

SISTEMA GENERADOR DE DOCUMENTOS DE CASOS DE USO - TESSERACT

Luis G. Jiménez C., Diego E. López O., Esteban P. Martínez I.

Yamile G. Olvera N., Hermes F. Montes C.

Escuela Superior de Cómputo

Instituto Politécnico Nacional

Ciudad de México

Resumen - TESSERACT utiliza el conocimiento generado en las primeras etapas del desarrollo de software para desarrollar un sistema Web que asista en la generación de un documento de análisis basado en casos de uso que coadyuve a los analistas, de tal manera que puedan construir y generar sus documentos de forma estandarizada, que eleve la disponibilidad de la información contenida en los proyectos y que ayude al control del registro, edición y revisión de los casos de uso.

1 INTRODUCCIÓN

La etapa de mantenimiento de software requiere mayor tiempo y costo que sus fases complementarias, por lo que resulta ser la etapa de mayor complejidad dentro del ciclo de vida de desarrollo de software. Se estima que aproximadamente dos tercios del costo total del software se dedican al mantenimiento [1]. Esta situación es causada por diversos problemas presentes durante las etapas precedentes, principalmente en la etapa de análisis, ya que es difícil contar con las bases sólidas de una documentación bien construida y estructurada que favorezca a la fase de mantenimiento. Específicamente, el proceso de documentación de los casos de uso requiere una gran cantidad de esfuerzos humanos y es habitualmente propenso a errores, generando un impacto negativo en el desarrollo e implementación del sistema [2].

El desarrollo de una herramienta de Ingeniería de Software Asistida por Computadora (Computer Aided Software Engineering, CASE por sus siglas en inglés) en un nivel alto favorecería la construcción y generación de la documentación de análisis ya que no solo colaboraría en la estandarización del estilo de trabajo que se emplea en la organización para documentar; también sería capaz de recolectar, almacenar y procesar los elementos que integran un proyecto para generar el documento; elevaría la disponibilidad de la información de tal manera que los integrantes accedan a ella; controlaría quién escribe, modifica y supervisa cada parte del documento; finalmente, ayudaría en la generación de los documentos que se le entregan al cliente.

2 ANÁLISIS

“El proceso de análisis dentro del desarrollo de software consiste en obtener los requerimientos del sistema para

crear una solución, identificar los problemas a resolver o necesidad a ser atendida, evaluar las restricciones que presenta, así como los insumos se requieren para su debida construcción. Al ser la primera etapa dentro del proceso de desarrollo es la más crítica y sensible, ya que cualquier error de gran impacto que surja dentro de esta perjudicará las etapas consecuentes ocasionando retrasos en el proceso.” [3] “Durante esta etapa se con-

struye el documento de análisis, en donde se obtienen y describen todas las necesidades y peticiones del cliente en forma de requerimientos. Con base en estos, el analista continua el proceso identificando en alto nivel que funcionalidades deberá poseer el sistema para cumplir los requerimientos. Una solución común para mapear cada funcionalidad es a través de CASOS DE USO. Este documento proporciona la descripción de la manera en la que se utilizará el sistema y emplea un lenguaje técnico especializado ya que busca ser comprendido por los diseñadores y programadores para su correcta construcción.” [4]

2.1 Documento de análisis

Con base en los enunciados del libro Ingeniería de Software, un enfoque práctico y la experiencia adquirida en el desarrollo de sistemas de la Coordinación de Desarrollo Tecnológico de la ESCOM, se propone la siguiente estructura para la construcción del documento de análisis.

Parte 1. Modelo de negocio:

- Glosario de términos.
- Modelo de información utilizado para representar la información que será almacenada en el sistema.
- Reglas de negocio mediante las cuales se normará el funcionamiento del sistema.

Parte 2. Modelo dinámico, el cual describe funcionalidad a partir de los siguientes capítulos:

- Arquitectura lógica.
- Máquinas de estados que modelarán el comportamiento de las entidades que así lo necesiten.
- Funciones y roles que tendrán los actores que interactuarán con el sistema.
- Casos de uso que describen funcionalidad.

Parte 3. Interacción con el usuario, que muestra las interfaces y mensajes a partir de los siguientes capítulos:

- Interfaces del sistema.
- Catálogo de mensajes.

2.2 Caso de Uso

En un libro que analiza cómo escribir casos de uso eficaces, Alistair Cockburn afirma que: *“un caso de uso capta un contrato que describe el comportamiento del sistema en distintas condiciones en las que el sistema responde a una petición de alguno de sus participantes.”* [5]

Pressman toma esta definición como fundamento y la complementa proponiendo que un caso de uso es una actividad que puede realizar un usuario dentro del software y que sirven para describir el funcionamiento de los componentes acorde a las acciones que los usuarios realizan dentro del software, asegurando que:

“En esencia, un caso de uso narra una historia estilizada sobre cómo interactúa un usuario final (que tiene cierto número de roles posibles) con el sistema en circunstancias específicas. La historia puede ser un texto narrativo, un lineamiento de tareas o interacciones, una descripción basada en un formato o una representación diagramática. Sin importar su forma, un caso de uso ilustra el software o sistema desde el punto de vista del usuario final.” [4]

Para escribir casos de uso eficientes, el autor Alistair Cockburn presenta la estructura un Caso de uso básico que describe la interacción entre el actor y el sistema. Menciona que dependiendo de los requisitos podría haber casos de uso más detallados.

Por ejemplo si se requiere hacer una descripción detallada del caso de uso sugiere el siguiente formato [5]:

- **Nombre del caso de uso**
- **Actor principal**
- **Objetivo en contexto**
- **Disparador**
- **Escenario**

Con base en los enunciados de Alistair Cockburn en su libro “Escribiendo Casos de Uso efectivos” (2001) y en la experiencia en el análisis y desarrollo de sistemas de la Coordinación de Desarrollo Tecnológico CDT, se propone la siguiente estructura para la presentación de un Caso de Uso, así como los elementos que lo componen y forman parte del caso de uso.

- **Actor:** Es la idealización de un rol que puede jugar una persona, otro sistema, proceso, un dispositivo o de alguna cosa que interactúa con el sistema. Los actores son objetos que residen fuera del

sistema, en tanto que los casos de uso están compuestos por objetos y acciones que residen dentro del sistema. Todo actor tiene uno o más objetivos cuando utiliza el sistema. [6]

- **Entidad :** Representación de un objeto exclusivo único en el mundo real que se está controlando. Algunos ejemplos de entidad son una sola persona, un solo producto o una sola organización. [7]
- **Atributo :** Es una especificación que define una propiedad de un objeto, elemento o archivo. También puede referirse o establecer el valor específico para una instancia determinada de los mismos. [7]
- **Entrada:** Es la información producida por el usuario para ser guardada o procesada en el sistema. El usuario comunica y determina qué clases de entrada aceptará el sistema (por ejemplo, secuencias de control o de texto escritas a máquina a través del teclado y el ratón).[8]
- **Salida:** Es la información producida por el sistema y percibida por el usuario. Las clases de salida los productos de programa, y las clases de entrada la que el programa acepta, definen la interfaz de usuario del programa. [8]
- **Acción:** Evento originado por el usuario mediante botones.
- **Pantalla:** Es la interfaz de usuario, utiliza imágenes, iconos y menús para mostrar las acciones disponibles entre las que el usuario puede escoger en un sistema. Su función es proporcionar un entorno visual amigable y sencillo de usar que facilite la comunicación del usuario con el software.[9]
- **Regla de Negocio:** Es aquella que rige los procesos de un negocio para garantizar el correcto funcionamiento del software. Las reglas de negocio establecen los procedimientos que se deben realizar y las condiciones sobre las que dichas actividades se van a ejecutar.
- **Mensaje:** Constituyen la mínima unidad de comunicación entre el usuario y el sistema. Se trata de un proceso de comunicación completa porque el sistema lanza un mensaje hacia el usuario que no se resuelve hasta que el usuario lo recibe o lo responde, completando así el proceso de comunicación con la realimentación correspondiente. [10]
- **Trayectoria:** Es un conjunto de pasos que describen la interacción entre el usuario y el sistema.
- **Paso:** Es una instrucción que realiza el usuario o el sistema.
- **Precondición:** Está formada por el conjunto de condiciones que se tienen que cumplir para que se

pueda iniciar un caso de uso. En muchos casos supone la ejecución de casos de uso previos. [11]

- **Postcondición:** Refleja el estado en que se queda el sistema una vez ejecutado el caso de uso. [11]
- **Puntos de extensión:** Es la incorporación implícita del comportamiento de otro caso de uso, el cuál no es parte del flujo principal. Modela la parte opcional del sistema, un subflujo que sólo se ejecuta bajo ciertas condiciones o varios flujos que se pueden insertar en un punto determinado. [12]

2.3 TESSERACT

Tesseract es un sistema Web que asiste en la generación de un documento de análisis basado en casos de uso ayudando a los analistas, de tal manera que pueden construir y generar sus documentos de forma estandarizada, que eleva la disponibilidad de la información contenida en los proyectos y que ayuda al control del registro, edición y revisión de los casos de uso.

El sistema gestiona:

- Un catálogo de colaboradores, en donde el administrador podrá llevar un control de todo el personal involucrado en el proceso de desarrollo de los proyectos de la organización.
- Un catálogo de proyectos de administrador, en donde se podrá llevar un control del registro de los proyectos y el administrador tendrá la responsabilidad de capturar la información que corresponde a cada proyecto, tal como lo es: clave, nombre, fechas de término y fin, descripción, presupuesto y estado, así como asignar un líder por proyecto (seleccionado de los colaboradores previamente registrados). El administrador también tendrá la facultad de asignar un conjunto de colaboradores al proyecto, los cuales trabajarán como analistas y tendrán los permisos de acceso para operar en los proyectos que se le fueron encomendados.
- Un catálogo de términos de glosario en donde el colaborador controlará aquellas expresiones cruciales para el entendimiento de un proyecto en específico y podrá definirlos registrando su nombre y descripción, con el fin de tener consistencia en la utilización de los nombres de los términos.
- Un catálogo de actores el cual explicará brevemente el objetivo del mismo, teniendo la siguiente estructura para definirlos: el nombre del actor, descripción del mismo y sus responsabilidades relacionadas con el sistema según aplique, con el fin de tener consistencia en la utilización de los nombres de los actores.
- Un catálogo de Reglas de negocio especificando lo siguiente: Identificador y nombre de la regla de negocio, de que tipo es, el nivel, una descripción explicando en qué consiste dicha regla, con el fin tener un control al momento de usarlas en diferentes casos de uso.
- Un catálogo de Mensajes el cual explicará brevemente el objetivo del mismo, este catálogo documentará los mensajes de la siguiente manera: identificador y nombre del mensaje, el tipo de mensaje, propósito, la redacción del mismo y que parámetros deben cumplirse para que el mensaje aparezca esto ayudará a que el usuario pueda reutilizar mensajes en diferentes casos de uso evitando la confusión de los nombres de los mensajes.
- Un catálogo de Entidades con sus respectivos atributos; las entidades documentadas unicamente con el nombre de la entidad y su descripción. Una vez registrada la entidad se podrán gestionar los atributos de la entidad correspondiente, en este caso la estructura para definir los atributos se compone de lo siguiente: el nombre del atributo, la descripción, si es un dato obligatorio o no, y por último el tipo de dato.
- Un catálogo de pantallas con sus respectivas acciones; las pantallas documentadas con el nombre de la pantalla, su respectiva descripción así como la imagen o interfaz representada. Una vez registrada la pantalla se podrán gestionar las acciones de la pantalla correspondiente, en este caso la estructura para definir las acciones se compone de lo siguiente: el nombre de la acción, la descripción y por último la imagen o icono.
- La agrupación de casos uso dividiéndolos por módulos. Con el propósito de obtener una estructura modular con alta cohesión, segmentando el conjunto de Casos de uso en partes más pequeñas y con objetivos similares entre sí para que los analistas operen sobre una configuración más ordenada.
- La generación de los casos de uso integrando los elementos que conforman el caso de uso (descritos anteriormente), así como los elementos que son parte del mismo y que se gestionan internamente en el catálogo de casos de uso como lo son: Trayectorias, Pasos, Precondiciones, Postcondiciones y Puntos de extensión.
- Un estándar de redacción y escritura definido para ordenar y evitar confusiones en la descripción de los casos de uso.

Generar de manera semiautomatizada documentos de casos de uso a través de una herramienta Middle Case es un desafío que propone la idea de transformar la escritura del lenguaje natural (comúnmente empleado en la elaboración de dichos documentos) a un lenguaje estándar, formal, específico y ordenado. De concretarse este desafío, el tiempo que actualmente toma solucionar los problemas que se presentan durante la elaboración del análisis y su documentación por el personal

de análisis será optimizado; coadyuvará a adquirir experiencia al equipo para disminuir errores, su uso representará una reducción en los recursos destinados al análisis y de este modo se generará un documento de análisis con mayor estructura y consistencia.

2.4 RESULTADOS

2.4.1 Gestor de proyectos y colaboradores

A través del cual el administrador registra los proyectos existentes así como la información personal de todo los involucrados en el desarrollo de los sistemas registrados. Posteriormente el administrador se encarga de asignar un líder a cada proyecto (de la relación de colaboradores). Una vez asignado, este actor podrá gestionar los proyectos como analista desde su sesión.

2.4.2 Concentración de la información de los elementos de casos de uso

Un caso de uso está compuesto por elementos que forman parte del proyecto y que intervienen en la redacción del caso de uso, estos elementos son: Actores, Términos del Glosario, Reglas de negocio, Entidades, Módulos, Mensajes y Pantallas. Por otro lado tenemos elementos que son propios del caso de uso, estos elementos son: Trayectorias, Pasos, Precondiciones, Poscondiciones y puntos de extensión. Toda esta información se encuentra organizada y centralizada dentro de un repositorio de datos.

2.4.3 Generador de tokens para la construcción de los casos de uso

Los elementos que componen la redacción del caso de uso pero que no son propios del este son gestionados de forma externa, sin embargo gracias a los valores parametrizados es posible referenciar en el editor del caso de uso cada uno de estos elementos para ser utilizados dentro de la gestión del caso de uso.

2.4.4 Revisión y libreación de los casos de uso generados

Al finalizar la elaboración de un caso de uso debidamente redactado y referenciado, los colaboradores de los proyectos indican al líder que culminaron su caso de uso, el líder procede a revisarlo y realizar las observaciones correspondientes dentro de la misma plataforma. Si hay detalles en el caso de uso registra una serie de comentarios, mismos que el analista podrá visualizar para corregirlos. Un caso de uso podrá ser liberado hasta que no haya observaciones por parte del líder y lo libere.

2.4.5 Generador del documento de análisis

El documento de análisis no solo contiene los casos de uso elaborados, también presenta el catálogo de todos los elementos del proyecto para facilitar la lectura de los casos de uso .

2.5 CONCLUSIONES

La etapa de mantenimiento de software representa un costo muy alto comparado con sus etapas precedentes en el ciclo de desarrollo de software, la solución a este problema radica en agilizar las actividades presentes en el proceso inicial. Es por eso que se ha desarrollado una plataforma que coadyuva en la etapa de análisis agilizando las tareas en la construcción de un documento de análisis de calidad basado en casos de uso.

La documentación de los casos de uso suele ser elaborada y redactada por más de un analista, esta situación provoca que el estilo de escritura pueda variar de un analista a otro, lo cual complica la creación del documento de casos de uso. Nuestro sistema proporciona a los analistas un estándar de escritura, redacción y composición de los casos de uso para obtener un documento final uniforme y fácil de comprender.

Durante los sprints desarrollados en este trabajo, se demostró que es posible gestionar y controlar todos y cada uno de los elementos de un caso de uso, para posteriormente construir el documento de análisis referenciando en el la información de cada elemento en un editor basado en tokens. Dicha información debe estar organizada y centralizada dentro de un repositorio de datos.

2.6 REFERENCIAS

- [1] Rui, K. Butler, G. (2003, April 21). Refactoring use case models: the metamodel [Online]. Available: <https://dl.acm.org/citation.cfm?id=783140>
- [2] Shuang, L. Sun, L. (2014, September 19). Automatic early defects detection in use case documents [Online]. Available: <https://dl.acm.org/citation.cfm?id=2642969>
- [3] Pressman, Roger. (2010). Ingeniería de Software. Un enfoque práctico / 7 ED.(777 páginas). USA: Mcgraw-Hill Interamericana.
- [4] Jacobson, I., Object-Oriented Software Engineering, Addison-Wesley, 1992
- [5] A. Cockburn, Writing effective use cases by Alistair Cockburn. Addison-Wesley: Pearson Professional Education, 2001.
- [6] D. West, "Use Cases Considered Valuable (but Optional) For Lean/Agile Requirements Capture", InfoQ, 2010. [Online]. Available: <https://www.infoq.com/news/2009/02/Use-Cases-Valuable-But-Optional>.
- [7] "IBM Knowledge Center", Ibm.com, 2019. [Online]. Available: https://www.ibm.com/support/knowledgecenter/es/SSWSR9_11.6.0/com.ibm.mdmhs.overview.doc/entityconcepts.html.
- [8] Federal Standard 1037C: Glossary of Telecommunications Terms", Its.bldrdoc.gov, 1996. [Online]. Available: <https://www.its.bldrdoc.gov/fs-1037/fs-1037c.htm>
- [9] S. Zorraquino Comunicación, "Interfaz gráfica de usuario — Zorraquino", Zorraquino, 2019. [Online]. Available: <https://www.zorraquino.com/diccionario/marketing->

digital/que-es-interfaz-grafica-de-usuario.html. [10] J. JUNOY, "Mensajes del sistema", Alzado.org, 2005. [Online]. Available: https://www.alzado.org/articulo.php?id_art=429. [Accessed: 29- Oct- 2019]. [11] "Las precondiciones y postcondiciones en los casos de uso", Jummp, 2012. [Online]. Available: <https://jummp.wordpress.com/2011/07/22/las-precondiciones-y-postcondiciones-en-los-casos-de-uso/>. [12] J. Barquinero, "Tipos de relaciones en diagramas de casos de uso. UML. — Blog SEAS", Blog de SEAS, 2013. [Online]. Available: <https://www.seas.es/blog/informatica/tipos-de-relaciones-en-diagramas-de-casos-de-uso-uml/>.