



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y
URBANISMO**

ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS

INFORME DE INVESTIGACIÓN

**UNA REVISIÓN DE LA LITERATURA ACERCA DE LA
COMPRENSIÓN DE REQUERIMIENTOS DE SOFTWARE
MEDIANTE EL LENGUAJE NATURAL**

Autor:

Otero Arrascue Danny Frank

Asesor:

Mg. Mejia Cabrera Heber Ivan

Línea de Investigación:

Ingeniería Tecnología y Urbanismo

Sub Línea de Investigación:

Ingeniería de Software

Pimentel – Perú

2019

DEDICATORIA

Dirigido con mucho cariño a mis padres, por ese apoyo incondicional y esas ganas de verme por segunda vez profesional, a mis hijos Dangelo y Santiago, por ser esa motivación extra para tener las fuerzas suficientes para realizarme como padre y profesional, a la paciencia incondicional de Anngye, la mujer que está a mi lado liderando este reto, también una dedicatoria especial a mis docentes, por todas las enseñanzas recibidas durante toda la 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 Mg. Heber Ivan Mejia Cabrera, por brindar sus conocimientos profesionales que ayudo a concluir este trabajo de pregrado.

RESUMEN

Los últimos avances sobre la ingeniería de requerimientos permiten utilizar una variedad de técnicas en diferentes procesos de la IR. Los errores que surgen durante Las actividades de la ingeniería de requerimiento son los desafíos más comunes que son enfrentando industrias de software. Estos errores ocurren en la concepción en las fases de desarrollo del producto software que causan problemas, los cuales conducen a la baja calidad software, fallas, proyectos incompletos o incluso dramáticamente detenidos.

Se han desarrollados métodos y técnicas para obtener la mayor precisión para la comprensión de requerimientos de software mediante el lenguaje natural tales como, Procesamiento de lenguaje natural, Ontologías de dominio expresadas en mapas conceptuales, lenguaje de modelado unificado entre otros.

Esta revisión de la literatura científica tiene como objetivo recopilar los documentos acerca de las diferentes técnicas en la ingeniería de requerimiento para la comprensión de los requerimientos de software, el cual facilitará de gran manera en la reducción de los errores en la definición, comunicación de los requerimientos de software durante el proceso de análisis de la ingeniería de requerimientos y así poder lograr una alta calidad en los proyectos de software.

De tal manera este material literario científico ha permitido ofrecer aportes a los analistas, programadores de software, docentes y estudiantes para los diferentes proyectos de software que realicen.

Palabras Claves:

Ingeniería de Requerimientos, Ontología, Mapas conceptuales, Lenguaje de Modelado Unificado, Rendimiento, Requerimientos de Software, Procesamiento de lenguaje Natural.

ABSTRACT

The latest advances in requirements engineering allow a variety of techniques to be used in different IR processes. Mistakes that arise during Requirement engineering activities are the most common challenges facing software industries. These errors occur on the basis of software projects that cause problems that lead to poor software quality, software failure, incomplete project or even dramatically crash.

Methods and techniques have been developed to obtain an understanding of the requirements in natural language such as, Natural language processing, Domain ontologies expressed in conceptual maps, unified modeling language among others.

This research aims to compile the bibliographic material documents of the different techniques in the requirements engineering that help in the understanding of the requirements, which will greatly facilitate the reduction of the errors of definition of the requirements during the process of requirements engineering analysis and thus achieve high quality software projects.

In this way, this bibliographic material has allowed us to offer contributions to analysts, software programmers, teachers and students for the different software projects they carry out.

KeyWord:

Requirements Engineering, Ontology, Concept Maps, Unified Modeling Language, Performance, Software Requirements, Natural Language Processig

CONTENIDO

DEDICATORIA.....	2
AGRADECIMIENTO.....	3
RESUMEN.....	4
ABSTRACT.....	5
CONTENIDO.....	6
ÍNDICE DE FIGURAS.....	7
ÍNDICE DETABLAS.....	8
I. INTRODUCCIÓN.....	9
1.1. Antecedentes de Estudio.....	11
1.2. Planteamiento del problema de investigación.....	17
1.3. Objetivos.....	17
1.3.1. Objetivo General.....	17
1.3.2. Objetivo Especifico.....	17
1.4. Marco teórico conceptual.....	17
1.4.1. Ingeniería de Requerimientos.....	17
1.4.1.1. Requerimiento.....	18
1.4.1.2. Requerimientos funcionales.....	19
1.4.1.2.1. Requerimientos de usuario.....	19
1.4.1.2.2. Requerimientos del sistema.....	19
1.4.1.3. Requerimientos no funcionales.....	20
1.4.1.3.1. Requerimientos de producto.....	20
1.4.1.3.2. Requerimiento de organización.....	21
1.4.1.3.3. Requerimientos externos.....	21
1.4.1.4. Procesos de Ingeniería de requerimientos.....	22
1.4.1.4.1. Elicitación.....	22
1.4.1.4.2. Análisis.....	23
1.4.1.4.3. Especificación.....	24
1.4.1.4.4. Validación.....	25
1.4.1.5. El procesamiento de lenguaje natural y la ingeniería de requerimientos.....	25
1.4.1.5.1. El procesamiento de lenguaje natural.....	26
1.4.1.6. Técnicas de la ingeniería de requerimientos.....	27
1.4.1.6.1. Entrevista.....	28
1.4.1.6.2. Observación.....	28
1.4.1.6.3. Lluvia de ideas.....	28
1.4.1.6.4. Diseño de aplicación conjunta (JAD).....	28
1.4.1.6.5. Modelamiento de lenguaje unificado (UML).....	29
1.4.1.6.5.1. Diagrama de caso de uso.....	29
1.4.1.6.5.2. Diagrama de clase.....	31
1.4.1.7. Ontología.....	31
1.4.1.7.1. Mapa conceptual basado en ontología utilizando la IR.....	31
II. MÉTODOS DE INVESTIGACIÓN.....	33
2.1. Plan de la Investigación.....	34
2.1.1. Interrogantes de la investigación.....	34
2.1.2. Protocolo de Revisión.....	34

2.1.3.	Validar Protocolo de revisión.....	36
2.2.	Documentación de la investigación	36
2.2.1.	Identificar las investigaciones relevantes	36
2.2.2.	Validar la información	38
2.2.3.	Evaluar la calidad de los datos	40
2.2.4.	Extraer datos requeridos.....	41
2.2.5.	Sintetizar los datos	41
2.3	Documentación de la investigación	59
2.3.1	Validar informe	59
III.	RESULTADOS.....	60
3.1	Análisis	61
IV.	CONCLUSIONES	72
4.1	Limitaciones.....	72
	REFERENCIAS	73

ÍNDICE DE FIGURAS

Figura 1.1	Tipos de requerimientos no funcionales.....	21
Figura 1.2	Proceso de la ingeniería de requerimientos.....	22
Figura 1.3	Proceso de descubrimiento de requerimientos.....	23
Figura 1.4	Proceso de análisis de requerimientos.....	24
Figura 1.5	Documentos de fase de especificación de requerimientos.....	25
Figura 1.6	Elementos de los diagramas de casos de uso fuente.....	30
Figura 1.7	Mapa conceptual extraído de una ontología.....	32
Figura 1.8	Arquitectura de la Ontología en la IR.....	32
Figura 2.1	Método de revisión bibliográfico.....	33
Figura 2.2	Resultado del filtro de la plataforma SCImago Journal & Country Rank.....	35
Figura 2.3	Resultado crudo de la Búsqueda de IEEE.....	37
Figura 2.4	Valoración de los artículos según el número de citas y la cantidad de referencia.....	39
Figura 3.1	Cantidad de artículos por año aplicando el filtro crudo tal cual arroja la regla de búsqueda.....	60
Figura 3.2	Tendencia a la cantidad de publicaciones de los artículos aplicando el filtro final según la regla de revisión.....	63
Figura 3.3	Selección de las técnicas de IR para la comprensión de los requerimientos.....	69

ÍNDICE DETABLAS

Tabla 1.1	Técnicas más utilizadas en las industrias de desarrollo de software en el proceso de la IR.....	21
Tabla 2.1	Búsqueda de base de datos y resultados	36
Tabla 2.2	Resumen de las propuestas planteadas desde el 2009 – 2018.....	39
Tabla 2.3	Resumen de impacto de las revistas (2009- 2018)	40
Tabla 2.4	Valoración de las palabras clave de acuerdo a Inspec.....	42
Tabla 3.1	Resultados más significativos sobre la comprensión de los requerimientos en lenguaje natural.....	62
Tabla 3.2	Cantidad de artículos publicados referencia sobre la comprensión de los requerimientos en lenguaje natural entre el año 2009 -2018.....	63
Tabla 3.3	Análisis de precisión de las técnicas de IR con mayor uso.....	70

I. INTRODUCCIÓN

Actualmente son muchas las técnicas de análisis de requerimiento en donde se encuentra sumergidos la ingeniería de requerimientos por recalcar alguna de estas son observación, lluvia de ideas, procesamiento de lenguaje natural (NLP), diseño de aplicación en conjunta, orientada a objetivos, modelo conceptual, check-list, mapas conceptuales basados en ontología, caso de uso (UML), entre otras. Todas estas se centran en intentar disminuir de manera eficiente la comprensión de los requerimientos de software mediante el lenguaje natural del cliente. En efecto, los requerimientos de usuario representan un valor altamente importante en el proceso de desarrollo de software, es por ello se busca mejorar todos los métodos desde la percepción de los requerimientos hasta llegar a la definición de los mismos, considerando la comprensión del cliente, siendo uno de los más importantes para asegurar la calidad del software. Constantemente en este punto es donde surge el principal problema ya que los requerimientos son expresados en lenguaje natural bajo la definición del usuario, la cual se supone serios inconvenientes al momento del entendimiento y claridad relacionadas a dichas exigencias, por ejemplo, es el uso del término requerimiento de usuario, como una notación de alto nivel, y requerimiento de sistema, para obtener una descripción detallada de lo que el sistema debe realizar generando una confusión del analista y el cliente donde se suelen asociar con la escasa comprensión teniendo problemas de comunicación y claridad en los requerimientos, por lo consecuente afecta el tiempo de desarrollo del producto[1,3]. Las industrias de desarrollo de software hoy en día generalmente aplica la técnica de mapa conceptual basado en ontología, que ayuda eficientemente en las actividades de desarrollo de software, ya que al reducir la comprensión y proveer un marco de unificación, ayudan a compartir conocimiento, facilitan la comunicación y permiten una alta reutilización del conocimiento[4]. La función de la ontología es facilitar el entendimiento común del conocimiento a los miembros del equipo de desarrollo de software, obteniendo una ontología integrando un conocimiento detallado para apoyar a cada fase del ciclo de vida del software[7]. Son las fases iniciales del ciclo de vida de software en las que se realiza un conjunto de procesos que parten desde el análisis que realizan los analistas hasta su especificación

en lenguajes formales y semiformales para el desarrollo del software.[7] esta práctica utilizando los mapas conceptuales ayudan de forma significativa el entendimiento compartido entre los participantes del desarrollo del software, donde la capa ontológica se puede transformar en un mapa conceptual de acuerdo a las similitudes entre sus elementos [8].

En la industria del desarrollo del software una de las técnicas más relevante es el uso de lenguaje de modelado unificado (UML), basándose en los diagramas de casos de uso, donde se describe las interacciones de entre el usuario y una pieza del software, partiendo desde descripciones de lenguajes naturales o controlados [9]. Para entender UML necesitamos construir un modelo conceptual [10], entendiendo su vocabulario donde engloba tres tipos de bloques de construcción que son: cosas, relaciones y diagramas, el objetivo es crear modelos de casos de uso a partir de los requerimientos del lenguaje natural, donde las oraciones en texto de entrada satisfacen algunos requisitos básicos [11]. Al día de hoy se han publicado investigaciones relevantes relacionadas con el tema en cuestión utilizando distintas técnicas antes mencionada. A pesar de ser un campo bastante estudiado los errores en la captura de requerimientos en las etapas iniciales del desarrollo de software siguen persistiendo ocasionado, atrasos de tiempo e pérdidas económicas para la industria de desarrollo de software de los países del mundo todavía sigue siendo un campo de investigación abierto, ya que es necesario superar muchos desafíos para poder garantizar la comprensión eficaz de los requerimientos. En consecuencia, el presente documento ofrece un estudio bibliográfico de las propuestas planteadas en cuanto al procedimiento mencionado.

1.1. Antecedentes de Estudio

[2] Proponen un método basado en las reglas heurísticas que permiten identificar los actores, los casos de usos y las relaciones especiales del diagrama de casos de uso, tomando como inicio una representación en lenguaje controlado de dominio del problema (Esquemas preconceptuales), conjuntamente implementan la herramienta heurística metaCASE AToM3. Obteniendo una construcción automática de esquemas conceptuales a partir del lenguaje natural finalizando en esquemas preconceptuales de UML.

[5] Proponen generar automáticamente especificaciones formales a partir del documento de requerimiento de lenguaje natural en coreano, utilizando el NPL para desarrollar el sistema de resolución Anafora. Dicho sistema consiste en analizar el documento de requerimiento, basándose a los antecedentes de los pronombres. Donde el pronombre se puede clasificar pronombres personal y demostrativo.

[7] Proponen la construcción de ontologías a partir de patrones de diseño basándose a un enfoque de ontologías de dominio durante el desarrollo del sistema de información. Se utilizo la generación automática de código conocida como síntesis de programa. Este proceso se basa en esquemas que representan el conocimiento computacional reutilizable y usan técnicas de la Inteligencia Artificial (IA) como los sistemas AutoBayes y AutoFilter. Obteniendo resultados eficientes en las etapas iniciales del desarrollo del software.

[10] Propusieron una adaptación de modelo MeRinde, utilizado las técnicas comúnmente de la IR, llegando a complementar los artefactos propios de MeRinde buscando mejorar la calidad del producto final, posteriormente se analizan e identifican a través de la metodología Systematic Review, las herramientas de software libre que den soporte a las técnicas seleccionadas. El modelo propuesto es descrito utilizando la notación estándar Software Pocess Engenieering Metamodel (SPEM) v2.0.

[16] Exponen que el lenguaje natural estructurado forma parte en la redacción de los requerimientos, donde el redactor de los mismos está limitado y los requerimientos se formulan de una manera estándar. Este artículo manifiesta la ventaja que el enfoque NLP mantiene la mayor parte de claridad y comprensión del lenguaje natural, propusieron utilizar un enfoque basado a NLP representados en formularios para especificar los requerimientos del sistema, para así definir formatos o plantillas específicas.

[11] Propusieron la visualización de los requerimientos funcionales, el modelo conceptual de entidades y el modelo de análisis y diseño orientado a objetos (OOAD), basados en lenguaje de modelado unificado (UML), se calculó las entidades similares aplicando las reglas de modelamiento conceptual implementado en Python usando la librería de kit de herramientas de lenguaje natural (NLTK), el resultado de la propuesta es de 81.82% de precisión.

[12] Presentaron un procesamiento de lenguaje natural y las técnicas basadas en ontología para aprovechar el modelo de dominio analizando la calidad de los requerimientos y la comprensión de los mismos. El prototipo brinda asesoramiento para que los requerimientos sean alineados con el estándar del dominio esperado, se utilizaron 38 requerimientos de varios proyectos en conjunto donde fueron validadas por un experto, se obtuvieron 7 de un total de 46 violaciones a las reglas dando un 86% de precisión.

[13] Presentaron la integración de la Inteligencia Artificial (IA) en la Ingeniería de que Requerimientos (IR), siendo clasificados en enfoques que transforman los requerimientos en modelos, tales como: Inspección, NLP, Heurísticas (reglas y métricas), Patrones/Gráfico y finalmente ontología. Las técnicas de NLP utilizaron una herramienta que proporcionaba una exportación a diagramas UML, códigos o archivos XML, todas las técnicas siguieron unos análisis computarizados, excepto Inspección que adopta un análisis manual, donde los requerimientos fueron tratados en lenguaje natural, transformándose en requerimientos mejorados para simplificar la extracción de datos como TESSI y QUARS y se obtuvo que los diagramas UML no fueron generados en su totalidad donde los enfoques fueron agrupados en 3 categorías principales: heurísticas,

NLP y ontología, donde se pudo detectar que el enfoque de la ontología producían más errores.

[14] Propusieron combinar un enfoque que adopta el uso de diagramas de actividad con especificaciones de lenguaje natural, para especificar las funciones del sistema, así obtener los errores de calidad del sistema, donde el diagrama de actividad crea una visión general temprana, la información contenida se transfiere al documento de requerimientos textuales, agregándose detalles y se refina el comportamiento del sistema, se pusieron a prueba 36 funciones del sistema Daimler, donde obtuvieron el 78% de inconsistencias entre documento de requerimiento y el diagrama de actividad. Para evitar problemas de calidad, utilizaron enfoques automáticos, para mantener sincronizados el diagrama de actividad y el documento de requerimientos, a partir de modelos gráficos, por ejemplo [8] Utilizaba modelos i * para derivar requerimientos, [9] propone un enfoque similar para el método orientado a objetivos (KAOS) y [26] con modelo soporte de herramientas, los resultados mostraron que 7 de cada 9 de cada problema de calidad fueron identificados obteniendo un 95% de precisión.

[15] Realizaron una encuesta en las diferentes industrias de desarrollo de software, donde se enfocaban cuáles eran las diferentes técnicas en la IR utilizadas en la actualidad, los participantes fueron los jefes de proyectos, analistas, programadores, tester, llegando a detallar que la puntuación abarca de 0 a 5 en la usabilidad de la técnica, mostrando resultados que las técnicas de procesamiento de lenguaje natural, obtuvieron un 4.08, donde implementaban la extracción de datos de los requerimientos textuales de los stakeholders transformándolos en lenguajes naturales estructurados, obteniendo un documentos de especificación de requerimientos más eficientes.

[27] Utilizaron las herramientas de procesamiento de lenguaje natural (NLP), para las especificaciones escritas de requerimientos en lenguaje natural (NL) convirtiéndolas en lenguajes formales de especificaciones o extrayendo relevante conocimiento de dominio tales como conceptos y relaciones, desde las especificaciones, obteniendo una reducción de ambigüedad, inconsistencia, carencia de legibilidad, comprensión, omisión y

redundancia.

[18] Plantearon utilizar 2 técnicas del NLP que son clustering semántico y reglas enriquecidas semánticas, para descubrir concerns ocultos en las especificaciones de casos de uso, donde la primera técnica combina NLP, algoritmos de clustering y métricas de similitud entre palabras para encontrar compartimientos repetidos que pueden ser requerimientos no funcionales denominados Crosscutting Concerns (CCCs) potenciales. La segunda técnica acumula mecanismos de búsqueda basados en reglas con información semántica de las oraciones para revelar familias de concerns mediante consultas específicas. Se realizó una evaluación experimental con 3 casos de estudios analizando los diferentes concerns encontrados en los diferentes documentos de requerimientos. Los resultados obtenidos fueron del 50%, 60% y 70% de precisión de detectar concerns ocultos.

[19] Realizaron una propuesta centrándose en los requerimientos escritos en inglés, proponiendo un enfoque para procesar los requerimientos escritos en tailandés. Extrayendo los diagramas de clases UML, usando las técnicas de procesamiento de lenguaje natural. La extracción del diagrama de clases UML se basa a las reglas de transformación que identifican clases y atributos a partir de los requerimientos. Los resultados mostraron que en 3 casos de estudios las clases extraídas tienen una precisión de 85% y recall de 90%, mientras tanto en atributos se tiene una precisión de 77% y recall 78%.

[20] Propusieron potenciar los procesos de comunicación pasando por 4 etapas: Identificación de técnicas de comunicación, especificación formal de las técnicas, definición metodológica de la IR, aplicación de lo propuesto en el desarrollo del sistema. En la propuesta metodológica para IR, se compone a 11 etapas donde se incorporan técnicas de elicitación de conocimiento (TEC) y técnicas de conocimientos (TC) al proceso de la IR. La etapas están conformadas por: Definición del modelos de descomposición de trabajo, identificación de las técnicas de ingeniería de requerimientos (TIR), formalización de TIR, clasificación TIR por actividades de IR, identificación de

TEC y TC, se recopilan de la bibliografías las técnicas a incorporar en IR, formalización de TEC y TC, elaboración de catálogo, identificación del problema de comunicación, identificación de estrategias de comunicación, aplicación de instrumentos, elaboración de matriz de decisión. Se realizo la propuesta en 5 empresas de Armenia. Los resultados son de entre 95% y 97% en la Precisión de la calidad de requerimientos obtenidos.

[21] Proponen ejecutar las técnicas de procesamiento de lenguaje natural (NLP) y recuperación de información, para clasificar los requerimientos de software (SRC), utilizando Deep Learning (DL). Basado en el estado del arte en la clasificación de oraciones en lenguaje natural (análisis de sentimientos, extracción de relaciones semánticas, etc.). Se utilizó el modelo de red neuronal convencional (CNN) para SRC escritos en lenguaje natural en las siguientes categorías: funcional, disponibilidad, legal, apariencia, mantenimiento, escalabilidad, seguridad, usabilidad, tolerancia a fallas y portabilidad, aplicando corpus PROMISE, que contiene un conjunto de requerimientos funcionales y 11 categorías diferentes de requerimientos no funcionales para evaluar el modelo. Los resultados promedios en las métricas de precisión son de 80% y recall 78.5% en la SRC.

[22] Desarrollaron un caso de estudio para encontrar la comprensión de los analistas de software y los stakeholders(interesados), utilizando mapas conceptuales (MP) como representación informal de requerimientos basados en ontologías como representación formal y explícita de los mismos. Para medir el grado de comparación entre métodos tradicionales y ontología, se utilizó la herramienta estadística Fleiss Kappa. Text2Onto integrada con General Architecture for Text (GATE), WorldNet y Treetagger son utilizados para el procesamiento del lenguaje natural y como la extracción de los componentes de la ontología. Para el refinamiento y modelado de las clases ontológicas. Implementaron Protege y las reglas de inferencia. Protege implementa la conversión de elementos OWL a elementos de MP para descubrir los errores de los requerimientos y luego superarlos. La fiabilidad de los MP producidos se midió en función de acuerdo entre 2 usuarios y un analista de software, para evaluar el grado de comprensión compartida y la calidad de información. El resultado muestra que el Kappa de Fleiss fue

73% en el marco provisto, 56% en ontología 40% y en métodos tradicionales.

[23] Propone un modelo formal de casos de uso de 3 niveles que consisten en caso de uso de funciones, nivel sistema y el nivel comercial. El caso de uso de funciones se formaliza con Hore Logic (Logica de Hore), un caso de uso del sistema se compone de un conjunto de casos de usos de funciones de acuerdo a 3 reglas de composición y caso de uso empresarial se formaliza por una red Petri. Después que se propone el modelo de caso de uso, se discute como una retroalimentación en el análisis de los requerimientos funcionales si tiene efecto sobre el modelo.

[24] Propusieron un marco de desarrollo de software MDA, donde el núcleo es un conjunto de transformaciones automáticos, como el modelo independiente de computadora (CIM). CIM es utilizado como un modelo de proceso de negocio. Se integra los modelos de requerimientos de lenguaje natural para que sean derivados a CIM. Luego se implementa una estrategia para obtener un diagrama de clase UML, llevando a cabo la transformación de reglas del mismo.

[34] Plantearon utilizar el proceso de análisis de requerimientos con las metodologías orientadas a objetos y las ágiles. El modelo conceptual de los requerimientos fue representado por UML, los cuales permite modelar los casos de usos, el comportamiento y los requerimientos, permitiendo involucrar a los actores del mismo en cuanto al funcionamiento del sistema. Obteniendo una precisión 85% de los requerimientos y su forma de representación.

[30] Desarrollaron una ontología basado en mapas conceptuales, implementando un método para construir la ontología de dominio de clase en forma de texto escrito, basado en el lenguaje natural (NL), usando diagramas UML en una arquitectura orientada al servicio, convirtiendo los requerimientos en dominio de clase expresados en un mapa conceptual, se midió la fiabilidad de los mapas conceptuales, con el apoyo de 2 expertos, utilizando la herramienta KAPPA como método estadístico para medir el grado de comprensión entre el equipo de software y el los stakeholders, tomando 35

requerimientos del proyecto, el resultado del experto de A fue de una precisión 82%, mientras tanto en el experto B obtuvo una precisión de 90%.

1.2. Planteamiento del problema de investigación

¿Cuál es el estado del conocimiento respecto al procesamiento de lenguaje natural utilizando las técnicas de la ingeniería de requerimientos para reducir los errores de definición de los requerimientos de software?

1.3. Objetivos

1.3.1. Objetivo General

Realizar una revisión del material bibliográfico sobre el procesamiento de lenguaje natural utilizando las técnicas de la ingeniería de requerimiento.

1.3.2. Objetivo Especifico

- a) Elaborar el plan de Investigación.
- b) Desarrollar el procedimiento de investigación.
- c) Crear la documentación de la investigación.

1.4. Marco teórico conceptual

Los fundamentos teóricos que soportan el proyecto son necesarios para entender el conocimiento de ciertos comportamientos de los requerimientos de software, siendo el procesamiento de lenguaje natural conjuntamente con las técnicas de las IR las más utilizado en la actualidad en la industria de desarrollo de software, lo que conlleva al uso de las ontologías, metodologías de procesos, UML, técnicas de NLP y las diferentes técnicas de la IR en sus diferentes ciclo de vida o también llamados procesos tales como; Elicitacion, Análisis, Especificación y Validación , para su correcta definición.

1.4.1. Ingeniería de Requerimientos

El proceso de recopilar, analizar y verificar las necesidades del cliente para un sistema de software es llamado Ingeniería de Requerimientos. La meta de la

ingeniería de requerimientos es entregar una especificación de requerimientos de software correcta y completa. La ingeniería de requerimientos apunta a mejorar la forma en que comprendemos y definimos sistemas de software complejos. Existen varias definiciones de requerimientos, de entre las cuales podemos citar las siguientes:

[51] El diseño y construcción de software de computadora es difícil, creativo y sencillamente divertido. En realidad, elaborar software es tan atractivo que muchos desarrolladores de software quieren ir directo a él antes de haber tenido el entendimiento claro de lo que se necesita. Argumentan que las cosas se aclararán a medida que lo elaboren, que los participantes en el proyecto podrán comprender sus necesidades sólo después de estudiar las primeras iteraciones del software, que las cosas cambian tan rápido que cualquier intento de entender los requerimientos en detalle es una pérdida de tiempo, que las utilidades salen de la producción de un programa que funcione y que todo lo demás es secundario. Lo que hace que estos argumentos sean tan seductores es que tienen algunos elementos de verdad. Pero todos son erróneos y pueden llevar un proyecto de software al fracaso.

1.4.1.1. Requerimiento

[52] Los requerimientos para un sistema son descripciones de lo que el sistema debe hacer: el servicio que ofrece y las restricciones en su operación. Tales requerimientos reflejan las necesidades de los clientes por un sistema que atienda cierto propósito, como sería controlar un dispositivo, colocar un pedido o buscar información. Al proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se le llama ingeniería de requerimientos (IR). El término “requerimiento” no se usa de manera continua en la industria del software. En algunos casos, un requerimiento es simplemente un enunciado abstracto de alto nivel en un servicio que debe proporcionar un sistema, o bien, una restricción sobre un sistema. En el otro extremo, consiste en una definición detallada y formal de una función del sistema.

1.4.1.2. Requerimientos funcionales

[52] Son enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse el sistema en situaciones específicas. En algunos casos, los requerimientos funcionales también explican lo que no debe hacer el sistema.

Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a las entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas pueden declarar explícitamente lo que el sistema no debe hacer [52,36]. Su representación es: lenguaje natural, modelos visuales, métodos formales.

Los requerimientos funcionales para un sistema refieren lo que el sistema debe hacer. Tales requerimientos dependen del tipo de proceso de software que se esté desarrollando, de los stakeholders y del enfoque general que adopta la organización cuando se escriben los requerimientos.

[52,36] los requerimientos funcionales definen: Cuáles entradas debe aceptar el sistema, Cuáles salidas debe producir el sistema, Qué datos debe almacenar el sistema que utilizarán otros sistemas, Que operaciones debe realizar el sistema, La sincronización y cronometraje de las actividades anteriores.

A los requisitos funcionales se le puede dividir:

1.4.1.2.1 Requerimientos de usuario

[52] son enunciados, en un lenguaje natural junto con diagramas, acerca de qué servicios esperan los usuarios del sistema, y de las restricciones con las cuales éste debe operar.

1.4.1.2.2 Requerimientos del sistema

Son descripciones más detalladas de las funciones, los servicios y las restricciones operacionales del sistema de software. El documento de requerimientos del sistema (llamado en ocasiones especificación funcional) tiene que definir con exactitud lo que se

implementará. Puede formar parte del contrato entre el comprador del sistema y los desarrolladores del software [52].

1.4.1.3 Requerimientos no funcionales

[52,36] Son requerimientos que no se relacionan directamente con los servicios específicos que el sistema entrega a sus usuarios. Pueden relacionarse con propiedades emergentes del sistema, como fiabilidad, tiempo de respuesta y uso de almacenamiento de forma alternativa, pueden definir restricciones sobre la implementación del sistema, como las capacidades de los dispositivos I/O las representaciones de datos usados en las interfaces con otros sistemas. L

Los requerimientos no funcionales, como el rendimiento, la seguridad o la disponibilidad, especifican o restringen por lo general características del sistema como un todo. Los requerimientos no funcionales a menudo son más significativos que los requerimientos funcionales individuales [36].

[52]La implementación de dichos requerimientos puede propagarse a lo largo del sistema. Para esto existen dos razones: Los requerimientos no funcionales afectan más la arquitectura global de un sistema que los componentes individuales. Por ejemplo, para garantizar que se cumplan los requerimientos de rendimiento, quizá se deba organizar el sistema para minimizar las comunicaciones entre componentes.

[52,36] Los requerimientos no funcionales surgen a través de necesidades del usuario, debido a restricciones presupuestales, políticas de la organización, necesidad de interoperabilidad con otro software o sistemas de hardware, o factores externos como regulaciones de seguridad o legislación sobre privacidad. Se tiene tipos de requerimientos no funcionales como: requerimiento de producto, organización, externos.

1.4.1.3.1 Requerimientos de producto

Estos requerimientos especifican o restringen el comportamiento del software. Los ejemplos incluyen requerimientos de rendimiento sobre qué tan rápido se debe ejecutar el sistema y cuánta memoria requiere, requerimientos de fiabilidad que establecen la tasa aceptable de fallas, requerimientos de seguridad y requerimientos de usabilidad [52].

1.4.1.3.2 Requerimiento de organización

Son requerimientos de sistemas amplios, derivados de políticas y procedimientos en la organización del cliente y del desarrollador. Los ejemplos incluyen requerimientos del proceso operacional que definen cómo se usará el sistema, requerimientos del proceso de desarrollo que especifican el lenguaje de programación, estándares del entorno o el proceso de desarrollo a utilizar, y requerimientos ambientales que definen el entorno de operación del sistema [52].

1.4.1.3.3 Requerimientos externos

Este término cubre todos los requerimientos derivados de factores externos al sistema y su proceso de desarrollo. En ellos se incluyen requerimientos regulatorios que establecen lo que debe hacer el sistema para ser aprobado en su uso por un regulador, como sería un banco central; requerimientos legislativos que tienen que seguirse para garantizar que el sistema opere conforme a la ley, y requerimientos éticos que garanticen que el sistema será aceptable para sus usuarios y el público en general [52].

Los tipos de requerimientos no funcionales se pueden describir a continuación en la figura 1.1.

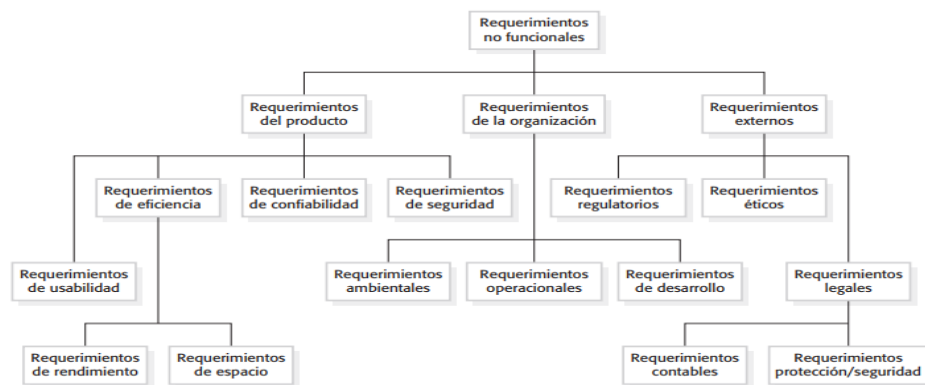


Figura 1.1: Tipos de requerimientos no funcionales **fuentes:**[52]

1.4.1.4 Procesos de Ingeniería de requerimientos

Según [53] refiere de manera detallada y precisa cada uno de los aspectos del ciclo de vida de un conjunto de requerimientos.

Los procesos utilizados en la IR, varían dependiendo del dominio de aplicación, de la gente implicada y de la organización que desarrolla los requerimientos [52]. Cada una de estas actividades que conforman el Desarrollo de Requerimientos consisten en: Elicitación, Análisis, Especificación, Verificación, así como se muestra a continuación en la **Figura 1.2**



Figura 1.2 : Proceso de la ingeniería de requerimientos. **fuentes:** [54]

1.4.1.4.1. Elicitación

Según [52] la elicitación de los requerimientos es un paso del ciclo de vida de los requerimientos en el ciclo de vida del software. Consiste en la indagación de los requerimientos por medio de técnicas.

Uno de los aspectos más cruciales y desafiantes en el desarrollo de proyectos de software es la definición de los requerimientos para la ejecución de la propuesta, en esta

actividad el equipo de desarrollo conjuntamente con los interesados identifica, modulan y alcanzan los requerimientos del software a desplegar. El hallazgo de los requerimientos envuelve a definir el dominio, los servicios, los problemas y las necesidades de los usuarios respecto al software a construir [46].

Es un proceso de descubrimiento que se utiliza para presentar o producir información relevante para el proyecto o programa al extraer información de las partes interesadas y otras fuentes. Este proceso tiene como objetivo identificar las causas del problema comercial o las razones para abordar una oportunidad, así como la información que se utilizará para derivar un nivel suficiente de requisitos para permitir el desarrollo y la implementación de una solución [46].

Los requisitos también pueden surgir a medida que se realicen análisis y otras actividades dentro del proceso de requisitos [54] como se grafica en la Figura 1.3.

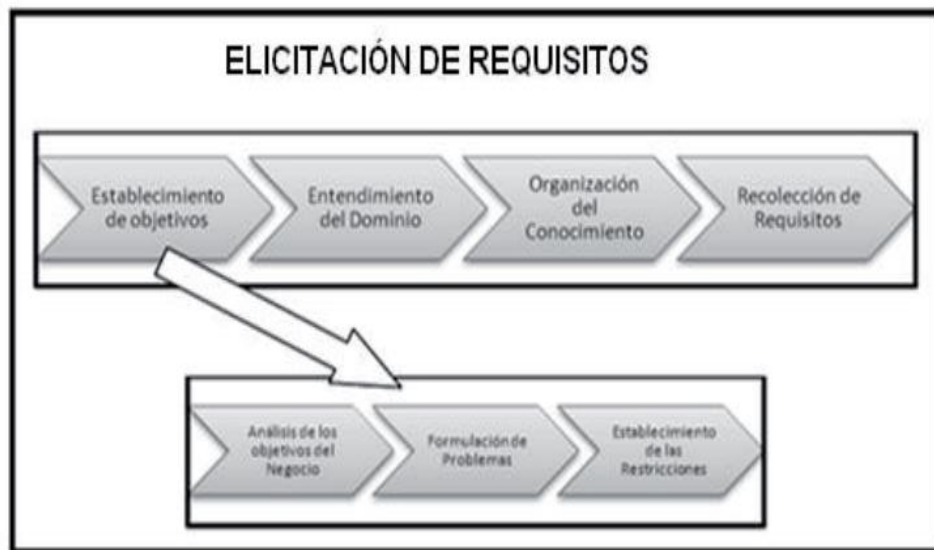


Figura 1.3: Proceso de descubrimiento de requerimientos **fuelle:**[54]

1.4.1.4.2. Análisis

Es el proceso mediante el cual obtiene una comprensión precisa de los requisitos, se analizan las necesidades identificadas por parte de los stakeholders de tal forma que se obtiene el Documento de definición de requisitos Validado [54]. A continuación, en la Figura 1.4 se grafica el proceso de análisis de requerimiento.



Figura 1.4: Proceso de análisis de requerimientos **fuentes:**[54]

El análisis de requisitos comprende las siguientes actividades:

- Analizar los requisitos funcionales (RF) recolectados.
 - Agrupar los requisitos funcionales recolectados y clasificarlos.
 - De la clasificación de los requisitos determinar: los que no son necesarios, son incompatibles entre sí, no son completos, no son factibles y los que están repetidos.
 - Aprobar la lista tentativa de requisitos funcionales definitivos por parte de los usuarios expertos en el dominio de la aplicación.
 - Estructurar el contenido de Documentos de Definición de Requisitos (DDR).
 - Elaborar el documento de Definición de Requisitos DDR con el listado de los requisitos funcionales; el cual debe estar aprobado por parte de los stakeholders.
- En esta fase el cuidado se debe tomar para describir requisitos con exactitud, suficientemente como para permitir que los requisitos sean validados, su implementación sea verificada, y sus costes estimados [55].

1.4.1.4.3. Especificación

Se refiere a la creación de un documento a su similar electrónico que esta sistemáticamente corregido, estimado y aceptado [56]. Desemparejar y documentar los requerimientos funcionales y no funcionales, identificar los atributos de calidad, requisitos importantes y restricciones, y validar que los requerimientos documentados sean completos y no sean ambiguos [58]. En esta fase se desarrollan 3 tipos de documentos: Documentos de definición de sistema, documento de requerimiento de sistema (SRS), documento de requerimiento de software. A continuación, se muestra en la figura 5 los tipos de documentos de especificación.

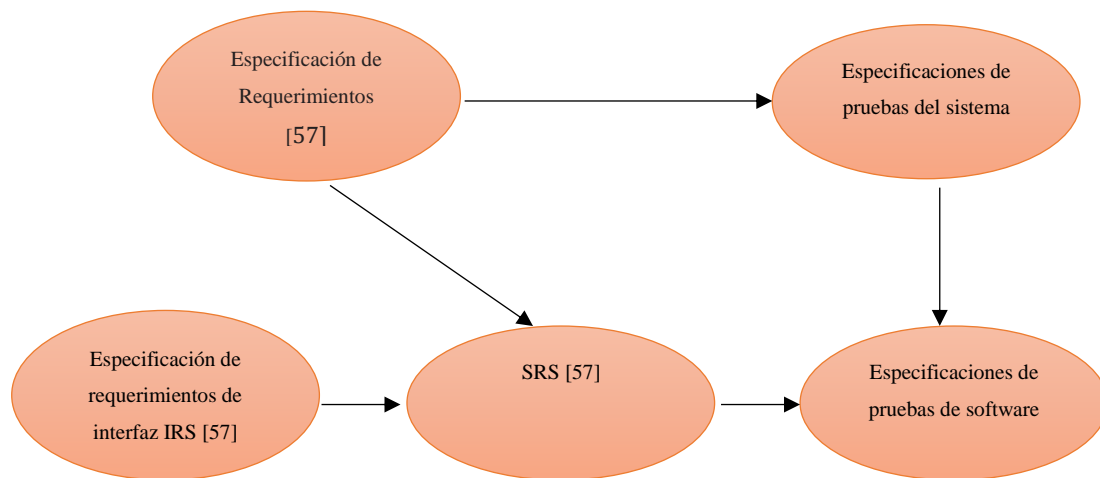


Figura 1.5: Documentos de fase de especificación de requerimientos **fuentes:** Elaboración propia, [55]

1.4.1.4 Validación

Los requisitos deben ser validados para asegurarse que el equipo de desarrollo de software haya entendido los requisitos; además se verifica que los documentos de los requisitos contemplan estándares, es comprensible, constante y finito. Con esta premisa se puede concluir que la validación de los requisitos es el proceso de examinar el documento de los requisitos para asegurarnos que este define el software correctamente [55].

1.4.1.5 El procesamiento de lenguaje natural y la ingeniería de requerimientos

En la Ingeniería de requerimientos, la educación es una de las fases iniciales del proceso de desarrollo de software, la cual implica obtener, analizar y especificar requisitos bajo la intervención de stakeholders, generando descripciones textuales o gráficas que plasman los conceptos más relevantes del dominio del interesado. En esta fase inicial ocurren dos sucesos que hacen complejo este proceso: la alta intervención de los stakeholders y el problema de comunicación usuario-analista. Por un lado, las técnicas de educación que se utilizan tradicionalmente requieren una muy alta intervención del interesado, debido a que en su mayoría se soportan en forma de diálogos y entrevistas [63].

El lenguaje natural (NL) es tradicionalmente la nota predominante para documentar, analizar y especificar los requerimientos de software [59]. NL se usa ampliamente no solo en las especificaciones, por ejemplo: historia de usuarios, casos de usos; sino durante todo el desarrollo del sistema. Los requerimientos escritos en NL son fáciles de escribir y entender, pero por otro lado son ambiguos, poco precisos [60].

La comunidad ha explorado temas tales como la identificación de defectos de calidad y ambigüedad, clasificación y agrupamiento de grandes colecciones de requisitos, extracción de abstracciones clave, generación de modelos y trazabilidad entre requisitos de NL y código [44]. Hasta hace poco, muchos de estos. Las aplicaciones de PNL se han limitado al mundo académico debido a la inaccesibilidad de las herramientas de PNL y la curva de aprendizaje empinada. Afortunadamente, los avances en el aprendizaje profundo y la disponibilidad de grandes cuerpos de NL han reducido significativamente las barreras de entrada al uso de PNL. Esto crea oportunidades sin precedentes para aplicar técnicas de PNL a la práctica de RE y para ayudar a analizar automáticamente los documentos relacionados con los requisitos.

1.4.1.5.1 El procesamiento de lenguaje natural.

Cuando se habla de procesar un lenguaje, se refiere a la traducción de la versión de un texto desde una lengua a otra [63]. Así, para ejecutar procesamiento de lenguaje natural (PLN) se requiere la transformación del texto en una representación semántica apta para razonar, tomar decisiones y ejecutar tareas específicas. Esta representación se logra por medio de procesos de Parsing o construcción de un árbol de análisis a partir de una gramática [64]. Si la gramática es sintáctica, por medio de dicho árbol de análisis se genera información sobre las categorías gramaticales de las palabras y la función sintáctica asociada (por ejemplo: identificación del sujeto, verbo, predicado y complementos).

1.4.1.6 Técnicas de la ingeniería de requerimientos

En la ingeniería de requerimiento existe 4 etapas: Elicitación, Análisis, Especificación y Validación [52], en cada uno de estas etapas se puede utilizar distintas 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 [15, 6, 16], que para esta investigación se considerara un criterio de importante de selección que representa a una evaluación empresarial como se muestra a continuación en la **Tabla 02**.

No	Técnica de Requerimientos	Etapas				Nivel (1-5)
		Elicitación	Análisis	Especificación	Validación	
1	Entrevista	X				4.39
2	Observación	X	X			1.89
3	Lluvia de ideas	X	X			4.02
4	Lenguaje Natural Estructurado (NLP)		X	X		4.08
5	JAD (Diseño de aplicación Conjunta)	X	X			4.37
6	Revisión por pares				X	4.17
7	Orientada a Objetivos		X			4.15
8	Etnografía	X				2.18
9	Modelo Conceptual		X	X		4.09
10	Especificación de requisitos blandos			X		4.05
11	Check-List (lista de verificación)	X	X	X	X	4.11
12	Especificación basada en ERD					4.23
13	Mapas Conceptuales Basados en Ontología	X	X	X		4.60
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

Tabla 1.1: Técnicas más utilizadas en las industrias de desarrollo de software en el proceso de la IR **fuente:** elaboración propia, [15,6, 16]

1.4.1.6.1 Entrevista

Es un enfoque metódico que se utiliza para obtener información de las partes interesadas al hacer preguntas relevantes y documentar las respuestas. Durante esta actividad, se plantean preguntas a los participantes del programa y del proyecto para identificar las funciones y capacidades que deberían existir en el producto final, el servicio o el resultado. Las entrevistas pueden ser estructuradas, donde las preguntas se preparan antes de una reunión, o desestructuradas, donde las preguntas se hacen de manera fluida en función de las respuestas a preguntas anteriores [62].

1.4.1.6.2 Observación

También conocida como "observación de empleos", proporciona una forma directa de ver a las personas en su entorno para ver cómo realizan sus trabajos o tareas y los procesos dentro de su entorno. Esta técnica es particularmente útil para obtener requisitos tácitos que son difíciles de verbalizar [54].

1.4.1.6.3 Lluvia de ideas

Es una técnica de grupo utilizada para generar múltiples ideas relacionadas con un tema en particular. Utiliza el aporte colectivo del grupo para comprender las diferentes perspectivas de un problema o solución y para construir sobre cada otra idea. Este método se basa en gran medida en la contribución de sus participantes para obtener información valiosa sobre un tema específico. Otros ejemplos de técnicas de creatividad grupal incluyen el uso de la técnica de grupo nominal, idea / mapeo mental, diagrama de afinidad y análisis de decisión multicriterio [54].

1.4.1.6.4 Diseño de aplicación conjunta (JAD)

Es una técnica de grupo utilizada para generar múltiples ideas relacionadas con un tema en particular. Utiliza el aporte colectivo del grupo para comprender las diferentes perspectivas de un problema o solución y para construir sobre cada otra idea. Este método se basa en gran medida en la contribución de sus participantes para obtener información valiosa sobre un tema específico. Otros ejemplos de técnicas de creatividad

grupales incluyen el uso de la técnica de grupo nominal, idea / mapeo mental, diagrama de afinidad y análisis de decisión multicriterio [15].

1.4.1.6.5 Modelamiento de lenguaje unificado (UML)

Es definido por sus creadores en [65] como un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Capta la información sobre la estructura estática y el comportamiento dinámico de un sistema, no obstante, carece de una semántica rigurosa, que permita utilizar razonadores automáticos sobre los modelos. Para realizar la comparación, este trabajo se enfocará en la representación de la información de estructura estática a través de los diagramas de clases, caso de uso de UML [65].

1.4.1.6.5.1 Diagrama de caso de uso

Es una representación gráfica de los casos de uso dentro del alcance de un sistema y cómo un actor (parte interesada) interactúa con ellos. Los casos de uso no muestran los requisitos, pero ayudan a organizar los requisitos para los esfuerzos de análisis y un documento de requisitos [52].

Según [2] se presentan los siguientes elementos de la especificación del diagrama de casos de uso:

Casos de uso: son las especificaciones de un conjunto de acciones realizadas por el actor sobre el sistema; es decir, son los pasos que describen de principio a fin un proceso.

Actores: son los roles que los usuarios desempeñan respecto del sistema y que emplean los casos de uso. Los actores pueden ser usuarios humanos u otros sistemas, que se comunican con el sistema que se requiere.

Relaciones: son interacciones que se pueden presentar entre casos de uso, entre actores y casos de uso y entre los actores. Las relaciones pueden ser de cuatro tipos:

Asociación: se establece entre los actores y casos de uso.

<<include>>: se presenta cuando una instancia del caso de uso origen incluye también el comportamiento descrito por el caso de uso destino; es decir, un caso de uso incluido describe un objetivo de bajo nivel de un caso de uso base [52].

<<extends>>: ocurre cuando el caso de uso origen extiende el comportamiento del caso de uso destino; en otras palabras, el caso de uso a extender invoca el caso de uso base bajo ciertas condiciones [52].

<<inheritance>>: se presenta cuando un caso de uso origen hereda la especificación del caso de uso destino y posiblemente la modifica y/o amplía. Este tipo de relación también se presenta entre los actores [52].

Escenario: es una secuencia específica de acciones que ilustra un comportamiento. Los escenarios se usan para ilustrar una interacción o la ejecución de una instancia de un caso de uso [52]. La representación gráfica de los elementos del diagrama de casos de uso se puede apreciar en la Figura 6.

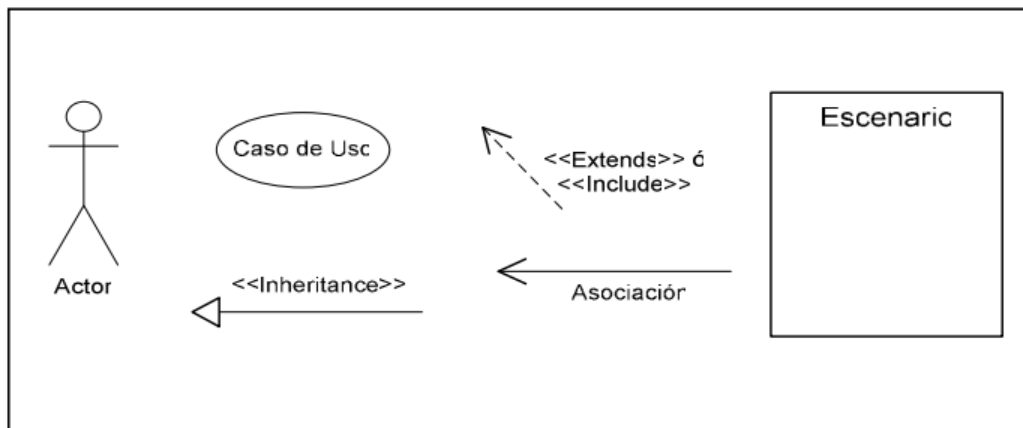


Figura 1.6: Elementos de los diagramas de casos de uso fuente: [52]

Las principales ventajas de utilizar el diagrama de casos de uso, según [52] son: La captura de los requisitos desde el punto de vista del usuario, lo cual permite el correcto desarrollo del sistema. La utilización de los casos de uso para Elicitación y análisis los requisitos funcionales que se recolectan en la fase de definición del ciclo de vida del software.

1.4.1.6.5.2 Diagrama de clase

Cuando se usan diagramas de clase UML para analizar requerimientos, los diagramas generalmente se usan para crear modelos conceptuales o modelos de dominio que representan entidades en requisitos. Los modelos comparten los requisitos en formas visuales y, por lo tanto, ayudan a los desarrolladores a comprender los tediosos detalles de los requisitos más fácilmente. Los modelos de dominio en forma de diagramas de clase UML se expanden durante el diseño y se transfieren al código fuente y a los productos finales.

1.4.1.7 Ontología

[37] Define los términos básicos y relaciones encerrando el vocabulario de un área, como las reglas para la composición de términos y relaciones para definir ampliaciones de un vocabulario. Esta tesis da los pasos a seguir para crear una ontología: asemejar términos básicos y relaciones entre los términos y las relaciones. [22] Así, una ontología no incluye solo los términos que son explícitamente definidos en ella, sino que también los términos que pueden ser deducidos usando reglas y expresándolas en mapas conceptuales. Uno de los principales objetivos de la ingeniería de requerimientos es obtener y comprender las necesidades de los clientes y luego reformular, estos necesitan proporcionar una imagen completa del proyecto de software para ayudar a las partes interesadas entender lo que hará el proyecto de software.

1.4.1.7.1 Mapa conceptual basado en ontología utilizando la IR.

[22] Produce una representación poderosa y el aprendizaje eficaz de los conocimientos de dominio. Por lo tanto, es muy útil y una técnica significativa en las actividades de ingeniería de requisitos. En otras palabras, se puede utilizar con fuerza en las primeras etapas de la ingeniería de software para eliminar los vacíos de conocimiento y comunicación entre todos los participantes. En otro aspecto, se proporcionan pequeñas contribuciones para el uso de mapas conceptuales y ontologías para apoyar la fase de ingeniería de requerimientos. A continuación, se muestra la figura 7 de un mapa conceptual basado en ontología.

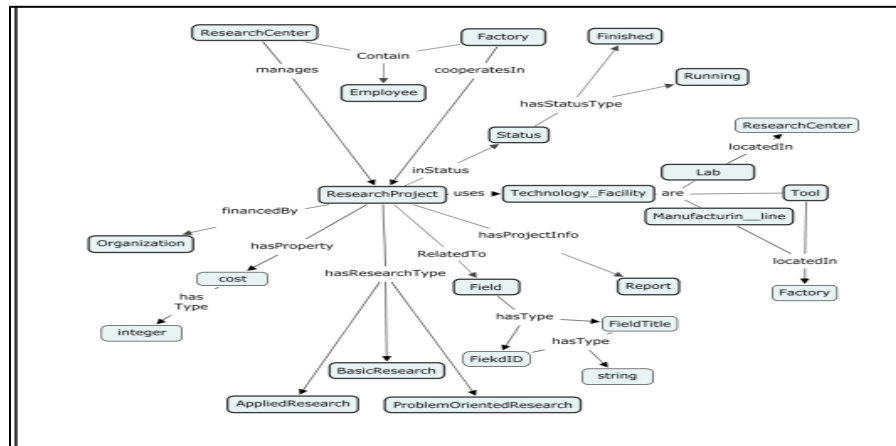


Figura 1.7: Mapa conceptual extraído de una ontología **fuentes:** [22]

[37] Se utiliza una ontología basada en mapas conceptuales para comprender, comunicar respecto a un dominio específico conceptualizado, para beneficiarse de su semántica, informalidad y fácil de entender, partiendo de los requisitos del dominio y los datos textuales relacionados con el mismo dominio existente. Por lo tanto, teniendo por finalidad proporcionar un proyecto de software, fácil y claro. La capa de ontología se puede transformar en mapa conceptual, según lo simétrico entre sus elementos. A continuación, se muestra en la figura 8 la arquitectura de la ontología conjuntamente con la IR.

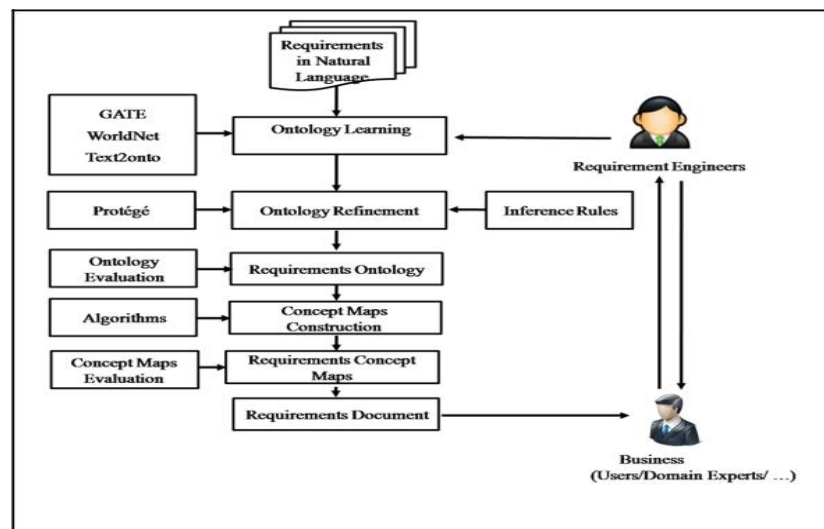


Figura 1.8: Arquitectura de la Ontología en la IR **fuentes:** [22]

II. MÉTODOS DE INVESTIGACIÓN

El procedimiento de revisión de la literatura científica del presente proyecto es un ajuste del método planteado por [62]. El cual tiene como objetivo proveer un marco para llevar a emprender el desarrollo de la de revisión de la literatura científica de un tema específico de forma concisa, valida y justificable. Teniendo en cuenta lo antes mencionado, se desarrolla el proyecto centrándose en la comprensión de los requerimientos de software mediante el lenguaje natural utilizando la ingeniería de requerimientos. Se debe agregar que se escogido por estudiar un rango de 9 años entre 2009 y 2018. A continuación, se realiza una representación del método mencionado en la figura.

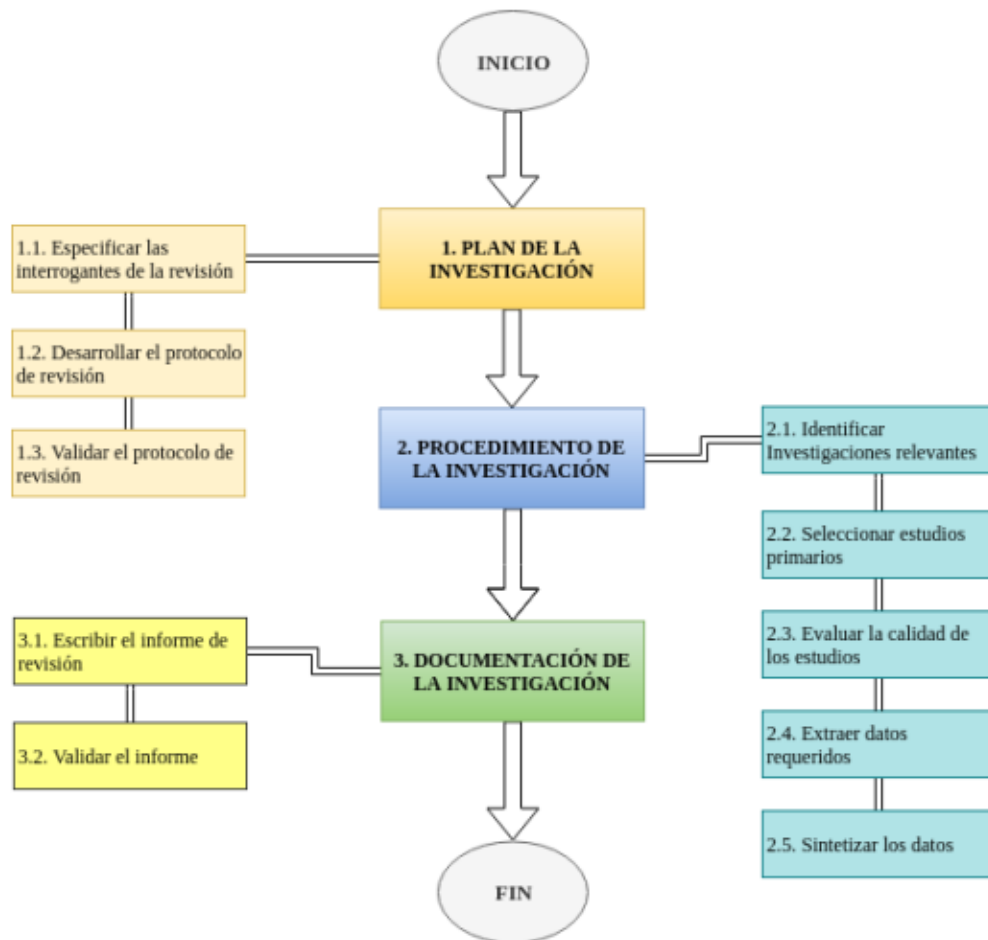


Figura 2.1: Método de revisión bibliográfico **fuentes:** Elaboración propia basado en [40].

2.1. Plan de la Investigación

Tiene como objetivo describir la planificación a llevar a cabo para realizar el procedimiento de revisión bibliográfica especificando a detalle los pasos a seguir para garantizar una firmeza de información a conocer. Este método refiere en los 3 pasos siguientes:

- Interrogantes de investigación que el estudio responderá
- Proceso que seguirá a la búsqueda de información
- Condición se aplique al seleccionar los artículos a incluir en la búsqueda de la información.

2.1.1. Interrogantes de la investigación

Las interrogantes que se plantea responder el estudio se describen a continuación.

PRE 1. ¿Cuánto material bibliográfico relacionado a la comprensión de lenguaje natural utilizando la visión de las técnicas de la ingeniería de requerimiento se ha publicado entre los años 2009 a 2018?

PRE 2. ¿Cuál es la técnica predominante para realizar la comprensión de requerimientos de lenguaje natural?

PRE 3. ¿Qué técnicas de ingeniería de requerimientos existen para realizar la comprensión del lenguaje natural de los requerimientos?

PRE 4. ¿En que convergen las conclusiones relacionadas a la comprensión de requerimientos en lenguaje natural?

2.1.2. Protocolo de Revisión

El protocolo de investigación define un conjunto de regla en las cuales se detalla el proceso de revisión de la información relacionada a la comprensión de los requerimientos de software mediante el lenguaje natural de modo de especificar el cómo se extrajo la indagación y por qué. Se debe agregar que para este proyecto el primer paso es clasificar las bases de datos en las cuales se realicen la búsqueda de los artículos relacionados, para ello se usó la plataforma SCImago Journal & Country Rank la cual incluye revistas y sus indicadores científicos a partir de la información contenida en la base de datos Scopus. Estos indicadores se utilizan para evaluar y analizar publicaciones

científicas. Se ha tomado los diferentes rankings de acuerdo a las bases de datos más relevantes que se necesita para esta investigación teniendo en cuenta que se encuentran en diferentes áreas de búsqueda, según el filtro SCImago al 2018, como se especifica en la Figura 2.2.

Si bien es cierto cada conjunto de datos tiene su propia configuración de búsqueda esta sigue patrones similares, pero esto no impide crear reglas y criterios de búsqueda para cada una de ellas. Lo dicho hasta aquí supone que se recuperó investigaciones relacionadas a las interrogantes planteadas [2.1.1], también se tiene en cuenta que el proceso de clasificación de frutas es considerado uno de los más importantes para garantizar la calidad del producto existe un amplio material bibliográfico, se debe aclarar que este proyecto solo considera artículos publicados en revistas según la base de datos antes mencionada. Además, para garantizar la revisión del material más reciente se tiene en cuenta el rango entre el 2009 y 2018. Por otra parte, se diseña la regla de búsqueda que se ajuste al objetivo real de la investigación según las especificaciones de las bases de datos las cuales producirán los mejores resultados efectivos y completos, luego se opta por elegir las propuestas que mejor se ajusten.

Posición	Title	Type	↓ SJR	H index	Total Docs. (2018)	Total Docs. (3years)	Total Refs. (2018)	Total Cites (3years)	Citable Docs. (3years)	Cites / Doc. (2years)	Ref. / Doc. (2018)	
FILTRO CIENCIAS DE LA COMPUTACION												
5	SoftwareX 	journal	4.539 Q1	10	77	98	1665	1130	97	2.14	21.62	
145	IEEE Transactions on Intelligent Transportation Systems	journal	1.412 Q1	112	571	911	13725	7211	900	7.42	24.04	
401	IEEE Transactions on Software Engineering	journal	0.732 Q1	151	166	197	3914	1260	181	6.52	23.58	
519	IEEE Access 	journal	0.609 Q1	56	6159	3908	239767	19132	3870	4.64	38.93	
FILTRO INGENIERIA DE CONTROL Y SISTEMAS												
66	IEEE Systems Journal	journal	0.824 Q2	54	547	504	10019	2574	480	5.28	18.32	

Figura 2.2: Resultado del filtro de la plataforma SCImago Journal & Country Rank
Fuente: Elaboración Propia

2.1.3. Validar Protocolo de revisión

Se ha validado el protocolo de revisión. Las búsquedas iniciales condujeron a revisiones del protocolo y las preguntas de investigación. Además, la validación será discutida más tarde utilizando un contenido proceso de análisis de las palabras clave de los incluidos en los artículos.

2.2. Documentación de la investigación

2.2.1. Identificar las investigaciones relevantes

En este punto se buscan las propuestas más relevantes que proporcionan la mayor información con el fin de responder las interrogantes planteadas [2.1.1]. Cabe mencionar que, la propuesta de revisión se centra estrictamente en la comprensión de requerimientos de software mediante el lenguaje natural por ello que se ha diseñado una regla genérica de búsqueda utilizando “and” booleano para unir los principales términos “, Procesamiento de lenguaje natural, ingeniería de requerimientos”, “natural lenguaje”, “requirements” como se muestra en la Tabla 2.1. Además, se ha revisado la base de datos IEEE, finalmente se determinó que revistas serán la fuente principal de esta revisión bibliográfica para garantizar la calidad de este estudio en la Figura 2.3 la cual expone el resultado crudo del filtro.

Nº	Base de Datos	Regla de Búsqueda	Resultados
1	IEEE Transactions on Software Engineering	(((natural language processing AND techniques AND requirements engineering AND natural lenguaje AND Requirements))	43
2	IEEE Systems Journal		
3	Software		
4	IEE Access		
5	IEEE Transactions on Intelligent Transportation Systems		

Tabla 2.1: Búsqueda de base de datos y resultados. **Fuente:** Elaboración propia.

IEEE.org | IEEE Xplore Digital Library | IEEE-SA | IEEE Spectrum | More Sites Cart (0) | Create Account | Personal Sign In

IEEE Xplore[®] Digital Library Institutional Sign In **IEEE**

Browse ▾ My Settings ▾ Get Help ▾ Subscribe

All ▾ Enter keywords or phrases (Note: Searches metadata only by default. A search for 'smart grid' = 'smart AND grid') Advanced Search | Other Search Options ▾

Search within results Per Page: 25 ▾ | Export ▾ | Set Search Alerts ▾ | Search History

Displaying results 1-25 of 43 for **(natural language processing and Requirements Engineering)** ✕

▼ Filters Applied: Journals ✕ 2008 - 2019 ✕

Conferences (533)
 Courses (26)
 Magazines (11)
 Early Access Articles (7)

Books (1)
 Standards (1)

Show
 All Results
 Open Access

Year ▾
 Single Year | Range
 2010 | 2019
 From: 2010 | To: 2019

Author ▾
Affiliation ▾
Publication Title ▾
Publisher ▾
Supplemental Items ▾
Index Terms ▾

Select All on Page
 Sort By: Relevance ▾

Automated Checking of Conformance to Requirements Templates Using Natural Language Processing
 Chetan Arora ; Mehrdad Sabetzadeh ; Lionel Briand ; Frank Zimmer
 IEEE Transactions on Software Engineering
 Year: 2015 | Volume: 41, Issue: 10 | Journal Article | Publisher: IEEE
 Cited by: Papers (30)
 ▶ Abstract (3559 Kb)

A Process Model for Standardization and Increase in the Requirements Quality
 Jose Andre Dorigan ; Rodolfo Miranda de Barros
 IEEE Latin America Transactions
 Year: 2014 | Volume: 12, Issue: 8 | Journal Article | Publisher: IEEE
 ▶ Abstract (877 Kb)

Empirical Principles and an Industrial Case Study in Retrieving Equivalent Requirements via Natural Language Processing Techniques
 Davide Falessi ; Giovanni Carlone ; Gerardo Carfora
 IEEE Transactions on Software Engineering
 Year: 2013 | Volume: 39, Issue: 1 | Journal Article | Publisher: IEEE
 Cited by: Papers (44)
 ▶ Abstract (3881 Kb)

Using Natural Language Processing to Automatically Detect Self-Admitted Technical Debt
 Everton da Silva Maidonado ; Emad Shihab ; Nikolaos Tzantalis
 IEEE Transactions on Software Engineering
 Year: 2017 | Volume: 43, Issue: 11 | Journal Article | Publisher: IEEE
 Cited by: Papers (9)
 ▶ Abstract (857 Kb)

Supporting Process Model Validation through Natural Language Generation
 Henrik Leopold ; Jan Mendling ; Artem Polyvyanyy
 IEEE Transactions on Software Engineering
 Year: 2014 | Volume: 40, Issue: 6 | Journal Article | Publisher: IEEE
 Cited by: Papers (32)
 ▶ Abstract (1317 Kb)

A Tool-Supported Methodology for Validation and Refinement of Early-Stage Domain Models
 Marco Aullì ; Antonia Bertolino ; Guglielmo De Angelis ; Davide Di Ruscio ; Alessio Di Sandro
 IEEE Transactions on Software Engineering
 Year: 2016 | Volume: 42, Issue: 1 | Journal Article | Publisher: IEEE
 Cited by: Papers (1)

Need Full-Text
 access to IEEE Xplore for your organization?
[REQUEST A FREE TRIAL](#) ▶

IEEE

MARS SPACE COLONY
 A GAME OF STANDARDIZATION

A Unique Team Building Activity

Experience technical standards development in this role play simulation activity.

[LEARN MORE](#)

Figura 2.3: Resultado crudo de la Búsqueda de IEEE. Fuente: IEEE Xplore Digital Library

2.2.2. Validar la información

La selección de los estudios incluidos en el proyecto se lleva a cabo gracias a los distintos indicadores (palabras clave, revistas, resumen, cantidad de referencias, número de citas) comparadas a partir del resultado de aplicar la búsqueda utilizando la regla planteada anteriormente, para tomar los estudios que describan de forma más ecuánime posible el objetivo de la revisión. A continuación, se describe las tendencias de los estudios filtrados haciendo énfasis en el año de la publicación, tipo de técnicas utilizadas en la ingeniería de requerimientos, característica de las técnicas, porcentaje de precisión, que se obtuvo en las propuestas planteadas y las referencias a estas.

Nº	Año	Técnicas	Característica	Eficiencia	Ref.
1	2009	NLP - UML	Diagramas de colaboración	-	[1]
2	2009	UML - NLP	Regla Heurística	-	[2]
3	2009	UML	Systems Modeling Language	-	[3]
4	2009	NLP - UML	Naïve Bayes, Caso de uso	-	[49]
5	2010	UML – NLP	Diagramas de clases (DC)	-	[4]
6	2010	UML-NLP-XML	Diagrama de colaboración, Diagrama de clase	-	[5]
7	2010	Ontología - NLP	Algoritmo de segmento, ontologías de dominio	79%, 81%, 79%	[50]
8	2010	Ontología – NLP - IA	OWL, AutoBayes y AutoFilter	-	[7]
9	2011	UML - NLP	Casos de uso, Petri	-	[23]
10	2011	UML- NL	Diagrama de Clase, CIM	-	[24]
11	2011	ISO / IEC / IEEE 29148	Artefactos Análisis, Especificación	80%	[25]
12	2011	NLP - UML	Reglas semánticas de negocio y vocabulario	85.76%,80.12%, 82.94%	[41]
13	2012	UML	Casos de uso	-	[39]
14	2013	Ontología - NLP	Modelo de dominio (MD)	82%	[12]
15	2013	UML	MD, Modelo de análisis, Diseño orientado a Objeto, NLTK Payton	81,82%	[11]
16	2013	Modelo MeRinde - IR	SPEM 2.0	-	[10]
17	2014	UML- NLP	clustering semántico, reglas enriquecidas semánticas, DC	50%,60%,70%	[18]
18	2014	Metodología ER	Modelo de requisitos	-	[42]

19	2014	UML	Marte, SysML	-	[43]
20	2015	NLP - UML	Deep Learning	-	[27], [21]
21	2015	Modelo de Proceso de Conceptualización	Modelo conceptual	-	[46]
22	2015	UML- MBD	Diagrama de actividad, Caso de uso, Diagrama de actividad	-	[47]
23	2016	Modelos MDE - UML	NL, DC, DA, Caso de uso	88%, 90%	[68]
24	2016	IA-NLP-UML-Ontología	Inspección, patrones gráficos, Heurísticas	-	[13]
25	2017	UML- NLP	Diagrama de actividad, modelos i*, KAOS	95%	[14]
26	2017	UML-NLP-Ontología	Diagrama de dominio, Mapa Conceptual	82%,90%	[30]
27	2017	UML - NLP	Diagrama de clase	85%, 90%,77%, 78%	[19]
28	2017	Metodología TEC	Matriz de decisión	95%, 97%	[20]
29	2017	UML-Ontología	Text2Onto, Gate, WorldNet Treetagger, Diagrama de clase, Mapa Conceptual, Kappa	73%,56%, 40%	[22]
30	2018	UML	Modelo Conceptual, Casos de uso.	85%	[34]
31	2018	NLP	Deep Learning, CNN	80%, 78,5%	[21]
32	2018	NLP	Weka, Matriz de confusión, Kappa	95%,90%,85%	[35]
33	2018	UML	Análisis de Procesos, Modelos conceptuales, Diseño BPM	-	[36]
34	2018	NLP	NLP Tool-kit, Apache OpenNLP	-	[44]
35	2018	NLP – Ontología	Rational DOORS, Jira, GitLab	-	[38]
36	2018	NLP	NPL tools	-	[45]
37	2018	NLP-UML	Historias de Usuario (AQUSA), extracción de modelo conceptual	72%, 93%	[48]

TABLA 2.2: RESUMEN DE LAS PROPUESTAS PLANTEADAS DESDE EL 2009 – 2018. **Fuente:** Elaboración propia.

2.2.3. Evaluar la calidad de los datos

En este paso, la calidad de investigación se abordó con el criterio de inclusión de la revisión integradora. La inclusión criterio para esta revisión integral de la literatura incluyó artículos de revistas que realizaron investigaciones relacionadas a la comprensión de requerimientos de software mediante el lenguaje natural (Regla (‘natural language processing AND Requirements Engineering’ AND natural lenguaje AND requirements)). Además, la evaluación del enfoque de investigación del documento, se evaluó el impacto global y el valor del H -índice de las revistas. El valor es “El número de citas ponderadas por año seleccionado por los documentos publicados en la revista seleccionada en tres años anteriores” [60]. El H-índice “cuantifica tanto la productividad científica de la revista como la productividad científica impacto” [60]. En la Tabla 2.3 se muestra la SCImago Journal Rank (impacto #) y el H-índice de cada documento incluido en este estudio. Cabe mencionar que, la mayoría de los documentos incluidos en este estudio fueron publicado en revistas de impacto en las que el valor medio de impacto es de 0.64 y la media de El H-índice es 93.88.

Artículo	Revista	Impact #	H-índice
1	IEEE Transactions on Software Engineering	0.73	151
2	IEEE Latin America Transactions	0.34	19
3	IEEE Access	0.61	56
4	IET Software	0.17	40
5	IEEE Transactions on Audio, Speech, and Language Processing	0.93	98
6	IEEE Systems Journal	0.82	54
7	IEEE Transactions on Intelligent Transportation Systems	1.41	112
8	IEEE Transactions on Very Large-Scale Integration (VLSI) Systems	0.41	95
9	IEEE Transactions on Information Forensics and Security	1.36	95
10	IEEE Transactions on Computers	0.83	110
11	IEEE Transactions on Geoscience and Remote Sensing	2.76	216
12	IEEE Transactions on Automation Science and Engineering	1.47	69
13	IEEE Geoscience and Remote Sensing Letters	1.52	89
14	IEEE Transactions on Knowledge and Data Engineering	1.14	148
15	IEEE Transactions on Industrial Informatics	1.68	100
16	Proceedings of the IEEE	1.59	50
#	PROMEDIOS	1.11	93.88

TABLA 2.3: RESUMEN DE IMPACTO DE LAS REVISTAS (2009- 2018). **Fuente:** Elaboración propia.

2.2.4. Extraer datos requeridos

Para responder a las preguntas de investigación, se extrajeron los siguientes datos de cada estudio:

1. Las palabras clave que se utilizarán en un análisis de contenido para determinar en última instancia las categorías de investigación en comunicación (la taxonomía)
2. Tipo de datos recogidos: cualitativos, cuantitativo, encuesta, revisión de la literatura.
3. Suficiente información para determinar el resultado principal.
4. La información de la revista para permitirnos determinar su valor de impacto y el valor del H-índice.

2.2.5. Sintetizar los datos

Los resultados de la síntesis de datos se utilizan para determinar los temas principales dentro de la investigación relacionada a la clasificación automatizada de frutas y para categorizar cada artículo seleccionado y así poder determinar las categorías de investigación. Se realizó un análisis del contenido de las palabras clave de cada artículo haciendo énfasis en el significado de cada palabra o concepto el cual va más allá de simplemente contar palabras para examinarlas. Se debe aclarar que para realizar este análisis se basa en la valoración de los términos controlados por INSPEC4 la cual es una importante base de datos de indexación de literatura científica y técnica, publicada por el Instituto de Ingeniería y Tecnología.

El resultado de esta evaluación se muestra Tabla 2.4, la cual expone el título de la publicación, autor, año de la publicación, palabras clave del autor, términos controlados por IEEE, términos controlados y no controlados por INSPEC número de citas y referencias de cada publicación. Además, en la Figura 2.4 se muestra una comparación entre el número de citas y la cantidad de veces que ha sido referenciado cada artículo revisado.

Tabla 2.4: VALORACION DE LAS PALABRAS CLAVE DE ACUERDO A INSPEC

N°	Título del documento	Autores	Años	Palabras clave del autor	Termino de IEEE	Términos controlados por INSPEC	Términos controlados no por INSPEC	# de citas	# de Ref.
1	A controlled natural language approach for integrating requirements and model-driven engineering	David de Almeida Ferreira; Alberto Rodrigues da Silva	2009	Requisitos de software, Herramientas y métodos de ingeniería de software	Lenguajes naturales, Ingeniería dirigida por modelos, Calidad de software, Ingeniería de software, Gestión de ingeniería, Automatización, Productividad, Ingeniería de diseño, Control automático, Procesamiento natural del lenguaje	especificación formal, verificación formal, procesamiento del lenguaje natural, Ingeniería de software	especificación formal, verificación formal, procesamiento del lenguaje natural, Ingeniería de software	6	22
2	An Automated Tool for Generating UML Models from Natural Language Requirements	Deva Kumar Deeptimahanti; Muhammad Ali Babar	2009	Ingeniería de requisitos, Procesamiento de lenguaje natural, Lenguaje de modelado unificado	Lenguaje de modelado unificado, Lenguajes naturales, Automatización, Herramientas de colaboración, Procesamiento del lenguaje natural, Humanos, Ingeniería de software, Software colaborativo, Colaboración internacional, Visualización	lingüística computacional, groupware, procesamiento del lenguaje natural, Lenguaje de modelado unificado	herramienta automatizada, requisitos de lenguaje natural, Generador de modelos UML a partir del análisis de requisitos, diagrama de caso de uso, modelo de clase de análisis, diagrama de colaboración, procesamiento del lenguaje natural, PNL, reglas de reconstrucción sintáctica, Analizador XMI, Archivos XMI	11	11
3	Conversion of Pre-Conceptual Schema into Use Case Diagrams by Using Atom3	Carlos m. Zapata; Paula a. Tamayo; Fernando Arango	2009	Meta modelamiento, diagrama de casos de uso, esquemas preconceptuales	Meta modelamiento, diagrama de casos de uso, esquemas preconceptuales	procesamiento del lenguaje natural, Lenguaje de modelado unificado	diagrama de caso de uso, modelo de clase de análisis, diagrama de colaboración,	5	8

4	Automated Identification of LTL Patterns in Natural Language Requirements	Allen P; Nikora Galen Balcom	2009	análisis de requisitos, requisitos temporales, aprendizaje automático, procesamiento natural del lenguaje	Lenguajes naturales, Propulsión, Automatas, Análisis de patrones, Procesamiento del lenguaje natural, automático, Especificaciones, Programación, frecuencia, Fiabilidad del software	especificación formal, aprendizaje (inteligencia artificial), procesamiento del lenguaje natural, clasificación de patrones, lógica temporal	Patrones LTL, identificación automatizada, análisis de requisitos temporales en lenguaje natural, consistencia automatizada de cheques, propiedades de corrección, subconjunto de requisitos, técnicas de aprendizaje automático, Misión JPL, clasificación de requisitos	5	18
5	Text-based Requirements preprocessing using nature language processing techniques	Huafeng Chen; Keqing He; Peng Liang; Rong Li	2010	requisitos, procesamiento, ontología de dominio, parte interesada no técnica, wiki	Ontologías, Computación distribuida, Helio, software, Web, Aceleración, cara, Colaboración, Sólidos, Minería de datos, Lenguaje de modelado unificado, Procesamiento del lenguaje natural	procesamiento distribuido, especificación formal, procesamiento del lenguaje natural, Web semántica, análisis de texto	preprocesamiento de requisitos basado en texto, procesamiento del lenguaje natural, ambiente distribuido, parte interesada no técnica, declaración de requisitos, documento de requisitos de texto libre, ingeniería de requisitos, requisito de nivel del sistema, requisito de nivel de instancia, expresión de requerimiento, Web semántica, segmento semántico mejorado, patrón de oración de dominio, extracción de objetivos	2	12

6	Anaphora resolution system for natural language requirements	Parque Ki Seon; Dong Un An; Yong Seok Lee	2010	Obtención de requisitos, Resolución de la anáfora, Decisión antecedente, Procesamiento natural del lenguaje	Lenguajes naturales, Procesamiento del lenguaje natural, Especificaciones formales, Análisis de información, Costos, Análisis de texto, Diseño para experimentos, Recuperación de información, Arquitectura de la computadora, Sociedades	especificación formal, procesamiento del lenguaje natural, clasificación de patrones	Sistema de resolución de anáfora, documento de requisitos de lenguaje natural, especificaciones formales, costo de desarrollo del sistema, procesamiento del lenguaje natural, clasificación de pronombres, pronombre personal, pronombre demostrativo	1	9
7	Class Diagram Extraction from Textual Requirements Using Natural Language Processing (NLP) Techniques	Mohd Ibrahim, Rodina Ahmad	2010	Procesamiento de lenguaje natural (PNL), Ontología de dominio, Diagrama de clase UML	Procesamiento del lenguaje natural, Ontologías, Lenguajes naturales, Lenguaje de modelado unificado, Minería de datos, Modelado orientado a objetos, Investigación y desarrollo, Ingeniería de software, Correo electrónico, Automatización	especificación formal, procesamiento del lenguaje natural, ontologías (inteligencia artificial)	requisitos textuales, procesamiento del lenguaje natural, PNL, requisitos de lenguaje natural, técnicas de ontología de dominio, análisis de requisitos y extracción de diagramas de clases, RAZA	8	14

8	From natural language requirements to requirement ontologies	Rong Li, Keqing He, Huafeng Chen	2010	requisito de ontología, procesamiento del lenguaje natural, Modelo superior generalizado, WordNet, anotación de concepto	Lenguajes naturales, Ontología, Bases de datos, Ingeniería de software, La informática, Puentes, Cognición, Ciencia cognitiva, Laboratorios, Diccionarios	sistemas de gestión de bases de datos, la lingüística, procesamiento del lenguaje natural, ontologías (inteligencia artificial), análisis de sistemas, análisis de texto	modelo superior generalizado, ontología lingüísticamente motivada, WordNet, base de datos léxica utilizada en todo el mundo, Idioma Inglés, procesamiento del lenguaje natural	1	11
9	Casos de uso y comentarios en el análisis de requisitos funcionales	Jiandong Xu; Tong Li; Zhongwen Xie; Tilei Gao	2011	Caso de uso; Comentarios, Requisito funcional; Hoare Logic; Petri Net	Negocios; Sistemas de software; Lenguaje de modelado unificado; Instituciones educativas; Modelos analíticos; Modelos de adaptación	La lógica formal; especificación formal; verificación formal; Redes de Petri, calidad de software	Proceso de software, calidad del producto, modelo de casos de uso de tres niveles, Hoare Logic, Red de Petri, análisis de requisitos funcionales	1	12
10	Una transformación ATL de modelos de requisitos de lenguaje natural a modelos comerciales de un proyecto MDA	Narayan Debnath; María Carmen Leonardi; Marcela Ridaio; María Virginia Mauco; Laura Felice; Germán Montejano; Daniel Riesco	2011	MDA; Transformaciones ATL; CIM; Modelos de requisitos de lenguaje natural	Lenguaje de modelado unificado; Empresas; Producción; Modelado computacional; Contratos; Fabricación integrada por computadora; Lenguajes naturales	reingeniería de procesos de negocio; arquitectura de software; Lenguaje de modelado unificado	Transformación ATL; modelo de requisitos de lenguaje natural; modelo de proceso de negocio; Proyecto MDA, arquitectura basada en modelos; marco de desarrollo de software; modelo independiente de la computadora; Diagrama de clase UML, Lenguaje de modelado unificado; Lenguaje de transformación ATLAS	1	13

11	El marco de ingeniería de requisitos basado en ISO 29148: 2011 y el marco de modelado de múltiples vistas	Debby Selvyanti; Yoanes Bandung	2011	Ingeniería de requisitos, marco de ingeniería de requisitos, marco de ingeniería de requisitos para agencias gubernamentales	Ingeniería de requisitos, Software, Normas ISO, Normas IEC, gobierno, Lenguaje de modelado unificado, Procesamiento del lenguaje natural	especificación formal, Normas IEC, Estándares IEEE, Normas ISO, gestión de proyectos, gestión de desarrollo de software, análisis de sistemas	marco de ingeniería de requisitos, marco de modelado multivista, agencias gubernamentales, proyectos de desarrollo de software fallan, estándar de desarrollo de software, área de requisitos de software, ingenieros de requisitos, documentos de requisitos, ISO 29148: 2011, herramienta de comunicación	0	19
12	Minimizar la ambigüedad en la especificación de requisitos de software de lenguaje natural	Ashfa Umer; Imran Sarwar Bajwa	2011	Especificaciones de requisitos de software; procesamiento de lenguaje natural; SBVR	Software, Negocios, Semántica, vocabulario, Horarios, Lenguajes naturales, Sintácticos, Lenguaje de modelado unificado, Procesamiento del lenguaje natural	especificación formal, java, procesamiento del lenguaje natural, arquitectura de software, vocabulario	especificación de requisitos de software de lenguaje natural, ingeniero de software, diseño de software formal, procesamiento de máquina, Representación de NL, modelo de software, vocabulario de negocios, Especificación de requisitos de software en inglés, Representación controlada basada en SBVR, fundamento matemático, Implementación basada en Java	6	15

13	Requisitos de modelado de soporte en el idioma malayo usando casos de uso esenciales	Massila Kamalrudin ; John Grundy ; John Hosking	2012	Requisitos de lenguaje natural, Casos de uso esenciales, Ingeniería de requisitos, Ingeniería de ida y vuelta, Modelado Humancentric	Resúmenes, Lenguajes naturales, Bibliotecas, Software, Conferencias, Modelado computacional, Instituciones educacionales, Lenguaje de modelado unificado, Procesamiento del lenguaje natural	interfaces de lenguaje natural, procesamiento del lenguaje natural, análisis de texto	modelado de requisitos, ambigüedad del lenguaje natural, Texto en inglés, modelado semiformal, refinamiento de ida y vuelta, enfoque centrado en el ser humano, ingeniería de requisitos, conjunto de herramientas de casos de uso esenciales, modelado de interacción esencial, Extracción de lenguaje natural malayo	1	10
14	Mapas conceptuales basados en ontologías para ingeniería de software	Karama Ali Mohamed ; Marwa Salah Farhan ; Mahmoud Mohamed Ahmed Abd Elatif	2013	ontología, ingeniería de software, mapas conceptuales	Software, Teoría de la complejidad, historia, Negocios, Cara, Lenguaje de modelado unificado, Procesamiento del lenguaje natural	ontologías (inteligencia artificial), Ingeniería de software	mapas conceptuales basados en ontología, representación formal, conceptualización compartida, campo de ingeniería de software, proyecto de software	1	37
15	Análisis de calidad de requisitos de lenguaje natural basado en modelos de dominio de negocios	KM Annervaz ; Vikrant Kaulgud ; Shubhashis Sengupta; Milind Savagaonkar	2013	Ingeniería de requisitos, Ontología, Procesamiento de lenguaje natural, Modelado de dominio comercial	Negocios, Ontologías, Lenguajes naturales, Modelos analíticos, Portales, OWL, Normas, Lenguaje de modelado unificado, Procesamiento del lenguaje natural	procesamiento de datos comerciales, procesamiento del lenguaje natural, ontologías (inteligencia artificial), ingeniería de software, análisis de sistemas	análisis de calidad de requisitos de lenguaje natural, modelos de dominio comercial, ingeniería de software, proyecto de entrega de software, conocimiento del dominio, procesamiento del lenguaje natural, técnicas basadas en ontología, comprensión de requisitos, analistas de negocios, estándar de dominio	1	21

16	Una comparación de técnicas de PNL semánticas para analizar casos de uso	Alejandro Rago; Claudia Marcos; J. Andrés Diaz-Pace	2014	Procesamiento del lenguaje natural, Software	Procesamiento del lenguaje natural, Software, Acoplamientos, Semántica, Compuestos de silicio, Manuales, Inteligencia artificial	procesamiento del lenguaje natural, Inteligencia artificial	software, conocimiento del dominio, procesamiento del lenguaje natural	0	15
17	Una metodología de ingeniería de requisitos que combina modelos y lenguaje natural controlado.	Markus Fockel; Jörg Holtmann	2014		Contexto, Lenguaje de modelado unificado, Modelado de contexto, Documentación, Lenguajes naturales, Modelos analíticos, Sistemas embebidos	sistemas embebidos, especificación formal, procesamiento del lenguaje natural, documentación del sistema	transformación bidireccional del modelo de varios pasos, Documentación basada en NL, documentación basada en modelos, metodología RE basada en modelos, Requisitos basados en NL, especificaciones de requisitos, técnicas de análisis automático, comprensión de requisitos, sistemas embebidos intensivos en software, lenguaje natural controlado, metodología de ingeniería de requisitos	7	30
18	Integrando UML, MARTE y sysml para mejorar la especificación de requisitos y la trazabilidad en el dominio incorporado	Milena Rota Sena Marques; Eliane Siegert; Lisane Brisolará	2014	Ingeniería de requisitos, Software embebido, Sistemas embebidos, Gestión de requisitos, Trazabilidad	Lenguaje de modelado unificado, ruedas, El tiempo, Sensores, Software incorporado, Modelado orientado a objetos	sistemas embebidos, especificación formal, Lenguaje de modelado unificado	Notación UML, Notación MARTE, Notación SysML, Lenguaje de modelado unificado, especificación de requisitos, trazabilidad de requisitos, dominio de software incorporado, enfoque de ingeniería de requisitos basado en modelos	10	20

19	NLP-KAOS para sistemas de obtención de objetivos: estudio de caso del sistema de medición inteligente	Erik Casagrande; Selamawit Woldeamlak; Wei Lee Woon; HH Zeineldin; Davor Svetinovic	2014	Ingeniería de requisitos, obtención de objetivos, PNL, minería de datos, bibliometría	Minería de datos, Taxonomía, Ontologías, Modelos de datos, Procesamiento del lenguaje natural, Resúmenes, Recopilación de datos, Lenguaje de modelado unificado, Procesamiento del lenguaje natural	minería de datos, especificación formal, procesamiento del lenguaje natural, contadores inteligentes	NLP-KAOS, sistemas de obtención de objetivos, estudio de caso del sistema de medición inteligente, método computacional, procesamiento del lenguaje natural, técnicas de minería de textos, ingenieros de requisitos, documentos textuales, Enfoque de obtención de objetivos basado en PNL, Método de ingeniería de requisitos orientado a objetivos KAOS, relaciones jerárquicas, sistemas de energía, rejillas inteligentes, frases relacionadas con objetivos, publicaciones de investigación, sistemas complejos	6	57
20	Apoyar la validación del modelo de proceso mediante la generación de lenguaje natural	Henrik Leopold; Jan Mendling; Artem Polyvyanyy	2014	Validación del modelo de proceso de negocio, generación de texto en lenguaje natural, verbalización	Lenguaje de modelado unificado, Lenguajes naturales, Negocios, Modelos analíticos, Modelos de adaptación, Modelado de contexto, Contexto	sistemas de información, procesamiento natural del lenguaje	validación del modelo de proceso, complejidad lingüística, complejidad de la estructura, generación de texto en lenguaje natural, generación de texto de aspecto natural, técnicas de verbalización, integridad del modelo de proceso, corrección del modelo de proceso, modelos de procesos de negocio, generación de lenguaje natural	6	110

21	Verificación automatizada de conformidad con plantillas de requisitos mediante procesamiento de lenguaje natural	Chetan Arora; Mehrddad Sabetzadeh; Lionel Briand; Frank Zimmer	2015	Plantillas de requisitos, Procesamiento del lenguaje natural (PNL), Estudio de caso de investigación	terminología, Procesamiento del lenguaje natural, oreja, Seguridad, Tuberías, La coincidencia de patrones, Lenguaje de modelado unificado	especificación formal, procesamiento del lenguaje natural, la coincidencia de patrones	verificación automatizada de conformidad, plantillas de requisitos, procesamiento del lenguaje natural, método generalizable, Combinador de patrones de PNL, comunidad de ingeniería de requisitos	13	71
22	Detección de fallas de sistemas estocásticos de tiempo discreto sujetos a requisitos de corrección de lógica temporal	Jun Chen; Ratnesh Kumar;	2015	Filtrado bayesiano, sistemas ciberfísicos, diagnosticabilidad, detección de fallas, lógica temporal en tiempo lineal, sistemas híbridos estocásticos	Autómatas, Detección de fallas, Sistemas de tiempo discreto, Métodos de Bayes, Sistemas estocásticos, Sistemas ciberfísicos, Lenguaje de modelado unificado, Procesamiento del lenguaje natural	teoría de autómatas, sistemas de tiempo discreto, diagnóstico de fallas, control tolerante a fallas, distribuciones estadísticas, procesos estocásticos, sistemas estocásticos, lógica temporal	detección de fallas, discreto, requisito, lógica temporal en tiempo lineal, Especificación LTL, entrada-salida automática estocástico híbrido, Noción I / O-SHA, ecuación de diferencia estocástica, propiedad de accesibilidad, distribución de probabilidad, tasa de falsas alarmas, nTasa de FA, tasa de detección perdida, Tasa de MD, noción de diagnóstico estocástico	2	40
23	Encuesta de trabajos que transforman los requisitos en diagramas UML	Mariem Abdouli; Wahiba Ben Abdessalem Karaa; Henda Ben Ghezala	2016	requisitos, transformación, Diagramas UML, estado del arte, la inteligencia artificial, aprendizaje	Lenguaje de modelado unificado, Software, Inspección, Pragmática, Ingeniería de requisitos, Modelado computacional, Medición	la inteligencia artificial, especificación formal, verificación formal, análisis de sistemas, Lenguaje de modelado unificado	transformación de requisitos, Diagramas UML, ingeniería de requisitos, análisis de requisitos, inteligencia artificial	0	38

24	Una metodología compatible con herramientas para la validación y el refinamiento de modelos de dominio de etapa temprana	Marco Autili; Antonia Bertolino; Guglielmo De Angelis; Davide Di Ruscio; Alessio Di Sandro	2016	Modelado de dominio, Modelo de etapa temprana, Ingeniería dirigida por modelos, Refinamiento del modelo, Validación de modelo, Cuestionarios de lenguaje natural, Calidad del modelo semántico	Lenguaje de modelado unificado, Semántica, Modelado de contexto, Modelado de carga, Motores, Modelado del sistema biológico, Contexto	procesamiento del lenguaje natural, Lenguaje de modelado unificado	modelo de dominio en etapa inicial, ingeniería basada en modelos, transformación automatizada del modelo, Técnica MDE, refinamiento del modelo soportado por herramientas, lenguaje natural, MOTHIA, UML, diagrama de actividad	0	50
25	Un estudio de caso sobre un enfoque de especificación utilizando diagramas de actividad en documentos de requisitos	Martin Beckmann; Andreas Vogelsang; Christian Reuter	2017		Lenguaje de modelado unificado, Partes interesadas, Modelos gráficos, Ingeniería de requisitos, Teoría de la complejidad, Lenguajes naturales	especificación formal, verificación formal, procesamiento del lenguaje natural, análisis de texto	Daimler AG, documento de requisitos textuales, funciones del sistema, especificaciones de lenguaje natural, ingeniería de requisitos, diagrama de actividad	2	28
26	Uso de mapas conceptuales basados en ontología para la ingeniería de requisitos: un estudio de caso	Mahmoud Abd Ellatif; Karama Ali Mohamed; Marwa Salah Farhan	2017	ontología, ingeniería de software, ingeniería de requisitos, ontología, mapas conceptuales	Ontologías, Software, Ingeniería de requisitos, Ingeniería de software, Herramientas, Partes interesadas, Sistemas de información, Lenguaje de modelado unificado, Procesamiento del lenguaje natural	especificación formal, verificación formal, ontologías (inteligencia artificial)	mapas conceptuales basados en ontología, ingeniería de requisitos, procesos de ingeniería de software, representación de información, éxito del proyecto de software, ingeniería de requisitos, representación de requisitos, Fleiss kappa	0	27

27	Extracción de diagramas de clase UML de los requisitos de software en tailandés utilizando PNL	Mathawan Jaiwai; Usa Sammapun	2017	ingeniería de software, análisis de requisitos, UML, diagrama de clase, PNL	Lenguaje de modelado unificado, Software, Discurso, Procesamiento del lenguaje natural, Herramientas, Etiquetado, Modelos analíticos	especificación formal, procesamiento del lenguaje natural, Lenguaje de modelado unificado	Extracción de diagrama de clase UML, requisitos de software, tailandesa, PNL, desarrollo de software, productos de software, procesamiento del lenguaje natural, Inglés, transformación, valores	0	11
28	Extracción automatizada y agrupamiento de requisitos Glosario de términos	Chetan Arora; Mehrddad Sabetzadeh; Lionel Briand; Frank Zimmer	2017	Requisitos glosarios, extracción a plazo, procesamiento del lenguaje natural, agrupamiento, estudio de caso de investigación	terminología, servidores, Tuberías, Lenguajes naturales, Monitoreo, Software, Algoritmos de agrupamiento, Lenguaje de modelado unificado, Procesamiento del lenguaje natural	procesamiento del lenguaje natural, agrupación de patrones, análisis de texto	documentos de requisitos de lenguaje natural, procedimiento basado matemáticamente, parámetros especificados por el usuario, términos del glosario candidato, términos técnicos, algoritmo de agrupamiento, enfoque automatizado, documento de requisitos de software, términos del glosario de requisitos, construcción del glosario, herramientas de extracción de términos genéricos	2	100
29	Evaluación de un enfoque para definir guías de obtención de requisitos no funcionales	Andreia Silva; Plácido R. Pinheiro; Adriano Albuquerque; Jonatas Barroso	2017	guías de obtención de requisitos no funcionales	proceso de obtención, participación del cliente, Lenguaje de modelado unificado, Procesamiento del lenguaje natural	verificación formal, lenguas naturales, aspectos organizacionales	lenguaje natural, obtención de requisitos, organización de desarrollo de software Proceso de evaluación ADEG-NFR, Guías de obtención de NFR	0	29

30	Uso del procesamiento del lenguaje natural para detectar automáticamente la deuda técnica autorremitida	Everton da Silva Maldonado; Emad Shihab; Nikolaos Tsantalis	2017	Deuda técnica, comentarios de código fuente, procesamiento del lenguaje natural, estudio empírico	Software, Procesamiento del lenguaje natural, Manuales, Entropía, Lenguaje de modelado unificado, java, lenguaje de consulta estructurado	delitos informáticos, java, procesamiento del lenguaje natural, gestión de proyectos, software de dominio público, mantenimiento de software, gestión de software, calidad de software, SQL	Procesamiento de lenguaje natural, deuda requerida, detección de deudas técnicas autoadmitidas, proyectos de código abierto, calidad del código fuente, clasificación del comentario del código fuente, PNL, deuda de diseño	10	58
31	Sistema para aplicaciones de modelo de superficie terrestre basado en la computación en la nube	Lihui Luo; Yaonan Zhang; Wei Ma; Yanli Zhuang	2017	Computación en la nube, modelo de superficie terrestre, post-procesamiento, preprocesamiento, Portal web	Computación en la nube, Modelado computacional, Modelos de datos, Superficie de la tierra, Portales, Visualización de datos, Lenguaje de modelado unificado	lenguajes de autoría, computación en la nube, computación geofísica, portales, paquetes de software	lenguajes de autoría, computación en la nube, computación geofísica, portales, paquetes de software	0	45
32	Construcción de casos de prueba utilizando procesamiento de lenguaje natural	Ahlam Ansari; Mirza Baig Shagufta; Ansari Sadaf Fatima; Shaikh Tehreem	2017	prueba, casos de prueba, requisitos funcionales, Procesamiento del lenguaje natural (PNL), Especificación de requisitos de software	Prueba, Procesamiento del lenguaje natural, Computadoras, Software, Conferencias, Documentación, Contexto	especificación formal, procesamiento del lenguaje natural, programa de prueba	probador de software, documento de especificación de requisitos de software, PNL, procesamiento del lenguaje natural, generación de casos de prueba	3	17

33	Hacia el apoyo a la ingeniería de software mediante el aprendizaje profundo: un caso de clasificación de requisitos de software	Raúl Navarro-Almanza; Reyes Juárez-Ramírez; Guillermo Licea	2018	Clasificación de requisitos de software, Red neuronal convolucional, Incrustación de palabras, Ingeniería de software	Software, Aprendizaje automático, Análisis de tareas, Redes neuronales convolucionales, Modelos de datos, Arquitectura de la computadora, Ingeniería de software	prealimentar redes neuronales, especificación formal, aprendizaje (inteligencia artificial), clasificación de patrones, calidad de software	SRC, aprendizaje profundo, tareas relacionadas con el lenguaje natural, requisitos etiquetados, requisitos no funcionales, clasificación de requisitos de software, proceso de desarrollo de software de alta calidad, tarea de clasificación de requisitos de software, ingeniería de características, ingeniería de software, red neuronal convolucional, CNN, Corpus PROMESA requerimientos funcionales	0	13
34	Extracción de características y medición del rendimiento del documento de ingeniería de requisitos (RE) utilizando la técnica de clasificación de texto	LP Saikia; Shilpi Singh	2018	Documento RE (documento de ingeniería de requisitos), índice kappa, Etiquetado de punto de venta, Cociente de probabilidad, Árbol de decisión basado en algoritmo C4.5	Extracción de características, Software, Categorización de texto, java, Índices, Pautas, Herramientas, Lenguaje de modelado unificado, Procesamiento del lenguaje natural	extracción de características, especificación formal, clasificación de patrones, análisis de texto	documento de ingeniería de requisitos, técnica de clasificación de texto, Documento RE, Fase SDLC, desarrollo de software, lenguaje natural, documento de texto, información semántica, índice kappa, la razón de verosimilitud, extracción de características	0	15

35	Cadena de proceso basada en eventos para el modelado y la verificación de requisitos comerciales: una revisión sistemática de la literatura	Anam Amjad; Farooque Azam; Muhammad Waseem Anwar; Wasi Haider Butt; Muhammad Rashid	2018	BPML, EPC, SLR, Verificación EPC, Herramientas EPC	Lenguaje de modelado unificado, Negocios, Herramientas, Modelos analíticos, Semántica, Modelos de datos, Modelado orientado a objetos	procesamiento de datos comerciales, especificación formal, Redes de Petri, Lenguaje de modelado unificado	metamodelo, Anotaciones EPC, Métodos de verificación EPC, procesamiento de eventos complejos, Redes de Petri, eventos, Lenguajes, BPML, verificación de requisitos comerciales, UML	7	100
36	Medición de la consistencia del proceso basada en datos basada en gráficos de restricción de actividad	Xuewei Zhang; Jiacun Wang; Jianchun Xing; Wei Song; Qiliang Yang	2018	Proceso de reconocimiento de datos , medición de consistencia, gráfico de restricción de actividad, restricciones de actividad esenciales	Lenguaje de modelado unificado, Negocios, Control de procesos, Correo electrónico, aeropuertos, Modelos analíticos, matrices lógicas programables	procesamiento de datos comerciales, manejo de datos, teoría de grafos	ACG, consistencia de proceso consciente de datos, integración de procesos, mapeo de procesos, gráfico de restricción de actividad, proceso de sustitución	1	41
37	Extracción de características y medición del rendimiento del documento de ingeniería de requisitos (RE) utilizando la técnica de clasificación de texto	LP Saikia; Shilpi Singh	2018	Documento RE (documento de ingeniería de requisitos), índice kappa, Etiquetado de punto de venta, Cociente de probabilidad, Árbol de decisión basado en algoritmo C4.5	Extracción de características, Software, Categorización de texto, java, Índices, Pautas, Herramientas	extracción de características, especificación formal, clasificación de patrones, análisis de texto	documento de ingeniería de requisitos, técnica de clasificación de texto, Documento RE, Fase SDLC, desarrollo de software, lenguaje natural, documento de texto, información semántica, índice kappa, la razón de verosimilitud, extracción de características	0	19

38	Procesamiento del lenguaje natural para la ingeniería de requisitos: lo mejor está por venir	Fabiano Dalpiaz; Alessio Ferrari; Xavier Franch; Cristina Palomares	2018	procesamiento del lenguaje natural, PNL, ingeniería de requisitos, NLP4RE, requisitos de software, ingeniería de software, desarrollo de software	Desarrollo de software, Análisis de tareas, Procesamiento del lenguaje natural, Ingeniería de software, Ingeniería de requerimientos			1	5
39	Hacia un corpus de documentos de requisitos enriquecidos con anotaciones de marco semántico	Waad Alhoshan; Riza Batista-Navarro; Liping Zhao	2018	Documentos de requisitos, Corpus de requisitos, Anotación semántica, Marcos semánticos, FrameNet	Semántica, Herramientas, Análisis de tareas, XML, Procesamiento del lenguaje natural, Ingeniería de requerimientos, Lenguaje de modelado unificado	especificación formal, gramáticas, procesamiento del lenguaje natural, ingeniería de software, análisis de texto	documentos de requisitos, anotaciones de marco semántico, requisitos de software, representación formal, análisis semántico, ingeniería de requisitos, técnicas de procesamiento del lenguaje natural, corpus de requisitos, FrameNet, FN-RE	0	15
40	Ingeniería ágil de requisitos con historias de usuarios	Fabiano Dalpiaz; Sjaak Brinkkemper	2018	Ingeniería ágil de requisitos, historias de usuarios, procesamiento natural del lenguaje	Ingeniería de requisitos, Conferencias, Tutoriales, Lenguaje de modelado unificado, Procesamiento del lenguaje natural	especificación formal, mecanismos de inferencia, procesamiento del lenguaje natural, herramientas de software, análisis de sistemas	calidad de la historia del usuario, ingeniería ágil de requisitos, procesamiento del lenguaje natural, modelos conceptuales, herramientas narradoras interactivas, compañía de software	0	9

41	Un enfoque de programación en lenguaje natural para pruebas de seguridad basadas en requisitos	Phu X. Mai; Fabrizio Pastore; Arda Goknil; Lionel C. Briand	2018	Pruebas de seguridad del sistema, Requisitos de lenguaje natural, Procesamiento de lenguaje natural, Programación de lenguaje natural	Ontologías, Fiabilidad del software, Lenguajes naturales, Generadores de patrones de prueba, Autorización, Lenguajes de computadora, Contraseña, Lenguaje de modelado unificado, Procesamiento del lenguaje natural	interfaces de programa de aplicación, especificación formal, procesamiento del lenguaje natural, prueba de programa, seguridad de datos	enfoque de programación en lenguaje natural, pruebas de seguridad basadas en requisitos, requisitos de seguridad de software, casos de prueba de seguridad ejecutables, Requisitos de NL, Programación de casos de mal uso, MCP, especificaciones del caso de uso, especificaciones de requisitos, casos de prueba ejecutables, API	0	77
42	Generación de un diagrama de relación de entidad a partir de la especificación de requisitos basada en PNL	PGTH Kashmira; Sagara Sumathipala	2018	Modelo de datos de relación de entidad, Procesamiento de lenguaje natural, Especificación de requerimiento	Lenguaje de modelado unificado, Erbio, Procesamiento del lenguaje natural, Herramientas, Modelos de datos, Modelado orientado a objetos	diagramas, modelado de entidad-relación, especificación formal, procesamiento natural del lenguaje	especificación de requisitos, modelo de datos de relación de entidad, modelo conceptual de alto nivel, diseño del diagrama de relación de entidad, PNL, base de datos, diseñador novato, procesamiento del lenguaje natural, idioma en inglés	0	18
43	Abordar la ambigüedad léxica y semántica en los requisitos del lenguaje natural	Fatima Zait; Nacereddine Zarour	2018	ingeniería de requisitos, lenguaje natural, ambigüedad, la ambigüedad léxica, significado, PLN	Herramientas, Semántica, Sintácticos, Ingeniería de requisitos, Pragmática, Software	especificación formal, verificación formal, procesamiento del lenguaje natural, web semántica	la ambigüedad léxica, ambigüedad semántica, requisitos de lenguaje natural, ingeniería de requisitos, detección de ambigüedad, PLN, palabras ambiguas, ingeniería de software,	0	24

44	Procesamiento de requisitos de lenguaje natural: de la investigación a la práctica	Alessio Ferrari	2018	ingeniería de requisitos, procesamiento natural del lenguaje	Procesamiento del lenguaje natural, Herramientas, Análisis de tareas, Recuperación de información, oro, Normas	especificación formal, verificación formal, recuperación de información, procesamiento natural del lenguaje	técnicas representativas preparadas industrialmente, detección de defectos, problemas de extracción de información, Investigación RE, Técnicas de PNL, procesamiento de requisitos de lenguaje natural, manipulación automatizada, investigadores de ingeniería de requisitos, procesamiento natural del lenguaje	0	0
45	Capa de registro basada en procesamiento de lenguaje natural	Hanji Shen; Chun Long; Wei Wan; Jun Li; Yakui Qin; Yuhao Fu; Xiaofan Song	2018	Datos de registro en tiempo real, Procesamiento de lenguaje natural, Compresión de datos	Redes de computadoras, Seguridad informática, Procesamiento del lenguaje natural, Extracción de características, Seguridad de la información, Codificación	compresión de datos, procesamiento del lenguaje natural, procesamiento de consultas, almacenamiento, monitoreo, análisis de texto	técnica de almacenamiento distribuido bien construida, eficiencia de consulta, segmentación de palabras, tiempo de descompresión, costo de almacenamiento, registros masivos, técnica de procesamiento de lenguaje de texto, datos de registro, procesamiento del lenguaje natural, técnica de almacenamiento en capas de registro en tiempo real, tasa de consulta	0	11

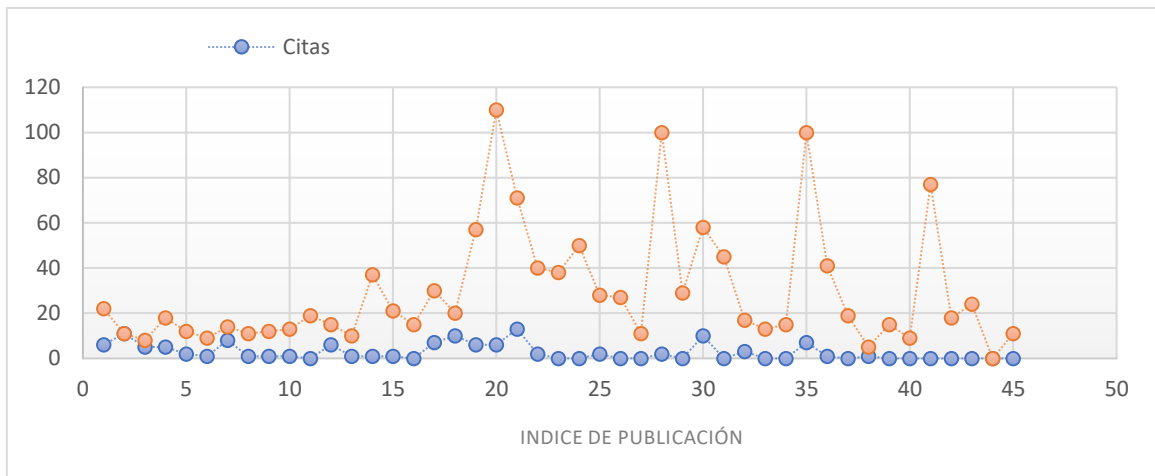


Figura 2.4: Valoración de los artículos según el número de citas y la cantidad de referencia

Fuente: Elaboración propia

2.3 Documentación de la investigación

El objetivo de la fase 3 es documentar las respuestas a las preguntas de investigación presentadas en el paso 1, así como validar el informe.

2.3.1 Validar informe

Los datos y el análisis de esta revisión integradora fueron validados en todo momento por los autores de este documento como parte del proceso de revisión.

III. RESULTADOS

La interpretación de los resultados obtenidos se basa a la revisión de la literatura de los artículos relacionados a la comprensión de los requerimientos de software mediante el lenguaje natural utilizando la visión de la ingeniería de requerimientos. A continuación, se muestra se muestra en la Figura 3.1 la cantidad de estudios relacionados al tema en cuestión ordenados de forma ascendente por año de publicación utilizando la regla de búsqueda [2.1] para aplicar dicho filtro. Cabe mencionar que se evidenció un crecimiento constante salvo en los años 2012, 2013 y 2016 en los cuales se muestra una caída en el aporte de publicaciones relacionadas. Así mismo, se muestra una curva de crecimiento muy importante en los últimos años, lo cual demuestra que la comprensión de requerimientos de software mediante el lenguaje natural utilizando la ingeniería de requerimientos se ha convertido en materia de investigación constante con un impacto muy importante en la industria de desarrollo de software.

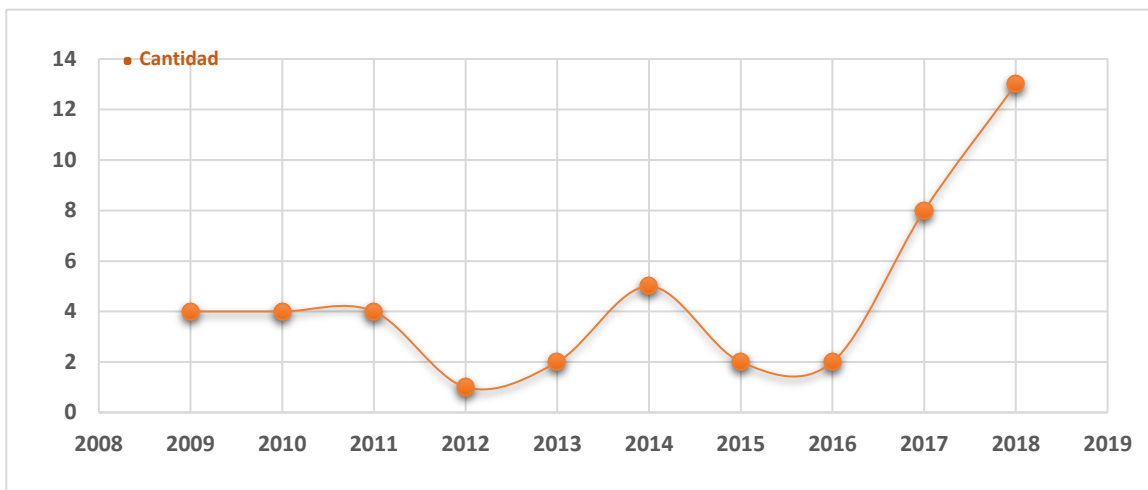


Figura 3.1: Cantidad de artículos por año aplicando el filtro crudo tal cual arroja la regla de búsqueda. **Fuente:** Elaboración propia

De los 45 artículos sobre la comprensión de requerimientos de software mediante el lenguaje natural fueron elegidos para este proyecto 20 para analizar sus porcentajes de precisión se muestran en la tabla 3.1. La cual es el resultado del análisis del contenido de palabras clave, las cuales sirve como descriptores críticos de cada documento para su

selección, además de su precisión en términos de porcentaje final de cada uno detallando las características de sus técnicas utilizadas con las cuales se extrae su información comprensión de requerimientos de software mediante el lenguaje natural.

3.1 Análisis

En esta sección se representa los resultados del análisis respondiendo las preguntas de la investigación planeadas anteriormente [2.1.1] desarrollado como parte de la literatura integradora proceso de revisión.

PRE 1. ¿Cuánto material bibliográfico relacionado a la comprensión de requerimientos de software mediante el lenguaje natural utilizando la visión de la ingeniería de requerimientos se ha publicado entre los años 2009 a 2018?

Si bien es cierto dar una valoración objetiva genérica de la cantidad de material bibliográfico existente relacionado a la comprensión de requerimientos de software mediante el lenguaje natural es muy amplio, ya que depende de la infinidad de bases de datos donde se alojan cada propuesta es por ello, para realizar la propuesta de revisión se hace énfasis en las dos revistas de IEEE filtrando los artículos de acuerdo a la regla de búsqueda planteada. A continuación, se muestra el resultado del filtro y su valoración en la Tabla 3.1, en la cual evidencian los 45 documentos de revistas encontrados en esta búsqueda integradora relacionado a la comprensión de requerimientos de software mediante el lenguaje natural utilizando ciertas características (técnicas IR), pero de los cuales se selecciona 20 documentos se muestran resultados en precisión en términos de porcentaje y las características trabajadas en cada documento las cuales son contribuciones clave de cada uno de estos. Es por ello si una contribución clave en particular es de interés, puede ser igualada a su factor de impacto diario descrito anteriormente en la Tabla 2.3.

N°	Año	Técnicas	Característica	Precisión	Ref.
1	2010	Ontología - NLP	Algoritmo de segmento, ontologías de dominio	79%, 81%, 79%	[50]
2	2011	ISO / IEC / IEEE 29148	Artefactos Análisis, Especificación	80%	[25]
3	2011	NLP - UML	Reglas semánticas de negocio y vocabulario	85.76%, 80.12%, 82.94%	[41]
4	2013	Ontología - NLP	Modelo de dominio (MD)	82%	[12]
5	2013	UML	MD, Modelo de análisis, Diseño orientado a Objeto, NLTK Payton	81,82%	[11]
6	2014	UML- NLP	clustering semántico, reglas enriquecidas semánticas, DC	50%, 60%, 70%	[18]
7	2016	Modelos MDE - UML	NL, DC, DA, Caso de uso	88%, 90%	[68]
8	2017	UML- NLP	Diagrama de actividad, modelos i*, KAOS	95%	[14]
9	2017	UML-NLP-Ontología	Diagrama de dominio, Mapa Conceptual	82%, 90%	[30]
10	2017	UML - NLP	Diagrama de clase	85%, 90%, 77%, 78%	[19]
11	2017	Metodología TEC	Matriz de decisión	95%, 97%	[20]
12	2017	UML-Ontología	Text2Onto, Gate, WorldNet Treetagger, Diagrama de clase, Mapa Conceptual, Kappa	73%, 56%, 40%	[22]
13	2017	UML - NLP	Lenguaje Natural, tool NLP	82%	[70]
14	2018	NLP - UML	DC, Deep learning	80%	[38]
15	2018	UML	Modelo Conceptual, Casos de uso.	85%	[34]
16	2018	NLP	Deep Learning, CNN	80%, 78,5%	[21]
17	2018	NLP	Weka, Matriz de confusión, Kappa	95%, 90%, 85%	[35]
18	2018	NLP-UML	Historias de Usuario (AQUUSA), extracción de modelo conceptual	72%, 93%	[48]
19	2018	NLP- UML	Extracción de datos, regla heurística	90%	[71]
20	2018	NLP-UML	Casos de Prueba, ERS	92%	[72]

Tabla 3.1: Resultados más significativos sobre la comprensión de los requerimientos en lenguaje natural **Fuente:** Elaboración propia.

Años de publicación de los artículos									
2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
4	4	4	1	2	5	2	2	8	13

Tabla 3.1: Cantidad de artículos publicados referencia sobre la comprensión de requerimientos de software mediante el lenguaje natural entre el año 2009 - 2018
Fuente: Elaboración propia.

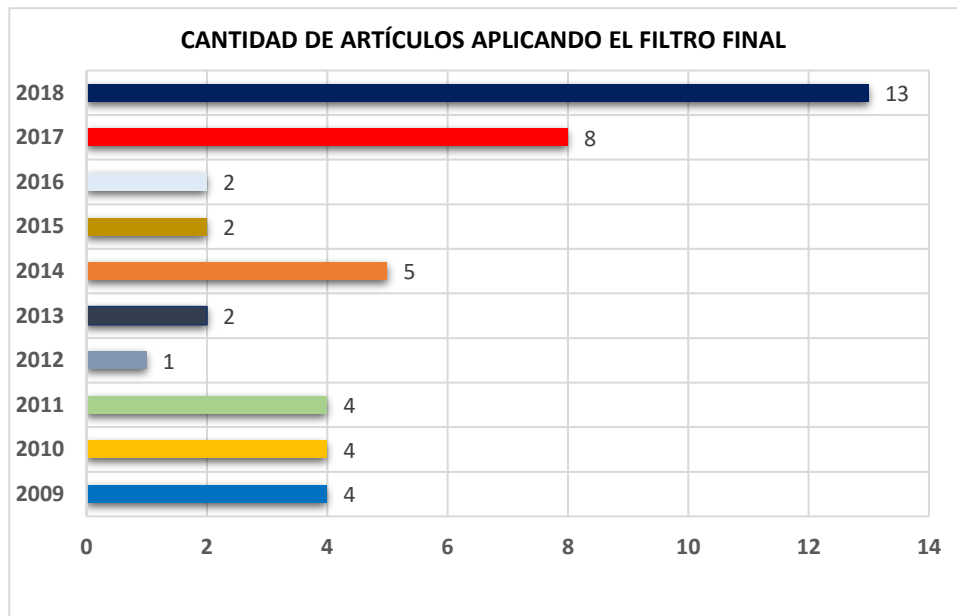


Figura 3.2: Tendencia a la cantidad de publicaciones de los artículos aplicando el filtro final según la regla de revisión **Fuente:** Elaboración propia

PRE 2. ¿Cuál es la técnica predominante para realizar la comprensión de requerimientos de software mediante el lenguaje natural?

El resultado del análisis demuestra que la técnica predominante por excelencia para realizar la comprensión de requerimientos en lenguaje natural utilizando el **Lenguaje de modelado unificado** la cual ha venido ganando muchísimo terreno en el mundo de la ingeniería de requerimientos relacionada a la investigación aplicada por ser sumo interés en áreas ya sea para la comprensión, ambigüedad, comunicación, entre otras. A continuación, se muestra una tabla en la cual se especifica el título del artículo, año de publicación, términos controlados por IEEE para dar fe del uso de la técnica y una valoración lógica del uso de la técnica. Se puede dar fe que esta técnica es una especie de estar para realizar este tipo de investigaciones ya como se puede evidenciar que todos los artículos filtrados por la regla de búsqueda según el protocolo establecido utilizan el procesamiento de lenguaje natural.

Título del documento	Años	Termino de IEEE	UML
A controlled natural language approach for integrating requirements and model-driven engineering	2009	Lenguajes naturales, Ingeniería dirigida por modelos, Calidad de software, Ingeniería de software, Gestión de ingeniería, Automatización, Productividad, Ingeniería de diseño, Control automático, Procesamiento natural del lenguaje, Lenguaje de modelado unificado	SI
An Automated Tool for Generating UML Models from Natural Language Requirements	2009	Lenguaje de modelado unificado, Lenguajes naturales, Automatización, Herramientas de colaboración, Procesamiento del lenguaje natural, Humanos, Ingeniería de software, Software colaborativo, Colaboración internacional, Visualización	SI
Conversion of Pre-Conceptual Schema into Use Case Diagrams by Using Atom3	2009	Meta modelamiento, diagrama de casos de uso, esquemas preconceptuales, Procesamiento del lenguaje natural	SI
Automated Identification of LTL Patterns in Natural Language Requirements	2009	Lenguajes naturales, Propulsión, Automatas, Análisis de patrones, Procesamiento del lenguaje natural, automático, Especificaciones, Programación, frecuencia, Fiabilidad del software	SI
Text-based Requirements preprocessing using nature language processing techniques	2010	Ontologías, Computación distribuida, Helio, software, Web, Aceleración, cara, Colaboración, Sólidos, Minería de datos, Procesamiento del lenguaje natural	SI
Anaphora resolution system for natural language requirements	2010	Lenguajes naturales, Procesamiento del lenguaje natural, Especificaciones formales, Análisis de información, Costos, Análisis de texto, Diseño para experimentos, Recuperación de información, Arquitectura de la computadora, Sociedades	SI
Class Diagram Extraction from Textual Requirements Using Natural Language Processing (NLP) Techniques	2010	Procesamiento del lenguaje natural, Ontologías, Lenguajes naturales, Lenguaje de modelado unificado, Minería de datos, Modelado orientado a objetos, Investigación y desarrollo, Ingeniería de software, Correo electrónico, Automatización	SI
From natural language requirements to requirement ontologies	2010	Lenguajes naturales, Ontología, Bases de datos, Ingeniería de software, La informática, Puentes, Cognición, Ciencia cognitiva, Laboratorios, Diccionarios	SI
Casos de uso y comentarios en el análisis de requisitos funcionales	2011	Negocios; Sistemas de software; Lenguaje de modelado unificado; Instituciones educativas; Modelos analíticos; Modelos de adaptación	SI
Una transformación ATL de modelos de requisitos de lenguaje natural a modelos comerciales de un proyecto MDA	2011	Lenguaje de modelado unificado; Empresas; Producción; Modelado computacional; Contratos; Fabricación integrada por computadora; Lenguajes naturales	SI
El marco de ingeniería de requisitos basado en ISO 29148: 2011 y el marco de modelado de múltiples vistas	2011	Ingeniería de requisitos, Software, Normas ISO, Normas IEC, gobierno, Lenguaje de modelado unificado	SI
Minimizar la ambigüedad en la especificación de requisitos de software de lenguaje natural	2011	Software, Negocios, Semántica, vocabulario, Horarios, Lenguajes naturales, Sintácticos, Lenguaje de modelado unificado	SI
Requisitos de modelado de soporte en el idioma malayo usando casos de uso esenciales	2012	Resúmenes, Lenguajes naturales, Bibliotecas, Software, Conferencias, Modelado computacional, Instituciones educativas	SI
Mapas conceptuales basados en ontologías para	2013	Software, Teoría de la complejidad, historia, Negocios, Cara	SI

ingeniería de software			
Análisis de calidad de requisitos de lenguaje natural basado en modelos de dominio de negocios	2013	Negocios, Ontologías, Lenguajes naturales, Modelos analíticos, Portales, OWL, Normas	SI
Una comparación de técnicas de PNL semánticas para analizar casos de uso	2014	Procesamiento del lenguaje natural, Software, Acoplamientos, Semántica, Compuestos de silicio, Manuales, Inteligencia artificial	SI
Una metodología de ingeniería de requisitos que combina modelos y lenguaje natural controlado.	2014	Contexto, Lenguaje de modelado unificado, Modelado de contexto, Documentación, Lenguajes naturales, Modelos analíticos, Sistemas embebidos	SI
Integrando UML, MARTE y sysml para mejorar la especificación de requisitos y la trazabilidad en el dominio incorporado	2014	Lenguaje de modelado unificado, ruedas, El tiempo, Sensores, Software incorporado, Modelado orientado a objetos	SI
NLP-KAOS para sistemas de obtención de objetivos: estudio de caso del sistema de medición inteligente	2014	Minería de datos, Taxonomía, Ontologías, Modelos de datos, Procesamiento del lenguaje natural, Resúmenes, Recopilación de datos	SI
Apoyar la validación del modelo de proceso mediante la generación de lenguaje natural	2014	Lenguaje de modelado unificado, Lenguajes naturales, Negocios, Modelos analíticos, Modelos de adaptación, Modelado de contexto, Contexto	SI
Verificación automatizada de conformidad con plantillas de requisitos mediante procesamiento de lenguaje natural	2015	terminología, Procesamiento del lenguaje natural, oreja, Seguridad, Tuberías, La coincidencia de patrones	SI
Detección de fallas de sistemas estocásticos de tiempo discreto sujetos a requisitos de corrección de lógica temporal	2015	Autómatas, Detección de fallas, Sistemas de tiempo discreto, Métodos de Bayes, Sistemas estocásticos, Sistemas ciberfísicos	SI
Encuesta de trabajos que transforman los requisitos en diagramas UML	2016	Lenguaje de modelado unificado, Software, Inspección, Pragmática, Ingeniería de requisitos, Modelado computacional, Medición	SI
Una metodología compatible con herramientas para la validación y el refinamiento de modelos de dominio de etapa temprana	2016	Lenguaje de modelado unificado, Semántica, Modelado de contexto, Modelado de carga, Motores, Modelado del sistema biológico, Contexto	SI
Un estudio de caso sobre un enfoque de especificación utilizando diagramas de actividad en documentos de requisitos	2017	Lenguaje de modelado unificado, Partes interesadas, Modelos gráficos, Ingeniería de requisitos, Teoría de la complejidad, Lenguajes naturales	SI
Uso de mapas conceptuales basados en ontología para la ingeniería de requisitos: un estudio de caso	2017	Ontologías, Software, Ingeniería de requisitos, Ingeniería de software, Herramientas, Partes interesadas, Sistemas de información	SI
Extracción de diagramas de clase UML de los requisitos	2017	Lenguaje de modelado unificado, Software, Discurso, Procesamiento del lenguaje natural,	SI

de software en tailandés utilizando PNL		Herramientas, Etiquetado, Modelos analíticos	
Extracción automatizada y agrupamiento de requisitos Glosario de términos	2017	terminología, servidores, Tuberías, Lenguajes naturales, Monitoreo, Software, Algoritmos de agrupamiento	SI
Evaluación de un enfoque para definir guías de obtención de requisitos no funcionales	2017	proceso de obtención, participación del cliente, Lenguaje de modelado unificado, Procesamiento del lenguaje natural	SI
Uso del procesamiento del lenguaje natural para detectar automáticamente la deuda técnica autorremitida	2017	Software, Procesamiento del lenguaje natural, Manuales, Entropía, Lenguaje de modelado unificado, java, lenguaje de consulta estructurado	SI
Sistema para aplicaciones de modelo de superficie terrestre basado en la computación en la nube	2017	Computación en la nube, Modelado computacional, Modelos de datos, Superficie de la tierra, Portales, Visualización de datos	SI
Construcción de casos de prueba utilizando procesamiento de lenguaje natural	2017	Prueba, Procesamiento del lenguaje natural, Computadoras, Software, Conferencias, Documentación, Contexto	SI
Hacia el apoyo a la ingeniería de software mediante el aprendizaje profundo: un caso de clasificación de requisitos de software	2018	Software, Aprendizaje automático, Análisis de tareas, Redes neuronales convolucionales, Modelos de datos, Arquitectura de la computadora, Ingeniería de software	SI
Extracción de características y medición del rendimiento del documento de ingeniería de requisitos (RE) utilizando la técnica de clasificación de texto	2018	Extracción de características, Software, Categorización de texto, java, Índices, Pautas, Herramientas	SI
Cadena de proceso basada en eventos para el modelado y la verificación de requisitos comerciales: una revisión sistemática de la literatura	2018	Lenguaje de modelado unificado, Negocios, Herramientas, Modelos analíticos, Semántica, Modelos de datos, Modelado orientado a objetos	SI
Medición de la consistencia del proceso basada en datos basada en gráficos de restricción de actividad	2018	Lenguaje de modelado unificado, Negocios, Control de procesos, Correo electrónico, aeropuertos, Modelos analíticos, matrices lógicas programables	SI
Extracción de características y medición del rendimiento del documento de ingeniería de requisitos (RE) utilizando la técnica de clasificación de texto	2018	Extracción de características, Software, Categorización de texto, java, Índices, Pautas, Herramientas	SI
Procesamiento del lenguaje natural para la ingeniería de requisitos: lo mejor está por venir	2018	Desarrollo de software, Análisis de tareas, Procesamiento del lenguaje natural, Ingeniería de software, Ingeniería de requerimientos	SI
Hacia un corpus de documentos de requisitos enriquecidos con anotaciones de marco semántico	2018	Semántica, Herramientas, Análisis de tareas, XML, Procesamiento del lenguaje natural, Ingeniería de requerimientos	SI
Ingeniería ágil de requisitos con historias de usuarios	2018	Ingeniería de requisitos, Conferencias, Tutoriales, , Lenguaje de modelado unificado	SI
Un enfoque de programación en lenguaje natural para pruebas de seguridad basadas en requisitos	2018	Ontologías, Fiabilidad del software, Lenguajes naturales, Generadores de patrones de prueba, Autorización, Lenguajes de computadora, Contraseña, Lenguaje de modelado unificado, Procesamiento del lenguaje natural	SI
Generación de un diagrama de relación de entidad a partir de la especificación de requisitos basada en PNL	2018	Lenguaje de modelado unificado, Erbio, Procesamiento del lenguaje natural, Herramientas, Modelos de datos, Modelado orientado a objetos	SI

Abordar la ambigüedad léxica y semántica en los requisitos del lenguaje natural	2018	Herramientas, Semántica, Sintácticos, Ingeniería de requisitos, Pragmática, Software. Lenguaje de modelado unificado, Procesamiento del lenguaje natural	SI
Procesamiento de requisitos de lenguaje natural: de la investigación a la práctica	2018	Procesamiento del lenguaje natural, Herramientas, Análisis de tareas, Recuperación de información, oro, Normas, Lenguaje de modelado unificado	SI
Capa de registro basada en procesamiento de lenguaje natural	2018	Redes de computadoras, Seguridad informática, Procesamiento del lenguaje natural, Extracción de características, Seguridad de la información, Codificación	SI

PRE 3. ¿Qué técnicas de ingeniería de requerimientos existen para realizar la comprensión de requerimientos de software mediante el lenguaje natural?

comprensión de requerimientos de software mediante el lenguaje natural es un tema ampliamente explorado teniendo el lenguaje natural como el proceso más estudiado para explorar los errores más significativos de los requerimientos como, la comunicación, comprensión, ambigüedad y la definición. La calidad, gastos sobre el presupuesto, tiempo de entrega, la extracción iterativa y el refinamiento de los requerimientos del producto se encuentran entre los principales problemas en las empresas de desarrollo de software basados. Las investigaciones más relevantes expuestas por el filtro demuestran que se han utilizado una variedad de técnicas en todo el proceso de desarrollo del software tales como: Elicitación, Análisis, Especificación y validación como se muestra en la tabla 1. Cabe recordar que dichas técnicas se pueden utilizar en las diferentes fases de la ingeniería de requerimientos tales como: Procesamientos de lenguaje natural, Lenguaje de modelado unificado, ontologías entre otras. El criterio para identificar dichas técnicas es el nivel de uso donde fueron estudiado por [1,2,3] y el filtro de búsqueda. A continuación, se muestra la tendencia porcentual en la Figura 3.3 la cual expone la usabilidad de las técnicas.

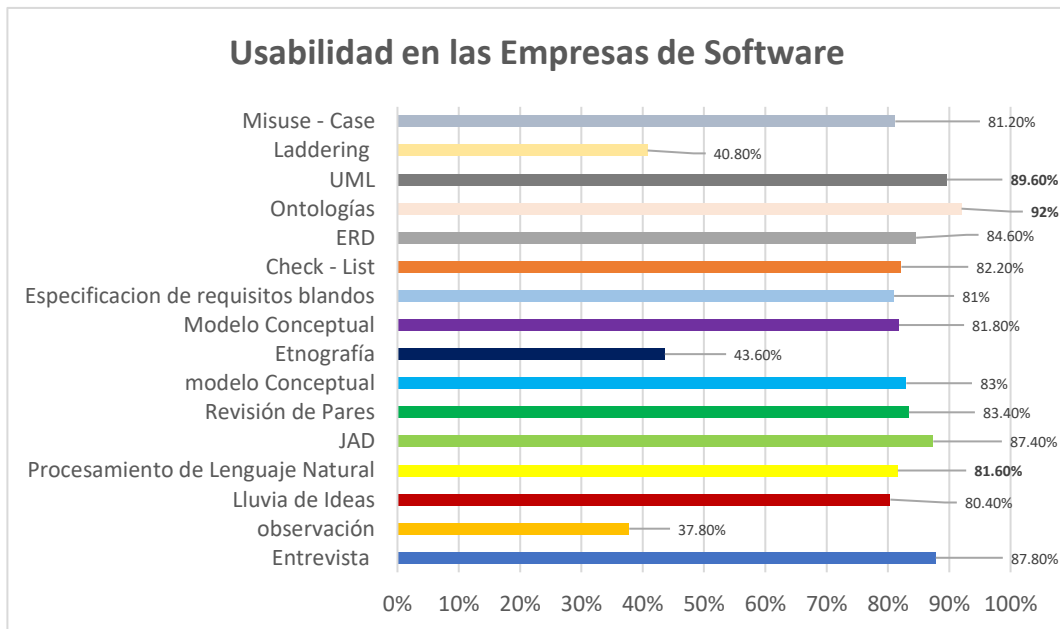


Figura 3.3: Selección de las técnicas de IR para la comprensión de los requerimientos **Fuente:** Elaboración propia

Tabla 2.3: Análisis de precisión de las técnicas de IR con mayor uso **Fuente:**
Elaboración propia

N°	Año	Técnicas	Característica	Precisión	Ref.
1	2011	NLP - UML	Reglas semánticas de negocio y vocabulario	85.76%,80.12%, 82.94%	[41]
2	2013	Ontología - NLP	Modelo de dominio (MD)	82%	[12]
3	2016	Modelos MDE - UML	NL, DC, DA, Caso de uso	88%, 90%	[68]
4	2017	UML-NLP-Ontología	Diagrama de dominio, Mapa Conceptual	82%,90%	[30]
5	2017	UML-Ontología	Text2Onto, Gate, WorldNet Treetagger, Diagrama de clase, Mapa Conceptual, Kappa	73%,56%, 40%	[22]
6	2017	UML - NLP	Lenguaje Natural, tool NLP	82%	[70]
7	2018	UML	Modelo Conceptual, Casos de uso.	85%	[34]
8	2018	NLP	Deep Learning, CNN	80%, 78,5%	[21]
9	2018	NLP-UML	Historias de Usuario (AQUA), extracción de modelo conceptual	72%, 93%	[48]

PRE 4. ¿En que convergen las conclusiones relacionadas a la comprensión de requerimientos en lenguaje natural?

La mayoría de las propuestas relacionado al proceso de comprensión de requerimientos utilizando las técnicas de ingeniería de requerimientos convergen en los siguiente

1. Los problemas de alcance de los proyectos, donde los límites de los sistemas de software están mal definidos o los clientes no especifican detalles técnicos innecesarios que confunden, más que se clarifiquen.
2. Problemas de entendimiento del lenguaje natural se basa que los stakeholders no están seguros de lo que necesitan, conjuntamente tiene dificultades para comunicar a los ingenieros de software llegando a omitir información que creen que no es relevante.
3. Las complicaciones cambiantes de los requerimientos durante todo el ciclo de vida del desarrollo de software son constantemente.

4. El uso de la técnica apropiada en las diferentes fases de la ingeniería de requerimientos, que les permita a los ingenieros de software analizar un requerimiento más concretamente, donde pueda detectar la incompletitud y la inconsistencia de los mismos.

IV.CONCLUSIONES

Basado en la literatura, este documento se presenta una revisión actualizada comprensión de requerimientos de software mediante el lenguaje natural y de las diferentes técnicas que se utilizan a lo largo del proceso de la ingeniería de requerimientos.

Cabe mencionar que, se han identificados retos significativos que deben superarse en términos de definición, comunicación, comprensión, completitud, inconsistencia de los requerimientos, conllevando a los retrasos de entrega del producto de software. Se debe aclarar que, la comprensión de los requerimientos en lenguaje natural es un tema ampliamente explorado teniendo a la técnica, Lenguaje de modelado unificado como el más estudiado para explorar sus diferentes diagramas tales como, diagrama de clases, para planificar funciones nuevas y demostrar las tareas y flujos básicos en tu sistema. Visualiza el estado actual y futuro de alto nivel de tu producto con el fin de garantizar entornos Agile productivos para desarrolladores de software, ingenieros, científicos de datos y otros profesionales técnicos. Investigaciones recientes han utilizado una variedad de técnicas en la ingeniería de requerimientos entre los más significativos se encuentran Procesamiento de lenguaje natural (NLP), Lenguaje de modelado unificado (UML), ontologías de dominio conjuntamente con tools de herramientas de procesamiento de lenguaje natural expresadas en mapas conceptuales, generando resultados extraordinarios siempre enmarcando en el ámbito de la industria de desarrollo de software.

4.1 Limitaciones

- Identificar necesidades que no se plasmaron en la definición de los requerimientos.
- Estimar en total del tiempo y gasto en el desarrollo de producto de software que no están estipulados.
- Si comparamos una solución por software a una solución dada por la mente humana, la limitación más evidente es que un software no tiene inteligencia de por sí, es decir, sólo tiene funciones predefinidas que abarcan un conjunto de soluciones que para ciertas aplicaciones puede llegar a ser limitado.

REFERENCIAS

- [1] D. K. Deeptimahanti and M. A. Babar, "An automated tool for generating UML models from natural language requirements," ASE2009 - 24th IEEE/ACM Int. Conf. Autom. Softw. Eng., pp. 680–682, 2009.
- [2] C. M. Zapata, P. Arango, and F. Arango, "DIAGRAMA DE CASOS DE USO EMPLEANDO AToM 3 CONVERSION OF PRE-CONCEPTUAL SCHEMA INTO USE CASE DIAGRAMS BY USING AToM 3," Dyna, vol. 74, no. 153, pp. 237–251, 2009.
- [3] D. D. A. Ferreira and A. R. Da Silva, "A controlled natural language approach for integrating requirements and model-driven engineering," 4th Int. Conf. Softw. Eng. Adv. ICSEA 2009, Incl. SEDES 2009 Simp. para Estud. Doutor. em Eng. Softw., no. 518, pp. 518–523, 2009.
- [4] M. Ibrahim and R. Ahmad, "Class diagram extraction from textual requirements using natural language processing (NLP) techniques," 2nd Int. Conf. Comput. Res. Dev. ICCRD 2010, pp. 200–204, 2010.
- [5] K. S. Park, D. U. An, and Y. S. Lee, "Anaphora resolution system for natural language requirements document in Korean," ICIC 2010 - 3rd Int. Conf. Inf. Comput., vol. 1, pp. 11–14, 2010.
- [6] D. J. Hurtado, "Comparative analysis between the techniques used in the Requirements Engineering , evaluating such techniques forehead to the characteristics of software projects," 2011.
- [7] C. M. Z. Jaramillo, G. L. Giraldo, and G. A. U. Giraldo, "Las ontologías de la ingeniería de software: un acercamiento de dos grandes áreas del conocimiento," Rev. Ing. Univ. Medellin, vol. 9, no. 16, pp. 91–99, 2010.

- [8] N. A. Maiden, S. Manning, S. Jones, and J. Greenwood, "Generating requirements from systems models using patterns: a case study," *Requirements Engineering*, vol. 10, no. 4, 2005.
- [9] R. De Landtsheer, E. Letier, and A. Van Lamsweerde, "Deriving tabular event-based specifications from goal-oriented requirements models," *Requirements Engineering*, vol. 9, no. 2, 2004.
- [10] C. R. S. G and F. Losavio, "Free Software tools," 2013 XXXIX Lat. Am. Comput. Conf. equipos, pp. 27–34, 2013.
- [11] V. Bhala, R. V. Sagar, and S. Abirami, "Conceptual modeling of natural language functional requirements," *J. Syst. Softw.*, 2013.
- [12] K. M. Annervaz, V. Kaulgud, S. Sengupta, and M. Savagaonkar, "Natural Language Requirements Quality Analysis Based on Business Domain Models," pp. 676–681, 2013.
- [13] M. Abdouli, W. B. A. Karaa, and H. Ben Ghezala, "Survey of works that transform requirements into UML diagrams," 2016 IEEE/ACIS 14th Int. Conf. Softw. Eng. Res. Manag. Appl. SERA 2016, pp. 117–123, 2016.
- [14] M. Beckmann, A. Vogelsang, and C. Reuter, "A Case Study on a Specification Approach Using Activity Diagrams in Requirements Documents," *Proc. - 2017 IEEE 25th Int. Requir. Eng. Conf. RE 2017*, pp. 253–262, 2017.
- [15] S. Besrou, L. B. A. Rahim, and P. D. D. Dominic, "Investigating requirement engineering techniques in the context of small and medium software enterprises," 2016 3rd Int. Conf. Comput. Inf. Sci. ICCOINS 2016 - Proc., pp. 519–523, 2016.

- [16] L. E. P. A. Toro, "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 Cienc. e Ing.*, vol. 10, no. 20, pp. 117–123, 2016.
- [17] X. Zhang, J. Wang, J. Xing, W. Song, and Q. Yang, "Measuring Data-Aware Process Consistency Based on Activity Constraint Graphs," *IEEE Access*, vol. 6, pp. 21005–21019, 2018.
- [18] A. Rago, C. Marcos, and J. A. Diaz-Pace, "Una Comparación de Técnicas de NLP Semánticas para Analizar Casos de Uso," 2014 IEEE Bienn. Congr. Argentina, ARGENCON 2014, vol. 2014-Janua, pp. 479–484, 2014.
- [19] M. Jaiwai and U. Sammapun, "Extracting UML class diagrams from software requirements in Thai using NLP," *Proc. 2017 14th Int. Jt. Conf. Comput. Sci. Softw. Eng. JCSSE 2017*, 2017.
- [20] J. C. Ramirez Leal, W. J. Giraldo Orozco, and R. Anaya Hernandez, "Una Propuesta Metodológica Para Mejorar La Comunicación En Ingeniería De Requisitos," *Rev. EIA*, vol. 13, no. 26, pp. 121–139, 2017.
- [21] R. Navarro-Almanza, R. Juarez-Ramirez, and G. Licea, "Towards Supporting Software Engineering Using Deep Learning: A Case of Software Requirements Classification," *Proc. - 2017 5th Int. Conf. Softw. Eng. Res. Innov. CONISOFT 2017*, vol. 2018-Janua, pp. 116–120, 2018.
- [22] K. A. Mohamed, M. A. Ellatif, and M. S. Farhan, "Using ontology-based concept maps for requirements engineering: A case study," 2017 13th Int. Comput. Eng. Conf., pp. 366–371, 2017.
- [23] J. Xu, "Use Cases and Feedback in Functional Requirements Analysis," pp. 55–58, 2011.

- [24] N. Debnath et al., “An ATL transformation from natural language requirements models to business models of a MDA project,” 2011 11th Int. Conf. ITS Telecommun. ITST 2011, pp. 633–639, 2011.
- [25] D. Selvyanti, “The Requirements Engineering Framework Based On ISO 29148 : 2011 and Multi-View Modeling Framework,” 2011.
- [26] M. Fockel and J. Holtmann, “A requirements engineering methodology combining models and controlled natural language,” in 4th International Model-Driven Requirements Engineering Workshop (MoDRE). IEEE, 2014
- [27] R. Gacitúa Bustos, “Procesamiento de Lenguaje Natural en Ingeniería de Requisitos: Contribuciones Potenciales y Desafíos de Investigación,” pp. 2–4, 2015.
- [28] A. Amjad, F. Azam, M. W. Anwar, W. H. Butt, and M. Rashid, “Event-Driven Process Chain for Modeling and Verification of Business Requirements-A Systematic Literature Review,” IEEE Access, vol. 6, pp. 9027–9048, 2018.
- [29] H. Shen et al., “Log Layering Based on Natural Language Processing,” Int. Conf. Adv. Commun. Technol. ICACT, vol. 2019-February, pp. 660–663, 2019.
- [30] M. A. Ellatif, “Using Ontology-Based Concept Maps to Reduce Requirement Engineering Errors,” vol. 15, no. 6, pp. 292–303, 2017.
- [31] P. G. T. H. Kashmira and S. Sumathipala, “Generating Entity Relationship Diagram from Requirement Specification based on NLP,” 2018 3rd Int. Conf. Inf. Technol. Res. ICITR 2018, pp. 1–4, 2018.

- [32] F. Zait and N. Zarour, "Addressing Lexical and Semantic Ambiguity in Natural Language Requirements," *5th Int. Symp. Innov. Inf. Commun. Technol. ISIICT 2018*, pp. 1–7, 2019.
- [33] H. Shen et al., "Log Layering Based on Natural Language Processing," *Int. Conf. Adv. Commun. Technol. ICACT*, vol. 2018-Febru, pp. 660–663, 2018.
- [34] G. Benigni and I. Marcano, "Incorporando modelos mentales y conceptuales en el análisis de requerimientos para desarrollar software educativo development of educational software," vol. 35, no. No 3, pp. 1–12, 2018.
- [35] L. P. Saikia and S. Singh, "Feature extraction and performance measure of requirement engineering (RE) document using text classification technique," *Proc. 4th IEEE Int. Conf. Recent Adv. Inf. Technol. RAIT 2018*, pp. 1–6, 2018.
- [36] J. J. P. Surelys G. Pérez Jiménez, "Procedimiento para la obtención de requerimientos funcionales " pp. 1–7, 2018.
- [37] M. Fockel and J. Holtmann, "A requirements engineering methodology combining models and controlled natural language," *2014 IEEE 4th Int. Model. Requir. Eng. Work. MoDRE 2014 - Proc.*, pp. 67–76, 2014.
- [38] F. Dalpiaz, A. Ferrari, X. Franch, and C. Palomares, "Natural Language Processing for Requirements Engineering: The Best Is Yet to Come," *IEEE Softw.*, vol. 35, no. 5, pp. 115–119, 2018
- [39] M. Kamalrudin, J. Grundy, and J. Hosking, "Supporting requirements modelling in the Malay language using essential use cases," *Proc. IEEE Symp. Vis. Lang. Human-Centric Comput. VL/HCC*, pp. 153–156, 2012.

- [40] J.DeFranco and P. A. Laplante, “Integrative Literature Review Review and Analysis of Software Development Team Communication Research,” *IEEE Transactions on Professional Communication*, vol. 00, no. 00, pp. 1–18, 2017.
- [41] A. UMBER and I. S. Bajwa, “Minimizing ambiguity in natural language software requirements specification,” 2011 6th Int. Conf. Digit. Inf. Manag. ICDIM 2011, pp. 102–107, 2011.
- [42] M. Fockel and J. Holtmann, “A requirements engineering methodology combining models and controlled natural language,” 2014 IEEE 4th Int. Model. Requir. Eng. Work. MoDRE 2014 - Proc., pp. 67–76, 2014.
- [43] M. R. Sena Marques, E. Siegert, and L. Brisolara, “Integrating UML, MARTE and sysml to improve requirements specification and traceability in the embedded domain,” *Proc. - 2014 12th IEEE Int. Conf. Ind. Informatics, INDIN 2014*, pp. 176–181, 2014.
- [44] A. Ferrari, “Natural language requirements processing,” *Proc. 40th Int. Conf. Softw. Eng. Companion Proceedings - ICSE '18*, pp. 536–537, 2018
- [45] F. Dalpiaz, A. Ferrari, X. Franch, and C. Palomares, “Natural Language Processing for Requirements Engineering: The Best Is Yet to Come,” *IEEE Softw.*, vol. 35, no. 5, pp. 115–119, 2018.
- [46] A. Hossian, “Modelo de Proceso de Conceptualización de Requisitos,” *Rev. Latinoam. Ing. Softw.*, vol. 1, no. 3, p. 79, 2015.
- [47] L. Lengyel, T. Mészáros, M. Asztalos, and P. Boros, “Quality Assured Model-Driven Requirements Engineering and Software Development,” vol. 58, no. 11, 2015.

- [48] F. Dalpiaz and S. Brinkkemper, “Agile requirements engineering with user stories,” Proc. - 2018 IEEE 26th Int. Requir. Eng. Conf. RE 2018, pp. 506–507, 2018.
- [49] A. P. Nikora and G. Balcom, “Automated identification of LTL patterns in natural language requirements,” Proc. - Int. Symp. Softw. Reliab. Eng. ISSRE, pp. 185–194, 2009.
- [50] H. Chen, K. He, P. Liang, and R. Li, “Text-based requirements preprocessing using nature language processing techniques,” 2010 Int. Conf. Comput. Des. Appl. ICCDA 2010, vol. 1, no. Iccda, 2010.
- [51] R. S. Pressman, Ingeniería del software. Un enfoque práctico. 2010.
- [52] I. Sommerville, Software Engineering. 2011.
- [53] J. Darío and Torres, “Guía Para La Integración De Métodos Formales De Ingeniería De Requerimientos En Procesos De Desarrollo Ágil,” pp. 1–67, 2014.
- [54] P. M. Institute, Gestión de requisitos: una guía de práctica. 2016.
- [55] G. Rodríguez, Departamento de Ciencias de la Computación y Electrónica Sección Ingeniería del Software y Gestión de Tecnologías de la Información Programación Avanzada Guía didáctica Autores : 2017.
- [56] J. M. Atlee, R. J. LeBlanc, T. C. Lethbridge, A. Sobel, and J. B. Thompson, “Software engineering 2004,” no. 0003263, p. 623, 2005.
- [57] I. Std, “Especificacion de Requerimientos según el estándar de IEEE 830,” 2008.
- [58] B. E. Gottesdiener, “A View To Agile Requirements,” pp. 1–8, 2009.

- [59] M. Kassab, C. Neill, and P. Laplante, "State of Practice in Requirements Engineering: Contemporary Data," *Innovations in Systems and Software Eng.*, vol. 10, no. 4, 2014, pp. 235–241
- [60] . D.M. Berry, E. Kamsties, and M.M. Krieger, *From Contract Drafting to Software Specification: Linguistic Sources of Ambiguity*, tech. report, School of Computer Science, Univ. of Waterloo,
- [61] K. Ryan, "The Role of Natural Lan- guage in Requirements Engineering," *Proc. 1993 IEEE Int'l Symp. Re- quirements Eng.*, 1993, pp. 240–242.
- [62] V. Casey, "Developing trust in virtual software development teams," *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 5, no. 2, pp. 41–58, 2010
- [63] Bell Manrique losada, "Natural Language Processing for Knowledge Acquisition : Approaches From," pp. 69–78, 2015.
- [64] Gavaldá, M, *La investigación en tecnologías de la lengua. Research in language technology*, 2011
- [65] E. Casagrande, S. Woldeamlak, W. L. Woon, H. H. Zeineldin, and D. Svetinovic, "NLP- KAOS for systems goal elicitation: Smart metering system case study," *IEEE Trans. Softw. Eng.*, vol. 40, no. 10, pp. 941–956, 2014.
- [66] H. Leopold, J. Mendling, and A. Polyvyanyy, "Supporting process model validation through natural language generation," *IEEE Trans. Softw. Eng.*, vol. 40, no. 8, pp. 818–840, 2014.
- [67] C. Arora, M. Sabetzadeh, L. Briand, and F. Zimmer, "Automated checking of conformance to requirements templates using natural language processing," *IEEE Trans. Softw. Eng.*, vol. 41, no. 10, pp. 944–968, 2015.

- [68] M. Autili, A. Bertolino, G. De Angelis, D. Di Ruscio, and A. Di Sandro, "A Tool-Supported Methodology for Validation and Refinement of Early-Stage Domain Models," *IEEE Trans. Softw. Eng.*, vol. 42, no. 1, pp. 2–25, 2016.
- [69] A. Silva, P. R. Pinheiro, A. Albuquerque, and J. Barroso, "Evaluation of an approach to define elicitation guides of non-functional requirements," *IET Softw.*, vol. 11, no. 5, pp. 221–228, 2017.
- [70] C. Arora, M. Sabetzadeh, L. Briand, and F. Zimmer, "Automated Extraction and Clustering of Requirements Glossary Terms," *IEEE Trans. Softw. Eng.*, vol. 43, no. 10, pp. 918–945, 2017.
- [71] E. D. S. Maldonado, E. Shihab, and N. Tsantalis, "Using Natural Language Processing to Automatically Detect Self-Admitted Technical Debt," *IEEE Trans. Softw. Eng.*, vol. 43, no. 11, pp. 1044–1062, 2017.
- [72] A. Ansari, M. B. Shagufta, A. Sadaf Fatima, and S. Tehreem, "Constructing Test cases using Natural Language Processing," *Proc. 3rd IEEE Int. Conf. Adv. Electr. Electron. Information, Commun. Bio-Informatics, AEEICB 2017*, pp. 95–99, 2017.