estereotipo <<include>> o <<extend>> al conector para indicar el tipo de dependencia

# Conectores

- Inclusión



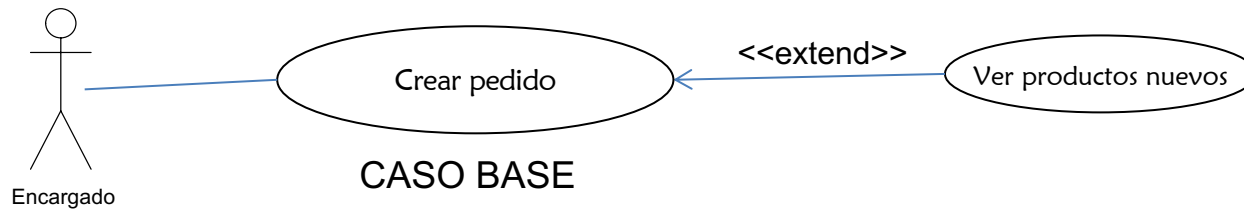
- Un caso de uso base incorpora explícitamente el comportamiento de otro caso de uso en el lugar especificado en el caso base
- Se usa para evitar describir el mismo flujo de eventos repetidas veces, poniendo comportamiento común en un caso de uso aparte

# Conectores

- Características de la Inclusión:
  - Aparecen como funcionalidad común
  - Los casos incluidos son casos de uso en sí mismos
  - Cuando el caso base se ejecuta **SIEMPRE** se ejecuta el caso incluido
  - Esto marca la diferencia con las extensiones, que son opcionales

# Conectores

- Extensión



- El caso base puede necesitar o no la funcionalidad del caso de uso extendido
- A diferencia de la inclusión, la extensión es **OPCIONAL**
- La flecha apunta siempre al caso de uso base

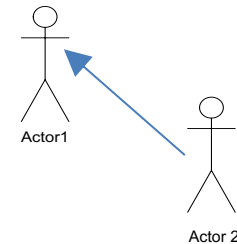
# Conectores

- Características de la relación de Extensión:
  - La extensión sólo se puede realizar en puntos indicados de manera específica dentro de la secuencia del caso de uso base
  - A estos puntos se les denomina ***puntos de extensión***
  - No necesariamente provienen de un error o excepción
- Jacobson ejemplifica los casos de uso con ir a cenar a un restaurante:
  - Para él, tomar café después de cenar es un ejemplo de una extensión

# Conectores

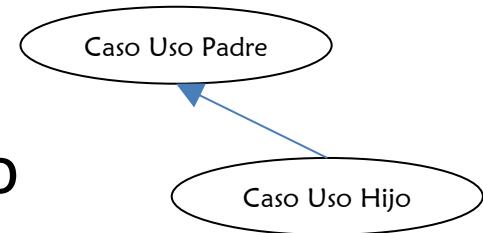
- Generalización entre actores

- El actor hijo hereda las características del actor padre
- El actor hijo puede añadir o redefinir características del actor padre



- Generalización entre casos de uso

- El caso hijo hereda el comportamiento y significado de caso de uso padre
- El hijo puede añadir o redefinir el comportamiento del padre





# ¿Extensión o alternativa?

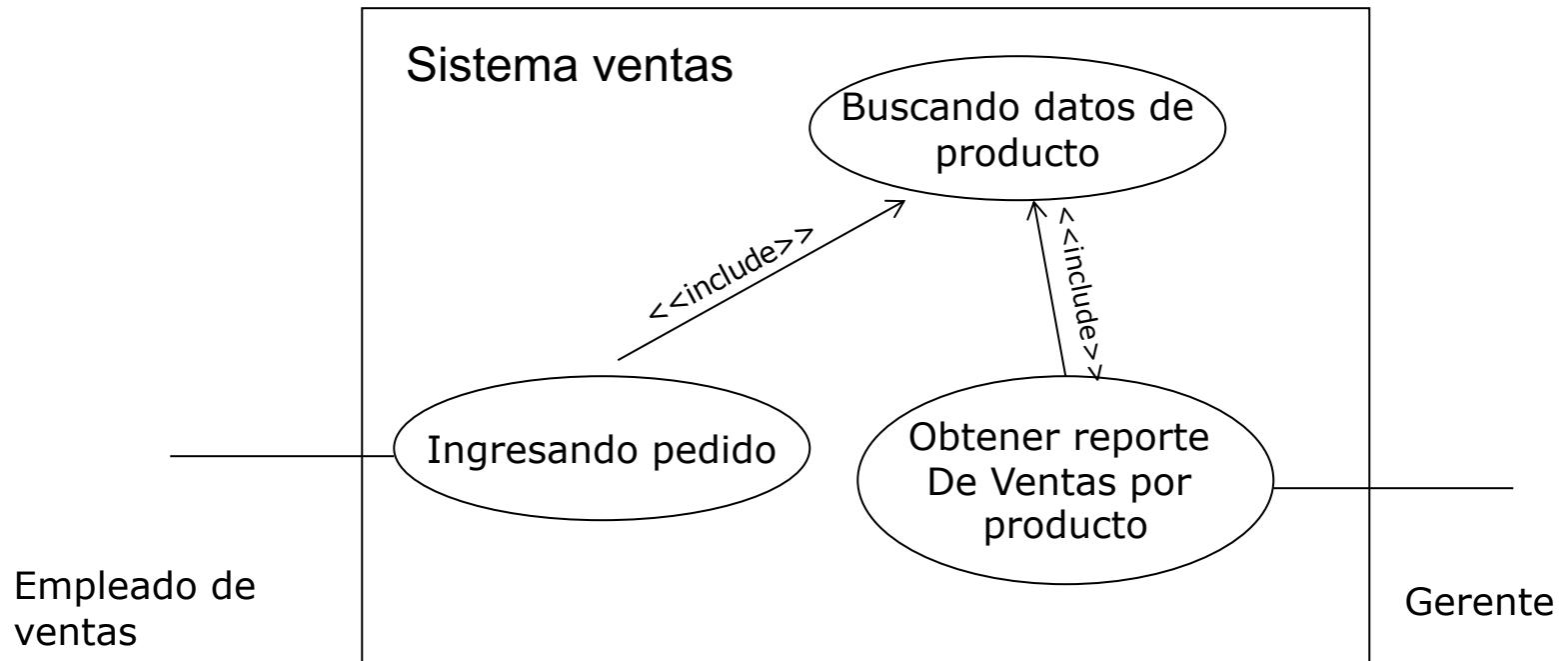
- En la práctica aparecen dudas si considerar algo optativo como:
  - Una alternativa
  - Una extensión
- Como regla aproximada:
  - Si algo opcional debe ser expresado con **más de un paso**, seguramente es una **extensión** y no una alternativa

# Diagrama de casos de uso

- Pasos para construir un diagrama:
  - Identificar los **límites** (alcance) del sistema.
  - Identificar los **actores** principales.
  - Para cada uno, identificar sus **objetivos**.
  - Definir **casos de uso** que satisfagan sus objetivos.

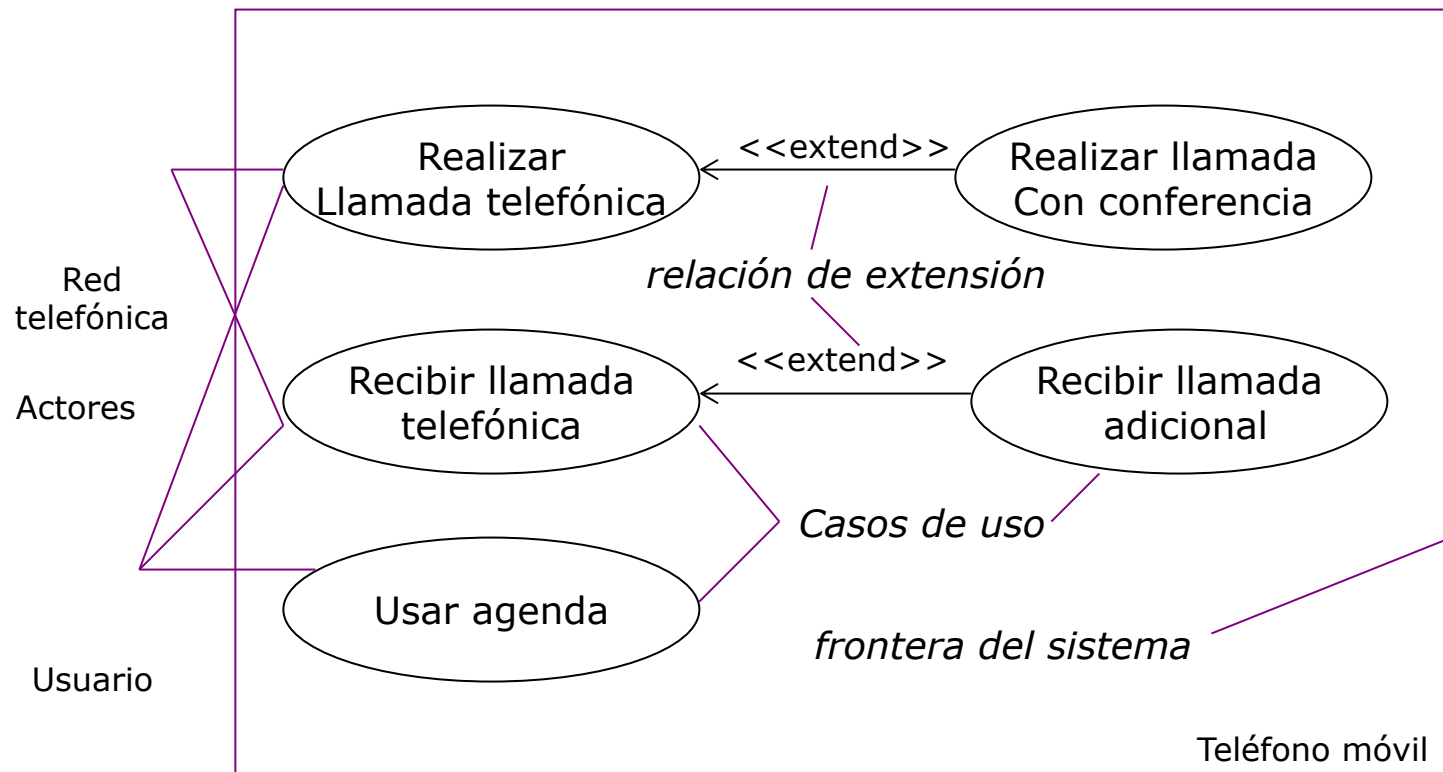
# Diagrama de casos de uso

- Ejemplo relaciones de inclusión:



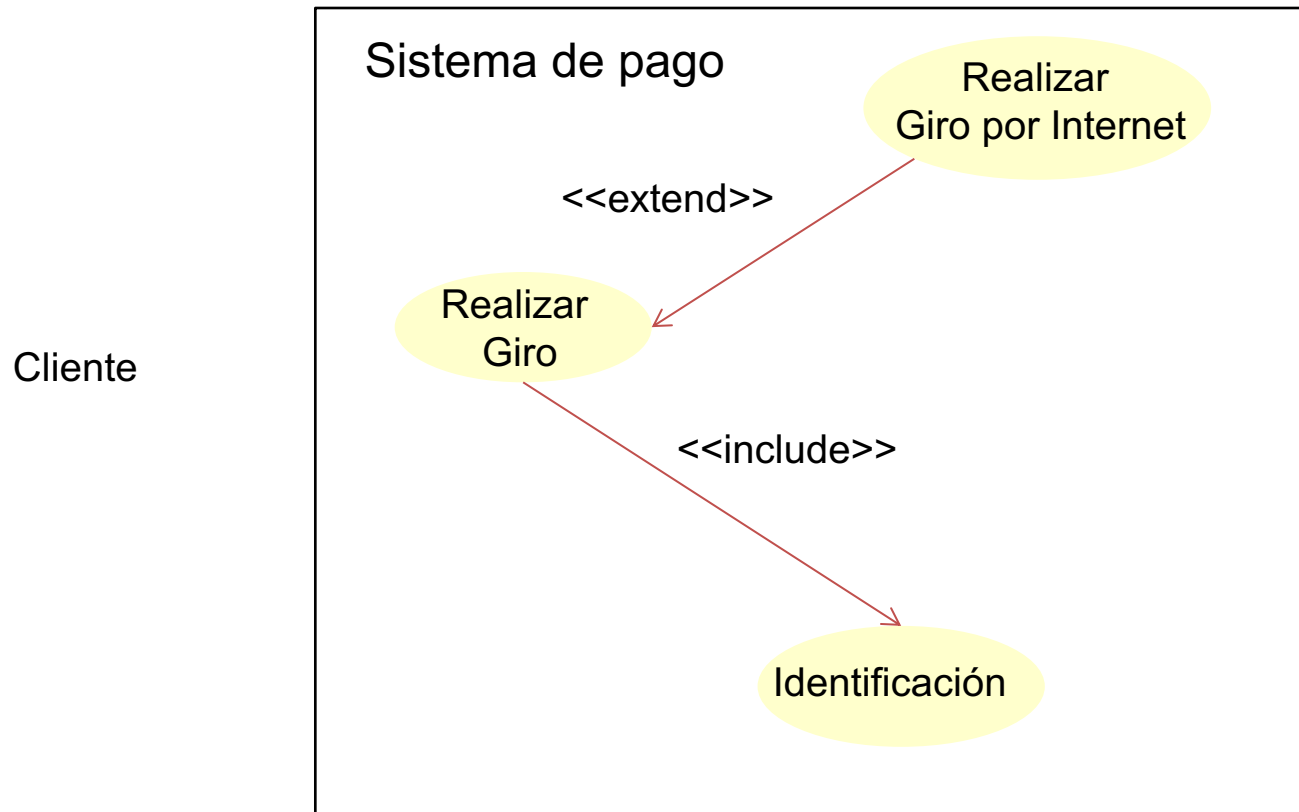
# Diagrama de casos de uso

- Ejemplo relaciones extensión:



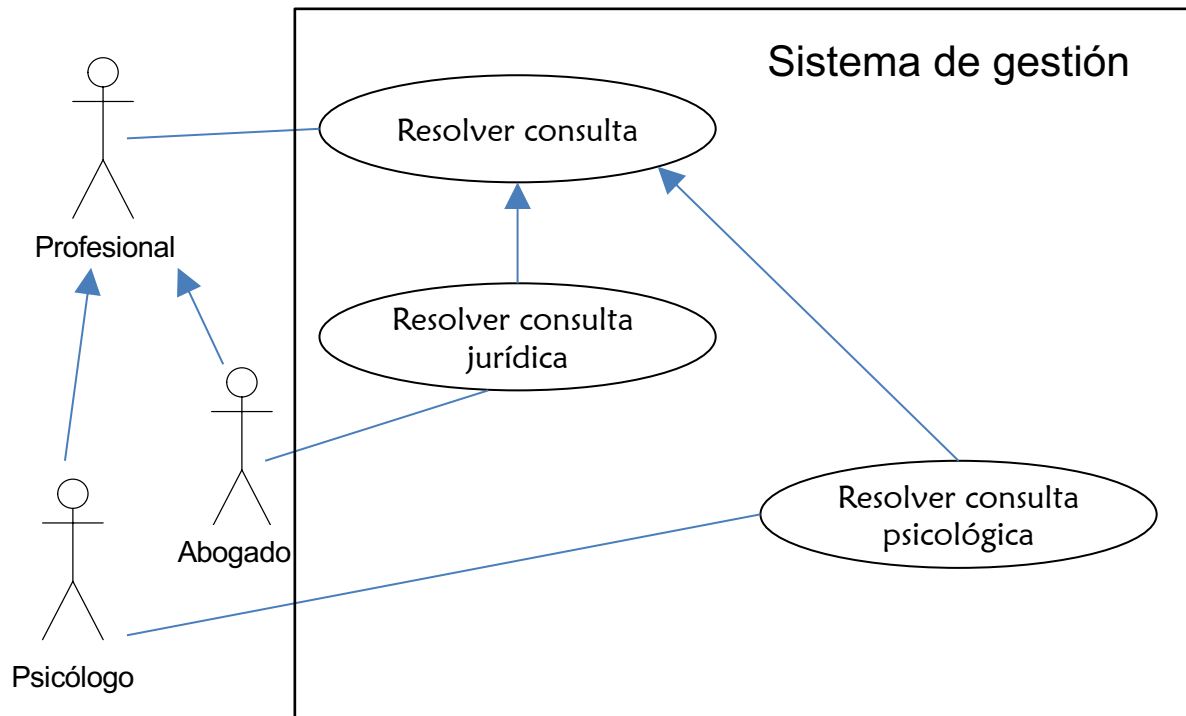
# Diagrama de casos de uso

- Ejemplo de todas las relaciones:



# Diagrama de casos de uso

- Ejemplo de generalizaciones:



# Ejercicio

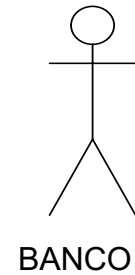
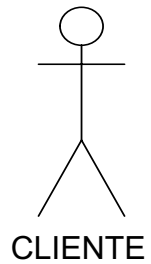
- Una entidad bancaria necesita ayuda para modelar el sistema que hará funcionar sus nuevos cajeros automáticos portátiles conectados con la entidad. Los cajeros permitirán al usuario realizar sólo las operaciones más simples: retirar, depositar y consultar saldo. Para ello hay que tener en cuenta que:
  - Al introducir la tarjeta es necesario validar la tarjeta e ingresar la clave del usuario
  - No se puede retirar más fondos de los que realmente hay, notificando de esta situación al usuario

# Ejercicio

- Una **entidad bancaria** necesita ayuda para modelar el **sistema** que hará funcionar sus nuevos **cajeros automáticos** portátiles conectados con la **entidad**. Los **cajeros** permitirán al **usuario** realizar sólo las **operaciones** más simples: retirar, depositar y consultar saldo. Para ello hay que tener en cuenta que:
  - Al introducir la **tarjeta** es necesario validar la tarjeta e ingresar la **clave** del usuario
  - No se puede retirar más **fondos** de los que realmente hay, notificando de esta situación al usuario



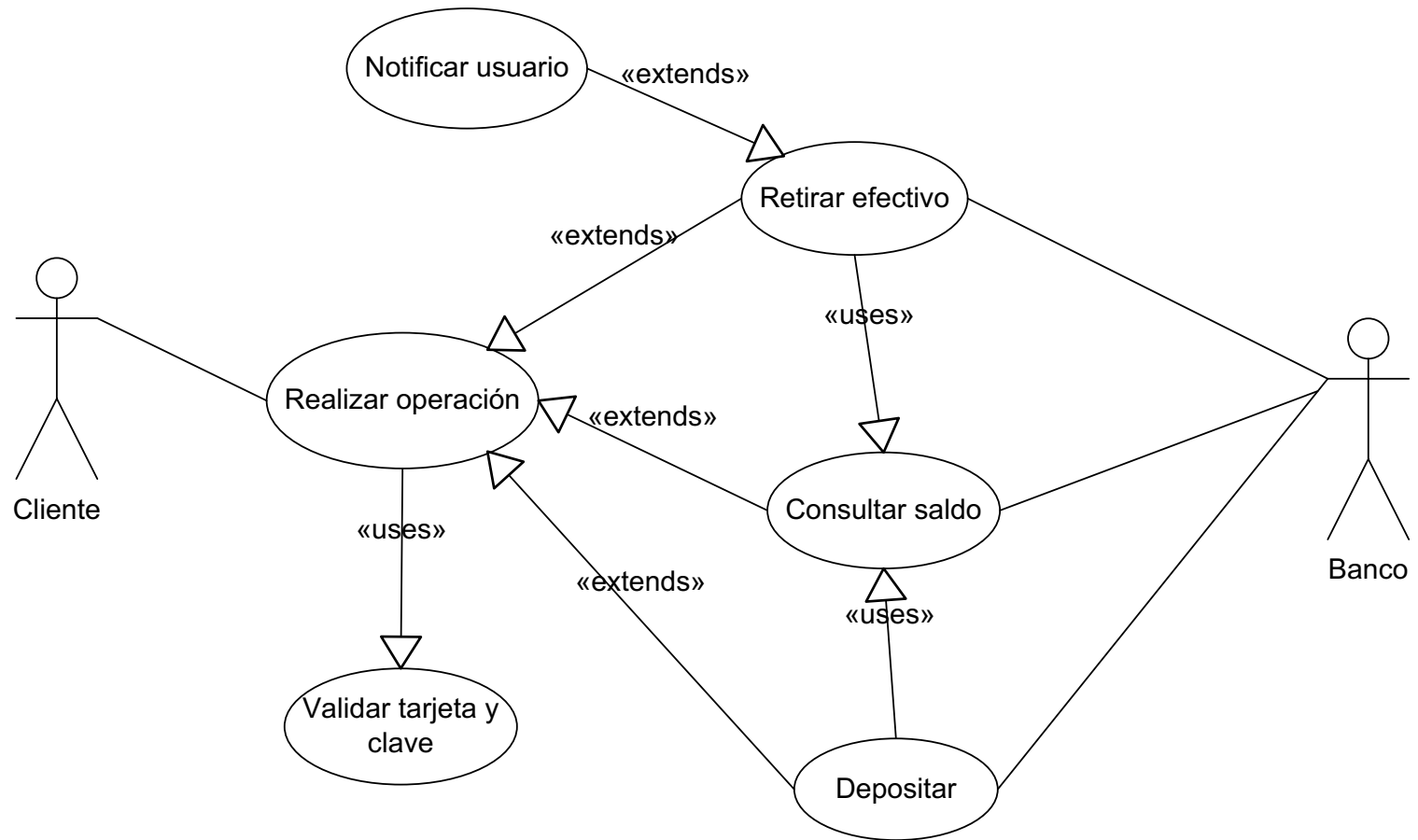
# Ejercicio



# Ejercicio

- Una entidad bancaria necesita ayuda para modelar el sistema que hará funcionar sus nuevos cajeros automáticos portátiles conectados con la entidad. Los cajeros permitirán al usuario **realizar sólo las operaciones** más simples: **retirar**, **depositar** y **consultar saldo**. Para ello hay que tener en cuenta que:
  - Al introducir la tarjeta es necesario **validar la tarjeta** e **ingresar la clave** del usuario
  - No se puede **retirar más fondos** de los que realmente hay, **notificando** de esta situación al usuario

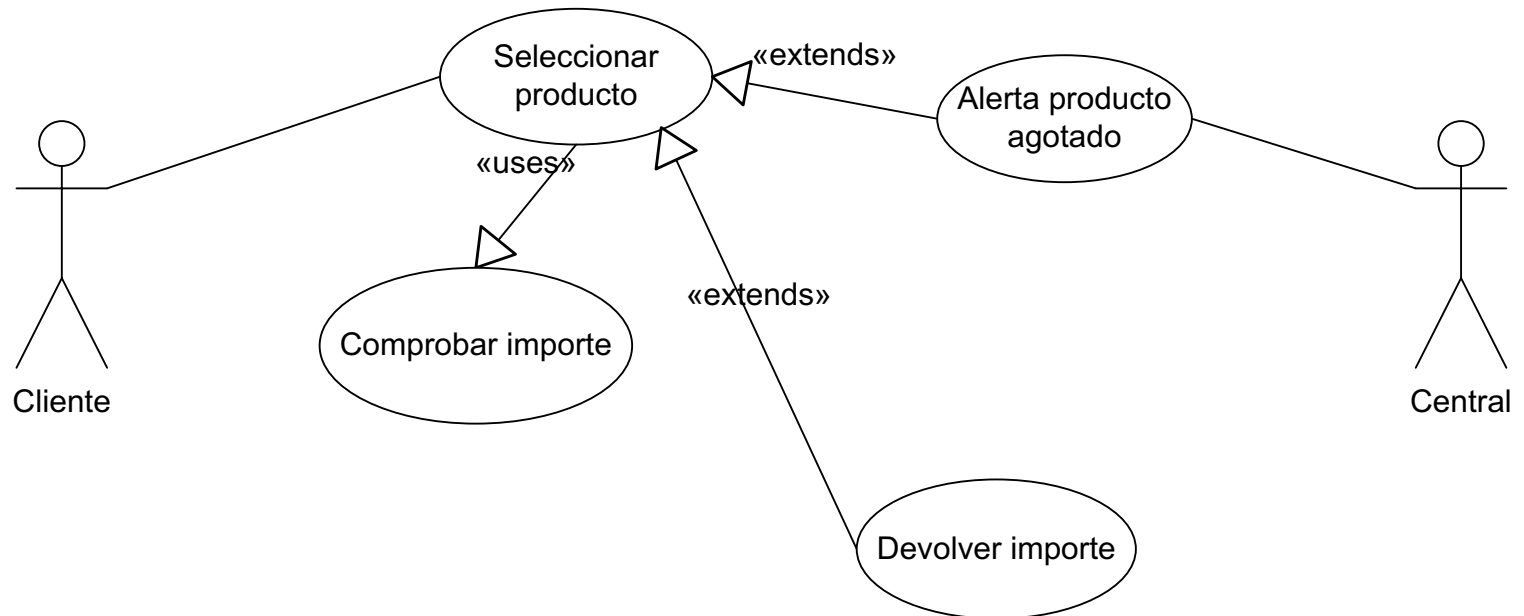
# Solución



# Ejercicio

- La empresa CaféOlé tiene planes para instalar una nueva máquina expendedora “inteligente” en el Aulario II. Inteligente porque cuando detecte que un cliente intenta comprar un producto agotado, se conectará automáticamente a la central de abastecimiento y dará aviso para realizar la reposición. Además, como buena máquina expendedora, debe devolver el dinero correspondiente y no dejar que se adquiera un producto si el dinero introducido es menor que el importe marcado en el producto.

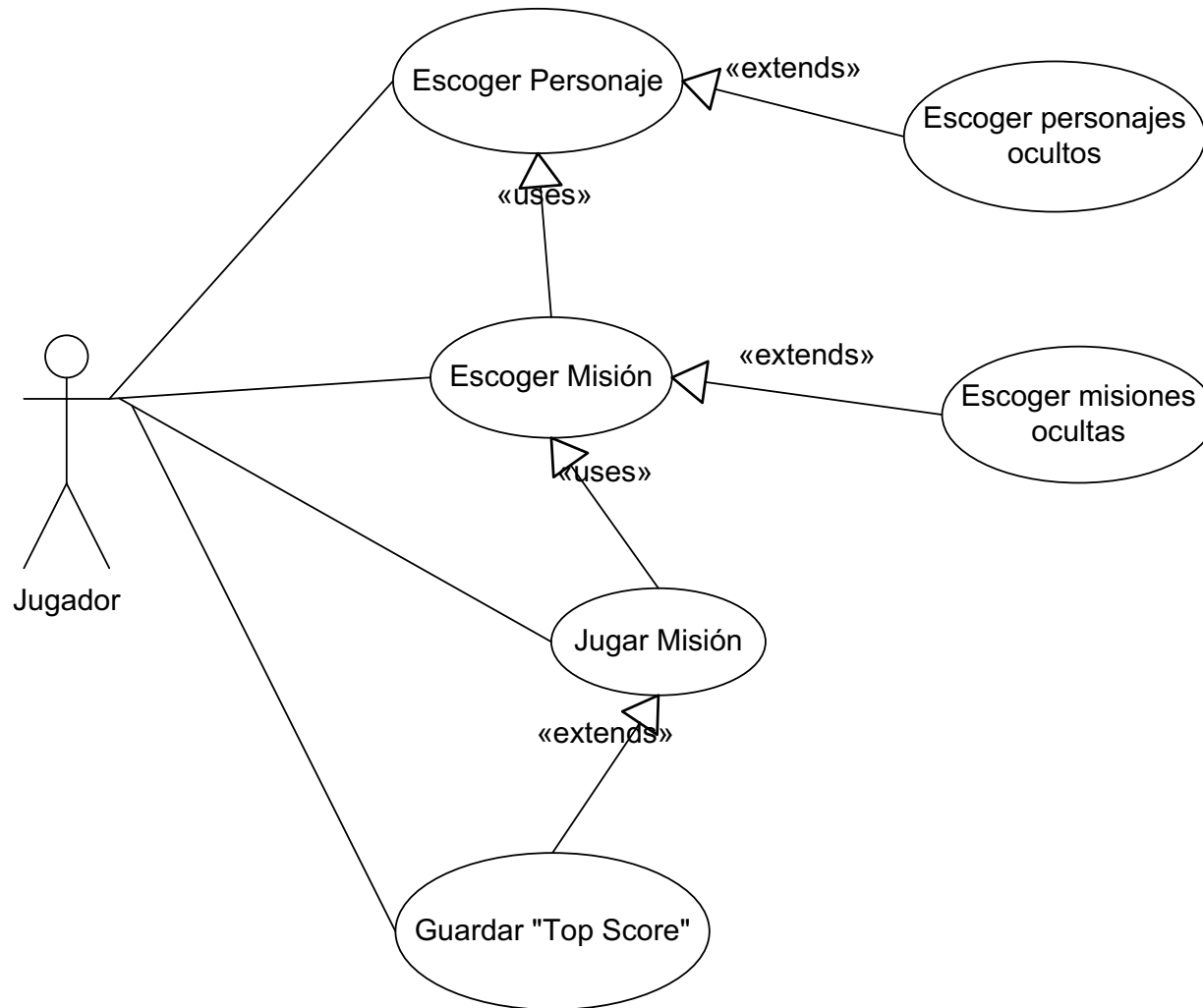
# Solución



# Ejercicio

- Para construir un emulador de videojuegos tipo arcade se solicita diseñar los casos de uso del sistema. En el emulador, el jugador puede escoger un personaje, una misión, jugar la misión y si logra una buena puntuación introducir su “top-score”. También se pide incluir los casos de uso necesarios para que el jugador que conoce los trucos active las claves para acceder a los personajes y misiones ocultas del juego.

# Solución

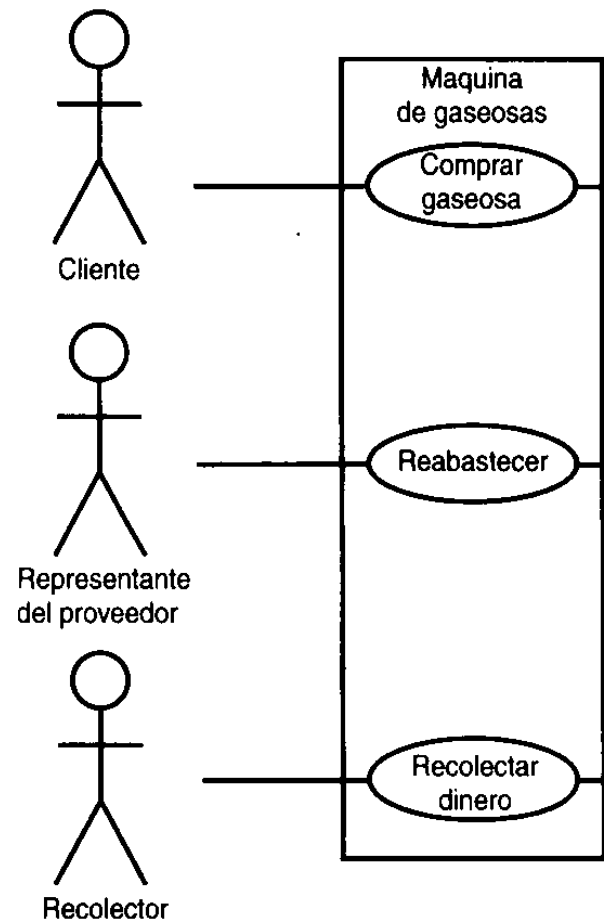


# Ejercicio

- Supongamos que estamos diseñando una máquina expendedora de refrescos:
  - Los clientes/usuarios deberán poder obtener el refresco solicitado
  - La máquina deberá permitir introducir monedas y devolver el cambio del dinero introducido
  - El reponedor o representante del proveedor podrá introducir nuevas bebidas
  - El encargado de la recaudación podrá recoger la recaudación y añadir monedas para devolver el cambio



# Ejercicio



# Preguntas

## **1. ¿Qué símbolo representa un caso de uso?**

- a. Una línea
- b. Una línea dirigida
- c. Un óvalo que contiene texto

## **2. Un actor solamente puede ser una persona.**

- a. Verdadero
- b. Falso

## **3. ¿Cómo se indica un estereotipo sobre un conector?**

- a. Texto entre un par de comillas angulares
- b. Texto llano próximo al conector
- c. La palabra estereotipo dentro del símbolo del óvalo

# Preguntas

- 4. Se usa una relación de inclusión para reutilizar el comportamiento modelado por otro caso de uso.**
  - a. Verdadero
  - b. Falso
- 5. Se usa una relación de extensión para modelar características opcionales del sistema.**
  - a. Verdadero
  - b. Falso
- 6. En una relación extendida, la flecha apunta hacia el**
  - a. caso de uso básico.
  - b. caso de uso de extensión.

# Resumen

- En UML cada caso de uso debe de tener al menos un actor
- Normalmente los diagramas de casos de uso contienen:
  - Casos de uso
  - Actores
  - Relaciones de dependencia, generalización y asociación

# Resumen

- Los **Casos de Uso** no son parte del diseño (cómo), sino parte del **análisis (qué)**
- Los Casos de Uso describen **qué hace el sistema** desde el **punto de vista de un observador externo**
- En una relación **<< extends >>**, un actor que lleve a cabo el caso de uso base puede realizar sus extensiones de forma **opcional**. Mientras que en una relación **<< include >>** el actor que realiza el caso de uso base realiza el caso de uso incluido de forma **obligatoria**

# Resumen

- **Actor** = Algo **con comportamiento** (persona, otro programa, organización...), que interactúa con el sistema
- **Escenario (instancia de caso de uso)** = **Secuencia de acciones e interacciones** entre los actores y el sistema con el sistema
- **Caso de Uso** = **Colección de escenarios** (éxito y fracaso) que describen actores que usan el sistema para conseguir un objetivo

# Bibliografía

- UML 2: iniciación, ejemplos y ejercicios corregidos. Laurent Debrauwer y Fien Van der Heyde
- UML2: Practique la modelización. Laurent Debrauwer y Naouel Karam
- UML gota a gota. Martin Fowler
- Manual de UML. Paul Kimmel