

## TEMA V

### UML (MODELIZACIÓN)

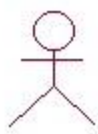
#### DIAGRAMA DE CASOS DE USO

Es un diagrama de comportamiento, permite visualizar los diferentes tipos de roles en un sistema y cómo esos roles interactúan con él. El diagrama de casos de uso representa la forma en cómo unos Clientes (Actores) operan con el sistema en desarrollo, además de la forma, tipo y orden en cómo los elementos interactúan (operaciones o casos de uso).

Este tipo de diagrama se utiliza para representar de forma general el funcionamiento de una determinada aplicación y quien interactúa con ella. Dicho de otra manera, muestra de manera visual las distintas funciones que puede realizar un usuario (más bien un tipo de usuario) de un Sistema de Información.

#### Elementos

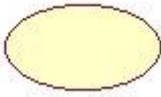
##### Actor:



Un **Actor** es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra **rol**, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino la función que realiza en la aplicación. Es una entidad que desempeña un papel.

Un actor puede ser una extensión de otro.

Como ejemplo a la definición anterior, tenemos el caso de un sistema de ventas en que el rol de Vendedor con respecto al sistema puede ser realizado por un Vendedor o bien por el Jefe del Local.

**Caso de Uso:**

Es una operación/tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.

Tiene que indicar una acción, por tanto, debe ser un verbo, nunca un sustantivo.

**Sistema****System**


El sistema se utiliza para definir el alcance del caso de uso y se dibuja como un rectángulo. Este es un elemento opcional pero útil cuando se visualizan sistemas grandes. Por ejemplo, puede crear todos los casos de uso y luego utilizar el objeto del sistema para definir el alcance que abarca su proyecto

**Paquete**

El paquete es otro elemento opcional que es extremadamente útil en diagramas complejos. De manera similar a los [diagramas de clase](#), los paquetes se utilizan para agrupar los casos de uso.

**Relaciones:**

Hay distintos tipos de relaciones:

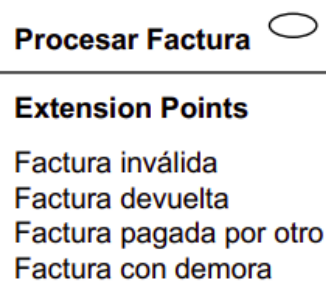
1. **Asociación**  (entre un actor y un caso de uso)

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. A veces con una línea (sin flecha)

2. **Generalización**

Este tipo de relación es uno de los más utilizados, cumple una doble función dependiendo de su estereotipo, que puede ser de **Uso** (<<uses>>) o de **Herencia** (<<extends>>).

**extends:** Se recomienda utilizar cuando un caso de uso es un caso particular de otro. Se utiliza tanto para casos de uso como para actores. Si hay varios puntos de extensión en un caso de uso se enumeran



**uses:** Se recomienda utilizar cuando se tiene un conjunto de acciones que son similares en más de un caso de uso. Sería el equivalente a una función. Se la llama también include

## Cómo crear un diagrama de caso de uso

### 1. Identificación de los actores

Los actores son entidades externas que interactúan con su sistema. Puede ser una persona, otro sistema o una organización. En un sistema bancario, el actor más obvio es el cliente. Otros actores pueden ser empleados del banco o cajeros, dependiendo del papel que intente mostrar en el caso de uso.

Un ejemplo de organización externa puede ser la autoridad tributaria o el banco central.

### 2. Identificación de los casos de uso

Una buena manera de hacerlo es identificar lo que los actores necesitan del sistema. En un sistema bancario, el cliente tendrá que abrir cuentas, depositar y retirar fondos, solicitar usuario online... Por lo tanto, todos ellos pueden considerarse como casos de uso.

Una vez que se identifican los actores y los casos de uso se tiene una idea básica del sistema.

### 3. Buscar funcionalidad común a utilizar

Busca una funcionalidad común que pueda ser reutilizada en todo el sistema. Si encuentras dos o más casos de uso que comparten una funcionalidad común, puedes extraer las funciones comunes y añadirlas a un caso de uso separado. Entonces puedes conectarlo a través de la relación de inclusión(uses) para mostrar que siempre se llama cuando se ejecuta el caso de uso original.

#### Funciones opcionales o funciones adicionales

Hay algunas funciones que se activan de manera opcional. En esos casos, se puede utilizar la relación de extensión.

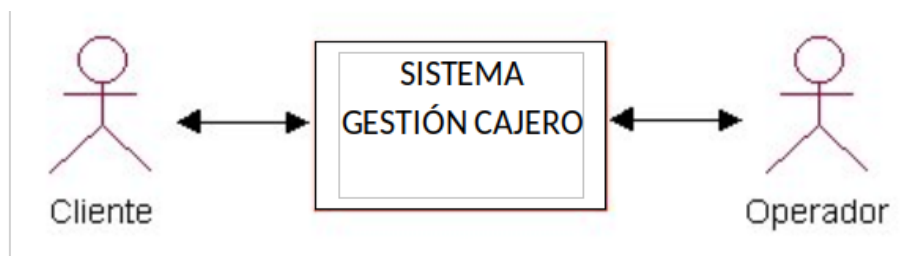
Extender no siempre significa que sea opcional. A veces el caso de uso conectado por extensión puede complementar el caso de uso base. Lo que hay que recordar es que el caso de uso básico debe ser capaz de realizar una función por sí mismo, aunque no se llame caso de uso extensivo.

#### Ejemplo:

Un cajero de un banco. Un cliente con tarjeta mediante login puede: consultar saldo, sacar dinero. También puede ingresar dinero en efectivo o ingresar un cheque sin logarme.

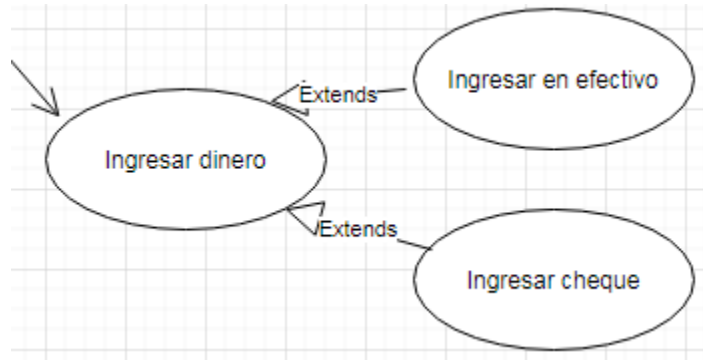
Todos los días el operador de mantenimiento retira las tarjetas que no han sido devueltas y recarga el dinero en el cajero.

Como una primera aproximación identificamos a los actores que interactúan con el sistema:

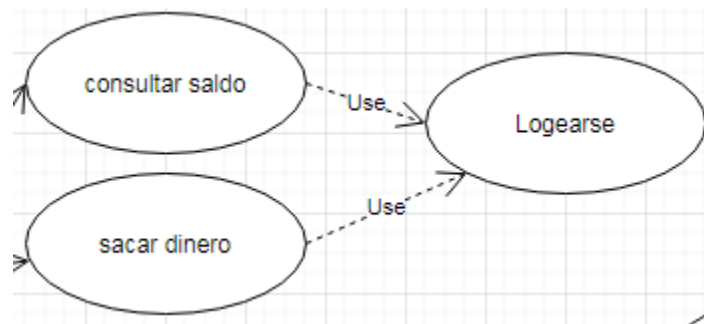


Un Cliente puede realizar unas operaciones y un Operador otras.

Ingresar dinero puede ser en efectivo o mediante un cheque.



Las operaciones de consultar el saldo y de sacar dinero llevan implícita la acción de logearse:



Entonces, el diseño completo del diagrama Use Case es:

