



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

CARRERA DE INGENIERÍA DE SISTEMAS

**APLICACIÓN DE ANÁLISIS DE DATOS CUALITATIVOS DENTRO DEL
PROCESO DE RECOPIACIÓN DE REQUERIMIENTOS EN LA ELABORACIÓN
DE UN SISTEMA INFORMÁTICO**

Trabajo de titulación previo a la obtención del
Título de Ingeniero de Sistemas

**AUTOR: CRISTIAN MARCELO MORILLO TORRES
TUTOR: GUSTAVO ERNESTO NAVAS RUILOVA**

Quito – Ecuador
2023

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo Cristian Marcelo Morillo Torres, con documento de identificación N.º 1719059691, manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 22 de septiembre de 2023

Atentamente,



.....
Cristian Marcelo Morillo Torres
1719059691

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Yo Cristian Marcelo Morillo Torres, con documento de identificación N.º 1719059691, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del artículo académico: “Aplicación de análisis de datos cualitativos dentro del proceso de recopilación de requerimientos en la elaboración de un sistema informático”, el cual ha sido desarrollado para optar por el título de: Ingeniero de sistemas, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En concordancia con lo manifestado, suscribo este documento en el momento que hago entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 22 de septiembre de 2023

Atentamente,



.....
Cristian Marcelo Morillo Torres
1719059691

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Gustavo Ernesto Navas Ruilova con documento de identificación N 1705675625, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: APLICACIÓN DE ANÁLISIS DE DATOS CUALITATIVOS DENTRO DEL PROCESO DE RECOPIACIÓN DE REQUERIMIENTOS EN LA ELABORACIÓN DE UN SISTEMA INFORMÁTICO, realizado por Cristian Marcelo Morillo Torres con documento de identificación N 1719059691, obteniendo como resultado final el trabajo de titulación bajo la opción Artículo Académico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 22 de septiembre de 2023

Atentamente,



.....
Ing. Gustavo Ernesto Navas Ruilova, Msc.
1705675625

APLICACIÓN DE ANÁLISIS DE DATOS CUALITATIVOS DENTRO DEL PROCESO DE RECOPIACIÓN DE REQUERIMIENTOS EN LA ELABORACIÓN DE UN SISTEMA INFORMÁTICO

APPLICATION OF QUALITATIVE DATA ANALYSIS WITHIN THE PROCESS OF COLLECTION OF REQUIREMENTS IN THE PREPARATION OF A COMPUTER SYSTEM

Cristian Marcelo Morillo Torres¹, Gustavo Ernesto Navas Ruilova²

Resumen

El propósito de este artículo es aplicar análisis de datos cualitativo dentro del proceso de recopilación de requerimientos en la elaboración de un sistema informático, esto haciendo uso de Teoría Fundamentada, la cual será el punto de apoyo para realizar toda la investigación. Se creó un cuestionario estándar del cual se recopiló la información necesaria a partir de la participación de 10 personas las cuales en su totalidad son profesionales en el área del desarrollo de software. Posteriormente haciendo uso de la herramienta Atlas Ti, se realizó un análisis de tipo cualitativo. Los resultados obtenidos reflejan el panorama actual de la elicitación de requerimientos dentro del desarrollo de sistemas y los pros y contras que, según los hallazgos obtenidos dentro de la investigación derivan en el éxito o fracaso de este proceso de toma de requerimientos y por consiguiente como afecta al resultado final del proyecto. Por último, se sugieren buenas prácticas en el proceso de elicitación de requerimientos ya que es un factor determinante para el éxito o fracaso del desarrollo de un sistema informático.

Palabras Clave: Teoría fundamentada, ingeniería de requerimientos, lenguaje natural, requerimiento, análisis cualitativo.

Abstract

The purpose of this article is to apply qualitative data analysis within the process of compiling requirements in the development of a computer system, this making use of Grounded Theory, which will be the point of support to carry out all the research.

A standard question bank was created from which the necessary information was collected from the participation of 10 people who work within the field of software engineering, later this information gave room to a qualitative analysis, where the Atlas Ti tool was used.

The results obtained reflect the current panorama of the Requirements Elicitation within the development of a system and the pros and cons that, according to the interview participants, derive in the success or failure of this process of taking requirements and therefore how it affects to the final result of the project.

Finally, good practices are suggested in the Requirements Elicitation process since it is a determining factor for the win or fail of the creation of a computer system.

Keywords: Grounded theory, requirement engineering, requirement, natural language, qualitative analysis.

¹Estudiante de Ingeniería de Sistemas-Universidad Politécnica Salesiana, Egresado – UPS – sede Quito. Autor para correspondencia: cmorillot@est.ups.edu.ec

²Máster en Software Libre en la Universidad Abierta de Cataluña. gnavas@ups.edu.ec

1. Introducción

La elicitación de requerimientos es el proceso de determinar los requerimientos para un sistema propuesto, especificar las necesidades de dicho sistema y las funcionalidades que este deberá entregar. [1]. Los requisitos son la base del sistema, y si estos son incorrectos entonces independientemente de los mejores procesos de desarrollo de software, el sistema implementado será incorrecto. [2]

El proceso de elicitación de requerimientos, analizarlos, validarlos y gestionarlos a lo largo del ciclo de vida de un proyecto se denomina ingeniería de requerimientos. [2] Cada etapa mencionada es tan importante como la que la precede sin embargo para efectos de nuestro estudio nos centraremos en la etapa de elicitación de requerimientos.

Como se había mencionado en esta etapa del proceso es donde se obtiene los requerimientos necesarios que el sistema deberá poseer a futuro, para esto se hace uso de diferentes técnicas como: entrevistas, encuestas, cuestionarios, observación entre otros, las cuales ayudarán a obtener la información de los interesados [3].

Sin embargo, esta etapa del proceso puede resultar un tanto compleja debido a que se debe tratar directamente con el interesado, quien en la mayoría de los casos será una persona con poco o nulo conocimiento acerca de lo que implica el desarrollo de un sistema, entre otras variables que dificultan el proceso de la obtención de requerimientos [1]. Por su parte la teoría fundamentada según [4], es una metodología de investigación cualitativa que tiene como objetivo desarrollar o probar una teoría sobre el tema en estudio. Se usa comúnmente para estudiar el comportamiento humano y las interacciones con el medio ambiente.

Esta metodología proporcionará un enfoque de investigación cualitativa que puede ser utilizado para comprender y

explorar en profundidad el proceso de elicitación de requerimientos. En lugar de partir de suposiciones o teorías preexistentes, la teoría fundamentada permite que los datos recopilados durante la elicitación de requerimientos guíen el análisis y la generación de teorías.

Dentro del campo de la Teoría Fundamentada, se utilizan tres elementos fundamentales para el análisis de los datos: conceptos, categorías y proposiciones. Estos elementos permiten organizar y desarrollar la teoría emergente a partir de los datos recopilados durante la investigación. Por su parte [5] define a estos conceptos como codificación abierta, selectiva y teórica respectivamente. Los conceptos surgen de los datos y son las ideas básicas que, cuando están relacionadas, conducen a las categorías. Finalmente, las proposiciones toman los conceptos y categorías anteriores que producen un conjunto discursivo de declaraciones teóricas. [5]

2. Metodología usada

En esta parte se expone la metodología empleada para cada etapa de la investigación, empezando desde el diseño de las preguntas, la muestra y los métodos empleados para recopilar la información.

Cabe mencionar que el enfoque que se usó para el análisis de datos fue un enfoque cualitativo, es decir que se analizó el conjunto de respuestas expresadas en lenguaje natural por los participantes de las entrevistas, con el propósito final de comprender las experiencias, sucesos, acontecimientos y luego analizarlos desde una perspectiva interpretativa.

Los datos se obtuvieron a través de grabaciones de entrevistas en formato de audio, para posteriormente ser transcrito a texto, a continuación, se hizo un barrido de toda la información obtenida con la finalidad de buscar categorías que nos

ayuden a entender de mejor manera el estado del arte de la elicitación de requerimientos como tal.

2.1 Diseño del banco de preguntas

El diseño del banco de preguntas consistió en 20 preguntas las cuales en su totalidad fueron preguntas de tipo abiertas, esto con el propósito de obtener la máxima cantidad de información de los participantes de las entrevistas.

2.2 Población

La población que se consideró para este estudio fue el conjunto de profesionales que se desempeñan dentro del campo de desarrollo de software, independientemente de su edad o años de experiencia.

Fueron 10 el número total de personas que participaron de esta investigación, quienes de acuerdo con su experiencia ofrecieron una perspectiva más detallada sobre cómo se lleva a cabo esta actividad tan importante dentro de la ingeniería de requerimientos y por consiguiente dentro del desarrollo de software.

2.3 Técnica de Recolección de Datos

La técnica utilizada para recolectar los datos en la investigación fue la entrevista semi estructurada, siendo esta la opción más idónea para conseguir el fin deseado. Esta técnica permitió obtener la información necesaria por parte de los participantes, quienes, al tratarse de preguntas abiertas, tuvieron la oportunidad de responder libremente a cada pregunta realizada.

Dichas preguntas fueron realizadas previamente como parte de la preparación para la entrevista, sin embargo, debido a la dinámica y a la naturaleza de la investigación, fueron apareciendo nuevas preguntas que se consideraron importantes y que fueron adjuntadas sobre la marcha, esto con el propósito de obtener toda la información posible.

La investigación se desarrolla siguiendo los siguientes pasos:

a) Obtención y análisis de la información. La recopilación de información se la realizó haciendo uso del banco de preguntas diseñado, en esta etapa participaron 10 personas. Cada entrevista fue grabada en audio y posteriormente se depuró la información obtenida.

b) Limpieza y depuración de la información

Una vez obtenidos los audios de cada entrevista se procedió a transcribirlos a texto, con la finalidad de obtener un documento.

c) Procesado de la información

Para el procesado de la información obtenida se hizo uso del programa ATLAS TI versión 9.1.3.0 el cual es un software de carácter privativo pero que brinda excelentes resultados en el análisis de datos de tipo cualitativo.

De este software se desprendieron un total de 56 códigos los cuales posteriormente fueron agrupados en 4 categorías que son: Elicitación de requerimientos, calidad del software, gestión de los requerimientos y objetivo del desarrollo, todo esto dentro del campo de la elicitación de requerimientos.

d) Análisis de resultados

A continuación, se detallará los pasos usados en esta metodología uno a uno.

3. Análisis de resultados

3.1 Análisis Cualitativo

En el siguiente apartado, se presenta el análisis inductivo de los resultados partiendo por el análisis de la primera red semántica denominada “Elicitación de requerimientos”, la cual está integrada por los siguientes códigos: Técnica para la elicitación, gestión de la información, técnicas para la interpretación de requerimientos, definición de

requerimientos, requerimiento no funcional, documentación, preparación preentrevista, comunicación con los interesados, información acerca del interesados, evaluación de la competencia. Ver red semántica 1.

Entiéndase como requerimiento a “Una característica necesaria para el sistema que se va a desarrollar, en el que se describen funcionalidades que serán de valor para los stakeholders” [6]

La correcta especificación de requerimientos dentro del desarrollo de software marcará la diferencia entre uno de alta calidad y uno de baja calidad. “Los requerimientos son la base del sistema, y si estos son incorrectos, entonces el sistema implementado será incorrecto.” [1]. Para que un requerimiento sea considerado de calidad deberá cumplir con ciertas características como: No ser ambiguo, ser correcto, ser completo, ser verificable, ser consistente, ser priorizable, ser modificable, ser explorable y ser utilizable. [7]

La primera red semántica hace referencia al inicio del proceso de elicitación de requerimientos a través de la “Comunicación con los interesados” en donde el encargado de obtener los requerimientos busca obtener un acercamiento a la parte interesada por lo que se necesita establecer canales de comunicación efectivos y apropiados para interactuar con ellos, el (Informante 1) señala, “Utilizo una combinación de reuniones, llamadas y correos electrónicos para mantener una comunicación clara con los stakeholders. Además, utilizo herramientas de colaboración en línea para la comunicación y mantener a todos actualizados”. Esta etapa busca captar la mayor cantidad de información acerca del interesado para poder obtener un panorama de su situación y necesidades.

Respecto al código Técnica para la elicitación son métodos y enfoques específicos para obtener información y capturar los requerimientos de un proyecto, existen varias técnicas entre las que

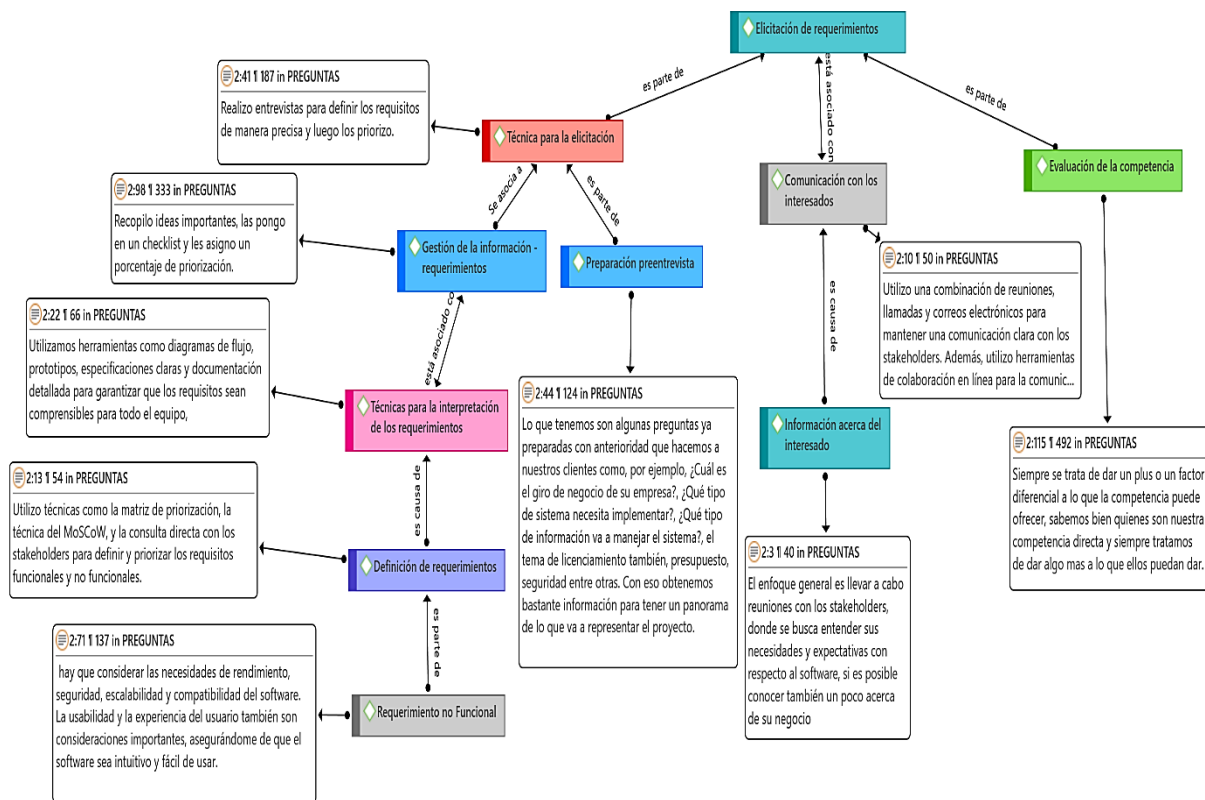


Figura 1 Primera red semántica: Elicitación de requerimientos

destacan entrevistas, cuestionarios y encuestas, observación, análisis de documentos existentes, análisis de casos de uso, el (Informante 3) señala *“Realizo entrevistas para definir los requisitos de manera precisa y luego los priorizo”*. Para esto normalmente se debe tener una preparación previa que ayude a tener una interacción más productiva con los interesados y obtener la información necesaria de manera efectiva. el (Informante 2) señala, *“Lo que tenemos son algunas preguntas ya preparadas con anterioridad que hacemos a nuestros clientes como, por ejemplo, ¿Cuál es el giro de negocio de su empresa?, ¿Qué tipo de sistema necesita implementar?, ¿Qué tipo de información va a manejar el sistema?, el tema de licenciamiento también, presupuesto, seguridad entre otras.”*

Por su parte el código gestión de la información se refiere a la manera de organizar la información recopilada con el fin de garantizar su almacenamiento, organización y uso eficiente.

Las técnicas para la interpretación de los requerimientos ayudan a comprender y analizar en profundidad los requisitos recopilados durante la elicitación, siendo un ejercicio fundamental para poder tener un mejor escenario de dichos requerimientos, el (Informante 1) señala *“Utilizamos herramientas como diagramas de flujo, prototipos, especificaciones claras y documentación detallada para garantizar que los requisitos sean comprensibles para todo el equipo”*.

Posteriormente aparece el código definición de requerimientos que es el resultado del análisis realizado, aquí se pueden ver los requisitos de manera más precisa y detallada y en consecuencia se los puede separar por categorías como su nivel de importancia, el (Informante 1) indica, *“Utilizo técnicas como la matriz de priorización, la técnica del MoSCoW, y la consulta directa con los stakeholders para definir y priorizar los requisitos funcionales y no funcionales.”*, La técnica MoSCoW es

un método utilizado en la gestión de requerimientos para establecer prioridades según la importancia y el grado de necesidad de los requisitos. El acrónimo "MoSCoW" se compone de las iniciales de los términos: Must (Debe), Should (Debería), Could (Podría) y Won't (No lo hará), esta técnica categoriza a los requerimientos desde la categoría más crítica que es Must en donde estarán los requerimientos esenciales para el éxito del proyecto, hasta aquellos requisitos que no serán implementados clasificados dentro de la categoría Won't.

Respecto al código requerimiento no funcional es una restricción a la funcionalidad del sistema, por ejemplo: tiempo, rendimiento, confiabilidad, disponibilidad, portabilidad, facilidad de uso, seguridad, confiabilidad o una restricción de hardware. [...] Si no se satisfacen las necesidades no funcionales, el sistema que se proporcione podría no ser útil o el cliente podría considerarlo insatisfactorio [1]. El (Informante 2) señala *“hay que considerar las necesidades de rendimiento, seguridad, escalabilidad y compatibilidad del software. La usabilidad y la experiencia del usuario también son consideraciones importantes (...)”*

Con respecto al código evaluación de la competencia puede brindar al desarrollador una visión sobre las soluciones existentes en el mercado, sus características, fortalezas y debilidades. Esta información es valiosa para diferenciarse y encontrar puntos únicos de venta en el desarrollo del software. Para el (Informante 8), *“Siempre se trata de dar un plus o un factor diferencial a lo que la competencia puede ofrecer, sabemos bien quienes son nuestra competencia directa y siempre tratamos de dar algo más a lo que ellos puedan dar.”*

La segunda red semántica denominada “Objetivo de desarrollo” está comprendida por los siguientes códigos: Problema, realismo y factibilidad, tiempo, recursos, presupuesto, especificidad y medibilidad,

capacidad operativa y alcance. Ver red semántica 2.

El objetivo del desarrollo es la finalidad del proyecto, en el cumplimiento del objetivo se verá plasmado el éxito o fracaso del desarrollo, este objetivo está directamente relacionado con los requerimientos y aspiraciones de los interesados en el proyecto.

Respecto al alcance, es el trabajo necesario para lograr el objetivo del proyecto, es un indicador de su tamaño y del esfuerzo necesario para terminarlo [8]. Para [9] crear una descripción exhaustiva del proyecto y sus resultados es el primer paso para definir el alcance. La principal ventaja de este proceso es que define los parámetros del producto, servicio o resultado final, esbozando cuáles de las necesidades recogidas se incluirán y cuáles no entrarán en el ámbito del proyecto.

El objetivo del proyecto también debe contar con ciertas características que ayudarán a guiar su planificación y ejecución hacia un cumplimiento exitoso,

una de estas características es que el objetivo debe enfocarse en el problema que se quiere resolver, la resolución de este problema a su vez generará un impacto positivo para el negocio como, por ejemplo: mejorará niveles de servicio, reducirá costos, agregará valor a la empresa, alineará normativas entre otras.

Otra característica para tomar en cuenta es el principio del “Realismo y factibilidad”, este principio hace notar que el objetivo no puede ser imposible de alcanzar, ya que puede llevar a la frustración y al fracaso de este, por esta razón tanto el presupuesto, los recursos y el tiempo deben ser analizados en función del objetivo y el alcance para de esta manera poder obtener una estimación lo más cercana posible de que es con lo que se cuenta para lograr el objetivo deseado.

Respecto al tiempo el (Informante 2) señala: “no es algo optimo trabajar especialmente con restricciones de tiempo porque en nuestro campo, de la programación como tú sabes se suelen

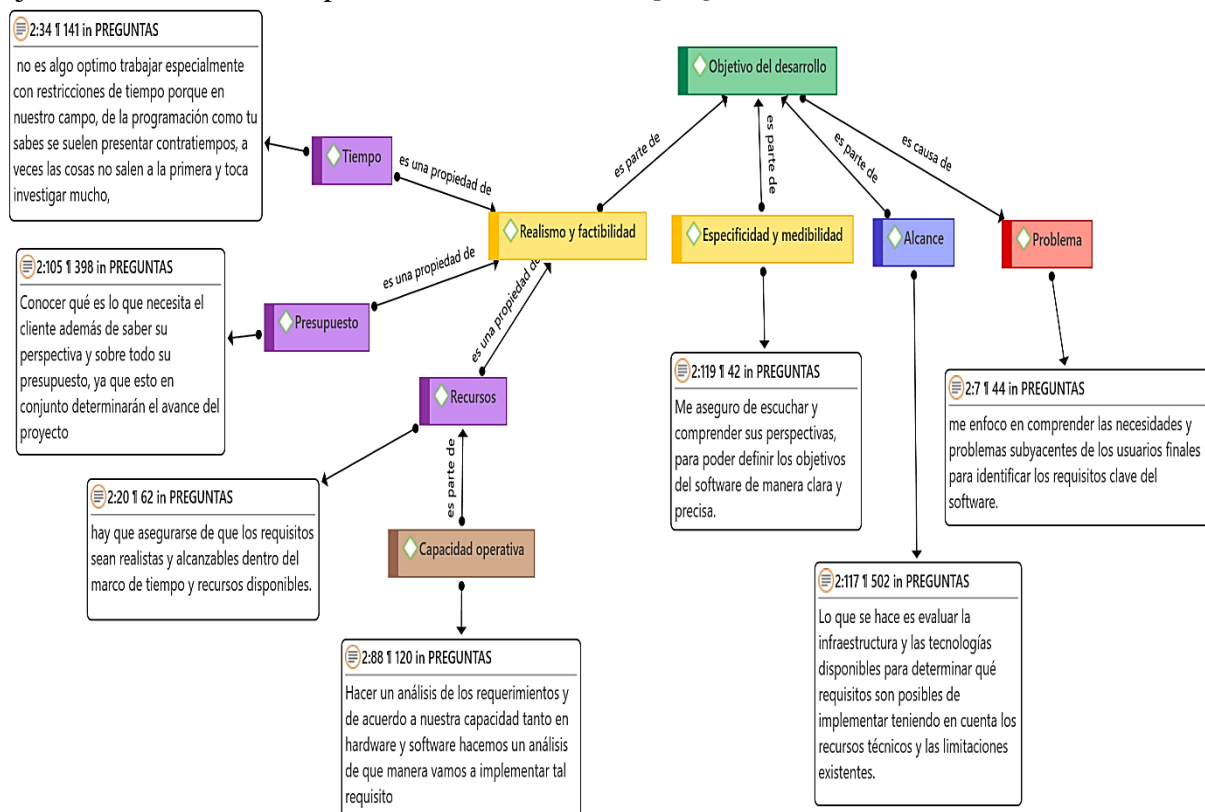


Figura 2 Segunda red semántica: Objetivo del desarrollo

presentar contratiempos, a veces las cosas no salen a la primera y toca investigar mucho”.

Referente al presupuesto para el desarrollo del proyecto, si bien no garantiza el cumplimiento del objetivo, puede mejorar algunos aspectos como la flexibilidad en el proceso de adquisición de equipos o materiales, mayores recursos disponibles, adquisición de mayor tecnología que podría facilitar y acelerar el proceso, permitir la realización de actividades complementarias, como capacitaciones o actividades de equipo que mejoren la cooperación y comunicación, . La (Informante 7) señala *“Conocer qué es lo que necesita el cliente además de saber su perspectiva y sobre todo su presupuesto, ya que esto en conjunto determinarán el avance del proyecto”*

Los recursos permiten que se realicen las tareas necesarias para alcanzar los objetivos del proyecto. Los recursos pueden ser financieros, humanos, físicos, tecnológicos, entre otros, y su uso adecuado es esencial para cumplir con los objetivos del proyecto.

Cada tipo de recurso juega un papel fundamental que puede optimizarse. Por lo tanto, es crítico asignar adecuadamente todos los recursos para la asignación correcta para garantizar que las tareas específicas del proyecto se realicen de manera eficiente, efectiva y en tiempo y forma. Además, es necesario el seguimiento y control para garantizar que los recursos estén siendo utilizados adecuadamente para cumplir los objetivos establecidos. Un adecuado uso hará que se amplíe la ventana de oportunidades de éxito explotadas en un proyecto. *“hay que asegurarse de que los requisitos sean realistas y alcanzables dentro del marco de tiempo y recursos disponible”* (Informante 1).

Otra de las características a tomar en cuenta es la “Especificidad y medibilidad” del objetivo, (Informante 1) *“Me aseguro de escuchar y comprender sus perspectivas,*

para poder definir los objetivos del software de manera clara y precisa.”

Sobre la tercera red semántica titulada “Calidad del software” emergen los siguientes códigos: tipo de aplicación, solución actual, mantenibilidad, escalabilidad, modularidad, arquitectura, seguridad de la información, estándares y normas, capacitación y soporte, participación de los interesados, usuario final. Ver red semántica 3.

Según [10] la calidad del software es el grado en que un sistema, componente o procedimiento satisface los deseos y expectativas del cliente o usuario, así como las especificaciones concretas. Satisface tanto las demandas o expectativas del usuario o cliente como los requisitos establecidos. Esto implica la evaluación de varios aspectos del software, como su funcionalidad, confiabilidad, usabilidad, eficiencia, seguridad y mantenibilidad. El propósito de la calidad del software es asegurarse de que el software sea confiable, eficaz, amigable para el usuario, seguro y que pueda acoplarse adecuadamente a las necesidades cambiantes del entorno.

La elección del tipo de aplicación es una parte fundamental en el desarrollo de un proyecto de software, ya que se puede asegurar que el sistema sea eficiente y escalable, sea útil para los usuarios, sea sostenible en costos y que se puedan reutilizar los códigos para futuros proyectos. Por lo tanto, es necesario analizar cuidadosamente las necesidades del proyecto para elegir el tipo de aplicación adecuado.

Otro aspecto importante es que se debe analizar la solución actual del problema, si es que la hay, esto con la finalidad de conocer de qué manera se está resolviendo el problema, con el objetivo de analizar si esa solución puede servir de algo en el futuro proyecto, ya que no todo debe ser malo, si es así, se puede empezar un desarrollo partiendo desde la solución actual en el caso de que el cliente esté de acuerdo o sino simplemente se empezará un

desarrollo desde cero. En el caso del (Informante 3) “*hay veces que el cliente ya cuenta con un sistema, pero no cumple con todas sus necesidades entonces es válido también analizar ese sistema y hasta analizar si es posible trabajar sobre ese mismo sistema*”.

Respecto al código mantenibilidad, es la capacidad del software para ajustarse a nuevos requerimientos a un costo justo y con baja probabilidad de introducir nuevos errores en el sistema al realizar los ajustes necesarios. [11]. (Informante 1) indica “*al mismo tiempo, diseño e implemento la arquitectura y estructura del software de manera flexible para acomodar cambios futuros sin afectar su funcionalidad principal*”, el código escalabilidad se refiere la capacidad de un sistema para adaptarse a demandas crecientes añadiendo más recursos. [11]. Es la capacidad de controlar mayores volúmenes de trabajo, usuarios, datos u otros recursos sin comprometer el rendimiento, la disponibilidad o la calidad del servicio.

Por su parte el modularidad del software es la capacidad que este tiene para

dividirse en módulos o componentes independientes, los módulos son unidades funcionales que se pueden desarrollar, probar y mantener de forma independiente, lo que facilita la reutilización, la actualización y la modificación del sistema, el (Informante 2) indica “*usamos microservicios para que todos los desarrollos sean modulares y que no esté todo centralizado en un mismo lugar*.” Esta característica facilita la escalabilidad al permitir agregar o eliminar componentes de manera flexible según las necesidades cambiantes del sistema.

Para que estos conceptos funcionen dentro de un desarrollo real es necesario que se haya tomado en cuenta la arquitectura de la aplicación, una arquitectura bien escogida hará que el software sea modular y por ende más escalable, el (Informante 2) señala, “*de nuestra parte nos manejamos con infraestructura en nube, Amazon Web Service (AWS), que prácticamente cuenta con todas las posibilidades en términos de almacenamiento, virtualización, hosting etc entonces adaptamos cada proyecto a lo que nos ofrece AWS como servicio*.” esto facilita

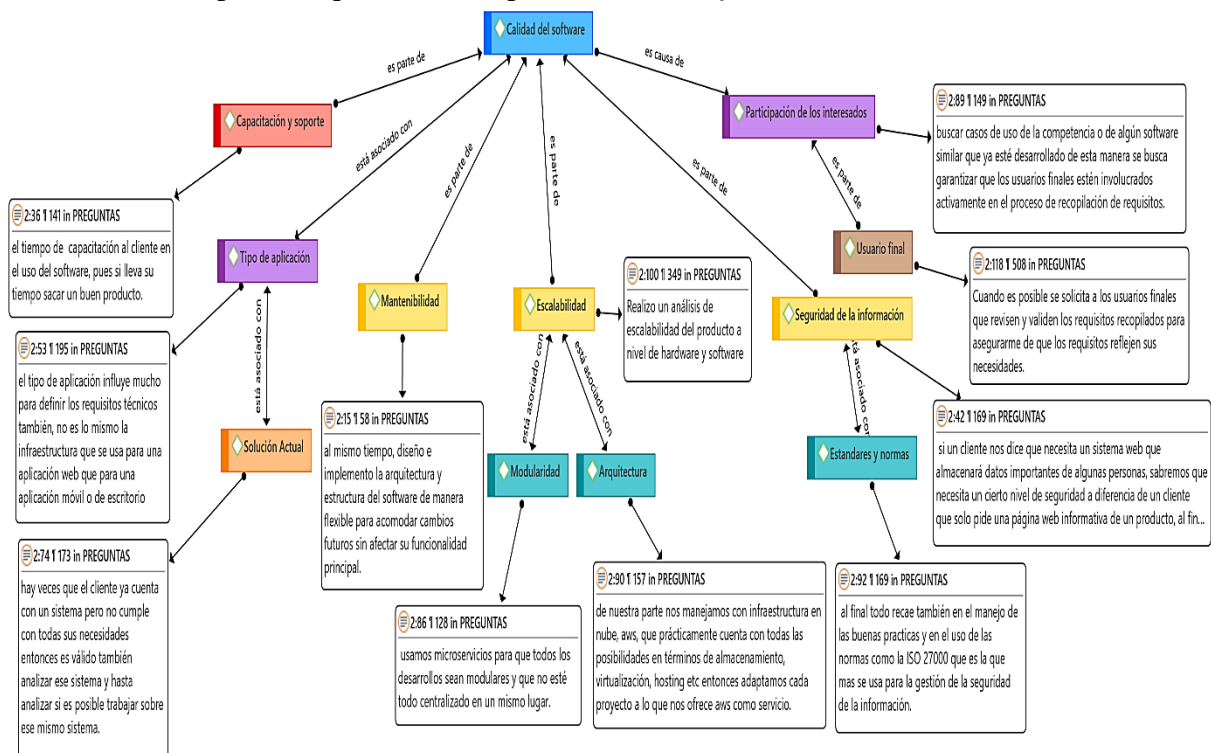


Figura 1 Tercera red semántica: Calidad del software

y permite el crecimiento y la adaptabilidad del sistema a medida que aumentan los requisitos y la demanda.

Todo sistema está destinado a manejar algún tipo de información para poder obtener un resultado y dependiendo de su propósito, su alcance y su ámbito de aplicación la información será más o menos significativa. Según el uso que se le dé al software existen diferentes categorías de información como información de usuario, información de negocio, información científica, información de salud entre otras. Todas y cada una con datos que pueden ir desde el teléfono de un usuario hasta registros médicos, resultados de exámenes o estadísticas de tratamiento. El (Informante 2) menciona *“si un cliente nos dice que necesita un sistema web que almacenará datos importantes de algunas personas, sabremos que necesita un cierto nivel de seguridad a diferencia de un cliente que solo pide una página web informativa de un producto (...)”*, información que debe estar protegida en todo momento.

Por este motivo, es esencial que los sistemas implementen diversas medidas de seguridad y privacidad, tales como encriptación, control de acceso, monitoreo constante, autenticación, entre otras. De esta manera se minimiza las posibilidades de que la información sea vulnerada de una u otra manera. El (Informante 2) señala *“(...) al final todo recae también en el manejo de las buenas prácticas y en el uso de las normas como la ISO 27000 que es la que más se usa para la gestión de la seguridad de la información”*.

Respecto a la capacitación y el soporte son componentes importantes de la calidad del software ya que, si los usuarios no están capacitados en el uso del software o si el soporte técnico no está disponible, esto puede afectar negativamente la calidad del software. Si los usuarios tienen problemas y no pueden obtener ayuda rápidamente, esto puede afectar la satisfacción del usuario e incluso causar daño a los datos y el negocio. Un soporte técnico sólido, que incluya

respuestas rápidas y resolución oportuna de problemas, es un punto clave para garantizar una buena calidad y además un alto índice de satisfacción por parte del usuario. El (Informante 2) menciona que *“el tiempo de capacitación al cliente en el uso del software, si lleva su tiempo sacar un buen producto.”*

Los usuarios finales en conjunto con patrocinadores, equipo de proyecto y stakeholders forman un conjunto de individuos que se denominan “La parte interesada”, la participación activa de este grupo de personas fomenta una comunicación mucho más clara, lo que desembocará en una toma de decisiones más efectiva, además, su participación, desde la planificación hasta la entrega del proyecto terminado, ayudará a asegurar que el software funcione correctamente, sea útil y satisfactorio para todas las partes involucradas, y asegurará una aceptación temprana y exitosa del mismo.

En la cuarta red semántica que se titula “Gestión de los requerimientos” emergen los siguientes códigos: Comunicación con el equipo de desarrollo, Herramientas para la gestión, metodologías ágiles, gestión de cambios en los requerimientos y experiencia del desarrollador. Ver red semántica 4.

Los requerimientos recopilados durante el proceso de elicitación necesitan ser gestionados y organizados en una estructura lógica para facilitar su comprensión y seguimiento. Esto puede implicar la creación de una matriz de trazabilidad de requerimientos, donde se establecen relaciones entre los diferentes requisitos y se rastrea su estado desde el inicio hasta el final del desarrollo del proyecto.

Respecto al código herramientas para la gestión están diseñadas para facilitar la recopilación, documentación, seguimiento, trazabilidad y gestión de los requerimientos. El (Informante 1) señala *“Utilizo herramientas como control de versiones y seguimiento de problemas para gestionar y*

documentar los cambios.” De igual manera el (Informante 1) indica “*Sí, suelo utilizar herramientas de gestión de proyectos como Jira o Trello para documentar y rastrear los requisitos recopilados*”, estas herramientas son muy parecidas la una con la otra, sin embargo, Jira fue construida especialmente para creadores de software por lo que podría tener una ventaja sobre Trello.

Ambas herramientas son capaces de trabajar con diferentes metodologías ágiles, y SCRUM es una de ellas, esta metodología es un marco de trabajo cuyo objetivo es mantener un control permanente sobre el estado actual del software. El cliente establece las prioridades, y el equipo SCRUM se organiza para averiguar cómo proporcionar los resultados de la forma más eficaz. [12]. . Esta metodología brinda una perspectiva flexible y colaborativa para la planificación, ejecución y entrega de proyectos. El (Informante 1) señala “*Suelo seguir la metodología de Scrum para gestionar los requisitos de manera iterativa, realizar revisiones con los stakeholders y adaptar el proceso de recopilación de requisitos según las necesidades cambiantes del proyecto*”

Debido a la naturaleza del software pueden existir cambios en los

requerimientos a medida que se desarrolla el proyecto por lo que es necesario afrontar esos cambios de una manera controlada para que el proyecto no sufra demasiados cambios especialmente en los tiempos de entrega. El (Informante 2) indica “*Junto con el equipo de desarrollo, analizamos el impacto de los cambios en el alcance, los plazos y los recursos del proyecto se ajusta la planificación y las iteraciones del proyecto según sea necesario para adaptarse a los nuevos requisitos*”. La experiencia del desarrollador siempre será una pieza fundamental cuando se trate de gestionar un cambio en los requerimientos, El (Informante 2) señala “*buscamos quien puede realizar cada tarea de acuerdo con la experiencia de cada miembro y de igual manera hacemos un aproximado del tiempo que nos tomaría realizar la tarea a cada uno.*”

3.2 Discusión

En el análisis de las cuatro redes semánticas, se identifican elementos clave relacionados con la elicitación de requerimientos en el proceso de un desarrollo de software.

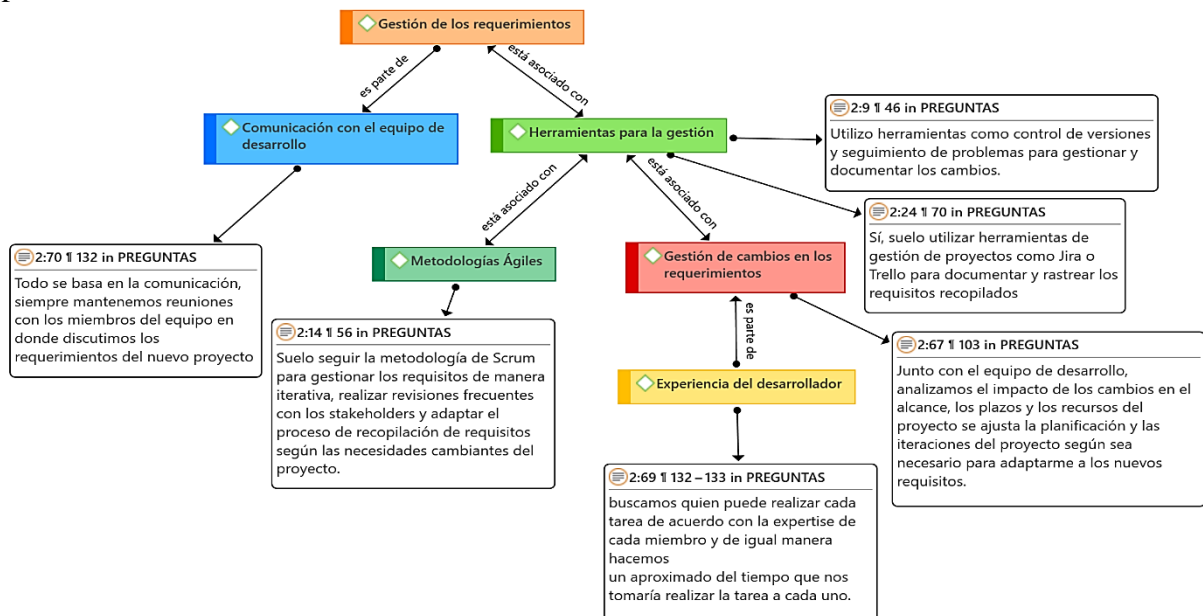


Figura 2Cuarta red semántica: Gestión de los requerimientos

En la primera red semántica, sobresale el valor del uso de canales para la comunicación bien definidos, efectivos y apropiados para interactuar con los interesados. Se mencionan reuniones, llamadas y correos electrónicos como medios utilizados para mantener una comunicación clara con los interesados. Además, se menciona el uso de herramientas de colaboración en línea para mantener a todos actualizados. Esta etapa inicial busca captar la mayor cantidad de información acerca de los interesados y sus necesidades, con el propósito de obtener una visión general completa de la situación.

Con relación a las técnicas de elicitación, se identifican diversas metodologías utilizadas para obtener información y capturar los requisitos del proyecto. Entre estas técnicas se mencionan entrevistas, cuestionarios y encuestas, observación, análisis de documentos existentes y análisis de casos de uso. Se resalta la importancia de una preparación previa que ayude a tener interacciones más productivas con los interesados y obtener la información necesaria de manera efectiva. Además, se menciona la utilización de preguntas predefinidas para abordar aspectos clave durante las entrevistas.

La gestión de la información recopilada se reconoce como un proceso fundamental que incluye el almacenamiento, organización y uso eficiente de los datos. La correcta organización de esta hace más fácil la manejabilidad, acceso y representación de los datos a lo largo del proyecto, mejorando así la toma de decisiones.

Las técnicas de interpretación de los requerimientos desempeñan un papel esencial en la comprensión y análisis profundo de los requisitos recopilados durante la elicitación. Se mencionan herramientas como diagramas de flujo, prototipos, especificaciones claras y documentación detallada, las cuales contribuyen a garantizar que los requisitos sean comprensibles para todo el equipo de desarrollo.

La definición de los requerimientos se aborda mediante técnicas de priorización, como la matriz de priorización y la técnica MoSCoW. Esta última categoriza los requisitos en función de su importancia y necesidad, desde aquellos críticos para el éxito del proyecto hasta los que no se implementarán. La técnica MoSCoW ayuda a establecer prioridades basadas en la importancia y el grado de necesidad de los requisitos, asegurando un enfoque adecuado en el desarrollo del software.

En cuanto a los requerimientos no funcionales, se destaca su importancia como especificaciones o restricciones que describen cómo debe ser el sistema al considerar aspectos como el rendimiento, la seguridad, la escalabilidad y la compatibilidad del software, así como la usabilidad y la experiencia del usuario.

Por último, se aborda la evaluación de la competencia como una estrategia para obtener una visión clara sobre las soluciones existentes en el mercado, sus características, fortalezas y debilidades. Esta información es valiosa para diferenciarse y encontrar puntos únicos de venta en el desarrollo del software.

4. Conclusiones

Concluida la investigación se puede afirmar que se ha tratado la gestión de requerimientos en el desarrollo de software desde una perspectiva integral y fundamentada. A través del análisis y la síntesis de diferentes aspectos clave, se ha evidenciado la importancia de un enfoque estructurado y efectivo para garantizar el éxito del proyecto y la calidad del software resultante.

Destaca la relevancia de establecer una comunicación clara y efectiva con los interesados desde las etapas iniciales, lo cual permite comprender sus necesidades y requisitos de manera precisa. Asimismo, se ha enfatizado la utilización de diversas técnicas de elicitación, como entrevistas, cuestionarios y observación, para recopilar información valiosa y priorizar los

requerimientos de manera adecuada, siendo la técnica de entrevistas la más usada por los participantes de la investigación.

La gestión de la información ha sido reconocida como un aspecto crítico en el proceso, ya que un adecuado almacenamiento y organización de los requerimientos permite un acceso fácil y eficiente desde el comienzo hasta el final del desarrollo del proyecto. Además, se ha subrayado la importancia de interpretar y definir los requerimientos de manera precisa mediante el uso de herramientas y técnicas como diagramas de flujo, prototipos y especificaciones detalladas.

La categorización y priorización de los requerimientos se presenta como un paso crucial para establecer un conjunto claro y detallado de requisitos funcionales y no funcionales. Además, se resalta la importancia de considerar aspectos de calidad del software, como el tipo de aplicación, la solución actual, la mantenibilidad, la escalabilidad, la modularidad, la arquitectura, la seguridad de la información, los estándares y normas, la capacitación y el soporte, así como la participación de los interesados y los usuarios finales.

Por último, se destaca la relevancia de utilizar herramientas adecuadas, como herramientas de control de versiones y de gestión de problemas, para facilitar la documentación, el seguimiento y la gestión de los requerimientos. Asimismo, se ha enfatizado la utilidad de las metodologías ágiles, como Scrum, para planificar y adaptarse de mejor manera a los cambios que puedan aparecer a lo largo del proceso.

En conjunto, las conclusiones obtenidas en este artículo científico respaldan la importancia de una gestión efectiva de los requerimientos en el desarrollo de software, proporcionando una base sólida para alcanzar los objetivos planteados al comienzo del proceso y la entrega de un software de calidad que cumpla con las expectativas de los interesados.

5. Referencias

- [1] G. O'Regan, *Concise Guide to Software Engineering From Fundamentals to Application Methods*, Gewerbestrasse 11, 6330 Cham, Switzerland: Springer, 2017.
- [2] S. U. J. H. Muhammad Tukur, «Requirement Engineering Challenges: A Systematic Mapping Study on the Academic and the Industrial Perspective,» King Fahd University of Petroleum & Minerals 2021, 2020.
- [3] M. Chemuturi, «Elicitation and Gathering of Requirements,» de *Requirements Engineering and Management for Software Development Projects*, New York, Springer, 2013, pp. 33,54.
- [4] G. Navas y A. Yagüe, «A New Way of Cataloging Research through Grounded Theory,» *applied sciences*, p. 3, 2023.
- [5] G. Navas y A. Yague, «Glaserian Systematic Mapping Study: An Integrating Methodology,» *Proceedings of the 17th International Conference on Evaluation of Novel Approaches to Software Engineering - ENASE*, 2022.
- [6] Y. R., *The requirements Engineering Handbook*, Artech House, 2004.
- [7] IEEE, *Especificación de Requerimientos de Software*, IEEE/ANSI 830-1998, Autor, 1998.
- [8] G. R. Pablo Lledó, *GESTIÓN DE PROYECTOS*, Buenos Aires: Prentice Hall - Pearson Education, 2007.
- [9] F. U. GARCÍA, *Guía para la gestión del alcance, tiempo y costos de los proyectos de desarrollo Mega*, COSTA RICA, 2018.

- [10] IEEE, IEEE Standard Glossary of Software Engineering Terminology, 1990.
- [11] I. Sommerville, Software Engineering, vol. 9, Madrid: Pearsn, 2011.
- [12] P. Abrahamsson, O. Salo y J. Ronkainen, Agile Software Development Methods:, Oulu: VTT Publications 478, 2002.