

Consideraciones para redactar casos de uso según OMT++

Héctor Antillanca E.

Los casos de uso en OMT++ son ejemplos de uso de un sistema de software. Un caso de uso describe en un alto nivel de abstracción el cómo el usuario hace algo con el sistema para lograr un objetivo. El nivel de abstracción es importante, pues el detalle sobrecarga la descripción con decisiones de diseño que se pueden tomar más adelante. El escritor trata de incluir en cada caso la mayor variedad de usos posibles en un contexto de trabajo dado de un futuro sistema. No es un manual de uso ni un procedimiento determinista sobre como hacer las cosas. El caso de uso menos aprovechado es el caso que describe sólo una operación del sistema, *es deseable que el caso incluya varias operaciones relacionadas*, consistentes con el objetivo que se plantea al definir el caso de uso. Una buena práctica es disponer de una página completa para especificar cada caso de uso. Si ocupa menos de una página el caso de uso se considera flaco. Si ocupa más de una página el caso de uso se considera gordo.

A continuación se describen los campos requeridos de una plantilla de caso de uso basado en la teoría de casos de uso de OMT++.

- **Título.** Se recomienda escribir el título en gerundio, de modo que sugiera al lector que lo que se describe es una secuencia de eventos que está ocurriendo ahora en el presente, tal como un guión de una obra de teatro que describe el dialogo entre dos actores, el que puede ser actuado por un usuario.

- **Resumen.** Incluye un resumen del caso de uso, o el objetivo que logra el usuario al jugar su rol en el caso de uso.

- **Lista de actores del caso de uso.** Tipos de usuarios a los cuales les está permitido jugar el rol de usuario en el caso de uso.

- Precondiciones:

Es una lista de *aseveraciones* sobre el estado del sistema. Los objetos que mantienen el estado del sistema podrían ser examinados para verificar la validez de la condición. Ejemplos: hay una sesión activa del usuario, hay usuarios conectados, existe un registro de nombres de usuarios válidos del sistema, hay carpetas que contienen archivos de tipo documento.

Cada precondición establece condiciones que deben cumplir los objetos a ser manipulados por la aplicación durante el caso de uso. No se recomienda incluir condiciones que deben ser cumplidas por el usuario, porque no aportan al diseño de la solución, por ejemplo, que el usuario sea un adulto mayor, es mejor incluir requisitos no funcionales que satisfagan condiciones o características de los usuarios.

La sección de precondiciones no es una especificación de deberes del usuario, por ejemplo, no es apropiado poner: el usuario debe estar autenticado, o el usuario está leyendo la ventana principal; más apropiado poner: el usuario está autorizado

o, el usuario está conectado o, el usuario tiene una sesión abierta o, la ventana principal (o cualquiera otra ventana) está activa o, el identificador del usuario pertenece al conjunto de los usuarios validos, etc.

No es una especificación de lo que desea hacer el usuario, por ejemplo, el usuario desea hacer cambios en el documento de trabajo.

- Poscondiciones:

Igual que las precondiciones, es una lista de aseveraciones.

Establece condiciones que deben cumplir los objetos que fueron manipulados por la aplicación durante el caso de uso (estado del sistema resultante).

No es necesario referirse a las acciones realizadas (en el pasado) por el usuario durante la ejecución del caso de uso, pues esta información ya está incluida en la descripción. Importa más que el estado pueda verificarse, especialmente en la etapa de prueba del sistema cuando se requiere comprobar que el caso de uso funciona.

El efecto del caso de uso se deduce contrastando estas condiciones con las precondiciones.

-Excepciones:

No explica la causa de la excepción. Debe indicar lo que hace el usuario en el caso que ocurra la excepción, en tiempo presente. Es una continuación de la descripción del caso de uso.

En algunos casos, una excepción puede implicar que el usuario tenga que ejecutar otro caso de uso.

Debe ser consistente con la descripción del caso de uso y las precondiciones. No es un procedimiento o una especificación de deberes del usuario.

-Descripción:

Debe ser consistente con las precondiciones.

Usa tiempo presente en la narración.

Es una historia feliz. Supone que todo funciona sin contratiempos. Describe lo que ocurre en la interacción con afirmaciones fuertes, que no dan lugar a dudas sobre lo que los actores hacen.

Cabe en media página aproximadamente.

No describir requisitos funcionales, por ejemplo, “el sistema le permite al usuario ordenar los objetos según algún criterio de interés”, o “el usuario puede ordenar los objetos por algún criterio de interés”.

No describe requisitos no funcionales, por ejemplo, “es obligatorio que el usuario ingrese información en los campos X, Y y Z”.

No es una especificación de lo que el usuario desea hacer.

Excluye expresiones ambiguas. Por ejemplo, el usuario *puede* buscar un objeto si lo desea cuando aparece una lista muy larga.

No hay sesgo de implementación. Por ejemplo, hay sesgo de implementación si escribe “el usuario abre un menú con un clic del botón derecho del mouse, luego se posiciona en la opción X y hace dos clics seguidos”. Las decisiones de diseño de la interacción del usuario con la interfaz se postergan hasta la etapa de especificación de la interfaz.

No introduce explicaciones del por qué se hace lo que se hace.

No incluye condiciones, por ejemplo, “si el usuario hace x entonces...”. No incluye posibilidades, por ejemplo, “el usuario podría hacer X”. Tampoco se incluyen deberes. Estos elementos aumentan la ambigüedad del caso.

- Requisitos no-funcionales:

Opcionalmente se pueden incluir requisitos no funcionales que se desprenden del caso de uso, como información complementaria del caso de uso. Por ejemplo, aquí se pueden incluir los requisitos de usabilidad.

Comentarios:

No seguir las prácticas anteriormente recomendadas no significa que el objetivo de especificar una solución no se logre, pero se hará con una sobrecarga de trabajo innecesaria para el analista y los que reciban los casos de uso como un insumo en las siguientes fases del desarrollo, pues tendrán que hacer un trabajo de destilado de la funcionalidad de la aplicación que pudiera estar inmersa en la narrativa de los casos de uso.

Los novicios tienden a redactar casos de uso que incluyen con detalles de diseño, en el intento de describir acciones precisas del usuario sobre la interfaz. Se recomienda concentrarse más en una descripción abstracta, simple y fácil de leer, describiendo qué hace el usuario con el sistema. Si el caso describe detalles de diseño, se reduce el espacio de soluciones posibles que el diseñador podría ofrecer para hacer una implementación. Por otro lado, hacer casos de uso flacos, aumenta la documentación para describir la solución. Además, se pierde la oportunidad que otorgan los casos de uso para detectar ciertas operaciones y por lo tanto tener un contexto que justifique la existencia de ellas. Para redactar casos de uso ricos en funcionalidad es necesario que el escritor haga un esfuerzo por imaginar escenarios donde el usuario logra objetivos significativos para su negocio, de no hacerse, ciertas operaciones necesarias pueden aparecer más tarde, cuando el costo de su implementación sea más caro.