



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y
URBANISMO**

**ESCUELA PROFESIONAL DE INGENIERÍA DE
SISTEMAS**

TESIS

**COMPARACIÓN DE MAPAS CONCEPTUALES
BASADO EN ONTOLOGÍA Y MODELO
CONCEPTUAL, PARA REDUCIR LA AMBIGÜEDAD
DE LOS REQUERIMIENTOS EN LA ELABORACIÓN
DEL PRODUCT BACKLOG DEL MARCO DE
GESTIÓN ÁGIL SCRUM**

**PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO DE SISTEMAS**

Autor:

Bach. Cantos Morante Augusto Enrique

Asesor:

Mag. Mejia Cabrera Heber Ivan

Línea de Investigación:

Infraestructura, tecnología y medio ambiente

Pimentel – Perú

2020



DEDICATORIA

Dirigido con mucho cariño a mis padres María Magdalena Morante Granados y Segundo Augusto Cantos Bazán, por ese apoyo incondicional y esas ganas de verme profesional, a mis hermanos por esa motivación extra para tener fuerzas suficientes para realizarme profesionalmente. A Jorge Ricardo Negrete Condori quien me ayudó incondicionalmente en mi educación y quien luchó por quererme ver profesional, a la paciencia de mi madre, la mujer que día a día con el esfuerzo, su lucha por sacarnos adelante e inculcarme valores, y también una dedicatoria especial a mis docentes por todas las enseñanzas recibidas duran mi carrera universitaria.

AGRADECIMIENTO

En primer lugar, a nuestro Dios padre que me brindó la oportunidad de cumplir uno de mis objetivos, por enseñarme el camino idóneo en mi vida, así también agradecer a todas las personas que estuvieron involucrados de manera indirecta y directa en la formación académica que obtuve en la universidad, a mi grupo de amigos en la universidad “LOS IRRESISTIBLES”, por esa unidad amical que siempre nos brindamos, tanto en lo académico y familiar, y no dejar de agradecer a nuestro tutor Heber Ivan Mejia Cabrera, por brindar sus conocimientos profesionales que ayudó a concluir este trabajo de grado.



RESUMEN

La Ingeniería de Requerimientos (IR), es un proceso que permite describir el comportamiento funcional de un sistema de software, es por ello que es el más importante para el éxito de los proyectos y también se puede decir que es el mayor de los retos que enfrenta, la especificación de ingeniería de requerimientos; muchas empresas sufren la mala claridad de los requerimientos del interesado, provocando en las mayorías de los casos en el software las malas interpretaciones generando la ambigüedad en el momento de la captura de requerimientos. Esta investigación tiene como objetivo proporcionar una posible solución en la ambigüedad de los requerimientos, el cual ayudará de gran manera en la reducción de los errores de los requerimientos durante el proceso de especificación de la IR. y así poder lograr una alta calidad en los proyectos de software. Por tales motivos se ha propuesto realizar un análisis comparativo de 7 técnicas de especificación de requerimiento que han sido elegidos según un top de técnicas encontradas por el criterio de uso en las diferentes investigaciones a empresas de desarrollo de software. (Besrou, Bin, et al., 2016)(Hurtado, n.d.)(A. Toro, 2016)(G & Losavio, 2013), y como regla de medición hemos considerado el factor rendimiento el cual ha contenido indicadores que han sido evaluados y comprobados durante el desarrollo de un caso de estudio de negocio a la empresa CelsySoft & Bussines S.R.L implementando las técnicas de mapas conceptuales basados en ontología y modelo conceptual, para posteriormente obtener resultados que prueben determinar cuál es la mejor técnica a utilizar. De tal manera ha permitido esta investigación brindar aportes a los analistas, programadores de software, docentes y estudiantes para los diferentes proyectos de software que desarrollen.

PALABRAS CLAVES: Ingeniería de requerimientos, mapas conceptuales, modelo conceptual, antología, ambigüedad, requerimientos de software.



ABSTRACT

Requirements Engineering (IR), is a process that allows describing the functional behavior of a software system, which is why it is the most important for the success of the projects and it can also be said that it is the greatest of the challenges that faces, the requirements engineer specification; Many companies suffer from the poor clarity of the requirements of the interested party, causing in most cases in the software the misinterpretations generating ambiguity at the time of the capture of requirements. This research aims to provide a possible solution in the ambiguity of the requirements, which will greatly help in reducing the requirements definition errors during the IR specification process. and thus be able to achieve high quality software projects. For these reasons it has been proposed to perform a comparative analysis of 7 requirement specification techniques that have been chosen according to a top of techniques found by the criteria of use in the different investigations of software development companies. (Besrour, Bin, et al., 2016) (Hurtado, nd) (A. Toro, 2016) (G & Losavio, 2013), and as a measurement rule we have considered the performance factor which has contained indicators that have been evaluated and verified during the development of a case study of business to the company CeslySoft & Bussines SRL implementing the technology of conceptual maps based on ontology and conceptual model, to subsequently obtain results that approve to determine which is the best technique to use. In this way it has allowed this research to provide contributions to analysts, software programmers, teachers and students for the different software projects they develop.

KEYWORDS: Requirements engineering, conceptual maps, conceptual model, anthology, ambiguity, software requirements.



Contenido

DEDICATORIA	2
AGRADECIMIENTO.....	3
RESUMEN	4
ABSTRACT.....	5
I. INTRODUCCIÓN.....	11
1.1. Realidad Problemática.....	11
1.2. Antecedentes de Estudio.....	15
1.3. Teorías relacionadas al tema.....	22
1.3.1. Ingeniería de Requerimientos.....	22
1.3.2. Tipos de Requerimiento	22
1.3.2.1. Requerimientos Funcionales.....	22
1.3.2.2. Requerimientos No Funcionales	23
1.3.3. Estepas de la Ingeniería de Requerimientos.....	24
1.3.3.1. Elicitación de Requerimientos	24
1.3.3.2. Análisis de Requerimientos	24
1.3.3.3. Especificación de Requerimientos	25
1.3.3.3.1. Procesos de especificación de requerimientos	26
1.3.3.3.2. Técnicas de Especificación	28
1.3.3.3.2.1. Mapa conceptual	28
1.3.3.3.2.2. Casos de uso.....	29
1.3.3.3.2.3. Modelo conceptual	29
1.3.3.4. Validación de Requerimientos.....	35
1.3.4. Ontología Basada en Ingeniería de Requerimientos.....	35
1.3.5. Marco de Gestión Ágil Scrum.....	37
1.3.5.1 Planificación del Backlog.....	38
1.4. Formulación del Problema.....	39
1.5. Justificación e importancia del estudio.....	39
1.6. Hipótesis	40
1.7. Objetivos.....	40
1.7.1. Objetivo General.....	40
1.7.2. Objetivo Especifico.....	40
II. MATERIAL Y METODOS.....	41
2.1. Tipo y Diseño de la Investigación.....	41
2.1.1. Tipo de la Investigación	41
2.1.2. Diseño de la Investigación	41
2.2. Población y Muestra.....	41
2.2.1. Población.....	41
2.2.2. Muestra.....	42



2.3.	Variables, Operacionalización	42
2.3.1.	Variable Independiente	42
2.3.2.	Variable dependiente	42
2.3.3.	Operacionalización	42
2.4.	Técnicas e instrumentos de recolección de datos, validez y confiabilidad	44
2.4.1.	Abordaje metodológico	44
2.4.2.	Método y técnicas de recolección de datos	44
2.4.2.1.	Instrumentos de recolección de datos.....	44
2.5.	Procedimiento de análisis de datos.....	45
2.6.	Criterios éticos	45
2.7.	Criterios de rigor científico	45
III.	RESULTADOS	46
3.1.	Resultado en tablas y figuras	46
3.1.1.	Estudio de Factibilidad	46
3.1.2.	Consistencia de Requerimiento	50
3.1.3.	No Ambiguo de Requerimiento.....	51
3.2.	Discusión de resultados	52
3.3.	Aporte Practico	53
3.3.1.	Comparar las técnicas de especificación de la ingeniería de requerimientos ..53	
3.3.1.1.	Características de ontología y Lenguaje de Modelado Unificado (UML) ..55	
3.3.2.	Seleccionar los indicadores de rendimientos para la comparación	61
3.3.3.	Implementar las técnicas seleccionadas en un proyecto modelo	62
3.3.3.1.	Ontología	62
3.3.3.1.1.	Mapeo y Reglas de Inferencia	63
3.3.3.2.	Modelo Conceptual	65
3.3.3.2.1.	Reglas de modelado conceptual	66
3.3.3.3.	Caso de estudio: Caso de Negocio CESLYSOFT & BUSSINES S.R.L68	
3.3.3.3.1.	Modelado conceptual basado en UML	72
3.3.3.3.2.	Mapa conceptual basado en ontología	79
3.3.3.3.2.1.	Construcción de una ontología.....	79
3.3.3.3.3.	Metodología Scrum	87
3.3.3.3.3.1.	Descripción de sistema	87
3.3.3.3.3.2.	Personas y Roles	88
3.3.3.3.3.3.	Historias de Usuario	88
3.3.3.3.3.4.	Product Backlog.....	89
IV.	CONCLUSIONES Y RECOMENDACIONES	94
	REFERENCIAS	96
	ANEXOS	100
	Anexo 01: Acta de Constitución del proyecto (Project Charter)	100
	Anexo 02: Especificación de requerimientos software	112
	Anexo 03: Focha de Observación	131



INDICE DE FIGURAS

Figura 1: Tipo de requerimientos no funcionales 23

Figura 2: Proceso de especificación de requerimientos de Software..... 26

Figura 3: Mapas conceptuales producidos..... 28

Figura 4: Diagrama de caso de uso 29

Figura 5: Clase y asociación UML..... 30

Figura 6: Diagrama de Clases UML..... 31

Figura 7: Clases con atributos y operaciones 32

Figura 8: Jerarquía de generalización 33

Figura 9: Jerarquía de generalización con detalles agregados 34

Figura 10: La asociación y agregación 35

Figura 11: Arquitectura ontológica 36

Figura 12: Ciclo de desarrollo ágil 37

Figura 13: Marco de gestión ágil Scrum 38

Figura 14: Tiempo de respuesta..... 49

Figura 15: Requerimientos de consistencia 50

Figura 16: Requerimiento de no ambiguo 51

Figura 17: Diagrama de clases a partir del modelo conceptual 78

Figura 18: Ontología generada a partir de los requerimientos iniciales..... 81

Figura 19: Arranque de la herramienta Protege 5.5.0 83

Figura 20: Levantamiento de plugin herramienta protege 5.5.0 84

Figura 21: Herramienta Protege 5.5.0..... 84

Figura 22: Creando mi ontología 85

Figura 23: Clases y sub clases de mi ontología..... 85

Figura 24: Grafico ontológico 86

Figura 25: Proceso Grafico ontológico 86



Figura 26: Mapa conceptual basado en ontología	87
Figura 27: Listado de requerimientos (Product Backlog)	89
Figura 28: Lista de requerimientos (Product Backlog)	90
Figura 29: Sprint mantenimiento cliente	90
Figura 30: Insertar tipo de trabajador	91
Figura 31: Actualizar tipo de trabajador	91
Figura 32: Eliminar tipo de trabajador	92
Figura 33: Mantenimiento tipo de trabajador	92



INDICE DE TABLAS

Tabla N° 0 1: El 71 % en el 2015 fracasaron o fueron desafiados	13
Tabla N° 0 2: Formas de escribir una ERS	26
Tabla N° 0 3: Indicadores y fórmulas para detallar las variables, técnicas e instrumentos de recolección de datos.	43
Tabla N° 0 4: Tiempo de respuesta de la Técnica Modelo Conceptual	47
Tabla N° 0 5: Tiempo de respuesta de la Técnica mapas conceptuales basado en ontología.....	48
Tabla N° 0 6: Cálculo de consistencia de requerimiento	50
Tabla N° 0 7: Cálculo de No ambiguo de requerimiento.....	51
Tabla N° 0 8: Cuadro comparativo basado en el uso de las técnicas en la IR.....	54
Tabla N° 0 9: Comparación de características OWL y UML	56
Tabla N° 10: Componentes y herramienta para una Ontología	63
Tabla N° 11: Componentes y herramienta para una UML.....	66
Tabla N° 12: Categorías de reglas de diseño y declaraciones	67
Tabla N° 13: Requerimientos del Cliente (Iniciales)	69
Tabla N° 14: Requerimientos Funcionales por Procesos	71
Tabla N° 15: Ejemplo de implementación – relaciones después de la extracción de entidades.	73
Tabla N° 16: Entidades extraídas del caso de estudio de negocio.....	74
Tabla N° 17: Ejemplo de implementación – relaciones después de la extracción de entidades.	75
Tabla N° 18: Entidades y atributos.....	77
Tabla N° 19: Definición de Requerimientos iniciales	80
Tabla N° 20: Requerimientos Funcionales por Procesos	82
Tabla N° 21: Persona y Roles.	88



I. INTRODUCCIÓN

1.1. Realidad Problemática

La Ingeniería de Requerimientos (IR), es un proceso que permite conocer el comportamiento funcional del software, ya que los desarrolladores deben comprender para construir el producto que satisfaga las necesidades del cliente (Sommerville, 2011).

Según (Davis, 1993) muchas empresas abordan los requerimientos de manera superficial realizando descripciones sencillas, sin embargo, algunas otras le dedican mayores recursos a la especificación detallada de los servicios que debe brindar el software a los procesos y al usuario.

Las empresas que no abordan con suficiente preocupación la especificación de los requerimientos, tienen resultados no esperados, todo esto debido a la falta de claridad entre los requerimientos de usuario y del software. Los requerimientos del usuario deben ser especificados en lenguaje natural sin ambigüedad o en artefactos gráficos que permitan tener plena comprensión. Por otro lado, los requerimientos del sistema son formas más detalladas con exactitud de lo que se implementará en el sistema, estos requerimientos de software también se pueden clasificar como requerimientos funcionales (RF) y requerimientos no funcionales (RNF): los RF es la parte principal de cómo debería de comportarse el software, y los RNF son restricciones o funciones que nos da el software, (Sommerville, 2011).

Según (Arias, 2014), la educación de los requisitos que se presenta en la fase 1 del ciclo de vida del software, se reflejan problemas entre el analista y el interesado, debido al poco interés de claridad del interesado, las malas interpretaciones del



analista de los requerimientos solicitados para el software, provocando que en el producto en las mayorías de los casos no satisfaga con las necesidades y las perspectivas del cliente, generando un discurso de ambigüedad léxica o morfológica, ambigüedad sintáctica o estructural y ambigüedad semántica (Polisémica y referencial), que dificulta a una buena comprensión y no a una mala ambigüedad de los requerimientos de software.

Según (De-la-Cruz-Londoño & Castro-Guevara, 2014), nos expresa que hoy en día para poder llevar una buena calidad de software todo depende de los requisitos y que va de la mano con los técnicas de la IR., que consta de 4 etapas: Elicitación, análisis, especificación y validación, dichas técnicas son las que ayudara a llevar una mejor calidad de requisitos para un mejor planteamiento del desarrollo del software solicitado por el cliente. Por consiguiente, que se dice que en las empresas de software de la ciudad de Cali (Colombia) el 48.7%, no son capaces de establecer puntos de criterios para una mejor aceptación de los requerimientos, el 43.6% no ejecutan un seguimiento de administración a los requerimientos de software, por otro lado, Chile y Nueva Zelanda no cuentan con un personal calificado, cuentan con recursos económicos muy bajos, se puede observar que las empresas no utilizan las herramientas que no brindan la ingeniería de requerimientos.

Según (CHAOS, Standish Group 2015), nos informa que hizo un estudio de 50000 proyectos de software en todo el mundo, estos resultados nos indica que hay mucho más trabajo por resolver, si queremos ver resultados con éxito. Analizando la **Tabla N° 01** se expresa que el éxito del desarrollo del software los cinco primeros años es un balanceo de porcentaje ni aumento y ni disminuye, nos muestra un oscilante sobre el 29%, el fracaso en un medio de 19% y un 50% en entorno de descuido, eso quiere decir que el 71% de los proyectos en el 2015 fracasaron o fueron desafiados.



Tabla N° 0 1: El 71 % en el 2015 fracasaron o fueron desafiados

	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

Fuente: CHAOS, Standish Group 2015

Es interesante este informe por se muestra que no hay mucha influencia ni de metodologías ni de ciclos de vida, etc. Es decir, hoy en día existen otras dificultades que impidan o afecten al éxito, ya que por ahora muchas de las empresas sufren de fracasos en los proyectos de software.

Según (Darío & Torres, 2014) explica con la “Guía para la Integración de Métodos Formales de Ingeniería de Requerimientos en Procesos de Desarrollo Ágil”, que el primordial problema que mantiene el proceso de uso común de Ingeniería de Requerimiento es la gestión sobre los requerimientos de software, teniendo una observación, de que las herramientas que existen alrededor del mundo no son inestables, para que los ingenieros de software desarrollen con facilidad y efectividad, este aporte por más de dar ayuda al lector, logrará proporcionar los elementos visuales verdaderamente necesarios que sirvan para tomar decisiones en el momento adecuado en la integración de todas las metodologías ágiles, utilizando los elementos o técnicas que facilita la IR con las etapas de elicitación, especificación, análisis y validación, sabiendo que en el ámbito de la ingeniera de requerimiento mucho de las empresas no cumplen con las especificaciones de requerimientos de calidad, retrasando en tiempo y costos para los proyectos de software planificados.



Por consiguiente, a los problemas antes observados respecto a la ambigüedad de la especificación de requerimientos, los investigadores en ingeniería de software han dedicado mucho tiempo en indagar esta situación mediante el desarrollo de diversos procesos, métodos, técnicas que permitan optimizar la elaboración de requisitos, en comprender las tendencias en el estado actual para identificar los retos que los ingenieros de software enfrentan en la actualidad, (Raunak & Binkley, 2017).

Hoy en día uno de los factores primordiales de los ingeniería de software es la ambigüedad, por lo cual es el mayor reto que enfrentan la ingeniería de requerimientos, como por consecuencia conllevan a muchos errores en el desarrollo del software, por esta razón, los investigadores se centran en el problema que emerge en las primeras etapas de la ingeniería de requisitos, desarrollaron una ontología en la cual se construyen en dos capas la primera capa, la ontología se basa en los requisitos que se han obtenido en una forma de lenguaje natural, esta capa proporciona los requisitos en un lenguaje formal legible, en los que se aplican las reglas de inferencia y razonamiento para detectar inconsistencia en los requisitos; La segunda capa convierte la requisitos producido ontología a un mapa conceptual mediante el uso de un algoritmo, (Mohamed, Ellatif, & Farhan, 2017), asimismo el modelo conceptual pueden desplegarse en diagramas de clase durante las fases de diseño y ejecución del proyecto de software, incluso un modelo conceptual parcialmente automatizado puede ahorrar un tiempo significativo durante la fase de requisitos, a partir de especificaciones funcionales escritas en lenguaje natural de forma automatizada, (Vidya Sagar & Abirami, 2014) .



1.2. Antecedentes de Estudio

(Buestán-Borja & Cuji-Torres, 2013), en la investigación “**Metodología para la Especificación de Requerimientos de Software Baso en el Estándar IEEE 830-1998**”, en la Universidad Politécnica Salesiana Sede Cuenca.

Muchas de las empresas utilizan servicios de sistemas automatizados, sin embargo, es uno de los factores que los sistemas fracasen, obteniendo retrasos en los proyectos de software y de la calidad, ya que dichos proyectos de software no llegan a culminar sus objetivos, debido a los problemas ya mencionados anteriormente, muchas de las empresas deciden obtener sistemas extranjeros, luego ser personalizados, obteniendo más pérdida de dinero y tiempo de lo pensado.

Es por ello el estudio realizado, menciona que a pesar de que existan herramientas que aporten a los procesos y estándares de calidad, mientras no cumplan las necesidades del usuario no se podrán obtener un buen sistema. Su total importancia en el proyecto en mención consistió, en que las especificaciones se describan con claridad y sin ambigüedad, que no tenga una mala gestión de requerimientos, con el fin de que el desarrollo de software no fracase.

Se propuso una metodología iterativa e incremental apoyada por (Sommerville, 2011); en esta propuesta utilizó 4 etapas de la IR: Elicitación, análisis, especificación y validación. También se incluyó a la metodología, el estándar (IEEE-STD-830, 1998), que ayuda a concluir las herramientas adecuadamente en base a las necesidades del usuario. En esta investigación se basó en las plantillas de (Duran T, 2000), en las plantillas se les permitió describir los RNF y RF del sistema. También se implementó un prototipo (SysToErs) para decepcionar toda la información del



documento de la especificación realizada.

Se verifico que la Ingeniería de Requerimiento no es muy utilizado para los desarrolladores, buscando la implementación de la calidad del software sin ser considerados todas las necesidades del usuario, desperdiciando tiempos y costos del producto.

(Silvana del Valle, 2013) en la investigación **“Elicitación y Especificación de Requerimientos No Funcionales en Aplicaciones Web”** en la Universidad Nacional de la Plata.

La complejidad es importante para el desarrollo del sistema, asimismo es la funcionabilidad del software que deberá ser implementada para satisfacer todas las necesidades del usuario, por otro lado, siempre existen dificultades en el desarrollo de las necesidades que limitan en la construcción del sistema.

En el ámbito de la Ingeniería de Requerimiento no se ofrece igualdad al momento de la captura de los requerimientos, se dice que las metodologías que realmente se utilizan en el entorno de aplicaciones Web no son completas y no se asemejan en el despliegue de los Requerimientos no funcionales a lo largo del ciclo de vida.

Los investigadores propusieron dar solución para desarrollar las técnicas y alineamientos para las estepas de la IIR de elicitación, especificación y plantillas definidas como soporte en formato Excel, que sean acorde en los RNF en el entorno de aplicaciones Web; plantillas de elicitación para obtener los requerimientos no funcionales web de calidad y matriz de control de autorización de accesos, plantillas para la elicitación de los requerimientos no funcionales web – restricciones, una



matriz de requerimientos no funcionales que da soporte y permite el seguimiento del etapa de los requerimientos durante la ejecución de los dos procesos del desarrollo.

En conclusión, esta investigación se realizó en un caso de estudio para la Elicitación y Especificación de Requerimientos no Funcionales, ejecutando las técnicas y plantillas demostradas, para que se puedan aplicar en un marco de desarrollo de aplicaciones Web.

(Segura-Fiquitiva & Lopez-Ruis, 2015) desarrollaron el trabajo de investigación **“Prototipo de Aplicativo para Especificar Requerimientos de Software”** en la Universidad Católica de Colombia.

Los investigadores consideran, que los inconvenientes más relevantes que carecen los proyectos de software son los malos entendimientos de los problemas y la poca claridad de las necesidades del cliente. Es por ello que en el desarrollo y aceptación de los proyectos realizados el ingeniero de software y el analista encuentra un 68% de proyectos fracasados por una mala interpretación o gestión de requerimientos.

Propusieron como solución a la metodología Scrum, estos investigadores aceptan que pueden equivocarse en transcurso que se va desarrollando el proyecto, pero la mejor manera de solucionar los errores es que deberían de poner en conocimientos las funcionalidades de la metodología y así poder construir el producto, quiere decir que las especificaciones sean correctas, se describan con claridad, sin ambigüedades, al momento de captar las necesidades de los usuarios o clientes; de esta forma, se pretende de minimizar los problemas a la mala interpretación de los requerimientos en el desarrollo de software.



Como resultado se obtuvo el diseño de un prototipo, que con gran importancia se logró capturar las necesidades de los stakeholders, creando el documento de elicitación de requerimientos en la norma IEEE 830, y a medida que los usuarios avanzan se crea el formato con la norma especificada como entregable.

(Cáceres-Saldaña, 2014) desarrollo el trabajo de investigación “**Metodología para el Reúso Efectivo de Patrones de Requisitos en la Ingeniería de Software**” en la universidad de Piura.

Para los investigadores el software en la actualidad es un factor primordial para las empresas, hoy en día se puede considerar como una disciplina de importancia y económica, que responde a las necesidades del mercado empresarial.

Muchos estudios realizados evidenciaron, que por la mala gestión de los requisitos fue la causa que fracasaron los proyectos de software, llegaron a la conclusión que los errores en la especificación de los requisitos en etapas posteriores consideran que resultan costosos, por último, afectan los proyectos de software que se están realizando.

Los investigadores pusieron como solución desarrollar una metodología para la indexación y recuperación efectiva de patrones de requisitos, esta propuesta servirá para mejorar la especificación de requisitos de software, que orientan pasos que deberían ser utilizados los analistas para el reúso efectivo de patrones a la etapa de elicitación de un proyecto de software.

La solución fue desarrollada para que el analista puedes cumplir con las etapas asignadas, cada etapa está dividida en elementos: entradas, herramientas y técnicas,



criterios de validación y salidas, asimismo se debe detallar la metodología en un conjunto para aprobar los procesos planteados y así poder tener un buen resultado.

Se obtuvo como resultado final la estructuración y el perfeccionamiento durante muchos años de manera empírica, consistente o inconsistente, empleando la metodología para facilitar su aplicación, esta metodología le otorgo una confiabilidad al reuso de patrones y que sea incluido con éxito en los proyectos de software realizados.

Por conclusión, se debe destacar que esta metodología es un aporte para aumentar la mejorara en los trabajos de los desarrolladores poniendo en conocimientos practico la ingeniería de requisitos, siendo base conceptual para el desarrollo de una herramienta CASE que permita implementar un repositorio de patrones de requisitos.

(De la Cruz-Londoño & Castro-Guevara, 2015) en la investigación **“Metodología para la Adquisición y Gestión de Requerimientos en el Desarrollo de Software para Pequeñas y Medianas Empresas (Pymes) del Departamento de Risaralda”** en la Universidad Tecnológica de Pereira.

En esta investigación su total importancia es en enfocarse en las empresas que desarrollan software, tales empresas no cuentan con fuerte económico para implementar y alimentar un prototipo de calidad, es por ello que los procesos de la ingeniería de software no están totalmente definidos.

Como solución los investigadores propusieron para realizar procesos de ingeniería de requerimientos, metodología aplicando técnicas para el problema de cómo se debe plantear en las empresas de Risaralda (Colombia), con el fin de disminuir los costos



y tiempo durante la elaboración del Software.

La solución fue desarrollada en tres fases en la cual se consideró: selección de datos, arquitectura de la metodología para la adquisición y gestión de requerimientos y por último se realizó la validación, empleando las Técnicas de Ingeniería de requerimientos: Elicitación, Especificación, Análisis y validación.

Con el fin de dar soporte a la metodología se desarrolló un aplicativo web, el objetivo de brindar apoyo a los procesos y herramientas para implementar la metodología en las empresas, identificando los inconvenientes que presenta la ingeniería de requerimientos para los desarrolladores de software.

Por conclusión, la propuesta metodológica se estructuró en una guía de trabajo que permita abordar a través de las fases planteadas por la metodología, con el propósito que la metodología cree proyectos de calidad, definiendo las herramientas que se aplican a las fases.

(Pardo-Pastrana, 2017) en la investigación **“Un Método de Elicitación de Requisitos para SCRUM Compuesto por Inception Deck y Modelos de Proceso de Negocios (BPMN)”** en la Universidad de San Buenaventura Seccional Cali.

Durante muchos años la ingeniería de software se han alarmado con la baja del éxito y alza del fracaso de los proyectos, buscando la manera de dar solución definitiva a este problema que se está enfrentado el área de ingeniería de software, así como resultados que brinda (CHAOS, Standish Group), que viene realizando desde los años de 1994, los reportes de los proyectos que han fracasado o han tenido éxito. Por otro ámbito los procesos de elicitación de requisitos sufren de manera contrastante



problemas con el éxito del proyecto, asimismo se puede expresar, que el fracaso es por la ambigüedad que existe en los documentos de los requisitos.

Los investigadores propusieron como solución un método de elicitación de requisitos, que les permite la mejora significativa en el análisis de software en los proyectos, unificando la técnica del Inception Deck manteniendo una documentación no ambigua mediante los modelos de BP.

La solución se desarrolló utilizando una serie de preguntas adecuadas de la actividad planteada en la investigación (elevator pitch) del Inception Deck y modelos de procesos de negocios desde el pre-análisis, estableciendo representaciones gráficas de los requisitos con BPMN del (walking skeleton), posteriormente con el método de elicitación de requisitos para SCRUM.

Como resultados se obtuvo de un caso de vehículos de pasajeros, llevando a cabo las técnicas Inception Deck con responsable asignado de maximizar el producto desarrollado y un líder de la empresa, además contando con equipo SCRUM de manera que se pudiera agrupar lo que el producto desea plantear y realizar un desarrollo a la medida, práctico y ajustándose a las necesidades que solicite la empresa.

Los investigadores llegaron a la conclusión que la propuesta apoyara el desarrollo y la disminución considerablemente de la ambigüedad en las historias de usuarios, al reemplazar los artefactos por modelos de procesos de negocios.



1.3. Teorías relacionadas al tema

1.3.1. Ingeniería de Requerimientos

En la etapa de la construcción del software, es necesario que los desarrolladores se involucren en el problema que se va a desarrollar, quiere decir que la solución que se quiera construir sea correcta y útil. Asimismo, se cabe mencionar que los requerimientos del sistema es uno de los factores que implica en la edificación de los proyectos de software, por lo tanto, los problemas que sufren los desarrolladores es la mala captura e interpretación de los requerimientos con el cliente (Cueva & Sucunuta, 2017).

1.3.2. Tipos de Requerimiento

Según (Sommerville, 2011) los requerimientos de software son objetivos y problemas que la empresa quiere resolver, estos se clasifican en requerimientos funciones y requerimientos no funcionales

1.3.2.1. Requerimientos Funcionales

Son enunciados que definen las funciones del sistema como debería de reaccionar, estos requerimientos van depender del software que se esté desarrollando, depende a la empresa que solicite los requerimientos para el software que se va a implementar. Sin embargo, los requerimientos funcionales más específicos del sistema son las que definen las funciones, sus entradas y salidas, sus excepciones, etc. (Sommerville, 2011).



1.3.2.2. Requerimientos No Funcionales

Son requerimientos que no tienen relación de forma específica con el sistema que se entrega a los usuarios, asimismo tiene que ver con propiedades que limitan el sistema, como fiabilidad, tiempo y espacio. Para poder implementar los requerimientos existen dos razones: estos requerimientos afectan más a la arquitectura general de un sistema, quiere decir, si un requerimiento no funcional individual, como un requerimiento de seguridad, estos podrían generar requerimientos que definan nuevos servicios que se requieran en el sistema (Sommerville, 2011).

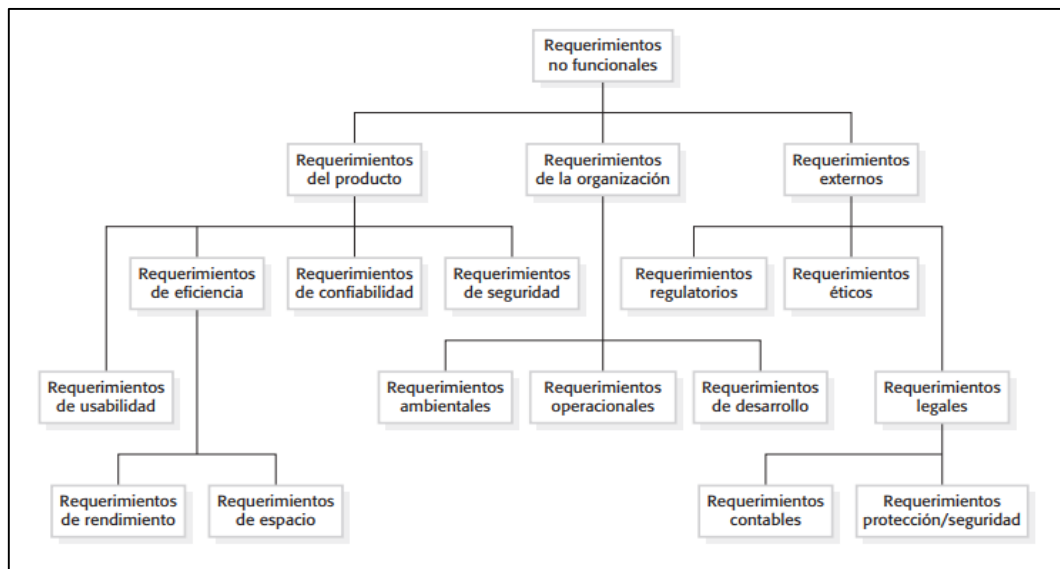


Figura 1: Tipo de requerimientos no funcionales
Fuente: (Sommerville, 2011).



1.3.3. Estapas de la Ingeniería de Requerimientos

Según (Darío & Torres, 2014), los aspectos más desafiantes que surgen en el desarrollo de los proyectos software, son las definiciones de los requisitos en el momentos de elegir las etapas de la ingeniería de requerimientos.

1.3.3.1. Elicitación de Requerimientos

Es la etapa principal en cualquier proceso de Ingeniería de Requerimientos, alcanzando todo lo relacionado con el descubrimiento de estos y el cómo el Ingeniero de Software los recolecta. Acá se plasman actividades tales como entrevistas, reuniones y talleres, análisis de documentos y prototipos, observación de actividades e historias de usuarios, entre otras, así poder obtener el mayor detalle del software que el desarrollador va realizar, la parte fundamental es la comunicación que tiene el usuario con el equipo de desarrollo (Darío & Torres, 2014).

1.3.3.2. Análisis de Requerimientos

Esta parte es primordial para el proyecto e implica un alto grado de compromiso y detalle por parte de los Analistas de Requerimientos. Los resultados de esta fase podrían definir en el éxito o fracaso del proyecto que se va a desarrollar. El Análisis de Requerimientos se encarga principalmente de detectar y resolver los conflictos entre los requerimientos obtenidos en la etapa anterior, descubriendo así los límites del software y cómo este podría interactuar con los ambientes organizacional y operacional (Darío & Torres, 2014).



En esta fase también es importante tener en cuenta la arquitectura del sistema, ya que los nuevos requerimientos deben satisfacer dicha arquitectura, y en algunas ocasiones se hace necesario modificarla para que se acople correctamente. Esta ya es la parte final de la fase, y una vez que se tiene en cuenta la priorización y la arquitectura, se procede a negociar los requerimientos con el usuario, ya que no siempre lo que este desea es posible (Darío & Torres, 2014).

1.3.3.3. Especificación de Requerimientos

La Especificación de Requerimientos fundamenta en, una vez establecidos los requerimientos que efectivamente tienen valor para el cliente y para el negocio, traducir estos a requerimientos formalmente constituido. Estos requerimientos integrarán parte de un documento que podrá ser revisado, evaluado y aprobado de manera sistemática por el desarrollador y del cliente. Dependiendo de la complejidad de cada sistema, lograrán ser generados hasta tres documentos diferentes en esta fase, documento de definición del sistema, especificación de requerimientos del sistema, y especificación de requerimientos de Software (SRS); para el caso de proyectos de software simples, sólo es necesario el SRS (Darío & Torres, 2014).

En la SRS se establecen los requerimientos negociados entre el cliente y la parte que va a desarrollar el software, de esta manera, se llega a una aprobación sobre lo que el software debe y no debe hacer. Algo importante en la construcción de este documento es el formalismo a la hora de escribir cada requerimiento, ya que entre más detallado y formal se escriba, permitirá reducir la posibilidad de errores en la especificación y con esto, tener que rediseñar el sistema en un futuro; también esto permitirá determinar con mayor precisión los costos, riesgos y el calendario (Darío & Torres, 2014).



Tabla N° 0 2: Formas de escribir una ERS

Notación	Descripción
Enunciados en lenguaje natural	Los requerimientos se escriben al usar enunciados numerados en lenguaje natural. Cada enunciado debe expresar un requerimiento.
Lenguaje natural estructurado	Los requerimientos se escriben en lenguaje natural en una forma o plantilla estándar. Cada campo ofrece información de un aspecto del requerimiento.
Lenguajes de descripción de diseño	Este enfoque usa un lenguaje como un lenguaje de programación, pero con características más abstractas para especificar los requerimientos al definir un modelo operacional del sistema. Aunque en la actualidad este enfoque se usa raras veces, aún tiene utilidad para especificaciones de interfaz.
Anotaciones gráficas	Los modelos gráficos, complementados con anotaciones de texto, sirven para definir los requerimientos funcionales del sistema; los casos de uso del UML y los diagramas de secuencia se emplean de forma común.
Especificaciones matemáticas	Dichas anotaciones se basan en conceptos matemáticos como máquinas o conjuntos de estado finito. Aunque tales especificaciones sin ambigüedades pueden reducir la imprecisión en un documento de requerimientos, la mayoría de los clientes no comprenden una especificación formal. No pueden comprobar que representa lo que quieren y por ello tienen reticencia para aceptarlo como un contrato de sistema.

Fuente: (Sommerville, 2011)

1.3.3.3.1. Procesos de especificación de requerimientos

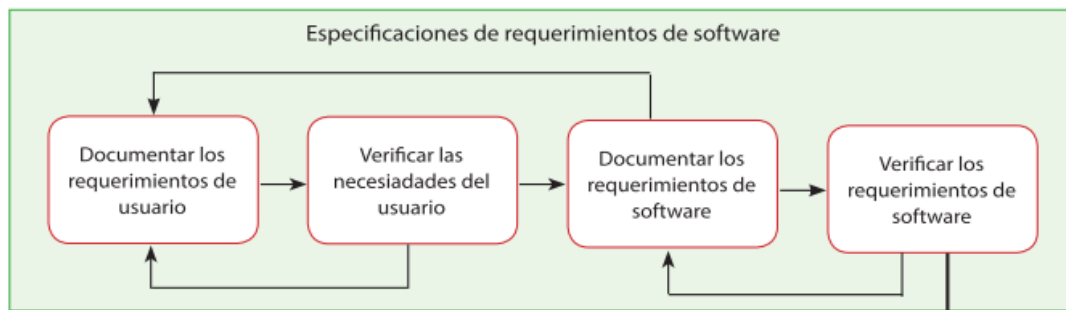


Figura 2: Proceso de especificación de requerimientos de Software

Fuente: (Sommerville, 2011)



a) Documentar los requerimientos de usuario

Según (Cueva & Sucunuta, 2017), tiene como finalidad elaborar un documento a corte de las necesidades de los usuarios; esta documentación nos ayudara a describir los comportamientos y características que el usuario propone, esta descripción tiene que ser las necesidades del usuario y la especificación de los requisitos del software

b) Verificar las necesidades del usuario

Según (Cueva & Sucunuta, 2017), se tiene que comprobar si la descripción del usuario tiene que ver con el sistema, estos requisitos provienen de los requerimientos que el negocio quiere plantear, asimismo es prudente que los Stakeholders comprueben que todos los requerimientos mencionados estén completos, consistentes y de alta calidad.

c) Documentar los requerimientos de software

Según (Cueva & Sucunuta, 2017), registrar los requerimientos del software en el documento de especificación para que el desarrollador identifique lo que va a realizar, esta especificación describe los requerimientos funcionales, los atributos de calidad, las interfaces del sistema, y las limitaciones de diseño e implementación.

d) Verificar los requerimientos de software

Según (Cueva & Sucunuta, 2017), en esta parte se verifica la descripción que sea correcta y las características del sistema, los requisitos se derivan de las necesidades del usuario, de los requisitos del sistema, y otras fuentes que se utilizaron.



1.3.3.2. Técnicas de Especificación

1.3.3.2.1. Mapa conceptual

Según (Mohamed et al., 2017) mapa conceptual produce una representación poderosa y el aprendizaje eficaz de los conocimientos de dominio. Por lo tanto, es muy útil, una técnica en las actividades de ingeniería de requisitos. En otras palabras, se puede utilizar con fuerza en las primeras etapas de la ingeniería de software, en otro aspecto, se proporcionan pequeñas contribuciones para el uso de mapas conceptuales y ontologías para apoyar la fase de ingeniería de requerimientos.

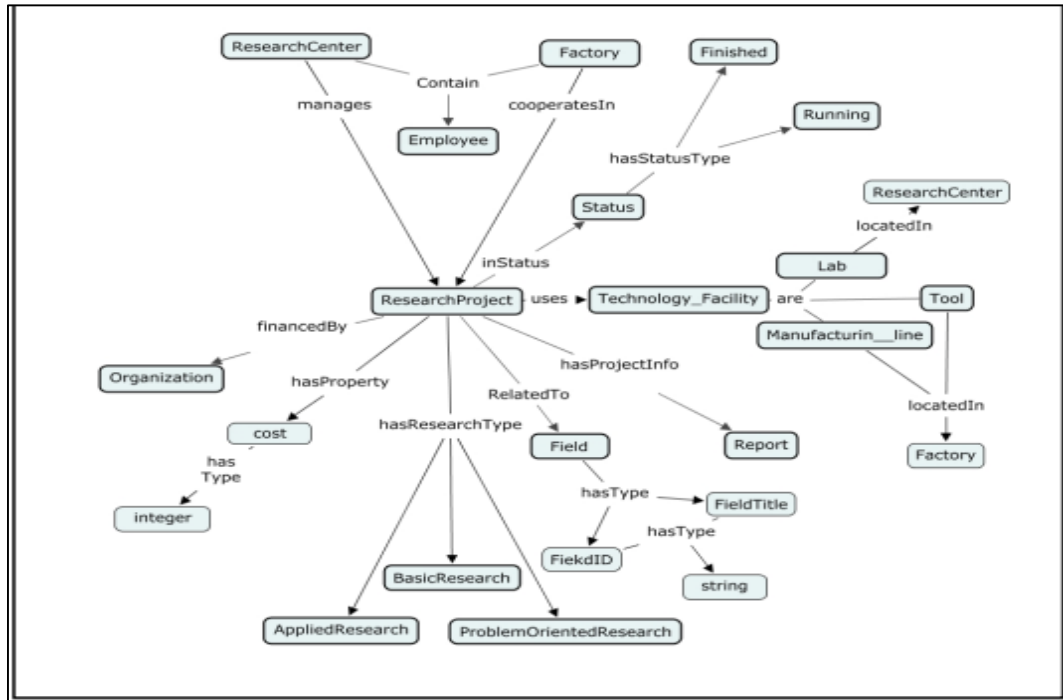


Figura 3: Mapas conceptuales producidos
Fuente: (Mohamed et al., 2017)



1.3.3.3.2. Casos de uso

Según (Sommerville, 2011) hoy en día los casos de uso se ha convertido uno de los factores primordiales del modelo lenguaje unificado, podríamos decir que, un caso de uso puedo identificar a sus actores que están implicados en una interacción, esto es complementado con la información de la interacción con el sistema, esta información puede ser representada de manera de garfios o de manera textual

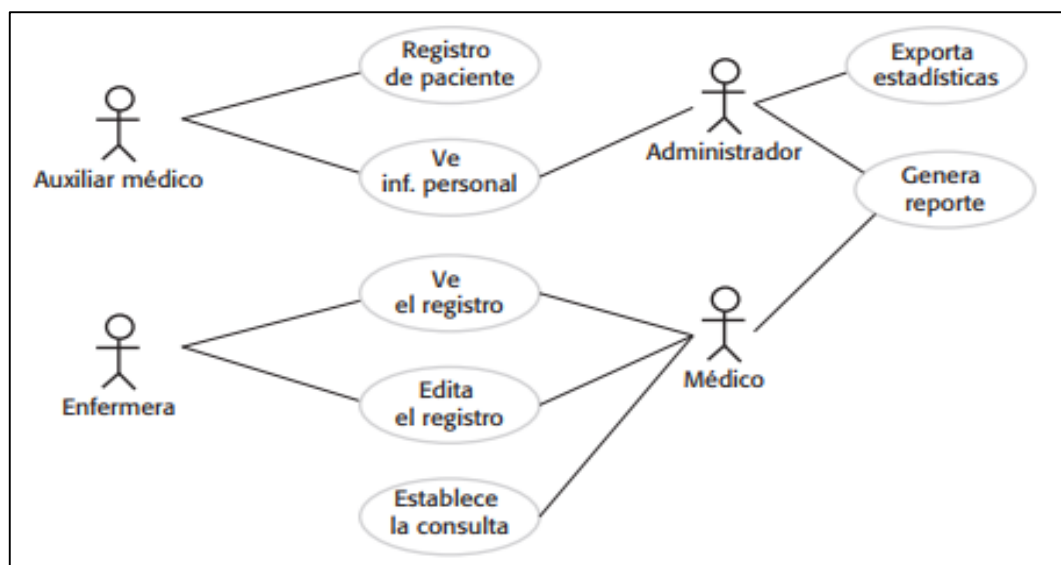


Figura 4: Diagrama de caso de uso
Fuente: (Sommerville, 2011)

1.3.3.3.3. Modelo conceptual

Según (Vidya Sagar & Abirami, 2014) los modelos conceptuales pueden desarrollarse en diagramas de clase durante las fases de diseño y ejecución del proyecto de software, incluso un modelo conceptual parcialmente automatizado puede ahorrar un tiempo significativo durante la fase de requisitos, acelerando el proceso de



comunicación gráfica y visualización. Se crea un modelo conceptual a partir de especificaciones funcionales, escritas en lenguaje natural de forma automatizada. Las clases y las relaciones se identifican automáticamente a partir de las especificaciones funcionales, esta identificación se basa en el análisis de los constructos gramaticales de oraciones y en los principios de diseño orientados a objetos.

a) Diagramas de Clase

Según (Sommerville, 2011) los diagramas de clase son utilizados cuando el usuario quiere ejecutar un sistema que se encuentra en orientado a objetos, demostrado en clases y en asociaciones, entre las clases que han sido seleccionadas. De tal manera que las clases de objetos son utilizadas como ilustraciones generales, asimismo las asociaciones son los parentescos entre las clases, eso demuestra que existe una relación entre ambas clases.

En UML se pronuncian como distintos niveles de detalles, esto quiere decir que cuando uno desarrolla un modelo, las clases seleccionadas pueden ser escritas los nombres de la clase en un recuadro, también se puede graficar las asociaciones simplemente con una línea entre las clases de las mismas (Sommerville, 2011).

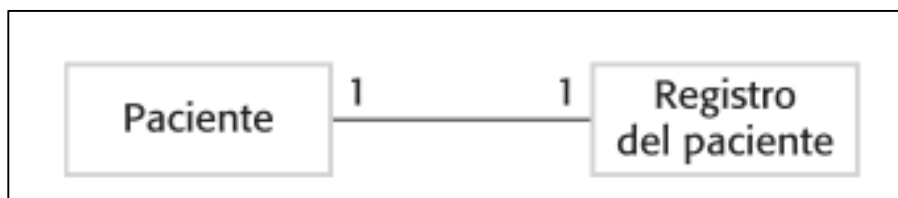


Figura 5: Clase y asociación UML
Fuente: (Sommerville, 2011)



Según (Sommerville, 2011) UML nos expresa que no es necesario una relación específica para este modelado de base de datos como se muestra en la **Figura 6**.

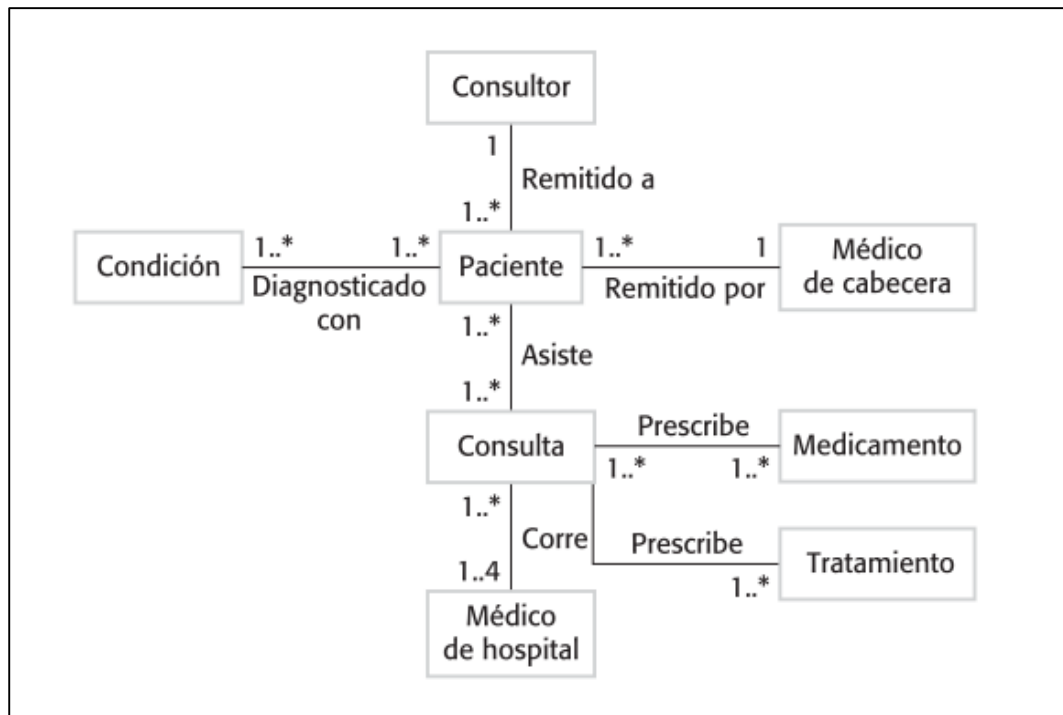


Figura 6: Diagrama de Clases UML
Fuente: (Sommerville, 2011)

Es posible que en UML se pueda utilizar modelo semántico de datos, esto se expresa, pensando en las entidades como objetos simplificadas, atributos y relaciones entre clases de objetos.



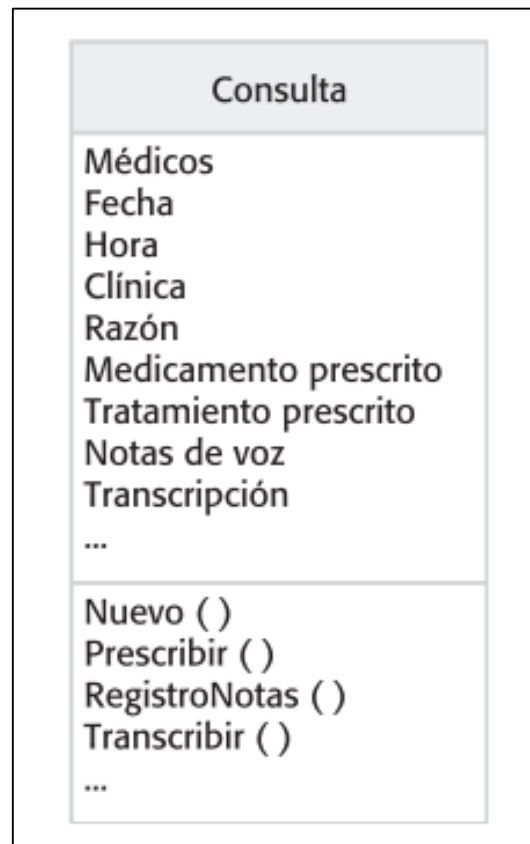


Figura 7: Clases con atributos y operaciones
Fuente: (Sommerville, 2011)

b) Generalización

Según (Sommerville, 2011) es una de las técnicas que día a día ha ido utilizándose para no contener confusión en el momento de las relaciones o asociaciones entre las clases, quiere decir que no es necesario aprender las características que detallan a cada entidad, estas entidades son clases más generales es por ellos nos permite disminuir la diferencia que existe en las clases más comunes.



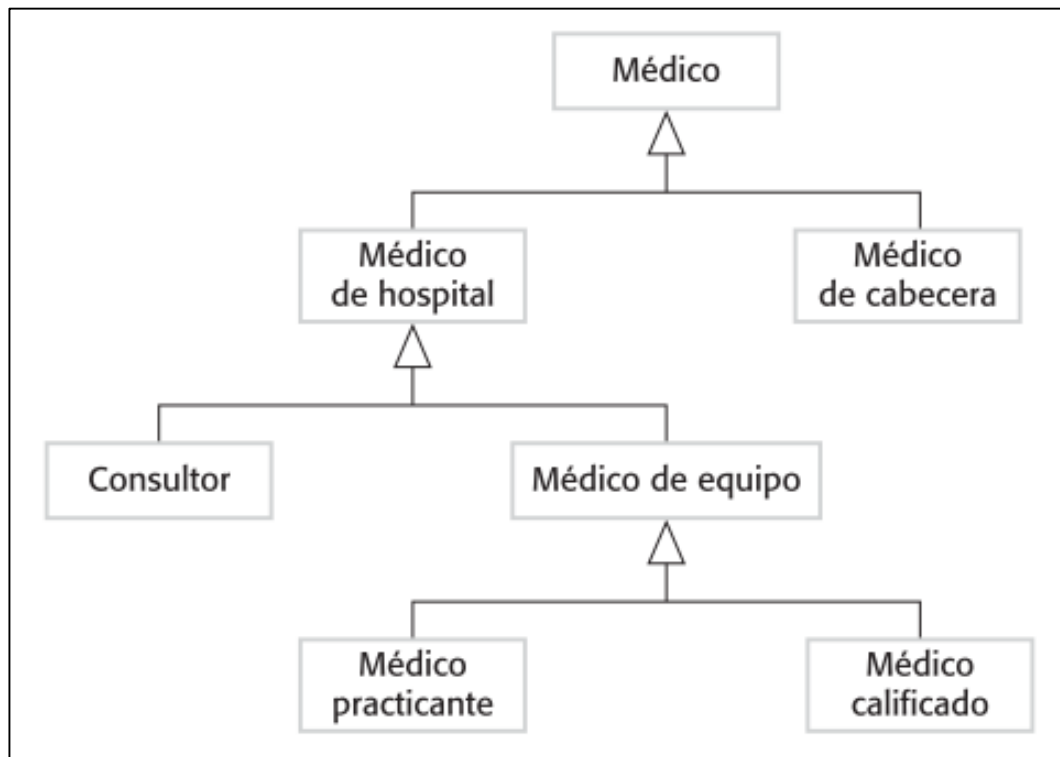


Figura 8: Jerarquía de generalización
Fuente: (Sommerville, 2011)

Asimismo, una generalización nos muestra que los atributos y operaciones asociados con las clases de nivel superior también pueden ser asociadas las del nivel inferior; es decir que las clases de nivel inferior son subclases que heredan los atributos y las operaciones de sus superclases. Por lo tanto estas clases de nivel inferior adicionan atributos y operaciones más específicos (Sommerville, 2011).



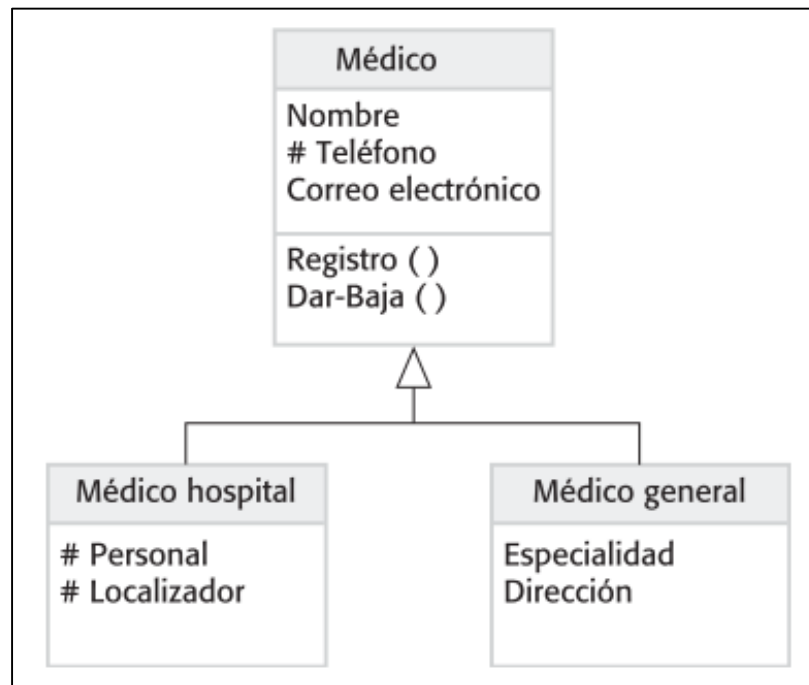


Figura 9: Jerarquía de generalización con detalles agregados
Fuente: (Sommerville, 2011)

c) Agregación

Según (Sommerville, 2011) nos facilitará en la asociación entre las clases llamado agregación, significa que un objeto se compone de otros objetos, quiere decir que el todo es agregado de las partes. Esto se expresar como se muestra en la **Figura 10** utilizando con trozo de diamante junto a la clase de objeto que es el todo.



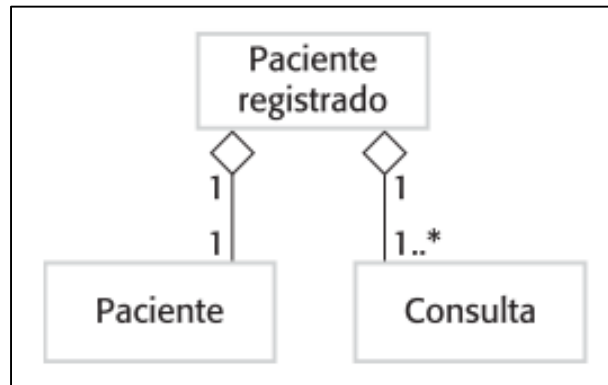


Figura 10: La asociación y agregación
Fuente: (Sommerville, 2011)

1.3.3.4. Validación de Requerimientos

Una vez terminada la especificación, viene la Validación de Requerimientos. En esta etapa básicamente se tiene que acordar entre los desarrolladores, si los requerimientos específicos realmente nos ayudara realizar una solución al problema que se enfrenta, y si satisfacen los objetivos del negocio. Para alcanzar lo anteriormente propuesto, es necesario asegurarse que los Ingenieros de Software hayan entendido correctamente los requerimientos, y esto es posible realizando un conjunto de actividades en compañía de los diferentes Stakeholders a fin de aprobar o desaprobar lo que hasta el instante se ha especificado (Sommerville, 2011).

1.3.4. Ontología Basada en Ingeniería de Requerimientos

Como mayor importancia a la ingeniería de requerimientos es lograr entender las principales necesidades del cliente y luego reformular las necesidades para proporcionar una imagen completa del proyecto de software para ayudar a las partes interesadas que comprendan lo que será el proyecto de software (Mohamed et al., 2017).



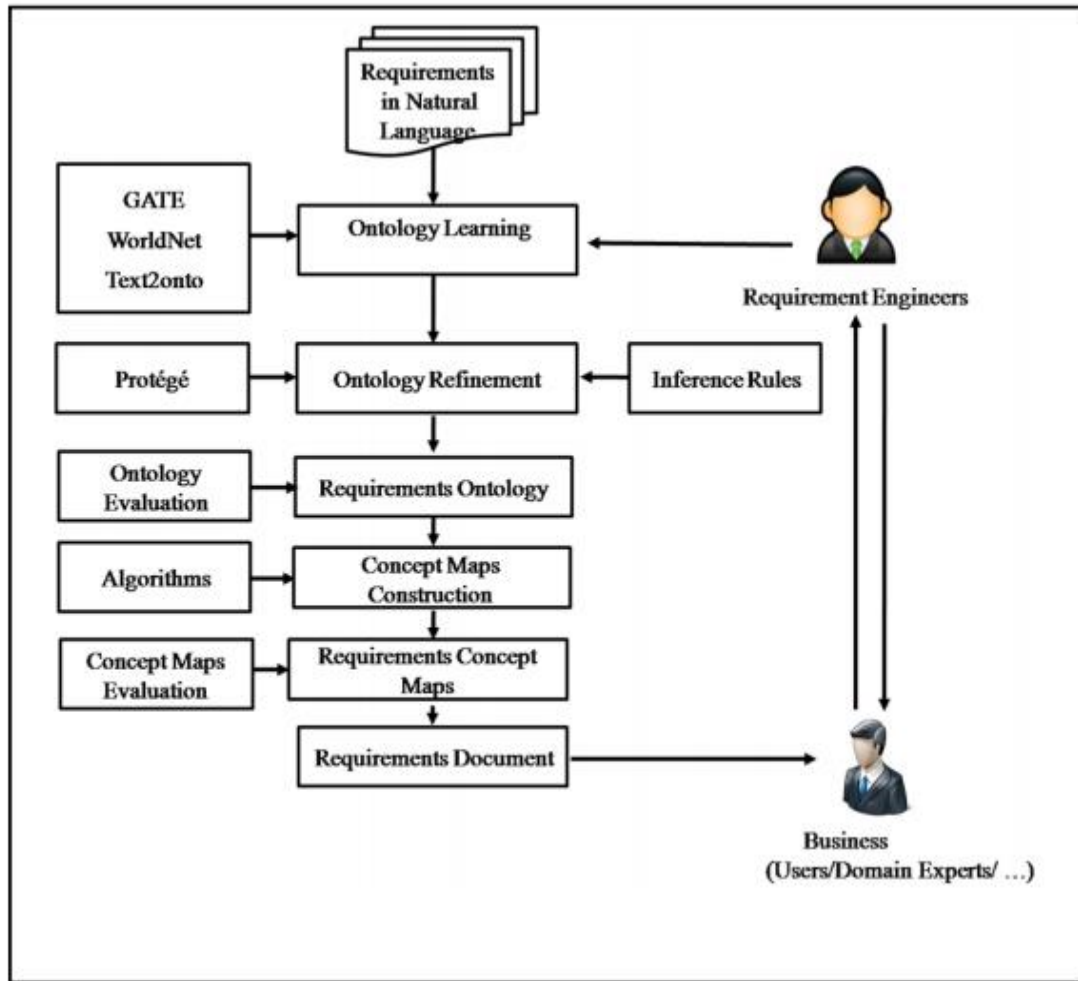


Figura 11: Arquitectura ontológica
Fuente: (Mohamed et al., 2017)

a) Lenguaje de Ontología Web (OWL)

La ontología tiene como significados definidos en lenguaje para la web semántica, desarrollado por el W3C Web Ontology Working Group. Asimismo, se encuentra diseñado para la facilidad del desarrollo Ontológico y el uso compartido de la web, con un solo objetivo de ser más sensibles los contenidos de la web. Las ontologías en



OWL proporcionan clases, propiedades, individuos y valores de datos, y son recolectados como documentos Semánticos de la Web (Ramírez, Alonso, Hernández, Arias, & Rosa, 2010).

1.3.5. Marco de Gestión Ágil Scrum

Según (Darío & Torres, 2014) nos define que Scrum es una metodología más popular hoy en día por su marco iterativo e incremental para proyectos y productos, que se desarrolla en ciclo de trabajo llamados sprint, por lo tanto scrum es un marco de trabajo que se puede emplear distintos proceso, técnicas y con distintos elementos como , Scrum Team, roles, eventos, artefactos y reglas asociadas, los cuales son mostrados como “interacción” en la Ilustración.



Figura 12: Ciclo de desarrollo ágil
Fuente: (Manuel & Cristina, 2012)



13.3.5.1 Planificación del Backlog

Según (Manuel & Cristina, 2012), es una lista donde se tiene todo el almacenamiento de las funcionalidad o requisitos en forma priorizada. Esta lista será la encargada de adquirir las interacciones creadas y gestionadas por el cliente con la ayuda del scrum master, quien será la persona encargado de indicar los tiempos para completar los requisitos, asimismo aguantará todo lo que contribuya un valor final al producto.

Las listas tendrán que contener los principales objetivos del producto para que puedan ser utilizados en las historias de usuario. En cada objetivo, se mostrará la importancia que el cliente indicara y el tiempo estimado; de este modo, se realiza la lista, prevaleciendo la importancia y tiempo, se tendrá que incluir los posibles desafíos e incluir las trabajos necesarias para poder solventarlos (Manuel & Cristina, 2012).

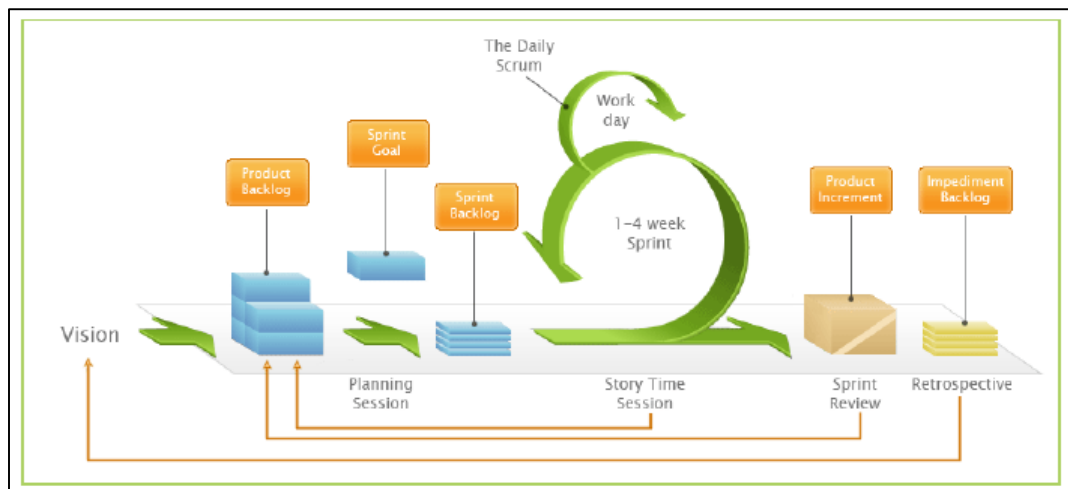


Figura 13: Marco de gestión ágil Scrum
Fuente: (Manuel & Cristina, 2012)



1.4. Formulación del Problema

¿Qué técnica puede ayudar a reducir la ambigüedad en la especificación de los requerimientos para la elaboración del product backlog bajo el marco de gestión ágil Scrum?

1.5. Justificación e importancia del estudio

Esta investigación está diseñada con la finalidad de aplicar las técnicas de especificación de ingeniería de requerimientos y así poder obtener un buen entendimiento en el momento de especificar los requerimientos del software.

Desde el punto de vista metodológica busca analizar el escenario de comparación de las técnicas de Especificación de requerimientos integrando al marco de gestión ágil Scrum (Trigas-Gallega, n.d.), esta investigación podría lograr en obtener una buena especificación de requisitos, sin ambigüedad en el desarrollo del Software, con el fin de ayudar en el índice de los proyectos que terminan con éxito y que no existan muchos fracasos ni pérdidas de costos y tiempo, cubriendo las necesidades del cliente a la hora del desarrollo del software.

En este sentido, la investigación servirá como aporte para un mejor enfoque para los analistas antes de la elaboración de los requerimientos de software para integrar al product backlog y que no haya algún fracaso y pérdida de tiempo en el software.

El desarrollo de la ingeniería de requerimiento aporta a la industria del software para construir mejores productos, sin embargo, también contribuye a mejor conocimiento actual respecto a los requerimientos y el tratamiento de ambigüedad con los estudiantes que priorizan la línea de desarrollo de software.



1.6. Hipótesis

La técnica de mapas conceptuales basado en ontología reducirá mejor la ambigüedad en la especificación de requerimientos.

1.7. Objetivos

1.7.1. Objetivo General

Comparar las técnicas de mapas conceptuales basado en ontología y modelo conceptual, en la reducción de la ambigüedad para la elaboración del product backlog del marco de gestión ágil Scrum.

1.7.2. Objetivo Especifico

- a) Seleccionar las técnicas de especificación de la ingeniería de requerimientos.
- b) Seleccionar los indicadores de rendimiento para la comparación entre técnicas.
- c) Implementar las técnicas seleccionadas en un proyecto modelo.
- d) Comparar y mostrar resultados obtenidos.

II. MATERIAL Y METODOS

2.1. Tipo y Diseño de la Investigación

2.1.1. Tipo de la Investigación

En la presente investigación corresponde a una investigación tecnológica aplicada de tipo cuantitativa, porque fundamenta en utilizar la recolección y el análisis de datos para contestar preguntas de investigación y probar la hipótesis señalada previamente. Establecido por (Hernández, Fernández, & Baptista, 2014) la investigación cuantitativa los datos son producto de medición numérica, el conteo y frecuentemente en el uso de la estadística.

2.1.2. Diseño de la Investigación

De acuerdo al tipo de investigación el diseño utilizado es de tipo cuasi experimental, como lo expresa (Hernández et al., 2014) que este tipo de diseño consiste en la observación de una variable controlando estadísticamente otras variables que se considera que no puedan afectar la variable dependiente.

2.2. Población y Muestra

2.2.1. Población

Para determinar la población se realizó un top de las 8 técnicas de Especificación de requerimientos, en base a investigación de estudios de artículos, por consiguiente, se podrá realizar la investigación comparando las 2 técnicas más usadas en el enfoque de la Especificación de IR, como se muestra en la **Tabla N° 08**.



2.2.2. Muestra

Ha sido determinada por conveniencia seleccionar las técnicas de Especificación de Requerimientos mapas conceptuales basado en ontología y modelo conceptual, teniendo en cuenta el top de técnicas en la cual son las más usadas en la investigación de proyectos ya concluidos.

2.3. Variables, Operacionalización

2.3.1. Variable Independiente

Mapas conceptuales basado en ontología.

Modelo conceptual.

2.3.2. Variable dependiente

Ambigüedad de requerimientos.

2.3.3. Operacionalización

Tabla N° 0 3: Indicadores y fórmulas para detallar las variables, técnicas e instrumentos de recolección de datos.

<i>Variable independiente</i>	<i>Indicadores</i>	<i>Formula</i>	<i>Descripción</i>
<i>Mapas conceptuales basados en ontología</i>	<i>Tiempo</i>	$TR = TF - TI$	<i>TR = Tiempo de repuesta.</i> <i>TF = Tiempo Final.</i> <i>TI = Tiempo Inicial.</i>
<i>Modelo conceptual</i>			
<i>Variable dependiente</i>	<i>Indicadores</i>	<i>Formula</i>	<i>Descripción</i>
	<i>Consistencia</i>	$\text{Consistencia} = \frac{ \{x \mid x \in \text{RCC} \wedge \neg \text{contradict}(x)\} }{ \text{RCC} }$	<i>ReqItem= Conjunto de elementos de requisitos.</i> <i>Con= Conjunto de conceptos.</i>
<i>Ambigüedad de requerimientos</i>	<i>No Ambiguo</i>	$\text{No Ambigüedad} = \frac{ \{x \mid x \in \text{ReqItem} \wedge F_{\text{int}}(x) \subseteq \text{Clo}\} }{ \text{ReqItem} }$	<i>Rel= Conjunto de relaciones.</i> <i>Clo= Significa un cierre de Con U Rel</i>

Fuente: Elaboración Propia.



2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad

2.4.1. Abordaje metodológico

El abordaje metodológico fue del subtipo cuasi – experimental, como lo expresa (Hernández et al., 2014) que este tipo de diseño utiliza la variable independiente para observar su efecto sobre una o más variables dependientes, la investigación estableció el efecto de las técnicas de especificación de requerimientos, mapas conceptos basado en ontología y modelo conceptual (variable independiente), sobre la Ambigüedad de requerimientos (variable dependiente).

2.4.2. Método y técnicas de recolección de datos

Se obtuvo por elegir la técnica de observación científica debido de lo que observamos en la actualidad, con el propósito de obtener los datos previamente que han sido definidos como interés para la investigación.

Se optó por aplicar como técnica el análisis documental debido al interés que tiene la orientación científica e informativa el cual es un proceso transversal utilizado en la investigación como son los artículos, las tesis y libros.

2.4.2.1. Instrumentos de recolección de datos

Ficha de Observación

El formato de ficha de observación es el documento más utilizado en la actualidad, con el propósito de querer evaluar y conocer un tema específico que se quiere plantear ya sea en grupo o en sociedad, es una herramienta fundamental de la investigación cuantitativa **Anexo N° 03**.

Formato de acta de entrevista

Este formato es uno de los documentos de prioridad en la investigación el cual a través de este queda específicamente plasmado que se cumplió con la entrevista y que en realidad se aplicó la veracidad de la recolección de los datos de dicha investigación.

2.5. Procedimiento de análisis de datos

- a) Seleccionar las técnicas de especificación de la ingeniería de requerimientos de acuerdo a los indicadores que se desean obtener.
- b) Elaborar un formato de ficha de observación.
- c) Elaborar un formato de acta de reunión para la entrevista.
- d) Implementar las técnicas seleccionadas en un proyecto modelo.
- e) Mostrar resultados obtenidos.

2.6. Criterios éticos

En la investigación se manipuló el principio ético de honestidad o integridad, ya que no se manipuló la información, exponiendo los resultados tal cual se obtuvieron. Además, se citó las referencias bibliográficas apropiadamente para no incurrir en plagio.

2.7. Criterios de rigor científico

La presente investigación se realiza siguiendo los juicios científicos señalados, estos permiten garantizar la calidad de la investigación. Así, seguimos la coherencia metodológica durante el desarrollo de la investigación, según el muestreo de datos, las cuales teniendo como características la fiabilidad cumpliendo con las expectativas de la investigación, validez los datos obtenidos fueron evaluados y analizados, consistencia presentó material consistente y certificada.



III. RESULTADOS

En esta sección se presentará el análisis ejecutado de la comparación de las técnicas previamente estudiadas y seleccionadas, obtenidas de la selección: mapas conceptuales basado en ontología y modelo conceptual, para reducir la ambigüedad de los requerimientos planteados en un caso de estudio, posteriormente poder evaluarlos mediante el estándar de la (IEEE-STD-830, 1998), y su estudio de factibilidad basado en el tiempo de respuesta.

Según (IEEE-STD-830, 1998) para poder medir los rendimientos de los requerimientos de software para reducción de la ambigüedad se tomaron las siguientes métricas: Consistencia y no ambiguo, los cual se detallaron las fórmulas matemáticas correspondientes a cada métrica.(Kaiya & Saeki, 2005).

3.1. Resultado en tablas y figuras

3.1.1. Estudio de Factibilidad

- **Tiempo de respuesta**

Se derivó a valorar las técnicas mapas conceptuales basado en ontología y modelo conceptual, bajo el tiempo de respuesta que tomó la reducción de la ambigüedad de los requerimientos, expresados en minutos teniendo en cuenta el desarrollo que se propuso en esta investigación.

A continuación, se mostrará la técnica monedo conceptual, la cantidad de requerimientos, el tiempo de respuesta, el total de los minutos, en fechas variadas como se representará en el diagrama de Gantt de las actividades que se desarrollaran



y también se calculan en minutos como se muestra en la **Tabla N° 04**; por siguiente se mostrará de la misma manera la técnica mapas conceptuales basado en ontología como se expresa en la **Tabla N° 05**.

Tabla N° 0 4: Tiempo de respuesta de la Técnica Modelo Conceptual

Técnica	N° RQ.	TI	TF	Total	N° RQ.	TI	TF	Total
		minutos	/minutos			minutos	/minutos	
		08-10-2018				09-10-2018		
Modelo Conceptual	RF -001	10:00 a.m	10:08 a.m	8	RF -017	1:00 p.m	1:04 p.m	4
	RF -002	10:09 a.m	10:14 a.m	5	RF -018	1:05 p.m	1:13 p.m	8
	RF -003	10:15 a.m	10:18 a.m	3	RF -019	1:14 p.m	1:19 p.m	5
	RF -004	10:19 a.m	10:25 a.m	6	RF -020	1:20 p.m	1:24 p.m	4
	RF -005	10:26 a.m	10:34 a.m	8	RF -021	1:25 p.m	1:30 p.m	5
	RF -006	10:35 a.m	10:42 a.m	6	RF -022	1:31 p.m	1:42 p.m	11
	RF -007	10:43 a.m	10:48 a.m	5	RF -023	1:43 p.m	1:48 p.m	5
	RF -008	10:49 a.m	10:57 a.m	8	RF -024	1:49 p.m	2:03 p.m	14
	RF -009	10:58 a.m	11:09 a.m	11	RF -025	2:04 p.m	2:08 p.m	4
	RF -010	11:10 a.m	11:15 a.m	5	RF -026	2:09 p.m	2:18 p.m	9
	RF -011	11:16 a.m	11:24 a.m	8	RF -027	2:19 p.m	2:30 p.m	11
	RF -012	11:25 a.m	11:29 a.m	4	RF -028	2:31 p.m	2:36 p.m	5
	RF -013	11:30 a.m	11:36 a.m	6	RF -029	2:37 p.m	2:46 p.m	9
	RF -014	11:37 a.m	11:40 a.m	3	RF -030	2:47 p.m	2:53 p.m	6
	RF -015	11:41 a.m	11:46 a.m	5	RF -031	2:54 p.m	2:59 p.m	5
	RF -016	11:47 a.m	11:55 a.m	8	RF -032	3:00 p.m	3:04 p.m	4
TIEMPO EN TOTAL EN MINUTOS								208

Fuente: Elaboración Propia.



Tabla N° 0 5: Tiempo de respuesta de la Técnica mapas conceptuales basado en ontología

Técnica	N° RQ.	TI	TF	Total	N° RQ.	TI	TF	Total
		minutos	minutos			minutos	minutos	
10-10-2018				11-10-2018				
MAPAS CONCEPTUALES BASADO EN ONTOLOGÍA	RF -001	08:30 a.m	8:35 a.m	5	RF -017	09:00 a.m	09:03 p.m	3
	RF -002	08:36 a.m	8:39 a.m	3	RF -018	09:04 a.m	09:07 p.m	3
	RF -003	08:40 a.m	8:43 a.m	3	RF -019	09:08 a.m	09:10 p.m	2
	RF -004	08:44 a.m	8:47 a.m	3	RF -020	09:11 a.m	09:14 p.m	3
	RF -005	08:48 a.m	8:53 a.m	5	RF -021	09:15 a.m	09:18 p.m	3
	RF -006	08:54 a.m	8:57 a.m	3	RF -022	09:20 a.m	09:32 p.m	12
	RF -007	09:58 a.m	09:03 a.m	5	RF -023	09:33 a.m	09:35 p.m	2
	RF -008	09:04 a.m	09:09 a.m	5	RF -024	09:36 a.m	09:48 p.m	12
	RF -009	09:10 a.m	09:22 a.m	12	RF -025	09:49 a.m	09:52 p.m	3
	RF -010	09:23 a.m	09:28 a.m	5	RF -026	09:53 a.m	10:03 p.m	10
	RF -011	09:29 a.m	09:34 a.m	5	RF -027	10:04 a.m	10:09 p.m	5
	RF -012	09:35 a.m	09:38 a.m	3	RF -028	10:10 a.m	10:13 p.m	3
	RF -013	09:39 a.m	09:42 a.m	3	RF -029	10:14 a.m	10:24 p.m	10
	RF -014	09:43 a.m	09:45 a.m	2	RF -030	10:25 a.m	10:28 p.m	3
	RF -015	09:46 a.m	09:51 a.m	5	RF -031	10:29 a.m	10:32 p.m	3
	RF -016	09:52 a.m	09:57 a.m	5	RF -032	10:33 a.m	10:35 p.m	2
TIEMPO EN TOTAL EN MINUTOS								151

Fuente: Elaboración Propia.



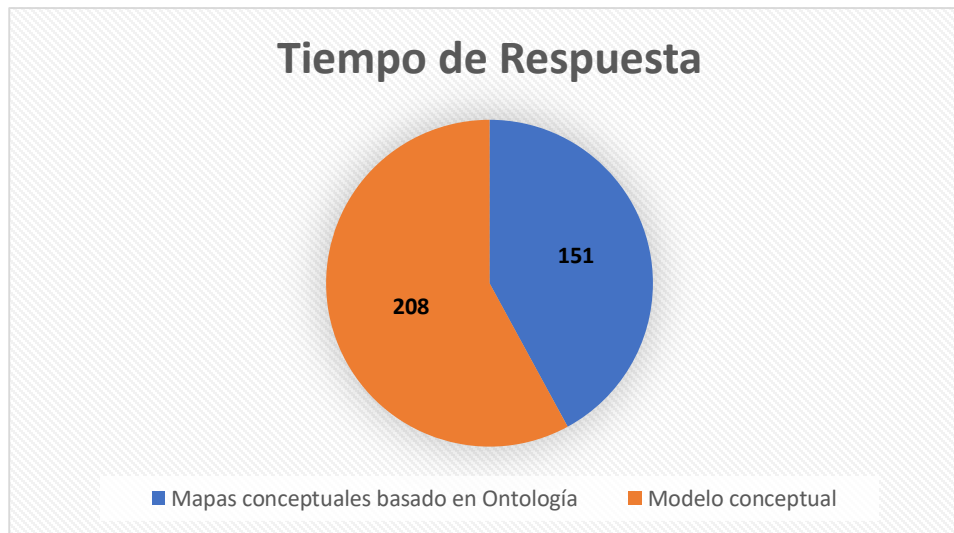


Figura 14: Tiempo de respuesta

Fuente: Elaboración Propia

Por consiguiente, los datos obtenidos de la guía de observación desarrollada en la empresa CESLYSOFT & BUSSINES S.R.L que se muestran en la **Figura 14**, los tiempos de respuesta en minutos de las técnicas de Mapas conceptuales basados en ontología y Modelo conceptual tienen una diferencia de 47", donde la primera técnica mencionada, nos ayudara para la reducción de los requerimientos. Por lo consecuente requiere de menos tiempo.



3.1.2. Consistencia de Requerimiento

Relaciones entre conceptos mencionados en el documento de requerimientos, esta métrica se enfoca en los requerimientos mencionados. Tales relaciones pueden ser representadas en forma de cálculo como en la **Tabla N° 06**.

Tabla N° 06: Cálculo de consistencia de requerimiento

TÉCNICA	CONTRADICCIÓN	RCC	RESULTADO
Mapas conceptuales basados en ontología	46	47	97.87 %
Modelo conceptual	29	44	65.91 %

Fuente: Elaboración Propia.

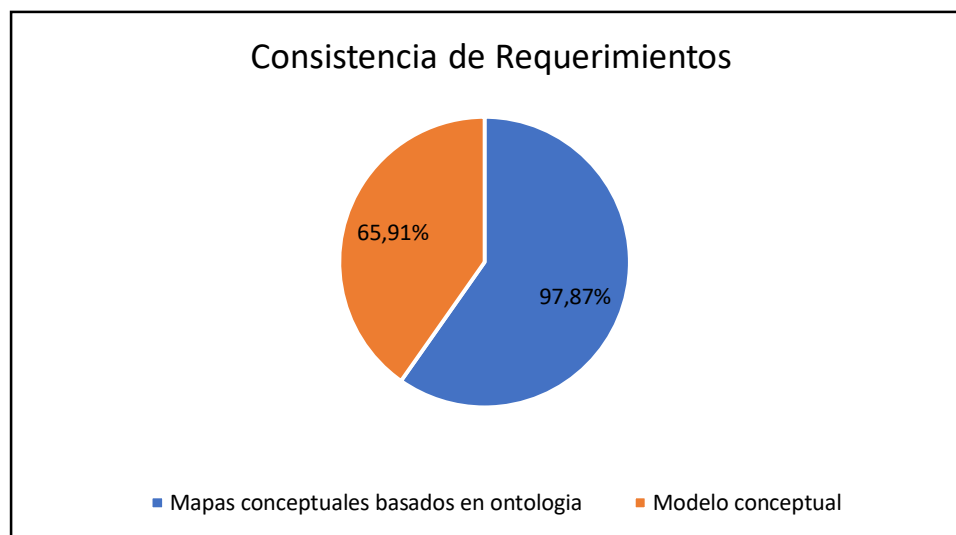


Figura 15: Requerimientos de consistencia
Fuente: Elaboración Propia



3.1.3. No Ambiguo de Requerimiento

Cuando un elemento de requisitos se asigna a varios elementos que no están relacionados semánticamente, el ítem es considerado como uno ambiguo, expresando la contabilidad como se muestra en la **Tabla N° 07**.

Tabla N° 07: Cálculo de No ambiguo de requerimiento

TÉCNICA	REQ.	CONCEPTOS	RESULTADO
Mapas conceptuales basados en ontología	30	32	93.75 %
Modelo conceptual	26	29	89.66 %

Fuente: Elaboración Propia.

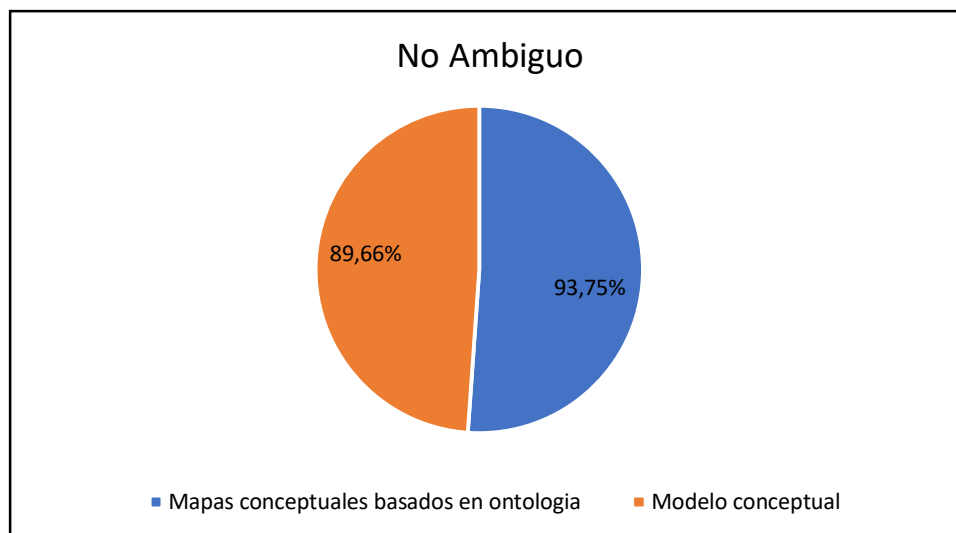


Figura 16: Requerimiento de no ambiguo
Fuente: Elaboración Propia



3.2. Discusión de resultados

Esta investigación tuvo como idea principal en disminuir la ambigüedad de los requerimientos de software, de aquellos proyectos no concluidos en la industria del desarrollo. Según (IEEE-STD-830, 1998), tiene como métricas fundamentales para la consistencia y no ambiguo de requerimientos, las cuales serán evaluadas y ejecutadas para llegar a un resultado con éxito en la ER.

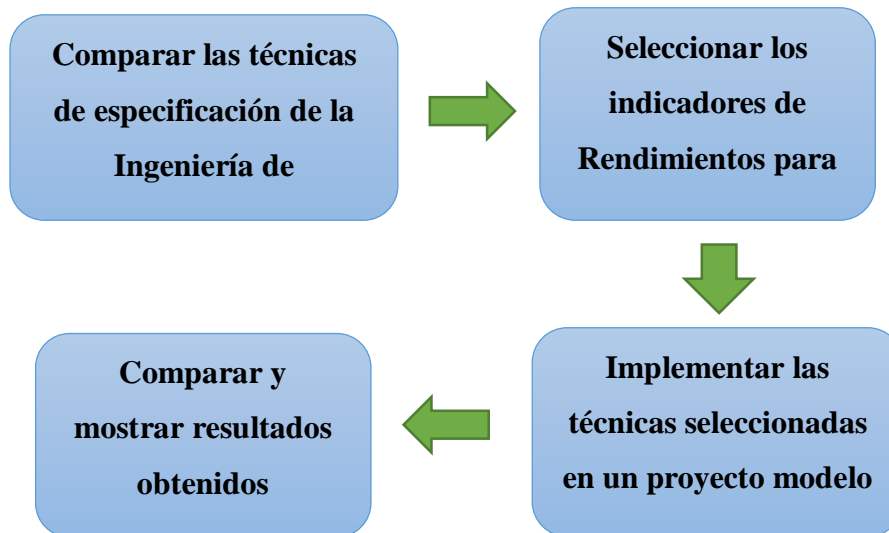
En la **Figura 15**, utilizando la métrica de Consistencia de Requerimiento, muestra un porcentaje de éxito con las técnicas propuestas, es decir tomando los requerimientos obtenidos del caso de uso de la empresa CESLYSOFT & BUSSINES S.R.L., estas técnicas fueron seleccionadas por un cuadro comparativo (TOP), siendo elegidas por criterio profesional, evaluadas empresarialmente en la línea de Ingeniería de Software; La técnica mapas conceptuales basado en ontología obtuvo un porcentaje de 97.87% y la técnica modelo conceptual con 65.91%, quiere decir que la técnica con mayor porcentaje nos servirá para mayor criterio de que los requerimientos obtenidos nos puede ayudar a que no haya contradictorio en los requerimientos. Por otro lado, en la **Figura 16**, se utilizó la métrica de No Ambiguo de Requerimiento, lo cual la técnica mapas conceptuales basado en ontología obtuvo un porcentaje mayor de 93.75 % a 89.65% de la técnica ya mencionada, siendo la mejor en porcentaje para reducir la ambigüedad, por lo tanto, podemos decir que la técnica en la cual se va utilizar en esta propuesta es la que mayor nos ayudara a disminuir la ambigüedad de los requerimientos para la ejecución del software con éxito.

Es preciso recalcar que si bien es cierto la propuesta planteada no significa que todos los analistas pueden utilizar esta técnica siendo el factor principal la caída del software a la hora de la ejecución. Por otro lado, se debe mencionar que se evidencia un promedio de efectividad superior a la otra técnica seleccionada.



3.3. Aporte Practico

Grafico 1. Diagrama de flujo que muestra cómo se desarrollaran los procesos de los objetivos planteados.



Fuente: Elaboración Propia

3.3.1. Comparar las técnicas de especificación de la ingeniería de requerimientos

En la ingeniería de requerimiento existe 4 etapas: Elicitación, Análisis, Especificación y Validación (Sommerville, 2011), en cada uno de estas etapas se puede utilizar distintas técnicas, y para esta investigación es necesario seleccionar 2 técnicas con las que se va a trabajar el caso de negocio que se logrará comparar dichas técnicas, para ello se ha considerado una evaluación de estas técnicas basado en las estepas de la ingeniera de requerimientos, el criterio para seleccionar las técnicas es el nivel de uso en las empresas, el nivel de uso de estas técnicas fue estudiado por (Besrou, Rahim,



& Dominic, 2016) (Hurtado, n.d.,2013), (A. Toro,2016), (G & Losavio, 2013), que para este trabajo se considerara un criterio de importante de selección que representa a una evaluación empresarial de uso como se muestra en la **Tabla N° 08**.

Tabla N° 0 8: Cuadro comparativo basado en el uso de las técnicas en la IR

No	Técnica de Requerimientos	Etapas				Nivel (1-5)	Fuente
		Elici.	Análi.	Espf.	Valid.		
1	Entrevista	X				4.39	
2	Observación	X	X			1.89	
3	Lluvia de ideas	X	X			4.02	
4	Lenguaje Natural Estructurado		X	X		4.08	(Besrou
5	JAD (Diseño de aplicación Conjunta	X	X			4.37	et al.,
6	Revisión por pares				X	4.17	2016)
7	Orientada a Objetivos		X			4.15	(Hurtado,
8	Etnografía	X				2.18	n.d., 2013)
9	Modelo Conceptual		X	X		4.09	(A. Toro,
10	Especificación de requisitos blandos			X		4.05	2016)
11	Check-List (lista de verificación)	X	X	X	X	4.11	(G &
12	Especificación basada en ERD					4.23	Losavio,
13	Mapas Conceptuales Basados en Ontología	X	X	X		4.60	2013)
14	Caso de Uso	X	X	X	X	4.17	
15	Laddering(Escalera)	X		X		2.04	
16	Misuse-Case (Caso de Uso Incorrecto)		X	X		4.06	

Fuente: Elaboración Propia



Identificando aquellas técnicas de mayor uso se seleccionó las técnicas mapas conceptual basado en ontología y modelo conceptual, ya que por criterio de puntuación de estudio realizados.

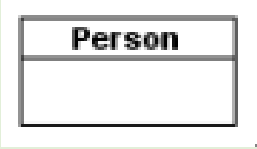

Es importante identificar las características más importantes de las técnicas seleccionadas que servirá de punto de partida para el uso de las mismas.

3.3.1.1. Características de ontología y Lenguaje de Modelado Unificado (UML)

Según (Ramírez et al., 2010) los lenguajes fueron creados bajo una perspectiva desiguales, donde la ontología se sostiene bajo el modelo conceptual, posteriormente la cual sea que se represente el conocimiento, por lo tanto ayudará de gran manera el desarrollo el software, teniendo en cuenta que ambos lenguajes emplean a los objetos y relaciones entre ellos mismos. Llegando a trabajar de una manera muy similar alineados a un fin común que se expresa con el conocimiento a la vez sea ilustrado e indicado, en cuanto al lenguaje modelado unificado tiene como prioridad modelar las limitaciones que a su vez deben compensar un conjunto de estados que será permitidos por el sistema. Por lo tanto, la representación del conocimiento se adhiere a conseguir un nuevo conocimiento que será representado por una ontología. A continuación, se muestra una asimilación (no completa, pero si específicas de las características comunes de ambos lenguajes).



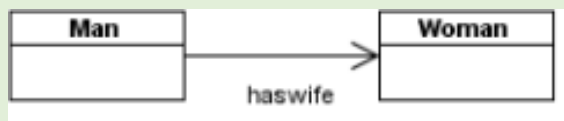
Tabla N° 0 9: Comparación de características OWL y UML

Elementos UML	Elementos OWL
Clases	Clases
<p>Ambos lenguajes tienen como arquitectura principal a un conjunto de clases, donde en el lenguaje de la ontología viene hacer un conjunto de ceros o más instancias. Mientras tanto en el lenguaje modelado unificado se basa en una simiente general</p>	
 <p>A UML class diagram for 'Person' with an empty class box.</p>	<p><i>Clase (: Persona)</i></p>
instancia	individual
<p>En algunos de los casos se puede decir que la construcción individual del lenguaje de ontología es similar a la instancia del lenguaje modelo unificado; por lo tanto, el lenguaje de ontología llena ayudará de diferentes formas dando definición de una clase.</p>	
 <p>A UML instance diagram for 'Person' with the attribute 'name : Mary'.</p>	<p><i>Individual (: Persona: Mary)</i> <i>Individual (: Rol :Persona)</i></p>
asociación binaria, propiedad atributo	Propiedad



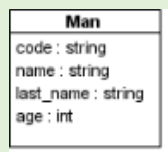
En el Lenguaje de ontología las relaciones entre clases son definidas como propiedades y constituidas por propiedad de objeto y propiedad de datos

La clase hombre tiene una relación con la clase mujer que significa que tiene esposa, que es representada en el lenguaje de modelo unificado como la asociación que interpreta que tiene esposa y en lenguaje de ontología como una propiedad de objeto del mismo nombre.



Propiedad Objeto (: tiene esposa dominio (: hombre) rango (: mujer))

La propiedad de datos se expresa de la asociación definida como propiedad atributo entre las clases y la propiedad, quiere decir que cuyo dominio es una clase y el rango es el tipo de la propiedad. Como se observa en el ejemplo del lenguaje de ontología para la propiedad atributo de la clase hombre : nombre.




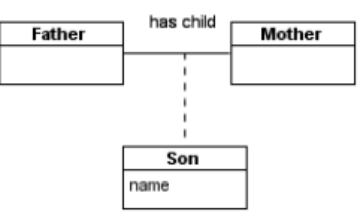
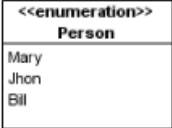
Propiedad tipo de datos (: nombre de dominio (: nombre: Persona) rango (: nombre xsd: cadena))

generalización

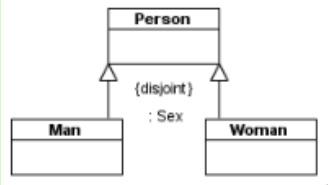
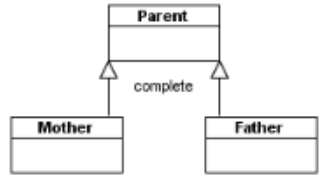
subclase subpropiedad

Ambas técnicas llevan la relación de subclase. Por lo tanto, el lenguaje de ontología solo se representa en sub clases y lenguaje de modelado unificado se presenta en generalización para identificar la inclusión de una clase.



	<p><i>Subclase (: mujer: persona)</i></p>
<p>Asociación N-aria, clase de asociación</p>	<p>Clase, propiedad</p>
<p>En el lenguaje de modelado unificado la asociación de una clase, también puede relacionarse con varias clases puede crearse como un array de clases. Para el lenguaje de odontología no existe una especificación de construcción, es por ello q se define como pares de conocimientos. Se presentó como solución un diseño de patrones llamado cosificación que consiste en crear una nueva clase, asimismo se crean nuevas relaciones funcionales una por cada entidad que participe en la asociación</p>	
	<p><i>Clase (: Hijo)</i> <i>Propiedad objeto (: padre_hijo)</i> <i>dominio (: hijo)</i> <i>rango (: padre)) Propiedad objeto (:</i> <i>dominio madre_hijo (: hijo)</i> <i>rango (: madre))</i></p>
<p>enumeración</p>	<p>uno de</p>
<p>Cada una de ellas son definidas con la misma extensión para una clase, sin embargo, en el lenguaje de modelado unificado se define que un enumerador es estimado como un tipo de datos antes que una clase</p>	
	<p><i>Clase (: Persona</i> <i>Uno de(:Mary :Jhon :Bill))</i></p>



disjunto, cubierta	disjunto con, unión de
<p>Estos lenguajes son capaces de permitir que sus subclases de las clases se declaren disjuntas. En el lenguaje de ontología normalmente se utiliza la edificación disjunta con. Además, se puede mencionar que para el array de subclases es declarado que el cubrimiento es completo, asimismo se dice que la instancia de la superclase es una de la instancia por lo menos de una de la subclases.</p>	
	<p><i>desunirse con (: mujer: hombre)</i></p>
	<p><i>Clase (: padre unión de (: madre: padre))</i></p>
multiplicity	min cardinalidad, max cardinalidad, cardinalidad
<p>En el lenguaje de ontología la restricción aplicada a una clase se puede imponer ante cardinalidad como se puede expresar ya sea cardinalidad mínima, máxima o exacta de la instancia que participen de una relación. En el lenguaje modelado unificado las asociaciones pueden ser expresadas en cardinalidad mínimas y máximas</p>	



	<p><i>Cardinalidad máxima (N: tiene hijo: padre)</i></p> <p><i>min Cardinalidad (N: tiene hijo: padre)</i></p> <p><i>cardinalidad (N: tiene hijo: padre)</i></p>
<p>También se expresa que en el lenguaje de ontología una propiedad puede ser considerada como propiedad funcional o inversa funcional, por lo tanto la propiedad funcional tiene como característica principal una cardinalidad máxima 1 sobre el rango, mientras que la propiedad inversa tiene como característica principal una cardinalidad máxima 1 sobre el dominio.</p>	
	<p><i>Propiedad Funcional (: tiene esposa)</i></p> <p><i>Propiedad Funcional Inversa (: tiene esposa)</i></p>
<p>Si en el caso que en el lenguaje modelado unificado tenga una multiplicidad sobre los dos extremos, entonces eso corresponde al lenguaje de ontología que será un par inversa</p>	
	<p><i>inverso de (: tiene padre: tiene hijo)</i></p> <p><i>Cardinalidad máxima (N: tiene hijo: padre)</i></p> <p><i>Cardinalidad máxima (1: tiene padre: hijo)</i></p>
<p>paquete</p>	<p>ontología</p>



Estos lenguajes son capaces de soportar el concepto de espacio de nombres como construcción de empaquetadura, se define como un modelo de una subclase del paquete tanto como en lenguaje modelado unificado como en lenguaje de ontología.

Fuente: (Ramírez et al., 2010)

Por lo tanto, es necesario como regla de medición el factor de rendimiento más adecuado, para poder realizar la comparación de las técnicas seleccionadas.

3.3.2. Seleccionar los indicadores de rendimientos para la comparación

Para la selección de los indicadores de rendimiento se tomó como base la Norma IEEE 830 “Especificación de Requerimientos de Software (ERS) (IEEE-STD-830, 1998), en la cual manifiesta los requerimientos a una elevada interpretación suficiente como para permitir que los diseñadores puedan efectuar y diseñar un sistema que satisfaga estos requisitos, por lo tanto, que permite si en realidad las pruebas que se hace satisfaga o no a los requerimientos planteados.

Según (Kaiya & Saeki, 2005) Al utilizar las 2 técnicas mapas conceptual basado en ontología y modelo conceptual, podemos calcular métricas para las características de los requerimientos. En la norma IEEE 830 hay ocho características y las cuatro características que se obtuvieron están relacionadas con la semántica de una especificación de requisitos.

Por lo siguiente se detalla: "ReqItem" significa el conjunto de elementos de requisitos (declaraciones) en un documento de requisitos, "Con" significa el conjunto de conceptos en una ontología, "Rel" significa el conjunto de relaciones en la ontología,



y "Clo" significa un cierre de $Con \cup Rel \setminus \{contradecir, antónimo\}$.

$$\text{Consistencia} = \frac{|\{x \mid x \in RCC \wedge \neg \text{contradict}(x)\}|}{|RCC|}$$

$$RCC = \{r \mid \exists a \exists b : Con \cdot \exists x \exists y : ReqItem \cdot a \in F_{int}(x) \wedge b \in F_{int}(y) \wedge r(a, b)\}$$

$$\text{No Ambigüedad} = \frac{|\{x \mid x \in ReqItem \wedge F_{int}(x) \subseteq Clo\}|}{|ReqItem|}$$

Teniendo los indicadores de rendimiento definidos. Se deberá implementar las técnicas señaladas en el Caso de Negocio CESLYSOFT & BUSSINES S.R.L.

3.3.3. Implementar las técnicas seleccionadas en un proyecto modelo

Se implantará las técnicas seleccionadas estudiado en el primer objetivo de esta investigación las cuales fueron: mapas conceptual basado en ontología y modelo conceptual, determinando que su nivel de uso fue superior de un rango del 1-5, en la cual fue estudiado por los investigadores (Besrou, Rahim, et al., 2016) (Hurtado, n.d.,2013), (A. Toro,2016), (G & Losavio, 2013).

3.3.3.1. Ontología

Según (Mohamed et al., 2017), teniendo en cuenta los componentes, lenguaje y herramientas que serán utilizadas en la investigación, se estará utilizando la herramienta Protegé, que ayudara a crear una ontología mediante instancias, relaciones y conceptos como se muestra en la **Tabla N° 10**.



a) Herramienta de modelado de ontologías: Protégé

Según (Marcelo et al., n.d.) expresa que para poder manejar el modelamiento de ontología, se tiene que efectuar una herramientas la cual implementa el lenguaje de ontología, se tiene que precisar que fue creada por la Universidad de Stanford para el desarrollo de Ontologías y Sistemas basados en el conocimiento, interactuando por una interface de usuario que logre al cliente un manejo flexible para creación en estructuras de marco que contienen clases, slots e instancias de una forma alineada .

Tabla N° 10: Componentes y herramienta para una Ontología

Ontologías	OWL	Protégé
Instancias	Individuos	Casos (Instance)
Relaciones	Propiedades	Slots
Conceptos	Clases	Clases

Fuentes: Elaboración Propia

Por los consiguiente se utilizó las reglas de inferencia para el mapeo de entidades o clases la cual se puede definir mediante los sinónimos o sustantivos (Tesauros)(Besrou, Rahim, et al., 2016), encontrados en el Caso de Negocio CESLYSOFT & BUSSINES S.R.L

3.3.3.1.1. Mapeo y Reglas de Inferencia

Según (Besrou, Rahim, et al., 2016), el mapeo de los requisitos se puede escribir formalmente a continuación:



$F_{int}: ReqItem \rightarrow 2^{Con \cup Rel}$ - "ReqItem" es un conjunto de elementos de requisitos (declaraciones) en un documento de requisitos, y llamamos a una función F_{int} como "una función de interpretación".

Un predicado InSpec se puede escribir de la siguiente manera utilizando F_{int} .

$InSpec(x, S) \equiv \exists r \in S \cdot (x \in F_{int}(r))$ - Tenga en cuenta que r muestra los elementos de requisitos en un documento S .

Al utilizar la función de interpretación y las reglas de inferencia, podemos analizar un documento de requisitos de forma semántica. En este estudio, usamos la lógica de predicado de primer orden para la inferencia.

Las fórmulas lógicas que incluyen reglas de inferencia provienen básicamente de dos tipos de recursos, uno es un tesoro en un sistema de ontologías y otro se describe explícitamente regla. Los conceptos en un tesoro se usan como valores constantes en las reglas, y las relaciones se usan como predicados.

De acuerdo con el significado de cada relación y concepto, podemos definir las reglas de inferencia por adelantado. Por ejemplo, una relación "generalizar" se utiliza para mostrar que las características son los tics de una súper clase se heredan de sus subclases. Este hecho se puede escribir en el siguiente esquema lógico.

Las características inherentes de cada relación también son representativas. Resentido como reglas de inferencia. Por ejemplo, como una relación "generalizar" es reflexiva y transitiva, usamos las siguientes reglas de inferencia.



$\forall x \cdot$ generalizar (x, x).

generalizar (x, y) \wedge generalizar (y, z) \rightarrow generalizar (x, z).

Por otro lado, como una relación "antónimo" es simétrica, usamos la siguiente regla.
antónimo (x, y) \rightarrow antónimo (y, x).

3.3.3.2. Modelo Conceptual

El modelo conceptual básico se crea como resultado de los primeros cuatro módulos funcionales, es decir, el preprocesamiento de texto, la extracción de características sintácticas, la extracción de elementos de diseño y la clasificación de tipo de relación. Aunque el objetivo de nuestra investigación es crear modelos a partir de los requisitos del lenguaje natural, asumimos que las oraciones en el texto de entrada satisfacen algunos requisitos básicos (Vidya Sagar & Abirami, 2014).

a) Lenguaje de Modelado Unificado (UML)

Según (Vidya Sagar & Abirami, 2014) expresa que la base primordial para la creación de un artefacto de software, es preciso observar, especificar, construir y evidenciar la información, donde el rol lo interpreta de una manera modelada visual la técnica del lenguaje de modelado unificado(UML), posteriormente capta la búsqueda de una estructura estática y a la vez el comportamiento dinámico del sistema, no obstante, necesita de una semántica rigurosa, que permita utilizar razonadores automáticos sobre los modelos.

b) Rational Rose Enterprise software

IBM Rational Rose Enterprise proporciona soporte de modelado para la aplicación desarrollo y trabajos con una serie de tecnologías de implementación. Todos los productos Rational Rose incluye soporte UML completo; sin embargo, Varían en términos de la implementación. tecnologías que soportan (Corporation, 2006).

Tabla N° 11: Componentes y herramienta para una UML

Modelo Conceptual	UML	Rational Rose
Casos	Casos	Casos
Asociaciones	Relaciones	Relaciones
Conceptos	Clases	Clases

Fuentes: Elaboración Propia

Por lo tanto, se utilizó las reglas de modelo conceptual por la cual se puede definir mediante los sinónimos, verbos, sustantivos dando como resultados a posibles candidatos a entidades (Vidya Sagar & Abirami, 2014), encontrados en el Caso de Negocio CESLYSOFT & BUSSINES S.R.L

3.3.3.2.1. Reglas de modelado conceptual

Según (Vidya Sagar & Abirami, 2014), expresa que las reglas de modelo conceptual ayudara a capturar las semánticas de los requisitos y los posibles candidatos a entidades, como se aprecia en la **Tabla N° 12**.



Tabla N° 12: Categorías de reglas de diseño y declaraciones

CATEGORÍA	N°	DECLARACIÓN DE REGLA DE DISEÑO
Transformación de Sujeto a objeto	RSO-01	Cuando una relación gramatical de modificador de posesión ocurre en una oración, el sujeto se cambia para referirse al dueño de la posesión.
	RSO-02	sustantivo modificadores compuestos se combinan con el sustantivo para generar palabras compuestas.
	RSO-03	Un adjetivo que califica un sustantivo, donde el adjetivo no se puede clasificar se combina con el sustantivo sujeto a generar palabras compuestas.
Clase	RC-01	Cualquier nombre que aparezca como sujeto es siempre una clase.
	RC-02	Los sustantivos que aparecen como objetos participan en las relaciones, pero no se crean como clases explícitamente.
	RC-03	Los sustantivos siempre se convierten a su forma singular.
	RC-04	Los gerundios se crean como clases.
Atributos	RA-01	Un adjetivo clasificable, ya sea en el predicado o en forma atributiva significa un atributo.
	RA-02	Un verbo intransitivo con un adverbio puede significar un atributo.
	RA-03	Una frase nominal que sigue a la frase "identificado por", "Reconocido por" indica la presencia de un atributo.



	RA-04	apóstrofe posesivo significa un atributo
	RA-05	El uso de tener en oraciones denota atributos.
	RR-01	Un verbo transitivo es candidato para un tipo de relación.
Relación	RR-02	Un verbo con una preposición es candidato para una preposición.
	RR-03	Una oración de la forma “la r de a es b”.
	RR-04	La oración de la forma “a es la r de b”.
	RO-01	Un verbo intransitivo es una operación.
Operaciones	RO-02	Un verbo que relaciona una entidad con una clase candidata que no sea creada que no sea entidad.

Fuentes: Elaboración Propia

Según (Kaiya & Saeki, 2005) y (Vidya Sagar & Abirami, 2014), las reglas y herramientas obtenidos nos ayudara al desarrollo de las técnicas ya seleccionadas en la investigación para Caso de Negocio CESLYSOFT & BUSSINES S.R.L

3.3.3.3. Caso de estudio: Caso de Negocio CESLYSOFT & BUSSINES S.R.L

La empresa CESLYSOFT & BUSSINES S.R.L. requiere una solución de software a medida para controlar la cartera de clientes y la gestión de proyectos de software. Por lo consiguiente el sistema debe permitir consultar y actualizar la información de clientes de forma eficiente y confiable, cada cliente debe estar asociado a uno o más servicios de software; de los clientes se debe almacenar información que permita la consulta remota para brindar soporte de acuerdo a los requerimientos del cliente, ya sea para actualizar o modificar funcionalidades del sistema, asimismo capacitar al



usuario; una de las características de la empresa es ofrecer soporte personalizado que permita dar un servicio de calidad.

Además, el sistema debe permitir gestionar los proyectos basados en requerimientos, para lo cual se verificará el estado, tipo y categoría, en seguida el trabajador asignará un orden y tiempo de atención de acuerdo a la prioridad de la petición, ya que el cliente podrá registrar su requerimiento en tiempo real utilizando la aplicación web; el requerimiento también podrá contener documentos o imágenes de apoyo para su mayor comprensión de parte de los trabajadores, cada trabajador será asignado a una tarea, será controlado por hora de inicio y hora fin de ejecución de dicho requerimiento.

Revisado el caso de negocio se procedió a la elaboración del Project charter ver (Anexo 01). Luego de dar inicio formalmente al proyecto se desarrolló la especificación de requerimientos que permite ampliar y formalizar los requerimientos del negocio ver (Anexo 02). Finalmente, en la fase de inicio del proyecto se extrajo los requerimientos iniciales del caso del negocio para la definición de los requerimientos como se muestra en la **Tabla N° 13**.

Tabla N° 13: Requerimientos del Cliente (Iniciales)

CODIGO	PRIORIDAD (A, M, B)	DESCRIPCIÓN BREVE
RI -001	A	Controlar la cartera de clientes de la empresa.
RI -002	A	Controlar detalladamente los proyectos que tiene cada cliente con la empresa.



RI -003	A	Gestionar los requerimientos registrados de los clientes.
RI -004	M	Listado de los requerimientos según la prioridad de cliente.
RI -005	A	Realizar el ingreso de requerimientos según el cliente lo establezca para el software.
RI -006	A	Controlar las operaciones (crear, modificar, eliminar) de los requerimientos de clientes.
RI -007	M	Realizar el mantenimiento del tipo de requerimiento de los clientes, que deberá permitir (crear, modificar, eliminar).
RI -008	M	Controlar todos los requerimientos que genera un cliente, así mismo clasificar en base al tipo de requerimientos y asignar la prioridad y estado.
RI -009	A	Realizar el ingreso, modificación y eliminación de documentos asignados a los proyectos de software.
RI -010	A	Crear Tareas de acuerdo al requerimiento solicitado por el cliente, así mismo dicha tarea deberá poderse asignar a un trabajador.
RI -011	A	Controlar las tareas asignadas a un trabajador, basándose en medir su rendimiento en horas.

Fuente: Elaboración Propia



Los requerimientos iniciales se definen por su identificación de código, por su nivel de prioridad (A-M-B) y una breve descripción que permita tener una visión general del requerimiento desde la perspectiva del cliente, esto se hace con el fin de que se pueda comprender tanto el equipo de desarrollo como el cliente.

Por consiguiente, es identificar los procesos negocios que están asociados al caso de negocio ver **Tabla N° 14**.

Tabla N° 14: Requerimientos Funcionales por Procesos

CODIGO	REQUERIMINETOS
RF -001	Registrar Proyectos
RF -002	Registrar Clientes
RF -003	Registrar Trabajador
RF -004	Listar características técnicas adicionales de los Clientes
RF -005	Registrar características técnicas adicionales de los Clientes
RF -006	Realizar mantenimiento de proyecto según la prioridad de cliente
RF -007	Realizar listado de cartera de clientes
RF -008	Registrar los requerimientos de cliente
RF -009	Modificar los requerimientos de cliente
RF -010	Eliminar los requerimientos de cliente
RF -011	Estado de requerimientos de cliente
RF -012	Listar los requerimientos priorizados del cliente
RF -013	Registrar el tipo de requerimiento de los clientes
RF -014	Modificar el tipo de requerimiento de los clientes
RF -015	Eliminar el tipo de requerimiento de los clientes
RE -016	Crear tipo de requerimiento de los clientes
RE-017	Modificar mantenimiento de tipo de requerimiento de los clientes



RE-018	Eliminar mantenimiento de tipo de requerimiento de los clientes
RE-019	Clasificar en base al tipo de requerimientos de los clientes
RE-020	Asignar la prioridad a los requerimientos del cliente
RE-021	Realizar el ingreso de documentos asignados a un proyecto
RE-022	Eliminar documentos asignados a un proyecto
RE-023	Registrar una tarea y asignarla a un trabajador
RE-024	Modificar una tarea asignada a un trabajador
RE-025	Eliminar una tarea asignada a un trabajador
RE-026	Registrar las horas que se trabajaron por tarea
RE-027	Modificar las horas que se trabajaron por tarea
RE-028	Eliminar las horas que se trabajaron por tarea
RE-029	Visualizar progreso del trabajador por tarea asignada

Fuente: Elaboración Propia

Los requerimientos funcionales por procesos se definen por su identificación de código y una breve descripción que permita tener una visión general del requerimiento desde la perspectiva de los requerimientos iniciales del cliente, esto se hace con el fin de que pueda comprender el equipo de desarrollo.

Por lo tanto, efectuaremos la metodología en cada una de las técnicas seleccionadas.

3.3.3.3.1. Modelado conceptual basado en UML

En la fase de pre procesamiento, el texto se divide en frases. El resultado de la extracción de características sintácticas proporciona la lista de adjetivos, adverbios, verbos, sustantivos propios y comunes. La lista final que clasifica las palabras en partes del discurso en este texto como se aprecia en la **Tabla N° 15**.



Tabla N° 15: Ejemplo de implementación – relaciones después de la extracción de entidades.

Característica	Extracción
Adjetivos	cartera, proyecto, información, cliente, trabajador, soporte, orden, tiempo, tarea, estado, cliente, categoría, prioridad.
Sustantivos	consultar, actualizar, servicio, asociado, cliente, proyecto, soporte, requerimiento, empresa, gestionar, tipo, categoría, trabajador, prioridad, documento, imágenes, tarea, sistema, usuario, software, área
Nombres Comunes	clientes, mantenimiento proyecto, sistema, usuario, requerimiento, documento, tarea, archivo, software, área, cliente, trabajador, Tipo_documento, Tipo_requerimiento.

Fuente: *Elaboración Propia*

Por lo consiguiente se aplica la extracción de entidades y atributos. **La Tabla N° 16** contiene las entidades extraídas de los requisitos de la **Tabla N° 13**. En la columna 2 representa las entidades, mientras en la tercera columna contiene la frecuencia de cada entidad que aparece al momento de la extracción de las características como se menciona en la **Tabla N° 15**.



Tabla N° 16: Entidades extraídas del caso de estudio de negocio.

N°	Entidades Encontradas	Frecuencia (TDS)
1	Cliente	2
2	Trabajador	3
3	Área	2
4	Proyecto	2
5	Documento	2
6	Empresa	2
7	Requerimiento	2
8	Tarea_requerimiento	1
9	Prioridad	2
10	Categoría	2
11	Estado	1
12	Tipo_requerimiento	1
13	Archivo	1
14	Tipo_documento	1
15	Tarea	3
16	Sistema	1
17	Usuario	1

Fuente: Elaboración Propia

De tal manera que, siguiendo las reglas de relaciones de modelo conceptual se podrá obtener una comparación de clase 1 y clase 2 como se muestra en la **Tabla N° 17**, así como se podrá describir las reglas más usadas con el código ya asignado en la **Tabla N° 12**.

Tabla N° 17: Ejemplo de implementación – relaciones después de la extracción de entidades.

N° Entidades	Clase 1	Calse2 (podría no existir)	Nombre de relación	Regla usada
1	Cliente	Empresa	Incluye	RC-01
2	Cliente	Cartera	Permite	RSO-03
3	Empresa	Cliente	Incluye	RC-01
4	Proyecto	Software	Servicio	RSO-03
5	Software	Proyecto	Servicio	ESO-01
6	Trabajador	Operaciones	Incluye	RO-02
7	Operaciones	Trabajador	soporte	RA-01
8	Trabajador	Área	Incluye	RC-01
9	Área	operaciones	Servicio	RO-02
10	Documento	Cliente	Incluye	RC-01
11	Cliente	Documento	Servicio	RC-01
12	Requerimientos	Cliente	Petición	RC-01
13	Trabajador	Requerimiento	Servicio	RC-01
14	Requerimiento	Software	Servicio	RSO-03
15	Tarea	trabajador	Soporte	RC-01
16	Tarea	Requerimiento	Soporte	RC-01
17	Trabajador	Tarea	Incluye	RC-01
18	Tarea_requerimiento	Trabajador	Incluye	RC-01
19	Prioridad	Cliente	Incluye	RC-01
20	Prioridad	Requerimiento	Tiempo	RA-01
21	Categoría	Requerimiento	Incluye	RA-01
22	Estado_requerimiento	trabajador	Incluye	RO-02



23	Tipo_requerimiento	Trabajador	incluye	RO-02
24	Cliente	Tipo_requerimiento	Incluye	RO-02
25	Archivo	Cliente	Soporte	RA-01
26	Tipo_documento	Requerimiento	Incluye	RO-02

Fuente: Elaboración Propia

La **Tabla N° 18** contiene los atributos de las respectivas entidades extraídas aplicando las reglas de extracción.



Tabla N° 18: Entidades y atributos.

ENTIDADES	ATRIBUTOS
Cliente	IdCliente, nombre, apellido, fecha_maci, sexo, dirección, teléfono, DNI, teamviewer, anydesk.
Trabajador	Id Trabajador, nombre, apellido, fecha_maci, sexo, dirección, teléfono, DNI.
Requerimiento	
Empresa	IdEmpresa, razón_social, ruc, dirección.
Proyecto	IdProyecto, nombre, descripción, fecha_inicio, fecha_fin.
Área	IdArea, descripción, estado.
Prioridad	IdPrioridad, Descripción estado.
Documento	IdDocumento, nombre, descripción.
Estado_requerimiento	IdEstado_requerimiento, descripción, estado.
Tarea	IdTarea, descripción, horas, estado.
Archivo	IdArchivo, nombre, descripción.

Fuente: Elaboración Propia

Después tener definidas las entidades, atributos y sus posibles relaciones en la **Figura 17** se mostrará un diagrama de clases a partir de las Especificación de casos de uso (Vidya Sagar & Abirami, 2014).



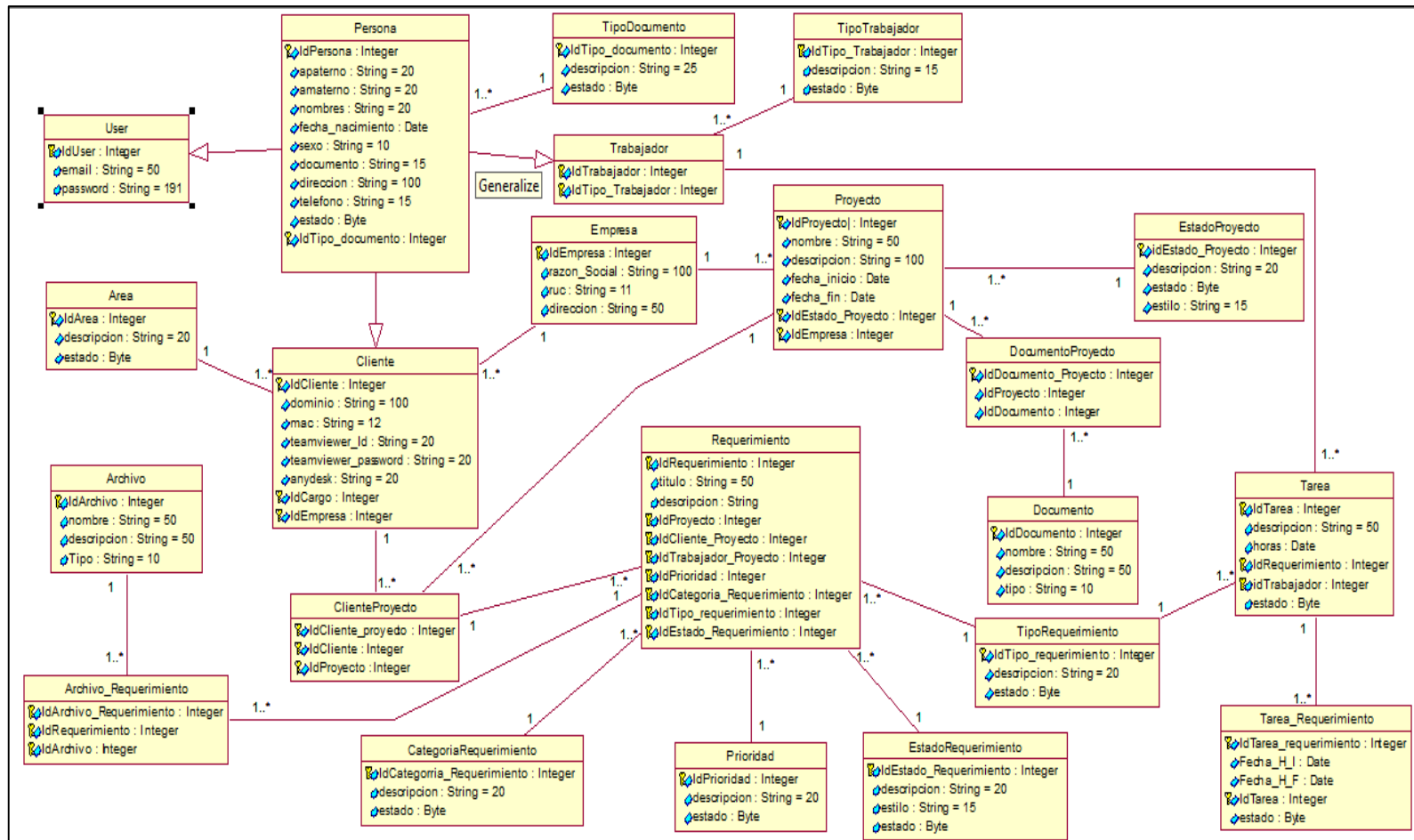


Figura 17: Diagrama de clases a partir del modelo conceptual

Fuente: Elaboración Propia



3.3.3.3.2. Mapa conceptual basado en ontología

Según (Ellatif, 2017), se define como una especificación explícita formal de una conceptualización compartida en su naturaleza depende de una lógica y facilidad para su formalización, llegando a eliminar la ambigüedad de un dominio específico y proporciona una forma de operación computacionales de tal manera que ayuda a la comprensión compartida del dominio claro.

3.3.3.3.2.1. Construcción de una ontología

Según (Ellatif, 2017) contiene 4 elementos básicos conceptos, relacione, instancias. Un concepto es un significado o una descripción de esa palabra o grupo de palabras que representa una clase o entidad de un dominio. Una instancia es un modelo de una clase o entidad. Las relaciones son la asociación entre conceptos. Los axiomas presentan el verdadero conocimiento requerido por razonar en un dominio para derivar hechos nuevos o faltantes. Reglas Indicar a las piezas de conocimiento lógico que solían formalizar las funciones y restricciones de un dominio.

La ontología es la herramienta más adecuada para la captura y representación de conocimiento formal que puede tomar de un texto. Según (Ellatif, 2017) La ontología se puede construir de las siguientes maneras :

- Manualmente, mediante el uso de editores de ontologías como Protégé.
- Automático, mediante el procesamiento de lenguaje natural con herramientas de aprendizaje ontológico con respeto al objetivo de reduciendo la interacción humana en esta etapa tal text2onto.
- Semiautomático, mediante el uso de algunas herramientas de aprendizaje ontológico reduce la participación humana en la construcción. Funcionamiento de la ontología.



En esta investigación utilizaremos la herramienta manualmente, mediante editores de ontología como Protege, partiendo desde una definición de requerimientos iniciales que se describe en la **Tabla N° 19**

Tabla N° 19: Definición de Requerimientos iniciales

ITEMS	REQUERIMIENTOS
RQ-01	Registrar clientes de las empresas.
RQ-02	Registrar los proyectos de los clientes.
RQ-03	Gestionar los requerimientos del cliente
RQ-04	Priorizar los requerimientos por cliente.
RQ-05	Realizar el requerimiento por cliente.
RQ-06	Registrar los mantenimientos de los clientes (crear, eliminar, modificar).
RQ-07	Realizar mantenimiento de tipo de requerimiento.
RQ-08	Registrar todos los requerimientos que genera un cliente, así mismo clasificar en base al tipo de requerimientos y asignar la prioridad y estado.
RQ-09	Realizar el ingreso, modificación y eliminación de documentos asignados a los proyectos de software.
RQ-10	Crear Tareas de acuerdo al requerimiento solicitado por el cliente, así mismo dicha tarea deberá poderse asignar a un trabajador.
RQ-11	Controlar las tareas asignadas a un trabajador, basándose en medir su rendimiento en horas.

Fuente: Elaboración Propia

Por lo tanto, se detallará la creación de la ontología basado al diagrama de clases el cual contiene entidades, relaciones o asociaciones, la cual nos ayudó a expresar los requerimientos iniciales como se muestra en la **Tabla N° 19**.



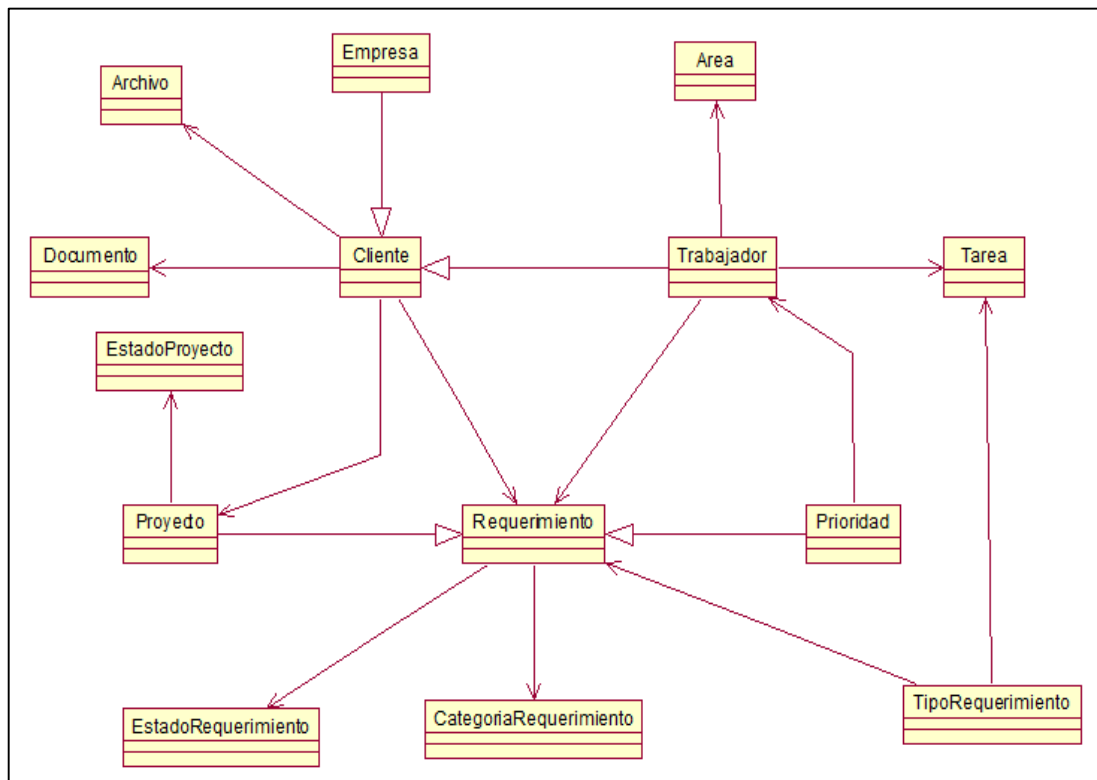


Figura 18: Ontología generada a partir de los requerimientos iniciales
Fuente: (Manuel & Cristina, 2012)

Teniendo los requerimientos iniciales se necesita detectar si un documento de requerimientos es consistente o completo utilizando un sistema de ontología, cada elemento de requerimiento se asigna en un conjunto de elementos (conceptos y relaciones). Para detectar inconsistencia en un documento de requisitos tratamos de encontrar elementos q se contradigan por ejemplo decimos que el documento es inconsistente si hay una relación contradictoria entre dos conceptos; para detectar incompletitud verificamos que la relaciones específicas de un concepto ya está mapeado por ejemplo un requerimiento vacío (Kaiya & Saeki, 2005).

Basándose en los requerimientos iniciales se procede a elaborar los requerimientos específicos como se muestra en la **Tabla N° 20**



Tabla N° 20: Requerimientos Funcionales por Procesos

CODIGO	REQUERIMINETOS
RF -001	Registrar Clientes
RF -002	Modificar Clientes
RF -003	Eliminar Clientes
RF -004	Eliminar Proyecto
RF -005	Registrar Proyecto
RF -006	Eliminar Proyecto
RF -007	Listar características técnicas adicionales de los Clientes
RF -008	Registrar características técnicas adicionales de los Clientes
RF -009	Realizar mantenimiento de proyecto según la prioridad de cliente
RF -010	Realizar listado de cartera de clientes
RF -011	Registrar los requerimientos de cliente
RF -012	Modificar los requerimientos de cliente
RF -013	Eliminar los requerimientos de cliente
RF -014	Estado de requerimientos de cliente
RF -015	Listar los requerimientos priorizados del cliente
RF -016	Registrar el tipo de requerimiento de los clientes
RF -017	Modificar el tipo de requerimiento de los clientes
RF -018	Eliminar el tipo de requerimiento de los clientes
RE -019	Crear tipo de requerimiento de los clientes
RE-020	Modificar mantenimiento de tipo de requerimiento de los clientes
RE-021	Eliminar mantenimiento de tipo de requerimiento de los clientes
RE-022	Clasificar en base al tipo de requerimientos de los clientes
RE-023	Asignar la prioridad a los requerimientos del cliente
RE-024	Realizar el ingreso de documentos asignados a un proyecto
RE-025	Eliminar documentos asignados a un proyecto
RE-026	Registrar una tarea y asignarla a un trabajador
RE-027	Modificar una tarea asignada a un trabajador



- RE-028** Eliminar una tarea asignada a un trabajador
- RE-029** Registrar las horas que se trabajaron por tarea
- RE-030** Modificar las horas que se trabajaron por tarea
- RE-031** Eliminar las horas que se trabajaron por tarea
- RE-032** Visualizar progreso del trabajador por tarea asignada

Fuente: Elaboración Propia

A partir de los requerimientos definidos efectuaremos el modelamiento de dominio ontológico, comenzaremos con el arranque de la herramienta **Protege versión 5.5.0**, como se muestra en la **Figura 19**. Por consiguiente, se abrirá la interface de la herramienta como se muestra en la **Figura 21**, así mismo se extrajo las entidades de los requerimientos de la **Tabla N° 20** como se grafica en la **Figura 23** y por tanto haciendo grafico ontológico como puede apreciar en la **Figura 24**.

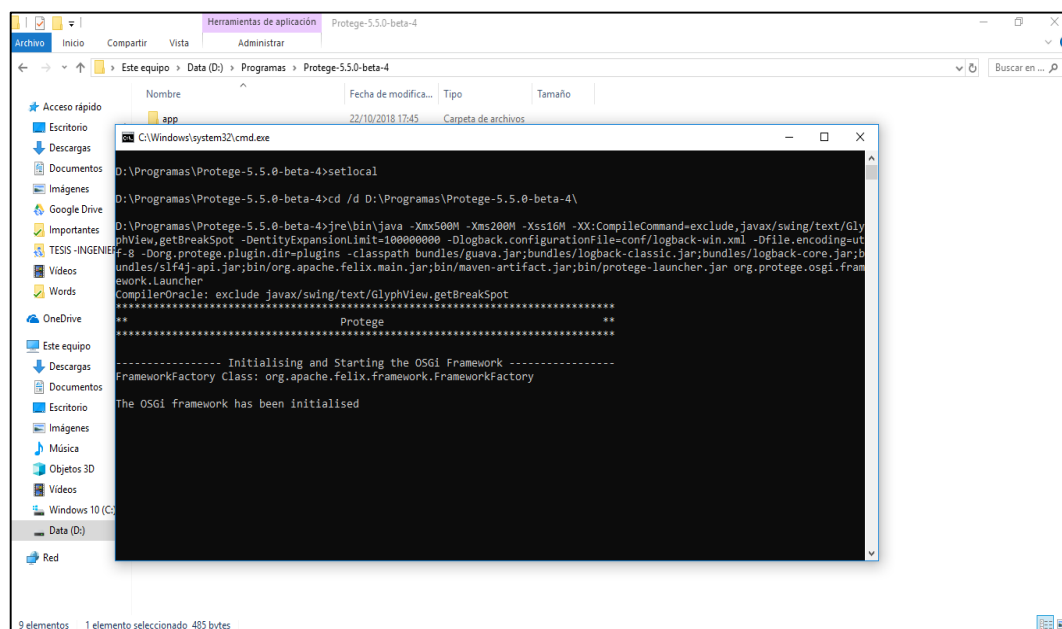


Figura 19: Arranque de la herramienta Protege 5.5.0

Fuente: Elaboración Propia



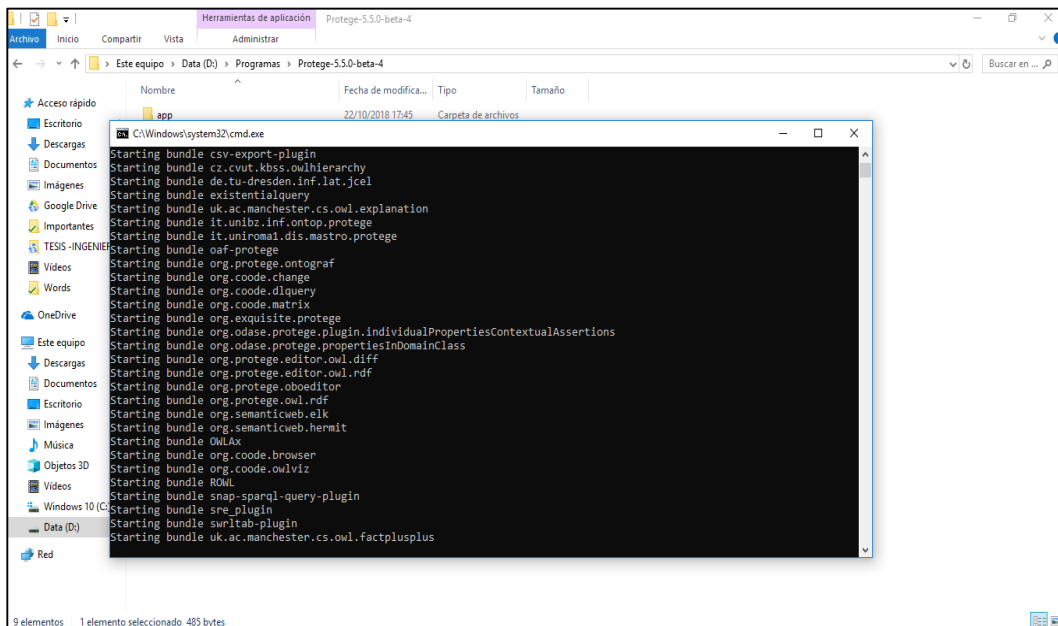


Figura 20: Levantamiento de plugin herramienta protege 5.5.0
Fuente: *Elaboración Propia*

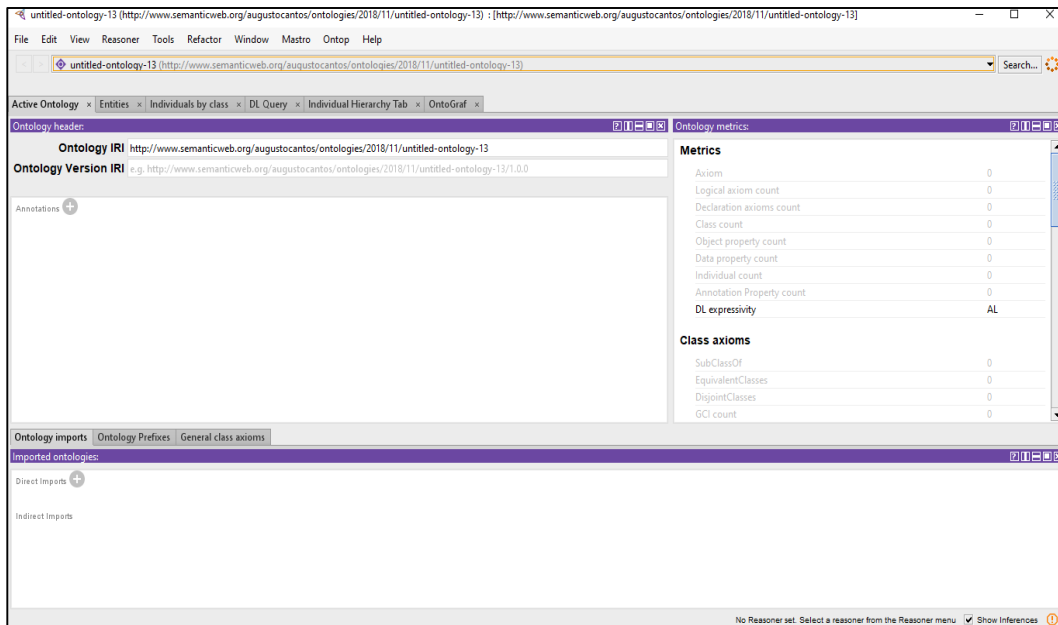


Figura 21: Herramienta Protege 5.5.0
Fuente: *Elaboración Propia*



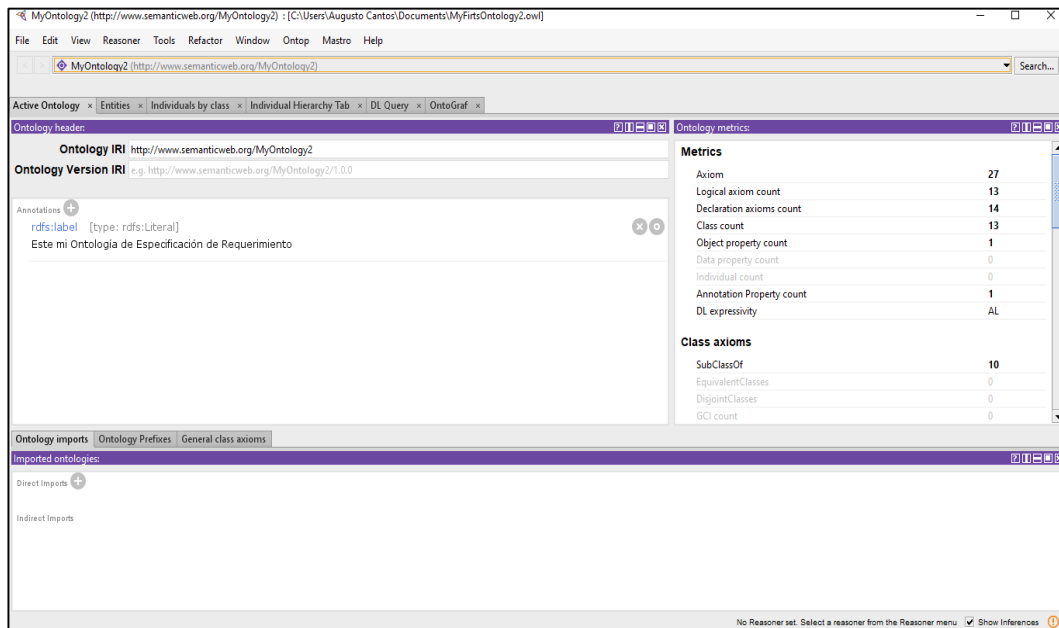


Figura 22: Creando mi ontología
Fuente: Elaboración Propia

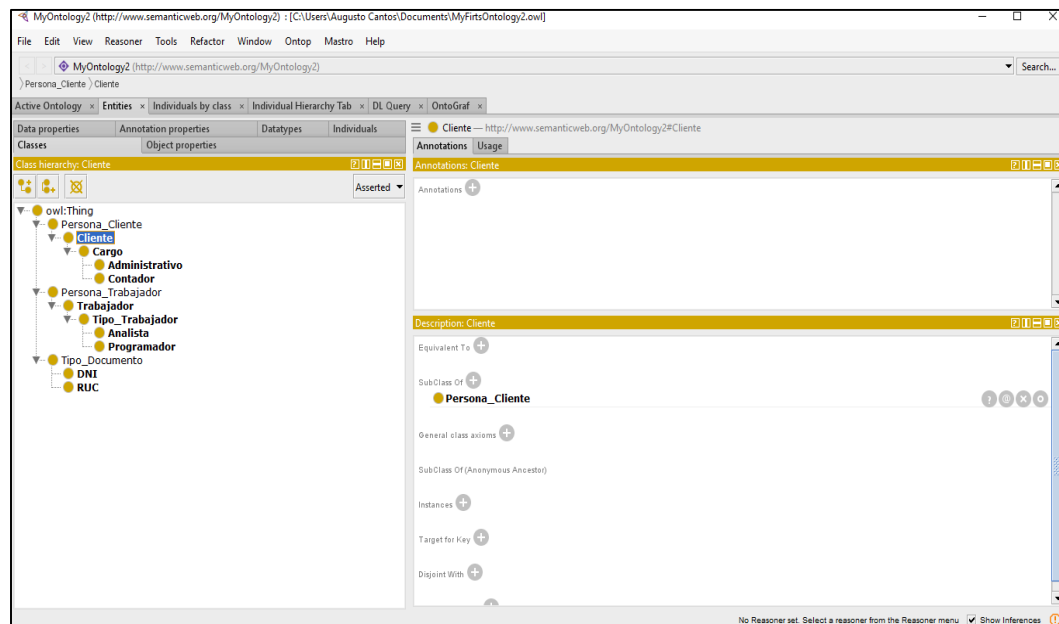


Figura 23: Clases y sub clases de mi ontología
Fuente: Elaboración Propia



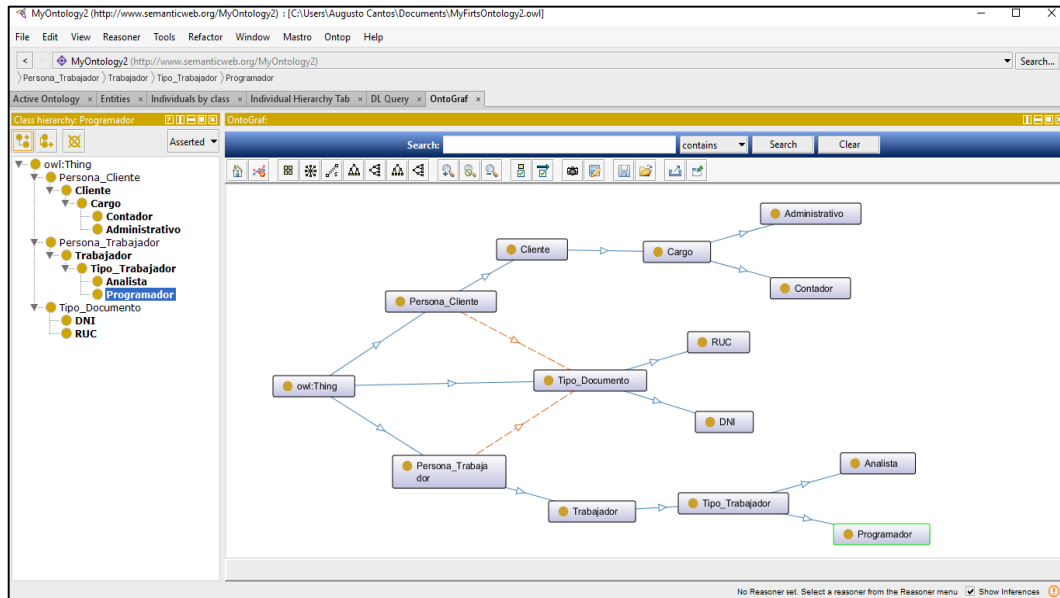


Figura 24: Grafico ontológico
Fuente: Elaboración Propia

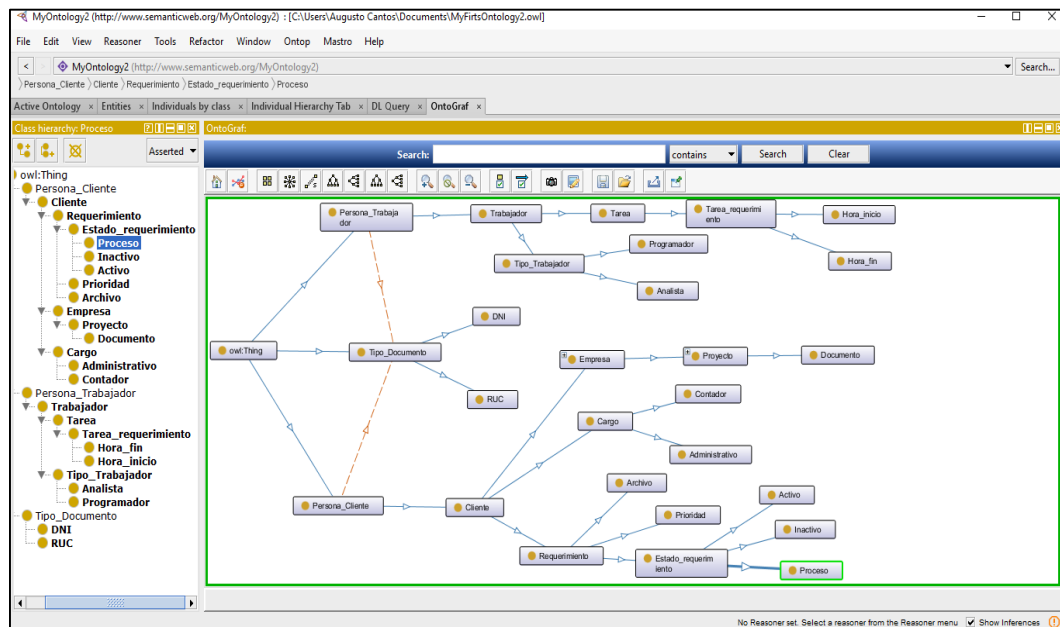


Figura 25: Proceso Grafico ontológico
Fuente: Elaboración Propia



Por lo tanto, siguiendo con los pasos de la ontología se requiere expresar el modelo Ontológico mediante un mapa conceptual basados en clases, subclases y relaciones como se muestra en la **Figura 26**,

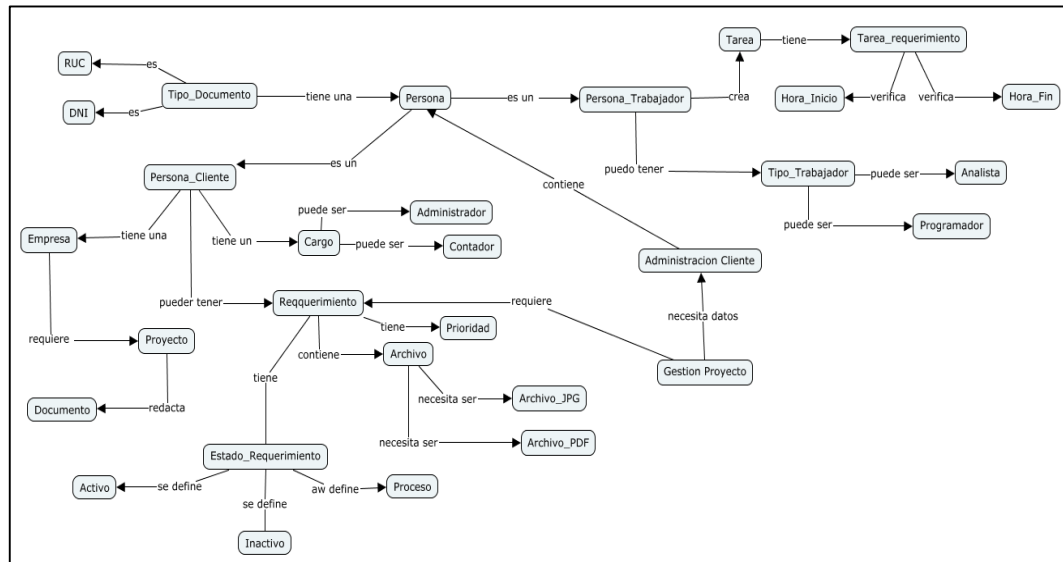


Figura 26: Mapa conceptual basado en ontología
Fuente: *Elaboración Propia*

Después de implementar las técnicas, teniendo los requerimientos definidos, elaboraremos el product backlog del marco de gestión ágil Scrum.

3.3.3.3. Metodología Scrum

3.3.3.3.1. Descripción de sistema

Se ha realizado el sistema de Gestión de proyectos de software y administración de clientes, para la gestión de proyectos de software, siendo una prioridad, las atenciones de los requerimientos, administración de la cartera de clientes y partiendo de la definición y aprobación de los requerimientos.



El sistema está integrado por los siguientes módulos: Gestión de proyectos y administración de clientes.

3.3.3.3.2. Personas y Roles

En el desarrollo del proyecto los participantes del Team de trabajo estuvo conformado por 2 personas con los siguientes datos y detalle de sus roles se mostrará en la siguiente **Tabla 21**.

Tabla N° 21: Persona y Roles.

PERSONA	EMAIL	ROL
Cesar Dávila Vallejos	Cesar.davila@gmail.com	Product Owner Scrum
Carlos Garcia Riojas	Carlos.riojas@gmail.com	Master Scrum
Danny F. Otero Arrascue	oarrascuedannyf@crece.uss.edu.pe	Equipo Scrum
Augusto Cantos Morante	cantosmo@crece.uss.edu.pe	Equipo

Fuente: Elaboración Propia

3.3.3.3.3. Historias de Usuario

Las historias de usuario (HU) se desarrollan en la ejecución de cada sprint, de acuerdo a la planificación del proyecto. Las HU son una descripción de una funcionalidad que debe implementar un sistema de software, partiendo de la incorporación de valor del cliente.



Tenemos que establecer prioridad de los requerimientos que ya están definidos en el documento de especificación de requerimiento. **(Anexo 02).**

Por lo consiguiente se planifica el product backlog en el marco de gestión ágil Scrum teniendo los requerimientos ya definidos, así como se describe en la siguiente tabla.

3.3.3.3.4. Product Backlog

Por resultante, se mostrará la lista de requerimientos del Product Backlog como se muestra en la **Figura 27**

Historia del usuario	Puntos	
Mantenimiento de Cliente	3	= ✎
Funcionalidad Login	2	= ✎
Mantenimiento Estado de Proyecto	0	= ✎
Mantenimiento Tipo de Requerimiento	0	= ✎
Mantenimiento Estado de Requerimiento	0	= ✎
Documentos de Proyectos	2	= ✎
Mantenimiento Cargo	0	= ✎
Mantenimiento de Tipo Documento	0	= ✎

Figura 27: Listado de requerimientos (Product Backlog)
Fuente: *Elaboración Propia*



Requerimiento	Cantidad	Acciones
Mantenimiento Cargo	0	[Edit] [Delete]
Mantenimiento de Tipo Documento	0	[Edit] [Delete]
Archivos de Requerimientos	2	[Edit] [Delete]
Mantenimiento de Empresa	0	[Edit] [Delete]
Prioridad de Empresa	0	[Edit] [Delete]
Prioridad de Requerimiento	0	[Edit] [Delete]
Categoría de Requerimiento	0	[Edit] [Delete]
Mantenimiento de Requerimiento de los Clientes	2	[Edit] [Delete]
Mantenimiento de Trabajador	2	[Edit] [Delete]
Mantenimiento de Proyectos	3	[Edit] [Delete]

Figura 28: Lista de requerimientos (Product Backlog)
Fuente: *Elaboración Propia*

Por lo tanto, seguiremos con los Sprint, en la **Figura 29** podemos observar el primer Sprint Mantenimientos de cliente: listar cliente, registrar cliente, modificar cliente y modificar cliente

Aún por hacer	Realizándose actualmente	Listas
	<ul style="list-style-type: none"> Modificar Cliente Eliminar Clientes 	<ul style="list-style-type: none"> Listar Clientes Registrar Clientes

Figura 29: Sprint mantenimiento cliente
Fuente: *Elaboración Propia*

Por conclusión se obtendrá un software establecido por el caso de estudio de la empresa CESLY SOFT & BUSINESS S.R.L, como se muestra en la **Figura 30**.



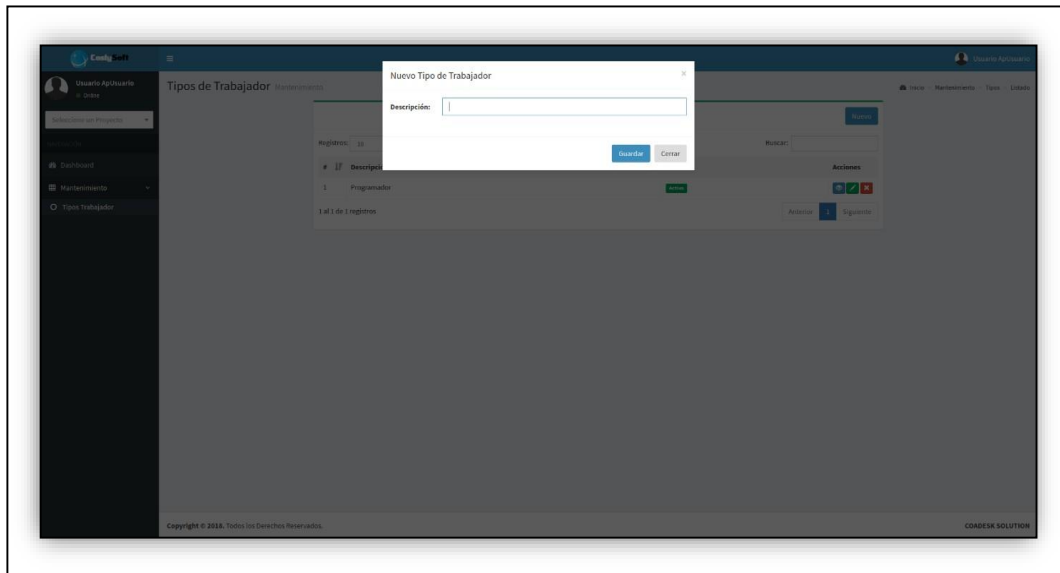


Figura 30: Insertar tipo de trabajador
Fuente: Elaboración Propia

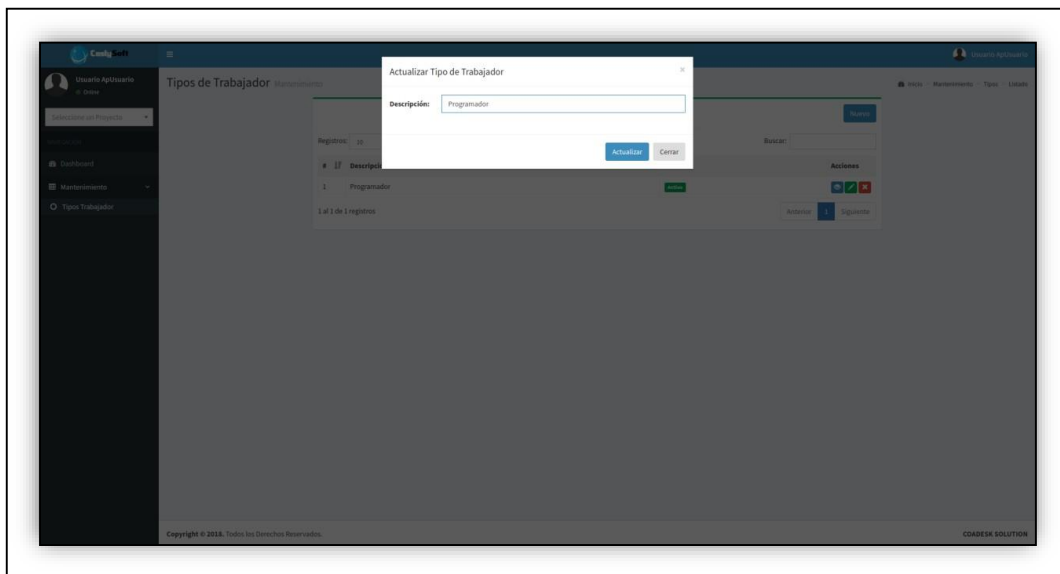


Figura 31: Actualizar tipo de trabajador
Fuente: Elaboración Propia



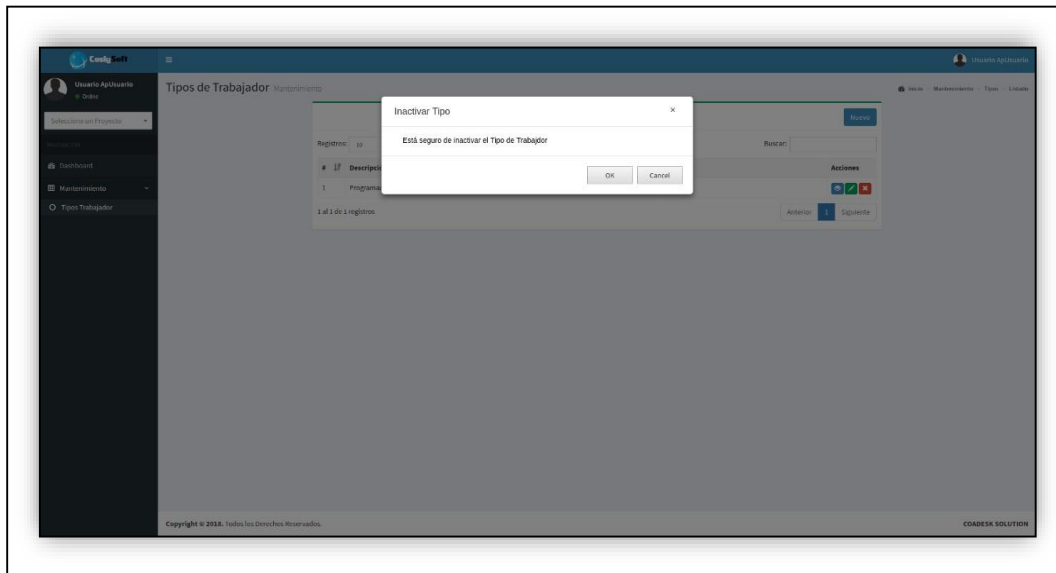


Figura 32: Eliminar tipo de trabajador
Fuente: Elaboración Propia

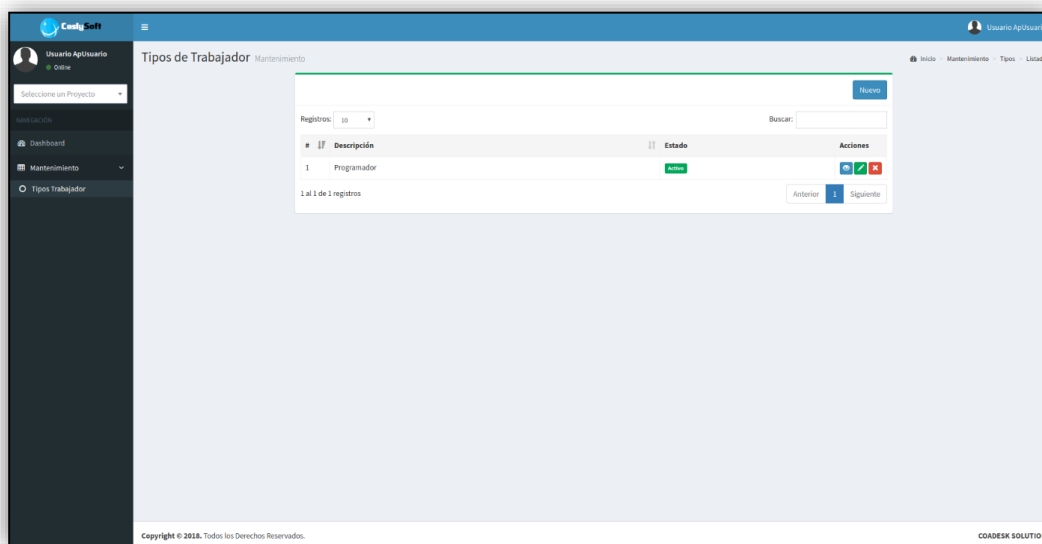


Figura 33: Mantenimiento tipo de trabajador
Fuente: Elaboración Propia



Habiendo implementado las técnicas de mapas conceptuales basado en ontología y modelo conceptual se llegó a definir el product backlog implementando el marco de gestión ágil Scrum, por lo consiguiente se tendrá que medir los indicadores de rendimiento para obtener y mostrar resultados de dichas técnicas de la etapa de especificación de requerimiento.

IV. CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- a. Se tomó el criterio de uso y se realizó una indagación bibliográfica de estudios previos lo cual nos permitió fundamentar la elección de las técnicas mapas conceptuales basado en ontología y modelo conceptual, donde los investigadores tuvieron como población a desarrolladores de software, analistas de software.
- b. La elección de los indicadores como: consistencia y no ambiguo, fueron seleccionadas por la (IEEE-STD-830, 1998), siendo la principal regla de medición que permitieron a dar a conocer que técnica se desempeña un óptimo rendimiento.
- c. Se implementó un sistema de gestión de proyectos de software y administración de cartera de clientes, el cual permitió de forma práctica evidenciar y conocer de forma amplia el desarrollo de cada técnica de especificación, mediante el cual se analizaron los indicadores.
- d. Se comparó las técnicas mapas conceptuales basado en ontología y modelo conceptual, obteniendo resultados con mayor porcentaje la técnica de mapas conceptuales basado en ontología, para mayor consistencia y no ambiguo, así logaremos mejorar la especificación de requerimientos a la hora del desarrollo de software del caso de estudio CESLY SOFT & BUSINESS S.R.L

RECOMENDACIONES

- a. Se recomienda utilizar la técnica de mapas conceptuales basado en ontología en diferentes proyectos de desarrollo de software, ya que permite un mejor desglosamiento de las entidades o clases, asimismo nos permitirá reducir la ambigüedad de los requerimientos captados por el usuario y determinar la fragilidad de la herramienta Protege 5.5.0.
- b. Para un estudio a futuro se recomienda hacer una comparación de otras técnicas de la etapa de especificación de ingeniería de requerimientos, añadiendo la utilización de las métricas de la (IEEE-STD-830, 1998).
- c. Se recomienda utilizar la herramienta Protege 5.5.0, donde ayudara a un mejor entendimiento de gráficos y procesos, para así poder reducir la ambigüedad a la hora del desarrollo del software, ya que hoy en día por las malas interpretaciones entre el desarrollador y el cliente llegan a fracasar la mayoría de los proyectos de software.

REFERENCIAS

- A. Toro, L. E. P. (2016). Ingeniería de Requisitos: de la especificación de requisitos de software al aseguramiento de la calidad. Cómo lo hacen las Mipymes desarrolladoras de software de la ciudad de Pereira. *Entre Ciencia e Ingeniería*, 10(20), 117–123.
- Ali Olow Jim'ale Sabriye, W. M. N. Z. (2017). Requirement Specification, 209–213.
- Aoki, Y., & Matsuura, S. (2014). Verifying security requirements using model checking technique for UML-based requirements specification. *2014 IEEE 1st International Workshop on Requirements Engineering and Testing (RET)*, 18–25. <https://doi.org/10.1109/RET.2014.6908674>
- Arias, S. A. G. (2014). Sebastián Alonso Gómez Arias.
- Besrou, S., Bin, L., Rahim, A. B., Dominic, P. D. D., Rahim, A. B., & Dominic, P. D. D. (2016). Investigating Requirement Engineering Techniques in The Context of Small and Medium Software Enterprises Investigating Requirement Engineering Techniques in The Context of Small and Medium Software Enterprises, 519–523.
- Besrou, S., Rahim, L. B. A., & Dominic, P. D. D. (2016). Investigating requirement engineering techniques in the context of small and medium software enterprises. *2016 3rd International Conference on Computer and Information Sciences, ICCOINS 2016 - Proceedings*, 519–523. <https://doi.org/10.1109/ICCOINS.2016.7783269>
- Buestán-Borja, C. D., & Cuji-Torres, V. A. (2013). Metodología para la especificación de requerimientos de software basado en el estándar IEEE 830-1998.
- Cáceres-Saldaña, E. (2014). METODOLOGÍA PARA EL REUSO EFECTIVO DE PATRONES DE REQUISITOS EN LA INGENIERÍA DE SOFTWARE.

- Corporation, I. B. M. (2006). IBM Rational Rose. Retrieved from hi
- Cueva, S., & Sucunuta, M. (2017). *Ingeniería de Requisitos Texto-guía*.
- Darío, J., & Torres. (2014). Guía Para La Integración De Métodos Formales De Ingeniería De Requerimientos En Procesos De Desarrollo Ágil, 1–67.
- De-la-Cruz-Londoño, C. A., & Castro-Guevara, G. A. (2014). La Ingeniería de Requerimientos en las Pequeñas Empresas del Departamento de Risaralda. *Lámpsakos*, 0(12), 110–119. <https://doi.org/10.21501/21454086.1354>
- De la Cruz-Londoño, C. A., & Castro-Guevara, G. A. (2015). METODOLOGÍA PARA LA ADQUISICIÓN Y GESTIÓN DE REQUERIMIENTOS EN EL DESARROLLO DE SOFTWARE PARA PEQUEÑAS Y MEDIANAS EMPRESAS (PYMES) DEL DEPARTAMENTO DE RISARALDA. *Tesis de Grado*, 217. Retrieved from <http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/5589/0053D278.pdf?sequence=1>
- Duran T, A. (2000). Un entorno metodológico de Ingeniería de Requisitos para Sistemas de Información, 373. Retrieved from http://fondosdigitales.us.es/media/thesis/30/O_Tesis-18.pdf
- Ellatif, M. A. (2017). Using Ontology-Based Concept Maps to Reduce Requirement Engineering Errors. *International Journal of Computer Science and Information Security (IJCSIS)*, 15(6), 292–303. <https://doi.org/10.1093/mnrasl/slv078>
- G, C. R. S., & Losavio, F. (2013). Free Software tools. *2013 XXXIX Latin American Computing Conference (CLEI) Equipos*, 27–34. <https://doi.org/10.1109/CLEI.2013.6670648>
- Hernández, R., Fernández, C., & Baptista, P. (2014). *Metodología de la investigación*. *Journal of Chemical Information and Modeling* (Vol. 53). <https://doi.org/10.1017/CBO9781107415324.004>
- Hurtado, D. J. (n.d.). Comparative analysis between the techniques used in



- the Requirements Engineering , evaluating such techniques forehead to the characteristics of software projects.
- IEEE-STD-830. (1998). Especificaciones de los requisitos del software., 27.
- Kaiya, H., & Saeki, M. (2005). Ontology based requirements analysis: Lightweight semantic processing approach. *Proceedings - International Conference on Quality Software, 2005*, 223–230. <https://doi.org/10.1109/QSIC.2005.46>
- Kumar, A., & Beniwal, R. (2016). 2Qwrroj \ Edvhg) Udphzrun Iru ' Hwhfwlqj \$ Peljxlwlhv Lq 6Riwzduh 5Htxluhphqwv, 3572–3575.
- Manuel, T. G., & Cristina, T. D. A. (2012). Metodología Scrum. GESTION DE PROYECTOS INFORMÁTICOS Tabla de contenido, 56. Retrieved from <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigastFC0612memoria.pdf>
- Marcelo, I., Facultad, M., Córdoba, R., Tecnológica, U., Pérez, M., Facultad, C., & Vigo, U. De. (n.d.). Construcción de una ontología utilizando Protégé para la elicitación de requerimientos.
- Mohamed, K. A., Ellatif, M. A., & Farhan, M. S. (2017). Using ontology-based concept maps for requirements engineering: A case study. *2017 13th International Computer Engineering Conference (ICENCO)*, 366–371. <https://doi.org/10.1109/ICENCO.2017.8289816>
- Pardo-Pastrana, A. M. (2017). Un Método de Elicitación de Requisitos Para SCRUM Compuesto Por Inception Deck y Modelos de Proceso de Negocios (BPMN).
- Ramírez, S. M., Alonso, Y., Hernández, V., Arias, A. C., & Rosa, D. La. (2010). Comparando UML y OWL en la representación del conocimiento: correspondencia sintáctica. *REICIS. Revista Española de Innovación, Calidad e Ingeniería Del Software*, 6(3), 84–94. Retrieved from <http://www.redalyc.org/resumen.oa?id=92218768008%5Cnhttp://www>



- .redalyc.org/pdf/922/92218768008.pdf
- Raunak, M. S., & Binkley, D. (2017). Agile and other trends in software engineering. *2017 IEEE 28th Annual Software Technology Conference (STC)*, 1–7. <https://doi.org/10.1109/STC.2017.8234457>
- Segura-Fiquitiva, N., & Lopez-Ruis, M. A. (2015). Prototipo de Aplicativo para Especificar Requerimientos de Software.
- Silvana del Valle, R. (2013). Elicitación y Especificación de Requerimientos no Funcionales en Aplicaciones Web, 228.
- Soares, H. A., & Moura, R. S. (2015). A methodology to guide writing Software Requirements Specification document. *Proceedings - 2015 41st Latin American Computing Conference, CLEI 2015*. <https://doi.org/10.1109/CLEI.2015.7360001>
- Sommerville, I. (2011). *Software Engineering*. (Castillo, Luis M. Cruz, Ed.), *Software Engineering* (PEARSON ED). Mexico. <https://doi.org/10.1111/j.1365-2362.2005.01463.x>
- Trigas-Gallega, M. (n.d.). Metodología Scrum . □.
- Vidya Sagar, V. B. R., & Abirami, S. (2014). Conceptual modeling of natural language functional requirements. *Journal of Systems and Software*, 88(1), 25–41. <https://doi.org/10.1016/j.jss.2013.08.036>



ANEXOS

Anexo 01: Acta de Constitución del proyecto (Project Charter)

Documento de Negocio:



(Project Charter)

GESTIÓN DE PROYECTOS DE SOFTWARE Y ADMINISTRACIÓN DE CARTERA DE CLIENTES



ACTA DE CONSTITUCIÓN DEL PROYECTO

PROYECTO	GESTIÓN DE PROYECTOS DE SOFTWARE Y CARTERA DE CLIENTES				
PATROCINADOR	CESLY SOFT & BUSINESS S.R.L				
PREPARADO POR:	Danny Frank Otero Arrascue	FECHA	06	09	2018
REVISADO POR	Carlos García Riojas	FECHA	10	01	2018
APROBADO POR	Cesar Vallejos Dávila	FECHA	10	01	2018

REVISIÓN	DESCRIPCIÓN	FECHA		
01	Preparación de Acta de Constitución	06	10	2018
02	Firma del Acta de Constitución	15	11	2018

BREVE DESCRIPCIÓN DEL PRODUCTO O SERVICIO DEL PROYECTO

El presente proyecto está orientado para la gestión de proyectos de software, siendo una prioridad, las atenciones de los requerimientos, administración de la cartera de clientes y la restructuración del sitio web de la empresa, partiendo de la definición y aprobación de los requerimientos.

Alinear los procesos establecidos por los interesados, de acuerdo al negocio, para así mejorar la flexibilidad en el manejo del sitio web. Aplicando la ingeniería de requerimientos, partiendo de los procesos de análisis y especificación, para así lograr una definición de los requerimientos, de manera precisa, correcta, consistente y no ambiguo.



La administración de la cartera de clientes tiene como objetivo incrementar la eficiencia de los esfuerzos realizados para retener y vincularse a sus clientes, obteniendo a los clientes satisfechos con el producto o servicio que se brinda, dando como resultado contar con la información más detallada y ordenada acerca de los clientes.

ALINEAMIENTO DEL PROYECTO

OBJETIVOS ESTRATÉGICOS DE LA ORGANIZACIÓN

PROPÓSITO DEL PROYECTO

Ejercer la administración de la cartera de clientes, gestionar los proyectos de software, dando priorización a las peticiones de los clientes, conllevando a una atención eficiente y eficaz a los problemas cotidianos, creando una buena actividad generadora de progreso económico en la empresa CESLY SOFT & BUSINESS S.R.L.

Desarrollar un sitio web, que permita administrar la cartera de clientes, controlando la gestión de los proyectos de software, priorizando las peticiones de los clientes de alto nivel, generando un manejo adecuado de los tiempos, en la empresa CESLY SOFT & BUSINESS S.R.L.

Desarrollar los procesos de la mejor manera a través del uso intensivo de las Tecnologías de la Información.

OBJETIVOS DEL PROYECTO

OBJETIVO GENERAL

OBJETIVOS ESPECÍFICOS

Desarrollar e implementar el sitio web,

- Analizar y especificar los requerimientos pedidos por la empresa Cesly Soft &



para la administración de carteras de clientes y gestión de proyectos de software de la empresa Cesly Soft & Business S.R.L.

- Business S.R.L utilizando las técnicas de Ingeniería de Requerimientos.
- Diseñar y crear la Base de Datos para la gestión de proyectos de software para el sitio web Cesly Soft & Business S.R.L.
 - Desarrollar el sitio web para la empresa Cesly Soft & Bussiness SRL alineándolos con los procesos de negocio solicitado.

FACTORES CRÍTICOS DEL ÉXITO DEL PROYECTO

Proceso y adquisición de los servicios del hosting y compra del dominio del sitio web, con la interactividad y la parte visual acorde a los requerimientos de los interesados.

Recopilar todos los datos relacionados, acerca al proyecto en la empresa Cesly Soft & Business S.R.L.

Componer un equipo multidisciplinario, para coordinar los plazos establecidos para el diseño y desarrollo del sitio web.

REQUERIMIENTOS DE ALTO NIVEL

El sitio web será desarrollado con varias tecnologías para poder alcanzar una alta concurrencia de usuarios al mismo tiempo.

1. Back – end

- a) SO Ubuntu 18.4
- b) Apache



- c) Php 5
- d) PostgreSQL
- e) Laravel Framework

2. Front – End

- a) HTML 5
- b) JS
- c) CSS
- d) Bootstrap
- e) VUEjs

DESCRIPCIÓN DEL ALCANCE DEL PROYECTO

1. EL ALCANCE DEL PROYECTO INCLUYE:

Se explorará los procesos de negocios de la empresa Cesly Soft & Business S.R.L, que permita desarrollar el sitio web que se necesita.

Requerimientos Funcionales del Negocio:

- Realizar un control de la cartera de clientes.
- Gestionar los proyectos de software.
- Gestionar los requerimientos de los clientes.
- Priorizar las peticiones de los clientes.
- Realizar el seguimiento de los requerimientos de los clientes, verificando el estado, categoría y tipo en que se encuentra.
- Calcular el tiempo de atención de las peticiones del cliente.
- El cliente pedirá las peticiones, asignándose un usuario y contraseña.
- Completitud Funcional: El sistema debe abarcar todo el proceso de: Gestión de Proyectos y cartera de clientes. El sistema debe estar completo en no menos de 3 meses.
- Corrección Funcional: El sistema debe generar los resultados precisos y Correctos para un 100 % de los RF



- Adecuación Funcional: El sistema debe adecuarse en no menos del 95% de las funciones de cada usuario.

Requerimientos no funcionales:

- **Concurrencia:** Usuarios concurrentes. Se refiere a la cantidad de usuarios que ingresan al Sistema al mismo tiempo
- **Madurez:** El sistema debe permitir, en condiciones normales, cubrir las funcionalidades al 100%.
- **Disponibilidad:** El sistema debe estar disponible el 95% del día, siendo el 4% restante para mantenimiento en horas de la noche y 1% por tolerancia a fallas
- **Capacidad de recuperación:** El sistema debe tener mecanismo que eviten la pérdida de información ante un fallo de software o Hardware.
- **Integridad:** Se debe contar con un sistema de cifrado. El sistema debe implementar la función creación de respaldos de la información
- **Adaptabilidad:** El sistema estará desarrollado bajo entorno para web.
- **Reusabilidad:** Se utilizará como base Framework de desarrollo.
- **Escalabilidad:** El sistema debe estar diseñado para soportar un crecimiento de 30% usuarios anuales durante 4 años.
- **Facilidad de Instalación:** El sistema no permitirá instalación por parte de usuario común, sin embargo, podrá permitir gestión de contenido por parte del usuario

1. Alcance	Desarrollo e implementación del sistema, con documentación y capacitación a los usuarios.	Aprobación de los entregables por parte del cliente.
2. Tiempo	Estimado de 06 meses.	Culminar el proyecto al 30 de diciembre del 2018.
3. Costo	Estimado de S/ 3,000	Culminar el proyecto con un incremento máximo de hasta 10%.



2. EL ALCANCE DEL PROYECTO NO INCLUYE:

El alcance de proyectos no incluye una aplicación móvil.

3. COSAS QUE SE ASUMEN:

- El equipo encargado del proyecto se encuentra capacitado para culminar satisfactoriamente el proyecto.
- La disponibilidad de los StakeHolders en el proyecto.

4. BENEFICIOS PARA EL CLIENTE:

Con la culminación del proyecto a realizar el cliente tendrá una mejor gestión de sus proyectos de software y la cartera de clientes de manera eficiente.

ORGANIZACIÓN DEL PROYECTO

EQUIPO DEL PROYECTO

- **Patrocinador:** Cesar Vallejos Dávila - Cesly Soft & Business S.R.L.
 - Gerente del proyecto: Augusto Enrique Cantos Morante
- **Miembros del equipo:**
 - Augusto Enrique Cantos Morante
 - Danny Frank Otero Arrascue.



STAKEHOLDERS / USUARIOS

CLIENTES: Empresas que han adquirido sistemas de información.

USUARIOS: Actores que tienen una autenticación.

GERENCIA: Gerente Cesar Vallejos Dávila

DESARROLLADORES: Equipo de desarrollo de software.

INVESTIGADOR: Carlos García Rioja

JEFE DE PROYECTOS: Carlos Vallejos Dávila

FASES E HITOS DEL PROYECTO

FASES DE LA INGENIERIA DE REQUERIMIENTOS

Existen cuatro actividades básicas (Elicitación, Análisis, Especificación y Validación) que se tienen que llevar a cabo para completar el proceso. Estas actividades ayudan a reconocer la importancia que tiene, para el desarrollo de un proyecto de software, realizar una especificación y administración adecuada de los requisitos de los clientes o usuarios.

Elicitación: Esta fase representa el comienzo de cada ciclo. Elicitación es el nombre comúnmente dado a las actividades involucradas en el descubrimiento de los requisitos del sistema.

Análisis: Sobre la base de la extracción realizada previamente, comienza esta fase. Usualmente se hace un análisis luego de haber producido un bosquejo inicial del documento de requisitos; aquí se leen los requisitos, se conceptúan, se investigan, se intercambian ideas con el resto del equipo, se resaltan los problemas, se buscan alternativas y soluciones, y luego se van fijando



reuniones con el cliente para discutir los requisitos.

Especificación: En esta fase se documentan los requisitos acordados con el cliente, en un nivel apropiado de detalle. En la práctica, esta etapa se va realizando conjuntamente con el análisis, pero se podría decir que la Especificación es el “pasar en limpio” el análisis realizado previamente aplicando técnicas y/o estándares de documentación, como la notación UML.

Validación: La validación es la etapa final de la IR. Su objetivo es verificar todos los requisitos que aparecen en el documento especificado para asegurarse que representan una descripción, por lo menos, aceptable del sistema que se debe implementar. Esto implica verificar que los requisitos sean consistentes y que estén completos.

HITO	FECHA
Acta de Constitución del Proyecto	06/10/2018
Plan de Gestión del Proyecto	08/10/2018
Plan de Iteración	10/10/2018
Plan de Gestión de Requerimientos	12/10/2018
Plan de Control de Cambios	13/10/2018
Plan de Desarrollo de Software	14/10/2018
Análisis de requerimiento (Inicio)	15/10/2018
Análisis de requerimiento (Fin)	20/10/2018
Especificación de requerimiento (Inicio)	21/10/2018
Especificación de requerimiento (Fin)	25/10/2018
Construcción (Inicio)	26/10/2018
Product backlog	26/10/2018



Sprint 1(Inicio)	27/10/2018
Sprint 1(Fin)	10/11/2018
Sprint 2(Inicio)	12/11/2018
Sprint 2(Fin)	24/11/2018
Construcción (Fin)	24/11/2018
Transición (Inicio)	25/11/2018
Transición (Fin)	26/11/2018
Acta de Cierre del Proyecto	30/11/2018

ENFOQUE DE TRABAJO

Para realizar este proyecto se tomará como referencia el proceso de desarrollo de la ingeniería de requerimientos, el cual utiliza las diferentes técnicas para el análisis y especificación de los requerimientos funcionales y no funcionales, en la construcción del software, se manejará el marco de gestión ágil Scrum, que constituye la creación del product backlog, donde se enunciará los requerimientos captados para el desarrollo de los Sprints (tareas).

Principales características que tiene este proceso de desarrollo:

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo
- Administración de requisitos
- Uso de arquitectura basada en componentes



- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software

RIESGOS

La información de la cartera de clientes no está organizada en un 90%. El seguimiento de los requerimientos no está registrado correctamente para su control. Perdidas en los bienes de la empresa, por el mal manejo de la priorización de los requerimientos de los clientes. Los requerimientos atendidos, no concuerdan con el listado de peticiones del cliente. Las ganancias no se incrementan de acuerdo a lo establecido por el área de gerencia.

FIRMA DE APROBACIÓN DEL ACTA

Cargo	Nombre	Firma	Fecha
Representante del Sponsor	Cesar Dávila Vallejos		06/10/2018
Coordinador del proyecto	Carlos García Riojas		06/10/2018
Gestor del Proyecto	Cesar Dávila Vallejos		06/10/2018

Observaciones:



Aprobado por:

Augusto Enrique Cantos Morante
Firma del Gerente del Proyecto

Fecha: _____

Cesar Vallejos Dávila
Firma del Patrocinador

Fecha: _____

Carlos García Riojas
Coordinador del Proyecto

Fecha: _____

Danny Frank Otero Arrascue
Analista de Sistema

Fecha: _____



Anexo 02: Especificación de requerimientos software

Documento de Negocio:

Septiembre 2018
CHICLAYO - PERÚ

Gestión de Proyectos de Software y Administración de Cartera de Clientes

Especificación de Requerimientos del Software

Versión 1.0

112



CONTROL DE VERSIONES

VERSIÓN	DESCRIPCIÓN DEL CAMBIO	FECHA DE CAMBIO	MODIFICADO POR
1.0	Se cambió el nombre del documento y se agregaron los puntos: personas autorizadas en definir los requerimientos y Cronograma de reunión de definición de requerimientos	16/09/2009	Equipo de Ingeniería del Proyecto
1.0	Se elaboró el informe de especificación de requerimientos del software	01/10/2009	Augusto Cantos Morante



CONTENIDO

1.	OBJETIVO DEL DOCUMENTO.....	115
2.	DESCRIPCION DEL PROYECTO	115
3.	DOCUMENTOS BASE PARA LA ESPECIFICACION DE REQUERIMIENTOS.....	115
4.	PERSONAS AUTORIZADAS EN DEFINIR REQUERIMIENTOS.....	116
5.	CRONOGRAMA DE REUNIONES DE DEFINICIÓN DE REQUERIMIENTOS	116
6.	DIAGRAMA DE CONTEXTO.....	117
7.	REQUERIMIENTOS DEL CLIENTE (INICIALES).....	85
8.	REQUERIMIENTOS FUNCIONALES POR PROCESO	118
	TABLA DE REQUISITOS	118
8.1	PROCESO DE NEGOCIO : GESTIÓN DE CARTERA DE CLIENTES	120
8.2	PROCESO DE NEGOCIO : GESTIÓN DE PROYECTOS.....	121
8.3	RELACION CON OTROS SISTEMAS (INTERFACES).....	128
9.	REQUERIMIENTOS NO FUNCIONALES.....	128
10.	FIRMAS DE APROBACION	130



1. OBJETIVO DEL DOCUMENTO

Definir y aprobar los requerimientos a ser desarrollados en el proyecto, especificando el alcance, restricciones y demás características de los mismos y lograr el compromiso de todos los involucrados en la ejecución del Proyecto.

2. DESCRIPCION DEL PROYECTO

El presente proyecto está orientado a la Gestión Proyecto y Directorio de Clientes, partiendo de la definición y aprobación de los requerimientos desarrollados, tomando en consideración los requerimientos funcionales propios de la gestión proyecto y directorio de clientes y los requerimientos no funcionales, infraestructura, herramientas de software, flujos de trabajo y actividades.

3. DOCUMENTOS BASE PARA LA ESPECIFICACION DE REQUERIMIENTOS

DOCUMENTO	DESCRIPCION BREVE
Base del Proyecto	El cliente no presentó ninguna base del proyecto se sustenta con la reunión inicial con el cliente (reglamentos preliminares y requerimientos de alto nivel coordinados con cliente).
Propuesta Técnica y Económica	Propuesta de solución presentada por los estudiantes de la Universidad Señor de Sipan.
Modelo del Negocio	Documento elaborado por el AS en conjunto con el cliente en donde se especifica el giro o movimiento del negocio, se da a conocer quien son los actores principales de las actividades que se realiza.



4. PERSONAS AUTORIZADAS EN DEFINIR REQUERIMIENTOS

NOMBRES Y APELLIDOS	CARGO	AREA
Cesar Vallejos Dávila	Gerente General	Gerencia
Carlos García Riojas	Área de Investigación	Investigación

5. CRONOGRAMA DE REUNIONES DE DEFINICIÓN DE REQUERIMIENTOS

FECHA	PARTICIPANTES	AGENDA
08/09/2018 de 8:00 am – 10:00 am	<i>Cesar Vallejos Dávila</i> <i>Carlos García Riojas</i> • Cantos Morante Augusto E. (AS)	<i>Especificación de RI001 al RI008</i>
22/09/2018 de 8:00 am – 10:00 am	<i>Cesar Vallejos Dávila</i> <i>Carlos García Riojas</i> • Cantos Morante Augusto E. (AS)	<i>Especificación de RF001 al RF020</i>
22/09/2018 de 8:00 am – 10:00 am	<i>Cesar Vallejos Dávila</i> <i>Carlos García Riojas</i> • Cantos Morante Augusto E. (AS)	<i>Especificación de RNF001 al RNF008</i>



6. DIAGRAMA DE CONTEXTO



7. REQUERIMIENTOS DEL CLIENTE (Iniciales)

CODIGO	PRIORIDAD (A, M, B)	DESCRIPCIÓN BREVE
RI -001	A	Controlar la cartera de clientes de la empresa.
RI -002	A	Controlar detalladamente los proyectos que tiene cada cliente con la empresa.
RI -003	A	Gestionar los requerimientos registrados de los clientes.
RI -005	M	Listado de los requerimientos según la prioridad de cliente.
RI -006	A	Realizar el ingreso de requerimientos según el cliente lo establezca para el software.



RI -007	A	Controlar las operaciones (crear, modificar, eliminar) de los requerimientos de clientes.
RI -008	M	Realizar el mantenimiento del tipo de requerimiento de los clientes, que deberá permitir (crear, modificar, eliminar).
RI -009	M	Controlar todos los requerimientos que genera un cliente, así mismo clasificar en base al tipo de requerimientos y asignar la prioridad y estado.
RI -010	A	Realizar el ingreso, modificación y eliminación de documentos asignados a los proyectos de software.
RI -011	A	Crear Tareas de acuerdo al requerimiento solicitado por el cliente, así mismo dicha tarea deberá poderse asignar a un trabajador.
RI -012	A	Controlar las tareas asignadas a un trabajador, basándose en medir su rendimiento en horas.

8. REQUERIMIENTOS FUNCIONALES POR PROCESO

Los requerimientos funcionales especifican lo que el sistema tiene que desarrollar, definiendo así el propósito y funcionalidad que el usuario requiere. A continuación se listan y especifican los requerimientos funcionales agrupados por proceso de negocio del cliente.

Tabla de Requisitos

CODIGO	REQUERIMINETOS
RF -001	Registrar Proyectos
RF -002	Registrar Clientes
RF -003	Registrar Trabajador
RF -004	Listar características técnicas adicionales de los Clientes
RF -005	Registrar características técnicas adicionales de los Clientes
RF -006	Realizar mantenimiento de proyecto según la prioridad de cliente



RF -007	Realizar listado de cartera de clientes
RF -008	Registrar los requerimientos de cliente
RF -009	Modificar los requerimientos de cliente
RF -010	Eliminar los requerimientos de cliente
RF -011	Estado de requerimientos de cliente
RF -012	Listar los requerimientos priorizados del cliente
RF -013	Registrar el tipo de requerimiento de los clientes
RF -014	Modificar el tipo de requerimiento de los clientes
RF -015	Eliminar el tipo de requerimiento de los clientes
RE -016	Crear tipo de requerimiento de los clientes
RE-017	Modificar mantenimiento de tipo de requerimiento de los clientes
RE-018	Eliminar mantenimiento de tipo de requerimiento de los clientes
RE-019	Clasificar en base al tipo de requerimientos de los clientes
RE-020	Asignar la prioridad a los requerimientos del cliente
RE-021	Realizar el ingreso de documentos asignados a un proyecto
RE-022	Eliminar documentos asignados a un proyecto
RE-023	Registrar una tarea y asignarla a un trabajador
RE-024	Modificar una tarea asignada a un trabajador
RE-025	Eliminar una tarea asignada a un trabajador
RE-026	Registrar las horas que se trabajaron por tarea
RE-027	Modificar las horas que se trabajaron por tarea
RE-028	Eliminar las horas que se trabajaron por tarea
RE-029	Visualizar progreso del trabajador por tarea asignada



8.1. PROCESO DE NEGOCIO: gestión de cartera de clientes

CODIGO REQUER	RF-001	DESCRIPCION BREVE	Registrar los datos propios de un cliente	PRIORIDAD	A
PRE REQUISITO	Registrar datos del cliente				
DESCRIPCION DETALLADA					
La empresa deberá registrar los datos de ésta con respecto a su empresa, como son su dirección, departamento, ciudad, provincia. Además de especificar la empresa el cual pertenece.					
POST CONDICION	Cliente registrado.				

CODIGO REQUER	RF-002	DESCRIPCION BREVE	Registrar a los trabajadores que ingresarán al sistema	PRIORIDAD	A
PRE REQUISITO	Registrar datos del trabajador				
DESCRIPCION DETALLADA					
La empresa deberá tener un registro de los trabajadores que manejarán el sistema, para que de ésta manera tenga un control de los responsables de las operaciones que se realicen.					
POST CONDICION	Trabajador registrado.				

CODIGO REQUER	RF-003	DESCRIPCION BREVE	Listar las características técnicas adicionales de los clientes	PRIORIDAD	M
PRE REQUISITO	Cliente listado.				
DESCRIPCION DETALLADA					
Se deberá listar, además de sus datos generales, ciertas características técnicas adicionales que tiene el cliente, como por ejemplo: datos para acceso remoto, teamviewer, anydesk.					
POST CONDICION	Característica técnica de cliente listado.				



CODIGO REQUER	RF-004	DESCRIPCION BREVE	Registrar las características técnicas adicionales de los clientes	PRIORIDAD	A
PRE REQUISITO	Cliente registrado.				
DESCRIPCION DETALLADA					
Se deberá registrar, además de sus datos generales, ciertas características técnicas adicionales que tiene todo cliente, como por ejemplo: datos para acceso remoto, teamviewer, anydesk.					
POST CONDICION	Característica técnica de proyecto registrado.				

8.2. PROCESO DE NEGOCIO: gestión de proyectos

CODIGO REQUER	RF-005	DESCRIPCION BREVE	Registrar los proyectos de cada cliente	PRIORIDAD	A
PRE REQUISITO	Datos de proyecto registrado				
DESCRIPCION DETALLADA					
La empresa tiene un control acerca de todos los proyectos, tanto de la misma empresa como para los clientes, registrando sus nombres, dirección, documento de identidad, números telefónicos, etc.					
POST CONDICION	Proyecto registrado.				

CODIGO REQUER	RF-006	DESCRIPCION BREVE	Registrar requerimientos de cliente	PRIORIDAD	A
PRE REQUISITO	Requerimiento registrado.				
DESCRIPCION DETALLADA					
A partir de ciertos requerimientos la empresa tendrá que solucionar el pedido o petición que solicite el cliente para alguna modificación o error de su proyecto de software. Esta Petición es, generalmente, realizada por pedido del cliente que desea tener preferencia de que si existieran algún error sea solucionada y que satisfagan todas las preferencias del cliente.					
POST CONDICION	Requerimiento solicitado por el cliente.				



CODIGO REQUER	RF-007	DESCRIPCION BREVE	Modificar requerimientos de cliente	PRIORIDAD	M
PRE REQUISITO	Requerimiento modificado.				
DESCRIPCION DETALLADA					
A partir de registrar un requerimiento solicitado por el cliente, por ende el cliente tendrá la facilidad de modificar el requerimiento ya que no haya sido realizado su petición del mismo, la empresa tendrá la facilidad de poder solucionar su requerimientos solicitado por el cliente.					
POST CONDICION	Requerimiento solicitado por el cliente.				

CODIGO REQUER	RF-008	DESCRIPCION BREVE	Eliminar requerimientos de cliente	PRIORIDAD	M
PRE REQUISITO	Requerimiento eliminado.				
DESCRIPCION DETALLADA					
A partir de registrar un requerimiento solicitado por el cliente, por ende el cliente tendrá la facilidad de eliminar el requerimiento.					
POST CONDICION	Requerimiento solicitado por el cliente.				

CODIGO REQUER	RF-009	DESCRIPCION BREVE	Estado de requerimiento de cliente	PRIORIDAD	A
PRE REQUISITO	Estado de requerimiento				
DESCRIPCION DETALLADA					
El estado del requerimiento solicitado por el cliente tendrá tres sub condiciones, si el requerimiento está en espera, en proceso o está atendido por los trabajadores de la empresa.					
POST CONDICION	Estado de requerimiento en procesos o atendido				



CODIGO REQUER	RF-010	DESCRIPCION BREVE	Listar los requerimientos priorizados de los clientes	PRIORIDAD	M
PRE REQUISITO	Requerimientos registrados. Ser Trabajador de la Empresa.				
DESCRIPCION DETALLADA					
Después de que el cliente haya ingresado un determinado requerimiento, un trabajador se encargará de realizar el trabajo de cambiar el estado del requerimiento, tales como En Espera, En Proceso y Atendido.					
POST CONDICION	Listado de priorización realizado.				

CODIGO REQUER	RF-011	DESCRIPCION BREVE	Realizar mantenimiento de proyecto según la prioridad de cliente	PRIORIDAD	A
PRE REQUISITO	Proyecto registrado.				
DESCRIPCION DETALLADA					
La empresa tendrá a su disposición la prioridad del cliente con respecto al proyecto que desea, para realizar un mantenimiento que satisfaga las exigencias del cliente. De no hacerlo el cliente ingresa a una lista de espera, o puede solicitar información acerca del por qué no se realizó su mantenimiento.					
POST CONDICION	Prioridad del cliente establecidos.				

CODIGO REQUER	RF-012	DESCRIPCION BREVE	Registrar el Tipo de Requerimiento de los Clientes	PRIORIDAD	A
PRE REQUISITO	Tipo de requerimiento registrado.				
DESCRIPCION DETALLADA					
Los Requerimientos enviados por los clientes deberán tener un Tipo obligatoriamente, por lo tanto se necesita el registro de los Tipos de requerimiento posibles.					
POST CONDICION	Registrar tipo de requerimiento				



CODIGO REQUER	RF-013	DESCRIPCION BREVE	Modificar el tipo de requerimiento de los clientes	PRIORIDAD	M
PRE REQUISITO	Tipo de requerimiento modificado.				
DESCRIPCION DETALLADA					
Los tipos de requerimientos deben ser modificados según se requiera, por motivos de equivocación al registrar o cambios en los nombres.					
POST CONDICION	Modificado tipo de requerimiento				

CODIGO REQUER	RF-014	DESCRIPCION BREVE	Eliminar el tipo de requerimiento de los clientes	PRIORIDAD	M
PRE REQUISITO	Tipo de requerimiento eliminado.				
DESCRIPCION DETALLADA					
Los Tipos de Requerimiento deberán poder ser eliminados según ser requiera.					
POST CONDICION	Eliminado tipo de requerimiento				

CODIGO REQUER	RF-015	DESCRIPCION BREVE	Clasificar en base al tipo de requerimientos de los clientes.	PRIORIDAD	A
PRE REQUISITO	Tipo de requerimiento clasificado.				
DESCRIPCION DETALLADA					
Los requerimientos enviados por los clientes deben poder ser clasificados de acuerdo al Tipo de Requerimiento, y poder dar cierta prioridad a los tipos de requerimientos más urgentes.					
POST CONDICION	Clasificación de tipo requerimiento				



CODIGO REQUER	RF-016	DESCRIPCION BREVE	Asignar la prioridad a los requerimientos del cliente	PRIORIDAD	A
PRE REQUISITO	Prioridad de requerimiento de cliente				
DESCRIPCION DETALLADA					
El Administrador podrá asignar la prioridad a un determinado requerimiento, esta prioridad podría ser: alta, media ,baja dependiendo de la gravedad del mismo.					
POST CONDICION	Prioridad Asignada				

CODIGO REQUER	RF-017	DESCRIPCION BREVE	Realizar el ingreso de documentos asignados a un proyecto	PRIORIDAD	A
PRE REQUISITO	Asignación de Documento.				
DESCRIPCION DETALLADA					
Cada proyecto deberá tener una sección de documentos asignados al mismo. Estos documentos pueden ser de los formatos: docx y pdf.					
POST CONDICION	Documentos asignados al proyecto				

CODIGO REQUER	RF-018	DESCRIPCION BREVE	Eliminar documentos asignados a un proyecto	PRIORIDAD	M
PRE REQUISITO	Asignación de Documento.				
DESCRIPCION DETALLADA					
Estos documentos pueden ser eliminados por un trabajado, ya sea en formato docx o pdf.					
POST CONDICION	Documento asignado eliminado				



CODIGO REQUER	RF-019	DESCRIPCION BREVE	Registrar una tarea y asignarla a un trabajador	PRIORIDAD	A
PRE REQUISITO	Registrar tarea				
DESCRIPCION DETALLADA					
El administrador asignara la tarea a un trabajador basándose en los requerimientos solicitados por un cliente.					
POST CONDICION	Tarea registrada				

CODIGO REQUER	RF-020	DESCRIPCION BREVE	Modificar una tarea asignada a un trabajador	PRIORIDAD	M
PRE REQUISITO	Modificar tarea				
DESCRIPCION DETALLADA					
El administrador modificara la tarea de un trabajador basándose en los requerimientos solicitados por un cliente					
POST CONDICION	Tarea Modificada por el administrador				

CODIGO REQUER	RF-021	DESCRIPCION BREVE	Eliminar una tarea asignada a un trabajador	PRIORIDAD	M
PRE REQUISITO	Eliminar tarea				
DESCRIPCION DETALLADA					
El administrador eliminara la tarea de un trabajador basándose en los requerimientos solicitados por un cliente					
POST CONDICION	Tarea eliminada por el administrador				



CODIGO REQUER	RF-022	DESCRIPCION BREVE	Registrar las horas que se trabajaron por tarea	PRIORIDAD	A
PRE REQUISITO	Horas trabajadas				
DESCRIPCION DETALLADA					
El trabajador podrá registrar las horas trabajadas por tareas, las cuales fueron asignadas por el administrador. Pueden haber múltiples ingresos para una sola tarea.					
POST CONDICION	Registros de horas trabajadas por tareas				

CODIGO REQUER	RF-023	DESCRIPCION BREVE	Modificar las horas que se trabajaron por tarea	PRIORIDAD	M
PRE REQUISITO	Horas trabajadas				
DESCRIPCION DETALLADA					
El trabajador podrá modificar las horas trabajadas por tareas, las cuales fueron asignadas por el administrador. Pueden haber múltiples ingresos para una sola tarea.					
POST CONDICION	Modificar horas trabajadas por tareas				

CODIGO REQUER	RF-024	DESCRIPCION BREVE	Eliminar las horas que se trabajaron por tarea	PRIORIDAD	M
PRE REQUISITO	Horas trabajadas				
DESCRIPCION DETALLADA					
El trabajador podrá eliminar las horas trabajadas por tareas, las cuales fueron asignadas por el administrador. Pueden haber múltiples ingresos para una sola tarea.					
POST CONDICION	Eliminar horas trabajadas por tareas				



CODIGO REQUER	RF-025	DESCRIPCION BREVE	Visualizar progreso del trabajador por tarea asignada	PRIORIDAD	M
PRE REQUISITO	Progreso de Trabajador				
DESCRIPCION DETALLADA					
El administrador podrá visualizar el progreso de un trabajador por tarea en forma porcentual. Para llevar un mejor control, y así tener conocimiento del tiempo que requiere un trabajador para realizar una determinada tarea.					
POST CONDICION	Visualización de progreso del trabajador.				

8.3. RELACION CON OTROS SISTEMAS (interfaces)

No hay sistemas que se relacionen con el sistema de Gestión.

9. REQUERIMIENTOS NO FUNCIONALES

1.REQUERIMIENTO DE INTERFACES DE USUARIO	
COD REQ	DESCRIPCION
RNF -001	La interfaz debe estar basada en formatos web estándares.
2.REQUERIMIENTO DE HARDWARE E INFRAESTRUCTURA	
COD REQ	DESCRIPCION
RNF -002	El sistema se debe implementar sobre la infraestructura existente en la empresa “CeslySoft”, que cuenta con un sistema operativo Linux, distribución Ubuntu 18.4, Core i5 y 8GB de RAM.
3.REQUERIMIENTO DE SOFTWARE	
COD REQ	DESCRIPCION
RNF -003	La aplicación deberá funcionar sobre la plataforma Web.



RNF -004	La aplicación se conectará con el motor de base de datos PostgreSQL 10
4.REQUERIMIENTO DE USABILIDAD OPERACIONAL	
COD REQ	DESCRIPCION
RNF -005	La aplicación generará entregables en formato PDF.
5.REQUERIMIENTO DE SEGURIDAD	
COD REQ	DESCRIPCION
RNF -006	El sistema contempla seguridad mínima de ingreso al sistema.
6.REQUERIMIENTO DE CAPACITACION	
COD REQ	DESCRIPCION
RNF -007	Se capacitará al usuario en un entrenamiento de 3 horas aproximadamente.
7.REQUERIMIENTO DE MIGRACION	
COD REQ	DESCRIPCION
El proyecto no presenta este tipo de requerimiento.	
8.REQUERIMIENTO LEGALES	
COD REQ	DESCRIPCION
El proyecto no presenta este tipo de requerimiento.	



10. FIRMAS DE APROBACIÓN

Cesar Vallejos Dávila
Gerente General
CESLY SOFT & BUSINESS SRL.

Carlos García Riojas
Coordinador del Proyecto
CESLY SOFT & BUSINESS SRL.

Augusto Enrique Cantos Morante
Analista de Sistemas
COADESK SOLUTION SRL.

Chiclayo, 01 de Octubre de 2018.



Anexo 03: Focha de Observación

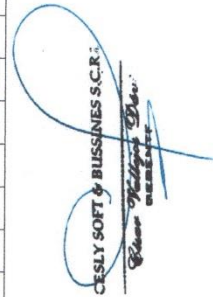
CeslySoft
www.ceslysoft.com

> Soluciones Empresariales

FICHA DE OBSERVACIÓN

Observado: Augusto Enrique Cantos Morante

N°	Nombres	Desempeños a observar									Comentarios u observaciones adicionales		
		Cuanta dificultad tuvieron a la hora de la definición en las técnicas.			Las técnicas ayudaron a tener un valor agregado para el desarrollo del software			Evalúa el conocimiento de dichos acuerdos y normas, y propone cómo mejorarlo los requerimientos definidos			Recomendaría utilizar las técnicas implementadas en los futuros proyectos de la empresa.		
		1	2	3	1	2	3	1	2	3	1	2	3
1	Cesar Vallejos Dávila	X					X						X
2	Carlos Garcia Riojas	X					X						X
3	Carlos Vallejos	X					X						X


 CESLY SOFT & BUSINESS S.C.R.L.
 Cesar Vallejos Dávila
 miembro

Dirección: Jr. Puerto Rico # 920 – Teléfono: 074-251042 – Email: informes@ceslysoft.com
 Chiclayo - Perú

