

Capítulo X

Diagramas de Use Case

Diagramas de Use Case

Tabla de contenido

1.-	¿Qué es un use case?	149
2.-	Diagramas de use case	150
3.-	Relaciones	152
3.1.-	Inclusión/Utilización	153
3.2.-	Extensión	154
3.3.-	¿Cuándo incluye y cuándo extiende?	154
4.-	Desarrollo de use cases	156
5.-	Definición de escenarios	159
6.-	Consideraciones de tipo práctico	160
7.-	Utilización de la técnica de use case	161
8.-	Ejemplos	163
8.1.-	Actualizar el catálogo de productos.....	166
8.1.1.-	Crear un renglón en el catálogo de productos.....	167
8.1.2.-	Modificar un renglón en el catálogo de productos	168
8.1.3.-	Desactivar un renglón en el catálogo de productos.....	169
8.2.-	Actualizar precios y ofertas	169
8.3.-	Modificar/eliminar un pedido.....	170
8.4.-	Modificar/eliminar un pedido (versión larga).....	171
9.-	Nomenclatura utilizada en UML	174

Diagramas de Use Case

1.- ¿Qué es un use case?

Un use case es una descripción del funcionamiento de un proceso, actividad o tarea. Muchos autores traducen el término use case como *Caso de Uso*, nosotros hemos preferido mantener la denominación en inglés, con el fin de evitar cualquier confusión.

Los autores Hans-Erik Eriksson y Magnus Pender, en su libro *UML Toolkit*, definen un use case como:

... un conjunto de secuencias de acciones que ejecuta un sistema, para producir un resultado observable, de valor para algún actor en particular ...

Estas acciones pueden ser de cualquier naturaleza, tareas o comunicación con otros actores, tanto usuarios como sistemas.

Un use case describe un proceso, actividad o tarea, en términos de:

- El nombre que lo distingue
- Las secuencias de pasos o flujos de tareas que lo conforman.
- Las condiciones previas y posteriores que “inician” o “finalizan” el caso.

Los autores Geri Schneider y Jason P. Winters, en su obra *Applying Use Cases* (Aplicando Use Cases) señalan que:

Los use cases se utilizan para describir los requerimientos de un sistema que se observan desde su exterior. Se utilizan en la fase de análisis de requerimientos de un proyecto y contribuyen en la elaboración de los planes de prueba y las guías de usuario. Se utilizan para crear y validar una propuesta de diseño y asegurar que se han incluido todos los requerimientos. Los use cases también se utilizan al establecer el plan de un proyecto, ayudando a definir qué debe incluirse en cada versión (release).

Siguiendo un poco la terminología que hemos adoptado para este libro, podemos decir que “los requerimientos de un sistema que se observan desde el exterior” no son otra cosa que el funcionamiento del

sistema; por lo que podemos rephrasear la definición de Schneider y Winters, diciendo que los use cases se utilizan para describir el funcionamiento de los procesos que conforman un sistema.

El autor Alistair Cockburn, autor de *Writing Effective Use Cases* -Escribiendo Use Cases Efectivos- y de multitud de artículos y conferencias sobre el tema, nos dice lo siguiente:

Los actores son básicamente usuarios del sistema. Son realmente tipos o categorías de usuario. Los actores son las entidades externas (gente u otros sistemas) que interactúan con el sistema para alcanzar una meta deseada.

Los use cases son lo que sucede cuando los actores interactúan con el sistema. Un actor utiliza el sistema para alcanzar una meta deseada. Registrando todas las formas en que nuestro sistema es utilizado (cajas del uso o casos del uso), podremos recopilar todas las metas o requerimientos de nuestro sistema.

Por lo tanto: Un use case es una colección de las posibles secuencias de interacción entre el sistema bajo discusión y sus usuarios (o los actores), relacionadas con una meta particular. La colección de use cases debe definir todo el comportamiento del sistema en lo que es relevante para que los actores aseguren el logro o cumplimiento correcto de sus metas. Ningún otro comportamiento del sistema, que no sea aplicable a los actores se debe incluir en los use cases.

En esta concepción, el Dr. Cockburn incluye un elemento muy importante que la mayoría de las definiciones omite; se trata de las metas u objetivos de los actores y del hecho que un use case describe lo que ocurre para que ese actor logre su objetivo.

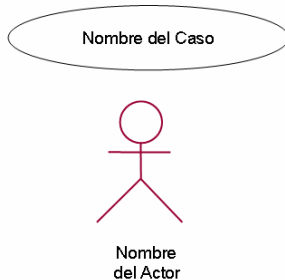
Así pues, podemos finalizar diciendo que un use case describe un proceso, actividad o tarea, en términos de:

- El nombre que lo distingue
- Todas las secuencias de pasos o flujos de tareas que se realizan para alcanzar una meta de los actores o usuarios.
- Las condiciones previas y posteriores que “inician” o “finalizan” el caso.

2.- Diagramas de use case

Un diagrama de use case presenta un conjunto de use cases, junto con los actores que participan en su ejecución, por lo que permite representar gráficamente el funcionamiento de una actividad, un proceso o un sistema; esto es, muestra cómo los usuarios operan (u operarán) y

ejecutan (o ejecutarán) sus tareas, independientemente de los componentes o estructura del sistema en uso (o a ser desarrollado).



Cada use case, dentro de un diagrama de use case, se muestra como una elipse, en la cual se inscribe su nombre, y cada actor se representa con un “muñeco de líneas”, debajo del cual se coloca el nombre del actor.

El diagrama de use cases se acompaña con las narrativas que describen cada uno de los use cases que lo componen. Esta descripción de

los pasos que se ejecutan en cada uno de los use case puede ser hecha de diversas formas; es muy común utilizar texto libre, aunque puede utilizarse pseudocódigo¹ o un diagrama de actividad.

En general, los actores pueden ser:

- Actores principales, que son las personas que usan el sistema.
- Actores secundarios, que son las personas que mantienen o administran el sistema.
- Otros sistemas con los que el sistema interactúa.

Es importante observar que la idea de actor no se refiere a personas específicas, sino a roles o funciones; una misma persona puede interpretar varios roles, cumpliendo las tareas de actores distintos.

La descripción del use case normalmente incluye:

- El inicio: cuándo y qué actor lo produce.
- Condiciones previas necesarias para la ejecución del use case.
- El objetivo del actor en el use case: ¿qué resultado persigue?
- La interacción de los actores con el use case: qué mensajes intercambian.
- Situación o secuencia normal de operaciones.

¹ El pseudocódigo es un lenguaje “tipo telegrama” que imita un lenguaje de programación y se utiliza para describir algoritmos. Permite que el diseñador de un algoritmo se centre en su lógica sin distraer su atención con los detalles de la sintaxis del lenguaje de programación que será utilizado.

- El fin: cuándo se produce y qué resultado genera.
- Situaciones especiales o ejecuciones alternativas.
- Condiciones posteriores a la ejecución del use case

Un use case debe incluir los siguientes elementos:

- **Inicio:**

Un use case es siempre lo inicia un actor, bien sea directa o indirectamente. Puede darse el caso en que se inicie un use case sin que el actor esté consciente de ello, sin embargo, éste se inicia “en su representación”.

- **Resultado:**

Un use case produce un resultado, genera “algo de valor”. Este resultado no es necesariamente un resultado que “salga” del use case -esto es, que se entregue a otro actor o use case-, pero si debe ser un resultado observable.

- **Fin:**

Un use case no está completo, hasta que no se señala su final.

- **Descripción:**

Recordemos que un use case es, en principio, una descripción de una tarea o actividad. En 1993, cuando Ivar Jacobson publicó su libro Object-Oriented Software Engineering (Ingeniería de Software Orientada a Objetos), introdujo los elementos gráficos que hemos descrito para los diagramas de use case, que, a su vez, han sido adoptados por la OMG para el UML. Por esta razón, muchas personas identifican los use case con su representación gráfica; sin embargo, es muy importante mantener en mente que la descripción del use case es fundamental.

3.- Relaciones

En un diagrama de use case, los casos -proceso, actividad o tarea- se conectan a los actores a través de asociaciones, con el fin de mostrar quiénes son los actores que los inician y con cuáles actores se establece comunicación.

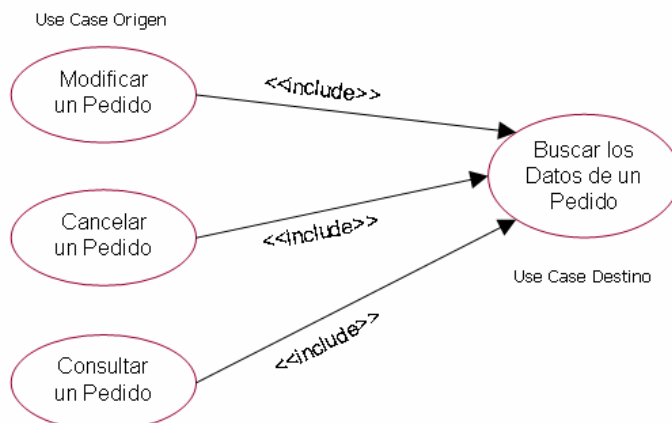
A su vez, los use cases se asocian entre si y estas asociaciones pueden ser de varios tipos: inclusión o utilización y extensión.

3.1.- Inclusión/Utilización

Decimos que existe una relación de inclusión entre dos use cases, cuando el use case origen, también incluye las acciones correspondientes al use case destino.

Para denotar estas relaciones se utiliza una flecha de cabeza triangular y el símbolo o estereotipo <<include>>. En algunos textos se encontrará la relación de inclusión con la denominación de utilización, utilizando el estereotipo <<uses>>.

Es importante señalar que los términos origen y destino, no tiene ninguna implicación, más que indicar la posición que guardan los use case en relación a la flecha que representa la relación. El use case origen es el que está al inicio de la flecha y el use case destino el que está al final de ésta; en el caso de la inclusión, el origen se coloca en el use case que incluye y el destino, en el use case incluido.



El ejemplo anterior nos muestra una relación de inclusión. En éste, si imaginamos un sistema de manejo de órdenes, en el cual se ingresan, modifican, consultan y cancelan pedidos de clientes, podemos observar que al realizar las operaciones de consulta, modificación o cancelación de órdenes, siempre será necesario “tener a la mano” los datos del pedido. Así pues, al modelar los use case correspondientes a estas operaciones, con el fin de evitar la repetición de la secuencia de pasos que corresponden a “buscar los datos del pedido”, será preferible colocar éstas tareas en un use case aparte, estableciendo un vínculo de inclusión.

3.2.- Extensión

En general, diremos que existe una relación de extensión, cuando el use case origen extiende las acciones del use case destino. Para denotar estas relaciones se utiliza una flecha de cabeza triangular y el símbolo o estereotipo <<extend>>.

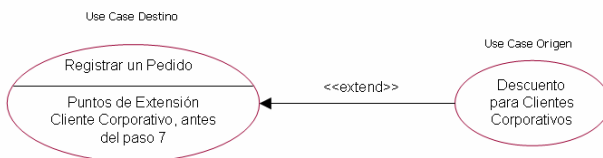
Las relaciones de inclusión y extensión constituyen una forma de factorizar las acciones de un use case. Esto es, tanto el use case incluido, como el use case que extiende representan un fragmento de las acciones que se cumplen en otro use case.

En el caso de la relación de inclusión, se busca evitar una repetición de las mismas acciones en distintos use cases, mientras que en el caso de la relación extensión, se busca describir una secuencia de acciones alternativa a las del comportamiento normal, permitiendo simplificar las especificaciones, cuando una variación pudiera oscurecer la comprensión del caso.

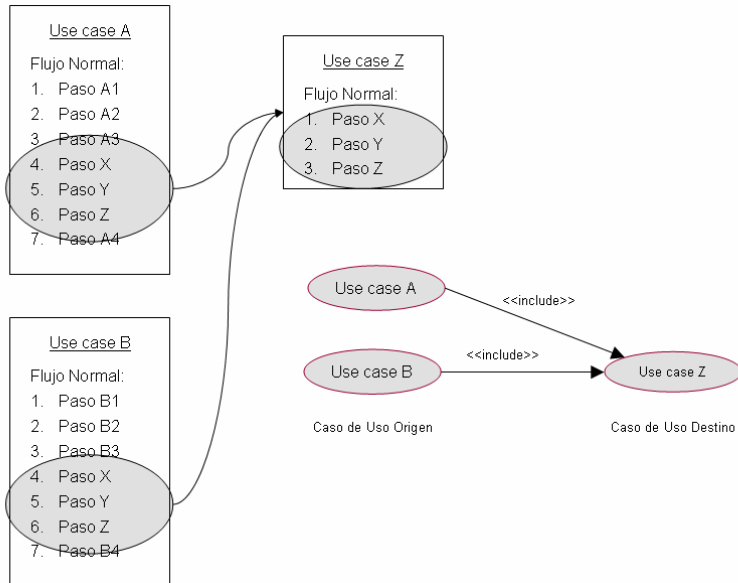
3.3.- ¿Cuándo incluye y cuándo extiende?

Siempre existen dudas acerca de el tipo de relación que existe entre dos use cases, ¿es inclusión o es extensión? Como regla general, podemos seguir los siguientes criterios:

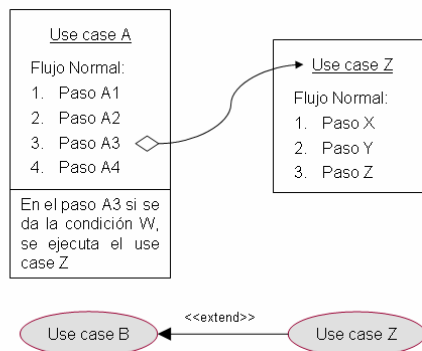
- Si el flujo de un use case A contiene una serie de pasos similares a los que contiene otro use case B y, para evitar la repetición, hemos decidido separar esos pasos comunes en otro use case Z, diremos que los use cases A y B incluyen el use case Z. En efecto, la ejecución de Z está incluida dentro de los use cases origen, A y B.



- Si, por el contrario, puede considerarse que el use case A “interrumpe” su flujo para que “se ejecuten” las funciones que están incluidas en otro use case Z, diremos que el use case Z extiende el use case A.



En algunas oportunidades, al modelar un proceso, se representa la “situación normal” o “escenario primario” y, al considerar las secuencias alternativas -como es el manejo de excepciones-, se prefiere modelar cada una de ellas en un use case aparte, con la finalidad de tener modelos más simples y claros. En estos casos, se utiliza una relación de extensión, como se muestra en el ejemplo de un sistema de entrada de órdenes, en el cual, normalmente, se calcula el monto total de la orden sobre la base de: las cantidades ordenadas, los precios unitarios de los ítems ordenados, rebajando los descuentos que se aplican a todos los clientes, con la excepción de ciertos clientes especiales a quienes se les otorga descuentos especiales.



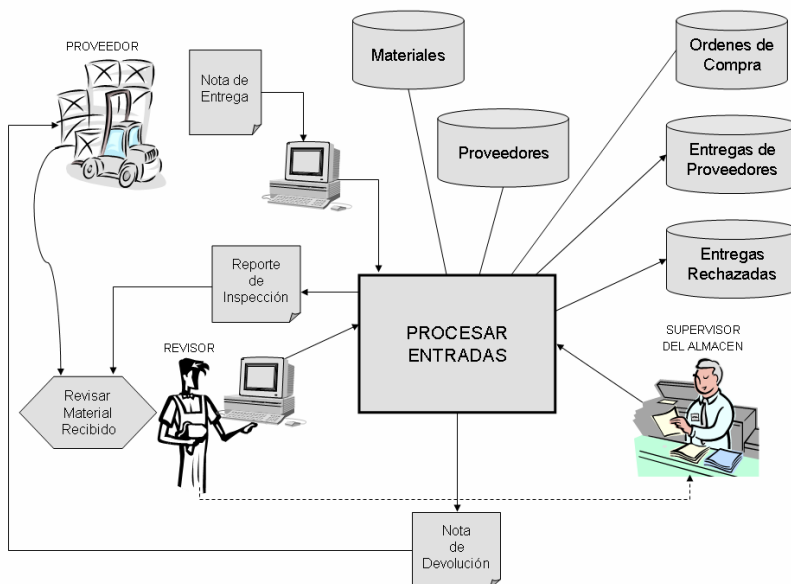
4.- Desarrollo de use cases

Para describir los pasos que deben cumplirse en la elaboración de un diagrama de use cases nos basaremos en un ejemplo que incluye cuatro grandes actividades:

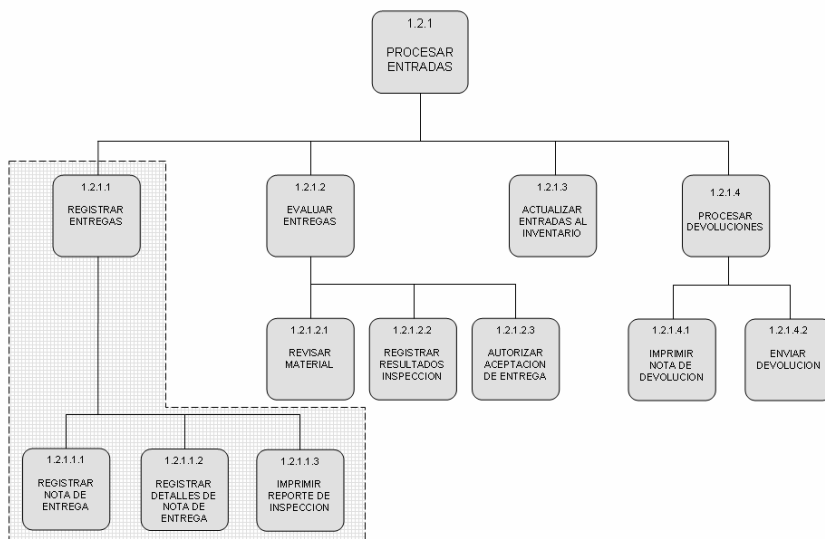
En un sistema de Control de Inventario, el encargado de recibir las entregas de los proveedores registrará los datos reflejados en las notas de entrega presentadas por los proveedores, al momento en que éstas son hechas en el almacén, y se crearán los registros para imprimir un Reporte de Inspección, con el cual se realizará la inspección de los ítems recibidos.

Una vez hecha la revisión, el revisor, encargado de realizar la inspección, registrará las cantidades aceptadas y rechazadas de cada material. Para cada rubro que no haya sido aceptado, indicará las razones para el rechazo.

El supervisor del almacén verificará los datos de las entregas registradas en el sistema y de su posterior inspección. Una vez hecha esta verificación, aceptará oficialmente la entrega dando inicio a los procesos de actualización del inventario, creación de la información de entregas aceptadas que servirá de base para las cuentas por pagar y devolución de ítems rechazados.



Este proceso tiene como objetivo principal registrar la entrada de los materiales entregados por los proveedores -recibidos y aceptados- y actualizar las bases de datos. Sin embargo, como puede observarse en el diagrama de procesos -diagrama de descomposición funcional-, para alcanzar este objetivo se cumple todo un conjunto de tareas.



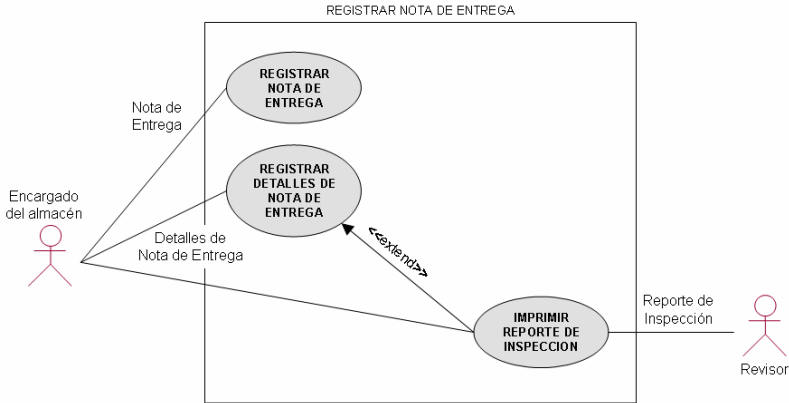
Para los efectos de nuestro ejemplo, solamente centraremos la atención en las tareas que aparecen dentro del área sombreada; esto es, centraremos nuestra atención en la actividad REGISTRAR ENTREGAS, la cual incluye tres tareas:

1.2.1.1.1 -REGISTRAR NOTA DE ENTREGA (crear, modificar o eliminar el encabezado de la nota de entrega, que contiene los datos de número de entrega, fecha de entrega y fecha y número de referencia que aparezcan en el documento del proveedor)

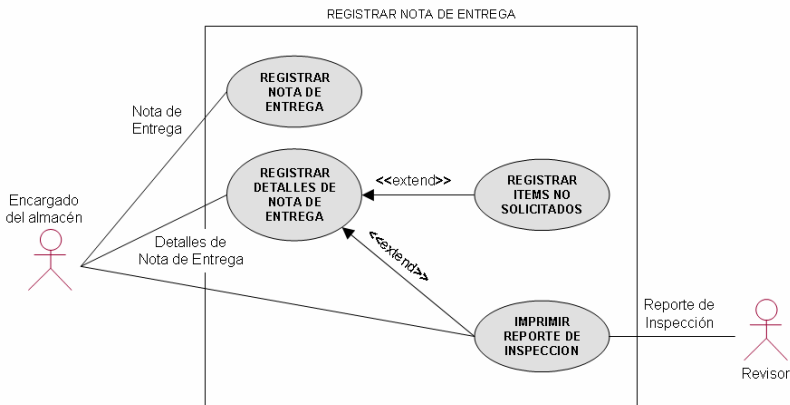
1.2.1.1.2 -REGISTRAR DETALLES DE NOTA DE ENTREGA (crear, modificar o eliminar las líneas de la nota de entrega, copiando la información de las órdenes de compra colocadas a ese proveedor y señalando las cantidades recibidas).

1.2.1.1.3 -IMPRIMIR REPORTE DE INSPECCION.

El funcionamiento de esta actividad, puede ser modelado con el siguiente diagrama de use cases:



Al desarrollar el use case REGISTRAR DETALLES DE NOTA DE ENTREGA y hacer las consideraciones de posibles situaciones especiales, observamos que una de ellas es la presencia en la nota de entrega de un ítem que la empresa no haya solicitado. Una alternativa para manejar esta situación sería ignorarla, pero si se requiere que el Reporte de Inspección incluya todos los detalles, deberemos crear un use case adicional, REGISTRAR ITEMS NO SOLICITADOS, que extienda las funciones del primero, atendiendo este tipo de situación especial con una secuencia totalmente diferente de pasos.



Para desarrollar la descripción de cada uno de los use case del diagrama daremos los siguientes pasos:

1. Identificar los actores (roles)
2. Identificar sus metas
3. Visualizar el funcionamiento del proceso y escribir un borrador del use case
4. Elaborar el primer borrador del diagrama de use cases
5. Visualizar situaciones especiales
6. Completar el diagrama de use cases, para incluir todos los casos
7. Identificar las interrelaciones entre use cases
8. Desarrollar las descripciones de cada uno de los use cases incluidos en el diagrama

5.- Definición de escenarios

Tal como hemos señalado, los escenarios en un use case constituyen el conjunto de todos los cursos de acción alternativos que pueden ser tomados por el sistema, para cumplir los objetivos del actor y obtener el resultado deseado.

Especialmente para un use case de diseño detallado, será importante que la definición de escenarios sea completa. En muchos casos, esta tarea puede ser bastante compleja, por lo que será conveniente echar mano de la técnica de tablas de decisión -descrita en el capítulo correspondiente-, con el fin de ordenar todo el conjunto de condiciones y acciones que, para un use case, puedan darse.

Normalmente, en un use case existirán dos grandes grupos de cursos de acción: las acciones que constituyen una finalización exitosa y las acciones que constituyen, por así llamarlas, el fracaso; esto es, aquellos escenarios en los que no puede obtenerse el resultado que se busca. Por ejemplo, en el caso del cobro de un cheque, si la cuenta corriente tiene saldo disponible, el cheque será pagado -éxito-; sin embargo, si la cuenta no dispone de suficientes fondos, el cheque será rechazado -fracaso-.

Dentro de estos dos grandes grupos de escenarios, normalmente existirán diversas combinaciones de situaciones que nos pueden llevar por caminos diferentes, que deben ser perfectamente identificados para tener un use case completo.

6.- Consideraciones de tipo práctico

Al desarrollar use cases, es conveniente tener presente que éstos definen la interacción de los usuarios con el sistema y que podemos verlos como si se tratara de un manual del usuario -el usuario hace esto y el sistema hace esto otro-, ya que básicamente, cuando el sistema esté listo, contendrán lo que incluye dicho manual.

Una vez hecha la definición inicial de un use case, debe refinarse hasta obtener una especificación completa, clara y concisa, en la que quede muy claro quiénes son los actores, las acciones normales y las acciones alternativas o escenarios.

Dentro de la descripción de un use case, normalmente se evita el uso de estructuras tipo “If-Then-Else”, ya que la sección de flujos alternativos permite indicar los diferentes caminos que puede tomar un use case; sin embargo, cuando se tienen distintas opciones, todas ellas consideradas normales o igualmente probables, pueden incluirse dentro del flujo normal, señalando los criterios de bifurcación para cada caso.

La experiencia señala que cuando comenzamos a utilizar esta técnica de modelaje, es frecuente que se cometan errores como los que, a continuación, señalamos:

- Describir sólo la interacción del usuario y no la respuesta del sistema. Los use case deben describir ambos lados de la comunicación, lo que hace el usuario y lo que hace el sistema en respuesta a las acciones del usuario.
- Prestar más atención a los atributos que al funcionamiento. No se deben incluir mayores detalles acerca de la forma en que se presentará la interfaz al usuario -menú, botones, ventanas-, ya que se trata de centrar la atención en lo qué el sistema hace (o hará) y no en cómo lo hace (o hará). En lugar de indicar que “el usuario presiona el botón CREAR”, es preferible describir que “el usuario selecciona la acción de crear”. No nos interesan la forma cómo el usuario seleccionará la acción de CREAR, nos interesan los pasos que el sistema dará, como respuesta a esa selección.
- Hacer descripciones demasiado breves. Al describir los use case, es preferible abundar que quedarse corto, es importante que se detallen todas las acciones de los actores y todas las respuestas del sistema.

- Escribir en voz pasiva. Los use case deben mostrar las acciones realizadas por el usuario y la respuesta del sistema, por lo que será mucho más sencillo y legible utilizar la voz activa para la descripción de los use case.
- Omitir cursos de acción alternativos. Los use case deben mostrar tanto el escenario normal, como los escenarios alternativos.

7.- Utilización de la técnica de use case

Al igual que ha ocurrido con muchas herramientas de la ingeniería de sistemas, los use case han recibido los más variados y extraños usos, por lo que es importante poner en perspectiva para qué utilizamos use case y cuál es el estilo más adecuado en cada circunstancia.

Cabe señalar que acerca de este tema, una multitud de calificados autores han expresado su opinión, por lo que, más que expresar la nuestra, hemos querido resumir las posiciones que hemos considerado más razonables o prácticas. En efecto, miles de artículos se han escrito sobre la técnica, cualquier consulta que hagamos en Internet sobre use cases, nos arrojará una lista casi interminable de libros, revistas, artículos, grupos de usuarios, etc.

Podemos afirmar que un use case es simplemente una manera de escribir, una de las tantas formas que existen para describir un proceso y las interacciones de éste con el mundo exterior. Esta forma particular de describir los procesos tiene dos secciones:

- La primera sección describe el proceso normal, que contiene las acciones e interacciones que normalmente se realizan.
- La segunda sección presenta una serie de variaciones de ese proceso normal, en las que se describe cómo varía el comportamiento “normal”, de acuerdo a las diferentes situaciones que puedan presentarse.

Evidentemente, esta forma de describir procesos puede ser utilizada para describir cualquier comportamiento y sus alternativas:

- Como descripción general de requerimientos, un use case nos dice:
 - el usuario hace esto
 - el sistema hace esto otro
 - el sistema habla con otro sistema

- algo va mal o algo es diferente, por lo que el sistema ahora hace esto en lugar de lo otro

Así pues, para una definición general de requerimientos, en las etapas de planificación de sistemas o de análisis y diseño general, un use case debe decir lo que el sistema hará, sin decir cómo el sistema deberá hacerlo y, como muy bien nos señala Alistair Cockburn en su obra *Writing Effective Use Cases* (Escribiendo Use Cases Efectivos), “de ninguna manera este tipo de use case presupone el uso de un estilo de diseño orientado a objetos -ni de cualquier otro-”.

- Como descripción de un proceso del negocio, un use case nos dice:
 - el cliente hace esto
 - el vendedor hace esto otro
 - el vendedor da algo a otro departamento
 - el otro departamento hace algo más; etcétera
 - cuando las cosas van mal o algo es diferente, un cierto departamento rechaza una parte del trabajo y va a otro departamento; etcétera.

Esta clase de descripciones son relativamente simples, cualquier persona del negocio, sin mayor entrenamiento previo, puede escribirlas y resultan muy fáciles de leer, por lo que pueden ser de gran utilidad en trabajos de reingeniería de los procesos del negocio.

- Si utilizáramos esta herramienta como documentación de un diseño, un use case describiría:
 - el usuario hace esto
 - un cierto primer componente hace algo y, como resultado, el componente transmite un mensaje a un segundo componente
 - el segundo componente hace otra cosa, etc.
 - bajo otras condiciones, el orden de ejecución de los componentes es diferente, etc.

Si bien existen los fanáticos de los use case, esta herramienta se utiliza con poca frecuencia para documentar un diseño detallado, porque existen mejores formas de hacerlo.

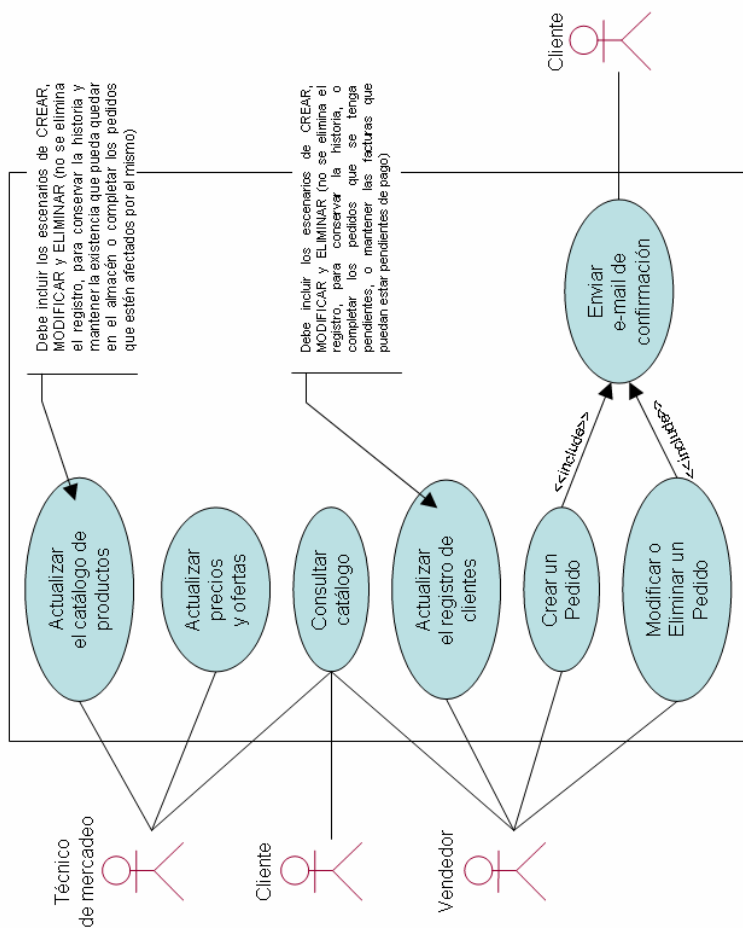
Así pues, podemos resumir diciendo que la técnica o herramienta de use cases es muy útil para modelar o documentar el funcionamiento de un sistema, esto es, las diferentes actividades y tareas que se cumplen para obtener los resultados -metas- que los usuarios -actores- requieren, así como también los diferentes cursos de acción que pueden ser tomados, bajo diferentes condiciones, para obtener dichos resultados.

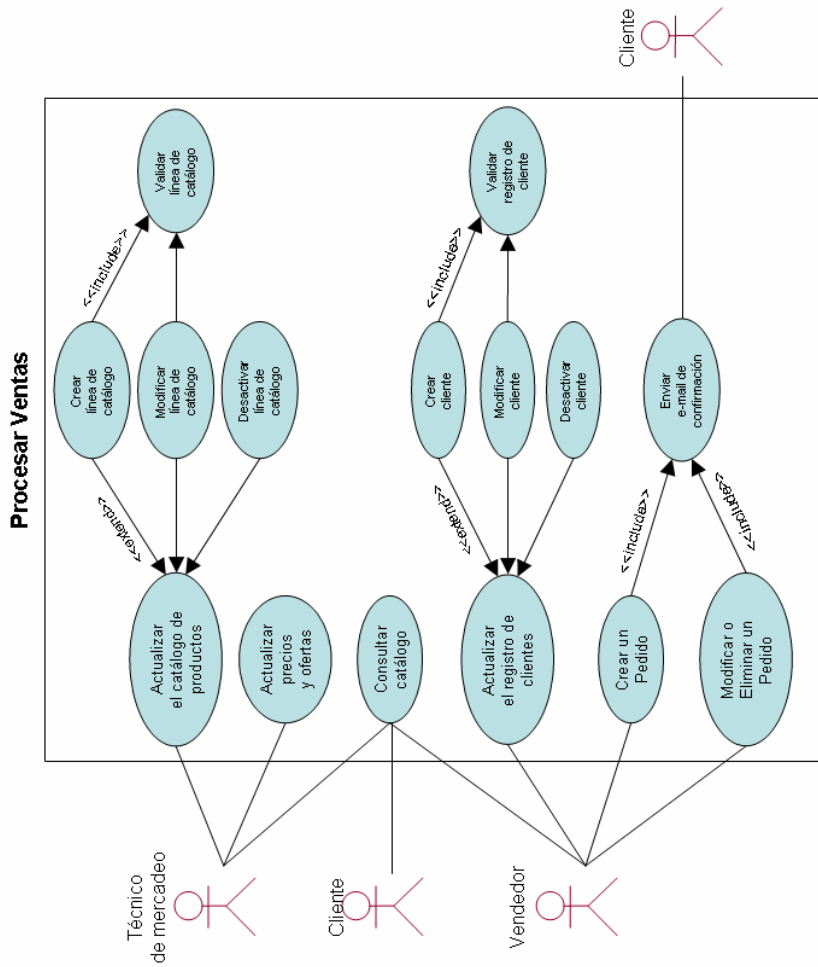
Queremos también señalar que, pese a los esfuerzos que han realizado muchos autores en señalar lo contrario, la herramienta de use cases no es la más indicada para realizar el análisis de un área del negocio, por cuanto, como veremos en la próxima sección, para elaborar un use case su redactor -el analista- debe partir de la definición de los actores y los resultados que éstos obtienen; es decir, para escribir un use case es necesario conocer algo y ese algo deberá ser identificado o conocido mediante la aplicación de algún procedimiento o de alguna técnica o herramienta de levantamiento de información. Más adelante, en la tercera parte de este libro, tendremos oportunidad de revisar con mayor detalle estas consideraciones.

8.- Ejemplos

Veamos, como ejemplo, el proceso de ventas de una imaginaria empresa mayorista de CDs y videos. Para algunos de los use cases, a continuación del diagrama, hemos incluido su descripción.

Procesar Ventas





8.1.- Actualizar el catálogo de productos

1. Descripción

El actor que inicia este use case es el “técnico de mercadeo”. Su objetivo en este use case es actualizar el catálogo de Videos y CD’s -añadir, eliminar, modificar- que contiene la información de descripción, características, proveedor, precio, etc.

2. Flujo de Eventos

2.1 Flujo Básico

- a) El sistema presenta las opciones: Crear, Modificar o Desactivar registro del catálogo de productos, también presenta la opción de Salir.
- b) Si el usuario selecciona la opción Salir, el sistema termina la transacción.
- c) Si el usuario selecciona la acción de Crear ejecuta la acción correspondiente.
- d) Si el usuario selecciona la acción de Modificar o Desactivar:
 - d.1 - El sistema presenta la lista de los productos registrados en la base de datos.
 - d.2 - Si el usuario selecciona un registro de la lista que el sistema le presenta.
 - d.3 - El usuario puede modificar los datos del registro seleccionado y elegir la acción de Modificar o simplemente puede seleccionar la acción de Desactivar.
 - d.4 - Si el usuario selecciona la acción de Modificar el sistema ejecuta la acción correspondiente.
 - d.5 - Si el usuario selecciona la acción de Desactivar el sistema ejecuta la acción correspondiente.
 - d.6 - Regresa al paso 4.1 para seleccionar otro ítem.

2.2 Flujos Alternativos

- a) De acuerdo con la selección del usuario, se ejecutarán las acciones de SALIR, CREAR, MODIFICAR o DESACTIVAR.

3. *Precondiciones*

- 3.1 Existen nuevos productos o se requiere modificar la información de algún producto o se requiere descontinuar la venta de algún producto.
- 3.2 En el caso de un nuevo producto la carátula correspondiente habrá sido escaneada previamente y se almacenará en el directorio de carátulas con el mismo código asignado al producto.

4. *Poscondiciones*

- 4.1 El catálogo de productos ha quedado actualizado.

5. *Puntos de Extensión*

8.1.1.- Crear un renglón en el catálogo de productos

1. Descripción

El actor que inicia este use case es el “técnico de mercadeo”. Su objetivo en este use case es crear un renglón en el catálogo de Videos y CD’s que contiene la información de descripción, características, proveedor, precio, etc.

2. *Flujo de Eventos*

2.1 Flujo Básico

- a) El sistema presenta el formato de creación de un nuevo producto, con los campos requeridos: tipo, código, descripción, precio, proveedor, etc.
- b) El usuario registra los datos correspondientes a los campos del formato e indica cuál es la imagen que corresponde a la carátula.
- c) El sistema ejecuta "Validar línea de catálogo".
- d) Si no hay errores
 - d.1 - El sistema presenta los datos y solicita confirmación.
 - d.2 - Si el usuario confirma, el sistema inserta el registro de catálogo en la base de datos.
 - d.3 - Si el usuario no confirma el sistema no toma acción.
- e) Si en el paso c se han detectado errores
 - e.1 - El sistema presenta los datos con los mensajes de error que correspondan.

e.2 - El usuario modifica los datos presentados por el sistema

e.3 - El sistema regresa al paso c

2.2 Flujos Alternativos

a) El usuario puede seleccionar la opción de SALIR

3. *Precondiciones*

3.1 La carátula correspondiente habrá sido escaneada previamente y se almacenará en el directorio de carátulas con el mismo código asignado al producto.

4. *Poscondiciones*

4.1 El catálogo de productos ha quedado actualizado con los nuevos productos.

5. *Puntos de Extensión*

8.1.2.- **Modificar un renglón en el catálogo de productos**

1. Descripción

El actor que inicia este use case es el “técnico de mercadeo”. Su objetivo en este use case es modificar un renglón en el catálogo de Videos y CD’s que contiene la información de descripción, características, proveedor, precio, etc.

2. *Flujo de Eventos*

2.1 Flujo Básico

a) El usuario modifica los datos correspondientes a los campos del formato

b) El sistema ejecuta "Validar línea de catálogo".

c) Si no hay errores

c.1 - El sistema presenta los datos y solicita confirmación.

c.2 - Si el usuario confirma el sistema reemplaza el registro en la base de datos.

c.3 - Si el usuario no confirma el sistema no toma acción.

d) Si en el paso b se han detectado errores.

d.1 - El sistema presenta los datos con los mensajes de error que correspondan.

d.2 - El sistema regresa al paso a.

2.2 Flujos Alternativos

- a) El usuario puede seleccionar la opción de SALIR

3. *Precondiciones*

4. *Poscondiciones*

- 4.1 El catálogo de productos ha quedado actualizado.

5. *Puntos de Extensión*

8.1.3.- Desactivar un renglón en el catálogo de productos

1. Descripción

El actor que inicia este use case es el “técnico de mercadeo”. Su objetivo en este use case es eliminar un renglón en el catálogo de Videos y CD’s.

2. *Flujo de Eventos*

2.1 Flujo Básico

- a) El sistema presenta los datos del registro y solicita confirmación de la eliminación.
- b) Si el usuario confirma, el sistema coloca el producto en estatus “inactivo” (no se elimina el registro, para conservar la historia y mantener la existencia que pueda quedar en el almacén o completar los pedidos que estén afectados por el mismo).
- c) Si el usuario no confirma, el sistema no toma acción

2.2 Flujos Alternativos

- a) El usuario puede seleccionar la opción de SALIR

3. *Precondiciones*

4. *Poscondiciones*

- 4.1 El catálogo de productos ha quedado actualizado, los productos eliminados han quedado en estatus “inactivo”.

5. *Puntos de Extensión*

8.2.- Actualizar precios y ofertas

1. Descripción

Este use case lo inicia el actor “técnico de mercadeo”. Su objetivo en este use case es actualizar los precios o incluir precios de oferta. El técnico de mercadeo consulta el catálogo de productos y

selecciona a los que desea aplicar un nuevo precio o un precio de oferta, también puede borrar o modificar ofertas anteriores.

2. *Flujo de Eventos*

2.1 Flujo Básico

- a) La pantalla muestra una lista con los productos del catálogo actual.
- b) Por cada línea:
 - b.1 - El sistema mostrará su descripción, el precio regular y el precio de oferta (si no hay ofertas anteriores este precio estará en cero), con su correspondiente fecha de vencimiento.
 - b.2 - El técnico de mercadeo puede modificar los campos de precio y fecha de vencimiento del precio de oferta.
 - b.3 - El sistema le presentará los datos del ítem modificado y le pedirá confirmación del cambio; si acepta, el registro correspondiente será actualizado. Si no confirma el cambio, el sistema no tomará ninguna acción sobre el registro.

2.2 Flujos Alternativos

3. *Precondiciones*

4. *Poscondiciones*

- 4.1 Los nuevos precios sólo afectan a los pedidos que ingresen al sistema después de la actualización.

5. *Puntos de Extensión*

8.3.- *Modificar/eliminar un pedido*

Este use case podemos presentarlo de dos formas, una versión larga y una versión más simple, separándolo en varios casos.

Como podrá observarse, la descripción de la “versión larga” resulta demasiado extensa, por lo que será preferible presentarlo mediante varios use cases cuya descripción sea más simple:

- El use case “padre” Modificar/Eliminar un pedido, en el cual indicaremos que se ejecutan las acciones alternativas de modificar o eliminar.

- El use case Modificar, incluido en el use case “padre”.
- El use case Eliminar -o desactivar-, incluido en el use case “padre”.

8.4.- *Modificar/eliminar un pedido (versión larga)*

1. *Descripción*

El actor que inicia este use case es el vendedor. Su objetivo en este use case es modificar un pedido de un cliente. El vendedor selecciona el cliente y el sistema muestra los pedidos pendientes para ese cliente. Los pedidos en estatus de “pedido pendiente” pueden ser modificados o eliminados.

2. *Flujo de Eventos*

2.1 Flujo Básico

- a) El vendedor, dentro de la lista de clientes, selecciona el cliente para el cual desea modificar o eliminar un pedido.
- b) El sistema presenta los datos del cliente y la lista de pedidos que estén en estatus de “pedido pendiente”.
- c) El vendedor puede seleccionar cualquier pedido, para ser modificado o eliminado.
 - c.1 - Si el vendedor indica que desea eliminarlo.
 - c.1.1 - El sistema presenta los datos del pedido, junto con un mensaje de solicitud de confirmación de la eliminación.
 - c.1.2 - Si el vendedor confirma, el sistema pone el registro en estatus de cancelado (con el fin de mantener la historia).
 - c.1.3 - Si el vendedor no confirma, el sistema no toma acción sobre el registro.
 - c.2 - Si el vendedor indica que desea modificar la dirección de envío.
 - c.2.1 - El sistema presenta los datos del pedido y la dirección de envío prevista para el pedido.

- c.2.2 - Si el vendedor modifica la dirección de envío, el sistema modifica ese dato en el registro.
- c.3 - Si el vendedor indica que desea modificar o eliminar algún ítem del pedido.
 - c.3.1 - El sistema presenta los datos del pedido y la lista de ítems que lo componen.
 - c.3.2 - Para cada ítem, el vendedor puede modificar la cantidad ordenada y solicitar la acción de modificar o simplemente, puede solicitar la acción de eliminar el registro.
 - c.3.3 - Si el vendedor modifica la cantidad ordenada e indica la acción modificar, el sistema modifica ese dato en el registro correspondiente.
 - c.3.4 - Si el vendedor indica que desea eliminar el ítem:
 - i) El sistema presenta los datos del ítem, junto con un mensaje de solicitud de confirmación de la eliminación.
 - ii) Si el vendedor confirma, el sistema pone el registro en estatus de cancelado (con el fin de mantener la historia).
 - iii) Si el vendedor no confirma, el sistema no toma acción sobre el registro.
- d) Cuando el vendedor indica que ha terminado de actualizar el pedido, el sistema ejecuta las

funciones de “Enviar un e-mail de confirmación” para generar el mensaje de confirmación al cliente.

2.2 Flujos Alternativos

a) En el punto 3.2.2

Si en el punto 3.2.2, al introducir alguno de los campos, el sistema detecta algún error se generará un mensaje indicando los errores.

b) En el punto 3.3.4.2

Si en el paso 3.3.4.2 el vendedor confirma la eliminación y el sistema determina que el todos los ítems del pedido han sido cancelados, el sistema también pone el registro del pedido en estatus de cancelado (con el fin de mantener la historia).

3. *Precondiciones*

3.1 El cliente tiene un pedido o una oferta registrada en la base de datos.

4. *Poscondiciones*

4.1 Los registros de pedidos pendientes habrán sido actualizados.

5. *Puntos de Extensión*

5.1 Añadir un ítem en el punto 3.4.1

En el paso 3.4.1, en caso de que el vendedor desee añadir un ítem al pedido, se invoca el use case “Crear una línea de pedido”.

5.2 Enviar un e-mail en el punto 4

En el paso 4, una vez que se ha completado la operación el sistema envía un e-mail al cliente para informarle que su pedido ha sido modificado o cancelado, de acuerdo a su solicitud.

9.- Nomenclatura utilizada en UML

