

Consejos Para Escribir Buenos Casos de Uso

Introducción

Escribir buenos casos de usos es más un arte que una ciencia. Y como en cualquier arte, no existen reglas absolutas para crear obras maestras. Finalmente, los casos de uso consisten en comunicar de forma clara información detallada a diversas audiencias, y alcanzar el objetivo de crear desarrollos de proyectos exitosos.

Ivar Jacobson inventó los Casos de Usos. María Ericsson trabajó junto a Jacobson y fue la coautora de uno de sus libros. Kurt Bittner junto a Ian Spence también escribieron un popular libro sobre Casos de Uso. Estos pioneros y mucha gente de IBM, que tienen años de experiencia trabajando con clientes para desarrollar buenos casos de uso, han contribuido a las tecnologías descritas en este documento.

Escribir un buen caso de uso no es fácil, pero, afortunadamente, nuestra experiencia puede ser su guía.

Los conceptos y principios aquí reunidos representan el trabajo de muchas personas en IBM, **los cuales forman un fundamento de mejores prácticas probadas**. Muchos de los consejos a los que hace alusión este documento son parte de la metodología RUP de IBM, otros son nuevos y no han sido incluidos dentro de RUP.

Entendiendo qué es un Caso de Uso (y qué no lo es)

Los Casos de Uso se han vuelto muy populares para establecer el esquema de la lógica de negocios del desarrollo de proyectos.

Esta popularidad, desafortunadamente, puede resultar también en confusión y en variedad de opiniones respecto a qué son y a cómo deberían ser estructurados y escritos.

Sin un estilo estándar y sin el establecimiento de pautas para escribir casos de uso, los escritores de casos de uso dentro de su organización escribirán los casos de uso a su manera, posiblemente confundiendo a los lectores y haciendo peligrar los proyectos así como también al **resultado deseado de una solución de calidad que provea valor de negocio inmediato.**

Los casos de uso son parte del Estándar Del Lenguaje Unificado de Modelado del Grupo de Gestión de Objetos (OMG).

Este estándar nos dice qué significan las partes de los diagramas de casos de uso, las figuras de palos **(monitos)**, óvalos y líneas, y nos da una definición de un caso de uso. Pero no nos dice cómo estructurarlos o escribirlos. Por lo que debemos leer libros o artículos (como este), para tratar de imaginarnos la forma correcta.

Así que, ¿cual es la forma correcta?, La mejor forma es la que funcione mejor para ti, sin salirse mucho de la definición de lo que es un caso de uso:

Un caso de uso es una especificación de una serie de acciones realizadas por un sistema, el cual produce un resultado observable que es, típicamente, de valor para uno o más actores u otros grupos de interés dentro del sistema (UML 2).

Una definición menos formal podría ser la siguiente: *Un caso de uso es una lista de pasos que especifican como un usuario interactúa con el negocio o sistema, teniendo en cuenta el valor que esta interacción le da al usuario o a otros grupos de interés. Aún más simple: Un Caso de uso es la historia de cómo un negocio o un sistema interactúa con el usuario.*

Un caso de uso no debe ser confundido con la historia de un usuario, por ejemplo un requerimiento del sistema de software formulado en una o dos frases en el lenguaje cotidiano del usuario. Los casos de uso son requerimientos formales con contexto y estructura que definen claramente el valor resultante.

Esto aún deja a los escritores de casos de uso un gran trabajo, descubrir realmente la mejor manera de escribir casos de uso, dada la significativa ambigüedad incorporada en la definición. El objetivo de este documento es ayudarle a tomar las decisiones correctas para que pueda realizar de forma exitosa ese trabajo.

Otro tipo de caso de uso – el caso de uso de negocio- es usado para desarrollar modelos que describen como un negocio interactuará con sus clientes y con otros grupos, y darle valore a esas entidades. Este documento no está orientado a los casos de uso de negocios y se centra exclusivamente en los casos de uso de sistemas.

Los Casos de uso son requerimientos de software que especifican funcionalidad

Las personas a menudo hablan de requerimientos y de casos de uso, implicando que son cosas diferentes. Los casos de uso son simplemente un método diferente para organizar los requerimientos funcionales, ellos aún son requerimientos.

Tradicionalmente, los requerimientos capturan la funcionalidad de un sistema usando una larga lista de declaraciones “deberá”, muchas veces cercanas a las miles. El objetivo de crear un caso de uso es transformar estas declaraciones “deberá” en pequeños grupos que provean valores y contextos observables, organizados desde una perspectiva del usuario.

Los casos de uso se refieren a comportamientos (al menos las salidas) que son observables por el usuario. Existen otro tipo de requerimientos aparte de los funcionales, usualmente llamados no funcionales o requerimientos suplementarios. Estos requerimientos son normalmente especificados como declaraciones “deberá”, pero ellos contienen información diferente a la que están en los casos de uso. Ellos describen la calidades necesarias del sistema:

cuan rápido debe ser, cuan confiable debería ser, cuan seguro debe ser y otra cualidades. Los casos de uso necesitan describir los que el sistema “debería” hacer. Ellos deberían expresar los pasos interactivos que un usuario debería tomar para obtener un resultado valorable.

Un caso de uso describe funcionalidad pero también debe describir un resultado. Esto es lo más importante, y está mencionado en la primera definición que los casos de uso dan “Valor a uno o más actores o grupos de interés del sistema”. Uno de los problemas observados más comunes en organizaciones que están recién comenzando con casos de uso es que ellos no entienden esta focalización en el valor. Y, como resultado, terminan escribiendo casos de uso más pequeños que no incrementan a la historia total. Ellos no se centra en las interacciones que conducen valor y que son de interés para los actores involucrados en los casos de uso.

Este acercamiento puede ser visto usualmente en la fase de pruebas de un proyecto. Si los casos de prueba están basados en casos de uso que reflejan unidades de prueba, en vez de pruebas de aceptación del usuario, probablemente existan muchos y sean de un nivel muy bajo.

Como ejemplo, un cliente con el que IBM se ha involucrado, tenía alrededor de 10 personas trabajando en un proyecto que estaba programado para durar alrededor de un año. Inicialmente, la organización tenía aproximadamente 150 casos de uso, porque había una falta de entendimiento de su propósito. Después de una examinación más profunda, se determinó que muchos de sus casos de uso eran realmente especificaciones de diseño, no requerimientos, y otros estaban centrados en interacciones tan pequeñas que nos mostraban ningún valor al actor o al grupo de interés.

Se tuvo que realizar un esfuerzo para tomar un paso atrás y centrarse en revisar sus casos de uso. Después de remover los casos inapropiados y combinando otros, el cliente terminó con menos de 10 casos de uso, cada uno centrado partes complementarias o comprensibles de la visión general, y mostrando valores significantes para cada actor o grupo de interés. Para este cliente, fue un momento de “ajá” cuando los desarrolladores de la compañía vieron como este nuevo, gran, y más completo caso de uso, serían más fácil de usar y de un gran uso para el diseño, desarrollo y prueba del sistema.

Consejo: No olvide el Diagrama

Los casos de uso son representados gráficamente como óvalos en los diagramas de casos de uso, y estos son también expresados como especificaciones textuales (ver Figuras 1 y 2). El texto es definitivamente la esencia del modelo de caso de uso, y ciertamente puede beneficiarse de escribir el texto sin dibujar el diagrama. Sin embargo, el diagrama puede ayudar a comunicar los requerimientos a un alto nivel. También ayuda a los grupos de interés a ver qué está dentro del alcance y qué está fuera. Los actores están fuera del sistema que está siendo desarrollado, y los casos de uso están dentro. Esto es especialmente importante para el diagrama de reconocimiento de los casos de uso, el cual muestra a todos los actores y a los casos de uso en un solo diagrama.

Consejo: Los actores humanos son el elemento más importante

Las figuras de palo utilizadas en los diagramas de caso de uso son aludidos como los actores. Ellos indican cualquier persona (o rol), otro sistema o quizás inclusive un dispositivo fuera del sistema que se está construyendo y que interactúa con él. Es el actor humano el que representa las interacciones mas interesantes y difíciles para la mayoría de los sistemas, y las interacciones humano/sistema son donde el verdadero valor de los casos de uso viene a la perspectiva. La funcionalidad puesta en un caso de uso ayudará a los actores a realizar su trabajo o tarea mejor- en una forma significativa.

A menos que el sistema que esté diseñando tenga una fuerte dependencia en otros sistemas o dispositivos, su tiempo y esfuerzo es mucho mejor que los utilice en orientarse a las necesidades de los actores humanos.

Consejo: Vaya con la corriente

Buenos casos de uso no son sólo párrafos de texto claro y conciso. Ellos poseen una estructura que definen como las partes de un caso de uso calzan juntas. Y ellos pueden ser estructurados en muchas formas- referidos como un estilo de casos de uso.

Una técnica estructural común al crear casos de uso es tener una larga lista de pasos de principio a fin. Este acercamiento puede causar problemas, ya que existen comportamientos condicionales y/o procesos de error que se benefician de la modularización – rompiendo cada pieza lógica para una parte específica de un caso de uso. Modularizar facilita escribir y leer, pero sólo si las piezas calzan juntas para describir la imagen completa.

Un caso de uso debería tener un único flujo principal y múltiples flujos alternativos. El flujo principal (ver figura 1) explica que sucede cuando todo sucede de forma correcta y el caso de uso cumple con su objetivo. Esto es llamado a veces el escenario del “camino feliz”, y ocurre la mayor parte del tiempo. A veces los escritores de casos de uso tratan de tener múltiples flujos principales, pero esto puede diluir el entendimiento del caso de uso ya que hace más difícil al lector el trabajo de entender el objetivo real.

Una buena y breve descripción ayuda a su lectura

Opciones del Actor:
Una escogida para el flujo principal o básico

Especificación de Casos de Uso – Registro para Cursos

Descripción Breve

Este caso de uso permite al estudiante registrarse a cursos ofrecidos en el presente semestre. El estudiante puede también modificar o borrar selecciones de cursos si se presentan cambios, adentro del período agrega/elimina al principio del semestre. El sistema de Catálogos de Cursos provee una lista de todos los cursos ofrecidos durante el presente semestre.

Actores

1.- Actor Primario – Estudiante

2.- Actor Secundario – Sistema de Catálogos de Cursos

Flujo de Eventos

1.- Flujo Principal

1.1 *Iniciar Sesión*

Este caso de uso inicia cuando un estudiante accede al Sistema de Registro de Cursos. El estudiante ingresa su ID y contraseña, el sistema valida al estudiante.

1.2 *Crear Horario*

El sistema despliega las funciones disponibles al estudiante. Estas funciones son: Crear Horario, Modificar Horario y Borrar Horario. El estudiante elige "Crear Horario".

1.3 *Seleccionar Cursos*

El sistema obtiene una lista de cursos disponibles ofrecidos desde el Sistema de Catálogos de Cursos y despliega esta lista al estudiante. El estudiante selecciona 4 cursos primarios ofrecidos y 2 cursos alternativos ofrecidos de la lista de cursos disponibles ofrecidos. El estudiante puede agregar y borrar cursos como desee hasta que elija enviar (submit) el horario.

1.4 *Enviar (Submit) Horario*

El estudiante indica que el horario está completo. El Sistema valida los cursos seleccionados y despliega el horario al estudiante. El sistema despliega el número de confirmación para el horario. El sistema guarda la información del horario del estudiante. El caso de uso termina.

Sin Referencias en el Flujo Principal

Los pasos deben tener nombres cortos

Flujos alternativos (ver figura 2) explican que sucede cuando algo causa una desviación desde el flujo principal. Estas desviaciones pueden ser o no etiquetadas como excepciones o errores. De cualquier manera, ellos no ocurren tan seguidos o no son tan importantes como el flujo principal. En un documento de caso de uso, cada tipo de flujo es listado en su propia sección. A veces los flujos alternativos están divididos en categorías más detalladas tales como errores de usuario y excepciones. Fíjese que los errores de código de bajo nivel están lejos del alcance de flujos alternativos.

Los Flujos tienen
Nombre

2.1 *Salir y Guardar*

El estudiante elige salir creando un horario y elige guardar un horario parcial antes de salir. Todos los cursos que no están marcados como “matriculado” son marcados como “seleccionados” en el horario. El sistema guarda el horario. El caso de uso termina.

2.2 *No se puede matricular*

En el caso de uso básico, en el paso Enviar Horario, el sistema determina que los prerequisites para un curso seleccionado no se cumplen o que el curso está lleno, o que existen conflictos en el horario, el sistema no matriculará al estudiante en el curso. El sistema desplegará un mensaje al estudiante y el caso de uso termina.

2.3 *Catálogo de Cursos no disponible*

En el caso de uso básico, en el paso Seleccionar Cursos, el sistema determina que el sistema de catálogo de cursos no está disponible. El sistema despliega un mensaje de error y el caso de uso termina.

2.4 *Registro Cerrado*

En el caso de uso básico, en el paso iniciar sesión, el sistema determina que el registro está cerrado; el sistema indica al estudiante que ya no puede seleccionar cursos y el caso de uso termina.

Un caso de uso con un solo flujo permanece inconcluso, ya que un buen caso de uso tendrá múltiples flujos o eventos. Recuerde que los casos de uso están hechos supuestamente para explicarle a los clientes, usuarios y equipos de desarrollo los requerimientos funcionales con un detalle suficiente para permitir a esos usuarios ya sea, aprobar o utilizar los casos de uso para diseñar

el sistema. Debe entender que los casos de uso no son un documento de diseño de cómo los requerimientos deben ser implementados; en cambio, es una indicación de lo que se necesita que sea construido. De esta manera, un solo flujo no es suficiente para mostrar el detalle necesario, o puede significar que existen muchos otros casos de uso más pequeños que podrían ser combinados en otros flujos alternativos.

Las referencias permiten a los usuarios reconstruir la historia completa

Es mejor práctica modularizar un caso de uso a través de su división en múltiples flujos. Pero para ser capaz de contar una historia completa, debe tener además alguna manera de colocar las piezas juntas otra vez para ilustrar los escenarios específicos de fin a fin. Esto es realizable utilizando referencias, lo que esencialmente define el comienzo y fin de un flujo alternativo, al señalar un paso en particular en el flujo principal o en otro flujo alternativo (ver figura 3).

Realizar un caso de uso fácilmente entendible requiere de una técnica consistente para la utilización de referencias.

Consejo: Colocar referencias en los flujos alternativos, no en los flujos principales

El flujo principal muestra que sucede cuando todo en el caso de uso sucede de la forma correcta, y el usuario alcanza su objetivo. Incluye una serie de pasos (eventos) ordenados en orden cronológico que explican los que el sistema hace y lo que el actor realiza para llevar a cabo el caso de uso. Las referencias son simples sentencias sobre donde inicia un flujo y como y donde termina.

Es recomendado que no utilice referencias en el flujo principal, que podrían llevar al usuario a un flujo alternativo o a otro caso de uso si una condición específica ocurre. Centrarse en el “camino feliz” mantiene el flujo simple y fácil de leer, y comunica claramente qué sucede cuando el caso de uso ha sido exitoso. Algunos estilos de casos de uso utilizan referencias en el flujo principal, los que usualmente se muestran como sentencias “if –then” y como ramas de muchos otros lugares en el caso de uso o fuera de él. Pero esta técnica puede resultar confusa, y es recomendado que todas las referencias ocurran en flujos alternativos.

Los flujos alternativos contienen las referencias (los flujos alternativos explican que sucede cuando ocurren errores o cuando ocurren resultados esperados, sin embargo ellos no son tan importantes como el flujo principal). Aquí es cuando aparece la importancia de las referencias. Cada flujo alternativo comienza en algún otro lugar dentro del caso de uso. Puede ramificarse en alguno de los pasos del flujo principal o extenderse desde otro flujo alternativo. Y cuando un flujo alternativo termina, el caso de uso usualmente continuará en algún otro lado.

Los flujos alternativos explican que causó su inicio y que es lo que el sistema hace en respuesta al flujo alternativo. Los flujos alternativos no sólo contienen las referencias de donde comenzaron y donde se reanudan, ellos también dicen qué causó su inicio y qué es lo que el sistema hace en respuesta. Mirando río abajo en el proceso de desarrollo, esta información va a ser útil cuando se escriban los casos de prueba.

Consejo: Los escenarios cuentan la historia completa

Los actores no pueden ejecutar flujos alternativos, pero pueden ejecutar escenarios. El concepto de escenario puede ser confuso porque las personas definen los escenarios de maneras muy distintas. En IBM, nosotros los definimos simplemente como los caminos a través de un caso de uso o como una serie de pasos y flujos que un actor podría tomar – de inicio a fin- a través de un caso de uso para proveer un resultado. Dados los muchos flujos alternativos posibles, habrá muchos escenarios, con muchos resultados – ambos, positivos y negativos-. El proceso de avanzar a través de los escenarios e identificar estos resultados negativos, brindan grandes plataformas para conversaciones de los grupos de interés, mientras los equipos tratan de satisfacer los requerimientos del negocio a través de todos los posibles escenarios. Tenga en mente que una completa serie de escenarios cuentan la historia completa de un caso de uso. Es esta completa serie de escenarios lo que será utilizado por los diseñadores y “testeadores”, como la base de su trabajo.

Consejo: Sea cuidadoso con las sentencias “if”

Los profesionales de las Tecnologías de Información (TI), particularmente aquellos con experiencia en programación, están familiarizados con las sentencias “if”, y por lo general puede observar dichas sentencias usadas en casos de uso. Un “if” en lenguajes de programación especifican comportamientos condicionales, y eso usualmente es verdad en los casos de uso también. Los problemas surgen cuando existen uno o más “if” en un flujo, ya que usualmente significa que se están especificando múltiples requerimientos. Esto puede resultar negativo para la legibilidad - puede resultar difícil seguir un flujo de múltiples sentencias “if” a través del texto. También resulta negativo para el diseño y prueba del sistema. Es mejor, tanto para el lector, como para el proyecto, que rompa cada “if” en su propio flujo. Se obtienen más flujos, pero el cambio vale la pena.

Consejo: Especificar elecciones (opciones) del actor

Los actores generalmente realizan elecciones en los casos de uso. Considere el ejemplo de un sistema de registro de una universidad siendo definido en caso de uso "Registro para cursos". En este ejemplo, un flujo principal podría ser: 1) Crear un horario, 2) modificar un horario y 3) Eliminar un horario. A menudo los escritores de casos de uso tratarán de usar sentencias "if" para mostrar al actor tomar una elección, pero por razones indicadas anteriormente, está podría no ser la mejor idea. Una mejor alternativa es tener todas las elecciones (opciones) del actor listadas en el punto apropiado, y tener una opción seleccionada y asignada al flujo actual y las otras repartirlas en flujos alternativos. Esta técnica mantiene cada flujo simple y reduce las ramificaciones.

Consejo: Aleja el CRUD

A menudo vemos a escritores dirigir las opciones del actor creando casos de uso separados para cada opción. Es una solución tentadora pero a la vez peligrosa. El peligro es la necesidad de manejar la complejidad exponencial que a menudo se crea, resultando en comportamientos repetitivos innecesarios, y, a la larga, casos de uso con valor mínimo.

En vez de eso, trate de centrarse en lo que el actor realmente quiere hacer, lo que generalmente no tiene relación con las acciones de Crear, Leer, Actualizar y Borrar (CRUD).

En nuestro ejemplo del sistema de registro de la universidad, hay opciones para 1) Crear un Horario, 2) Modificar un Horario y 3) Borrar un Horario. Todas están en el caso de uso “registro para cursos” porque al estudiante realmente no le interesa crear o modificar horarios, el estudiante sólo quiere registrarse. Tomando el camino fácil y listando todas las actividades CRUD, da como resultado 3 hasta 4 veces más casos de usos de los necesarios, y puede fácilmente hacer el proceso más complicado y difícil de manejar.

Para un ejemplo de un modelo de caso de uso con mucho CRUD, mire el ejemplo más debajo de un proyecto para actualizar la administración de una base de datos de un programa de refugios del gobierno local. Si sacamos todas las actividades de tipo CRUD, ¿cuáles son los reales beneficios para el negocio, derivados del esfuerzo de definir este modelo específico?

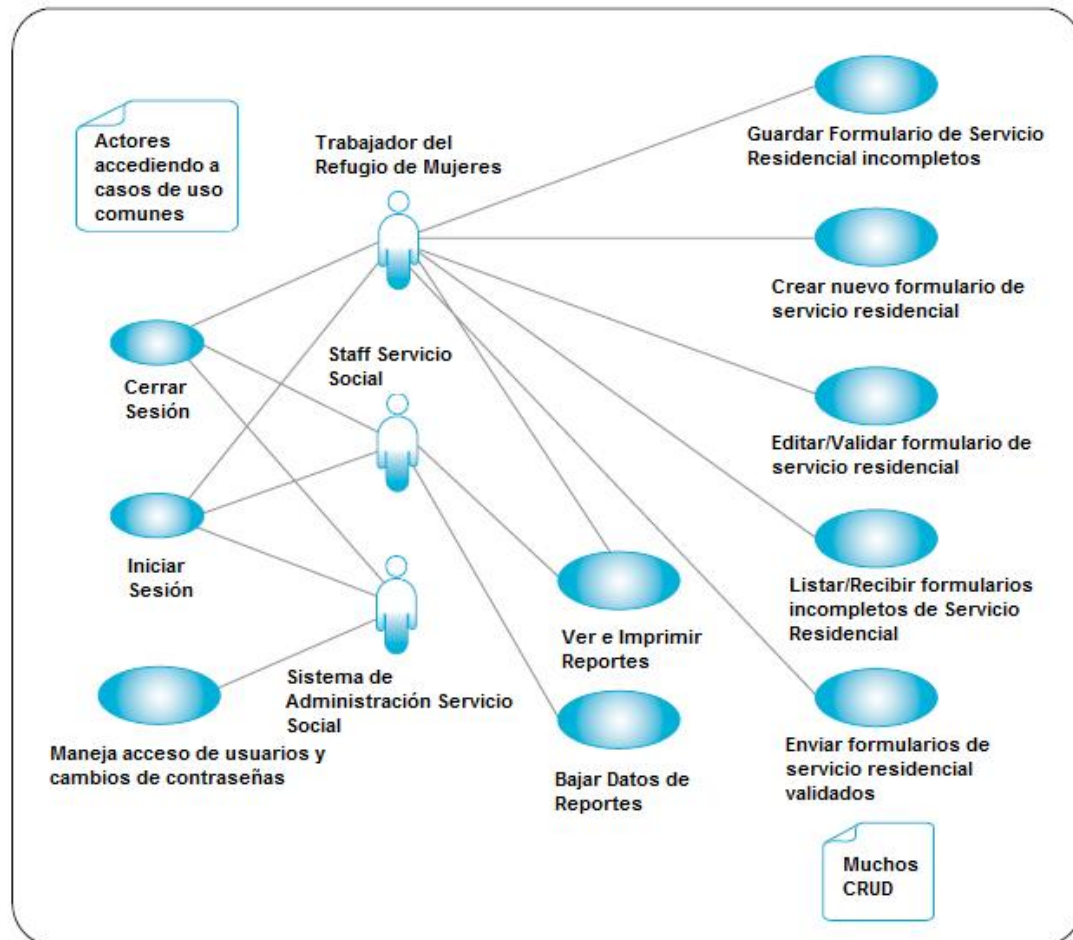


Figura 3: Un caso de Uso pobremente escrito posee demasiadas actividades del tipo CRUD incluidas.

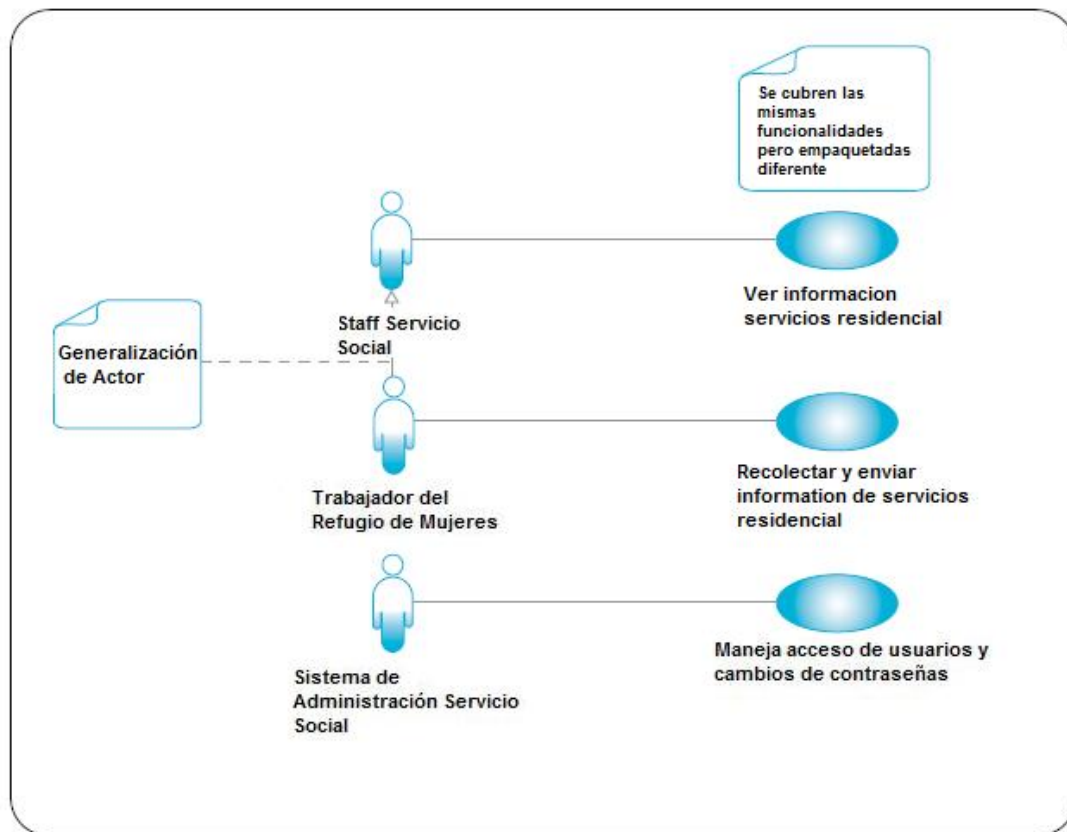


Figura 4: Un caso de uso bien escrito indica claramente caminos y valores

Consejo: La secuencia de los eventos puede ser opcional

Hemos dicho que los pasos en los casos de uso, usualmente están ordenado cronológicamente, pero se debe considerar si el orden de los pasos es un requerimiento. Por ejemplo, ¿El actor tiene que haber completado el paso cuatro antes de completar el paso cinco?, generalmente la respuesta sería SI, pero no siempre. Es una cosa muy simple, al desarrollar el texto de un caso de uso, para aclarar cualquier restricción temporal, agregue comentarios como este “Este paso puede ocurrir en cualquier orden” o “Este paso puede ocurrir en cualquier momento antes que este paso” a una línea. El punto es, usted no quiere restringir a los diseñadores del sistema teniendo requerimientos temporales implícitos cuando no son realmente necesarios.

Consejo: Utilice el nivel correcto de detalle

Una de las preguntas más comunes es “¿Cuanto detalle debe haber en un caso de uso?” Recuerde, los casos de uso son requerimientos. No son documentos de diseño, y ciertamente no son diseños de interfaz de usuario. Uno nunca debería referencias elementos de la interfaz de usuario, tales como página principal, pantalla de ingreso o “clickear” botones en un caso de uso.

De manera similar, los detalles arquitectónicos deben ser evitados. Por ejemplo, en el modelo de registro de cursos, una sentencia como “el horario es guardado en la base de datos SQL Server” no debería ir en un caso de uso. La parte de guardar el horario es apropiada, pero dónde se almacena es un detalle de arquitectura, que podría ser modificado después si el administrador de la base de datos decide que el horario debería ser guardado en una base de datos diferente.

Consejo: Póngase en los zapatos del actor

Los casos de uso son diferentes con respecto al uso de sentencia “deberá”. Dado que ellos especifican pasos ordenados cronológicamente, ellos crean más contexto que las autónomas sentencias “deberá”, las cuales no son temporales. Es por esto que ellos son más reflexivos para la experiencia de los usuarios planeados. Tenga esto en mente. Si los casos de uso son requerimientos (y lo son), los diseñadores y desarrolladores estarán restringidos por lo que ellos dicen. Recuerde esto, sienta pena por el actor y escriba sus casos de uso de una forma que haga que el sistema sea lo más simple posible para el usuario.

Conclusión

Cuando se escribe bien y con un acercamiento consistente, los casos de uso pueden ser una gran manera de especificar los requerimientos de una forma que sea entendible por los usuarios y los grupos de interés, y a la vez, directamente usables por el equipo de desarrollo. Pero la calidad y consistencia sólo puede ser alcanzada si existe un acuerdo en la estructura del caso de uso y los procesos. Por lo que aquí está el consejo final: Documente las decisiones que usted y su organización realicen sobre como realizar buenos casos de uso. Cree una guía de estilos para caso de uso o una presentación, déle a los escritores muchos ejemplos de buenos casos de uso y respete las guías establecidas.

Como todo el esfuerzo puesto en un caso de uso resultará (esperadamente) en un exitoso desarrollo del proyecto, el esfuerzo puesto en establecer prácticas estándares y en seguir estos consejos, resultará en mejores casos de uso que serán valorados por todos los involucrados en el proceso.