

Casos de uso

Tabla de contenido

Casos de uso	2
Definición (informal)	2
Definición (formal)	2
Actor	2
Definición	2
Casos de uso	2
Tecnológicamente neutros	2
El modelo visual: diagramas de casos de uso en UML	2
Escenario	3
Definición	3
Casos de uso	3
Un contrato	3
Formato breve	3
Formato secuencia de acciones	3
Formato de dos columnas o diálogo	4
Más completo: precondiciones, postcondiciones, escenarios alternativos	4
¿Cómo encontrarlos?	5
Casos de uso	5
Relaciones de inclusión	5
Relaciones de extensión	5
Relaciones de generalización	6
Adicionales	6
Casos de uso CRUD / ABMC	7
Casos de uso en el proceso de desarrollo	7
Proceso unificado	7
Proceso unificado	7
Modelo de análisis en el proceso unificado	8
Modelo de casos de uso	8
à la Larman	8
Contratos y operaciones	8
Diagrama de secuencia del sistema	9
Operaciones del sistema	9
Casos de uso	10
2.0	10
Conclusión	10

Casos de uso

Definición (informal)

Informalmente, podemos decir que un caso de uso es **una historia acerca de cómo un actor (el usuario), utiliza un sistema para alcanzar sus objetivos**. Por ejemplo, al especificar el sistema de inscripciones de una universidad, probablemente nos encontraremos con un caso de uso como este:

Consultar cursos disponibles

El alumno consulta los cursos disponibles para una materia en la que se quiere inscribir. Para cada curso disponible, el sistema le informa días y horarios, docentes y cantidad de vacantes.

Que describe un escenario de utilización del sistema en particular y correspondiente a la consulta de los cursos disponibles.

Definición (formal)

Más formalmente, un caso de uso es **una secuencia de acciones realizadas por un sistema que generan un resultado observable de valor para un actor en particular**.

Actor

Definición

En este contexto, un actor es **cualquier entidad con comportamiento que interactúa con el sistema: personas, organizaciones, e inclusive otros sistemas**. Volviendo al ejemplo, un alumno es en este caso un actor que interactúa con el sistema de inscripciones.

Casos de uso

Tecnológicamente neutros

Es importante destacar que **los casos de uso son tecnológicamente neutros**, es decir, que son esenciales, no describen la solución, no describen la interfaz del usuario. Describen cuáles son las acciones del usuario, cuáles son las respuestas del sistema, pero lo hacen en términos tecnológicamente neutros, es decir, independientemente de los aspectos de implementación.

El modelo visual: diagramas de casos de uso en UML

Los casos de uso también se pueden representar en forma visual. Esto se hace utilizando la notación propuesta por UML. En el gráfico podemos apreciar los actores, los casos de uso y los límites del sistema analizado, que se representa a través de un rectángulo. Es muy importante destacar que este diagrama no reemplaza, de manera alguna, a las descripciones, a las especificaciones de los casos de uso en formato texto. Como se puede ver, el diagrama aporta muy poca información. No podemos sacar ninguna conclusión acerca de cuáles son los objetivos de cada caso de uso, de cuáles son los objetivos de cada actor, del comportamiento que implica cada caso de uso, de las acciones que hace el actor. Todo este detalle es lo que vamos a encontrar en la especificación de cada caso de uso.



Un caso de uso...

- ☐ es una secuencia de acciones que realiza un actor que otorgan un tipo de resultado
- ☐ detalla claramente cómo el sistema elabora las respuestas
- ☒ **Es tecnológicamente neutro**

Escenario

Definición

Los casos de uso están compuestos de escenarios. Un escenario es **una secuencia particular de acciones e interacciones dentro de un caso de uso**. Normalmente nos vamos a encontrar con un **escenario principal**, con lo que muchos llaman “el camino feliz”, o sea que, muestra las acciones del actor y las respuestas del sistema sin indicar situaciones particulares, excepciones u errores; y uno o varios **escenarios alternativos o secundarios** que describen situaciones particulares que se pueden dar bajo determinadas circunstancias.

Consultar cursos disponibles

Escenario principal

El **alumno** consulta los cursos disponibles para una materia en la que se quiere inscribir. Para cada curso disponible para la materia, el sistema le informa días y horarios, docentes y cantidad de vacantes.

Escenario alternativo

De no existir cursos disponibles, el sistema le informa al alumno “No hay cursos disponibles para la materia ingresada”

Por ejemplo, en este caso de uso, nos encontramos con que hay un escenario alternativo que describe qué es lo que pasa cuando no hay cursos disponibles para una materia que el alumno ha consultado.

Casos de uso

Un contrato

Resumiendo:

- Un caso de uso describe un **contrato** entre las partes interesadas acerca del **comportamiento** de un **sistema**.
- Dicho comportamiento describe qué responde el sistema a las acciones de una de las partes interesadas en particular llamada **actor primario**.
- El actor primario inicia una **interacción** con el sistema para **lograr** algún tipo de **objetivo**. El sistema responde protegiendo los intereses de todas las partes interesadas.
- El caso de uso recoge, incluye, varios **escenarios** que son el resultado de distintas condiciones y acciones de los actores.

Formato breve

Para describir un caso de uso hay varios formatos posibles. En el llamado **formato breve**, simplemente hay un título que siempre es un verbo en voz activa más un objeto, y una descripción resumida del comportamiento.

Consultar productos disponibles

El cliente consulta los productos disponibles, filtrándolos por categoría, descripción o código de producto.

Formato secuencia de acciones

En este otro formato, se describen las acciones del actor y las respuestas del sistema uno por uno. Es importante destacar que el sistema se escribe como una **caja negra**, **no hay detalles de cómo el sistema elabora las respuestas a cada una de las acciones del actor**.

Consultar productos disponibles

El cliente consulta los productos disponibles, filtrándolos por categoría, descripción o código de producto.

- 1) El cliente ingresa una categoría de producto, un código de producto o una descripción de producto (total o parcial)
- 2) El sistema muestra el código, la descripción, la categoría y el stock disponible del producto o de los productos que cumplan el criterio de búsqueda.

En el ejemplo vemos que un cliente ingresa una categoría de producto y que el sistema muestra el código, la descripción, la categoría, etc. El caso de uso no describe cómo el sistema elabora esa respuesta, más allá de los aspectos de implementación que claramente están totalmente prohibidos en el caso de uso.

Formato de dos columnas o diálogo

En este otro formato, llamado de dos columnas o diálogos se opta por separar las acciones y respuestas en 2 columnas separadas.

Consultar productos disponibles	
El cliente consulta los productos disponibles, filtrándolos por categoría, descripción o código de producto.	
Actor	Sistema
Ingresa una categoría de producto, un código de producto o una descripción de producto (total o parcial)	Informa el código, la descripción, la categoría y el stock disponible del producto o de los productos que cumplan con el criterio de búsqueda.

Más completo: precondiciones, postcondiciones, escenarios alternativos

Por supuesto, hay otros campos que se pueden agregar y que se deben agregar al caso de uso:

- ◆ Uno de esos campos son las **precondiciones**. Las precondiciones son prerequisites para que el caso de uso se pueda ejecutar, y que por supuesto valga la pena contarle al lector. Precondiciones puede ser el estado inicial del sistema o la ejecución de otro caso de uso anterior. Las precondiciones no describen la intención del usuario ni de obviedad del tipo “el sistema está disponible” o “el usuario está logueado”. Nuevamente tenemos que pensar en quién es el que va a leer, qué información necesita para entender lo que pasa en el caso de uso.
- ◆ Las **postcondiciones**, describen qué es lo que debe ser cierto para cumplir con el objetivo del actor con respecto al caso de uso. Puede ser algo observable, o no; un cambio en el estado del sistema. Por ejemplo, objetos, asociaciones del modelo de dominio que se han creado, actualizado o eliminado.
- ◆ También se puede incluir un **disparador**, que es un **trigger**.
- ◆ Los requisitos no funcionales asociados a el caso de uso.

Agregar producto a un pedido de compra	
El cliente agrega un producto a su pedido de compra.	
Precondiciones	
El cliente ya consultó los detalles del producto.	
Actor	Sistema
1. Ingresa el código de producto que desea y la cantidad	2. Informa la descripción, el precio por unidad, el precio total del ítem
Postcondiciones	
El producto y la cantidad de unidades ingresadas han sido agregadas al pedido de compra.	
Escenarios alternativos	
2.1 Si no hay stock del producto ingresado, el sistema informa “No hay actualmente unidades disponibles del producto seleccionado. La entrega de este producto podrá verse demorada”	

[V / F] Que un sistema se describa como una caja negra significa que no hay detalles de cómo el sistema elabora las respuestas.

¿Cómo encontrarlos?

¿Cómo hacemos para encontrar los casos de uso? Bueno es importante:

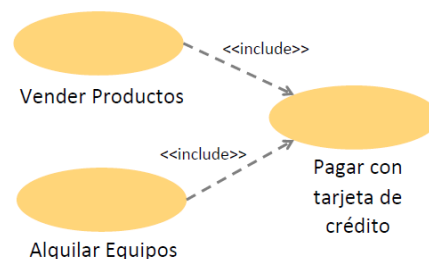
1. Establecer, entender cuáles son los **límites del sistema** que es lo que de alguna manera vamos a incluir y qué es lo que vamos a dejar afuera, a quiénes son los actores que van a interactuar con el sistema.
2. Tenemos que identificar cuáles son los **actores** primarios y cuáles son sus **objetivos**.
3. **Para cada uno de los objetivos**, en general de cada uno de los actores, en general, vamos a encontrar un caso de uso.
4. Para cada uno de los casos de uso, lo que vamos a hacer es identificar las pre y las post condiciones.
5. Después vamos a describir los escenarios principales de cada uno de los casos de uso.
6. Buscaremos y describiremos los escenarios alternativos de cada uno de los casos de uso.
7. Revisaremos, refinaremos y ajustaremos: Buscaremos relaciones de inclusión, relaciones de extensión y especializaciones.

Casos de uso

Relaciones de inclusión

Muchas veces nos vamos a encontrar con que hay comportamiento que se repite en varios casos de uso. Para evitar duplicar el texto una y otra vez, lo que se puede hacer es crear un nuevo caso de uso y poner en ese caso de uso el comportamiento compartido que hasta ahora hemos visto repetido en más de un caso de uso, e invocarlo, a ese nuevo caso de uso, desde los casos de uso que lo necesiten. Esos casos de uso son llamados casos de uso **base**, y esta relación que tenemos entre los casos de uso base y ese caso de uso nuevo que hemos creado, son **relaciones de inclusión**.

CU010: Vender Productos
– Escenario principal:
• 1) El cliente arriba al punto de venta con los productos para pagar.
• 2) ...
• 6) El cliente paga su compra (CU020 Pagar con tarjeta de crédito).
CU011: Alquilar equipos
– Escenario principal:
• ...
• 3.a) El cliente paga el alquiler (CU20 Pagar con tarjeta de crédito).
CU20: Pagar con tarjeta de crédito
– Escenario principal:
• ...

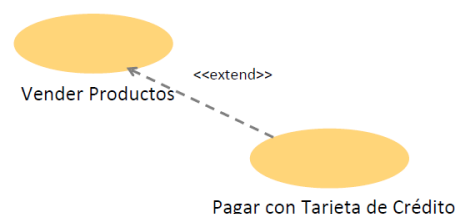


En este ejemplo vemos que se ha decidido factorizar el comportamiento relacionado con el pago mediante tarjeta de crédito, que se encuentra repetido en 2 casos de uso, lo hemos puesto en un caso de uso aparte que es invocado desde “Vender Productos” y “Alquilar equipos”. La relación se dibuja como una flecha con línea punteada que va desde el caso de uso base al caso de uso incluido, es decir, el caso de uso que necesita al caso de uso que provee esa funcionalidad.

Relaciones de extensión

Cuando en un caso de uso encontramos comportamiento adicional u opcional, también podemos crear un nuevo caso de uso para ubicar esas acciones y así simplificar la lectura del caso de uso base.

CU010: Vender Productos
– Escenario principal:
• 1) El cliente arriba al punto de venta con los productos para pagar.
• 2) ...
• 6) El cliente paga su compra con efectivo
– Escenarios alternativos:
• 6.a) CU030 Pagar con tarjeta de crédito
• ...



En este ejemplo, los usuarios normalmente pagan en efectivo, pero en determinadas circunstancias pueden hacerlo mediante tarjetas de crédito. Aquí se ha decidido extraer ese comportamiento y ubicarlo en un nuevo caso de uso, informalmente llamado **extendido**. Hay que tener cuidado porque aquí el sentido de la flecha es al revés que en la inclusión. “Pagar con Tarjeta de Crédito” extiende el comportamiento de “Vender Productos”, por eso se pone esa relación en ese sentido.

Relaciones de generalización

También es posible encontrar situaciones en donde es necesario especializar un caso de uso por alguna circunstancia en particular.

CU130: Realizar transferencia entre cuentas*

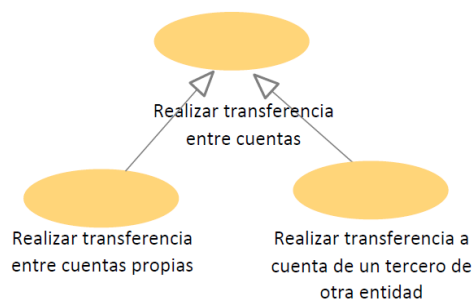
- 1) Ingresar número cuenta origen
- 2) Ingresar número cuenta destino
- 3) Ingresar monto a transferir
- 4) ...

CU131: Realizar transferencia entre cuentas propias

- 1) Ingresar número cuenta origen [I-UC130-1]
- 2) Ingresar número cuenta destino [I-UC130-2]
- 3) Ingresar monto a transferir [I-UC130-3]
- 4) ...

CU131: Realizar transferencia a cuenta de un tercero en otra entidad

- 1) Ingresar número cuenta origen [I-UC130-1]
- 2) Ingresar número cuenta destino (CBU) [S-UC130-2]
- 3) Ingresar monto a transferir [I-UC130-3]
- 4) Enviar notificación al tercero [N]
- 5) ...



Lo que vemos aquí es que hay un caso de uso **padre** con el comportamiento compartido entre 2 casos de uso **hijos**. No es algo que se utilice mucho, pero la posibilidad esta disponible. Lamentablemente, no hay consenso acerca de cómo especificar correctamente este tipo de casos de uso. Una alternativa es indicar en los casos de uso hijos, cuál es el comportamiento que se hereda sin cambiar, cuál se hereda pero se especializa y cuál comportamiento es nuevo propio del caso de uso hijo.

[V / F] La *inclusión* en casos de uso se refiere a la *repetición* de comportamiento *común* entre los casos de uso.

[V / F] La *extensión* en casos de uso se refiere a la *adición* de comportamiento entre los casos de uso.

Adicionales

Algunos temas adicionales:

- A partir de cada caso de uso, se pueden identificar **casos de prueba**: Qué es lo que se debe probar en cada escenario, bajo qué condiciones, con qué datos, etc. Si leemos detenidamente el caso de uso nos vamos a encontrar con que hay distintos caminos, distintas combinaciones. Básicamente el escenario principal nos va a determinar un camino de prueba posible, pero la combinación de ese escenario principal con los distintos escenarios alternativos nos da otros casos de prueba posible.
- También vamos a encontrarnos con que en los casos de uso se mencionan objetos de negocio u objetos de dominio, asociaciones, atributos o propiedades. Todos ellos que se mencionan en los casos de uso, deberían existir o **deben existir** en el modelo de dominio o en el modelo que se esté utilizando para representar ya sea el dominio o los datos.

Casos de uso CRUD / ABMC

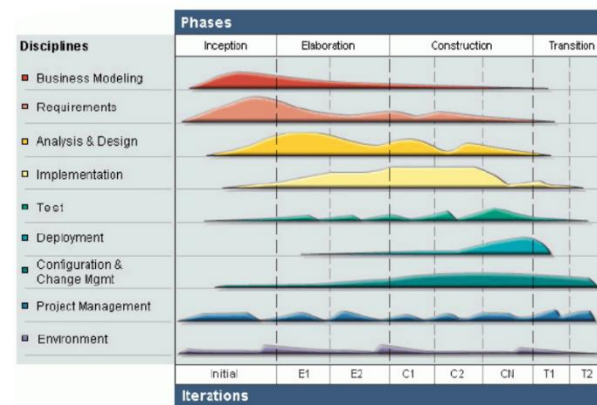
Otro tema bastante particular es el de los casos de uso CRUD (por Create, Read, Update, Delete), o en castellano, como se suele llamar ABM o ABMC que es Altas, Bajas, Multiplicaciones y Consultas. No hay consenso acerca de si debe haber un único caso de uso para hacer altas, bajas, modificaciones de una determinada entidad, o si va a haber un caso de uso para el alta, otro para la baja, otra para modificaciones, otra para la consulta. Ahí depende, claramente, del contexto y de cuánto se complique el modelo. Lo que sí es importante es que debe haber casos de uso que permitan custodiar, mantener, las entidades del modelo de dominio o las entidades del modelo de datos, dependiendo que otro modelo hayamos construido para representar los objetos del negocio.

[V / F] Todas las entidades y asociaciones referenciados en los casos de uso deben aparecer en el modelo de dominio o modelo de datos.

Casos de uso en el proceso de desarrollo

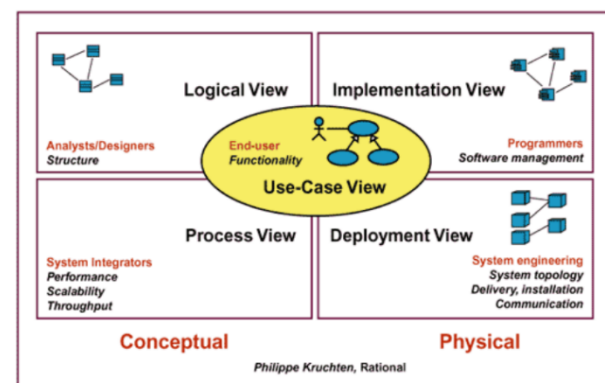
Proceso unificado

Bueno hablemos un poco de la historia de los casos de uso. Los casos de uso se popularizan a comienzos de la década de 1990 de la mano de Ivar Jacobson. Ivar Jacobson había escrito un libro muy conocido en esa época llamado “Object-Oriented Software Engineering” en el que además de presentar esta herramienta como una pieza central, describía una serie de disciplinas de ingeniería de software que, con el tiempo, constituirían un modelo de proceso de desarrollo de software llamado en esa época “objectory”, qué más tarde se incorporó al proceso unificado de desarrollo de software desarrollado por Booch, por Rumbagh, por Jacobson, los 3 también autores de UML. Este proceso unificado tenía o tiene una versión comercial llamada Rational Unified Process y actualmente IBM Rational Unified Process.

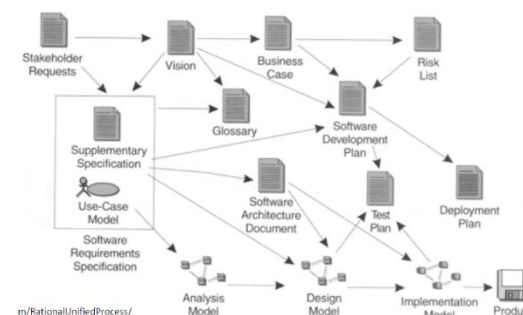


Proceso unificado

En el proceso unificado claramente los casos de uso juegan un papel fundamental, ya que son la piedra angular alrededor de la cual se organizan las diversas actividades y modelo. Philippe Krutchen, miembro del equipo de desarrollo del Unified Process, publicó un famoso paper acerca del modelo de arquitectura de software, conocido popularmente como “4+1”. En este paper, Krutchen sostiene que para poder representar adecuadamente los distintos aspectos de la arquitectura de un producto de software, es necesario emplear más de una vista, más de una perspectiva. Cada una de estas perspectivas está enfocada en un aspecto particular. La vista funcional, la vista central o la que identificamos como la “+1” en el modelo “4+1”, es representada en Unified Process mediante casos de uso.

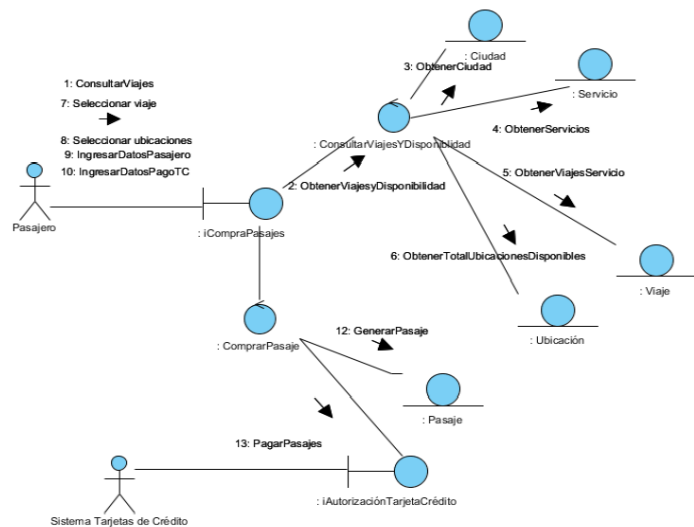


Esta perspectiva se presenta en el Unified Process mediante el modelo de casos de uso que está compuesto por uno o varios diagramas en notación UML y las especificaciones de cada uno de los casos de uso. También se propone una especificación suplementaria para poner ahí los requisitos no funcionales y quizás algún otro requisito funcional que no se pueda especificar mediante un caso de uso. Esa especificación suplementaria y el modelo de caso de uso constituyen la especificación de requisitos de software. En un entorno más tradicional, esto sería simplemente un documento con la descripción de cada uno de los requisitos funcionales y no funcionales que el sistema a construir debería satisfacer.



Modelo de análisis en el proceso unificado

Con esta información del Unified Process, propone desarrollar el **modelo de análisis que es una descripción de lo que ellos llaman la realización de cada caso de eso**. Una realización es la descripción del comportamiento interno del sistema en términos tecnológicamente neutros, necesario para elaborar las respuestas a las acciones de los actores que se describen en los casos de uso. En este modelo, el comportamiento se describe mediante lo que en Unified Process se llama objetos de análisis, que son abstracciones que tienen propiedades y comportamiento. Colaboran entre sí para elaborar las respuestas que necesitan los actores. Estos objetos de análisis pueden ser: entidades, equivalentes a nuestros objetos de dominio o entidades de datos, pero con el agregado de comportamiento; objetos de interacción, que son las interfaces con el mundo exterior; y objetos de control.



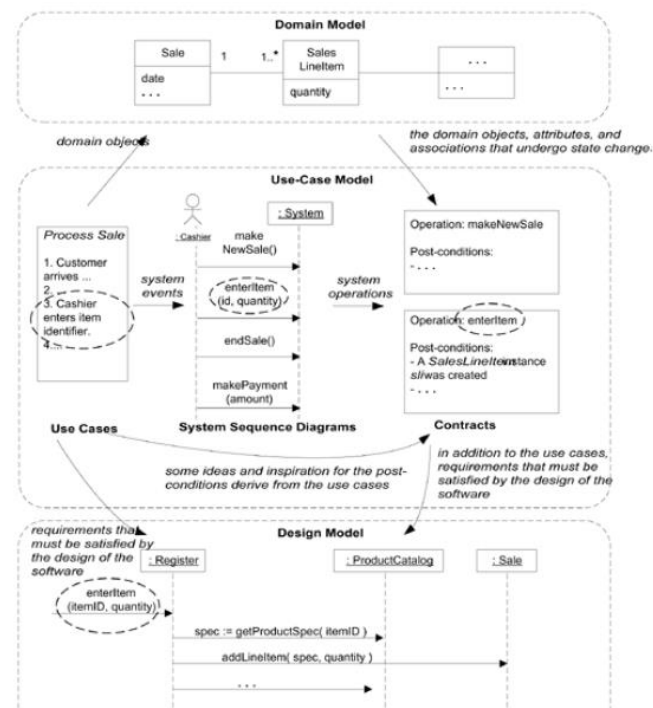
Lo que vemos en el gráfico es, para un caso de uso, cómo colaboran una serie de clases de análisis para poder elaborar las respuestas a cada una de las acciones del actor. A partir de este modelo de análisis se puede pasar al modelo de diseño en donde lo que vamos a encontrar son equivalentes, clases de diseño equivalentes, a estas clases de análisis que estamos viendo en el gráfico.

[V / F] El modelo de análisis representa una realización de un caso de uso.

Modelo de casos de uso

à la Larman

Una alternativa a este enfoque es el que propone Craig Larman, un autor muy reconocido y autor de varios libros de textos clásicos respecto al tema de procesos unificado y UML. Larman propone desarrollar, en lugar de un modelo de análisis como el propuesto por el Unified Process, **un modelo de caso de uso más completo que, además de los casos de uso propiamente dicho, contiene contratos y operaciones**. Todo esto constituye luego el input para las actividades de diseño. En esta propuesta nos encontraremos con diagramas de secuencia a nivel de sistema y contratos para cada una de las operaciones.



Contratos y operaciones

Entonces para cada caso de uso nos vamos a encontrar con lo siguiente:

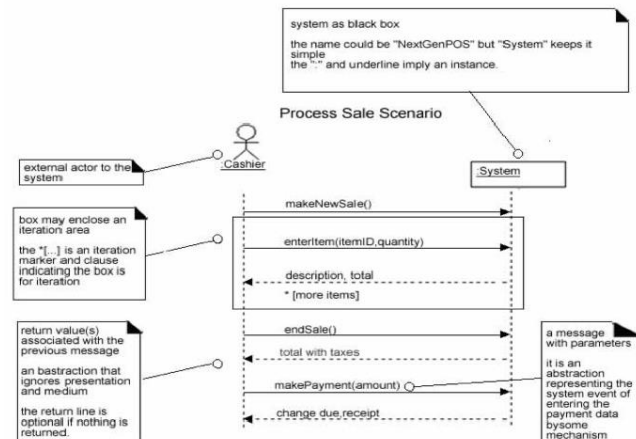
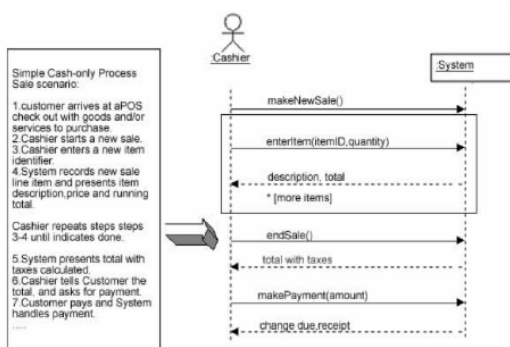
- ❖ Una operación se va a disparar como resultado de un **evento**, es decir, de una acción del actor. Cuando vemos una acción del actor en el caso de uso, en definitiva, para este modelo vamos a encontrarnos con un evento.
- ❖ Como resultado de la operación se pueden producir **cambios** en los objetos de **dominio**. Nuevamente, el modelo de dominio es una representación de los objetos y asociaciones del dominio del problema no es un modelo de base de datos. Entonces, cada vez que hay un evento, esos cambios pueden impactar en la creación de nuevas instancias o la creación de nuevas asociaciones entre objetos. Por ejemplo, si estamos describiendo el caso de uso correspondiente a la inscripción de un alumno en un curso, probablemente "Alumno", "Curso"

sean entidades del modelo de dominio y, probablemente, en ese caso de uso, se cree una instancia de un objeto de dominio llamado "Inscripción".

- ❖ Cada operación, al menos las más complejas, debe ser definida mediante un **contrato**.
- ❖ Lo que siempre tenemos que acordarnos, es que estamos hablando de la **esencia** del sistema, no estamos hablando de aspectos de implementación.

Diagrama de secuencia del sistema

Aquí podemos ver el ejemplo un poco más detallado. El sistema nuevamente es una caja negra, no describe cómo se elabora la respuesta. Fijémonos que en este diagrama de secuencia lo único que tenemos es el actor y el sistema. Las acciones del actor se representan como un mensaje que se envía del actor al sistema, las respuestas se dibujan como una flecha del sistema hacia el actor, como un mensaje del sistema hacia al actor (ahí cuando hace falta hacerlo, sino la respuesta está implícita).



Aquí tenemos otro ejemplo, tomado del libro del Larman. A la izquierda vemos el caso de uso, y para cada acción, vemos a la derecha como se dibujan los eventos del sistema.

Operaciones del sistema

Decíamos antes que cada evento va a disparar una operación. Esa operación es una operación si se quiere abstracta, conceptual. La vamos a necesitar especificar mediante un contrato. El contrato de operación va a incluir:

- ✎ Nombre de la operación y parámetros.
- ✎ Responsabilidades
- ✎ Precondiciones: estado del sistema o de los objetos del dominio antes de la ejecución de la operación.
- ✎ Postcondiciones: estado de los objetos en el modelo de dominio luego de hacerse ejecutado la operación (nuevas instancias o asociaciones creadas, cambios a los atributos, instancias o asociaciones destruidas).

Contract	
Name:	enterStudentIdentifier (studentIdentifier)
Responsibilities:	Accept and validate the Student's identifier. Verify that the Student is eligible to register.
Type:	System
Exceptions:	If the student identifier is not valid, indicate that it was an error. If the Student is not eligible to register, inform the Student.
Preconditions:	Student is known to the system.
Postconditions:	None

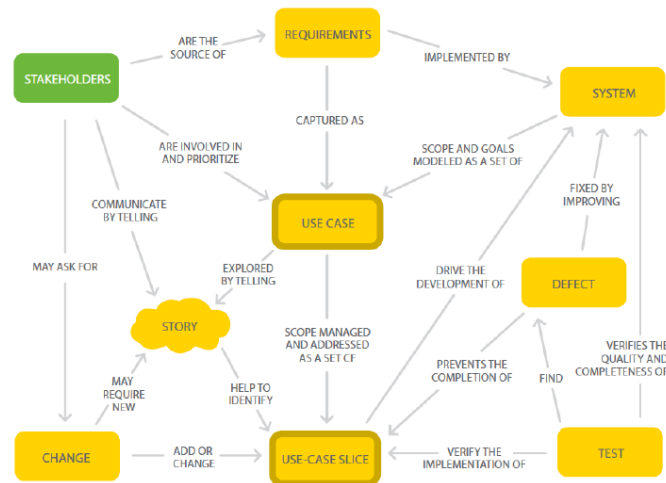
Repetimos que esta no es una descripción funcional, es una descripción esencial. Entonces, con esta descripción lo que estamos haciendo es complementando un poco qué es lo que hace el sistema. Convengamos que el modelo de casos de uso, los casos de uso en general, al tratar a los sistemas como una caja negra, no nos cuentan que es lo que debería estar pasando del lado de adentro del sistema, independientemente de cómo lo implementemos. Hay un comportamiento que el sistema tiene, para elaborar esas respuestas, que no tiene que ver con la implementación, tiene que ver con la lógica del sistema. Entonces si yo consulto los cursos disponibles, y el sistema me responde cuáles son los cursos disponible, ¿cómo hago yo para elaborar esa respuesta? Bueno, el sistema se tendrá que fijar cuáles son los cursos que están dentro de la oferta académica, cuál es la materia que yo estoy consultando, cuál es la materia de los cursos que quiero conocer y voy a tener que ver cuáles son los cursos que están habilitados, cuáles son los horarios, etc. Bueno, todo eso, si yo lo veo en un caso de uso, no me describe cómo pasa eso dentro del sistema, entonces con este esquema que estamos presentando, con este tema de las operaciones, o con el modelo de análisis que proponen en Unified Process, estamos de alguna manera, construyendo una especificación de lo que debe hacer el software, mientras que con los casos de uso estamos, en general, planteando una visión más desde el punto de vista del usuario.

[V / F] En el modelo propuesto por Larman cada operación abstracta se especifica a través de un contrato que es una descripción esencial.

Casos de uso

2.0

Si bien han pasado más de 20 años desde la invención de los casos de uso, aún se siguen produciendo desarrollos novedosos. Recientemente Jacobson y su equipo, han impulsado la adopción de lo que han de llamar casos de uso 2.0. La novedad más importante es que los casos de uso en esta nueva versión ya no son un conjunto de escenarios, sino un conjunto de historias, como las historias de usuario, que ayudan a organizar las porciones o slice en las cuales se puede dividir un caso de uso para poder organizar la implementación. Todavía es bastante prematuro hablar del impacto que podría llegar a tener esta nueva versión. Claramente vale la pena profundizar un poco en los detalles.



Conclusión

Bien para finalizar, hemos dado un panorama general de los casos de uso. Si bien parecen haber perdido protagonismo, hoy en día, todavía tienen su lugar en situaciones muy particulares. Por ejemplo, en aquellos casos en donde no se puede contar con un representante del usuario para que se sienta junto al equipo de desarrollo. También, en situaciones contractuales; también en situaciones en donde los equipos están geográficamente dispersos. Nuevamente hay que utilizar muy cuidadosamente los casos de uso. No todos los casos de uso merecen ser especificados con un gran nivel de detalle. Hay que tomar decisiones con respecto cuáles valen la pena profundizar y cuáles no. Sí es importante identificar la mayor cantidad de casos de uso posibles, identificar las relaciones entre esos casos de uso y que los casos de uso guarden consistencia con el resto de los modelos que construyamos.

[V / F] La novedad de los Casos de Uso 2.0 es la incorporación de *historias* y *slices*.