

Análisis de Sistemas

**Casos de Uso**

Ing. Ramiro Garbarini

Ing. Laura Recchini

- 2013 -

**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 5

2.2.1 Casos de Uso y UML 5

2.2.2 Flujos de Eventos 6

2.2.3 Precondiciones y Postcondiciones 8

2.2.3.1 Precondiciones: 8

2.2.3.2 Poscondiciones: 8

2.2.3.3 Ejemplos vinculados al caso de uso Procesar Venta: 9

2.2.4 Escenario de un Caso de Uso 9

2.2.4.1 Características de los escenarios 9

2.2.5 Nivel de detalle de los Casos de Uso 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

3.2 Relaciones entre Actores y Casos de Uso 21

4 Diagrama de Casos de Uso 22

4.1 Utilidad del Diagrama de Casos de Uso 22

4.2 Actividades para la generación de un Diagrama de Casos de Uso 23

5 Práctica Nº 1 26

6 Glosario 32

7 Anexo 1: Plantilla Caso de Uso 34

8 Anexo 2: UML – Elementos de Casos de Uso 35

9 Anexo 3: UML – Relaciones entre elementos 36

10 Bibliografía 37

# Objetivo

El objetivo del presente documento es introducir al lector sobre el concepto de la técnica denominada Casos de Uso. Se pondrá énfasis en sus fortalezas y debilidades como técnica aplicada al análisis de sistemas, destacando además, su importancia en esta etapa del aprendizaje de la materia y su aplicación general en la carrera informática.

Su aplicación facilitará la comprensión de los aspectos vinculados al análisis de sistemas. El presente documento se encuentra complementado por ejemplos y gráficos, que intentan afianzar el conocimiento que se vaya adquiriendo sobre la técnica C.U.

A su vez, se hace mención de la nomenclatura usada por el UML (Unified Modelling Lenguaje) para definir justamente los casos de uso, sus participantes y sus correspondientes relaciones.

Dicha nomenclatura será adoptada a lo largo de todo el documento.

# 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 el comportamiento deseado de un sistema desde el punto de vista del usuario, describiendo QUÉ debe hacer el sistema o subsistema, pero no imponen sobre el CÓMO se llevará a cabo ese comportamiento. Permiten definir el sistema, sus límites y las relaciones con 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 sobre la base de lo que es representativo para el usuario, que es quién interactúa finalmente con el sistema.
* Se documentan con texto informal.
* Describen tanto lo que hace el usuario como lo que debe hacer el sistema, poniendo énfasis en la interacción entre ambos.
* Son iniciados por un único elemento externo.

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

## 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.

## 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](#Anexo1)*.*

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.

### 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, y es 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.

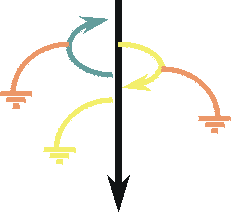
Dada la utilidad e importancia de los casos de uso, UML los incluye dentro del conjunto de herramienta de modelado, definiendo un estándar de representación gráfica, la cual será adoptada en el presente documento.

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](#Anexo2)

### 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**

(Funcionamiento normal)

**Flujo Alternativo - Excepciones**

(Variantes al comportamiento normal)

***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:***

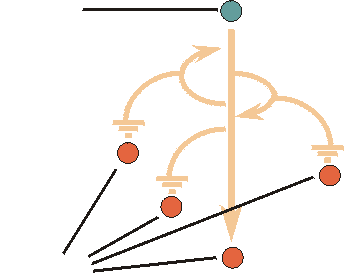
*7.a. Pago en Efectivo:*

* 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.*
   1. *El sistema recibe la denegación del pago:*
      1. *El sistema indica la denegación al cajero.*
      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.*

### Precondiciones y Postcondiciones



**Precondiciones**

**Postcondiciones**

#### 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:

*El agente de viajes debe estar autenticado en el sistema de reservas.*

#### Poscondiciones:

Las poscondiciones, 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*:

* Poscondición:

*Se registra la venta. Se actualiza el stock. Se imprime la factura.*

#### Ejemplos vinculados al caso de uso Procesar Venta:

***Caso de Uso:*** *Procesar Venta*

***Actor:*** *Cajero*

***Precondición:*** *El cajero se identifica y autentica.*

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

### 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.

#### 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*: tendiendo 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:

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 recibe la denegación del pago.*
10. *El sistema indica la denegación al cajero.*
11. *El Cajero solicita al cliente un modo de pago alternativo.*

### 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.

Algunos de 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.

## Relaciones entre Casos de Uso

### Relación de extensión

Pueden existir 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. 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 “Revisando Presentación de nuevos Productos “.

Ingresar Pedido

Revisar Presentación

de nuevos Productos

<<extend>>



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.

### Relación de Inclusión

Una relación de **inclusión** (include), entre casos de uso, significa que un caso de uso incorpora el comportamiento de otro. 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), permitiendo de esta manera la reutilización de este caso de uso.

Se aplica 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”.

Ingresar Pedido

Empleado de Ventas

Buscar datos de Producto

<<include>>

<<include>>

Obtener Estadística

de venta

Gerente



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 *includ*e se define como la inclusión de un Caso de Uso en otro, pero no precisa de una condición.

De esta manera, se puede decir que la relación *include* puede considerarse como un caso particular de la relación *extend.*

## 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 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, es decir, su definición y escritura, y no su organización. Esta organización irá evolucionando y refinando a lo largo del proceso de desarrollo de un sistema.

### 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.

### 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.

# 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 uno o varios roles. De esta manera, un usuario puede acceder al sistema como distintos actores y un actor puede representar a varios usuarios. Varios usuarios puede 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 en análisis.

Para entender completamente el propósito del sistema, se debe conocer quién es el destinatario del 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:



Encargar mercadería

Operador

Modelando el caso de uso



Operador

Operador

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.

Modelando el caso de uso



Estudiante

Ayudante

María

Estudiante

Ayudante

Realizar

Trabajo Práctico

Corregir Parcial

## ¿Cómo identificar actores?

Comunicaciones



Usuario



Informes



Mantenimiento

Otros Sistemas

¿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.



Pasajera

Agente de viaje



Sistema de Reservas

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.

Pasajero



Sistema de Reservas



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

### 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.

### 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.

## 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:*

Corregir Parcial

Ayudante

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

# 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.

Cliente

Comercio

Entidad Financiera

Realizar transacción con tarjeta

Procesar Factura del Cliente

Gestionar Cuenta

del Cliente



## 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: ver Glosario.

## 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.

1. 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.

1. 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.

1. 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).

1. 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.

1. 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 especificas (generalización).
  + Extraer descripciones de funcionalidades adicionales u opcionales que pueden extender descripciones más específicas (extensión).

# Práctica Nº 1

**Enunciado Nº 1:**

Teniendo en cuenta el siguiente enunciado:

* + 1. Identificar actores y casos de uso.
    2. Redactar un caso de uso.

La empresa Sweety se dedica a la venta por mayor de golosinas y artículos para kioscos, utilizando el siguiente procedimiento para vender sus productos.

Cada corredor, existen cinco en la empresa, visita a los clientes que tiene asignados, les ofrece los nuevos productos y les informa sobre las ofertas especiales del mes. Cuando el cliente decide comprar, confecciona la Nota de Pedido, asentando todos los datos del cliente, sus datos, el número y nombre de cada artículo y la cantidad pedida. Entrega el original al cliente y el duplicado lo lleva al sector Ventas al finalizar su recorrido.

Ventas recibe las Notas de Pedido y las envía a Cuentas Corrientes para su aprobación.

Cuentas Corrientes, controla que el monto del pedido más la deuda actual no superen el límite de crédito. Si no lo supera, sella Aprobado y devuelve la Nota de Pedido a Ventas. Caso contrario, envía la Nota de Pedido a Créditos para que analice la posibilidad de aumentar el límite de crédito. Para ello, Créditos, analiza el legajo del cliente, analiza su cumplimiento con la cuenta corriente, teniendo en cuenta el monto de las compras efectuadas en los últimos seis meses y las fechas de pago. En caso de aceptar, modifica el límite de crédito, lo actualiza en el legajo, informa el nuevo límite a Cuentas Corrientes para futuros controles y envía la Nota de Pedido a ventas.

Al final del día, Ventas, controla por cada pedido si existe stock suficiente, actualiza el stock y confecciona el remito y la factura, ambos por triplicado. Los remitos los envía a Almacenes junto con el original de la factura, el duplicado de la factura lo manda a Cuentas Corrientes y archiva el triplicado junto con la Nota de Pedido.

Almacenes, prepara la mercadería en base al remito recibido, se guarda el triplicado y entrega al camionero el paquete junto con los remitos, para que sean entregados al cliente.

El camionero, devuelve a Almacenes el duplicado del remito firmado por el cliente como aceptación de la entrega.

A fin de mes, Ventas liquida las comisiones a los corredores, según el total de ventas que realizó y la comisión fijada por el Gerente de Ventas. Entrega el duplicado de la liquidación a Tesorería para que proceda al pago. El original, se lo da al corredor y guarda el triplicado.

**Resolución:**



Corredor

Cuenta Corriente

Crédito

Ventas

Almacén

Confeccionar

Nota de Pedido

Aprobar

Nota de Pedido

Evaluar Límite

de Crédito

Procesar

Venta

Liquidar

Comisiones

Preparar

Mercaderías

?

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de Uso** | | Procesar Venta | |
| **Actores** | | Ventas | |
| **Descripción** | | Se procesan las ventas, confeccionando la factura y remito correspondiente. Además, se actualiza el stock. | |
| **Precondición** | | Fin del día. | |
| **Flujo Principal** | | | **Flujo Alternativo / Excepciones** |
| 1 | Ventas solicita procesar los pedidos aprobados del día. | |  |
| 2 | El sistema para cada pedido controla el stock. | | 2.a - No hay stock suficiente (falta relevar). |
| 3 | El sistema actualiza el stock. | |  |
| 4 | El sistema emite el remito. | |  |
| 5 | El sistema emite la factura. | |  |
| 6 | Ventas envía los remitos a Almacén. | |  |
| 7 | Ventas envía los originales de las facturas a Almacén. | |  |
| 8 | Ventas envía los duplicados de las facturas a Cuenta Corriente. | |  |
| 9 | Ventas archiva los triplicados de las facturas. | |  |
|  |  | |  |
| **Observaciones** | |  | |
| **Postcondición** | | Se actualiza stock. Se emite Remito. Se emite Factura. | |

Nota:Este caso en particular, estamos trabajando sobre un sistema de información, por lo tanto, al momento de la especificación de casos de uso se tiene en cuenta como es el manejo de toda la información, independientemente si se interactúa con un sistema software o no.

En caso de tratarse de un sistema software, no se deberán tener en cuenta los pasos 6 a 9.

**Enunciado Nº 2:**

Teniendo en cuenta el siguiente enunciado:

1. Identificar actores y casos de uso.
2. Redactar un caso de uso.

El área de Recepción del Hotel Colonia tiene a su cargo, entre otras cosas, las tareas de: efectuar reservas, asignar habitaciones y facturar estadías. Muchos huéspedes hacen sus reservas con bastante anticipación. Para ello, se comunican telefónicamente o se dirigen al hotel, en donde Recepción les informa sobre los distintos tipos de habitaciones y servicios que brinda el hotel en base a los catálogos confeccionados por el sector de Marketing, al igual que, las distintas tarifas que trimestralmente prepara la gerencia, teniendo en cuenta, los precios de la competencia y de los proveedores.

Si existen habitaciones disponibles, lo cual se chequea entre las reservas realizadas y el listado de habitaciones libres, se registra la reserva, informándole al futuro huésped, el número de habitación, el horario de entrada y el número de reserva asignado.

En ocasiones llegan personas sin aviso ni solicitud previa de alojamiento. En estos casos, el hotel sólo puede aceptar más huéspedes en función del número de habitaciones libres y las reservas confirmadas. Si la persona tenía una reserva, se le asigna una habitación, borrando la reserva del registro. Conforme se asignan habitaciones, se las borran y se registran en el formulario general de habitaciones ocupadas.

El hotel mantiene la reserva de habitaciones a sus clientes, aún cuando estos no lleguen en la fecha prevista. Cuando por error se asigna un huésped a una habitación reservada, el hotel se compromete a conseguirle alojamiento en un hotel próximo, pagando todos los gastos resultantes del traslado. En el momento en que el cliente se presenta y se percibe el error, se consulta telefónicamente, al los hoteles cercanos, hasta que se obtenga una habitación. Se le informa al cliente y se contrata un remís para el traslado.

Cuando el pasajero se retira del hotel, Recepción prepara una factura con los gastos de cada huésped, considerando el valor de estadía, según el registro de habitaciones ocupadas, la lista de precios de habitaciones y la declaración de gastos que ha ido presentando diariamente el Servicio de Habitación. Siempre que se retira un huésped del hotel, se deben actualizar los registros de habitaciones ocupadas y disponibles.



Realizar

Reserva

Brindar

Información

Asignar

Habitación

Facturar

Servicios

Recepción

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de Uso** | | Asignar Habitación. | |
| **Actores** | | Recepción. | |
| **Descripción** | | Asignar una habitación al cliente que llega. El cliente puede o no haber efectuado previamente la reserva. | |
| **Precondición** | |  | |
| **Flujo Principal** | | | **Flujo Alternativo / Excepciones** |
| 1 | Cliente llega a la Recepción y solicita habitación. | |  |
| 2 | Recepción consulta al cliente si posee reserva. | | *2.a - Posee reserva.* |
|  |  | | 1. Recepción consulta si está la habitación disponible. |
|  |  | | 1.1 Si está disponible, Recepción actualiza el registro de reserva y la asigna. |
|  |  | | 1.2 Si No está disponible, Recepción busca habitación en hotel cercano y consigue remís. |
|  |  | | *2.b - No posee reserva* |
|  |  | | 1. Recepción verifica si hay habitaciones disponibles. |
|  |  | | 1.1. Si hay habitación la asigna. |
|  |  | | 1.2. Si no hay habitación se informa al cliente. |
| 3 | Recepción asigna un número de habitación al cliente. | |  |
| 4 | Recepción actualiza el registro de habitaciones ocupadas. | |  |
| **Observaciones** | |  | |
| **Postcondición** | | Se asigna habitación. Se actualizan registro de reservas y de habitaciones ocupadas. | |

# Glosario

***Actor:*** representa un rol o 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***: es una representación gráfica a modo de diagrama, que muestra un conjunto de casos de uso, actores y sus relaciones. Se utiliza 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.

***Poscondiciones***: 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 condicione (opcionalmente) el comportamiento del otro caso de uso.

***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.

# Anexo 1: Plantilla Caso de Uso



# 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. |

# Anexo 3: UML – Relaciones entre elementos

| Relación | Gráfico |  | *Descripción* |
| --- | --- | --- | --- |
| Extensión | «extend» | «extend» | 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 | «include» | «include» | 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. |
| Generalización |  | Empleado  Cajero | Utilizado para indicar herencia. El origen (cajero) hereda las características del destino (empleado) de la relación.Indica que un elemento, ya sea un actor, caso de uso, clase, etc, realiza las mismas funciones que otro elemento, más otras funciones particulares. |
| 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. |

# 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