

INSTITUTO POLITÉCNICO NACIONAL ESCUELA SUPERIOR DE CÓMPUTO

Trabajo terminal

Sistema generador de documentos de casos de uso

"TESSERACT"

2018-B140

Presentan:

Jiménez Chávez Luis Gerardo

López Orozco Diego Efrain

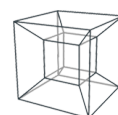
Martínez Ibáñez Esteban Pablo

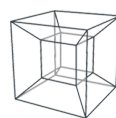
Olvera Neria Yamile Giselle

Directores: M. en C. José Jaime López Rabadán, M. en C. Hermes Francisco Montes Casiano
Escuela Superior de Cómputo, IPN

17 de abril de 2019

1. Introducción	5
1.1. Problemática	5
1.2. Propuesta	6
1.3. Objetivos	6
1.3.1. Objetivo General	6
1.3.2. Objetivos Específicos	6
1.4. Justificación	7
1.5. Estructura del Documento	7
2. Estado del Arte	9
2.1. Antecedentes	10
2.1.1. UCD-Generator - Una aplicación LESSA para el diseño de casos de uso	10
2.1.2. Generación automatizada de diagramas de casos de uso a partir de requerimientos de usuarios	10
3. Marco teórico	11
3.1. Ingeniería de software	11
3.1.1. Proceso de desarrollo de software	12
3.1.2. Modelado	13
4. Análisis de mercado	15
4.1. Situación actual y evolución del mercado	15
4.1.1. Industria Mexicana del Software	16
5. Bibliografía	17

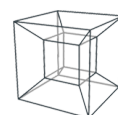




La etapa de mantenimiento de software requiere mayor tiempo y costo que sus fases complementarias, por lo que resulta ser también la etapa de mayor complejidad dentro del ciclo de vida de desarrollo de software. Se estima que aproximadamente dos tercios del costo total del software se dedican al mantenimiento [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].

1.1. Problemática

La obtención de requerimientos es crucial para la generación de casos de uso desde el punto de vista del analista [3]. La inadecuada especificación de requerimientos es una de las causas predominantes en el fracaso del desarrollo de los sistemas de software hoy en día [4]. Del mismo modo, es común que el equipo de análisis se enfrente a situaciones que dificultan y prolongan la tarea de documentar casos de uso, algunos de los problemas más comunes son: La falta de consistencia en la utilización de los nombres de los actores, reglas de negocio y mensajes, la incorrecta agrupación de casos de uso en gestiones determinadas, la confusión entre escenarios [5], la falta de adaptación a un estándar de escritura y redacción de los elementos del documento y la incorrecta descripción de derechos funcionales (permisos), todos esto, resultado de la falta de experiencia de los analistas y el gran esfuerzo humano que requiere obtener un producto final óptimo [6].



1.2. Propuesta

Se propone construir un sistema que asista a la generación de un documento de análisis basado en casos de uso, que permita gestionar:

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 los 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. La agrupación de casos uso dividiéndolos por módulos. Un estándar de redacción y escritura definido para evitar confusiones en la descripción de los casos de uso. Generar de manera automatizada documentos de casos de uso 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 formal y específico. 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 de tiempo (el cual no podrá ser medido debido al tiempo con el que se cuenta para desarrollar el proyecto) y en los recursos destinados al análisis. Lo anterior permitirá generar un documento de análisis con mayor estructura y consistencia.

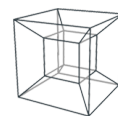
1.3. Objetivos

1.3.1. Objetivo General

Desarrollar un sistema que asista en la generación de la documentación de casos de uso de un proyecto de software con base en una plantilla predefinida a fin de contribuir en el proceso de su creación.

1.3.2. Objetivos Específicos

Generar un módulo de gestión de usuarios encargado del control de acceso y la administración de los usuarios, en este módulo se asignan los roles y permisos a los diferentes usuarios que se registren. Generar un módulo de gestión proyectos encargado de la administración de proyectos, a partir de este módulo se hará el registro, lectura, modificación, eliminación de los componentes necesarios para la documentación del proyecto; contendrá un apartado para la asignación de usuarios al proyecto. Generar un módulo de gestión de casos de uso encargado de la creación, lectura, modificación y eliminación de casos de uso así como la asociación de analistas. Generar un módulo de gestión de elementos encargado de la definición y registro de todos los elementos necesarios para la creación de casos de uso, con el objeto de registrarlos en una base de datos y poder reutilizarlos al momento de escribir un caso de uso. Generar un módulo de revisión y validación de elementos de casos de uso encargado de mostrar los elementos que los conforman para su



revisión y validación por usuarios permitidos. Generar un módulo de generación de documento de análisis encargado de la generación del documento final de casos de uso para el desarrollo de un sistema con base en la plantilla predefinida.

1.4. Justificación

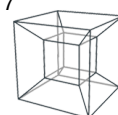
Un proyecto de software bien construido y formado es esencial para la competitividad de una organización, e incluso para su propia supervivencia, del mismo modo, la documentación es un elemento partícipe que determina la calidad del sistema dado que facilita su interpretación y comprensión, provee los antecedentes que permiten conocer cómo fué diseñado, que hace y cómo está operando, sirve de base para auditorias, elimina los riesgos de dependencia con respecto al personal, es fundamental para la capacitación de los usuarios del sistema facilitando la comunicación, provee antecedentes esenciales, concretos y permanentes para evaluar modificaciones a su funcionamiento y/o para decidir la sustitución de los mismos y aumenta la seguridad y eficiencia en su mantenimiento reduciendo su costo.

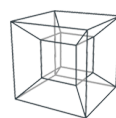
El proceso de construcción del documento no es sencillo, al analista le toma tiempo aprender y hacer de manera entendible la redacción, la inclusión de elementos del caso de uso y la especificación correcta de las trayectorias. La curva de aprendizaje es extensa y es común que una persona inexperta en el tema tenga complicaciones y retrasos al realizar el documento. Una herramienta web capaz de recolectar, almacenar y procesar los elementos que integran un proyecto para generar el documento de análisis será de gran apoyo para los analistas, reduciendo de manera considerable el tiempo, costo y gastos de dicho documento.

Éste proyecto de trabajo terminal se considera un trabajo terminal porque coadyuvará a formación de los autores en áreas de investigación, autoaprendizaje, y resolución de problemas, en la generación de este sistema se utilizarán conocimientos del área de Ingeniería de software, bases de datos, programación, tecnologías web, algoritmos y diseño orientado a objetos.

1.5. Estructura del Documento

En el capítulo ?? ... En el capítulo ?? ...



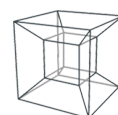


Es común que dentro del área de la ingeniería de software, se confundan los términos: "Caso de uso" y "Diagrama de caso de uso", sin embargo es importante resaltar las diferencias para comprender el objetivo principal del proyecto terminal.

Un caso de uso narra una historia detallada sobre cómo interactúa un usuario final (con 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 de casos de uso. Sin importar su forma, un caso de uso ilustra el software o sistema desde el punto de vista del usuario final [7].

En otras palabras, un caso de uso es aquel que describe en forma de secuencia de acciones o pasos la interacción entre un actor y el sistema, en cambio, un diagrama de casos de uso es el modelo de un grafo con dos tipos de nodos (Actor y caso de uso), el cual ilustra gráficamente el comportamiento del caso de uso, un diagrama de casos de uso no describe la interacción detallada del sistema con los actores ni reemplaza el documento de casos de uso

Ahora bien, en la red hay una gran variedad de sistemas que permiten la generación de diagramas de casos de uso en UML, a partir de distintas técnicas, sin embargo no hay herramientas comerciales o gratuitas que posibiliten la generación del documento con las especificaciones y la gestión de sus componentes.



2.1. Antecedentes

2.1.1. UCD-Generator - Una aplicación LESSA para el diseño de casos de uso

Las herramientas CASE convencionales requieren una comprensión completa del negocio, una gran cantidad de tiempo y esfuerzos adicionales por parte del analista del sistema durante el proceso de creación, organización, etiquetado y finalización de los diagramas de casos de uso. Es por esto que se diseñó un sistema que proporciona una manera rápida y confiable de generar diagramas de casos de uso para ahorrar tiempo y presupuesto tanto para el usuario como para el analista del sistema.

Objetivo

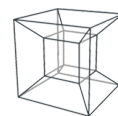
Este sistema presenta un enfoque basado en el procesamiento del lenguaje natural LESSA (Language Engineering System for semantic analysis) que se utiliza para comprender automáticamente el texto en lenguaje natural y extraer la información requerida. Esta información se utiliza para dibujar los diagramas de casos de uso. El usuario escribe sus preferencias basadas en la interfaz en inglés, en unos pocos párrafos y el sistema diseñado tiene una capacidad notable para analizar el script dado. Después del análisis compuesto y la extracción de información asociada, el sistema diseñado en realidad dibuja los diagramas de casos de uso [8].

2.1.2. Generación automatizada de diagramas de casos de uso a partir de requerimientos de usuarios

Con el estado actual de la tecnología de procesamiento de lenguaje natural (PNL), muchos investigadores han demostrado que automatizar el proceso de análisis de requisitos es posible, lo que ahorra una cantidad significativa de tiempo invertido por los analistas. Se han desarrollado numerosas herramientas semiautomáticas que ayudan al analista en este proceso. Sin embargo, una técnica comúnmente utilizada para usar la gramática en el texto obtenido como la base para identificar información útil, ha estado enfrentando problemas de escalabilidad debido a que el formato textual de los requisitos consiste en lenguaje natural no estructurado (NL).

Objetivo

Este proyecto utiliza una técnica probabilística para identificar actores y casos de uso. El resultado prometedor demuestra que las mejoras adicionales de este enfoque pueden automatizar completamente la fase de análisis, propone una metodología para la asistencia automática de análisis de requisitos a los analistas de software mediante la extracción de un diagrama de caso de uso del documento de requisitos del usuario. Este proyecto ha intentado con éxito extraer actores y usar casos utilizando un modelo de clasificación probabilística junto con una asistencia mínima de enfoque basado en reglas. Los casos de uso son nítidos y consistentes independientemente del tamaño del texto de los requisitos. Debido al pequeño tamaño de los datos utilizados, el rendimiento no se ha logrado precisar. Sin embargo, se pueden utilizar mejores modelos de clasificación con un conjunto de datos más grande que incluya otros dominios de software para mejorar los resultados. El desafío restante aquí se relaciona con abordar los requisitos no funcionales y también para incorporar funciones de inclusión y extensión al diagrama de casos de uso. Un gráfico bien diseñado [9].



3.1. Ingeniería de software

Una de las primeras definiciones de ingeniería de software fue dada por Fritz Bauer en el año de 1969, quien define que la ingeniería de software es “el establecimiento y uso de principios robustos, orientados a obtener software económico que sea fiable y que funcione de manera eficiente sobre máquinas reales” [7], aunque esta definición omite algunos términos referentes a tiempos de entrega, procesos eficaces, y calidad de software, nos da un panorama de sus principios fundamentales y es también la base de la definición que la IEEE ha desarrollado de una manera más completa:

”La ingeniería de software es: 1) La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la ingeniería al software. 2) El estudio de enfoques según el punto 1”

La ingeniería de software está formada por un proceso, un conjunto de métodos (prácticas) y un arreglo de herramientas que permite a los profesionales elaborar software de cómputo de alta calidad.





Figura 3.1: Capas de la ingeniería de software

La ingeniería de software es una tecnología con varias capas, como se muestra en la figura 3.1, existen 4 capas: herramientas, métodos, procesos y compromiso con la calidad. Cada una de ellas es importante, sin embargo, la capa de proceso es fundamental para el desarrollo de software, ya que es donde se define la estructura básica del producto hasta la culminación del mismo.

El proceso de software forma la base para el control de la administración de proyectos de software, y establece el contexto en el que se aplican métodos técnicos, se generan productos del trabajo (modelos, documentos, datos, reportes, formatos, etc.), se establecen puntos de referencia, se asegura la calidad y se administra el cambio de manera apropiada.

3.1.1. Proceso de desarrollo de software

Cuando se trabaja en la construcción de un producto o sistema, es importante ejecutar una serie de pasos predecibles, una estructura general para la ingeniería de software se define en cinco actividades elementales:

1. Comunicación
2. Planeación
3. Modelado
4. Construcción
5. Despliegue

Existen diferentes metodologías de desarrollo con modificaciones y adecuaciones al esquema general de construcción antes mencionado, algunas de ellas son las metodologías tradicionales y ágiles. Este proceso puede tener diferentes variaciones, sin embargo, sea cual sea la metodología aplicada, las etapas de Modelado (Análisis y Diseño) y Construcción (Codificación y Pruebas) son las más críticas e importantes para un producto final exitoso.

Durante el desarrollo, se realizan tareas específicas para cada etapa, por ejemplo, para la etapa de modelado se elabora el documento de análisis (donde se describe el funcionamiento del sistema), así como el



diseño (en donde se generan los diagramas que describen el funcionamiento establecido en el análisis); en la fase de construcción se genera el código del software y en la etapa de pruebas se valida y verifica que el software cumpla con lo asentado en las fases precedentes.

3.1.2. Modelado

Análisis

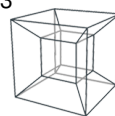
El proceso de análisis dentro del desarrollo de software consiste en comprender y definir los requerimientos del sistema, evaluar las restricciones presentas, así como los insumos se requieren para su debida construcción. Al ser la primera etapa dentro del proceso de desarrollo es la más crítica y sensible, ya que cualquier error que surja dentro de esta perjudicará las etapas consecuentes ocasionando retrasos en el proceso. En esta etapa se construye el documento de análisis, en donde se describen todos los requerimientos que el cliente ha solicitado mediante diferentes componentes como lo son:

- Reglas de negocio
- Mensajes
- Pantallas
- Casos de uso
- Entidades

Este documento emplea un lenguaje técnico especializado ya que busca ser comprendido por los diseñadores y programadores para su correcta construcción.

Caso de Uso

Un caso de uso es una actividad que puede realizar un usuario dentro del software. Estas actividades sirven para describir el comportamiento del producto en distintas condiciones en las que el sistema responde a alguna de las peticiones realizadas por el usuario, es decir, describe el funcionamiento de los componentes acorde a las acciones que los usuarios realizan dentro del software. Un caso de uso está compuesto por distintos elementos, los cuales se describen a continuación:





Análisis de mercado

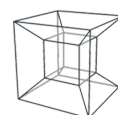
En este apartado se demuestra la viabilidad comercial del trabajo terminal TESSERACT en México, así mismo se realiza un estudio en donde se determina el campo en donde un sistema con las características del generador de documento de casos de uso podría generar un mayor impacto y aceptación por parte de los equipos de construcción de software. Cabe resaltar que TESSERACT no pretende ser comercializado por el momento.

4.1. Situación actual y evolución del mercado

El software es un elemento consustancial a la economía moderna. Se usa en casi todos los productos manufacturados y en los servicios; aun cuando hay empresas especializadas en su desarrollo que parecen constituir una industria distinta, son apenas un segmento de una más compleja; esta clase de industria no refleja en realidad su verdadero desarrollo, ya que la elaboración de programas de cómputo figura en casi todas las industrias y es, de hecho, factor de éxito de todos los sectores de la economía.

El desarrollo de software, es uno de los sectores tecnológicos más competitivos y no es algo nuevo, ya que durante muchos años lo ha sido, sin embargo ha tenido una evolución constante en lo que se refiere a las metodologías o bien, las formas en las cuales se realiza la planeación para el diseño del software, básicamente con el objetivo de mejorar, optimizar procesos y ofrecer una mejor calidad.

En el campo del desarrollo de software, existen dos grupos de metodologías, las denominadas tradicionales (formales) y las ágiles. Las primeras son un tanto rígidas, exigen una documentación exhaustiva y se centran en cumplir con el plan del proyecto definido totalmente en la fase inicial del desarrollo del mismo; mientras que la segunda enfatiza el esfuerzo en la capacidad de respuesta a los cambios, las habilidades del equipo y mantener una buena relación con el usuario. Ambas propuestas tienen sus propias ventajas y desventajas; de cualquier manera, las metodologías de desarrollo nos dicen el ¿Qué hacer? más no el ¿Cómo hacer?, esto significa que la metodología que elijamos, debe ser adaptada al contexto del proyecto, teniendo en cuenta los recursos técnicos y humanos; tiempo de desarrollo y tipo de sistema.



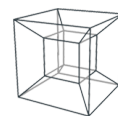
4.1.1. Industria Mexicana del Software

Antes de describir el perfil de las empresas desarrolladoras de software en México, es importante destacar que los diversos análisis que hasta la fecha se han realizado con respecto al panorama de este sector no resultan aún generalizables a toda la industria, ya que cada estudio analiza sólo un subconjunto del total de empresas, por lo tanto se hace la aclaración que lo aquí se presenta son datos representativos, y no necesariamente significa que sean generalizables.

Localización Geográfica de las Empresas Participantes Las empresas participantes en el estudio se localizan en 11 de los 32 estados de la República Mexicana, presentando la siguiente distribución: 2.9 % Chihuahua, 1.5 % en Coahuila, 44.1 % en la Ciudad de México, 11.8 % en Durango, 2.9 % en el Estado de México, 1.5 % en Guanajuato, 2.9 % en Jalisco, 2.9 % en Michoacán, 2.9 % en Morelos, 23.5 % en Nuevo León y 2.9 % en Querétaro. Esta concentración es similar a la de otros estudios realizados para este sector en México [1, 2].

Número de Empresas Desarrolladoras de Software en México La respuesta a esta pregunta no tiene una cifra exacta. De acuerdo con estimaciones realizadas por ESANE consultores [2] sobre del número total de empleados y empresas de la Industria del Software en México, el número aproximado de empresas de la industria mexicana del software podría ser del orden de 1,500 empresas.

Tamaño de las Empresas El estudio revela que el 85.29 % de las empresas del sector de la Industria Mexicana del Software son de tamaño micro (54.41 %) y pequeño (30.88 %), el 5.8 % mediana, y tan sólo el 8.82 % son de tamaño grande (con un número de empleados mayor a 100).



Bibliografía

- [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] J. Lee. (1999, August). Analyzing user requirements use cases a goal driven approach. [Online]. Available: <https://ieeexplore.ieee.org/document/776956>
- [4] L. Julijana. (2007, August). "Information Systems Modeling with Use Cases" IEEE Computer [Online]. Available: <https://ieeexplore.ieee.org/document/4283759>
- [5] Jason Gorman, J. G. (2007, 9 marzo). 10 Common Use Case Pitfalls. Recuperado 23 abril, 2018, [Online]. Available: <http://codemanship.co.uk/parlezuml/blog/?postid=364>
- [6] Susan Lilly [2002, August]. "Use Case Pitfalls: Top 10 Problems from Real Projects Using Use Cases" [Online]. Available: <https://ieeexplore.ieee.org/document/787547>
- [7] Pressman, Roger. (2010). Ingenieria de Software. Un enfoque práctico / 7 ED.(777 páginas). USA: McGraw-Hill Interamericana.
- [8] B. Imran S. and H. Irfan, ÜCD-generator - a LESSA application for use case design - IEEE Conference Publication", ieeexplore.ieee.org, 2007. [Online]. Available: <https://ieeexplore.ieee.org/document/4381333>.
- [9]

