

## Introducción

Los CU forman parte de las especificaciones de UML 2.0, así como de metodologías de desarrollo, los mismos son empleados para la especificación de requerimientos funcionales.

Este documento está dirigido a los analistas de requerimientos y los revisores internos de los documentos relacionados con los CU con el objeto de establecer los criterios que permitan evaluar si los CU han sido escritos de forma útil, comprensible, portable, completa y sin ambigüedades.

Cada analista de requerimientos debe conocer profundamente el significado de escribir CU y el empleo que a éstos se les dará. Los casos de uso son una manera de capturar requerimientos, para ser expresados a los equipos de análisis, quienes encontrarán en estos su principal fuente para identificar objetos y clases.

## ¿Cuándo utilizar los Casos de Uso?

Los casos de uso son un tipo de requerimientos utilizados para especificar funcionalidad, especialmente en sistemas con un alto grado de interacción hombre/máquina (y pueden ser utilizados hasta en sistemas de batch). En esencia los casos de uso describen los intercambios entre el sistema que se está describiendo y las personas o sistemas externos que interactúan con el primero, por lo tanto, son muy útiles para describir funcionalidades a varios tipos de usuarios y con muchas interfaces

## ¿Para qué sirven los Casos de Uso?

Los casos de uso son útiles para capturar requerimientos, ayudar a definir la arquitectura, establecer las pautas para el diseño y las pruebas funcionales. Los CU son una guía de los elementos que serán incluidos en los documentos de usuarios para las aplicaciones, así como la forma en como éstos deben ser empleados. Los CU también establecen las bases para los protocolos de comunicación entre aplicaciones y el diseño de las interfaces gráficas, entre otros.

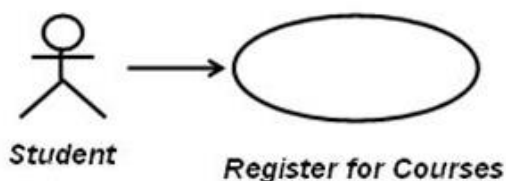
La validez de los casos de uso viene dada cuando los usuarios e involucrados del sistema aceptan el funcionamiento propuesto en los CU, siempre que la redacción de los mismo sea buena, no dejando cabida a ambigüedades.

Entonces los casos de uso deben ser útiles y ofrecer valor tanto al equipo de usuarios e involucrados como a los desarrolladores del proyecto.

## ¿Qué es un Modelo de CU y que son los CU?

Los casos de uso describen secuencias de acciones que realiza un sistema y que lleva a un resultado de valor a un actor específico.

Un modelo de CU está compuesto por dos partes, un diagrama (gráfico) y una parte textual. El diagrama muestra las relaciones entre actores y casos de uso, así como las relaciones entre los CU y entre actores (en caso de que existan). La parte textual muestra la descripción escrita en lenguaje natural que narra los pasos y demás características del caso de uso.



### ¿Qué son los actores y cómo identificarlos?

Actor es algo o alguien fuera del Sistema que interactúa con el Sistema.

Un actor especifica un rol que alguna entidad externa adopta cuando interactúa con el sistema directamente. Puede representar un rol de usuario o un rol jugado por otro sistema o hardware que toca la frontera del sistema.

La siguiente es la lista de preguntas que permiten identificar a los actores que interactuarán con el Sistema:

- ¿Quién o qué utiliza el Sistema?
- ¿Qué roles toman en la interacción?
- ¿Quién toma información del Sistema?
- ¿Quién provee información al Sistema?
- ¿En qué parte de la compañía es utilizado el Sistema?
- ¿Quién instala, soporta y mantiene el Sistema?
- ¿Quién inicia y termina la ejecución del sistema?
- ¿Qué otros sistemas utilizan el Sistema?
- ¿Ocurre algo en algún momento específico?

En la definición del caso de uso existe un apartado para definir los actores. La descripción se refiere a una breve reseña del rol del actor para el sistema y las fuentes son los involucrados del negocio que ayudaron a definir y describir al actor.

### ¿Qué son Casos de Uso y cómo identificarlos?

Los Casos de Uso representan la funcionalidad de lo que un actor desea que haga el Sistema. Los casos de uso definen una secuencia de acciones ejecutadas por un sistema que producen un resultado observable de valor para un actor.

En el libro “The UML Reference Manual” de RUMBAUGH se definen los casos de uso como: la especificación de secuencia de acciones, incluyendo variaciones a la secuencia y secuencia de los errores, que un sistema, subsistema o clase realizan con la interacción de actores externos.

Los casos de uso son siempre iniciados por un actor y son escritos desde el punto de vista del actor.

La siguiente es la lista de preguntas que permiten identificar los casos de uso dentro de las fronteras de un sistema:

- ¿Qué funciones del sistema son requeridas por un actor específico?
- ¿El sistema guarda o recupera información? Si es así ¿Qué actores disparan esta acción?
- ¿Qué ocurre cuando el sistema cambia de estado? ¿Algún actor es notificado?
- ¿Algún evento externo afecta al sistema? ¿Qué notifica el sistema respecto de estos eventos?
- ¿El sistema interactúa con algún sistema externo?
- ¿El sistema genera algún reporte?

### ¿Como especificar Casos de Uso?

La especificación de los casos de uso se refiere a la descripción de cada una de las partes definidas para lograr su descripción completa. En la organización, la especificación de los Casos de Uso se hará bajo el formato presentado a continuación. El siguiente cuadro muestra las partes y las indicaciones básicas para iniciar su redacción. En las siguientes secciones del documento se presentan las recomendaciones que hacen que la redacción de CU sea más fácil, sencilla de leer y escribir.

Esta definición es válida para casos de uso de alto nivel o casos de uso de negocio.

<b>CUXX – Nombre</b>	Cuxx = Código de Caso de Uso
<b>Descripción</b>	Enunciar la finalidad del caso de uso
<b>Actores que intervienen</b>	Enumeración de los actores que intervienen en el caso de uso
<b>Precondición</b>	La precondición está formada por el conjunto de condiciones que se tienen que cumplir para que se pueda iniciar un <u>caso de uso</u> . En muchos casos supone la ejecución de casos de uso previos.
<b>Resumen de la funcionalidad</b>	Se espera que en este apartado se explique de manera tecnológica que es lo que debe resolver la funcionalidad a alto nivel y que reglas de negocio se deben aplicar para poder implementar la misma.  Dicha descripción debe indicar claramente para cada uno de los actores definidos en dicha funcionalidad que tareas realiza cada uno de ellos de manera parcial o general dentro de la misma.
<b>Postcondición</b>	La postcondición refleja el estado en que se queda el sistema una vez ejecutado el caso de uso.
<b>Observaciones</b>	Las que considere pertinentes

## *Caso de Uso*

El nombre del CU debe comenzar por un verbo y ser lo más corto posible, pero que, a su vez, describa lo que el CU hace. El nombre del CU debe indicar el valor u objeto que genera para el actor. El nombre del CU comienza por su identificación CU.xx donde xx es el número asignado a este CU.

Pero por lo que veremos mas adelante los casos de uso no poseen un orden de ejecución entre uno y otro con lo cual ese número asignado es simplemente una referencia para ver cuantitativamente si todas las funcionalidades relevadas por el analista se encuentran definidas por casos de uso en el documento.

## *Actores*

Los actores que interactúan directamente con el sistema, tanto los primarios quienes inician el CU, como los secundarios que interactúan con el sistema luego que este ha iniciado. Cuando se nombran los actores en la sección de actores del CU. En caso de haber actores secundarios, el CU debe indicar cuales son secundarios, el resto se asumen primarios, de lo contrario se asume que todos son primarios. Los actores secundarios se les coloca luego del pseudónimo un guion y la palabra «secundario» en letra itálica.

### *Pre-condiciones*

Las precondiciones son elementos opcionales de los CU. Cada precondición listada tiene: 1) número, 2) título 3) descripción (opcional). El número es un consecutivo. El título debe resumir muy brevemente la condición. La descripción puede dar detalles de cómo la condición es evaluada, o el rango de valores que pueden ser aceptados para la condición.

Bittner y Spence hacen énfasis en que las precondiciones no son eventos que disparan o activan el CU en el sistema. Sin embargo, son declaraciones en las cuales el caso de uso puede comenzar. Las precondiciones son necesariamente ciertas para que el CU pueda comenzar, pero no suficientes. El CU sólo puede ser comenzado por el actor cuando las precondiciones son ciertas.

Las precondiciones no son evaluadas en la descripción de la funcionalidad del caso de uso pues se asumen como ciertas. Si las precondiciones aplican para todos los CU del sistema entonces es un requerimiento y no debe indicarse.

Para hallar precondiciones fórmúlese las siguientes preguntas:

- ¿En qué estado debe encontrarse el sistema para que el CU se pueda disparar?
- ¿Qué elementos deben estar presentes para que el CU pueda iniciar?
- ¿Cuáles son los supuestos asumidos?
- ¿Qué restricciones aplican al empleo del CU por los actores?

Note que las precondiciones son opcionales de no existir no agregue precondiciones que no aporten a la explicación del CU. Evite precondiciones como «El usuario debe estar vivo» o «La máquina debe estar encendida y el sistema debe estar activado y funcionando», pues no agregan valor al diseño y son obvias.

### *Resumen de la funcionalidad*

Es un párrafo que resume el objetivo del caso de uso. Sin dar detalles del cómo, la descripción del caso de uso resume todo lo que el caso de uso hace, que es el valor que da al actor (o actores) primario, así como la necesidad de la existencia de actores secundarios, para los casos que aplique.

Ejemplo. "El caso de uso permite al Administrador crear, modificar y borrar usuarios en el sistema" ó "El caso de uso permite al Cliente obtener la información sobre su estado de cuenta del Sistema de Manejo de Clientes".

### *Post-condiciones*

Las postcondiciones son elementos opcionales de los CU. Los elementos y patrones para describir postcondiciones son iguales a los de las precondiciones.

Bittner y Spence explican que las postcondiciones deben ser ciertas cuando el CU termina, sin importar el curso de eventos que se sucedieron. Si algo puede fallar Ud. debe cubrirlo en las postcondiciones para describir el estado en el que el sistema se encontrará cuando el CU se completa.

Las postcondiciones son importantes puesto que dan las luces sobre las condiciones que garantizan que siempre que termine el CU el sistema queda en un estado válido y los datos inherentes (en caso de existir) se encuentran consistentes. Las postcondiciones son igualmente útiles para verificar que las pruebas que se realicen sobre este CU aseguren que estas condiciones se darán al salir, sea cual fuere el curso de acciones tomadas.



Para hallar postcondiciones formúlese las siguientes preguntas:

- ¿En qué estado debe quedar el sistema luego que termina el CU?
- ¿Qué debe garantizarse cuando termine para que el sistema no quede inconsistente o no utilizable?
- ¿Cuáles son las únicas condiciones válidas en las que puede acabar una ejecución del CU?

Armour y Miller aclaran que tanto las precondiciones como postcondiciones se refieren a estados del sistema, no del ambiente fuera del mismo, ello debido a que las condiciones del ambiente no se modelan dentro del sistema.

### *Observaciones / Notas*

Las notas son elementos opcionales muy importantes de los CU pues reflejan el análisis y la comprensión del funcionamiento del sistema, más allá de la descripción de las interacciones entre los usuarios. Las notas en los CU se utilizan para plasmar explicaciones, detalles sobre la información, formatos de archivos que ya se encontraban establecidos y otros acuerdos con los que la aplicación debe cumplir. Es importante que todas las notas se encuentren referenciadas con algún elemento del Caso de Uso, ejemplo la descripción, las referencias justifican la existencia de la nota.

Las notas son características muy importantes de los CU puesto que permiten capturar conocimientos propios del sistema que no se está describiendo, que son útiles a la hora de hacer el diseño y la implementación, sin embargo, no representan una funcionalidad propia del CU que se está especificando.

Recuerde que, en su rol de analista de requerimientos, su misión es plasmar la mayor cantidad posible de detalle posible, ello debido a que es Ud. quién se encuentra más cercano a los funcionales que puedan estar revelando la información.

### La restricción ACID sobre los Casos de Uso

Pedimos que en su definición los casos de uso cumplan la condición de ACID. (Atómicos, Cohesivos, Aislados (Isolation) y Durables.

**Atómico:** Los casos de uso deben ser atómicos en su definición y conceptualización. Esto implica que la funcionalidad descrita por el caso de uso se cumple de manera total y completa o la misma no ocurre. Pero no puede existir ejecuciones parciales de la misma sin finalizar la totalidad de la funcionalidad ya que generaría inconsistencias funcionales. Ej Se transfiere dinero de una cuenta A a una cuenta B. El concepto de atómico es que o se realiza la transferencia en cuestión de forma completa o no ocurre la misma pero no puede retirarse plata de la cuenta A que nunca llegue a la cuenta B.

**Cohesivo:** Los casos de uso deben poseer una **única responsabilidad** dentro de la definición de la funcionalidad que resuelven. Esto implica que no pueden tener más de una y solo una funcionalidad definida dentro de su definición funcional. En caso de que un caso de uso no sea cohesivo el mismo tendrá acoplamiento funcional y dependerá de otros casos de uso para poder resolver el mismo. Ej. No es válido que un caso de uso cuya funcionalidad se confeccionar factura imprima la misma. Solamente con el concepto de cohesivo debe estar totalmente dedicado a confeccionar la misma.

**Aislado:** Dado las características de Atómico y Cohesivo los casos de uso deben ser aislados entre sí y mutuamente excluyentes. Esto quiere decir que los casos de usos son independientes y no hay dependencias ni orden de ejecución preestablecidas en primera instancia.



**Durable:** Implica que mientras las reglas de negocio no se modifiquen la funcionalidad definida por el caso de uso se preserva en el tiempo.

### Casos de uso de negocio (CUN)

Para Bittner et al. (2006) los casos de uso pueden ser utilizados para definir procesos de negocio, en estos casos son llamados casos de uso de negocio (CUN) o “Business Use Cases (BUC)”, este término fue introducido por Jaccobson. Los actores en el modelo de casos de uso de negocio son externos al negocio, ellos son clientes, accionistas, proveedores y otras partes que participan en el negocio en algún tipo de relación.

Los casos de uso de negocio sirven para clarificar el ambiente en el que un sistema será utilizado, ellos pueden ser útiles para:

- Clarificar el contexto del sistema y obtener el acuerdo de los requerimientos
- Cuando se están modelando más de un sistema para soportar uno o más procesos de negocio. En este sentido los CUN especifican el alcance y las responsabilidades de cada uno de los sistemas.
- En la construcción de aplicaciones que serán utilizados por varias organizaciones. En este caso muestran la flexibilidad de los sistemas para adaptarse a los procesos de dichas organizaciones.
- Para definir sistemas que serán utilizados en nuevas líneas de negocio.
- En proyectos complejos de reingeniería.

Los CUN muestran como los procesos son llevados a cabo por personas y los activos de la organización.

En la sección anterior se explicó que los CU, por ser herramientas para capturar requerimientos no tienen niveles. Trabajar con BUC y CU en un mismo proyecto no representa múltiples niveles de casos de uso, sin embargo, los BUC permiten identificar claramente, con base en un proceso de negocio, cuáles serían las oportunidades de automatización y en ese caso sería una entrada importante de información para desarrollar los CU con base en los requerimientos del negocio.

Es importante aclarar que, así como los CU no explican como el sistema realiza las operaciones ni la forma de las interfaces con el usuario, los BUC no revelan detalles de cómo los procesos son implementados por los actores, ni los artefactos de uso interno del proceso, más sí lo que hacen en el marco funcional.

### Recomendaciones generales

Esta sección contiene recomendaciones que deben ser consideradas a la hora de redactar, tanto documentos de este tipo, como cualquier otro documento para una organización.

### Requerimientos trazados

Los requerimientos son formas de enlazar los CU o especificaciones funcionales con el resto de las especificaciones no funcionales. Estas especificaciones o requerimientos deben encontrarse en el repositorio de requerimientos con la información completa de los atributos definidos. Aunque la trazabilidad con los requerimientos especiales se da por demostrada en el mapa de trazabilidad de los requerimientos, el tenerlos presentes en el CU hace más fácil su lectura para el diseñador e constructor.

Para los requerimientos se debe especificar el número, el título y la descripción del requerimiento (opcional). El título puede contener el código del requerimiento y el título del mismo según la definición que se establezca para las especificaciones no funcionales del sistema. La descripción puede omitirse puesto que se asume que la misma se encuentra en el repositorio de requerimientos, sin embargo, se pueden dar indicaciones adicionales de cómo el CU debe cumplir con este requerimiento.

### Empleo de estilos del procesador de palabras

Los procesadores de palabras cuentan en la actualidad con manejo de estilos o formatos preconfigurados de texto. Estos estilos, al igual que ocurre con los CSS (Cascade Style Sheets) en Web, hacen del documento un documento mucho más fácil de escribir y de modificar, puesto que con modificar el estilo se modifica todo el documento sin tener que identificar párrafo por párrafo donde se deben cambiar las fuentes, tamaños, etc.

### Redacciones cortas, concisas y completas

Como a Ud., el resto de las personas que leen estos documentos probablemente no tengan tiempo de leerlo, por lo tanto, considere escribir de forma de expresar todas las ideas relevantes sin emplear expresiones que abulten el texto que escribe sin agregar valor.

La forma estructurada de este documento tiene inmerso el espíritu de «solo escribir lo necesario y hacerlo sin ambigüedades». Procure mantener en mente esta expresión una vez que comience a escribir CU.

Lo otro que debe tener en mente es que alguien como Ud. va a leer este documento para hacer el análisis, implementar un sistema, escribir la documentación, identificar las pruebas y vender el funcionamiento del sistema, por lo que el documento debe ser: 1) lo suficientemente detallado para que el diseñador, programador, probador y documentalista no deba leer ningún documento no referenciado desde éste para realizar su trabajo en relación con el sistema, 2) lo suficientemente preciso para no aburrir a un usuario a quién se le explique cómo trabaja el sistema, sino que se obtenga de él el feedback que permita identificar los cambios necesarios en la fase de redacción de CU en vez de cuando se entregue el sistema, y 3) los suficientemente claro para que Ud. pueda entregar el documento y no tener que dar más explicaciones sobre lo que está escrito.

### Nivel de detalle esperado

Se espera que el nivel de detalle del caso de uso no contemple detalles sobre la interfaz, ni la arquitectura, ni el procesamiento interno de los datos cuando estos no sean concernientes a los involucrados. Los casos de uso presentan detalles de requerimientos, no de diseño, otras herramientas son utilizadas para el diseño, como el prototipo. Esta separación se realiza para hacer más fácil de entender los CU.

El caso de uso debe contener el detalle justo y necesario para explicar los intereses (requerimientos funcionales) del involucrado y entregar el sistema.

El nivel de detalle esperado para el caso de uso es el necesario para guiar al diseñador en la forma en que se desea presentar los datos y expresar los resultados, en ningún momento se debe limitar al diseñador en este aspecto. En caso de existir restricciones o requerimientos de algún otro tipo que limiten las opciones, los mismos deben ser referenciados en la sección de restricciones o requerimientos especiales según el caso.

Las siguientes palabras, como regla general, indican un nivel de detalle incorrecto: Base de Datos, archivo, etc., botón, HTML, http, SMTP, radio button.

### ¿Dónde colocar información necesaria sobre requerimientos de interfaz o arquitectura?

El documento que contiene los requerimientos debe contener esta información. Es deber del analista de requerimientos tomar en consideración todos los requerimientos recolectados y presentarlos de manera total y consistente para que el diseñador y desarrollador los puedan tener en cuenta a la hora de hacer su trabajo. Los requerimientos pueden ser presentados en el listado de requerimientos especiales.



### Creatividad

Debe recordar que Ud. Está escribiendo flujos que afectarán la experiencia del usuario en la herramienta, y que eso que se defina será validado con el usuario. A la hora de escribir, el CU hágase la pregunta ¿Qué es lo que el sistema debería hacer mejor? ¿Qué hay detrás de los requerimientos? ¿A qué están acostumbrado los usuarios? ¿Dónde el CU genera valor? ¿Las necesidades de los usuarios quedarán totalmente satisfechas con la solución propuesta? ¿Qué otra cosa se podría hacer para al sistema sea más usable?

Recuerde que cuando escribe los CU, Ud. Está escribiendo como el sistema va a trabajar y como la experiencia del usuario se verá afectada por el uso del sistema.

Sin embargo, debe tomar en cuenta que si bien el CU debe agregar valor a los procesos que el actor realizará con la herramienta, el mismo debe poderse implementar en los tiempos solicitados, que el analista de requerimientos no debe introducir requerimientos que no agreguen valor a la cadena.

Especificaciones fáciles pueden hacer sistemas fáciles de utilizar. Especificaciones complicadas, harán, con toda seguridad, sistemas complejos, lo cual incrementa la oportunidad de rechazo del sistema por los usuarios, agregando a todo el equipo la posibilidad de fracaso, pues su éxito está atado al éxito de la experiencia que los usuarios con el uso del sistema que se está analizando.

### Nombre de los actores en todas las secciones del caso de uso

Cuando el CU especifica los pasos que el actor está realizando en el sistema, el CU debe indicar claramente quién es el actor, no deben emplearse expresiones como: «El actor hace.... » en vez debe especificarse el nombre completo del actor el pseudónimo. Nótese que el pseudónimo puede ser el mismo en varios CU para hacer referencia a actores diferentes, entonces se debe estar claro que el alcance del pseudónimo se restringe, de manera estricta, al CU.

Los nombres de los actores siempre comienzan en mayúscula y todas las palabras que los componen comienzan en mayúsculas, como el formato de cualquier nombre de persona (menos los artículos, ejemplo: de, la, el, para, del, con, etc.