



Análisis de Sistemas

Casos de Uso

Ing. Ramiro Garbarini

- 2022 -

Índice

1	Objetivo.....	3
2	Casos de Uso.....	4
	2.1 Utilidad de los Casos de Uso	4
	2.2 Documentación de Casos de Uso	4
	2.2.1 Casos de Uso y UML.....	5
	2.2.2 Flujos de Eventos.....	6
	2.2.3 Precondiciones y Postcondiciones.....	8
	2.2.4 Escenario de un Caso de Uso.....	9
	2.2.5 Nivel de detalle de los Casos de Uso	10
	2.2.6.....	10
	2.3 Relaciones entre Casos de Uso	11
	2.3.1 Relación de extensión	11
	2.3.2 Relación de Inclusión.....	13
	2.4 Algo más sobre las relaciones entre casos de uso.....	14
	2.4.1 Algo más sobre la relación “Include”	14
	2.4.2 Algo más sobre la relación “Extend”	14
3	Actores	15
	3.1 ¿Cómo identificar actores?	18
	3.1.1 Breve descripción del actor	20
	3.1.2 Checkpoint: Actor	20
	3.1.3 Las relaciones entre actores	21
	3.2 Relaciones entre Actores y Casos de Uso	22
4	Diagrama de Casos de Uso	23
	4.1 Utilidad del Diagrama de Casos de Uso	23
	4.2 Actividades para la generación de un Diagrama de Casos de Uso ..	24
5	Glosario.....	27
6	Anexo 1: Plantilla Caso de Uso	29
7	Anexo 2: UML – Elementos de Casos de Uso	30
8	Anexo 3: UML – Relaciones entre elementos	31
9	Bibliografía	32

1 Objetivo

El objetivo del presente apunte busca introducir al lector principalmente en el concepto de Casos de Uso, como así también en sus limitaciones, su utilidad e importancia. Se utilizará para facilitar la comprensión, una serie de ejemplos y gráficos aclaratorios.

A su vez, se hará mención de la nomenclatura usada por el UML (Unified Modelling Lenguaje) para los casos de uso, sus participantes y sus correspondientes relaciones. Dicha nomenclatura será adoptada a lo largo de todo el documento.

2 Casos de Uso

Los casos de uso son una descripción de un conjunto de secuencias de acciones (interacciones con elementos externos al sistema), que ejecuta un sistema para obtener un resultado que agregue valor.

Un caso de uso es iniciado por un elemento externo al sistema. A partir de ese momento, el sistema intercambia datos con el entorno.

Los casos de uso especifican un comportamiento deseado (describen qué hace el sistema o subsistema), pero no imponen cómo se llevará a cabo ese comportamiento.

Permiten definir los límites del sistema y las relaciones entre el sistema y el entorno.

El nombre que posee el caso de uso, siempre está expresado desde el punto de vista del usuario y no desde el punto de vista del sistema.

Los casos de uso presentan las siguientes características:

- Están expresados desde el punto de vista del usuario (es quién interactúa con el sistema).
- Se documentan con texto informal.
- Describen tanto lo que hace el usuario como lo que hace el sistema cuando interactúa con él, pero poniendo énfasis en la interacción.
- Son iniciados por un único elemento externo.

Cabe aclarar, que el término elemento externo se asocia al Actor, concepto definido más adelante.

2.1 Utilidad de los Casos de Uso

Los casos de uso se utilizan para lo siguiente:

- Para modelar el contexto de un sistema.
- Para identificar y organizar actores.
- Proporciona un medio para capturar los requerimientos funcionales, enfocándose en el punto de vista del usuario (¿Qué hace el sistema?...por los usuarios).
- Para especificar qué debería hacer el sistema desde el punto de vista externo sin importar como se haga.
- Para documentar los requisitos de un sistema.
- Para documentar las funciones del sistema y los roles de cada actor interviniente.
- Permite que los desarrolladores y los clientes lleguen a un acuerdo sobre los requisitos del sistema (contrato).
- Permite generar la documentación de usuario y las pruebas funcionales del sistema en paralelo con el desarrollo.

2.2 Documentación de Casos de Uso

Una vez que se identifican los casos de uso, se comienzan a documentar sus pasos. Este documento se crea para cada caso de uso, detallando lo que el sistema debe proporcionar al actor cuando el caso de uso es ejecutado.

La documentación de casos de uso podría consistir en lo siguiente:

- Nombre.
- Descripción.
- Describir cómo comienza el caso de uso y cómo termina.
- Realizar un flujo normal de eventos.
- Realizar un flujo alterno de eventos.
- Detallar las excepciones al flujo de eventos.
- Precondiciones.
- Postcondiciones.

Ver [Anexo 1: Plantilla Caso de Uso](#).

Es importante que a medida que se van encontrando los casos de uso, se identifiquen y documenten los términos que forman parte del dominio del problema en un Glosario.

2.2.1 Casos de Uso y UML



¿Qué es UML?

UML (Unified Modelling Lenguaje), es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software.

UML esta relacionado con el paradigma orientado a objetos.

UML presenta un set de herramientas (diagramas y técnicas).

<http://www.uml.org/>

Los casos de uso son una técnica "de redacción" del conjunto de secuencias de acciones que ejecuta un sistema. Los casos de uso son una técnica independiente del paradigma que se adopte, es decir se podrán utilizar tanto en el paradigma estructurado como en el paradigma orientado a objeto. En base a lo antedicho, se puede concluir en que los casos de uso no son una herramienta propia del UML.

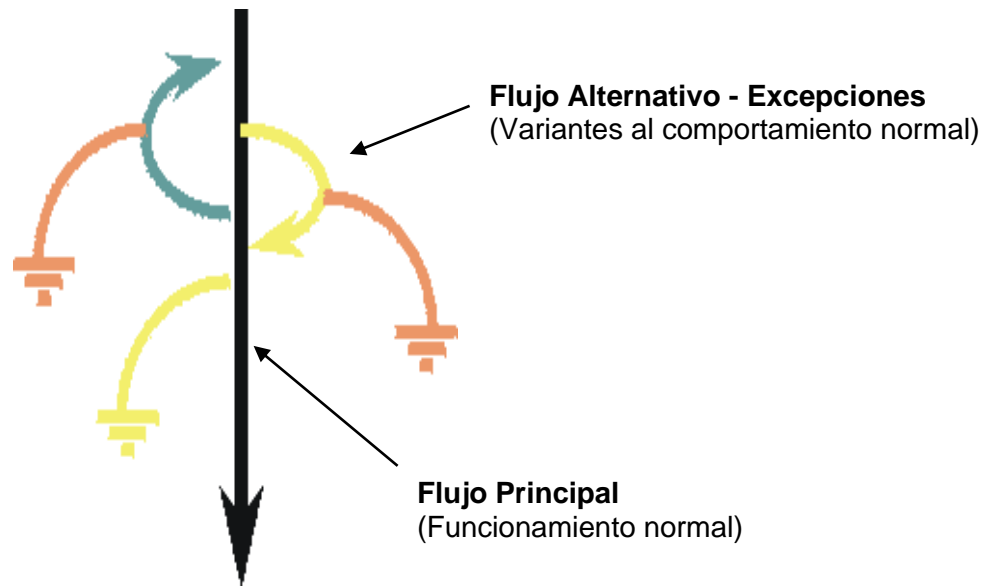
Dada la utilidad e importancia de los casos de uso, UML los incluye dentro del conjunto de herramienta de modelado, definiendo un estándar de representación gráfica, la cual será adoptada en el presente documento.

Los actores representan un conjunto de roles que los usuarios de los casos de uso juegan al interactuar con ellos. UML incluye también dentro de su conjunto de herramientas de modelado el concepto de actores y establece un estándar de representación gráfica.

Ver [Anexo 2: UML – Elementos de Casos de Uso](#)

2.2.2 Flujos de Eventos

Un caso de uso está compuesto por un flujo principal y uno o más flujos alternativos, como se muestra en la siguiente imagen.

**Flujo Principal:**

Es la ejecución del curso normal del caso de uso.

Ejemplo:

Caso de Uso: Procesar Venta

Actor: Cajero

Flujo Principal:

1. El Cliente llega a la caja con los productos que desea comprar.
2. El Cajero inicia un nuevo proceso de venta.
3. El Cajero registra el producto ingresando el código del mismo.
4. El Cajero repite el paso 3 hasta que no haya mas productos para ingresar.
5. El Sistema totaliza la operación.
6. El Cajero informa el importe.
7. El Cliente paga y el Sistema gestiona el pago.
8. El Sistema emite la factura.
9. El Cliente recibe la factura y se retira con los productos.

Flujo Alternativo:

Durante la ejecución de un caso de uso, suelen aparecer errores o excepciones al curso principal. Esas desviaciones del curso normal del caso de uso se llaman alternativas y representan un error o excepción en el curso normal del caso de uso.

Ejemplo: se detallará el flujo alternativo del paso 7 del caso de uso Procesar Venta, el sistema solo admite pago en efectivo y pago con tarjeta de crédito.

Caso de Uso: Procesar Venta

Actor: Cajero

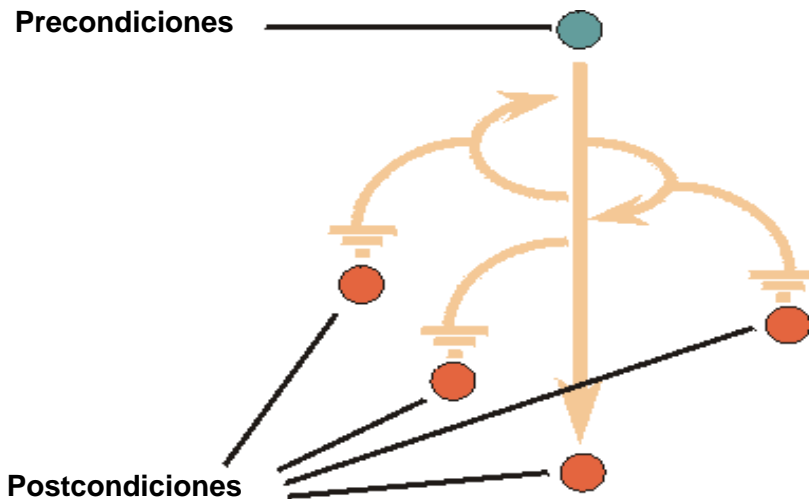
Flujo Alternativo:*7.a. Pago en Efectivo:*

1. El Cajero ingresa el monto en efectivo recibido.
2. El Sistema emite el vuelto a entregar.
3. El Cajero guarda el dinero en la caja y retira y entrega el vuelto.
4. El Sistema registra el pago en efectivo.

7.b. Pago con Tarjeta de Crédito:

1. El Cliente entrega la tarjeta de crédito
2. El Cajero ingresa los datos de la tarjeta de crédito.
3. El Sistema solicita autorización del pago a un sistema externo.
4. El Sistema recibe la aprobación del pago, y lo notifica al cajero.
 - 4.1. El sistema recibe la denegación del pago:
 - 4.1.1. El sistema indica la denegación al cajero.
 - 4.1.2. El Cajero solicita al cliente un modo de pago alternativo.
5. El Sistema registra el pago con tarjeta de crédito.
6. El Sistema emite el comprobante de pago por tarjeta de crédito.
7. El Cajero solicita al Cliente que firme el comprobante.
8. El Cliente firma el comprobante.

2.2.3 Precondiciones y Postcondiciones



2.2.3.1 Precondiciones:

Las precondiciones, reflejan el estado en el que debe estar el sistema y su entorno para que pueda comenzar la ejecución del caso de uso.

Las precondiciones establecen lo que debe cumplirse antes de comenzar un caso de uso. Las precondiciones no se prueban en el caso de uso, sino que se asumen como verdaderas, generalmente, implica que un caso de uso anterior se haya completado exitosamente. Las precondiciones indican suposiciones importantes que deben cumplirse.

Ejemplos:

- Precondición válida:
El agente de viajes debe estar autenticado en el sistema de reservas.

2.2.3.2 Postcondiciones:

Las postcondiciones, reflejan el estado en el que queda el sistema y su entorno luego de la ejecución del caso de uso.

Establecen qué debe cumplirse cuando el caso de uso se completa con éxito.

Ejemplo:

- Postcondición válida:
Se registra la venta. Se actualiza el stock. Se imprime la factura.

2.2.3.3 Ejemplos vinculados al caso de uso Procesar Venta:

Caso de Uso: *Procesar Venta*

Actor: *Cajero*

Precondición: *El cajero se identifica y autentica.*

Postcondición: *Se registra la venta. Se actualiza la contabilidad y el inventario. Se genera la factura. Se registran las autorizaciones de pago aprobadas.*

2.2.4 Escenario de un Caso de Uso

Un escenario es una instancia de un caso de uso, es decir, un camino concreto que puede tomar un caso de uso.

2.2.4.1 Características de los escenarios

- Los escenarios no contienen condiciones, ya que describen una de las posibles instancias del caso de uso.
- Todos los escenarios de un caso de uso comienzan igual, pero pueden terminar de muchas maneras diferentes.
- No se deben mostrar sólo instancias (escenarios) exitosas del caso de uso, también se deben mostrar algunas en las que falla.
- Cada uno de los escenarios definidos, van a servir como casos de prueba a la hora de realizar testing (test funcional) del sistema.
- Dentro del marco de la Metodología Orientada a Objetos, los casos de uso y por lo tanto los escenarios, son base de otros diagramas de UML. Por ejemplo: Diagrama de Actividad, Diagrama de secuencias (por cada escenario), Diagrama de Colaboración (por todo el caso de uso).

El desarrollo de un caso de uso es un conjunto de todos los escenarios posibles que puede presentar el caso de uso.

Ejemplo: teniendo en cuenta el caso de uso *Procesar Venta* de los ejemplos anteriores, se describirán tres escenarios posibles del mismo:

Escenario 1:

1. *El Cliente llega a la caja con los productos que desea comprar.*
2. *El Cajero inicia un nuevo proceso de venta.*
3. *El Cajero registra el producto ingresando el código del mismo.*
4. *El Cajero repite el paso 3 hasta que no haya mas productos para ingresar.*
5. *El Sistema totaliza la operación.*
6. *El Cajero informa el importe.*
7. *El Cliente paga en efectivo y el cajero ingresa el monto recibido.*
8. *El Sistema emite el vuelto a entregar.*
9. *El Cajero guarda el dinero en la caja y retira y entrega el vuelto.*
10. *El Sistema registra el pago en efectivo.*
11. *El Sistema emite la factura.*
12. *El Cliente recibe la factura y se retira con los productos.*

Escenario 2:

1. *El Cliente llega a la caja con los productos que desea comprar.*
2. *El Cajero inicia un nuevo proceso de venta.*
3. *El Cajero registra el producto ingresando el código del mismo.*
4. *El Cajero repite el paso 3 hasta que no haya mas productos para ingresar.*
5. *El Sistema totaliza la operación.*
6. *El Cajero informa el importe.*
7. *El Cliente paga con tarjeta de crédito.*
8. *El Cajero ingresa los datos de la tarjeta de crédito.*
9. *El Sistema solicita autorización del pago a un sistema externo.*
10. *El Sistema recibe la aprobación del pago, y lo notifica al cajero.*
11. *El Sistema registra el pago con tarjeta de crédito.*
12. *El Sistema emite el comprobante de pago por tarjeta de crédito.*
13. *El Cajero solicita al Cliente que firme el comprobante.*

Escenario 3:

14. *El Cliente llega a la caja con los productos que desea comprar.*
15. *El Cajero inicia un nuevo proceso de venta.*
16. *El Cajero registra el producto ingresando el código del mismo.*
17. *El Cajero repite el paso 3 hasta que no haya mas productos para ingresar.*
18. *El Sistema totaliza la operación.*
19. *El Cajero informa el importe.*
20. *El Cliente paga con tarjeta de crédito.*
21. *El Cajero ingresa los datos de la tarjeta de crédito.*
22. *El sistema recibe la denegación del pago.*
23. *El sistema indica la denegación al cajero.*
24. *El Cajero solicita al cliente un modo de pago alternativo.*

2.2.5 Nivel de detalle de los Casos de Uso

El nivel de detalle de un caso de uso puede variar, según el enfoque que se le desee dar. Estos niveles pueden ser:

- *Desarrollo de Alto nivel:* similar a un documento de requerimientos, dónde se detalla el dominio del sistema (breve descripción de lo que debe hacer el sistema), usuarios involucrados, requerimientos funcionales y requerimientos no funcionales.
- *Desarrollo Expandido:* incluye todos los escenarios posibles, descriptos desde el punto de vista del usuario.
- *Desarrollo Real:* muestra cómo debería comportarse internamente el sistema.

2.3 Relaciones entre Casos de Uso

2.3.1 Relación de extensión

En los sistemas se pueden encontrar casos de uso que son especializaciones de otros casos de usos, como así también, casos de usos incluidos como parte de otro y casos de usos que *extienden* el comportamiento de otros casos de uso básicos.

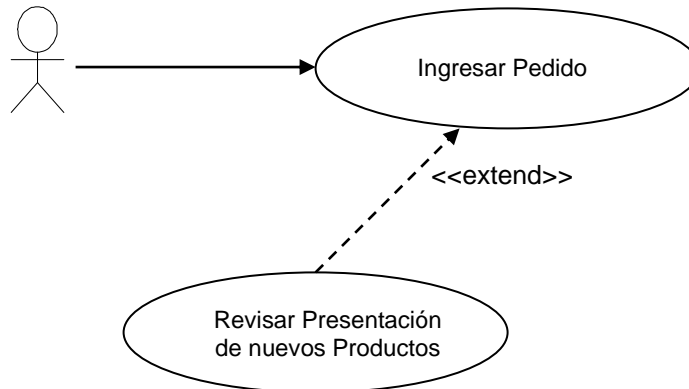
Una relación de **extensión** (extend) entre casos de uso, significa que uno de ellos incorpora implícitamente el comportamiento del otro caso de uso. Una relación de extensión se utiliza para modelar la parte de un caso de uso que el usuario puede ver como comportamiento opcional del sistema. De esta manera, se separa el comportamiento opcional del obligatorio o del de aquel que se ejecuta bajo ciertas condiciones, ya que muchas veces, la funcionalidad de un caso de uso incluye un conjunto de pasos que ocurren sólo en algunas oportunidades.

Se utiliza cuando se desea indicar que mientras se está ejecutando un caso de uso, se puede utilizar *opcionalmente* otro caso de uso, en algún punto de la ejecución del primero. Dentro de un caso de uso, algunos flujos alternativos pueden representar un comportamiento opcional seleccionado por el usuario. Cuando ese comportamiento puede ser a su vez utilizado por separado, resulta conveniente extraerlo a un nuevo caso de uso relacionado como extensión del original.

La relación de extensión se representa por una línea de trazos desde el caso que 'extiende a' al caso que es 'extendido'.

Ejemplo de Extensión:

El caso de uso “Ingresar Pedido” extiende al caso de uso “Revisar Presentación de nuevos Productos”.



Las extensiones tienen las siguientes características:

- Representan una parte de la funcionalidad del caso que no siempre ocurre.
- Son un caso de uso en sí mismas.
- No necesariamente provienen de un error o excepción.



Diferencia entre una alternativa y una extensión:

- Una extensión es un caso de uso en sí mismo, mientras que una alternativa no.
- Una alternativa es un error o excepción, mientras que una extensión puede no serlo.

2.3.2 Relación de Inclusión

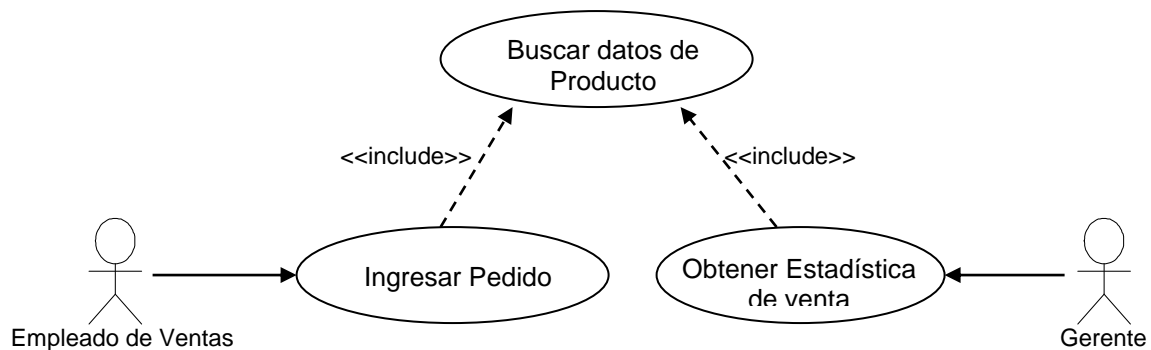
Una relación de **inclusión** (include), entre casos de uso, significa que un caso de uso incorpora el comportamiento de otro caso de uso. Es decir, el caso de uso base toma el comportamiento del caso de uso proveedor.

La relación de inclusión se usa para evitar describir el mismo flujo de eventos repetidas veces, poniendo el comportamiento común en un caso de uso aparte (que será incluido por un caso de uso base).

Se utiliza cuando se detecta que en al menos dos casos de uso diferentes conceptualmente existe alguna secuencia de pasos o flujo de eventos repetidos. De esta manera, esta secuencia puede extraerse dando lugar a un nuevo caso de uso relacionado con los otros dos mediante una relación de inclusión.

Ejemplo de Inclusión:

En el ejemplo se puede ver que el caso de uso “Ingresar Pedido” incluye el caso de uso “Buscar datos de Producto”.



Las inclusiones tienen las siguientes características:

- Aparecen como funcionalidad común, luego de haber especificado varios casos de uso.
- Los casos usados son casos de uso en sí mismos.
- El caso es usado siempre que el caso que lo usa es ejecutado. Esto marca la diferencia con las extensiones, que son opcionales.

Diferencia entre la relación extend y la include

La principal diferencia que existe entre las relaciones Include y la Extend, es que la relación *extend* cuenta con una condición a ser evaluada. En cambio, la relación *include* se define como la inclusión de un Caso de Uso en otro, pero no precisa de una condición.

2.4 Algo más sobre las relaciones entre casos de uso

Es habitual, que varios casos de uso tengan algún comportamiento en común. En vez de repetir el mismo texto en cada caso de uso, es conveniente crear un nuevo caso de uso que represente este comportamiento y se relacione con los otros casos de uso. Esto es sencillamente factorizar (sacar factor común) y de esta manera tener una funcionalidad común evitando duplicarla, que se relacionará con otros casos de uso.

Por ejemplo, tanto el caso de uso “Procesar Ventas On Line” como el caso de uso “Procesar Ventas en Locales” deben emitir facturas para entregarle al cliente. Factorizando estos casos de uso, obtenemos el caso de uso “Emitir Factura”. En consecuencia, “Procesar Ventas On Line” y “Procesar Ventas en Locales” son los casos de uso *base* que se relacionan con el caso de uso *adicional* “Emitir Factura”.

La organización de casos de uso mediante relaciones es una forma de mejorar la comunicación y comprensión, reduciendo texto duplicado y mejorando la gestión de los documentos de casos de uso.

Muchas veces, se dedica mucho tiempo debatiendo como relacionar los casos de uso, es decir, se pierde tiempo discutiendo si la relación entre dos casos de uso es una “extend” o una “include”. Hay que tener presente que lo más importante es la especificación de los casos de uso, la cual se hace escribiendo los casos de uso y no organizándolos. La organización de los casos de uso irá evolucionando y refinando a lo largo del proceso de desarrollo de un sistema.

2.4.1 Algo más sobre la relación “Include”

Ésta es la relación más común e importante.

Diversos autores recomiendan:

- utilizar “include” cuando un comportamiento se está repitiendo en dos o más casos de uso separados y se quiere evitar repeticiones.
- Utilizar “include” cuando se descompone un caso de uso largo (o complejo) en subunidades para mejorar la comprensión.
- Utilizar siempre la relación “include” entre los casos de uso.

2.4.2 Algo más sobre la relación “Extend”

Suponiendo que un caso de uso se haya establecido como estable, y su especificación no se pueda modificar, ¿Cómo hacemos para agregarle nuevas extensiones o pasos condicionales?.

En estos casos, se crea un nuevo caso de uso que extiende o añade comportamiento a un caso de uso base.

Algunas guías recomiendan utilizar “extend” para modelar el comportamiento condicional u opcional de un caso de uso base. Es decir, extender un caso de uso para ello. Sin embargo, también el comportamiento condicional u opcional se puede registrar como texto en la sección de Extensiones del caso de uso (como un curso alternativo).

Lo que motiva esencialmente el uso de extensiones es cuando no es conveniente modificar el caso de uso base.

3 Actores

Otro concepto muy relacionado con los casos de uso son los actores.

Un **Actor** representa un conjunto de roles que los usuarios de los casos de uso juegan al interactuar con ellos. Este rol o roles es llevado a cabo por una persona, un dispositivo de hardware u otro sistema.

La misma persona física puede interpretar varios papeles como actores distintos. El nombre del actor describe el papel desempeñado.

Los actores son externos al sistema. Es por ello, que al identificar a los actores se está delimitando el sistema.

Cabe aclarar la diferencia existente entre usuario y actor. Un actor es una clase de rol, mientras que un usuario es una persona que, cuando usa el sistema, asume un rol. De esta manera, un usuario puede acceder al sistema como distintos actores y un actor puede representar a varios usuarios. Varios usuarios pueden desempeñar el mismo rol y por lo tanto ser representados por el mismo actor.

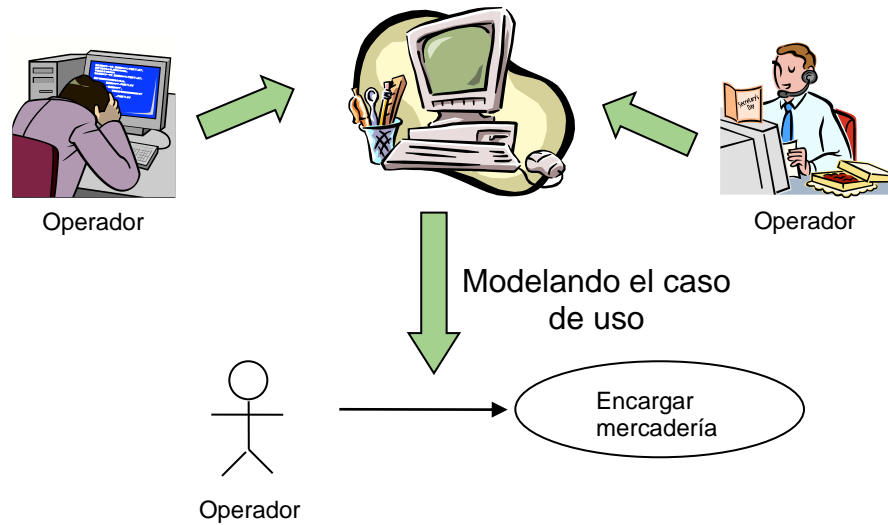
Identificar a los actores es el primer paso para usar la técnica de casos de uso.

La representación de un Actor apunta al rol que cumple el usuario, dispositivo o sistema respecto del sistema analizado.

Para entender completamente el propósito del sistema, se debe conocer para quién es el sistema, es decir, quién utilizará el sistema.

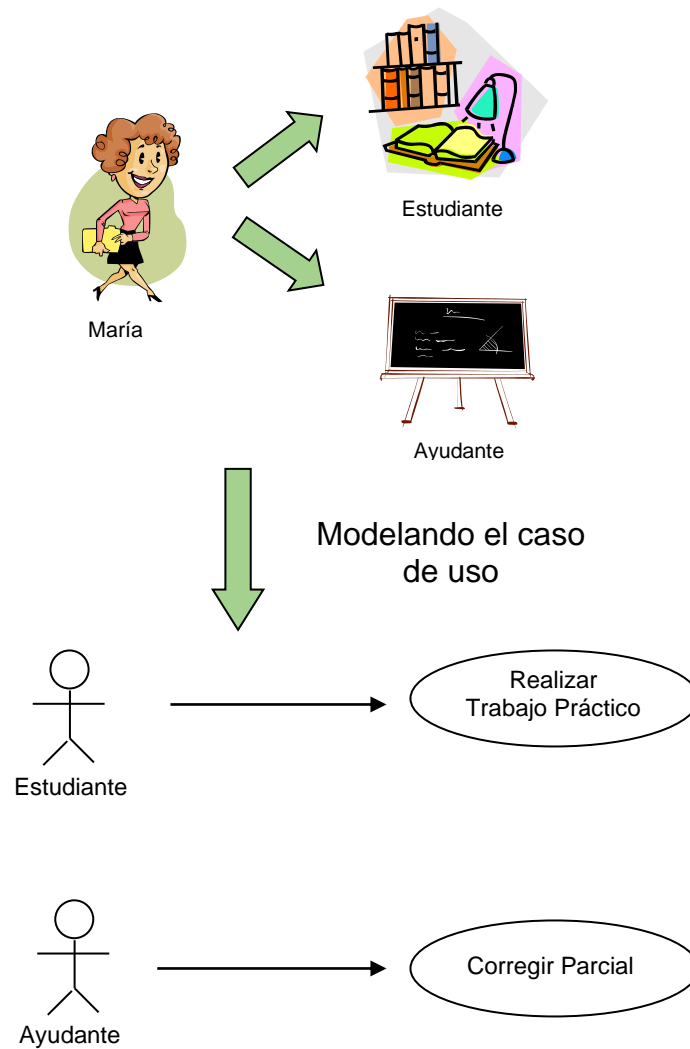
Ejemplo:

- Julián y Pablo son operadores (usuarios) del sistema de Compras, al cual ingresan semanalmente para realizar los pedidos de compras de los diferentes sectores de la compañía en la que trabajan:

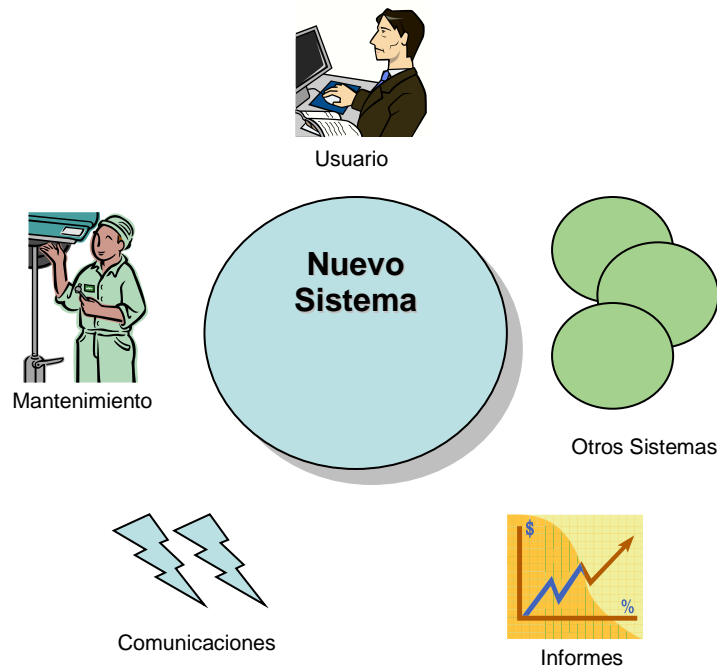


Otro caso que puede darse es el siguiente:

- María, está estudiando en la facultad y a su vez, es ayudante de una de las materias que se dicta en la misma.



3.1 ¿Cómo identificar actores?



¿Qué elementos del entorno serán actores del sistema?

Hay que empezar pensando en las personas que utilizarán el sistema. ¿Cómo se los puede categorizar?

Las siguientes preguntas son útiles cuando se busca identificar actores:

- ¿Quién proveerá, utilizará, o eliminará la información?
- ¿Quién utilizará esta funcionalidad?
- ¿Quién está interesado en cierto requerimiento?
- ¿Qué áreas de la organización utiliza el sistema?
- ¿Quién dará soporte y quién el mantenimiento del sistema?
- ¿Cuáles son los recursos externos del sistema?
- ¿con qué sistemas se deberá integrar?

Existen diversos aspectos del entorno de un sistema que se representan con actores diferentes:

- **Usuarios que ejecutan las funciones principales del sistema.**

Ejemplo:

En un sistema de mantenimiento de maquinaria, hay varias categorías de usuarios: operarios, supervisores, etc. Todas estas categorías tienen roles específicos en el sistema y se debe, por lo tanto representar cada uno como un actor separado.

- **Usuarios que ejecutan las funciones secundarias del sistema, tales como administración del sistema.**

Ejemplo:

En una máquina para reciclar latas, botellas, el Cliente es el actor principal, el sistema se construye para él. Pero alguien tiene que manejar la máquina, y este rol es representado por el operador de la máquina.

- **El sistema interactúa con Hardware externo.**

Ejemplo:

Un sistema de ventilación que controla la temperatura de un edificio, toma información de los distintos sensores distribuidos. Los sensores se representan con un Actor.

- **El sistema interactúa con otro sistema.**

Ejemplo:

Cuando el sistema de control de stock, detecta que la existencia de un artículo está bajo del nivel de reposición, envía un pedido de material al sistema de compras.

Los actores ayudan a definir el límite del sistema:

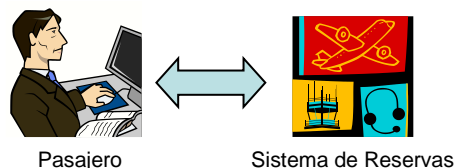
Encontrar a los actores significa establecer los límites del sistema, también ayuda a entender el propósito y el tamaño del mismo. Solamente los roles que se comunican directamente con el sistema deben ser considerados actores. Si se están incluyendo más roles que los existentes en el entorno del sistema, se está modelando “el negocio” en el cual el sistema será utilizado, no el sistema sí mismo.

Ejemplo:

¿En un sistema de reservas de pasajes, cuáles serían los actores? Esto depende si el sistema de reservas será utilizado por un agente de viaje de la empresa, o si el pasajero puede realizar las reservas vía Internet.



Si las reservas se realizan a través de un agente de viaje, el agente de viaje será el actor. El pasajero no interactúa directamente con el sistema, por lo tanto no es un actor.



Si el pasajero puede hacer la reserva directamente a través de Internet, sin necesidad de intermediarios, el pasajero es un actor.

3.1.1 Breve descripción del actor

La descripción debe incluir información acerca de:

- Qué y quién representa el actor.
- Por qué es necesario este actor.
- Qué intereses tiene el actor en el sistema.

Ejemplo:

Para la máquina de reciclaje, los actores pueden ser descriptos como:

Cliente: El cliente recoge las botellas y las da al negocio.

Operador: El operador es responsable del mantenimiento de la máquina de reciclaje.

Encargado: El encargado es responsable sobre el dinero y el servicio que se brinda.

3.1.2 Checkpoint: Actor

- ¿se han considerado y modelado todos los roles del sistema? Solo se puede estar seguro cuando se haya encontrado y escrito todos los casos de uso.
- ¿Cada actor está relacionado con algún caso del uso? Se deben quitar los actores que no tengan caso de uso asociado.
- ¿Varios actores desempeñan roles similares? Si es así, deben combinarse en un solo actor.
- ¿Dos actores desempeñan el mismo rol en lo referente a un caso del uso? Si es así se debe utilizar generalizaciones entre actores para modelar su comportamiento compartido.
- ¿Un actor usa el sistema de varias maneras (totalmente diversas) o él tiene varios propósitos (totalmente diversos) para usar el caso del uso? Si es así, probablemente se debe tener más de un actor.
- ¿Los actores tienen nombres intuitivos y descriptivos? ¿Pueden los usuarios y los clientes entender los nombres? Es importante que los nombres del actor correspondan a sus roles.

3.1.3 Las relaciones entre actores

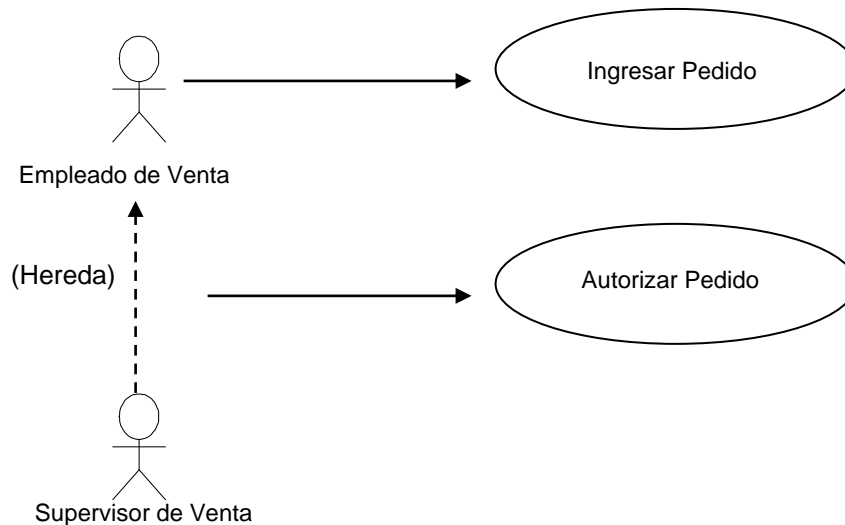
Relación de generalización (Herencia):

Se utiliza cuando se tienen actores que representan un rol muy parecido pero con algunas diferencias entre sí.

Se utiliza para indicar que un actor específico puede hacer lo mismo que otro más general, más otras cosas particulares de él.

Ejemplo de generalización:

Puede ocurrir que un actor ejecute todos los casos que ejecuta otro actor, y algunos más. De esta manera, el supervisor de ventas puede hacer todo lo que hace el empleado de ventas, pero además puede autorizar pedidos. Por lo tanto, se puede decir que el Supervisor de Ventas *hereda* al Empleado de Ventas. De esta forma, toda la funcionalidad que está habilitada para el Empleado de Ventas también lo está para el Supervisor.

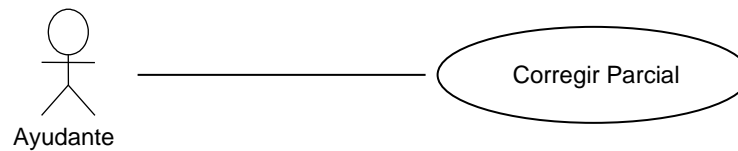


3.2 Relaciones entre Actores y Casos de Uso

Los actores pueden conectarse a los casos de uso sólo a través de *asociaciones*.

Una **asociación** indica una relación entre un actor y un caso de uso y la posibilidad que tienen éstos de comunicarse, es decir, enviar y recibir mensajes.

Ejemplo:



[Ver Anexo3: UML – Relaciones entre elementos](#)

4 Diagrama de Casos de Uso

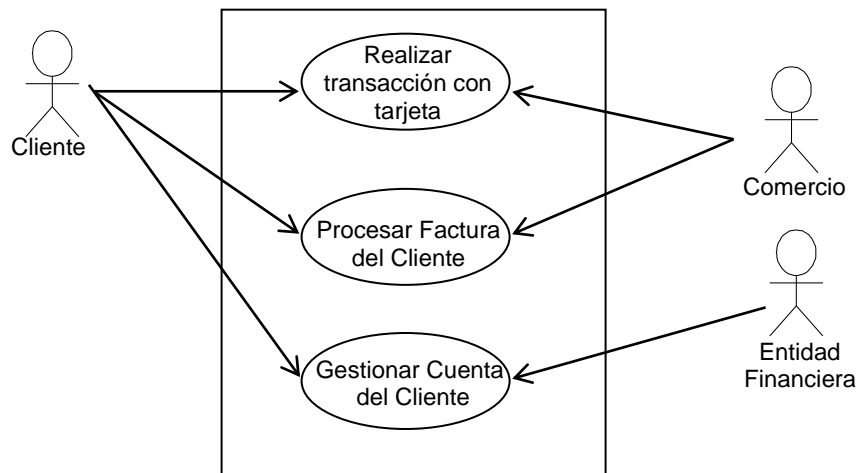
Un diagrama de casos de uso muestra un conjunto de casos de uso, actores y sus relaciones.

Se utilizan para visualizar el comportamiento de un sistema o un subsistema, de forma que los usuarios puedan comprender cómo utilizar ese elemento.

Los diagramas de casos de uso contienen:

- Casos de uso.
- Actores.
- Relaciones de asociación.

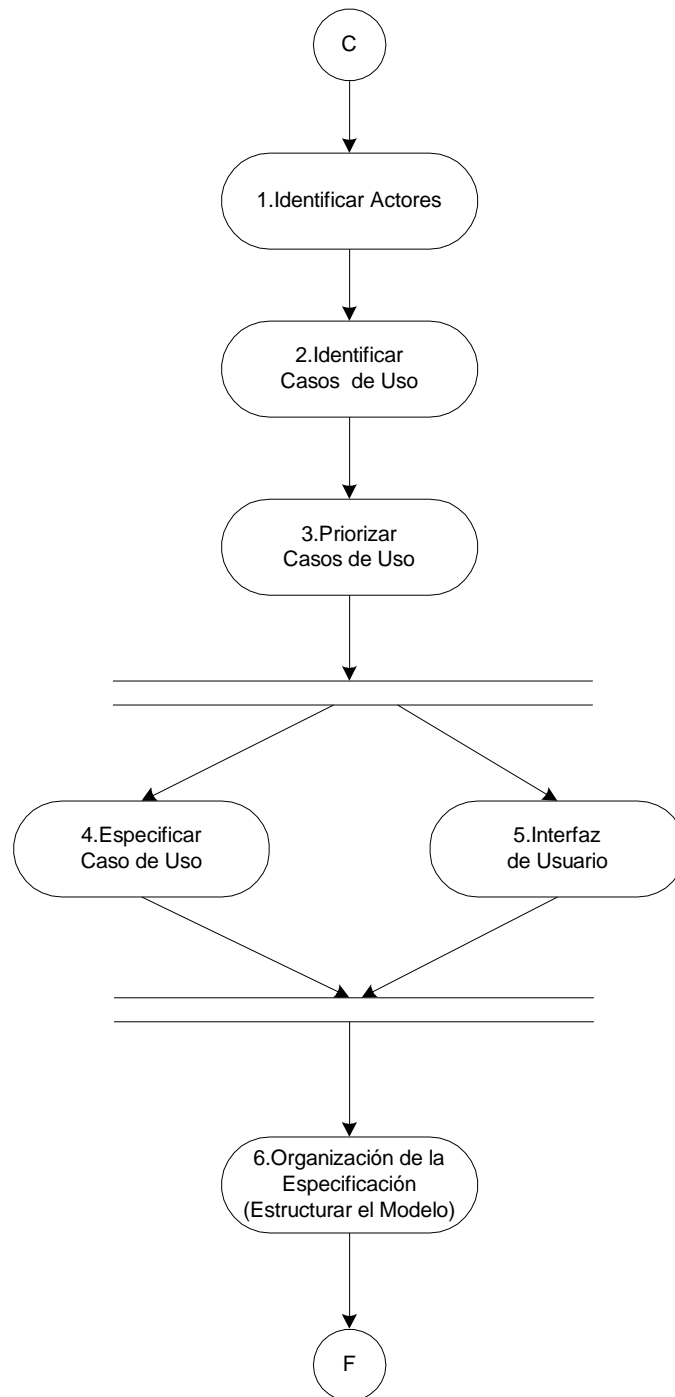
Cada relación o interacción comienza con un evento inicial que el actor envía al sistema y continúa con una serie de eventos entre el actor, el sistema y posiblemente otros actores involucrados. Muestra el comportamiento del sistema o de una parte del mismo y su interacción con los actores exteriores al sistema. Representa el aspecto estático de los casos de uso.



4.1 Utilidad del Diagrama de Casos de Uso

- Muestran los actores y los casos de uso de un sistema (o de una porción de un sistema), y las relaciones que existen entre ellos.
- Sirven como mapas donde se puede apreciar la funcionalidad del sistema completo, los actores del sistema y los casos de uso que utilizan cada uno de ellos.
- Especificaciones suplementarias, Glosario.

4.2 Actividades para la generación de un Diagrama de Casos de Uso



Descripción de las actividades:

1) Identificar Actores:

- Permite delimitar el sistema de su entorno.
- Esbozar quién va interactuar con el sistema.
- Glosario (Documento en el cual se van identificando todos los términos comunes del dominio del problema).

Encontrar los actores:

- Dos criterios para la elección.
 - Debe existir al menos un usuario que represente al actor candidato.
 - Debe existir coincidencia mínima entre los roles.

2) Identificar Casos de Uso:

- Esbozar qué funcionalidad se espera del sistema.
- Capturar y definir un glosario de términos comunes esenciales para la creación de descripciones detalladas de las funcionalidades del sistema.
- Lista de características.
- Modelo de Casos de Uso (esbozo).
- Comprender todas las especificaciones no funcionales del sistema.
- Glosario (Documento en el cual se van identificando todos los términos comunes del dominio del problema).

Encontrar los casos de uso:

- El analista de sistemas da nombre a los actores y los describe brevemente.
- Identificar los actores que representan sistemas externos y los actores para el mantenimiento y operación del sistema.
- El analista va repasando los actores y va proponiendo los casos de usos para cada actor.
- Elegir un nombre para cada caso de uso de forma que haga pensar en la secuencia de acciones concreta que añade valor a un actor. El nombre empieza generalmente con un verbo, y debe reflejar cuál es el objeto de la interacción entre el actor y el sistema.
- Recordar que un caso de uso entrega un resultado que se puede observar y que añade valor a un actor en concreto.

3) Priorizar Casos de Uso :

- El propósito de esta actividad es determinar el grado de importancia de cada caso de usos, es decir cuales son:
 - Necesarios para el desarrollo en las primeras iteraciones.
 - Más importantes para la definición de la arquitectura.
 - Más exigentes en requerimiento no funcionales.
 - Los resultados se muestran en la vista de arquitectura del modelo de casos de uso.

4) Especificar Casos de Uso:

- Describe su flujo de sucesos en detalle, incluyendo cómo comienza, termina e interactúa con los actores
- El resultado de esta actividad es la descripción detallada de un caso de uso en particular en forma de texto y diagramas.
- ¿Que incluir en la descripción?
 - Estado inicial (precondición).

- Cómo y cuándo comienza el caso de uso.
- El orden requerido en el que las acciones se deben ejecutar.
- Cómo y cuándo terminan.
- Estado finales (post condición).
- Los caminos no permitidos.
- Descripción de caminos alternativos.
- La interacción del sistema con los usuarios y que cambios producen.
- La utilización de objetos, valores y recursos
- Especificación Suplementaria (todas las especificaciones no funcionales del sistema. Requerimientos especiales” de cada caso de uso).
- Glosario.
- Describir explícitamente que hace el sistema (y separar la responsabilidad de los actores).

5) Crear interfaz de usuario:

- dibujos en papel o en herramienta de dibujo o herramientas de interfaces gráficas.
- Se utilizan como ayuda para comprender y especificar las interacciones entre los actores humanos y el sistema.

6) Organización de la Especificación (Estructurar el Modelo):

- Describir las relaciones de comunicación entre actores y casos de uso.
- Agrupar casos de uso en paquetes, cuando el número es significativo.
- Actualizar los diagramas de casos de uso.
- Generar un reporte del modelo más fácil de entender y de trabajar con él.

El modelo de casos de uso se estructura para:

- Extraer descripciones de funcionalidad generales y compartidas que pueden ser utilizadas por descripciones más específicas (generalización).
- Extraer descripciones de funcionalidades adicionales u opcionales que pueden extender descripciones más específicas (extensión).

5 Glosario

Actor: representa un conjunto de roles que los usuarios de los casos de uso juegan al interactuar con ellos

Caso de Uso: descripción de un conjunto de secuencias de acciones que ejecuta un sistema para obtener un resultado, el cual es de valor para un actor.

Diagrama de Casos de Uso: diagrama que muestra un conjunto de casos de uso, actores y sus relaciones. Se utilizan para visualizar el comportamiento de un sistema o un subsistema.

Escenario: es una instancia de un caso de uso, un camino concreto que puede tomar un caso de uso.

Evento: es un suceso que requiere una respuesta del sistema. Puede ser originado por el contexto o por condiciones de tiempo.

Flujo Alternativo: es una desviación del curso normal del caso de uso. Representa un error o excepción en el curso normal del caso de uso.

Flujo de Eventos: secuencia de pasos por los que atraviesa un caso de uso.

Flujo Principal: secuencia de pasos del curso normal del caso de uso.

Glosario: Documento en el cual se identifican los términos comunes del dominio del problema (propios del sistema).

Modelo: representación simplificada de la realidad.

Requerimientos funcionales: aquellas necesidades que especifican la función que un sistema o subsistema debe ser capaz de realizar.

Requisitos: condición o característica que debe tener o cumplir un sistema o componente de un sistema para satisfacer un contrato, norma, especificación u otro documento formalmente impuesto.

Postcondiciones: estado en el que queda el sistema y su entorno luego de la ejecución del caso de uso.

Precondiciones: estado en el que debe estar el sistema y su entorno para que pueda comenzar la ejecución del caso de uso.

Relación Extend (Extensión): relación a través de la cual un caso de uso incorpora implícitamente y bajo ciertas condiciones (opcionalmente) el comportamiento del otro caso de uso.

Relación Generalización: relación que indica que un elemento, ya sea un actor, caso de uso, clase, etc, realiza las mismas funciones que otro elemento, más otras funciones particulares.

Relación Include (Inclusión): relación a través de la cual un caso de uso incorpora el comportamiento de otro caso de uso. Se usa para evitar describir el mismo flujo de eventos repetidas veces, poniendo el comportamiento común en un caso de uso aparte.

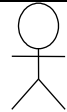



Rol: conjunto de actividades que lleva a cabo un usuario.

Usuario: persona que interactúa directamente con el sistema.

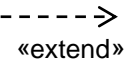
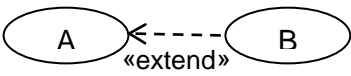
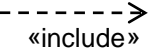
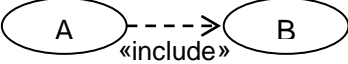


6 Anexo 1: Plantilla Caso de Uso

Caso de Uso		
Actores		
Descripción		
Precondición		
Curso Normal	Alternativo / Excepciones	
Observaciones		
Postcondición		

7 Anexo 2: UML – Elementos de Casos de Uso

Elemento	Gráfico	Descripción
Actor	 Cliente	Representación gráfica de un elemento externo que interactúa con el sistema.
Caso de Uso		Representación gráfica de Caso de Uso. Nomenclatura: Verbo en Infinitivo + Sustantivo.
Paquete		Organiza en grupo a elementos relacionados. Sirve para dividir un modelo en partes más pequeñas.
Comentario		Anotación asociada a un elemento, no tiene significado funcional sino que muestra información significativa para la comprensión del modelo.

8 Anexo 3: UML – Relaciones entre elementos

Relación	Gráfico		Descripción
Extensión			Una relación de extensión se utiliza para indicar que un caso de uso amplía el comportamiento de otro. Las extensiones se utilizan para indicar que un caso de uso (opcionalmente) amplía el comportamiento de otro.
Inclusión			Refleja que un caso de uso incluye el comportamiento de otro. Se utiliza una relación de inclusión para evitar tener el mismo subconjunto de comportamiento en muchos casos del uso.
Asociación			Implica que dos elementos tienen una relación. La asociación es el tipo general de la relación entre los elementos. Forma de enviar y recibir mensajes. Posee sentido.

9 Bibliografía

- *UML y Patrones* - Craig Larman -2da. Edición – Editorial: Pearson -Prentice Hall.
- *El lenguaje Unificado, El Proceso Unificado de Desarrollo de Software* - Booch, Jacobson y Rumbaugh – Addison Wesley.
- *Rational Unified Process*, Version 2003.06.13. - IBM Corp.
- *Guía de trabajos prácticos* – Sistemas y Organizaciones / Análisis de Sistemas UTN – FRBA – Edición 2003