

# ANÁLISIS: CASOS DE Uso

# INDICE

1	INTRODUCCIÓN .....	4
2	¿ QUÉ SON LOS CASOS DE USO ? .....	4
3	SINTAXIS UML PARA LA ESPECIFICACIÓN DE ACTORES, CASOS DE USO Y SUS RELACIONES 8	
3.1.1	Sintaxis UML básica: Actores y Casos de Uso .....	8
3.1.1	Sintaxis UML avanzada: Generalización de Actores y Casos de Uso .....	11
3.1.2	Sintaxis UML básica: Relaciones <<communicates>> .....	12
3.1.3	Sintaxis UML avanzada: Relaciones <<include>> y <<extend>> .....	12
4	ESPECIFICACIÓN TEXTUAL DE CASOS DE USO .....	17
4.1	ESPECIFICACIÓN TEXTUAL INFORMAL DE CASOS DE USO .....	17
4.2	ESPECIFICACIÓN TEXTUAL DETALLADA DE CASOS DE USO .....	17
5	CASOS DE USO ESENCIALES Y REALES O CONCRETOS .....	30
6	GRANULARIDAD DE LOS CASOS DE USO .....	30
7	IDENTIFICACIÓN DE ACTORES Y CASOS DE USO .....	33
8	VENTAJAS Y LIMITACIONES DE LOS CASOS DE USO .....	35
8.1	VENTAJAS .....	35
8.2	LIMITACIONES .....	35
9	BIBLIOGRAFÍA .....	36
10	ANEXO 1: PLANTILLA EN LENGUA INGLESA PARA LA ESPECIFICACIÓN COMPLETA DE UN CASO DE USO .....	37
10.1	GUIDANCE FOR USE CASE TEMPLATE .....	37
10.2	USE CASE IDENTIFICATION .....	37
10.2.1	Use Case ID .....	37
10.2.2	Use Case Name .....	37
10.3	USE CASE HISTORY .....	37
10.3.1	Created By .....	37
10.3.2	Date Created .....	37
10.3.3	Last Updated By .....	38
10.3.4	Date Last Updated .....	38
10.4	USE CASE DEFINITION .....	38
10.4.1	Actors .....	38
10.4.2	Trigger .....	38
10.4.3	Description .....	38
10.4.4	Preconditions .....	38
10.4.5	Postconditions .....	38
10.4.6	Normal Flow .....	39
10.4.7	Alternative Flows .....	39
10.4.8	Exceptions .....	39
10.4.9	Includes .....	39
10.4.10	Priority .....	40
10.4.11	Frequency of Use .....	40
10.4.12	Business Rules .....	40

10.4.13	Special Requirements .....	40
10.4.14	Assumptions.....	40
10.4.15	Notes and Issues .....	40
10.5	USE CASE LIST .....	40
10.6	USE CASE TEMPLATE.....	41
10.7	REVISION HISTORY .....	41

# 1 INTRODUCCIÓN

El presente documento aborda la especificación de Casos de Uso<sup>1</sup>, tanto desde un punto de vista descriptivo, como la notación propuesta por el Lenguaje de Modelado Unificado necesaria para su representación gráfica mediante Diagramas de Casos de Uso. Además, se proporcionan pautas para la identificación de Actores y Casos de Uso y, en el caso de estos últimos, para que los Casos de Uso identificados tengan además un grado de granularidad adecuado.

## 2 ¿QUÉ SON LOS CASOS DE USO ?

En Ingeniería de Software, los Casos de Uso son una herramienta vinculada fundamentalmente a la Actividad de Análisis del Proceso Unificado de Desarrollo. Además, **los Casos de Uso son una técnica que permite la Captura de los Requisitos<sup>2</sup> potenciales de un nuevo sistema, o de cualquier sistema en el caso de que lo que se pretenda llevar a cabo sea una actualización de este, ya que los Casos de Uso son una técnica que permite capturar información bien acerca de cómo un sistema trabaja en la actualidad, o bien acerca de cómo se desea que trabaje.**

**Los Casos de Uso permiten capturar la funcionalidad o Requisitos Funcionales (RF)<sup>3</sup> del sistema, así un Caso de Uso describe el comportamiento del sistema al afrontar un requisito funcional que enfatiza el valor proporcionado por el sistema a sus Actores que son entidades externas tales como operadores humanos u otros sistemas. Por tanto, los Casos de Uso permiten expresar los requisitos funcionales que indican qué hará el sistema, aunque también es posible especificar a través de los Casos de Uso, los Requisitos no Funcionales (RNF)<sup>4</sup>, en la sección de Requisitos Especiales tal y como se explicará más adelante, siempre que estos requisitos no funcionales se encuentren fuertemente relacionados con el Caso de Uso en cuestión. Los requisitos no funcionales excluidos de la descripción de los Casos de Uso aparecerán reflejados en un Catálogo de Requisitos o en un Documento de Especificación de Requisitos<sup>5</sup>.**

Normalmente, en la especificación de los Casos de Usos, al igual que en la de los Requisitos, debe evitarse el empleo de una jerga técnica, prefiriéndose en su lugar un lenguaje cercano al del usuario. Además, en la especificación de los Casos de Uso es importante la colaboración de los potenciales usuarios o clientes del sistema con los ingenieros de software.

Los Casos de Uso especifican **interacciones típicas entre los Actores<sup>6</sup> y el Sistema** que captan funciones visibles para los Actores y logran un resultado observable útil para alguno de ellos. Los Casos de Uso especifican lo que hace el sistema desde el punto de vista de sus Actores, es decir, describen un

<sup>1</sup> En los textos ingleses los Casos de Uso se denominan `Use Cases`.

<sup>2</sup> En los textos ingleses la Captura de Requisitos se denomina `Requirements Elicitation`.

<sup>3</sup> En los textos ingleses los Requisitos Funcionales se denominan `Functional Requisites o Requirements (FRQ)`. Un Requisito Funcional es una característica requerida del sistema que expresa una capacidad de acción del mismo, es decir, una funcionalidad, generalmente expresada en una declaración en forma verbal. Véase más información sobre el concepto de Requisito Funcional en el documento dedicado a este tema de forma específica.

<sup>4</sup> En los textos ingleses los Requisitos no Funcionales se denominan `Non Functional Requisites o Requirements (NFR)`. Un Requisito no Funcional es una característica ya sea del sistema, del proyecto o del servicio prestado, que es requerida por el cliente junto con la especificación del propio sistema pero que, en principio, no se satisface añadiendo código, sino cumpliendo con ésta como si de una restricción se tratara. Véase más información sobre el concepto de Requisito no Funcional en el documento dedicado a este tema de forma específica.

<sup>5</sup> Véase más información en el documento dedicado a este tema de forma específica.

<sup>6</sup> El concepto de Actor se analizará con detalle más adelante en la presente documentación.

uso del sistema o cómo los Actores interactúan con el sistema para satisfacer un objetivo. Además, se trata de una técnica que pone énfasis en la relación del sistema con el exterior.

Se podría decir que **los Casos de Uso cuentan historias** puesto que cada Caso de Uso contiene una colección de escenarios de éxito y fallo relacionados, que indican cómo debería interactuar el sistema con sus Actores para conseguir un objetivo específico, así como lo que puede ocurrir si dicho objetivo específico no se logra durante la interacción del Actor con el Sistema.

Además, **conviene tener en cuenta que se utiliza una aproximación de caja negra** puesto que no se describe el funcionamiento en detalle del sistema, es decir, no se describe el funcionamiento interno del sistema, sus componentes o diseño, sino que se describe el sistema en base a las responsabilidades que tiene. Así, los Casos de Uso contienen una descripción textual de todas las maneras en que los Actores previstos podrían trabajar sobre el sistema, pero no describen ninguna funcionalidad interna, es decir oculta al exterior, del sistema, ni explican cómo se implementará dicha funcionalidad. Simplemente muestran las tareas que cada Actor puede realizar sobre el sistema.

Mediante la definición de las responsabilidades del sistema con Casos de Uso de caja negra, es posible especificar qué debe hacer el sistema (requisitos funcionales) sin decidir cómo lo hará (diseño). Es decir, que los Casos de Uso no son parte de la tarea de Diseño (cómo), sino parte de la tarea de Análisis (qué).

Otro aspecto importante es que no se trata de una técnica orientada a objetos, es decir, que no se está realizando un análisis orientado a objetos si se especifican Casos de Uso. Esto no es una desventaja sino una aclaración. Sí es cierto que durante la década de los 90' los Casos de Uso se convirtieron en una de las prácticas más comunes para la captura de requisitos funcionales de forma paralela al desarrollo del paradigma de la Programación Orientada a Objetos (POO) en el marco del cual se originaron, no obstante, pueden utilizarse con resultados igualmente satisfactorios cuando se empleen otros paradigmas de programación<sup>7</sup>. Es decir, que los Casos de Uso son una técnica para la captura de requisitos funcionales que sirve igual para, por ejemplo, un enfoque procedimental que para una filosofía orientada a objetos.

Si el lector se ha enfrentado alguna vez al Lenguaje de Modelado Unificado, es posible que haya visto algún Diagrama de Clases y que espere que los Casos de Uso sean también una forma visual de representar la información. Sin embargo, si bien los Casos de Uso se pueden representar de forma gráfica agrupados mediante **un Diagrama de Casos de Uso, dicho diagrama debe complementarse con un documento que describa cada Caso de Uso de manera que se explique la forma de interactuar entre el sistema y sus Actores.**

No obstante, cuando se tiene un número considerable de Casos de Uso surge la necesidad de contar con una visión general, en cuyo caso los Diagramas de Casos de Uso [ver Figura 1, Figura 2 y Figura 3] resultan de mucha utilidad. En estos diagramas aparecen los Actores y los Casos de Uso y las relaciones entre ellos, consiguiéndose una visión general de cómo los diferentes Actores interactúan con el sistema a través de los distintos Casos de Uso, es decir, de cómo los Actores utilizan el sistema. En estos diagramas, típicamente aparecerá un rectángulo que representará la frontera del sistema, de forma que los Actores y los Casos de Uso quedarán separados por dicha frontera: los Casos de Uso se mostrarán como parte del sistema que está siendo modelado, mientras que los Actores se mostrarán en el exterior del sistema, ya que no forman parte del mismo, pero ambos, Casos de Uso y Actores, estarán unidos por líneas que indiquen uso o participación, especificándose así las relaciones existentes.

---

<sup>7</sup> Un paradigma de programación proporciona y determina la visión y métodos de un programador en la construcción de un programa. Diferentes paradigmas resultan en diferentes formas de pensar la solución de problemas y, con ello, en diferentes estilos de programación. Véase más información sobre el concepto de paradigma de programación en el documento dedicado a este tema de forma específica.

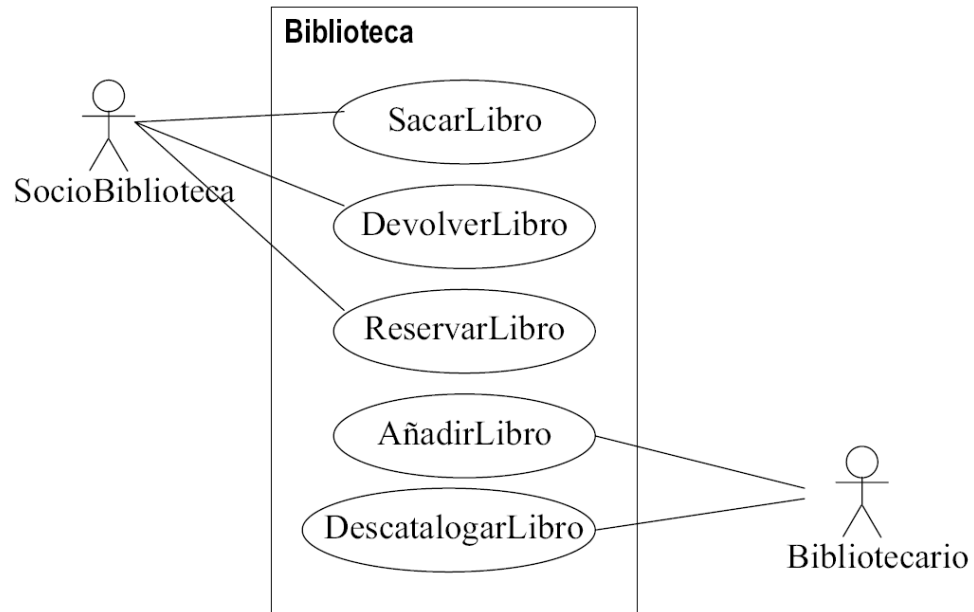


Figura 1: Ejemplo de Diagrama de Casos de Uso (I).

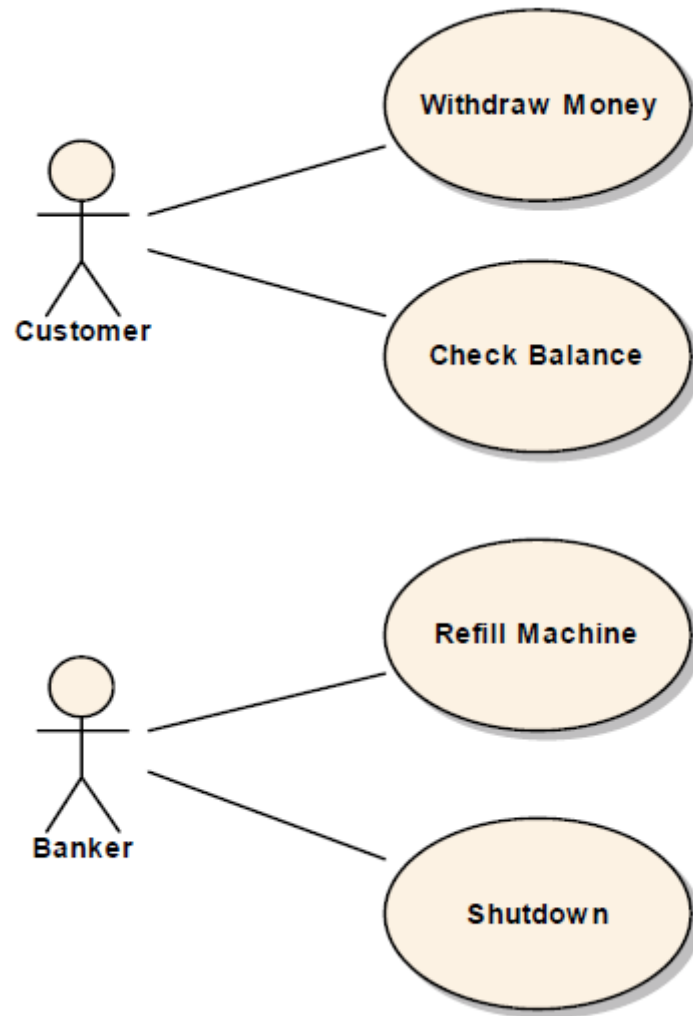


Figura 2: Ejemplo de Diagrama de Casos de Uso (II).

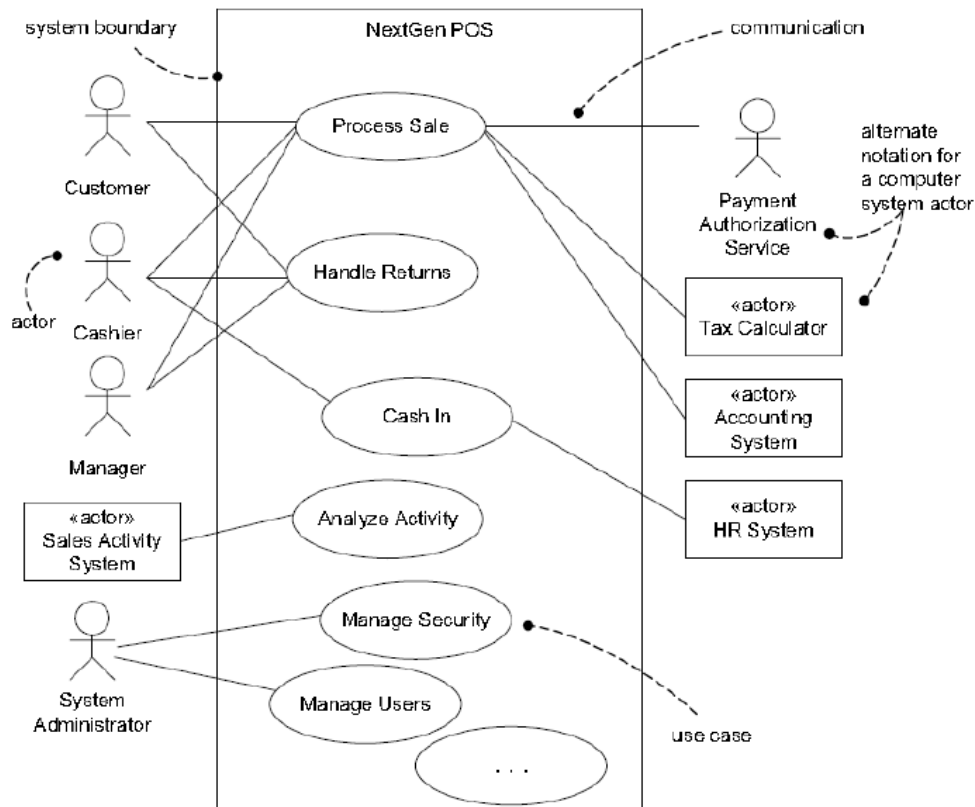


Figura 3: Ejemplo de Diagrama de Casos de Uso (III).

Los Diagramas de Casos de Uso permiten ilustrar de forma visual los requisitos funcionales del sistema. Así, con un buen Diagrama de Casos de Uso se puede conseguir saber quiénes usan el sistema (roles de personas, máquinas, otros sistemas, etc.), así como saber qué quieren del sistema, lo que permite hacer sistemas pensando en los clientes o usuarios.

Un Diagrama de Casos de Uso conforma un buen diagrama de contexto, ya que muestra los límites de un sistema, lo que permanece fuera de él, lo que forma parte de él y cómo se utiliza, por lo que sirve como herramienta de comunicación que resume el comportamiento de un sistema y las interacciones de sus Actores con el mismo. Sin embargo, únicamente proporciona una panorámica general simple de los Casos de Uso de un sistema. Y será el conjunto de Casos de Uso coherente y bien especificado lo que proporcione una imagen más completa del comportamiento del sistema a la par que permita un entendimiento común entre los clientes o usuarios y el equipo de desarrollo formado por los ingenieros de software.

### 3 SINTAXIS UML PARA LA ESPECIFICACIÓN DE ACTORES, CASOS DE USO Y SUS RELACIONES

#### 3.1.1 Sintaxis UML básica: Actores y Casos de Uso

El Lenguaje de Modelado Unificado no define, como cabe esperar, un estándar para describir de forma escrita los Casos de Uso, sin embargo, sí propone una notación gráfica para la representación de



los Diagramas de Casos de Uso. En concreto, en la representación gráfica de los Diagramas de Casos de Uso intervendrán los siguientes elementos:

**Actor:** Un Actor es una entidad externa al sistema que guarda una relación con éste y que, típicamente, le demanda una funcionalidad. Esto incluye a los operadores humanos, pero también a sistemas externos, máquinas, objetos mecánicos, organizaciones, etc., o incluso a algún evento temporal<sup>8</sup> como, por ejemplo, el final del mes o el último día trabajado, lo cual es frecuente en los Sistemas en Tiempo Real (STR)<sup>9</sup>, ya que un Actor puede ser cualquiera que desencadene un Caso de Uso. En el caso de los seres humanos se pueden ver a los Actores como definiciones de rol, por lo que un mismo individuo puede vincularse a uno o más Actores. Ejemplos de Actores serían Bibliotecario, Profesor, Alumno, etc.

Aunque no es una formalidad del Lenguaje de Modelado Unificado, en algunos textos se distinguen tres tipos de Actores con relación al sistema bajo estudio:

- **Actor principal o primario:** Tiene objetivos de usuario que se satisfacen mediante el uso de los servicios del sistema, es decir, que este tipo de Actores utilizan el sistema como ayuda. Por ejemplo, el Cajero en un sistema de ventas.
- **Actor de apoyo o secundario:** Proporciona un servicio, por ejemplo, información, al sistema bajo estudio. Normalmente se trata de un sistema, pero podría ser una organización o una persona. Por ejemplo, el Servicio de Autorización de Pago en un sistema de ventas.
- **Actor pasivo:** Está interesado en el comportamiento del Caso de Uso, pero no es principal ni de apoyo. En ocasiones, se identifican para asegurar que todos los intereses se han identificado y satisfecho, ya que es fácil no tener en cuenta los intereses de este tipo de Actores a menos que se hayan identificado explícitamente. Por ejemplo, el Ministerio de Hacienda en un sistema de ventas.

En la Figura 4 se observa la representación gráfica de un Actor según la notación del Lenguaje de Modelado Unificado que, tal y como puede observarse, se realiza mediante un monigote acompañado del nombre del Actor, aunque también puede utilizarse, de forma alternativa, la representación que aparece en la Figura 5, reservándose en ocasiones esta segunda notación para los Actores que no sean operadores humanos, sino que sean otros sistemas, máquinas, etc., de forma que visualmente se pueda identificar de forma rápida qué Actores son humanos y cuáles no. Nótese también que en la representación de los Actores hay quienes prefieren adoptar el convenio de presentar los Actores principales a la izquierda y el resto de los Actores (de apoyo, etc.) a la derecha. No obstante, debe tenerse en cuenta que ambas reglas son únicamente una práctica común y no una imposición o formalidad del Lenguaje de Modelado Unificado.

<sup>8</sup> Algunos autores definen Actor como una *entidad que obtiene algún beneficio del Caso de Uso*, en cuyo caso no podrían considerarse como Actores por ejemplo los eventos temporales. No obstante, la anterior no es una exigencia impuesta por la especificación del Lenguaje de Modelado Unificado, sino que se trata más bien de una regla práctica general (en inglés *rule of thumb*) que facilita su identificación en un caso real.

<sup>9</sup> Al utilizar el término Sistemas en Tiempo Real es importante aclarar el significado de la palabra *tiempo* que hace referencia a que el correcto funcionamiento del sistema depende no solo del resultado lógico que devuelve la computadora, sino también del tiempo en que se produce ese resultado. En definitiva, un Sistema en Tiempo Real no solo debe emitir respuestas correctas, sino que además debe cumplir restricciones temporales, siendo esta última la principal característica que los distingue de otro tipo de sistemas.



Figura 4: Representación gráfica de un Actor según la notación propuesta por el Lenguaje de Modelado Unificado.



Figura 5: Representación gráfica alternativa de un Actor, típicamente no humano.

- **Caso de Uso:** Representa una unidad funcional del sistema que puede ejecutar un determinado Actor o Actores. Por tanto, en un Caso de Uso existe comunicación entre el Actor y el sistema. Un Caso de Uso siempre lo inicia un Actor que puede, por ejemplo, solicitar información del sistema, modificar la información del sistema o informar al sistema de que ha ocurrido algo en su entorno que le incumbe. Por su parte, el sistema puede comunicarse con el Actor para comunicarle por ejemplo que ha sucedido algo en el sistema que le concierne, para que le ayude a tomar una decisión necesaria para cumplir los objetivos del sistema o para delegar alguna responsabilidad en el Actor. En la Figura 6 se observa la representación gráfica, mediante una elipse que contiene el nombre del Caso de Uso, de un Caso de Uso según la notación propuesta por el Lenguaje de Modelado Unificado.

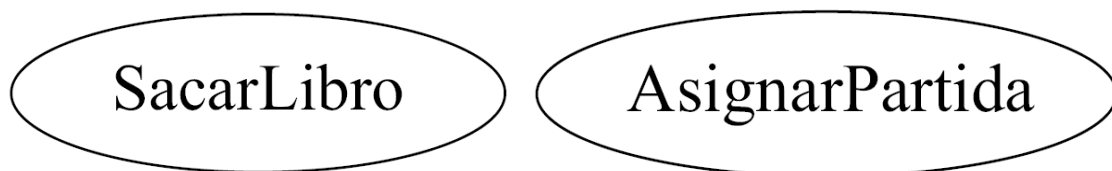


Figura 6: Representación gráfica de un Caso de Uso según la notación propuesta por el Lenguaje de Modelado Unificado.

### 3.1.1 Sintaxis UML avanzada: Generalización de Actores y Casos de Uso

Además, la **sintaxis avanzada** propuesta por el Lenguaje de Modelado Unificado permite:

- La generalización de Actores para agrupar varios Actores en uno, cuya representación gráfica puede verse en la Figura 7.
- La generalización de Casos de Uso para agrupar varios Casos de Uso en uno, cuya representación gráfica puede verse en la Figura 8.

Debe tenerse en cuenta que:

- Un Actor especializado hereda los Casos de Uso y Asociaciones de su generalización y puede agregar más elementos.
- Un Caso de Uso especializado hereda los objetivos y Actores de su generalización y puede agregar objetivos más específicos y los pasos para llevarlos a cabo.

Cuando existe una relación del tipo **generalización – especialización entre Actores** eso significa que un Actor es más general que otro. Igualmente, en lo relativo a los Casos de Uso, un Caso de Uso puede ser una forma especializada de otro Caso de Uso existente. La notación es una línea sólida terminada en un triángulo dibujado desde el Actor/Caso de Uso especializado al Actor/Caso de Uso general. Este tipo de relación puede ser útil, por ejemplo, en lo relativo a los Casos de Uso, **para describir los comportamientos comunes** una única vez mediante el Caso de Uso general y enfrentarse a los detalles excepcionales en los Casos de Uso especializados.

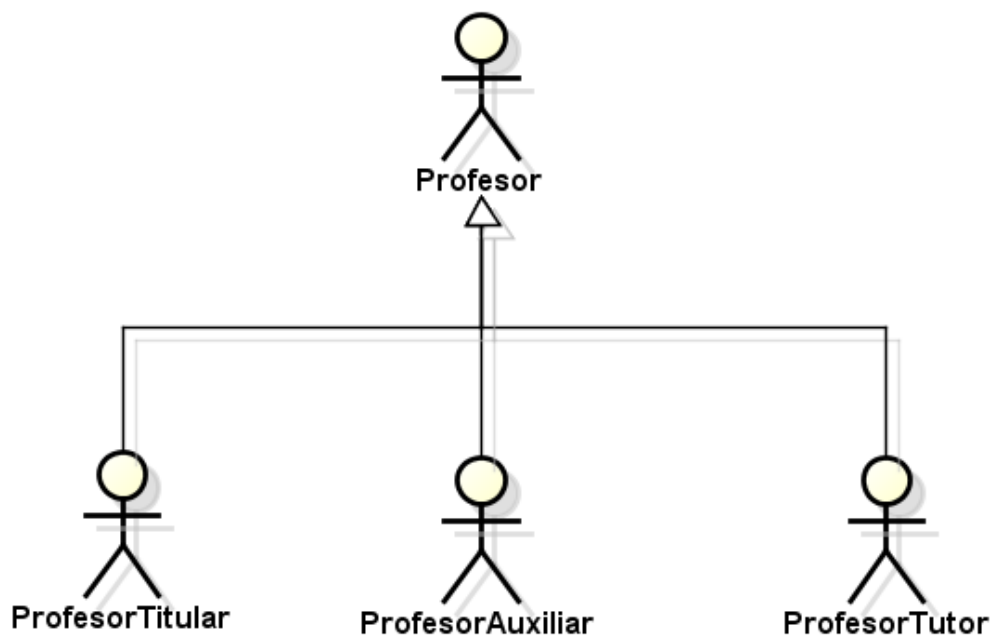


Figura 7: Representación gráfica de la Generalización de Actores en la notación avanzada del Lenguaje de Modelado Unificado.

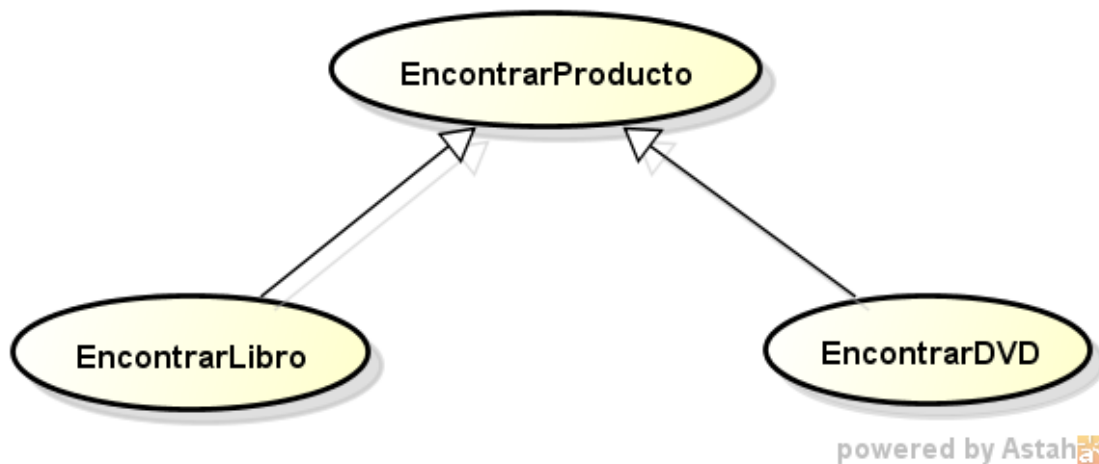


Figura 8: Representación gráfica de la Generalización de Casos de Uso en la notación avanzada del Lenguaje de Modelado Unificado.

Nótese, no obstante, que el uso de la sintaxis avanzada del Lenguaje de Modelado Unificado solo está recomendada si simplifica la representación del sistema software a modelar.

### 3.1.2 Sintaxis UML básica: Relaciones <<communicates>>

Con respecto a los tipos de relaciones o de conexiones entre los elementos del modelo, cuya notación gráfica ha sido definida por el Lenguaje de Modelado Unificado, la más habitual es comunica <<communicates>> que consiste en una relación o asociación entre un Actor y un Caso de Uso que denota la participación del Actor en el Caso de Uso.

Al mostrar este tipo de relación entre un Actor y un Caso de Uso en un Diagrama de Casos de Uso, no se especificará el estereotipo en cuestión <<communicates>> sobre la línea continua que dibuja la relación entre ambos puesto que el objetivo de dicha asociación está bien definido.

### 3.1.3 Sintaxis UML avanzada: Relaciones <<include>> y <<extend>>

No obstante, la sintaxis avanzada propuesta por el Lenguaje de Modelado Unificado también permite especificar los siguientes tipos de relaciones:

- Incluye <<include>> que permite especificar una relación de dependencia entre dos Casos de Uso que denota la inclusión del comportamiento de uno en el otro. Además, el primer Caso de Uso puede depender del resultado del Caso de Uso incluido. La notación a utilizar puede verse en la Figura 9 y en la Figura 10 y consiste en una línea discontinua con una punta de flecha abierta que apunta al Caso de Uso incluido y sobre la que se especifica la etiqueta «include». El Caso de Uso inicial incluye en su comportamiento el Caso de Uso incluido. Una relación A <<include>> a B significa que una instancia de A incorporaría el comportamiento especificado en B. Es decir, que el Caso de Uso incluido siempre existe cuando se produce el Caso de Uso inicial.

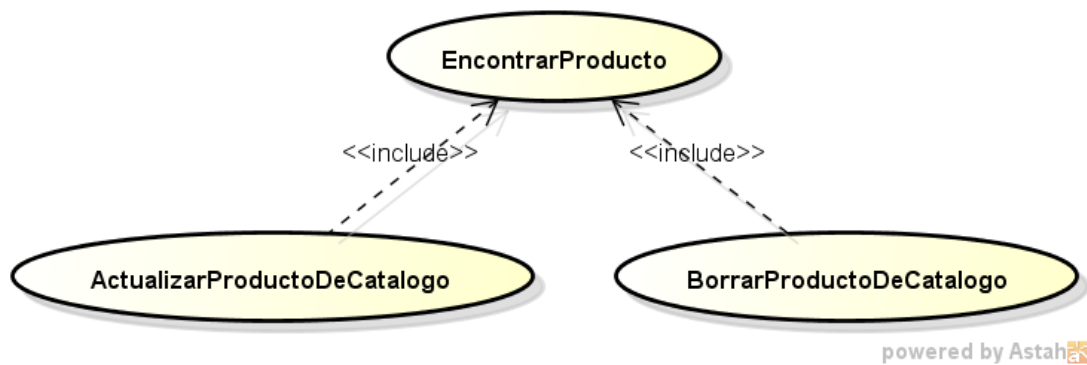


Figura 9: Representación gráfica de la Inclusión de Casos de Uso en otro en la notación avanzada propuesta por el Lenguaje de Modelado Unificado (I).<sup>10</sup>

La lectura del ejemplo de la Figura 9 sería la siguiente:

Tanto para actualizar un producto como para borrarlo, hay que realizar una búsqueda de producto.

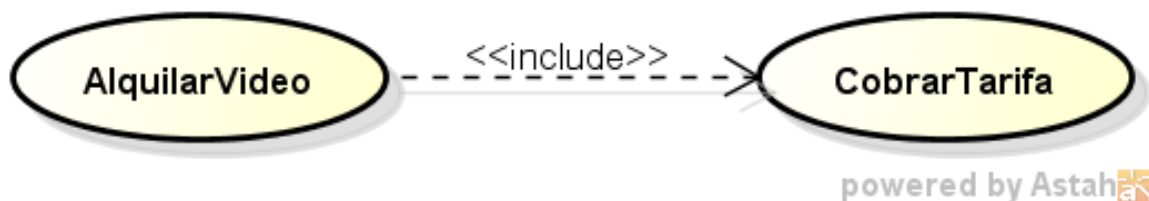


Figura 10: Representación gráfica de la Inclusión de Casos de Uso en otro en la notación avanzada propuesta por el Lenguaje de Modelado Unificado (II).

La lectura del ejemplo de la Figura 10 sería la siguiente:

Para alquilar un vídeo, es necesario cobrar la tarifa correspondiente.

Nótese que, tal y como se ha comentado y puede observarse en la Figura 9 y en la Figura 10, al mostrar este tipo de relación entre dos Casos de Uso en un Diagrama de Casos de Uso, sí se especificará el estereotipo en cuestión <<include>> sobre la línea discontinua con punta de flecha abierta que dibuja la relación entre ambos.

- Extiende <<extend>> que permite especificar una relación de dependencia entre dos Casos de Uso que denota que un Caso de Uso es una extensión de otro. Esta relación indica que el comportamiento del Caso de Uso extensión puede ser insertado en el Caso de Uso extendido o Caso de Uso base bajo ciertas condiciones. O dicho de otro modo, que el Caso de Uso extendido o Caso de Uso base se puede extender con el comportamiento del

<sup>10</sup> Ver también los ejemplos de especificación textual detallada de Casos de Uso de la Tabla 3 y de la **Error! Reference source not found.** en la sección 4 Especificación Textual de Casos de Uso, que incluyen una relación de este tipo.

Caso de Uso extensión en un punto concreto del primero bajo ciertas condiciones. Así, una relación `A <<extend>> B`, significa que una instancia del Caso de Uso B podría incorporar el comportamiento especificado en A siempre que se cumplan las condiciones especificadas en la extensión. El comportamiento se añadiría en el punto de extensión de B, referenciado por la relación `<<extend>>`. Este mecanismo puede ser útil para especificar casos especiales que supongan variaciones del comportamiento normal. La notación a utilizar puede verse en la Figura 11 y consiste en una flecha de punta abierta con línea discontinua con la etiqueta `<<extend>>` que se inicia en el Caso de Uso extensión.

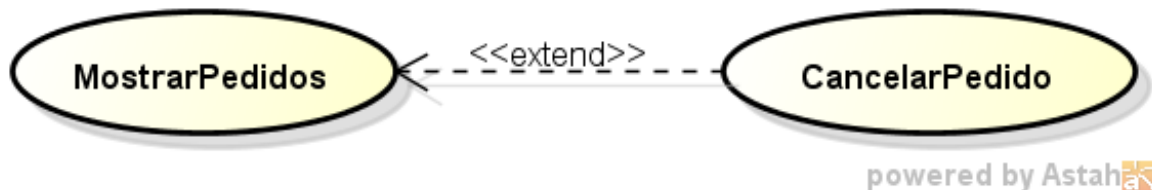


Figura 11: Representación gráfica de la Extensión de un Caso de Uso mediante otro en la notación avanzada propuesta por el Lenguaje de Modelado Unificado (I).

La lectura del ejemplo de la Figura 11 sería la siguiente:

Cuando se muestran los pedidos, en ocasiones, se cancela alguno de ellos.

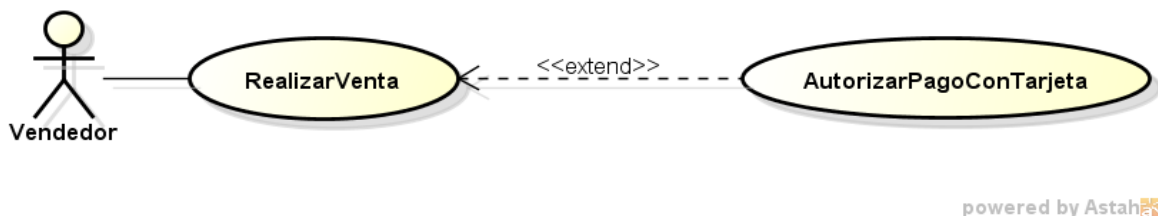


Figura 12: Representación gráfica de la Extensión de un Caso de Uso mediante otro en la notación avanzada propuesta por el Lenguaje de Modelado Unificado (II).

La lectura del ejemplo de la Figura 12 sería la siguiente:

Cuando se realiza una venta, en ocasiones, es necesario autorizar el pago realizado con la tarjeta.

Nótese que, tal y como se ha comentado y puede observarse en la Figura 11 y en la Figura 12, al mostrar este tipo de relación entre dos Casos de Uso en un Diagrama de Casos de Uso, sí se especificará el estereotipo en cuestión `<<extend>>` sobre la línea discontinua con punta de flecha abierta que dibuja la relación entre ambos.

Normalmente se utiliza una relación de tipo `<<extend>>` entre Casos de Uso cuando se tiene un Caso de Uso similar a otro pero que hace algo más<sup>11</sup> que este, es decir, que es una variante con respecto al primero. Por contra, se utiliza una relación de tipo `<<include>>` cuando existe una parte de comportamiento similar<sup>12</sup> en dos Casos de Uso y no se desea repetir la descripción de dicho comportamiento común. Dicho de otro modo, típicamente se podrá usar `<<extend>>` cuando se presente una variación del comportamiento normal e `<<include>>` cuando se repita un comportamiento en dos Casos de Uso y se desee evitar dicha repetición.

En una relación de tipo `<<extend>>`, un Actor que lleve a cabo el Caso de Uso base puede realizar o no sus extensiones. Mientras que en una relación de tipo `<<include>>` el Actor que realiza el Caso de Uso base también realiza el Caso de Uso incluido.

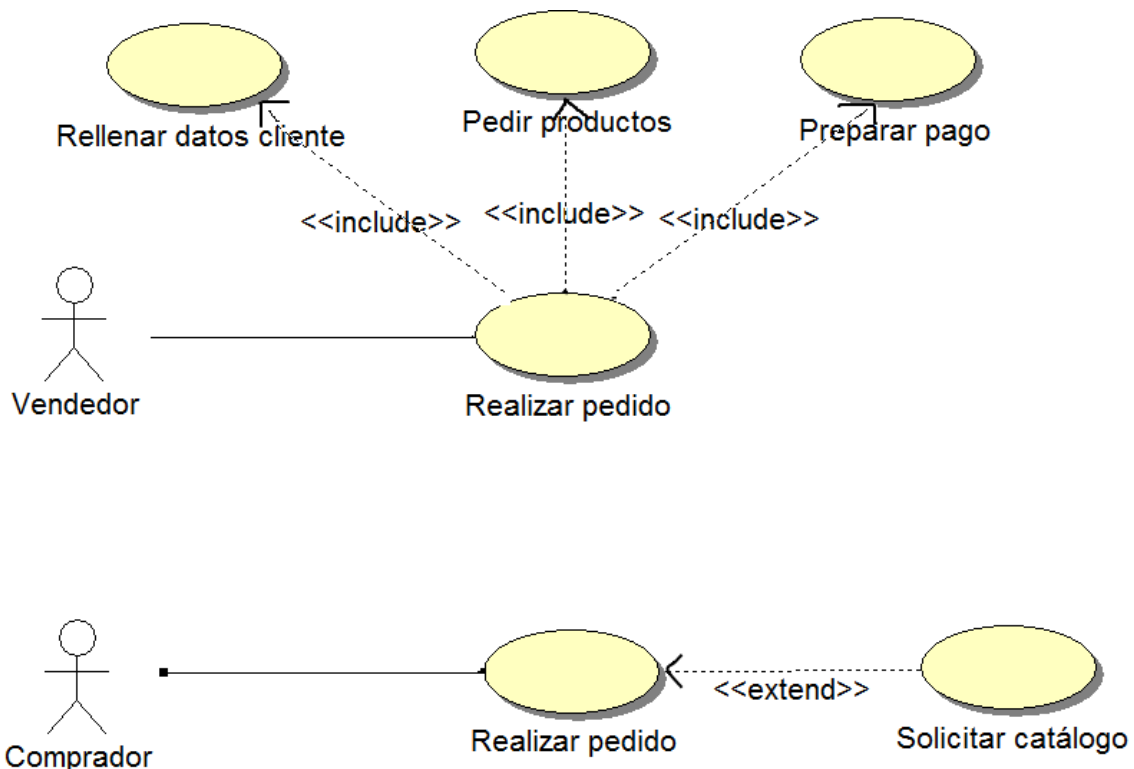


Figura 13: Ejemplo de Diagrama de Casos de Uso con inclusiones y extensiones de Casos de Uso (I).

En la Figura 13, el Vendedor puede Realizar un pedido lo cual implicaría que debería encargarse de Rellenar los datos del cliente, Pedir los productos y Preparar el pago, mientras que el Comprador puede Realizar un pedido y, opcionalmente, Solicitar el catálogo de productos.

<sup>11</sup> Ese algo más es lo que se modelaría como Caso de Uso extensión.

<sup>12</sup> Esa parte de comportamiento común es lo que se modelaría como Caso de Uso incluido.

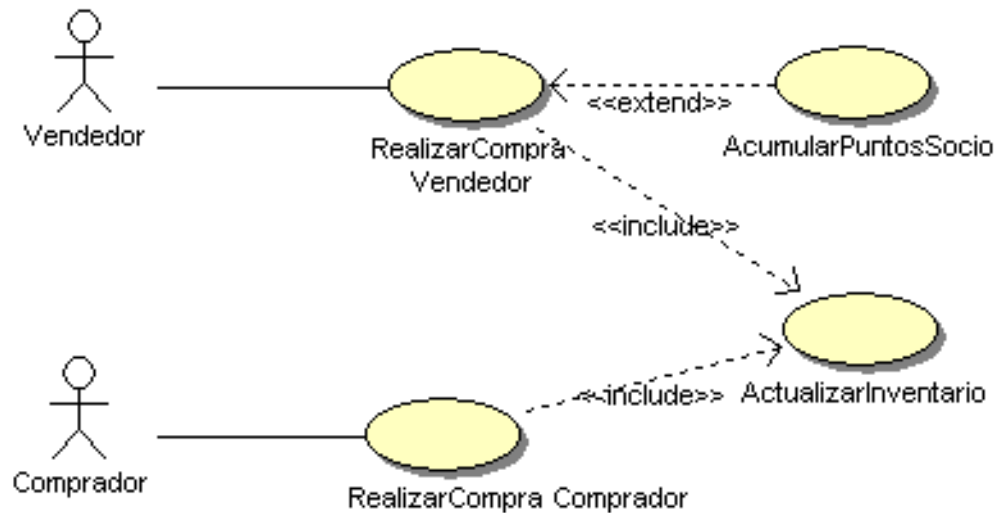


Figura 14: Ejemplo de Diagrama de Casos de Uso con inclusiones y extensiones de Casos de Uso (II).

De forma análoga, en la Figura 14, tanto el Vendedor como el Comprador pueden Realizar una Compra, que en ambos supondría Actualizar el Inventario. Y, por otra parte, cuando el Vendedor Realiza una Compra podría o no Acumular Puntos de Socio.

Finalmente, la Figura 15 muestra Casos de Uso incluidos, extensión y especializados. Cuando se Hace un Pedido, se Suministrarán los Datos de los Clientes, se Pedirá el Producto y se Pagará. Opcionalmente se podrá Solicitar un Catálogo. Por otra parte, se puede Pagar al Contado o a Crédito.

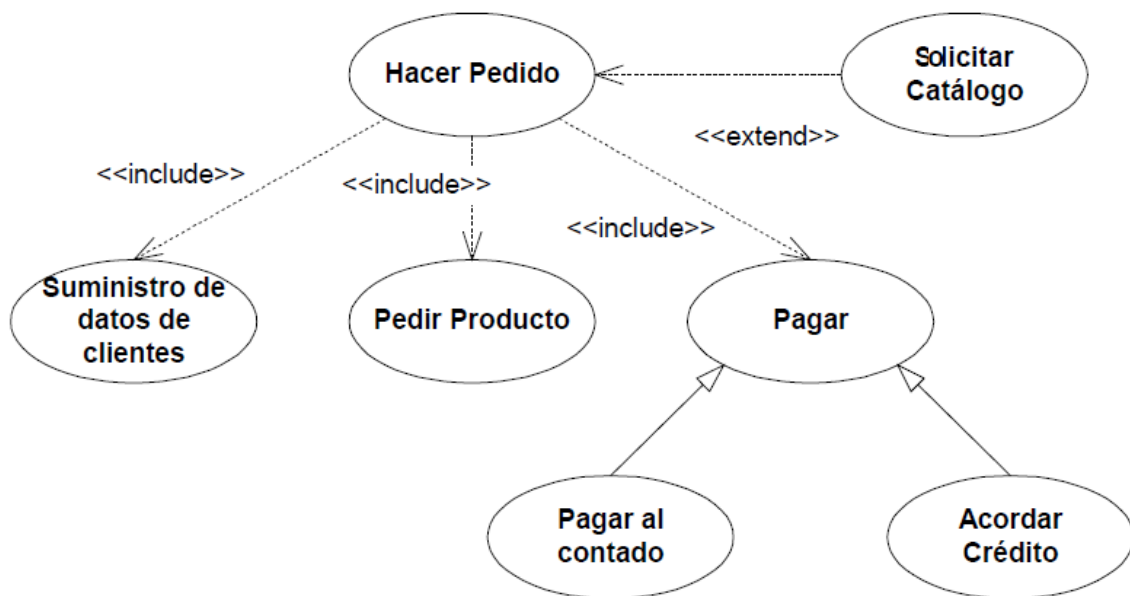




Figura 15: Ejemplo de Diagrama de Casos de Uso con inclusiones, extensiones y generalización de Casos de Uso.

## 4 ESPECIFICACIÓN TEXTUAL DE CASOS DE USO

### 4.1 ESPECIFICACIÓN TEXTUAL INFORMAL DE CASOS DE USO

Un Caso de Uso podría especificarse de manera informal de la siguiente forma:

Un cliente llega a la caja de una tienda con varios artículos. El cajero pasa uno a uno los artículos por una cinta para que el sistema los detecte. El sistema emite un recibo con el precio final. El cajero introduce la tarjeta de crédito del cliente para pagar, la cual es validada por el sistema. El sistema actualiza el inventario de productos de la tienda.

Texto 1: Especificación informal de un Caso de Uso.

### 4.2 ESPECIFICACIÓN TEXTUAL DETALLADA DE CASOS DE USO

La especificación informal de un Caso de Uso mostrada como ejemplo en el Texto 1 podría resultar válida, sin embargo, la especificación de los Casos de Uso suele hacerse de una forma más sistemática y organizada utilizando una tabla como la que se presenta a continuación. [ver Tabla 1]

Identificador	<Identificador del Caso de Uso> que comienza típicamente por un verbo	
Versión	<Número de Versión>	
Fecha	<Fecha>	
Autores	<Autores>	
Descripción	<Descripción del Caso de Uso>	
Personal involucrado o Actores	<Personas o entidades que se ven afectadas porque este Caso de Uso se lleve a cabo>	
Precondición	<precondición del Caso de Uso, es decir, las condiciones que se han de cumplir para que el flujo de secuencia se pueda llevar a cabo>	
Escenario principal de éxito o Secuencia o Flujo Normal	Paso	Acción
	1	{El <Actor>, el Sistema} <acción realizada por el actor o sistema>
	2	{El <Actor>, el Sistema} <acción realizada por el actor o sistema>

	3	[...]
	4	
	5	
	6	
	n	
Postcondición	<postcondición del Caso de Uso, es decir, los hechos que se han de cumplir si el flujo de eventos normal se ha ejecutado correctamente>	
Alternativas o Flujo Alternativo	Paso	Acción
	1	Si <condición de excepción>, {el <Actor>, el Sistema} <acción realizada por el actor o sistema> Permiten especificar el comportamiento del sistema en los casos menos frecuentes e inesperados.
	2	
	3	
	n	
Requisitos especiales	<Restricciones sobre la interfaz, de rendimiento, de seguridad, de compatibilidad, etc., es decir, Requisitos no Funcionales relacionados>	
Frecuencia esperada	<nº de veces> veces / <unidad de tiempo>	
Importancia	{sin importancia, importante, vital} u otra escala acordada	
Urgencia	{puede esperar, hay presión, inmediatamente} u otra escala acordada	
Comentarios	<comentarios adicionales>	

Tabla 1: Tabla con una plantilla para la especificación de un Caso de Uso.<sup>13</sup>

Nótese, no obstante que, en algunas ocasiones y, tal y como puede observarse en algunos de los ejemplos propuestos, **se utilizan tablas para la definición de los Casos de Uso** que pueden presentar alguna variación en los campos con respecto a la Tabla 1 puesto que la estructura y campos de dicha tabla no están estandarizados. Así, por ejemplo, con respecto a los autores del Caso de Uso, en ocasiones se especifican tanto los autores iniciales en un campo denominado, por ejemplo, *Creado por*, como aquellos que realizaron la última actualización en otro campo diferente denominado, por ejemplo, *Última Actualización por*. O bien, respecto a la fecha, en algunos casos se especificará tanto la fecha en la que inicialmente se creó el Caso de Uso indicándolo en un campo denominado, por ejemplo, *Fecha de Creación*, como la fecha en la que se realizó la última

<sup>13</sup> En alguno de los campos de la Tabla 1 como, por ejemplo, Importancia o Urgencia, también sería posible indicar que aún está por determinar utilizando las siglas PD, que son equivalentes al TBD (To Be Determined) empleado en los textos ingleses.

actualización indicándolo en un campo denominado, por ejemplo, Fecha de Última Actualización. También en ocasiones en vez de Requisitos Especiales se especifica ese campo como algún tipo de requisito concreto, por ejemplo, Rendimiento, tal y como ocurre en el Caso de Uso AsignarPartida especificado en la Tabla 2. O bien el campo Personal involucrado o Actores que describe las personas o entidades que se ven afectadas porque este Caso de Uso se lleve a cabo, se divide en dos campos, por una parte Actores primarios que describiría quién interactúa con el sistema y por otra Actores interesados que describiría a quiénes le interesa este Caso de Uso<sup>14</sup>.

Otros campos que también podrían aparecer en la especificación detallada de un Caso de Uso y que no se reflejan en la Tabla 1 son los siguientes:

- **Nombre del sistema:** Sistema a diseñar al cual corresponde el presente Caso de Uso.
- **Nivel del Caso de Uso:** Objetivo de Usuario o Subfunción. Esto hace referencia a la categorización de Casos de Uso propuesta por Alistair Cockburn [ver Figura 16]<sup>15</sup>:
  - o Objetivos globales.<sup>16</sup>
  - o Objetivos de Usuario.<sup>17</sup>
  - o Subfunciones.

Los Casos de Uso de nivel de Objetivo Global hacen referencia a funciones del negocio de alto nivel y resultan útiles cuando se desea capturar el detalle del sistema a alto nivel. Sin embargo, este nivel de detalle habitualmente se modelará mediante el Modelado de Procesos del Negocio<sup>18</sup> y no mediante Casos de Uso que típicamente se centrarán en el nivel de los Objetivos de Usuario. Por su parte, los Casos de Uso de nivel de Subfunción describen pequeñas tareas que se espera que un Actor realice como parte de un Caso de Uso de nivel de Objetivo de Usuario. A veces resulta interesante identificar los Casos de Uso de nivel de Subfunción y modelarlos como Casos de Uso extensión o incluidos, pero en otras ocasiones esto únicamente añade una complejidad innecesaria y no resulta útil.

<sup>14</sup> Es decir, el tipo de Actores que se ha denominado Actores pasivos.

<sup>15</sup> Alistair Cockburn es un investigador reconocido por su trabajo en el campo de la Ingeniería Software.

<sup>16</sup> En los textos ingleses a los Objetivos globales propuestos por Alistair Cockburn se les denomina Summary Goals.

<sup>17</sup> En los textos ingleses a los Objetivos de Usuario propuestos por Alistair Cockburn se les denomina User Goals.

<sup>18</sup> En los textos ingleses el Modelado del Proceso del Negocio se denomina Business Process Modeling (BPM). Un proceso de negocio es una colección de actividades diseñadas para producir una salida específica para un cliente o un mercado en particular. Esto implica un fuerte énfasis en cómo se realiza el trabajo dentro de una organización, en contraposición con un enfoque en el producto que se produce.

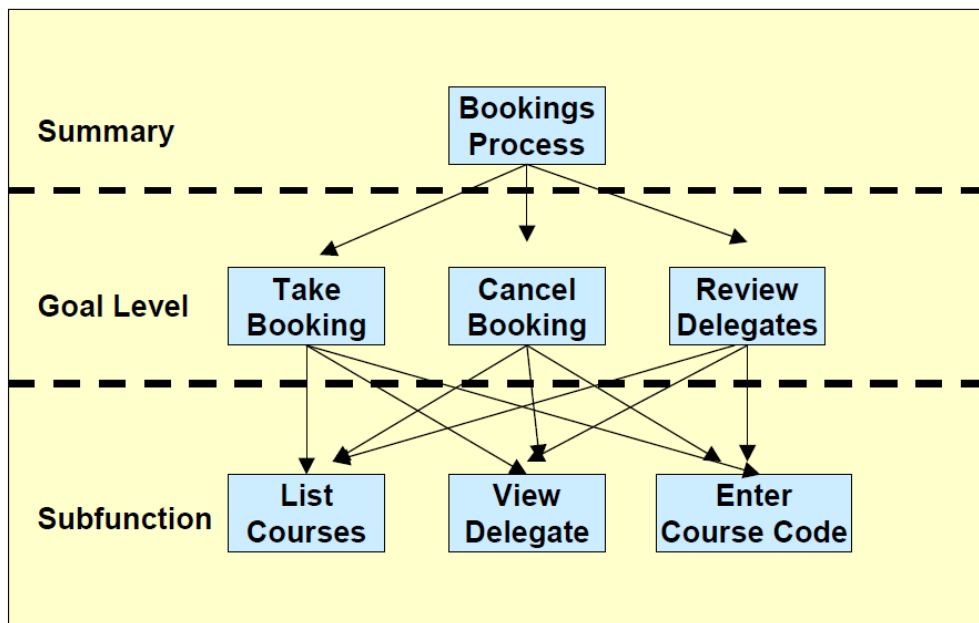


Figura 16: Niveles de Casos de Uso definidos por Alistair Cockburn. (Ariadne)

También debe tenerse en cuenta que durante la fase de Arranque del Proceso Unificado de Desarrollo de Software, la mayor parte de los Casos de Uso se especificarán de una forma resumida y breve, mientras que durante la fase de Elaboración, los Casos de Uso se especificarán de una forma más detallada, elaborada y extensa.

A continuación, se presentan diversos ejemplos de especificación de Casos de Uso [ver Tabla 2, Tabla 3, Tabla 4, Tabla 5, **Error! Reference source not found.** Tabla 6 y Tabla 7]:

Identificador	AsignarPartida	
Descripción	El GestorUniversal asigna una partida presupuestaria a un GestorPrograma.	
Precondición	Existe un presupuesto aprobado y existen en el sistema usuarios con rol de GestorPrograma.	
Escenario principal de éxito	Paso	Acción
	1	El GestorUniversal selecciona la opción Asignar Partida.
	2	El sistema muestra la lista de partidas en el presupuesto, proporcionando las claves orgánica, funcional y económica, así como la descripción textual de la partida.
	3	El GestorUniversal selecciona una de las partidas.
	4	El sistema muestra la lista de usuarios con rol de GestorPrograma, proporcionando nombre, apellidos y departamento del Ayuntamiento en el que trabajan.

	5	El GestorUniversal selecciona uno o más GestorPrograma, y solicita la asignación.
	6	El sistema realiza la asignación y muestra en una pantalla los datos de la partida (mismos datos que en el paso 2 para cada partida) y la lista de todos los GestorPrograma asignados (mismos datos que en el paso 4 para cada GestorPrograma) a la misma.
Postcondición	La partida presupuestaria ha quedado asignada a los GestorPrograma seleccionados.	
Alternativas	Paso	Acción
	1 - 5	Si el GestorUniversal selecciona Cancelar, el Caso de Uso termina sin que se haya realizado ninguna asignación de partidas a ningún GestorPrograma.
Rendimiento	No hay restricciones especiales.	

Tabla 2: Especificación del Caso de Uso AsingarPartida.

Identificador	GestionarUsuario	
Versión	1.0	
Fecha	14 Octubre 2021	
Autores	Pedro Pérez Pérez	
Descripción	El GestorUsuarios puede Crear, Ver, Modificar o Borrar usuarios en el sistema.	
Personal involucrado	El GestorUsuarios.	
Precondición	El sistema ha reconocido al usuario como GestorUsuarios.	
Escenario principal de éxito	Paso	Acción
	1	El GestorUsuarios selecciona la opción Gestionar Usuarios.
	2	El sistema pregunta qué se desea hacer: Crear Usuarios, Ver Usuarios, Modificar Usuarios o Borrar Usuarios.
	3a	El GestorUsuarios selecciona la opción Crear Usuarios.
	4a	<<include>> CrearUsuarios. <sup>19</sup>

<sup>19</sup> Ejemplo de relación de tipo <<include>> entre el Caso de Uso GestionarUsuario y el Caso de Uso CrearUsuarios. Véase más información en la sección del presente documento dedicada a este tema de forma específica.

	3b	El <code>GestorUsuarios</code> selecciona la opción Ver Usuarios.
	4b	<<include>> <code>VerUsuarios</code> . <sup>20</sup>
	3c	El <code>GestorUsuarios</code> selecciona la opción Modificar Usuarios.
	4c	<<include>> <code>ModificarUsuarios</code> . <sup>21</sup>
	3d	El <code>GestorUsuarios</code> selecciona la opción Borrar Usuarios.
	4d	<<include>> <code>BorrarUsuarios</code> . <sup>22</sup>
Postcondición	Las postcondiciones de los Casos de Uso incluidos.	
Alternativas	Paso	Acción
	1-4	Si el <code>GestorUsuarios</code> selecciona Cancelar, el Caso de Uso termina sin ninguna postcondición.
Requisitos especiales	Ninguno.	
Frecuencia esperada	Decenas de veces al día en el comienzo del uso de la aplicación. Muy infrecuente después.	
Importancia	Vital.	
Urgencia	Inmediatamente.	
Comentarios	No hay.	

Tabla 3: Especificación del Caso de Uso `GestionarUsuario`.

Id	<code>CrearUsuario</code>
Versión	1.0
Fecha	14 Octubre 2021
Autores	Pedro Pérez Pérez

<sup>20</sup> Ejemplo de relación de tipo <<include>> entre el Caso de Uso `GestionarUsuario` y el Caso de Uso `VerUsuarios`. Véase más información en la sección del presente documento dedicada a este tema de forma específica.

<sup>21</sup> Ejemplo de relación de tipo <<include>> entre el Caso de Uso `GestionarUsuario` y el Caso de Uso `ModificarUsuarios`. Véase más información en la sección del presente documento dedicada a este tema de forma específica.

<sup>22</sup> Ejemplo de relación de tipo <<include>> entre el Caso de Uso `GestionarUsuario` y el Caso de Uso `BorrarUsuarios`. Véase más información en la sección del presente documento dedicada a este tema de forma específica.

Descripción	El <code>GestorUsuarios</code> crea nuevos usuarios.	
Personal involucrado	El <code>GestorUsuarios</code> .	
Precondición	El sistema ha reconocido al usuario como <code>GestorUsuarios</code> .	
Escenario principal de éxito	Paso	Acción
	1	El <code>GestorUsuarios</code> selecciona la opción Crear Usuarios.
	2	El sistema solicita el nombre, apellidos, nombre de usuario, palabra clave y departamento de la empresa al que pertenece el nuevo usuario (para esto último muestra la lista de departamentos para que el <code>GestorUsuarios</code> elija uno).
	3	El <code>GestorUsuarios</code> proporciona los datos solicitados.
	4	El sistema crea la cuenta de usuario y le muestra al <code>GestorUsuarios</code> el número único asignado en el sistema al nuevo usuario y todos los datos proporcionados en el paso 3.
Postcondición	El nuevo usuario ha sido guardado en el sistema.	
Alternativas	Paso	Acción
	1-3	Si el <code>GestorUsuarios</code> selecciona Cancelar, el Caso de Uso termina sin ninguna postcondición.
	3.1	Si el <code>GestorUsuarios</code> no ha proporcionado alguno de los datos pedidos, se vuelve al paso 2, añadiendo un mensaje que indique que faltan datos.
	3.2	Si el nombre de usuario elegido ya existe en el sistema, se vuelve al paso 2, añadiendo un mensaje que indique que debe elegirse un nombre de usuario diferente.
Requisitos especiales	Ninguno.	
Frecuencia esperada	Decenas de veces al día en el comienzo del uso de la aplicación. Muy infrecuente después.	
Importancia	Vital.	
Urgencia	Inmediatamente.	
Comentarios	No hay.	

Tabla 4: Especificación del Caso de Uso `CrearUsuario`.

Id	<code>RealizarInscripción</code>
----	----------------------------------

Versión	1.0	
Fecha	14 Octubre 2021	
Autores	Pedro Pérez Pérez	
Descripción	El Participante se inscribe en el congreso.	
Personal involucrado	El Participante.	
Precondición	El proceso de inscripción está abierto.	
Escenario principal de éxito	Paso	Acción
	1	El Participante selecciona la opción Realizar inscripción.
	2	El sistema solicita nombre, apellidos, filiación y correo electrónico de quién se inscribe.
	3	El Participante proporciona los datos solicitados.
	4	El sistema busca los artículos entre cuyos autores figura el indicado anteriormente, a partir del correo electrónico, y muestra la lista de dichos artículos, preguntando cuáles va a presentar.
	5	El Participante selecciona los artículos que va a presentar.
	6	El sistema muestra la lista de talleres disponibles, con los horarios y precios, y solicita la selección de cero o más talleres.
	7	El Participante selecciona los talleres en los que desea inscribirse.
	8	El sistema calcula el importe total y solicita los datos de pago.
	9	El Participante proporciona un número de tarjeta de crédito, el nombre que aparece en ella y la fecha de caducidad.
	10	El sistema realiza el cobro y muestra un comprobante de dicho cobro.
Postcondición	El Participante se ha inscrito en el congreso y en cero o más talleres.	
Alternativas	Paso	Acción
	4.1	Si no hay artículos firmados por este asistente, se pasa directamente al paso 6.
	1-9	Si el usuario decide Cancelar, el Caso de Uso termina sin que la inscripción se haga efectiva.
	10.1	Si no se acepta el pago, el sistema muestra un aviso y la inscripción no se hace efectiva.



Requisitos especiales	La transmisión de los datos de la tarjeta de crédito debe llevarse a cabo de manera segura.
Frecuencia esperada	Cinco veces al día.
Importancia	Vital.
Urgencia	Hay presión.
Comentarios	No hay.

Tabla 5: Especificación del Caso de Uso `RealizarInscripción`.

Identificador	<code>CrearMensajeForo</code>	
Fecha	14 Octubre 2021	
Autores	Pedro Pérez Pérez	
Descripción	El <code>UsuarioForo</code> crea un mensaje en el foro de discusión.	
Personal involucrado	El <code>UsuarioForo</code> .	
Precondición	El <code>UsuarioForo</code> debe haberse autenticado en el sistema.	
Escenario principal de éxito	Paso	Acción
	1	El <code>UsuarioForo</code> pulsa sobre el botón que permite crear un nuevo mensaje.
	2	El sistema muestra una caja de texto para introducir el título del mensaje y un área de texto para introducir el cuerpo del mensaje.
	3	El <code>UsuarioForo</code> introduce el título del mensaje y el cuerpo de este.
	4	El sistema comprueba la validez de los datos y los almacena.
Postcondición	El nuevo mensaje ha sido creado en el foro.	
Alternativas	Paso	Acción
	1-3	Si el usuario decide Cancelar, el Caso de Uso termina sin que el mensaje se haya creado en el foro.
	4.1	Si el sistema comprueba la validez de los datos y los datos resultan no ser correctos, se avisa al <code>UsuarioForo</code> de ello permitiéndole que los corrija.
Requisitos especiales	No hay.	

Tabla 6: Especificación del Caso de Uso `CrearMensajeForo`.

Identificador	ApuntarseAGrupo	
Fecha	14 Octubre 2021	
Autores	Pedro Pérez Pérez	
Descripción	El <code>Alumno</code> puede apuntarse en un grupo de las asignaturas que se ha matriculado.	
Personal involucrado	El <code>Alumno</code> .	
Precondición	El <code>Alumno</code> debe haberse autenticado en el sistema.	
Escenario principal de éxito	Paso	Acción
	1	El <code>Alumno</code> selecciona la opción apuntarse a un grupo.
	2	El sistema presenta la lista de asignaturas de las que el <code>Alumno</code> está matriculado y no tiene aún grupo asignado.
	3	El <code>Alumno</code> selecciona una asignatura de las mostradas en la lista.
	4	El sistema presenta una lista de los grupos de dicha asignatura, tanto los ocupados como los libres.
	5	El <code>Alumno</code> selecciona uno de los grupos que aún se encuentran libres.
	6	El sistema muestra los datos y pide confirmación.
	7	El <code>Alumno</code> confirma los datos.
	8	El sistema registra al alumno en el grupo escogido.
Postcondición	El <code>Alumno</code> se ha apuntado a un grupo.	
Alternativas	Paso	Acción
	1-7	Si el <code>Alumno</code> decide Cancelar, el Caso de Uso termina sin que el <code>Alumno</code> quede registrado en ningún grupo de los disponibles en la asignatura.
	2.1	Si el sistema informa al <code>Alumno</code> de que no hay ninguna asignatura posible para apuntarse, el Caso de Uso termina sin que el <code>Alumno</code> quede registrado en ningún grupo de ninguna asignatura.
	4.1	Si el sistema informa al <code>Alumno</code> de que no hay ningún grupo disponible para apuntarse en la asignatura escogida, el Caso de Uso termina sin que el <code>Alumno</code> quede registrado en ningún grupo de dicha asignatura.

Rendimiento	Deben soportarse 10/20 operaciones concurrentes de este tipo, ya que se espera una carga proporcional al número de matriculados en la asignatura.
-------------	---

Tabla 7: Especificación del Caso de Uso `ApuntarseAGrupo`.

A continuación, se incluyen comentarios adicionales sobre algunos de los campos de la tabla utilizada para la especificación detallada de los Casos de Uso.

#### Requisitos Especiales:

Si un requisito no funcional o restricción se relaciona de manera específica con un Caso de Uso, se puede recoger en dicho Caso de Uso. Como ya es sabido, esto incluye cualidades tales como rendimiento, fiabilidad o facilidad de uso, o, en ocasiones, restricciones de diseño, por ejemplo, en relación a dispositivos de entrada/salida, que sean obligatorias o se consideren probables. Tal y como se ha comentado, una posibilidad es registrar estos requisitos con el Caso de Uso correspondiente. Sin embargo, muchos expertos encuentran útil el reunir todos los requisitos no funcionales y de otro tipo en un Catálogo de Requisitos o en un Documento de Especificación de Requisitos con una estructura más formal, con el objetivo de favorecer el análisis, validación y gestión de los mismos y de poderles considerar de manera conjunta.

Ejemplos de Requisitos Especiales a mayores del que aparece en la especificación del Caso de Uso `RealizarInscripción` enunciado como La transmisión de los datos de la tarjeta de crédito debe llevarse a cabo de manera segura [ver Tabla 5] son los siguientes:

- Interfaz de Usuario con pantalla táctil en un gran monitor de pantalla plana. El texto debe ser visible a 1 metro de distancia.<sup>23</sup>
- Tiempo de respuesta para la autorización de crédito de 30 segundos el 90 % de las veces.
- [...]

#### Precondiciones y Postcondiciones o Garantías de Éxito:

Las precondiciones establecen lo que siempre debe cumplirse antes de comenzar un escenario en un Caso de Uso. Las precondiciones no se comprueban en el Caso de Uso, sino que son condiciones que se asume que son verdad. Una precondición puede implicar que un escenario de otro Caso de Uso que se ha completado con éxito, como un inicio de sesión y/o autenticación del usuario, así precondiciones del tipo `El UsuarioForo` debe haberse autenticado en el sistema que aparece en la especificación del Caso de Uso `CrearMensajeForo` [ver Tabla 6] son habituales. Hay otras condiciones que deben ser verdad como, por ejemplo, que `El sistema` tiene energía, pero que desde un punto de vista práctico no tiene valor reflejarlas. Por el contrario, las precondiciones deben comunicar suposiciones importantes de las que el autor del Caso de Uso piensa que los lectores deberían ser avisados.

<sup>23</sup> Nótese que hay expertos como Alistair Cockburn que consideran que los Casos de Uso no deberían incluir detalles sobre la Interfaz de Usuario. Mientras que otros expertos consideran que los Casos de Uso reales o concretos sí podrían contender detalles sobre la Interfaz de Usuario. El concepto de Caso de Uso real o concreto se analiza, frente al de Caso de Uso esencial, más adelante en la presente documentación. Véase más información en la sección del presente documento dedicada a este tema de forma específica.

Las garantías de éxito o postcondiciones establecen qué debe cumplirse cuando el Caso de Uso se completa con éxito, o bien el escenario principal de éxito o bien algún camino alternativo, aunque en este último caso, es frecuente que el Caso de Uso finalice sin ninguna postcondición. Las postcondiciones deberían satisfacer las expectativas de todo el personal involucrado.

#### Escenario principal de éxito o Secuencia o Flujo Normal y Alternativas o Flujo Alternativo:

Los Casos de Uso deben representar los principales procesos extremo a extremo, distintos y completos de utilización de un sistema. Esto implica que el flujo normal no estará formado por un único paso, lo cual por otra parte es un error típico, sino que describirán una historia completa.

También debe tenerse en cuenta que el escenario principal de éxito supone un acuerdo entre clientes y usuarios e ingenieros de software sobre las responsabilidades del sistema. Mientras que las extensiones definidas proporcionarán a los ingenieros de software una lista de aspectos a los que deben prestar atención.

El escenario principal de éxito describe el camino de éxito típico que satisface los intereses del personal involucrado. Con frecuencia, no incluye ninguna condición o bifurcación, ya que muchos expertos consideran que resulta más claro el postergar todo el manejo de caminos condicionales a la sección de alternativas o flujo alternativo.<sup>24</sup>

El primer paso del escenario principal o de éxito debería ser el evento que desencadena el comienzo de la secuencia normal, el resto de los pasos de dicha secuencia pueden incluir validaciones, normalmente a cargo del sistema, o un cambio de estado realizado por el sistema, por ejemplo, almacenando o modificando información, etc.

Los flujos alternativos indican el resto de los escenarios o bifurcaciones posibles, tanto de éxito como de fracaso, lo cual hace que, en ocasiones, en la descripción de un Caso de Uso detallado, la sección de alternativas sea más larga y compleja que la correspondiente al escenario principal de éxito. La combinación del escenario principal y de los escenarios alternativos deberían satisfacer todos o, al menos gran parte, de los intereses del personal involucrado, ya que debe tenerse en cuenta que existe cierta limitación puesto que algunos intereses podrían capturarse mejor como requisitos no funcionales y contemplarse en el Catálogo de Requisitos o en el Documento de Especificación de Requisitos en lugar de en la sección de Requisitos Especiales de los Casos de Uso.

Como los flujos alternativos especifican bifurcaciones del escenario principal de éxito conviene etiquetarlos en relación con los pasos de la secuencia normal. Así, por ejemplo, si se supone que el Paso 3 del escenario principal de éxito es 3. El Cajero introduce el identificador del artículo y este resulta inválido, bien porque no se introdujo correctamente, bien porque el sistema no lo reconoce, una extensión de este paso se podría etiquetar como 3.1 o 3 a<sup>25</sup> y primero identificaría la condición y después la respuesta del sistema, igualmente una extensión alternativa al Paso 3.1 o 3 a se podría etiquetar como 3.2 o 3 b y así sucesivamente. Nótese también que una extensión definida para un determinado paso, tal y como se acaba de comentar tendrá, típicamente, dos partes, que son la condición que da lugar a ese camino alternativo y cómo dicha condición se maneja, siendo recomendable expresar la condición como algo que pueda ser detectado por el sistema o por un Actor. Así por ejemplo, entre 5.1 El sistema detecta un fallo en la comunicación con el servicio externo del sistema de cálculo de impuestos... y 5.1 El sistema de

<sup>24</sup> Entre otros motivos porque esto complicaría la realización posterior de los Diagramas de Secuencia del Sistema correspondientes a los escenarios principales de éxito de los Casos de Uso. Y además no debe perderse de vista que lo que intenta representarse es un escenario principal de éxito.

<sup>25</sup> O simplemente 3 como se hace en varios de los ejemplos propuestos en la Tabla 2, la Tabla 3, la Tabla 4 y la Tabla 5.

cálculo de impuestos externo no funciona es preferible la primera opción porque se trata de algo que el sistema puede detectar, mientras que en el segundo caso se expresa como una inferencia<sup>26</sup> por lo que el comportamiento del sistema – El sistema detecta un fallo en la comunicación con el servicio externo del sistema de cálculo de impuestos – debe deducirse u obtenerse como conclusión a partir de la información proporcionada – El sistema de cálculo de impuestos externo no funciona–.

Además, el manejo de la extensión alternativa de un paso del flujo normal podrían ser varios pasos, tal y como puede observarse en la especificación del Caso de Uso `CrearUsuario` [ver Tabla 4] para la extensión del Paso 4 de la secuencia normal. O bien suceder lo contrario, es decir que la extensión alternativa de un conjunto de pasos del flujo normal se resumiera en un único paso, tal y como puede observarse en la especificación del Caso de Uso `AsignarPartida` [ver Tabla 2] para la extensión de los Pasos 1–6 de la secuencia normal.

Al final del manejo de la extensión, por defecto, el escenario indicado por la extensión se une de nuevo con el escenario principal de éxito, a menos que en la extensión se indique algo diferente.

#### Reglas del Dominio:

Las reglas del dominio en general no se describen en los Casos de Uso<sup>27</sup>. No obstante, durante la especificación de los Casos de Uso es recomendable identificar aquellas reglas del dominio que permitan esclarecer el contenido de Casos de Uso ambiguos o incompletos. Así, por ejemplo, en un sistema de Ventas alguien podría preguntar si es necesario agregar a un Caso de Uso denominado `ProcesarVenta`, un flujo alternativo de pago a crédito en el que no fuera necesario solicitar la firma – o el pin – al cliente, pero si existe una regla del dominio que establece que ninguna compañía de crédito autoriza un pago sin la firma – o el pin –, no será necesario agregar dicho curso alternativo.

Igualmente, en ocasiones, los expertos del dominio bajo estudio proporcionan informaciones adicionales que es importante tener en cuenta para la correcta comprensión del dominio por parte de los ingenieros de software y para el planteamiento de los Casos de Uso del sistema. Así por ejemplo, en un sistema de Ventas podría ser importante entender la interpretación del código de barras.

El grado de la formalidad de un proyecto particular de software, su alcance y complejidad y la etapa del proyecto (Fase de Proceso Unificado de Desarrollo de Software) en la que este se encuentre, influenciarán el nivel del detalle requerido en cada Caso de Uso.

En el Anexo 1: Plantilla en lengua inglesa para la especificación completa de un Caso de Uso se proporciona una plantilla en lengua inglesa para una descripción completa de un Caso de Uso<sup>28</sup> con el único objetivo de que el alumno se familiarice con la jerga utilizada en los documentos de Casos de Uso en dicho idioma por si en algún momento le resultara de utilidad.

<sup>26</sup> El término inferencia significa deducción de una cosa a partir de otra, conclusión.

<sup>27</sup> Nótese que esto no siempre es así y que, en ocasiones, sí se incluye una sección en la descripción de los Casos de Uso que contenga las Reglas de Dominio que afecten al Caso de Uso en cuestión. Así por ejemplo, en la plantilla contenida en el Anexo 1: Plantilla en lengua inglesa para la especificación completa de un Caso de Uso del presente documento aparece, en el apartado 10.4.12 Business Rules, una sección dedicada a especificar las Reglas del Negocio que afectan a dicho Caso de Uso. Tal y como es sabido, el término Reglas del Negocio es más restrictivo que el término Reglas del Dominio, no obstante, en ocasiones se intercambian y se utilizan como si fueran sinónimos. Véase más información en el documento dedicado a este tema de forma específica.

<sup>28</sup> En los textos ingleses la especificación completa de un Caso de Uso se denomina *full-dressed Use Case*.

## 5 CASOS DE USO ESENCIALES Y REALES O CONCRETOS

En ocasiones los Casos de Uso se clasifican en esenciales y reales o concretos.<sup>29</sup> Los esenciales se centrarían en las intenciones de los Actores y en las responsabilidades del sistema, postergando los detalles de la Interfaz de Usuario y otro tipo de decisiones, mientras que los reales o concretos sí podrían contener dichos detalles. Es decir, que los Casos de Uso esenciales utilizarían un lenguaje abstracto que típicamente no incluye referencias a las tecnologías a utilizar para la implementación, ni suposiciones relativas a la interfaz de usuario u otras cuestiones, mientras que en los Casos de Uso concretos o reales sí pueden existir compromisos con ciertas tecnologías o contener detalles acerca de la interfaz de usuario.

- Un ejemplo de un Caso de Uso esencial sería El Titular de la cuenta introduce sus datos en el cajero automático.
- Un ejemplo de un Caso de Uso real o concreto sería:
  - o El Titular introduce su tarjeta en el lector del cajero automático.
  - o El cajero automático muestra la ventana A.
  - o El Titular introduce su número de identificación personal (PIN) mediante el teclado numérico, etc.

A medida que nos desplazamos del Análisis al Diseño, se irá optando por las descripciones reales o concretas para los Casos de Uso, dejando atrás las esenciales.

Estos dos tipos de Casos de Uso serían los tipos extremos ya que en la práctica los Casos de Uso especificados no suelen ser ni puramente esenciales ni puramente reales, ya que en un Caso de Uso esencial, muchas veces el simple hecho del orden de los pasos en los flujos normal y/o alternativos implica la toma de una determinada decisión, y por mucho detalle que se especifique en un Caso de Uso real o concreto siempre queda un margen a los ingenieros de software para tomar ciertas decisiones, por ejemplo, con respecto a la tecnologías de implementación.

## 6 GRANULARIDAD DE LOS CASOS DE USO

En ocasiones es complicado decidir si se está especificando un Caso de Uso válido o útil, ya que las tareas se pueden agrupar a muchos niveles de granularidad, desde un paso pequeño hasta una actividad a nivel de empresa. Es decir, que es posible hablar de objetivos globales o compuestos, a nivel de empresa, como por ejemplo, *ser rentable*, que incluye muchos objetivos intermedios a nivel de uso de la aplicación, como por ejemplo, *se capturan las ventas*, que a su vez incluyen objetivos de nivel de subfunción dentro de las aplicaciones, como por ejemplo, *la entrada es válida*.<sup>30</sup>

<sup>29</sup> En la primera edición de la obra de Larman *Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design and Iterative Development* se habla de Casos de Uso esenciales y reales, mientras que a partir de la segunda edición se habla de Casos de Uso esenciales y concretos.

<sup>30</sup> Esto hace referencia a la categorización de Casos de Uso propuesta por Alistair Cockburn ya presentada en esta documentación. Véase más información en la sección del presente documento dedicada a este tema de forma específica.

El Caso de Uso EBP es una regla que propone centrarse en los Casos de Uso al nivel de Procesos del Negocio Elementales<sup>31</sup>, es decir, que los Casos de Uso de una aplicación deberían corresponderse con los Procesos Elementales del Negocio. Proceso Elemental del Negocio es un concepto que procede del campo de la Ingeniería de Procesos del Negocio y que puede definirse como *una tarea realizada por una persona en un lugar, en un instante, como respuesta a un evento del negocio, que agrega un valor cuantificable para el negocio y deja los datos en un estado consistente, por ejemplo, autorizar crédito*.

Un Caso de Uso no debería abordar un pequeño paso como *eliminar una línea de pedido o imprimir el documento o introducir una tarjeta de crédito*, sino que el escenario principal de éxito debería estar formado por una serie de pasos, o, dicho de otra forma, el escenario principal de éxito debería contar una historia. Aunque tampoco debería el Caso de Uso abordar una tarea que durara días o múltiples sesiones, como *negociar un contrato con el proveedor*, sino que debería plantearse una tarea que típicamente se abordara en una única sesión con una duración limitada. Además, idealmente, el sistema y los datos se encontrarán al terminar en un estado estable.

Los Actores tienen objetivos o necesidades y utilizan las aplicaciones o sistemas software para ayudarles a satisfacerlos, por este motivo, en ocasiones un Caso de Uso de nivel EBP se denomina Caso de Uso de nivel de Objetivo de Usuario<sup>32</sup>, para remarcar que sirve, o debería servir, para satisfacer un objetivo de nivel de usuario típicamente de un Actor principal del sistema. Así, el procedimiento para modelar Casos de Uso podría ser el que a continuación se describe. En primer lugar, trazar los límites del sistema. A continuación, se identifican los Actores principales como aquellos que tienen objetivos de nivel de usuario que se satisfacen mediante el uso de los servicios del sistema. Para después identificar los objetivos de nivel de usuario que, para cada usuario, se elevan hasta el nivel de objetivos de usuario más alto que satisfaga la regla EBP. Finalmente, se definen los Casos de Uso que satisfagan los objetivos de usuario definidos y se nombran de acuerdo con los mismos. Es decir, que en lugar de preguntar cuáles son los Casos de Uso, se preguntaría a los usuarios del sistema cuáles son sus objetivos y el nombre del Caso de Uso para cada objetivo de usuario debería reflejar ese objetivo, así por ejemplo, si el objetivo de usuario es *Capturar o procesar una venta*, el Caso de Uso podría denominarse *ProcesarVenta*.

Además, una recomendación a la hora de determinar los objetivos de usuario es preguntar al usuario cuáles son sus objetivos, y a continuación cuál es el objetivo de nivel más alto que motiva el objetivo anterior, etc., de manera que se pueda ascender en la jerarquía de objetivos y visualizar así objetivos que agreguen un valor añadido para descubrir qué es lo que el personal involucrado quiere del sistema que se está estudiando.<sup>33</sup>

Si se plantea el ejemplo de un sistema de tipo cajero automático, para cumplir el objetivo de retirar dinero de un cajero, se seguiría aproximadamente la serie de pasos descrita a continuación:

- Introducir tarjeta.
- Introducir número pin.
- Seleccionar cantidad a retirar.
- Confirmar la cantidad solicitada.
- Retirar tarjeta.

<sup>31</sup> En los textos ingleses los Procesos del Negocio Elementales se denominan *Elementary Business Processes (EBP)*.

<sup>32</sup> Esto hace referencia nuevamente a la categorización de Casos de Uso propuesta por Alistair Cockburn ya presentada en esta documentación. Véase más información en la sección del presente documento dedicada a este tema de forma específica.

<sup>33</sup> Esta estrategia para obtener Casos de Uso es la sugerida por Larman en su obra *Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design and Iterative Development* (Larman).

- Retirar dinero.
- Recoger recibo.

Cada uno de los pasos anteriormente descritos no debería constituir un Caso de Uso en sí mismo, puesto que esto daría lugar a un gran número de pequeños Casos de Uso que no satisfacen ningún objetivo del usuario por sí solos, por lo que el Caso de Uso a identificar debería corresponder al objetivo del usuario de *retirar dinero del cajero* y podría denominarse `RetirarDinero`.

Aunque los Casos de Uso base<sup>34</sup> de una aplicación deberían satisfacer la regla EBP, también podrían existir Casos de Uso que no fueran EBP, es decir, definidos a un nivel inferior, que fueran típicamente subtareas de un Caso de Uso base. Así, por ejemplo, una subtask o extensión como *pago a crédito* podría resultar adecuado separarla en un Caso de Uso `PagarACredito` propio, aunque no satisfaga la directriz EBP, y conectarlo a uno o varios Casos de Uso base. Por otra parte, una subtask como *identificarse y ser validado o iniciar sesión* tampoco sería un objetivo de usuario, sino un objetivo de nivel más bajo, es decir, un objetivo de nivel de subfunción<sup>35</sup> que da soporte a un objetivo de usuario y que también podría modelarse como un Caso de Uso `AutenticarUsuario` que probablemente sería necesario vincular de alguna forma, por ejemplo usando su resultado como precondition, o como un Caso de Uso incluido, en varios Casos de Uso de nivel de objetivo de usuario.

En relación con esta cuestión, hay expertos que consideran que no es erróneo escribir Casos de Uso de nivel de subfunción. No obstante, como ya se ha comentado, solo deberían escribirse Casos de Uso para objetivos de subfunción de manera ocasional. El motivo es que hacerlo no siempre es útil puesto que agrega complejidad al Modelo de Casos de Uso ya que puede haber cientos de objetivos de nivel de subfunción y, por lo tanto, cientos de Casos de Uso de nivel de subfunción en un sistema y el número y granularidad de los Casos de Uso influyen en el tiempo y la dificultad para entender, mantener y gestionarlos adecuadamente.

Los Casos de Uso se pueden especificar a niveles diferentes y pueden hacer referencia a Casos de Uso de nivel inferior. Estos diferentes niveles, de objetivos y de Casos de Uso, son una fuente típica de confusión en la identificación del nivel adecuado para los de los Casos de Uso de una aplicación. La regla EBP pretende proporcionar una orientación para eliminar Casos de Uso de nivel excesivamente bajo.

Normalmente, los Casos de Uso de nivel de objetivo de usuario se corresponderán uno a uno con los objetivos de usuario identificados. No obstante, aunque, tal y como se acaba de comentar, lo habitual es definir un Caso de Uso de nivel EBP por cada objetivo de usuario con un nombre similar al del objetivo de usuario pero comenzando por un verbo, una excepción común a la regla de *un Caso de Uso por objetivo*, es agrupar objetivos separados del tipo CRUD (Create, Retrieve, Update, Delete), es decir, crear, recuperar o ver, actualizar o modificar y eliminar o borrar una entidad en un solo Caso de Uso, denominado por convenio `GestionarEntidad`. Así, por ejemplo, el Caso de Uso `GestionarUsuario` equivaldría a `CrearUsuario`, `VerUsuario`, `ModificarUsuario` y `BorrarUsuario`<sup>36</sup>, o dicho de otro modo, los objetivos de crear, ver, modificar y borrar usuarios quedarían agrupados en el Caso de Uso `GestionarUsuarios`.<sup>37</sup>

<sup>34</sup> Véase Casos de Uso incluidos y extensión en la sección del presente documento dedicada a este tema de forma específica.

<sup>35</sup> Esto hace nuevamente referencia a la categorización de Casos de Uso propuesta por Alistair Cockburn ya presentada en esta documentación. Véase más información en la sección del presente documento dedicada a este tema de forma específica.

<sup>36</sup> Nótese, no obstante que, en realidad, será necesario especificar todos los Casos de uso mencionados, es decir, `GestionarUsuario`, `CrearUsuario`, `VerUsuario`, `ModificarUsuario` y `BorrarUsuario`.

<sup>37</sup> Ver ejemplo en el Caso de Uso `GestionarUsuario` definido en la Tabla 3.



## 7 IDENTIFICACIÓN DE ACTORES Y CASOS DE USO

Para lograr identificar los Actores del sistema, se puede plantear las cuestiones que a continuación se presentan:

- ¿Quién usa el sistema?
- ¿Quién lo mantiene?
- ¿Quién lo arranca y para?
- ¿Quién gestiona los usuarios y la seguridad?
- ¿Con qué sistemas externos interactúa?
- ¿Quién proporciona o recibe información del sistema?<sup>38</sup>
- ¿Quién evalúa la actividad o el rendimiento del sistema?
- ¿Ocurren cosas en momentos determinados? Es decir, ¿es el Tiempo un Actor porque el sistema hace algo como respuesta a un evento relacionado con el tiempo?
- ¿Existe un proceso de control que reinicie el sistema si falla?
- ¿Cómo se gestionan las actualizaciones de software? ¿Son automáticas o no?
- [...] <sup>39</sup>

Además, a la hora de identificar los Actores debe tenerse en cuenta que los Actores son agentes externos al sistema y que, por lo tanto, no es el ingeniero de software quién decide cómo actúan. E igualmente que los Actores interactúan con el sistema y que, si se trata de seres humanos, habitualmente representarán roles, lo cual implica que una misma persona puede desempeñar varios roles. Igualmente, conviene no olvidar que aunque los Actores son frecuencia operadores humanos, existen otros tipos de Actores.

Por su parte, para identificar los Casos de Uso resultará útil plantear las siguientes cuestiones:

- ¿Cuáles son las principales funciones o tareas realizadas por los Actores? ¿Y qué funciones esperan del sistema?
- ¿Se notifica a los Actores cuándo hay cambios de estado?
- ¿Hay eventos externos que afecten al sistema? ¿Cómo? ¿Deberá el Actor informar al sistema de los cambios producidos en el entorno?
- ¿El sistema almacena y captura información? ¿Cómo? ¿Qué información del sistema adquiere, produce o transforma el Actor?

Además, a la hora de identificar Casos de Uso deberá tenerse en cuenta, tal y como ya es conocido, que siempre son comenzados por un Actor y que deben describirse desde el punto de vista del Actor, es decir que un Caso de Uso debe satisfacer un objetivo de un Actor.<sup>40</sup>

<sup>38</sup> Ya que se trata de identificar no solo a los Actores activos, sino también a los Actores pasivos.

<sup>39</sup> Se trata de una lista de preguntas abierta que el alumno puede completar para conseguir descubrir todos los Actores de un sistema de la forma más eficaz posible.

Antes de realizar el Diagrama inicial de Casos de Uso, lo ideal es organizar una reunión con el cliente o usuario con el objetivo de realizar una lluvia de ideas<sup>41</sup> que permita identificar los Actores y los Casos de Uso para cada Actor, así como encontrar una justificación para cada Actor y para cada Caso de Uso. Posteriormente, se expondrá el trabajo realizado al cliente o usuario para tomarlo como base a la hora de discutir y concretar.

Además, tal y como ya se ha comentado, tanto en las conversaciones con el cliente o usuario, como en la descripción textual de los Casos de Uso como tal, se debería evitar la jerga excesivamente técnica, prefiriéndose un lenguaje cercano al del cliente o usuario que, típicamente, será un experto en el campo del saber o dominio en el que se ubique el sistema software. También, como ya es sabido, se tratará de identificar Casos de Uso que describan Procesos de Negocio Elementales no a tan alto nivel como *ganar dinero para mi empresa*, pero tampoco a tan bajo nivel como *iniciar el cobro*.<sup>42</sup>

Cada Caso de Uso debe centrarse en describir una única tarea del negocio y ser lo suficientemente sencillo y claro para que su desarrollo pueda ser abordado de forma adecuada por el ingeniero de software. Así, un Caso de Uso describiría una característica del sistema, y para la mayoría de los proyectos de software, esto significa que será necesario especificar numerosos Casos de Uso para definir completamente un sistema.

Además, debe tenerse en cuenta que, sobre cómo escribir Casos de Uso existen diferentes puntos de vista, así Alistair Cockburn proporciona en su web<sup>43</sup> una serie de recomendaciones<sup>44</sup> que hemos completado. Los buenos Casos de Uso:

- Son fundamentalmente texto. Los Diagramas de Casos de Uso y las relaciones entre los Casos de Uso son secundarios en el trabajo con los Casos de Uso. Trabajar con Casos de Uso significa escribir texto. Lo que verdaderamente es importante es el hecho de detallar por escrito junto con el cliente o usuario los diferentes Casos de Uso del sistema, ya que los Casos de Uso son historias del cliente. (Larman)
- No incluyen detalles sobre la Interfaz de Usuario ni sobre los formatos de datos.<sup>45</sup>
- Contienen típicamente entre 3 y 9 pasos en el escenario principal de éxito.
- Son fáciles de leer. La especificación de los Casos de Uso no debe complicarse ya que, en ocasiones, lo mejor es enemigo de lo bueno. De hecho, es importante evitar exceso de detalle.
- Están definidos a nivel de los objetivos del cliente o usuario y no de los objetivos del software a construir ya que lo importante es dar respuesta a las necesidades del cliente o usuario.
- Son la secuencia de las acciones que tienen lugar entre los Actores y el sistema, por lo que constituyen un conjunto de escenarios de éxito y de fallo de los Actores y del sistema.

---

<sup>40</sup> Se trata de una lista de preguntas abierta que el alumno puede completar para conseguir descubrir todos los Casos de Uso de un sistema de la forma más eficaz posible.

<sup>41</sup> En los textos ingleses una lluvia de ideas se denomina *brainstorming*.

<sup>42</sup> Véase Niveles de los Casos de Uso en la sección del presente documento dedicada a este tema de forma específica.

<sup>43</sup> Véase <http://alistair.cockburn.us/> para más información.

<sup>44</sup> Véase <http://alistair.cockburn.us/Sampler+of+good+and+bad+use+cases> para más información.

<sup>45</sup> Nótese, que como ya se ha comentado, algunos autores opinan lo contrario.

- La suma del conjunto de Casos de Uso definido debe equivaler al sistema, por lo que aquello que no quede cubierto por alguno de los Casos de Uso identificado se entiende que estará fuera del sistema.
- Los Casos de Uso no solo hay que escribirlos, sino que además deben revisarse cuidadosamente con el cliente o usuario.
- Suponen la toma de diferentes decisiones.

Finalmente, también debe tenerse en cuenta que la elaboración de Casos de Uso no debe ser una actividad analítica, sino sintética, ya que no se trata de analizar y desmenuzar algo que ya existe, sino de crear, junto con el cliente o usuario, una concepción común del sistema a desarrollar o a mejorar.

## 8 VENTAJAS Y LIMITACIONES DE LOS CASOS DE USO

### 8.1 VENTAJAS

Como técnica de captura de Requisitos (Funcionales) permite que el analista se centre en las necesidades del usuario, es decir, en qué espera éste lograr al utilizar el sistema, evitando que los ingenieros de desarrollo software decidan la funcionalidad del nuevo sistema basándose solo o principalmente en criterios tecnológicos.

A su vez, durante la captura de Requisitos, el analista se concentra en las tareas centrales del usuario describiendo paralelamente los Casos de Uso que mayor valor aportan al negocio, facilitándose así la priorización de Requisitos y Casos de Uso.

Además, facilitan la comunicación entre el cliente o usuario y los ingenieros de software. Y se considera que guían todo el proceso de desarrollo del proyecto de software<sup>46</sup> y que permiten una adecuada planificación del trabajo, además de proporcionar el plan de pruebas y de facilitar el desarrollo de manuales de usuario.

Finalmente hay que señalar que, por todo lo anterior, los Casos de Uso son mucho más que una simple expresión de los Requisitos Funcionales del sistema.

### 8.2 LIMITACIONES

Los Casos de Uso pueden ser útiles para establecer requisitos de comportamiento, pero no permiten establecer completamente todos los Requisitos (Funcionales, no Funcionales, etc.). Por este motivo, los Casos de Uso deben complementarse con una especificación complementaria que recoja al menos los Requisitos no Funcionales y los especiales como son los Requisitos de Información y las Reglas de Dominio, de ser estos relevantes para el sistema bajo estudio.

---

<sup>46</sup> Recuérdese que el Proceso Unificado de Desarrollo de Software está guiado por Casos de Uso.

## 9 BIBLIOGRAFÍA

Larman, C. (2005). *Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design and Iterative Development*. Prentice-Hall.

Nota de los profesores: Si detectas algún error, piensas que alguna información está presentada de manera confusa, o que sobra o falta algún contenido, por favor, envía un mensaje a [mperez@tel.uva.es](mailto:mperez@tel.uva.es). Gracias por anticipado. Vuestras sugerencias se utilizarán para mejorar esta documentación.

## 10 ANEXO 1: PLANTILLA EN LENGUA INGLESA PARA LA ESPECIFICACIÓN COMPLETA DE UN CASO DE USO

### 10.1 GUIDANCE FOR USE CASE TEMPLATE

Document each use case using the template shown in the Appendix. This section provides a description of each section in the use case template.

### 10.2 USE CASE IDENTIFICATION

#### 10.2.1 Use Case ID

Give each use case a unique integer sequence number identifier. Alternatively, use a hierarchical form: X . Y. Related use cases can be grouped in the hierarchy.

#### 10.2.2 Use Case Name

State a concise, results-oriented name for the use case. These reflect the tasks the user needs to be able to accomplish using the system. Include an action verb and a noun. Some examples:

- View part number information.
- Manually mark hypertext source and establish link to target.
- Place an order for a CD with the updated software version.

### 10.3 USE CASE HISTORY

#### 10.3.1 Created By

Supply the name of the person who initially documented this use case.

#### 10.3.2 Date Created

Enter the date on which the use case was initially documented.

### 10.3.3 Last Updated By

Supply the name of the person who performed the most recent update to the use case description.

### 10.3.4 Date Last Updated

Enter the date on which the use case was most recently updated.

## 10.4 USE CASE DEFINITION

### 10.4.1 Actors

An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor that will be initiating this use case and any other actors who will participate in completing the use case.

### 10.4.2 Trigger

Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.

### 10.4.3 Description

Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.

### 10.4.4 Preconditions

List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition. Examples:

1. User's identity has been authenticated.
2. User's computer has sufficient free memory available to launch task.

### 10.4.5 Postconditions

Describe the state of the system at the conclusion of the use case execution. Number each postcondition. Examples:

1. Document contains only valid SGML tags.<sup>47</sup>
2. Price of item in database has been updated with new value.

### 10.4.6 Normal Flow

Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. This description may be written as an answer to the hypothetical question, *How do I <accomplish the task stated in the use case name>?* This is best done as a numbered list of actions performed by the actor, alternating with responses provided by the system. The normal flow is numbered  $X . 0$ , where  $X$  is the Use Case ID.

### 10.4.7 Alternative Flows

Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative flow and describe any differences in the sequence of steps that take place. Number each alternative flow in the form  $X . Y$ , where  $X$  is the Use Case ID and  $Y$  is a sequence number for the alternative flow. For example,  $5 . 3$  would indicate the third alternative flow for use case number 5.

### 10.4.8 Exceptions

Describe any anticipated error conditions that could occur during execution of the use case and define how the system is to respond to those conditions. Also, describe how the system is to respond if the use case execution fails for some unanticipated reason. If the use case results in a durable state change in a database or the outside world, state whether the change is rolled back, completed correctly, partially completed with a known state, or left in an undetermined state as a result of the exception. Number each alternative flow in the form  $X . Y . E . Z$ , where  $X$  is the Use Case ID,  $Y$  indicates the normal ( $0$ ) or alternative ( $>0$ ) flow during which this exception could take place,  $E$  indicates an exception, and  $Z$  is a sequence number for the exceptions. For example,  $5 . 0 . E . 2$  would indicate the second exception for the normal flow for use case number 5.

### 10.4.9 Includes

List any other use cases that are included (called) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality.

---

<sup>47</sup> SGML (Standard Generalized Markup Language, Lenguaje de Marcado Estándar Generalizado) es un metalenguaje. Los metalenguajes sirven para definir Lenguajes de Marcado (ML, Markup Languages). El vínculo entre un Lenguaje de Marcado y el metalenguaje a partir del cual está definido es un DTD (Document Type Definition, Definición de Tipo de Documento). SGML es el metalenguaje a partir del cual está definido el Lenguaje de Marcas de Hipertexto (HTML, Hypertext Markup Language) que se utiliza para la creación de lo que conocemos como páginas web, aunque existe una reformulación de HTML, denominada XHTML (extensible HTML), como una *aplicación* de XML. El término *aplicación* se refiere a los Lenguajes de Marcado, en ocasiones se les denominada *aplicaciones*, puesto que se crean *aplicando* las normas de un determinado metalenguaje.

#### 10.4.10 Priority

Indicate the relative priority of implementing the functionality required to allow this use case to be executed. The priority scheme used must be the same as that used in the software requirements specification.

#### 10.4.11 Frequency of Use

Estimate the number of times this use case will be performed by the actors per some appropriate unit of time.

#### 10.4.12 Business Rules

List any business rules that influence this use case.

#### 10.4.13 Special Requirements

Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.

#### 10.4.14 Assumptions

List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.

#### 10.4.15 Notes and Issues

List any additional comments about this use case or any remaining open issues or TBDs (To Be Determined) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.

### 10.5 USE CASE LIST

Primary Actor	Use Cases



## 10.6 USE CASE TEMPLATE

Use Case ID:			
Use Case Name:			
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actors:	
Description:	
Trigger:	
Preconditions:	
Postconditions:	
Normal Flow:	
Alternative Flows:	
Exceptions:	
Includes:	
Priority:	
Frequency of Use:	
Business Rules:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

## 10.7 REVISION HISTORY

Name	Date	Reason For Changes	Version