DEVELOPMENT
DESIGN
ANALYZE
IMPLEMENT

CASO PRÁCTICO

En *GoyoProg S.L.* continúan inmersos en el mundo de UML. A pesar de que han trabajado duro y han aprendido bastante acerca de este lenguaje de especificación, Ada se ha dado cuenta de que apenas han empezado a arañar la superficie de todas las posibilidades que les ofrece. De momento ya saben cómo crear un diagrama de clases y como analizar un problema propuesto, sin embargo, hay muchos aspectos del problema que no pueden modelar todavía, por ejemplo, con solo el diagrama de clases no pueden saber qué se espera del sistema que van a construir, o en qué se deben basar para codificar los métodos, o simplemente, ¿Cómo colaboran los objetos de las clases que han creado para hacer alguna tarea que sea útil?

El equipo decide que no pueden parar ahora, y que hay que hacer un esfuerzo final para que los conocimientos del equipo sean globales y puedan enfrentarse a cualquier desarrollo software con solvencia.

Tema5. Diagramas de Comportamiento. Parte I.

1. Clasificación de los Diagramas de Comportamiento

En el tema anterior vimos cómo crear un diagrama de clases para un problema determinado, esto nos ayuda a ver el problema con otra perspectiva y descubrir información nueva, sin embargo, no tiene en cuenta elementos como la creación y destrucción de objetos, el paso de mensajes entre ellos y el orden en que deben hacerse, qué funcionalidad espera un usuario poder realizar, o cómo influyen elementos externos en nuestro sistema. Un **diagrama de clases** nos **da información estática** pero no dice nada acerca del **comportamiento dinámico** de los objetos que lo forman, para incluir este tipo de información **utilizamos los diagramas de comportamiento** que incluyen:

- ✓ Diagramas de casos de uso.
- ✓ Diagramas de actividad.
- ✓ Diagramas de estados.
- ✓ Diagramas de interacción.
    - Diagramas de secuencia.
    - Diagramas de colaboración o comunicación.

2. Diagramas de Casos de Uso

Los diagramas de casos de uso son un elemento fundamental del análisis de un sistema desde la perspectiva de la orientación a objetos porque resuelven uno de los principales problemas en los que se ve envuelto el proceso de producción de software: la falta de comunicación entre el equipo de desarrollo y el equipo que necesita de una solución software. Un diagrama de casos de uso nos ayuda a determinar **QUÉ** puede hacer cada tipo diferente de usuario con el sistema, en una forma que los no versados en el mundo de la informática o, más concretamente el desarrollo de software, pueda entender.

Los diagramas de casos de **uso documentan el comportamiento de un sistema desde el punto de vista del usuario.** Por lo tanto, los casos de uso determinan los **requisitos funcionales del sistema** (acciones fundamentales que debe realizar el software al recibir información, procesarla y producir resultados. Suelen venir definidos por el cliente), es decir, representan las funciones que un sistema puede ejecutar.

Un diagrama de casos de uso es una visualización gráfica **de los requisitos funcionales del sistema, que está formado por casos de uso (se representan como elipses) y los actores que interactúan con ellos (se representan como monigotes).** Su principal **función** es dirigir el proceso de creación del software, definiendo

qué se espera de él, y su **ventaja** principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el cliente.

Los diagramas de casos de uso se crean en la primera etapa de desarrollo del software, y se enmarcan en el proceso de análisis, para definir de forma detallada la funcionalidad que se espera cumpla el software, y que, además, se pueda comunicar fácilmente al usuario, pero ¿termina aquí su función?

En absoluto, de los diagramas de casos de uso se desprenden otros (normalmente se realiza antes que el diagrama de clases) que **describen tanto la estructura del sistema como su comportamiento**, lo que influye directamente en la implementación (Paso a código de las especificaciones que se han definido durante la fase de análisis y diseño de un sistema software) del sistema y en su arquitectura (Conjunto de decisiones significativas acerca de la organización de un sistema software, la selección de los elementos estructurales a partir de los cuales se compone el sistema, y las interfaces entre ellos, junto con su comportamiento, tal y como se especifica en las colaboraciones entre esos elementos, la composición de estos elementos estructurales y de comportamiento en subsistemas progresivamente mayores y el estilo arquitectónico que guía esta organización: estos elementos y sus interfaces, sus colaboraciones y su composición) final. Por otra parte, al describir específicamente qué se espera del software también se usa en la **fase de pruebas**, para verificar que el sistema cumple con los requisitos funcionales, creándose muchos de los casos de prueba (***Pruebas de caja negra*** (Se realizan cuando una aplicación es probada usando su interfaz externa sin preocuparnos de la implementación de la misma) directamente a partir de los casos de uso.

2.1. Actores

Los **actores** representan un **tipo** de **usuario del sistema**. Se entiende como usuario **cualquier cosa externa que interactúa con el sistema**. No tiene por qué ser un humano, puede ser otro sistema informático o unidades organizativas o empresas.

Siempre hay que intentar independizar los actores de la forma en que se interactúa con el sistema. Por ejemplo, un usuario del sistema puede interpretar diferentes roles según la operación que esté ejecutando, cada uno de estos roles representará un actor diferente, es decir, un actor en un diagrama de casos de uso representa un rol que alguien puede estar jugando, no un individuo particular, por lo tanto puede hacer personas particulares que puedan estar usando el sistema de formas diferente en diferentes ocasiones. Suele ser útil mantener una lista de los usuarios reales para cada actor.

Tipos de Actores:

- ✓ **Primarios**: Interaccionan con el sistema para explotar su funcionalidad. Trabajan directa y frecuentemente con el software.
- ✓ **Secundarios**: Soporte del sistema para que los primarios puedan trabajar. Son precisos para alcanzar algún objetivo.
- ✓ **Iniciadores**: No interactúan con el sistema, pero desencadenan el trabajo de otro actor.

Los actores se representan mediante la siguiente figura.

Es posible que haya casos de uso que no sean iniciados por ningún usuario, o algún otro elemento software, en ese caso se puede crear un actor 'Tiempo' o 'Sistema'.

2.2. Casos de uso

Un **caso de uso** especifica una **secuencia de acciones, incluyendo variantes, que el sistema puede llevar a cabo, y que producen un resultado observable de valor para un actor concreto**. El conjunto de casos de uso forma el "**comportamiento requerido**" de un sistema.

El objetivo principal de elaborar un diagrama de casos de uso no es crear el diagrama en sí, sino la descripción que de cada caso se debe realizar, ya que esto es lo que ayuda al equipo de desarrollo a crear el sistema a posteriori. Para hacer esto utilizamos, sobre todo otros diagramas que permiten describir la dinámica del caso de uso, como el diagrama de secuencia que veremos después, y una descripción textual, en la que se deben incluir, al menos, los siguientes datos (a los que se denomina **contrato**):

- ✓ **Nombre:** nombre del caso de uso.
- ✓ **Actores**: aquellos que interactúan con el sistema a través del caso de uso.
- ✓ **Propósito**: breve descripción de lo que se espera que haga.
- ✓ **Precondiciones**: aquellas que deben cumplirse para que pueda llevarse a cabo el caso de uso.
- ✓ **Flujo normal**: flujo normal de eventos que deben cumplirse para ejecutar el caso de uso exitosamente, desde el punto de vista del actor que participa y del sistema.
- ✓ **Flujo alternativo**: flujo de eventos que se llevan a cabo cuando se producen casos inesperados o poco frecuentes. **No** se deben incluir aquí errores como escribir un tipo de dato incorrecto o la omisión de un parámetro necesario.
- ✓ **Postcondiciones**: las que se cumplen una vez que se ha realizado el caso de uso.

La **representación gráfica** de un caso de uso se realiza mediante un óvalo o elipse, y su descripción se suele hacer rellenando una o más tablas.

2.3. Relaciones

Los diagramas de casos de uso son **grafos no conexos** en los que los nodos son actores y casos de uso, y las aristas son las relaciones que existen entre ellos. Representan qué actores realizan las tareas descritas en los casos de uso, en concreto **qué actores inician un caso de uso**. Pero además existen otros tipos de relaciones que se utilizan para especificar relaciones más complejas, como uso o herencia entre casos de uso o actores.

Existen diferentes tipos de relaciones entre elementos:

- ✓ **Asociación**: representa la relación entre el actor que lo inicia y el caso de uso.
- ✓ **Inclusión**: se utiliza cuando queremos dividir una tarea de mayor envergadura en otras más sencillas, que son utilizadas por la primera. Representa una relación de uso, y son muy útiles cuando es necesario reutilizar tareas.
- ✓ **Extensión**: se utiliza para representar relaciones entre un caso de uso que requiere la ejecución de otro en determinadas circunstancias.
- ✓ **Generalización**: se utiliza para representar relaciones de herencia entre casos de uso o actores.

A continuación, vemos este tipo de relaciones con un poco más de detalle.

2.3.1. Interacción o Asociación

Hay una asociación entre un actor y un caso de uso si el actor interactúa con el sistema para llevar a cabo el caso de uso o para iniciarlo.

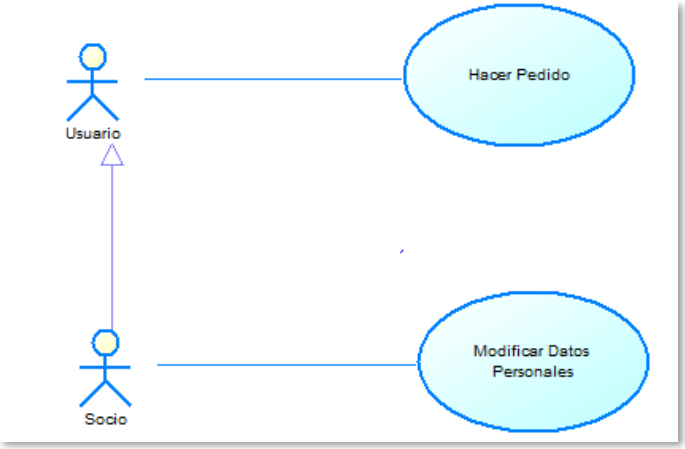
Una asociación se representa mediante una línea continua que une un actor con un caso de uso. Por ejemplo, un usuario de un sistema de venta por Internet puede hacer un pedido, lo que se representa del siguiente modo:

2.3.2. Generalización

Es posible que, igual que con los diagramas de clases, existan casos de uso que tengan comportamientos semejantes a otros que los modifican o completan de alguna manera. **El caso base se define de forma abstracta y los hijos heredan sus características** añadiendo sus propios pasos o modificando alguno. Normalmente **la herencia se utiliza menos en diagramas de casos de uso** que en diagramas de clases.

Por ejemplo, el usuario del sistema de venta por Internet puede a su vez darse de alta en la página web para que tengan sus datos registrados a la hora de hacer el pedido, en este caso el usuario es la generalización del socio.

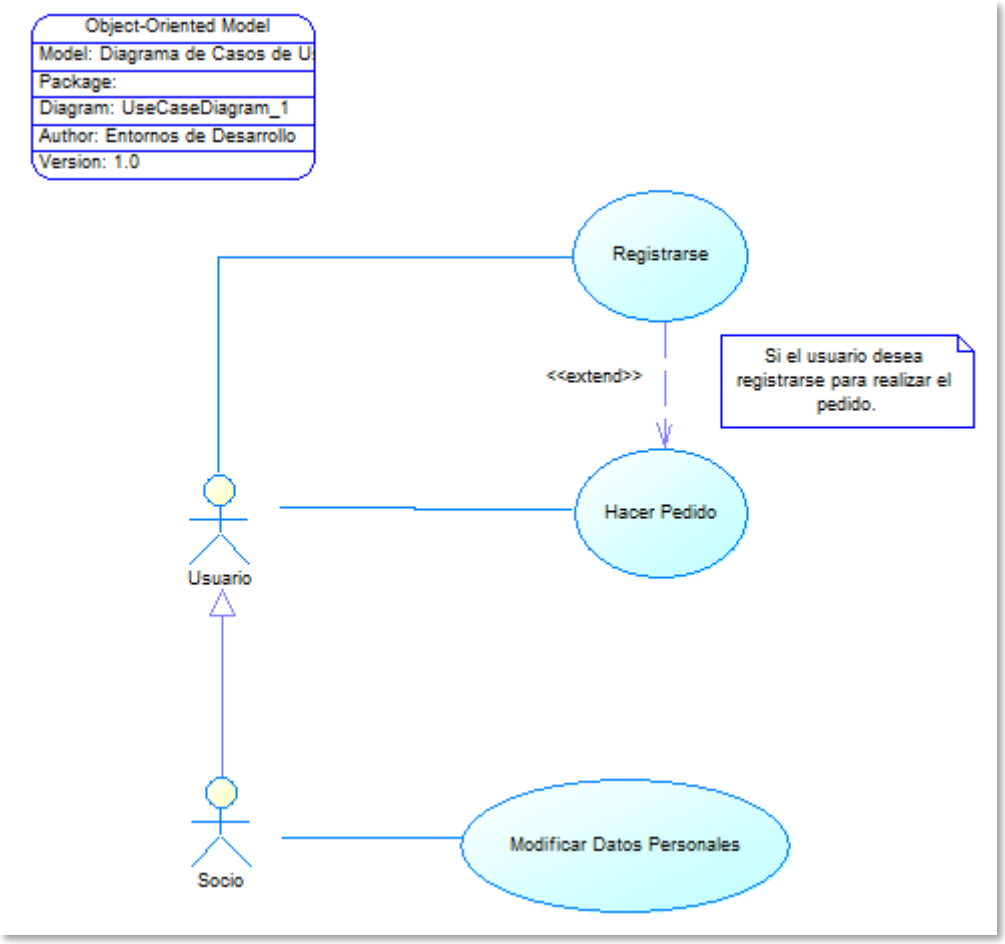
Ambos actores pueden hacer un pedido, pero solo el socio puede modificar sus datos en el sistema.

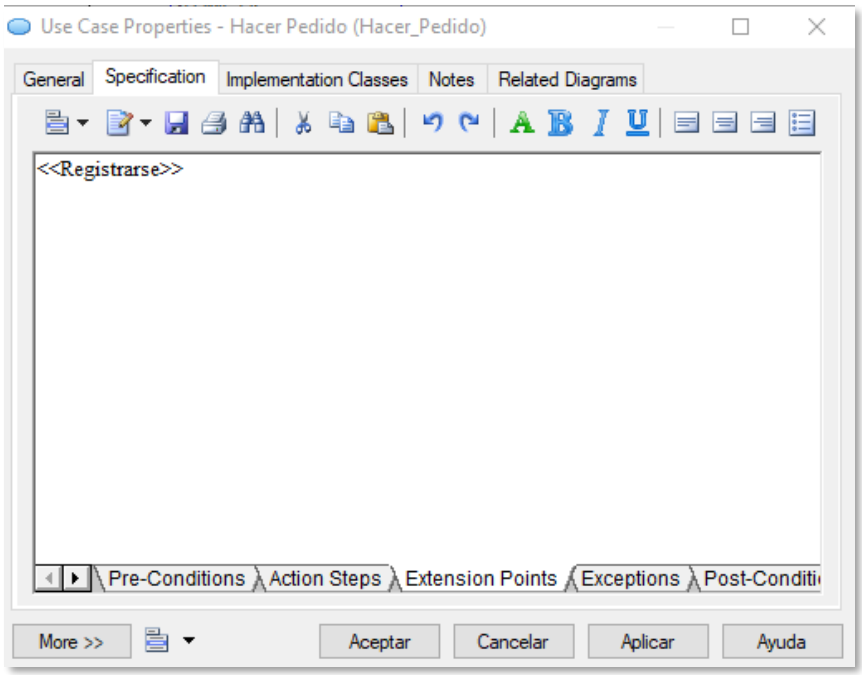


2.3.3. Extensión

Se utiliza una relación entre dos casos de uso de tipo "**extends**" cuando se desea especificar que el comportamiento de un caso de uso es diferente dependiendo de ciertas circunstancias.

La principal función de esta relación es simplificar el flujo de casos de uso complejos. **Se utiliza cuando existe una parte del caso de uso que se ejecuta sólo en determinadas ocasiones, pero no es imprescindible para su completa ejecución.** Cuando un caso de uso extendido se ejecuta, se indica en la especificación del caso de uso como un **punto de extensión**. Los puntos de extensión se pueden mostrar en el diagrama de casos de uso.





Por ejemplo, cuando un usuario hace un pedido, si no es socio se le ofrece la posibilidad de darse de alta en el sistema en ese momento, pero puede realizar el pedido aunque no lo sea.

2.3.4. Inclusión

Se incluye una relación entre dos casos de uso de tipo "**include**" cuando la ejecución del caso de uso incluido se da en la rutina normal del caso que lo incluye.

Esta relación es muy útil cuando se desea especificar algún comportamiento común en dos o más casos de uso, aunque es frecuente cometer el error de utilizar esta técnica para hacer subdivisión de funciones, por lo que se debe tener mucho cuidado cuando se utilice.

Por ejemplo, a la hora de hacer un pedido se debe buscar la información de los artículos para obtener el precio, es un proceso que necesariamente forma parte del caso de uso, sin embargo, también forma parte de otros, como son el que visualiza el catálogo de productos y la búsqueda de un artículo concreto, y dado que tiene entidad por sí solo se separa del resto de casos de uso y se incluye en los otros tres.

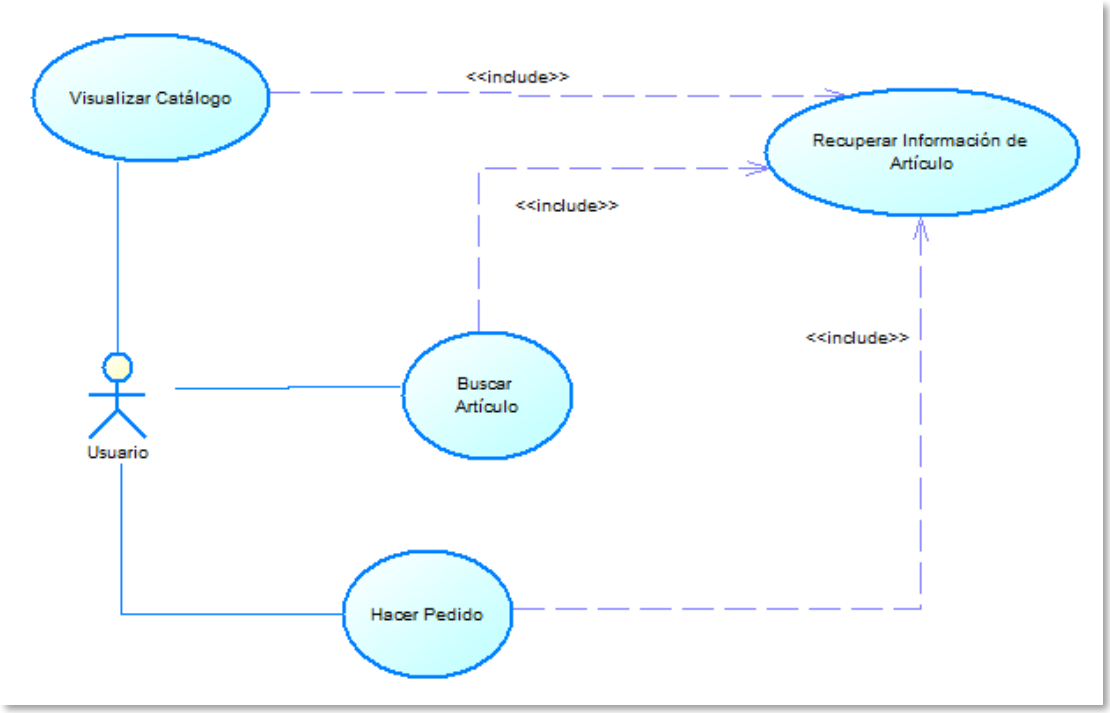
Las **ventajas** de esta asociación son:

- ✓ Las descripciones de los casos de uso son más cortas y se entienden mejor.
- ✓ La identificación de funcionalidad común puede ayudar a descubrir el posible uso de componentes ya existentes en la implementación.

Las **desventajas** son:

- ✓ La inclusión de estas relaciones hace que los diagramas sean más difíciles de leer, sobre todo para los clientes.

Cuando usamos relaciones de inclusión o extensión no podemos olvidar que los casos de uso extendidos o incluidos deben cumplir con las características propias de un caso de uso, es decir, deben representar un flujo de actividad completo desde el punto de vista de lo que un actor espera que el sistema haga por él, así como no utilizar estas herramientas sólo para descomponer un caso de uso de envergadura en otros más pequeños, piedra angular del diseño estructurado y no del orientado a objetos.

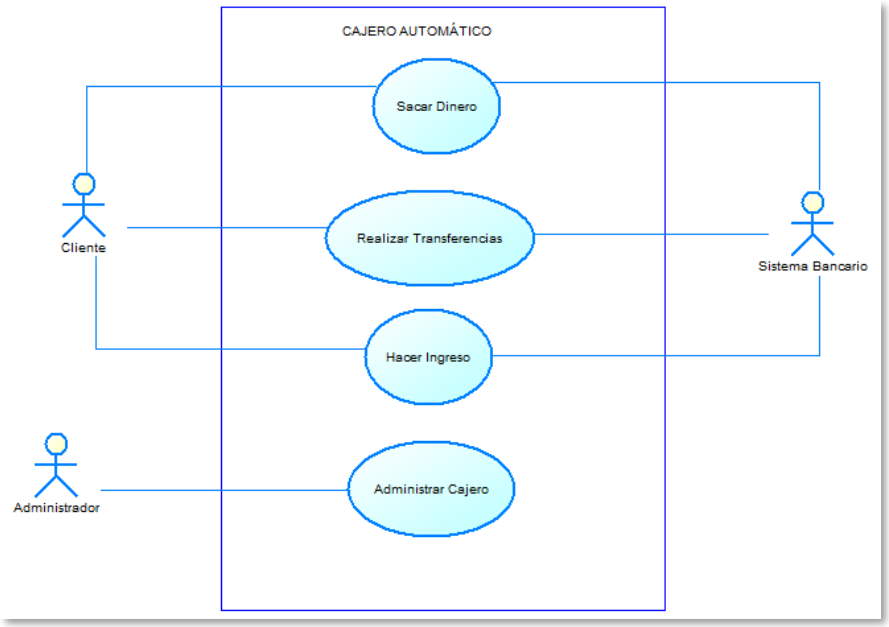


2.4. Elaboración de Casos de Uso

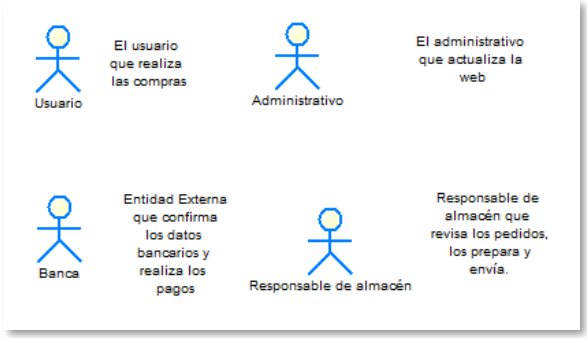
Sea cual sea el tipo de diagrama que estemos creando, cuando lo hacemos realizamos un proceso de abstracción por el cual representamos elementos de la realidad esquemáticamente, y **en el diagrama de casos de uso** pasa igual, **necesitamos abstraer la realidad en un diagrama, en el que representamos qué cosas pueden hacerse en nuestro sistema y quien las va a hacer**. Necesitamos diagramas que incluyan suficiente información para que el equipo de desarrollo tome las decisiones más adecuadas en la fase de análisis y diseño para una construcción de software que cumpla con los requerimientos, así como que sean útiles en la fase de implementación en un lenguaje orientado a objetos.

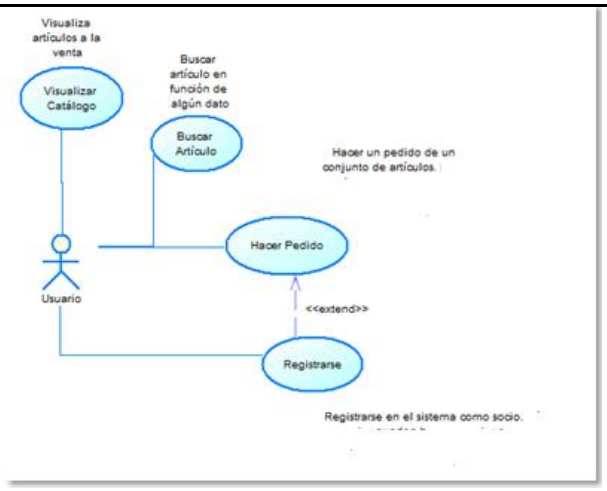
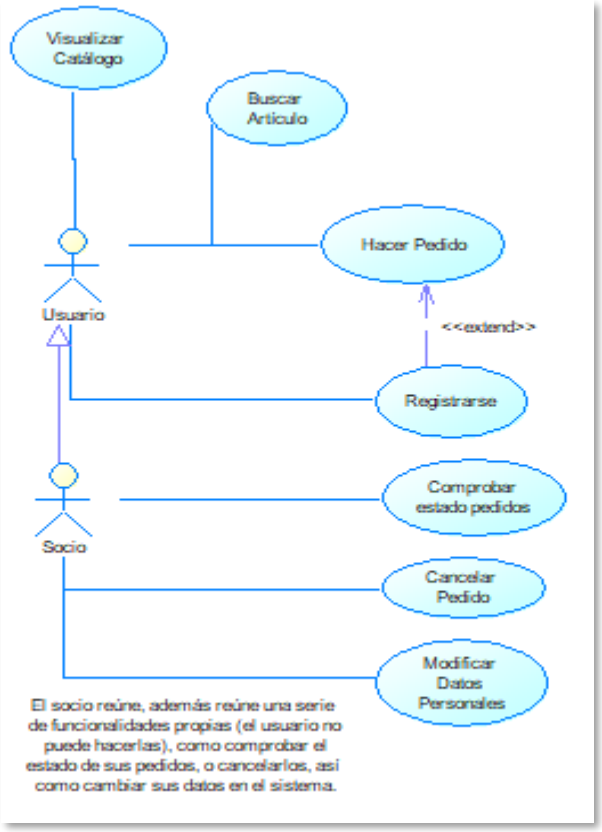
Partiremos de una descripción lo más detallada posible del problema a resolver y trataremos de detectar quien interactúa con el sistema, para obtener los actores diagrama de casos de uso, a continuación, buscaremos qué tareas realizan estos actores para determinar los casos de uso más genéricos. El siguiente paso es refinar el diagrama analizando los casos de uso más generales para detectar casos relacionados por inclusión (se detectan fácilmente cuando aparecen en dos o más casos de uso generales), extensión y generalización. Al diagrama generado se le denomina ***diagrama frontera***.

Se conoce como **diagrama frontera** al diagrama de casos de uso que incluye todos los casos de uso genéricos del sistema, que podrán ser desglosados después en nuevos diagramas de casos de uso que los describan si es necesario. Se especifica enmarcando los casos de uso en un recuadro, que deja a los actores fuera.

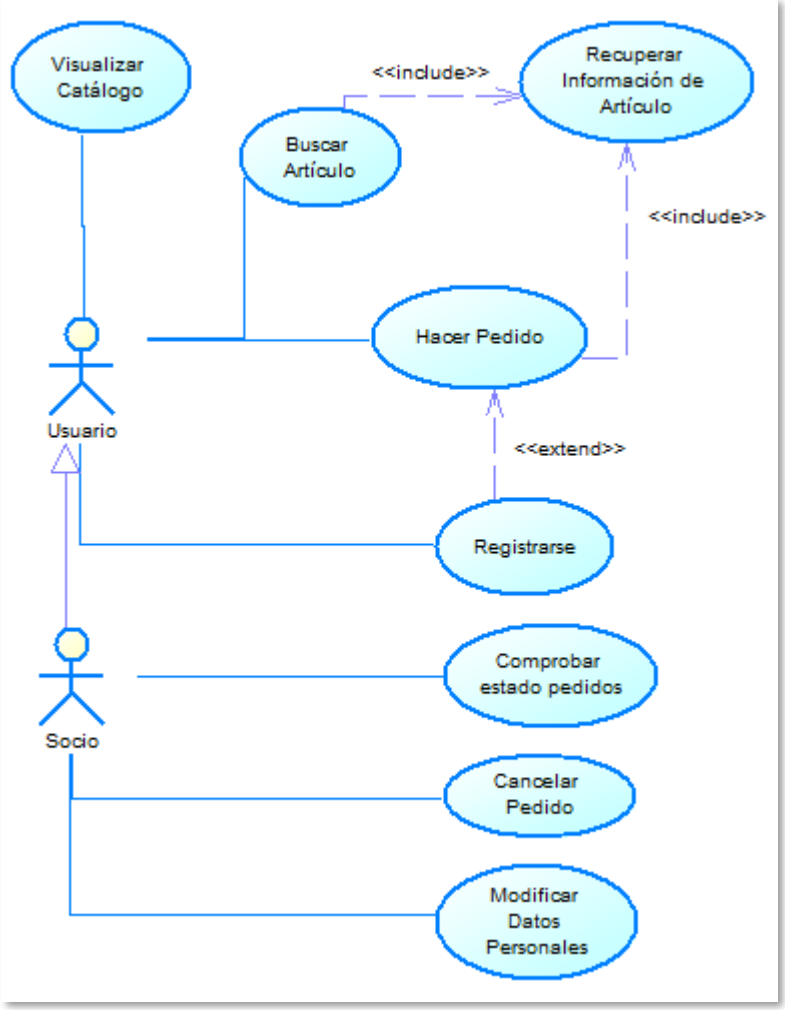


Ejemplo de Diagrama de Casos de Uso.

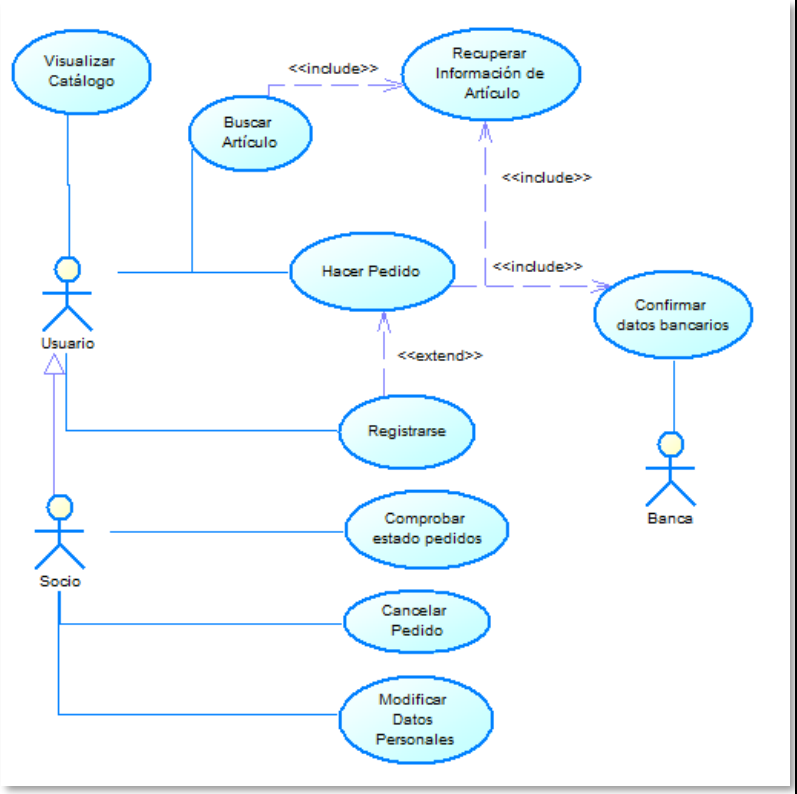
CREACIÓN DE CASOS DE USO	
Primeros pasos	Antes de elaborar el diagrama tendremos que **leer con detenimiento el documento con la especificación de requisitos y del problema a resolver y asegurarnos de que entendemos la idea central del problema**: crear una tienda virtual en la que se puedan realizar pedidos de los productos a la venta (zapatos). El proceso se centra en el pedido, desde poner a disposición del cliente los artículos en venta, pasando por la selección de artículos a pedir, la cumplimentación de toda la información necesaria para el pedido, pago, confección del pedido, envío y reajuste del stock en almacén, todo ello, a través de la web
Identificar actores	En nuestro sistema ¿Quién interviene? 
Identificar funcionalidades	Para facilitar la creación del diagrama vamos a ir sacando funcionalidades para cada usuario. Debemos recordar que un caso de uso representa una interacción de un actor con el sistema, que está **relacionado con los requisitos funcionales de la aplicación final** y que, en definitiva representa tareas que llevará a cabo el sistema.
Funcionalidad del usuario	Cuando una persona se conecta al sistema lo primero que podrá hacer será **visualizar** el catálogo de la temporada. También puede hacer un **pedido** con uno o varios artículos del catálogo, para ello visualizará los artículos de forma que pueda seleccionar algunos de ellos e indicar la cantidad que quiere comprar. También puede hacer **búsquedas** por datos concretos de artículos. Cualquier persona que acceda al sistema puede **darse de alta** para ser socio. Así mismo, si es socio, podrá **comprobar el estado** de sus pedidos y **cancelarlos**.

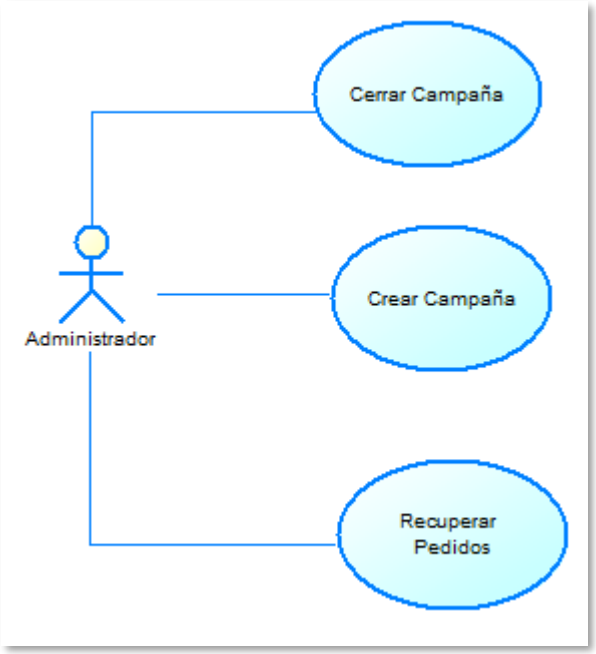
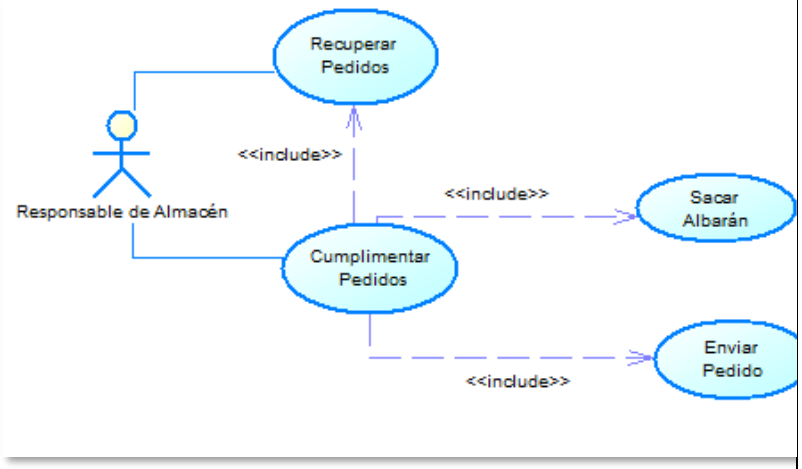
Un Usuario como tal, puede:	
Al hacerse socio la identidad del usuario con respecto del sistema cambia, por lo que surge un nuevo tipo de actor que hereda de usuario. Un socio es un usuario que se ha registrado en el sistema. Los socios almacenan cierta información, como sus datos personales y bancarios, email, etc.	

Al revisar un poco la funcionalidad de los casos de uso del usuario podemos comprobar que en los casos **Buscar articulo** y **Hacer pedido** es necesario buscar en el sistema y recuperar la información de un artículo del que tenemos algún dato, en el primer caso para obtener todos los datos del artículo buscado y en el segundo para recuperar el precio de los artículos que se añaden al pedido, por lo que extraemos el caso de uso "**Recuperar información de artículo**" que se incluye en los otros dos.



Cuando se realiza el pedido es obligatorio hacer una comprobación de los datos bancarios del cliente, que dependen de una entidad externa, por lo que se añade otro caso de uso para esta función, "*Confirmar datos bancarios*" que trae de la mano la inclusión del actor Banca.



Funcionalidad del administrador web	El objetivo del administrador web es gestionar los contenidos de la web, en concreto de las diferentes campañas, ya que cada temporada se debe cerrar la campaña antigua, retirando los artículos de la temporada anterior y abrir la temporada nueva, añadiendo sus artículos. Para que se pueda cerrar una temporada es necesario que en el almacén se hayan gestionado todos sus pedidos, por lo que es obligatorio comprobarlo, antes de cerrar.
Se incluye el caso de uso **Recuperar pedidos** en **Crear campaña** por dos motivos: ✓ Es una función que puede llevarse a cabo de forma independiente, tanto por el administrador de la web como desde el almacén. ✓ Es de comprobación obligatoria antes de cerrar la temporada.	
Funcionalidad del responsable de almacén	Es el encargado de leer los pedidos de los usuarios y cumplimentarlos. Ésta será su única función, si bien, es una función complicada, ya que implica realizar una serie de tareas: ✓ Seleccionar el pedido más antiguo. ✓ Buscar los artículos a servir. ✓ Empaquetarlos junto con un albarán para el socio. ✓ Colocarlos en su ruta de envío
El primer paso es **recuperar la lista de pedidos sin procesar.** Esta operación recupera el proceso más antiguo para ser procesado. **Sacar albarán** produce un listado en papel con información del pedido para el socio. **Enviar pedido** es colocar el pedido en la ruta de envío más apropiada para su destino.	

Especificación del problema a modelar para una tienda *e-commerce*:

Los usuarios del sistema navegan por la web para ver artículos, zapatos, bolsos y complementos que se venden en la tienda. De los artículos nos interesa su nombre, descripción, material, color, precio y stock. De los zapatos nos interesa su número y el tipo. De los bolsos nos interesa su tipo (bandolera, mochila, fiesta). De los complementos (cinturones y guantes) su talla. Los artículos se organizan por campañas para cada temporada (primavera/verano y otoño/invierno) de cada año.

Los artículos son de fabricación propia, pero opcionalmente pueden venderse artículos de otras firmas. De las firmas nos interesa saber su nombre, CIF y domicilio fiscal. La venta de artículos de firma se realiza a través de proveedores, de forma que un proveedor puede llevar varios artículos de diferentes firmas, y una firma puede ser suministrada por más de un proveedor. Los artículos pertenecen a una firma solamente. De los proveedores debemos conocer su nombre, CIF, y domicilio fiscal. Los usuarios pueden registrarse en el sitio web para hacerse socio. Para ello debe proporcionar los siguientes datos: nombre completo, correo electrónico y dirección.

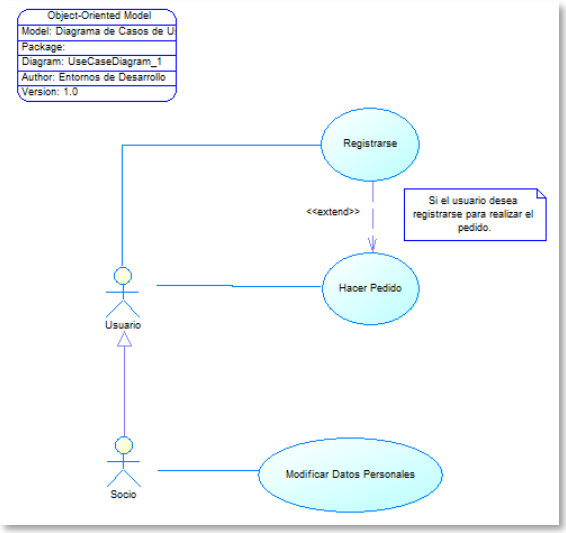
Los socios pueden hacer pedidos de los artículos. Un pedido está formado por un conjunto de detalles de pedido que son parejas formadas por artículo y la cantidad. De los pedidos interesa saber la fecha en la que se realizó y cuánto debe pagar el socio en total. El pago se hace a través tarjeta bancaria, cuando se va a pagar una entidad bancaria comprueba la validez de la tarjeta. De la tarjeta interesa conocer el número. Las campañas son gestionadas por el administrativo que se encargará de dar de baja la campaña anterior y dar de alta la nueva siempre que no haya ningún pedido pendiente de cumplimentar. Existe un empleado de almacén que revisa los pedidos a diario y los cumplimenta. Esto consiste en recopilar los artículos que aparecen en el pedido y empaquetarlos. Cuando el paquete está listo se pasa al almacén a la espera de ser repartido.

Del reparto se encarga una empresa de transportes que tiene varias rutas preestablecidas. Según el destino del paquete (la dirección del socio) se asigna a una u otra ruta. De la empresa de transportes se debe conocer su nombre, CIF y domicilio fiscal. Las rutas tienen un área de influencia que determina los destinos, y unos días de reparto asignados. Se debe conocer la fecha en la que se reparte el pedido. Si se produce alguna incidencia durante el reparto de algún pedido se almacena la fecha en la que se ha producido y una descripción. Los socios pueden visualizar sus pedidos y cancelarlos siempre y cuando no hayan sido cumplimentados por el usuario de almacén. Así mismo puede modificar sus datos personales.

Documentación del caso de Uso “Hacer Pedido”

Como se indicaba en los contenidos del apartado anterior, la parte más importante de la elaboración de un diagrama de casos de uso es la documentación de los propios casos de uso, lo que permitirá desarrollar otros diagramas que ayuden en la codificación del sistema, y la elaboración de los casos de prueba de caja negra. A modo de ejemplo vamos a desarrollar la documentación del caso de uso **Hacer Pedido**, ya que, por su complejidad abarca todos los apartados que hemos visto.

Recordamos el aspecto del caso de uso:

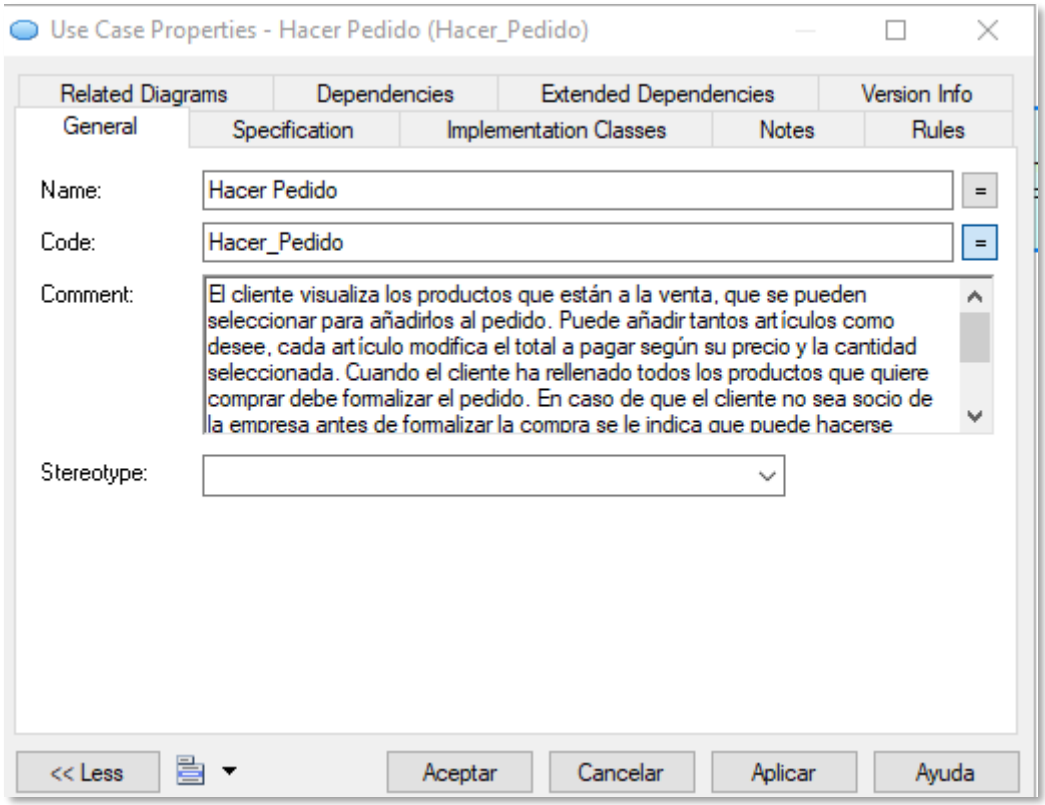


Los datos que debemos incluir para elaborar la documentación del caso de uso eran:

- ✓ **Nombre**: Nombre del caso de uso.
- ✓ **Actores**: aquellos que interactúan con el sistema a través del caso de uso.
- ✓ **Propósito**: breve descripción de lo que se espera que haga.
- ✓ **Precondiciones**: aquellas que deben cumplirse para que pueda llevarse a cabo el caso de uso.
- ✓ **Flujo normal**: flujo normal de eventos que deben cumplirse para ejecutar el caso de uso exitosamente.
- ✓ **Flujo alternativo**: flujo de eventos que se llevan a cabo cuando se producen casos inesperados o poco frecuente. No se deben incluir errores de escribir un tipo de dato incorrecto o la omisión de un parámetro necesario.
- ✓ **Postcondiciones**: las que se cumplen una vez que se ha realizado el caso de uso. Para incluir el nombre, actores, propósito, pre y post condiciones, abrimos la especificación del caso de uso. Esto da lugar a la aparición de una ventana con un conjunto de pestañas que podemos rellenar:

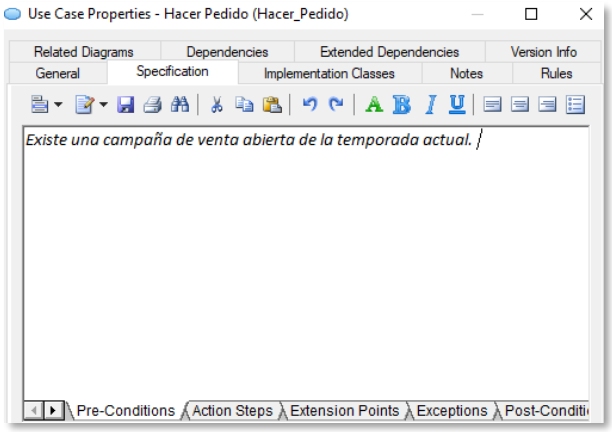
En la pestaña "**General**" rellenamos el nombre "**Hacer pedido**", y tenemos un espacio para escribir un breve comentario, como por ejemplo:

"El cliente visualiza los productos que están a la venta, que se pueden seleccionar para añadirlos al pedido. Puede añadir tantos artículos como desee, cada artículo modifica el total a pagar según su precio y la cantidad seleccionada. Cuando el cliente ha rellenado todos los productos que quiere comprar debe formalizar el pedido. En caso de que el cliente no sea socio de la empresa antes de formalizar la compra se le indica que puede hacerse socio, si el cliente acepta se abre el formulario de alta, en caso contrario se cancela el pedido. En caso de que se produzca algún problema con los datos bancarios se ofrecerá la posibilidad de volver a introducirlos. Al finalizar un pedido se añade al sistema con el estado pendiente."

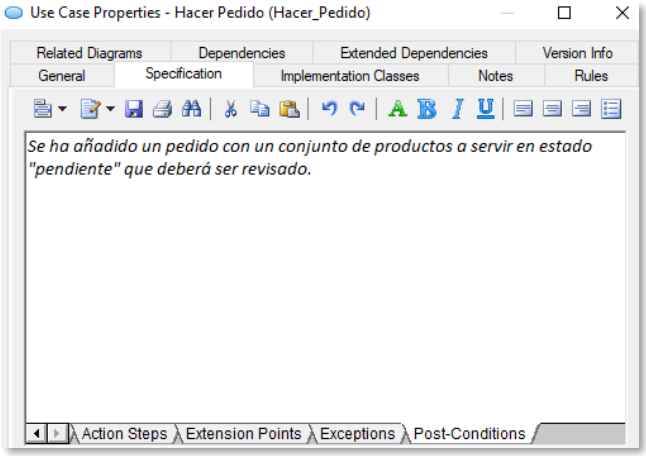


En la pestaña "**Especificación"** encontramos unos campos predefinidos entre los que se encuentran precondiciones y postcondiciones, que podemos rellenar de la siguiente manera:

Precondiciones: *Existe una campaña de venta abierta de la temporada actual.*



Postcondiciones: *Se ha añadido un pedido con un conjunto de productos a servir en estado "pendiente" que deberá ser revisado.*



Dentro de la misma pestaña '**Especificación'**, en la pestaña inferior de ***'Action Steps'***, deberemos describir el flujo normal de eventos que se producen cuando se ejecuta el caso de uso sin ningún problema. Lo haremos de la siguiente forma, resumiéndolo dentro de una tabla.

Use Case Properties - Hacer Pedido (Hacer_Pedido)

Dependencies | Extended Dependencies | Version Info
General | Specification | Implementation Classes | Notes | Rules | Related Diagrams

	Actor Input	System Response
1	Inicia el pedido	
2		Se crea un pedido en estado 'en construcción'
3	Selecciona un articulo	
4	Selecciona la cantidad	
5		Recupera la información del articulo para obtener el precio y modifica el precio total del pedido.
6	El proceso anterior se repite hasta completar la lista de articulos.	
7	Se acepta el pedido.	
8		Se comprueba si el usuario es socio, si no lo es se le muestra un aviso para que se registre en el sitio.
9		Se comprueban los datos bancarios con una entidad externa.
10		Se calcula el total, sumando los gastos de envio.
11		Se realiza el pago a través de la entidad externa.
12		Se almacena la información del pedido en el estado 'pendiente'.

Pre-Conditions | Action Steps | Extension Points | Exceptions | Post-Conditions

Añadiremos un par de descripciones alternativas para indicar que hacer cuando el usuario no es socio y cuando los datos bancarios no son correctos.

Flujo de caso de uso Alternativo para cuando el usuario no está registrado:

	Actor Input	System Response
1	Inicia el pedido	
2		Se crea un pedido en estado 'en construcción'
3	Selecciona un articulo	
4	Selecciona la cantidad	
5		Recupera la información del articulo para obtener el precio y modifica el precio total del pedido.
6	El proceso anterior se repite hasta completar la lista de articulos.	
7	Se acepta el pedido.	
8		Se comprueba si el usuario es socio, si no lo es se le muestra un aviso para que se registre en el sitio.
9		Se comprueban los datos bancarios con una entidad externa.
10		Se calcula el total, sumando los gastos de envio.
11		Se realiza el pago a través de la entidad externa.
12		Se almacena la información del pedido en el estado 'pendiente'.

Pre-Conditions | Action Steps | Extension Points | Exceptions | Post-Conditions

Flujo de caso de uso Alternativo para cuando los datos bancarios no son correctos:

	Actor Input	System Response
1	Inicia el pedido.	
2		Se crea un pedido en estado 'en construcción'.
3	Selecciona un articulo.	
4	Selecciona la cantidad.	
5		Recupera la información del articulo para obtener el precio y modifica el precio total del pedido.
6		Añade la información al pedido en creación.
7	El proceso anterior se repite hasta completar la lista de articulos.	
8	Se acepta el pedido.	
9		Se comprueban los datos bancarios con una entidad externa, fallando la comprobación.
10		Se solicitan los datos de nuevo.
11	Introduce los datos de nuevo.	
12		Se repiten el proceso hasta que se acepten los datos bancarios o se cancele la operación.
13		Se calcula el total, sumando los gastos de envio.
14		Se realizar el pago a través de la entidad externa.
15		Se almacena la información del pedido con el estado 'pendiente'.

Pre-Conditions | Action Steps | Extension Points | Exceptions | Post-Conditions

2.5. Escenarios

Un caso de Uso debe especificar un comportamiento deseado, pero no imponer como se llevará a cabo ese comportamiento, es decir, debe decir QUÉ pero no CÓMO. Esto se realiza utilizando escenarios que son casos particulares de un caso de uso.

Un **escenario** es una ejecución particular de un caso de uso que se describe como una secuencia de eventos.

Un caso de uso es una generalización de un escenario.

Por ejemplo, para el caso de uso hacer pedido podemos establecer diferentes escenarios:

Un posible escenario podría ser:

Realizar pedido de unos zapatos y unas botas.

1. El usuario inicia el pedido.
2. Se crea el pedido en estado "en construcción".

3. Se selecciona un par de zapatos “Lucía” de piel negros, del número 39.
4. Se selecciona la cantidad 1.
5. Se recupera la información de los zapatos y se modifica la cantidad a pagar sumándole 45 €.
6. Se selecciona un par de botas “Aymara” de ante marrón del número 40.
7. Se selecciona la cantidad 1.
8. Se recupera la información de las botas y se modifica la cantidad a pagar sumándole 135€.
9. El usuario acepta el pedido.
10. Se comprueba que el usuario es socio.
11. Se comprueban los datos bancarios, que son correctos.
12. Se calcula el total a pagar añadiendo los gastos de envío.
13. Se realiza el pago a través de una entidad externa.
14. Se genera un pedido para el usuario con los dos pares de zapatos que ha comprado, con el estado “pendiente”.

Los **escenarios** pueden y deben posteriormente **documentarse** mediante **diagramas de secuencia**, como veremos a continuación.