



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

**CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

**TRABAJO DE TITULACIÓN PREVIO LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN SISTEMAS E INFORMÁTICA**

**TEMA: MODELO DE PROCESOS ALTERNATIVO DE INGENIERÍA DE  
REQUISITOS EN POS DE MEJORAR LA CALIDAD DEL SOFTWARE**

**AUTORES: ESPARZA ECHANIQUE RICHARD SEBASTIÁN,  
PELÁEZ RAMOS PAOLA STEFANÍA**

**DIRECTOR: PhD. FONSECA CARRERA, EFRAÍN RODRIGO**

**SANGOLQUÍ**

**2019**

## CERTIFICADO



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

### CERTIFICACIÓN

Certifico que el trabajo de titulación, "*MODELO DE PROCESOS ALTERNATIVO DE INGENIERÍA DE REQUISITOS EN POS DE MEJORAR LA CALIDAD DEL SOFTWARE*" fue realizado por los señores *ESPARZA ECHANIQUE RICHARD SEBASTIÁN* y *PELÁEZ RAMOS PAOLA STEFANÍA*, el mismo ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustenten públicamente.

Sangolquí, 29 de abril del 2019

Dr. Efraín R. Fonseca C.

**DIRECTOR**

## AUTORÍA DE RESPONSABILIDAD



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

### AUTORÍA DE RESPONSABILIDAD

Nosotros, **ESPARZA ECHANIQUE RICHARD SEBASTIÁN** y **PELÁEZ RAMOS PAOLA STEFANÍA**, declaramos que el contenido, ideas y criterios del trabajo de titulación: **“MODELO DE PROCESOS ALTERNATIVO DE INGENIERÍA DE REQUISITOS EN POS DE MEJORAR LA CALIDAD DEL SOFTWARE”** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, 29 de abril del 2019

ESPARZA ECHANIQUE  
RICHARD SEBASTIÁN  
C.C. 1723266407

PELÁEZ RAMOS  
PAOLA STEFANÍA  
C.C. 1104703058

**AUTORIZACIÓN**

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**  
**CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

**AUTORIZACIÓN**

*Nosotros, **ESPARZA ECHANIQUE RICHARD SEBASTIÁN** y **PELÁEZ RAMOS PAOLA STEFANÍA**, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: "**MODELO DE PROCESOS ALTERNATIVO DE INGENIERÍA DE REQUISITOS EN POS DE MEJORAR LA CALIDAD DEL SOFTWARE**" en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra autoría y responsabilidad.*

Sangolquí, 29 de abril del 2019

  
ESPARZA ECHANIQUE  
RICHARD SEBASTIÁN  
C.C. 1723266407

  
PELÁEZ RAMOS  
PAOLA PELÁEZ  
C.C. 1104733058

## DEDICATORIA

Esta tesis se la dedicamos a:

Dios, quien supo guiarnos durante todo este proceso académico y darnos la fortaleza y salud para seguir adelante y no rendirnos antes los problemas que muchas veces se presentaron.

A nuestros padres, quienes con su amor, trabajo y sacrificio durante todos estos años nos han permitido llegar a cumplir un sueño más y con sus consejos nos han motivado a seguir adelante sin rendirnos.

A nuestros hermanos por su cariño y apoyo incondicional, durante todo este proceso. A toda nuestra familia porque con sus oraciones, consejos y palabras de aliento hicieron de nosotros mejores personas.

A nuestros amigos por que supieron brindarnos su apoyo cuando más lo necesitábamos.

“La dicha de la vida consiste en tener siempre algo que hacer, alguien a quien amar y alguna cosa que esperar”

Thomas Chalmers

## **AGRADECIMIENTO**

Queremos agradecer primero a Dios, por habernos guiado a lo largo de nuestra carrera universitaria y por ser nuestra fortaleza en los momentos de debilidad.

A nuestros padres y hermanos, quienes con su gran amor, apoyo incondicional y paciencia nos guiaron por el camino del bien y nos enseñaron a luchar por nuestros objetivos.

Agradecemos a la Universidad de las Fuerzas Armadas ESPE y a todos nuestros docentes quienes compartieron con nosotros su valioso conocimiento que nos permitió crecer día a día como profesional, gracias por su paciencia, dedicación, apoyo incondicional y amistad.

Finalmente, queremos expresar nuestro más grande agradecimiento al PhD. Efraín R. Fonseca C. principal colaborador durante todo este proceso, quien con su dirección, conocimiento, enseñanza y colaboración permitió el desarrollo de este trabajo.

## ÍNDICE DE CONTENIDOS

CERTIFICADO .....	i
AUTORÍA DE RESPONSABILIDAD.....	ii
AUTORIZACIÓN.....	iii
DEDICATORIA.....	iv
AGRADECIMIENTO.....	v
ÍNDICE DE TABLAS .....	xi
ÍNDICE DE FIGURAS.....	xiii
RESUMEN.....	xv
ABSTRACT .....	xvi
CAPÍTULO I.....	1
INTRODUCCIÓN .....	1
1.1 ANTECEDENTES .....	1
1.2 PROBLEMÁTICA .....	3
1.3 JUSTIFICACIÓN.....	3
1.4 OBJETIVOS.....	5
1.4.1. Objetivo General.....	5
1.4.2. Objetivos Específicos .....	5
1.5 ALCANCE .....	5

1.6	HIPÓTESIS .....	6
	CAPÍTULO II .....	7
	MARCO METODOLÓGICO .....	7
2.1.	MARCO TEÓRICO .....	7
2.1.1.	Señalamiento de variables.....	7
2.1.1.1.	Operacionalización de variables.....	7
2.1.2.	Propuestas respecto a los problemas de ingeniería de requisitos.....	8
2.1.2.1.	Problemas Comunes .....	9
2.1.2.2.	Causas Comunes .....	10
2.1.2.3.	Tipos de Propuestas.....	11
2.1.3.	Modelo de procesos de ingeniería de requisitos .....	12
2.1.3.1.	Calidad .....	12
2.1.3.2.	Calidad de software .....	13
2.1.3.3.	Calidad de requisitos .....	14
2.1.3.4.	Métricas de calidad de Software .....	15
2.1.4.	Propuestas de modelos de procesos de ingeniería de requisitos .....	17
2.1.4.1.	Modelo de Pohl .....	17
2.1.4.2.	El modelo espiral.....	19
2.1.4.3.	El modelo SWEBOK .....	20

2.1.4.4.	Otros Modelos .....	21
2.1.5.	Métodos de Investigación .....	25
2.1.5.1.	Estudio de caso .....	26
2.1.5.2.	Encuestas .....	26
2.1.5.3.	Revisión Sistemática de Literatura.....	26
2.1.5.4.	Mapeo Sistemático de Literatura.....	26
2.1.5.5.	Estudio Terciario .....	27
2.2.	ESTADO DEL ARTE .....	27
2.2.1.	Planteamiento del objetivo de búsqueda.....	27
2.2.2.	Conformación del grupo de control (GC).....	28
2.2.3.	Construcción y afinación de la cadena de búsqueda.....	29
2.2.4.	Proceso de Selección de Estudios Primarios y Extracción de Datos .....	30
2.2.5.	Determinación de las características del estado del arte .....	31
2.2.5.1.	Resumen de los estudios primarios .....	31
2.2.5.2.	Características del estado del arte .....	34
2.3.	METODOLOGÍA DE LA INVESTIGACIÓN .....	35
2.3.1.	Metodología Design Science .....	35
2.3.1.1.	Identificación del problema y motivación.....	35
2.3.1.2.	Definición de los objetivos de la propuesta .....	36

2.3.1.3.	Desarrollo del artefacto .....	36
2.3.1.4.	Demostración del artefacto.....	37
2.3.1.5.	Evaluación del artefacto .....	37
2.3.2.	Design Science aplicado .....	38
CAPÍTULO III .....		39
MAPEO SISTEMÁTICO DE LITERATURA .....		39
3.1.	PROPÓSITO DE LA INVESTIGACIÓN.....	39
3.2.	DEFINICIÓN DE LAS PREGUNTAS DE INVESTIGACIÓN .....	39
3.3.	DEFINICIÓN DE CRITERIOS DE INCLUSIÓN Y EXCLUSIÓN .....	39
3.4.	DEFINICIÓN DE LA ESTRATEGIA DE BÚSQUEDA.....	40
3.4.1.	Configuración del GC .....	41
3.4.2.	Construcción de la cadena de búsqueda.....	42
3.4.3.	Búsqueda de estudios candidatos.....	44
3.5.	DEPURACIÓN DE ESTUDIOS CANDIDATOS.....	45
3.6.	SELECCIÓN DE ESTUDIOS.....	46
3.6.1.	Selección de Estudios Primarios.....	46
3.7.	EXTRACCIÓN DE RESULTADOS .....	48
3.8.	SÍNTESIS DE RESULTADOS DE LA SMS .....	72

CAPITULO IV .....	75
MODELO PROPUESTO .....	75
4.1. PROPUESTA DEL MODELO .....	75
4.1.1. Elicitación de Requisitos.....	77
4.1.2. Análisis y Negociación .....	77
4.1.3. Especificación de Requisitos .....	79
4.1.4. Validación.....	79
4.1.5. Evolución .....	80
4.2. VALIDACIÓN DEL MODELO .....	83
4.2.1. Diseño y planificación del estudio caso.....	83
4.2.2. Preparación y recolección de datos.....	89
4.2.3. Análisis y presentación de datos .....	95
CAPÍTULO V .....	110
CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS .....	110
5.1. CONCLUSIONES.....	110
5.2. RECOMENDACIONES .....	112
5.3. TRABAJOS FUTUROS.....	112
REFERENCIAS BIBLIOGRÁFICAS .....	113

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> <i>Preguntas de Investigación</i> .....	6
<b>Tabla 2.</b> <i>Estudios del Grupo de Control</i> .....	29
<b>Tabla 3.</b> <i>Artículos que conforman el Grupo de Control</i> .....	41
<b>Tabla 4.</b> <i>Cadena de Búsqueda (Estudios recuperados)</i> .....	43
<b>Tabla 5.</b> <i>Estudios resultantes por base digital</i> .....	45
<b>Tabla 6.</b> <i>Estudios duplicados por base digital</i> .....	45
<b>Tabla 7.</b> <i>Estudios Seleccionados</i> .....	46
<b>Tabla 8.</b> <i>Estudios Primarios</i> .....	47
<b>Tabla 9.</b> <i>Estudios de Ingeniería de Requisitos</i> .....	49
<b>Tabla 10.</b> <i>Problemas Relacionados con la IR</i> .....	50
<b>Tabla 11.</b> <i>Categorías de Problemas de Requisitos</i> .....	53
<b>Tabla 12.</b> <i>Causas Identificadas</i> .....	56
<b>Tabla 13.</b> <i>Resumen de Métodos de Investigación identificados en el EP12</i> .....	62
<b>Tabla 14.</b> <i>Tipos de Contribución relacionadas con Ontologías encontrados en el EP3</i> .....	66
<b>Tabla 15.</b> <i>Técnicas de requisitos más aceptadas encontradas en el EP5</i> .....	66
<b>Tabla 16.</b> <i>Checklist aplicable para participación de usuarios encontrada en el EP8</i> .....	68
<b>Tabla 17.</b> <i>Propuestas de Ingeniería de Requisitos identificadas en el EP10</i> .....	69
<b>Tabla 18.</b> <i>DRSR Sistema A</i> .....	85
<b>Tabla 19.</b> <i>DRSR Sistema B</i> .....	87
<b>Tabla 20.</b> <i>Plantilla Historia de Usuario</i> .....	88
<b>Tabla 21.</b> <i>Plantilla Partial Delivery Story</i> .....	88

<b>Tabla 22.</b> <i>Tipos de Requisitos para la métrica de evolución</i> .....	91
<b>Tabla 23.</b> <i>Nivel de adecuación del documento (NAD)</i> .....	95
<b>Tabla 24.</b> <i>Requisitos Evolucionados Grupo 1</i> .....	96
<b>Tabla 25.</b> <i>Requisitos Evolucionados Grupo 2</i> .....	97
<b>Tabla 26.</b> <i>Requisitos Evolucionados Grupo 3</i> .....	98
<b>Tabla 27.</b> <i>Requisitos Evolucionados Grupo 4</i> .....	99
<b>Tabla 28.</b> <i>Requisitos con Defectos encontrados en la ERS del Grupo 1</i> .....	99
<b>Tabla 29.</b> <i>Requisitos con Defectos encontrados en la ERS del Grupo 2</i> .....	100
<b>Tabla 30.</b> <i>Requisitos con Defectos encontrados en la ERS del Grupo 3</i> .....	101
<b>Tabla 31.</b> <i>Requisitos con Defectos encontrados en la ERS del Grupo 4</i> .....	102
<b>Tabla 32.</b> <i>Porcentaje de Requisitos Soportados Sistema A (G1 y G2)</i> .....	103
<b>Tabla 33.</b> <i>Porcentaje de Requisitos Soportados Sistema B (G3 y G4)</i> .....	105
<b>Tabla 34.</b> <i>Calidad General de los Documentos de ERS resultantes</i> .....	107

## ÍNDICE DE FIGURAS

<i>Figura 1.</i> Reporte del Caos de Proyectos Software.....	1
<i>Figura 2.</i> Variables identificadas.....	7
<i>Figura 3.</i> Métricas de Calidad de Boehm.....	16
<i>Figura 4.</i> Métricas de Calidad de McCall .....	16
<i>Figura 5.</i> Métricas de Calidad de la ISO 9126.....	17
<i>Figura 6.</i> Modelo de Procesos de IR propuesto por Pohl.....	18
<i>Figura 7.</i> Modelo Espiral de Ingeniería de Requisitos.....	19
<i>Figura 8.</i> Modelo SWEBOOK de procesos de ingeniería de requisitos.....	20
<i>Figura 9.</i> Modelo de Procesos de Ingeniería de Requisitos .....	22
<i>Figura 10.</i> Proceso de IR basado en Ontologías .....	25
<i>Figura 11.</i> Design Science.....	35
<i>Figura 12.</i> Design Science aplicado a la investigación .....	38
<i>Figura 13.</i> Detalle Contextos.....	43
<i>Figura 14.</i> Relación Estudios Primarios.....	48
<i>Figura 15.</i> Distribución Anual de estudios de IR.....	50
<i>Figura 16.</i> PRC – Causas .....	57
<i>Figura 17.</i> GDC – Causas.....	57
<i>Figura 18.</i> PRE – Causas.....	58
<i>Figura 19.</i> HYT – Causas.....	58
<i>Figura 20.</i> PRR – Causas .....	58
<i>Figura 21.</i> PRP – Causas.....	59

<b>Figura 22.</b> Métodos de Investigación Identificados en el EP3 .....	60
<b>Figura 23.</b> Métodos de Investigación Identificados en el EP11 .....	62
<b>Figura 24.</b> Métodos de Investigación identificados en el EP14 .....	63
<b>Figura 25.</b> Prácticas de Ingeniería de Requisitos identificadas en el EP2 .....	65
<b>Figura 26.</b> Propuestas de Ingeniería de Requisitos identificadas en el EP11 .....	70
<b>Figura 27.</b> Tendencia de Propuestas identificada en la literatura.....	72
<b>Figura 28.</b> Modelo de Procesos de IR propuesto .....	76
<b>Figura 29.</b> Fase Evolución.....	81
<b>Figura 30.</b> Proceso de realización de estudios de caso.....	83
<b>Figura 31.</b> Proceso de aplicación del estudio de caso .....	89
<b>Figura 32.</b> Configuración de los grupos de trabajo.....	94
<b>Figura 33.</b> Cronograma de Aplicación del Estudio de Caso.....	94
<b>Figura 34.</b> Métricas de calidad de los documentos del sistema A.....	108
<b>Figura 35.</b> Métricas de calidad de los documentos del sistema B.....	108
<b>Figura 36.</b> Resumen general de la calidad resultante según el modelo aplicado.....	109

## **RESUMEN**

La investigación en Ingeniería de Requerimientos (IR) ha aumentado significativamente en los últimos años, debido a que los requisitos son el primer aporte primario en el proceso de desarrollo de software. Sin embargo, siguen existiendo graves problemas con respecto a los requisitos que incluso han causado el fracaso de varios proyectos de desarrollo de software. Ante esta realidad nos propusimos como objetivo estructurar un artefacto que proporcione una vista panorámica de los problemas relacionados con la IR, sus causas y las características de las soluciones propuestas. Para lo cual realizamos un proceso de mapeo sistemático de literatura (SMS). Como resultado de la información disponible en los 15 estudios primarios, encontramos que las actividades implícitas de ingeniería de requisitos han sido objeto de un arduo trabajo de investigación, que se ha intensificado en los últimos años. A partir de los hallazgos encontrados, estructuramos un modelo de procesos para la mejora del proceso de Ingeniería de Requisitos con el fin de tratar los problemas que se presentan en la IR y así contribuir a la mejora de la calidad del software, ya que los resultados obtenidos del SMS apuntan a que pese de la importancia de la IR en el proceso de desarrollo de software y de una importante cantidad de investigación llevada a cabo en este campo; no existe un modelo sólido para guiar la práctica de IR basado en una vasta base de conocimiento empírico.

### **KEYWORDS:**

- **MAPEO SISTEMÁTICO DE LITERATURA**
- **MODELO DE PROCESOS**
- **INGENIERÍA DE REQUISITOS**
- **INGENIERÍA DE SOFTWARE**

## **ABSTRACT**

Research in Requirements Engineering (RE) has improved in recent years, because the requirements are the first primary input in the software development process. However, there are still graves. Given this reality, we set ourselves the objective of structuring an artifact that provides a panoramic view of the problems related to the ER, its causes and the characteristics of the proposed solutions. For what we do a process of systematic literature mapping (SMS). As a result of the information available in the 15 primary studies, we found that the implicit engineering activities of the requirements have been the subject of hard research work, which has intensified in recent years. Based on the findings, we structured a process model for the improvement of the Requirements Engineering process in order to deal with the problems that arise in the IR and as well as the improvement of the software quality, since the results SMS reports suggest that despite the importance of the ER in the software development process and a significant amount of research carried out in this field; there is no solid model to guide the practice of ER based on an empirical knowledge base.

### **KEYWORDS:**

- **SYSTEMATIC MAPPING STUDY**
- **PROCESS MODEL**
- **REQUIREMENTS ENGINEERING**
- **SOFTWARE ENGINEERING**

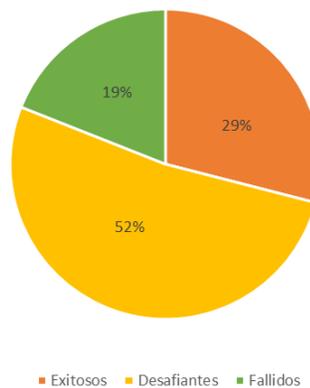
# CAPÍTULO I

## INTRODUCCIÓN

### 1.1 ANTECEDENTES

La evolución de la tecnología informática ha obligado a que se incremente exponencialmente el desarrollo de productos software (Sun & Shinde, 2016). En la actualidad los productos software se los puede encontrar en casi todas las actividades que realizan a diario las personas en todo el mundo, lo que ha incidido en la transformación de la producción de sistemas a nivel mundial (Nadir, Streitferdt, & Burggraf, 2016).

A pesar de la importancia del producto software, un alto porcentaje de proyectos orientados a construir software fracasan por distintos factores (Fernández Sanz & Bernad Silva, 2014). El reporte del caos publicado por Standish Group en 2015, reporta que el 29% de productos software fueron un éxito, el 52% implicaron ser desafiantes y el 19% fracasaron, como lo indica la Figura 1. En general, un proyecto de desarrollo de software es considerado exitoso o de calidad cuando es entregado en el tiempo acordado, no se sale del presupuesto y a la vez contempla todos los requisitos del cliente (Frese & Sauter, 2014).



**Figura 1.** Reporte del Caos de Proyectos Software

Fuente: (Frese & Sauter, 2014)

Está claro entonces que la búsqueda de la calidad es vital en los proyectos de desarrollo de software, ya que por ejemplo con la entrega oportuna del software las empresas se ubican estratégicamente en la industria de Software y compiten con otras (Olaperi & Sanjay, 2015). Es por esto que existen una gran cantidad de estudios enfocados en la calidad de software (e.g.: (Fitzgerald, Letier, & Finkelstein, 2011) (Nagappan, Murphy, & Basili, 2008) (Nagappan, Ball, & Zeller, 2006) (Kpodjedo, Ricca, Galinier, Guéhuéneuc, & Antoniol, 2011) (Rempel & Mäder, 2016)), los cuales muestran la presencia de distintos factores que inciden en la calidad del software. Siendo que los requisitos juegan un papel fundamental en el éxito o fracaso de un proyecto de software, se constituyen en uno de los factores más importantes en la calidad del software (Kalinowski, et al., 2015), lo cual es ampliamente aceptado.

De acuerdo con la IEEE, por ejemplo, la ingeniería de requisitos es “la fase inicial y la más importante en el ciclo de vida del Software ya que su principal objetivo es obtener de forma clara el alcance del problema así como los requisitos del cliente, lo cual contribuirá a la calidad resultante del producto software” (Ouhbi, Idri, Fernandez-Aleman, & Toval, 2013) (IEEE, 2014). Podría decirse entonces, que si los requisitos no están bien definidos y comprendidos, la calidad resultante del sistema se verá afectada (Solemon, Sahibuddin, & Abd , 2008) (Suranto, 2015). Sin embargo, comprender y definir claramente los requisitos no es una tarea sencilla; de hecho, que por ejemplo (Kalinowski, y otros, 2015) afirman que un gran número de proyectos de software fallan debido a la existencia de problemas en el proceso de Ingeniería de Requisitos.

## **1.2 PROBLEMÁTICA**

Los proyectos de desarrollo de software presentan problemas en el proceso de Ingeniería de Requisitos, afectando a la calidad del software resultante (Suranto, 2015). Dentro de estos problemas se encuentran defectos en la comunicación, pobre entendimiento del contexto, requisitos ausentes y requisitos incompletos (Méndez Fernández, et al., 2017). Sin embargo, y pese a los numerosos esfuerzos que se han hecho en torno a esta problemática, no se han encontrado las causas o peor aún las soluciones para estos problemas, debido a la naturaleza altamente cambiante y compleja de los requisitos y también al costo que representa el corregir problemas relacionados con los mismos (Ouhbi, Idri, Fernandez-Aleman, & Toval, 2013) (Mafra, Kalinowski, Méndez Fernández, Felderer, & Wagner, 2016).

Es por esta razón que el presente trabajo de investigación plantea un modelo de procesos para la mejora del proceso de Ingeniería de Requisitos con el fin de tratar los problemas que se presentan en la IR y así contribuir a la mejora de la calidad del software.

### **Formulación del Problema**

¿Es posible desarrollar un Modelo de procesos alternativo de Ingeniería de Requisitos que permita aportar a la mejora en la calidad de los productos de software?

## **1.3 JUSTIFICACIÓN**

La ingeniería de requisitos es la fase inicial y la más importante al momento de desarrollar software (Suranto, 2015); por lo que la aplicación de buenas prácticas en el proceso de ingeniería de requisitos de software es una condición fundamental para lograr productos software de calidad (Londoño, Anaya , & Tabares, 2008).

Desarrollar productos software de calidad es de suma importancia debido a que estos optimizan las tareas, incrementan las ganancias, aumentan los ingresos y optimizan los tiempos;

permitiendo realizar actividades como: monitoreo e implementación de las plataformas gubernamentales, gestión de carteras financieras, realizar procedimientos médicos, transporte, sistemas de defensa, manejo de energía, construir e incluso controlar sistemas en tiempo real que no pueden permitirse fallar (Olaperi & Sanjay, 2015) (G. Martins & Gorscheck, 2016). Las fallas durante el control de estos sistemas pueden causar graves daños al medio ambiente, a las propiedades y a las personas, afectando a las empresas, al mercado, así como a la calidad de vida de las personas y de la sociedad en general (Heimdal , 2007). Uno de los desafíos más importantes para las empresas de desarrollo es crear y establecer una especificación y/o entendimiento de requisitos completos, correctos, no ambiguos, comprobables e incluso comprensibles entre las partes interesadas (Hatcliff, Wassyn, Kelly, Comar, & Jones, 2014). Sin embargo, y a pesar de su importancia, no existe aún una guía, modelo o técnica que brinde una solución definitiva a los problemas que se presentan en la Ingeniería de Requisitos (Nolan, Abrahão, & Clements, 2011); esto se debe principalmente a la naturaleza altamente cambiante de los requisitos, lo que los convierte en un factor de riesgo fundamental y un determinante de éxito para los proyectos de desarrollo de software (Rempel & Mäder, 2016).

Ante esta realidad, es preciso realizar esfuerzos (proponer modelos, herramientas o metodologías) enfocados a tratar los problemas que se presentan en la Ingeniería de Requisitos para contribuir a la mejora de la calidad del software resultante.

## **1.4 OBJETIVOS**

### **1.4.1. Objetivo General**

Plantear un Modelo de procesos alternativo de Ingeniería de Requisitos con el fin contribuir a la mejora de la calidad de los productos de software, a través de la síntesis de resultados de un mapeo sistemático de literatura.

### **1.4.2. Objetivos Específicos**

- Realizar un estudio exploratorio sobre las propuestas existentes respecto a los problemas en la IR con el propósito de identificar los problemas más comunes y sus causas, a través de un mapeo sistemático de literatura.
- Plantear un modelo de procesos para tratar los problemas críticos identificados de ingeniería de requisitos, a partir de la síntesis de resultados del mapeo sistemático de literatura.
- Validar el modelo propuesto a través de un estudio de caso aplicado a los estudiantes de la asignatura de Desarrollo de sistemas de la Universidad de las Fuerzas Armadas ESPE.

## **1.5 ALCANCE**

El proyecto de investigación consiste en el planteamiento de un modelo de procesos de ingeniería de requisitos para aportar a la mejora en la calidad de los productos de software. Como base para el planteamiento de dicho modelo se partirá de la realización de un mapeo sistemático de literatura (SMS) que permita recopilar información relevante en torno a la Ingeniería de Requisitos.

Una vez establecido el modelo, será validado a través de un estudio de caso aplicado en un grupo de estudiantes de la asignatura de Desarrollo de sistemas de la carrera de Ingeniería en Sistemas de la Universidad de las Fuerzas Armadas ESPE.

Para definir el alcance de la presente investigación se establecen las siguientes preguntas de investigación asociadas a los objetivos específicos, tal como se muestra en la Tabla 1:

**Tabla 1.**  
*Preguntas de Investigación*

OBJETIVO ESPECÍFICO	PREGUNTA DE INVESTIGACIÓN
<p>i. Realizar un estudio exploratorio sobre las propuestas existentes respecto a los problemas en la IR con el propósito de identificar los problemas más comunes y sus causas, a través de un mapeo sistemático de literatura.</p> <p>ii. Plantear un modelo de procesos para tratar los problemas críticos identificados de ingeniería de requisitos, a partir de la síntesis de resultados del mapeo sistemático de literatura.</p>	<p><b>RQ1.</b> ¿Cuál es la frecuencia de los estudios realizados cuyo enfoque es la Ingeniería de Requisitos?</p> <p><b>RQ2.</b> ¿Qué problemas y causas son más habituales dentro de la Ingeniería de Requisitos?</p> <p><b>RQ3.</b> ¿Cuáles son los métodos de investigación empleados en los trabajos que se han llevado a cabo en torno a la Ingeniería de Requisitos?</p> <p><b>RQ4.</b> ¿Cuál es la tendencia en torno a las propuestas realizadas con miras a tratar los problemas vinculados a la Ingeniería de Requisitos?</p> <p><b>RQ5.</b> ¿Es posible estructurar un modelo de procesos alternativo de Ingeniería de Requisitos?</p>
<p>iii. Validar el modelo propuesto a través de un estudio de caso aplicado a los estudiantes de la asignatura de Desarrollo de sistemas de la Universidad de las Fuerzas Armadas ESPE.</p>	<p><b>RQ6.</b> ¿Cuáles son los beneficios obtenidos al aplicar el modelo de procesos alternativo de IR con respecto al tradicional?</p>

## 1.6 HIPÓTESIS

El desarrollo de un Modelo de procesos alternativo de Ingeniería de Requisitos permitirá aportar a la mejora en la calidad de los productos de software en el estudio de caso propuesto.

## CAPÍTULO II

### MARCO METODOLÓGICO

#### 2.1. MARCO TEÓRICO

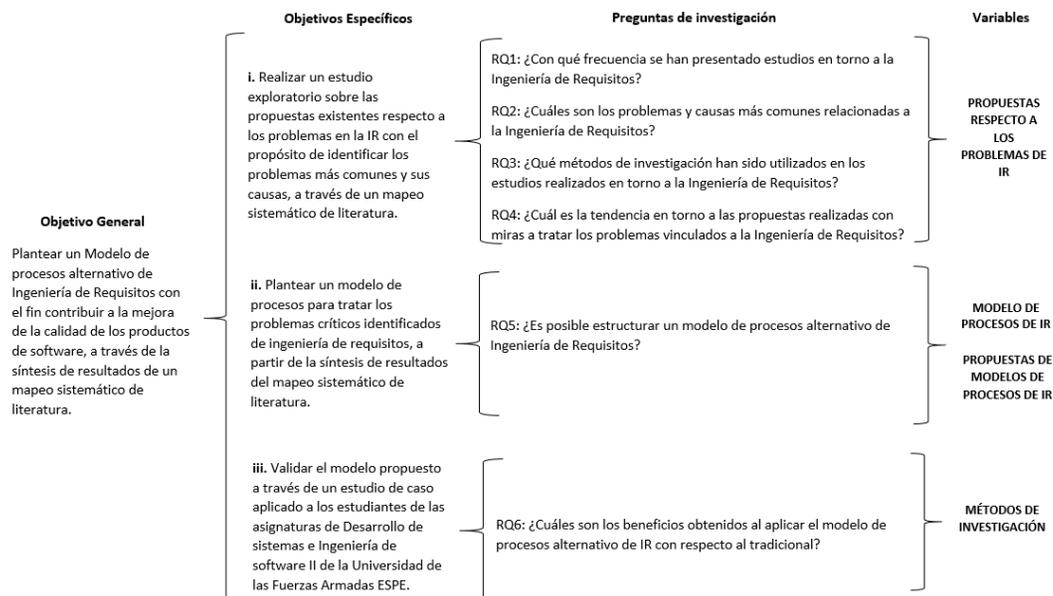
A continuación, se indican las variables que conforman la hipótesis propuesta, detallada en el capítulo anterior.

##### 2.1.1. Señalamiento de variables

- **Variable independiente:** modelo de procesos alternativo de Ingeniería de Requisitos
- **Variable dependiente:** calidad de los productos de software en el estudio de caso propuesto.

##### 2.1.1.1. Operacionalización de variables

Con la finalidad de definir el marco teórico que será el sustento de este trabajo se identificaron las variables que se detallan en la *Figura 2*.



**Figura 2.** Variables identificadas

### **2.1.2. Propuestas respecto a los problemas de ingeniería de requisitos**

La Ingeniería de Requisitos (IR) representa una fase fundamental del proceso desarrollo de software, dado que tiene como finalidad comprender de forma clara, concisa y precisa los requisitos del cliente para el desarrollo del sistema (Fernández Sanz & Bernad Silva, 2014). La IR se creó con el objetivo de incrementar la calidad del software (Kalinowski, et al., 2015). Los costos implicados en detectar y corregir errores se relacionan directamente con la fase del proceso en que son detectados; lo que quiere decir que, de detectar todos estos errores en las etapas iniciales, no sería tan alto el impacto en el desarrollo del sistema. Lo que contribuye en minimizar los costos y tiempo y a la vez se cumple con las expectativas del cliente (Kalinowski, et al., 2015).

Sin embargo, pese a los grandes intentos que apuntan al proceso de Ingeniería de Requisitos en las Industrias de Software, los principales problemas relacionados a la IR permanecen. Estos problemas se relacionan con aquellos requisitos ambiguos e incompletos, así como con la inadecuada gestión de herramientas para este proceso (Khankaew & Riddle, 2014). Daniel Méndez, investigador de NaPiRE (Fernández Méndez & Wagner, 2014), asegura que la ejecución de un proceso adecuado de Ingeniería de Requisitos, incide notablemente en la calidad del producto software; no obstante, también menciona que los requisitos son altamente volátiles e inherentemente complejos por naturaleza.

Badariah Solemon y colegas en su estudio (Solemon, Sahibuddin, & Abd , 2008) realizado a 63 Pymes de software de Malasia identificaron desafíos cruciales para la Ingeniería Requisitos, indica que actualmente son más las empresas que concuerdan en que la IR es un pilar fundamental en el proceso de desarrollo de software.

De la misma manera, Kalinowski y colegas (Kalinowski, et al., 2015), afirman en su estudio que son varios los proyectos que fracasan debido a retos presentados al momento de llevar

a cabo el proceso de IR. Este estudio fue ejecutado en Brasil y consistió en replicar un familia de encuestas a 74 empresas, de todos los tamaños, las mismas que empleaban tanto metodologías ágiles como tradicionales (Fernández Méndez & Wagner, 2014). Como resultado, los autores obtuvieron un diagrama causal sobre los desafíos habituales en la IR. Mientras que, en el estudio llevado a cabo por la UADE en varias compañías de Argentina (Oliveros, Napolillo, & López Infesta, 2016), se constató que existen varias falencias en los estudios previos sobre IR.

El proceso de descubrimiento científico no está limitado a científicos profesionales que trabajan en laboratorios. La experiencia diaria de deducir qué está sucediendo con el automóvil que no enciende comparte similitudes con los descubrimientos en la ciencia. Dichas actividades involucran realizar observaciones, analizar evidencia, comprobar ideas y aferrarse a aquellas que sí funcionan (Undersea, Northwest, & Bauer, 2013).

#### **2.1.2.1. Problemas Comunes**

De acuerdo a los estudios realizados por (Fernández Méndez & Wagner, 2014), (Fernández Sanz & Bernad Silva, 2014), (Kalinowski, et al., 2017), (Kalinowski, et al., 2015), (Méndez Fernández, et al., 2017), y (Méndez Fernández, Tießler, Kalinowski, Felderer, & Kuhrmann, 2017) como parte de la iniciativa NaPiRE (Naming the Pain in Requirements Engineering) en varias instancias alrededor del mundo, los problemas más comunes que se presentan en torno a la Ingeniería de Requisitos son:

- Responsabilidades poco claras
- Falta de conocimiento del producto
- Falta de conocimiento del contexto
- Falta de comunicación y retroalimentación

- Recursos insuficientes para el entendimiento y mantenimiento de los requerimientos.
- Requerimientos ocultos o incompletos
- Objetivos cambiantes
- Requerimientos no especificados que son muy abstractos
- Tiempo Insuficiente
- Stakeholders con dificultad para separar requerimientos de diseños de soluciones existentes
- Falta de colaboración del cliente
- Requerimientos inconsistentes
- Dificultad en el acceso a las necesidades del cliente y a la información del negocio
- Falta de habilidades
- Personal de trabajo inestable
- Falta de Involucramiento del usuario final
- Requerimientos no funcionales poco claros
- Requerimientos implícitos no especificados
- Falta de colaboración del líder de proyecto
- Implementación de características que no corresponden a los requerimientos especificados
- Barreras en el lenguaje

#### **2.1.2.2.Causas Comunes**

De acuerdo a los estudios realizados por (Fernández Méndez & Wagner, 2014), (Fernández Sanz & Bernad Silva, 2014), (Kalinowski, et al., 2017), (Kalinowski, et al., 2015), (Méndez Fernández, et al., 2017), y (Méndez Fernández, Tießler, Kalinowski, Felderer, & Kuhrmann, 2017) como parte de la iniciativa Naming the Pain in Requirements Engineering (NaPiRE), las causas

más comunes reportadas en la literatura que se presentan en torno a la disciplina de la Ingeniería de Requisitos son:

- Poco tiempo para contratación de personal
- Stakeholders con diferentes requerimientos.
- Stakeholders con dificultad para separar requerimientos de diseños de soluciones existentes
- Barreras en el lenguaje
- Falta de colaboración del cliente
- Tiempo Insuficiente
- Falta de conocimiento del contexto
- Requerimientos ocultos o incompletos
- Falta de experiencia
- Técnicas de elicitación pobres
- Requerimientos no especificados que son muy abstractos

### 2.1.2.3. Tipos de Propuestas

Los tipos de propuestas de investigación, según lo expuesto por (Ouhbi, Idri, Fernandez-Aleman, & Toval, 2013), (Petersen, Feldt, Mujtaba, & Mattsson, 2008) y (Barmi, Ebrahimi, & Feldt, 2011) se pueden clasificar en:

- **Método:** una forma de procedimiento y una serie de pasos para adquirir conocimiento en torno a un tema en particular.
- **Herramienta:** un medio para lograr una tarea o propósito enfocado en un tema en específico.

- **Modelo:** una representación de un sistema que permite que se investiguen las propiedades del tema a tratar.
- **Guía:** una indicación de la política por la cual se puede determinar un curso de acción en torno a una temática establecida.
- **Marco:** una estructura real o conceptual que pretende servir de guía para la construcción de conocimiento sobre un tema.
- **Técnica:** un procedimiento con el cual se realiza una tarea, por ejemplo. técnica de minería de datos.
- Otro, por ejemplo un proceso o una métrica.

### **2.1.3. Modelo de procesos de ingeniería de requisitos**

De acuerdo con (Deepti, P, & Parameshwar, 2011), un modelo de procesos de Ingeniería de Requisitos puede describirse como el conjunto de actividades dirigidas a descubrir, obtener y validar continuamente un conjunto de varios tipos diferentes de requisitos que debe cumplir un sistema en desarrollo.

La ingeniería de requisitos es de naturaleza multidisciplinaria y de acuerdo con (Nuseibeh & Easterbrook, 2000) y (Cheng & Atlee, 2007) está compuesto aproximadamente por las cinco fases de Redacción de requisitos, Modelado y análisis de requisitos, Comunicación de requisitos y Acuerdo o validación, Evolución de requisitos y Gestión de requisitos generales.

#### **2.1.3.1. Calidad**

La calidad es cuestión de competitividad (Cubillos Rodríguez & Rozo Rodríguez, 2009); es por esta razón que el término “calidad” es un concepto utilizado frecuentemente, sin embargo, su significado se percibe de distintas formas.

Por un lado (Reeves & Bednar, 1994), define a la calidad desde tres puntos de vista: valor, cumplir o exceder las expectativas del cliente y la conformidad con las especificaciones y requisitos.

Por otro lado, Joseph M. Juran la define como “la integración e interrelación de todas las funciones y procesos en una organización para lograr el mejoramiento continuo de los productos y servicios y obtener la satisfacción del cliente” (Juran, 1990).

En base a las definiciones mencionadas anteriormente podemos decir que la calidad se relaciona directamente con las exigencias de los clientes respecto a la satisfacción de sus necesidades, de ahí que las empresas se ven en la necesidad de ofrecer servicios de calidad en busca de la presencia y permanencia de los clientes en el mercado.

### **2.1.3.2. Calidad de software**

Calidad en el lenguaje de la ingeniería de Software significa “cumplir los requerimientos y tener un alto grado de usabilidad” (Lewis, 2004). Esto implica que el software cumple con los requerimientos de los usuarios como está escrito en la especificación de requerimientos y hace exactamente lo que el usuario necesita.

Por un lado, la calidad es considerada un requerimiento vital de los productos software, ya que por ejemplo con la entrega oportuna del software las empresas se ubican estratégicamente en la industria de Software y compiten con otras (Olaperi & Sanjay, 2015); es decir la calidad del producto software es determinada de acuerdo a que tanto cumple el producto con los requerimientos y especificaciones del usuario y el número de defectos existentes en este.

Por otro lado, (Nakai, Tsuda, Honda, Washizaki, & Fukazawa, 2016) consideran que la calidad de software es el grado de en qué las necesidades requeridas son satisfechas (por ejemplo: propósito del software, desempeño, usabilidad del software y facilidad de mantenimiento) si esta

calidad es insuficiente el producto software podría incluir incidentes, altos costo de desarrollo o mantenimiento y violaciones de las necesidades del usuario.

Sin embargo, los problemas en la calidad de software y errores ocurren a lo largo del ciclo de vida del producto software, los cuales pueden afectar la eficiencia, rendimiento y calidad del proceso de Ingeniería de Software (Rahman & Nurdatillah, 2015).

Dada la importancia de la calidad de software esta ha sido objeto de numerosos estudios empíricos en el pasado, lo que ha generado una gran cantidad de conocimiento empírico sobre este tema. De esta manera gran variedad de factores influyentes en la calidad del software han sido estudiados (Rempel & Mäder, 2016), pero muy pocos estudios se han centrado en los factores de la ingeniería de requerimientos que afectan a la calidad de Software.

### **2.1.3.3. Calidad de requisitos**

Los requisitos de software son la base que determina la calidad de los productos software (Suranto, 2015); es decir que la falta de concordancia de los requisitos afecta directamente al producto final de software y por ende a la calidad del mismo.

Según (Fotrousi, Fricker, & Fiedler, 2014) la definición correcta de los requisitos otorga al sistema atributos tales como: idoneidad funcional, rendimiento, confiabilidad, facilidad de uso, seguridad y portabilidad; atributos indispensables para alcanzar los objetivos de los stakeholders; caso contrario estos tendrán un impacto negativo para los Stakeholders.

Existen modelos y normas de Ingeniería de software disponibles para definir correctamente los requisitos no funcionales (es decir calidad) del software (Laporte & April, 2018).

**Modelo de Calidad:** Según la ISO, un modelo de Calidad es un conjunto definido de características y de relaciones entre ellas, que proporciona un marco para especificar los requisitos de calidad y evaluar la calidad (ISO, 2007).

Un modelo de Calidad de Software se utiliza para que el usuario pueda:

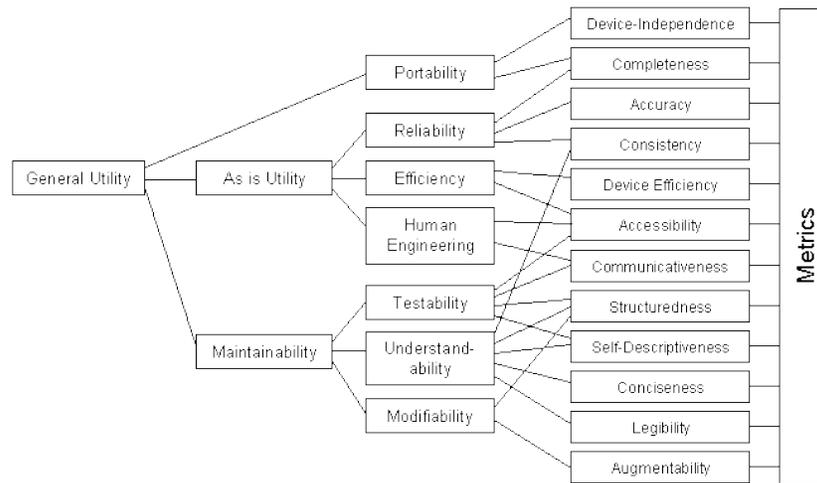
- Especificar las particularidades de calidad de software que puedan ser valoradas;
- Contrastar las diferentes perspectivas del modelo de calidad que entran en juego (es decir, perspectivas internas y externas);
- Elija cuidadosamente un número limitado de características de calidad que servirán como los requisitos no funcionales para el software (es decir, requisitos de calidad);
- Establecer una medida y sus objetivos para cada uno de los requisitos de calidad.

Por lo tanto, es necesario que el modelo demuestre su capacidad para respaldar la definición de los requisitos de calidad y, posteriormente, su medición y evaluación (Laporte & April, 2018).

#### **2.1.3.4.Métricas de calidad de Software**

La calidad del proyecto consiste en una serie de elementos de calidad, cada uno de los elementos es de calidad según una medida. Las mediciones de calidad abarcan todo el proceso de Ingeniería de Software así como la entrega del software, ejemplo de estas medidas son la complejidad del programa, la efectividad del módulo y el tamaño total del programa. La medida después de la entrega del proyecto incluye principalmente el número de defectos residuales y la mantenibilidad del sistema. En circunstancias normales las características y métricas de calidad del software se pueden definir de acuerdo a un modelo (Huang & Sun, 2013).

(Boehm, 1989), en su obra *Software Risk Managment*, propuso por primera vez un modelo jerárquico de las métricas de calidad del proyecto software. Las métricas de este modelo se detallan en la Figura 3.



**Figura 3.** Métricas de Calidad de Boehm.

Fuente: (Boehm, 1989)

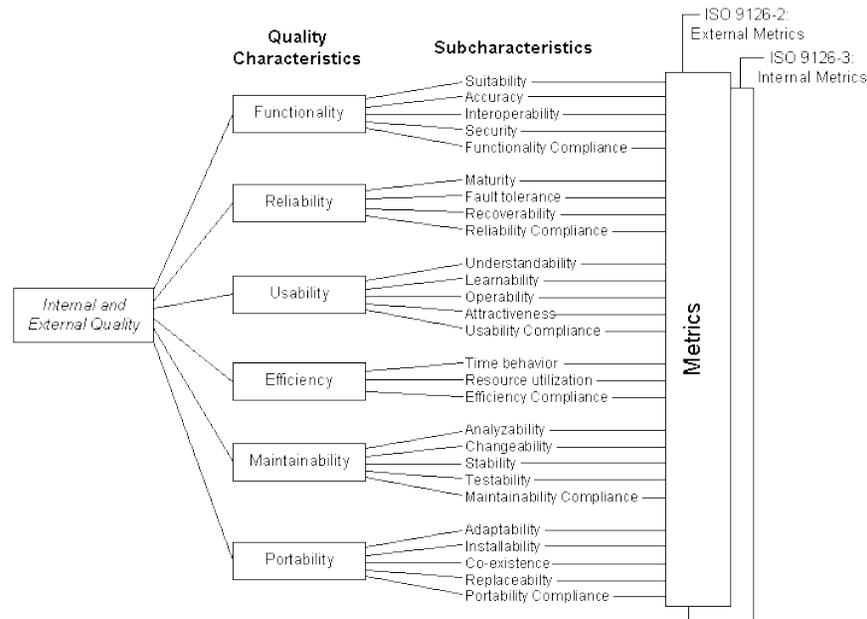
(McCall, Richards, & Walters , 1977); proponen un modelo en donde se detallan los factores que afectan la calidad del software (ver Figura 4). Principalmente se enfocan en tres aspectos de vital importancia del software como son: sus características operativas, su capacidad de ser modificado y su adaptabilidad a nuevos ambientes.



**Figura 4.** Métricas de Calidad de McCall

Fuente: (McCall, Richards, & Walters , 1977)

La ISO 9126 propone un modelo que incluye tres capas (ver Figura 5), la principal: requisitos de calidad del proyecto y criterios de evaluación; la del medio: la calidad de los criterios de evaluación del diseño del proyecto y la baja: los criterios de evaluación (Sukoco, Marzuki, & Cucus, 2012).



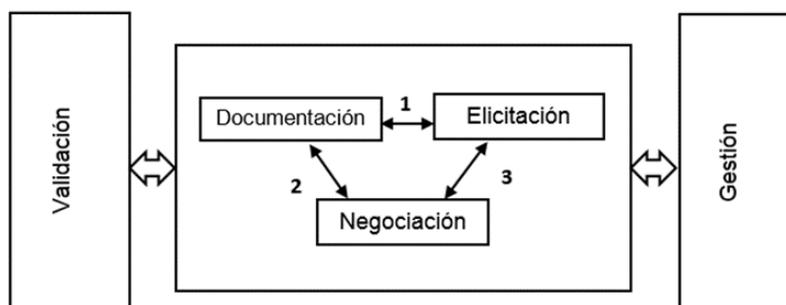
**Figura 5.** Métricas de Calidad de la ISO 9126

Fuente: (Sukoco, Marzuki, & Cucus, 2012)

## 2.1.4. Propuestas de modelos de procesos de ingeniería de requisitos

### 2.1.4.1. Modelo de Pohl

(Pohl, 2010) Propone realizar un conjunto de actividades interrelacionadas con la finalidad de adquirir los requisitos del sistema, donde todas las actividades deberán ser registradas en documentos siguiendo los estándares correspondientes, que en instancias posteriores permitirán la validación y gestión de todo el proceso, mismo que se describe en la Figura 6.



**Figura 6.** Modelo de Procesos de IR propuesto por Pohl

Fuente: (Pohl, 2010)

Para llevar a cabo este proceso se debe realizar las siguientes actividades a continuación descritas (Pohl, 2010):

- **Documentación y elicitación:**

En esta fase se produce información, la cual debe ser documentada conforme a las normas y estándares definidos por la organización y en caso de encontrar alguna inconsistencia en los requisitos se deberá llevar a cabo un proceso de elicitación adicional que permita recolectar la información faltante (Pohl, 2010).

- **Documentación y negociación:**

La documentación obtenida con los requisitos deberá ser analizada en busca de conflictos, los cuales en caso de existir deberán ser solventados llevando a cabo actividades de negociación. Además, tanto los conflictos como su resolución deberán ser documentados de acuerdo a las normas establecidas (Pohl, 2010).

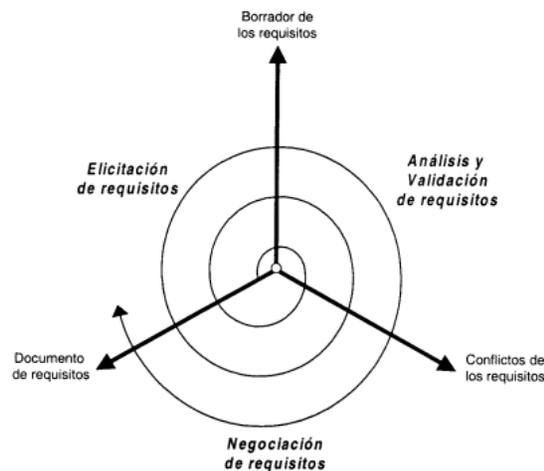
- **Elicitación y Negociación:**

Durante el proceso de elicitación de requisitos, los stakeholders pueden encontrarse con diferentes conflictos relacionados con los requisitos, para lo cual se hace evidente llevar a cabo una

negociación que permita llegar acuerdos y soluciones posibles de modo que se garantice la satisfacción de las partes interesadas (Pohl, 2010).

#### 2.1.4.2.El modelo espiral

El modelo espiral propuesto por (Sawyer, Sommerville, & Viller, 1997), consta de tres actividades que se ejecutan en forma iterativa, ya que conforme incrementa el número de iteraciones, también lo hace el nivel de detalle y calidad de los requisitos. Sin embargo, entre más iteraciones se realiza mayor es el impacto sobre el costo del proyecto.



**Figura 7.** Modelo Espiral de Ingeniería de Requisitos.

Fuente: (Sawyer, Sommerville, & Viller, 1997)

#### **Elicitación de Requisitos**

En esta fase se debe identificar todos los stakeholders del proyecto y se debe recabar información que permita comprender el dominio del problema así como las necesidades de los interesados. Sin embargo, dada la informalidad en que se ejecuta este proceso, los requisitos pueden no estar especificados adecuadamente, pueden ser incorrectos o inconsistentes.

### **Análisis y Validación de requisitos**

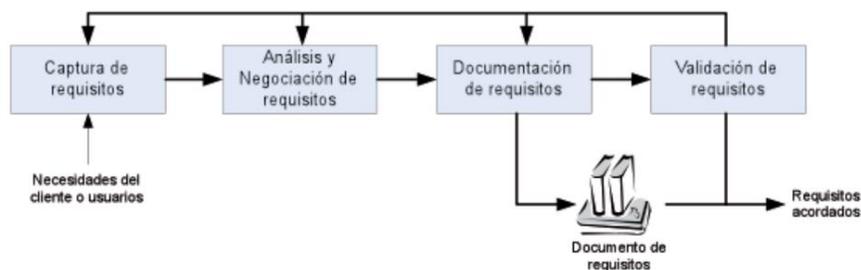
Los requisitos obtenidos en la fase anterior deberán ser entonces integrados y sujetos de análisis por parte de los Stakeholders con el fin de detectar cualquier inconveniente que se pudiera haber presentado en los mismos.

### **Negociación de requisitos**

En esta fase se debe llegar a acuerdos que permitan dar solución a los problemas detectados en los requisitos elicitados, siendo necesaria en muchos de los casos acudir a las fuentes de los requisitos para recuperar cualquier tipo de información faltante.

#### **2.1.4.3.El modelo SWEBOK**

SWEBOK (Software Engineering Body of Knowledge) es un proyecto en el cual se abordan diversas áreas de conocimiento (Bourque, Dupuis, & Abran , 1999); entre estas áreas se encuentra la Ingeniería de Requisitos dentro de la cual se propone un modelo de procesos (Ver Figura 8).



**Figura 8.** Modelo SWEBOOK de procesos de ingeniería de requisitos.

Fuente: (Bourque, Dupuis, & Abran , 1999)

### **Captura de Requisitos**

También conocida como Elicitación de requisitos esta actividad es la primera indispensable para lograr entender los problemas que se debe resolver; es una actividad fundamentalmente humana.

### **Análisis y negociación de requisitos**

Esta actividad tiene como objetivo detectar y resolver los conflictos entre los requisitos, y a la vez determinar el alcance del sistema y como va a interactuar con su entorno para poder transformar los requisitos del usuario en requisitos de software.

### **Documentación de Requisitos**

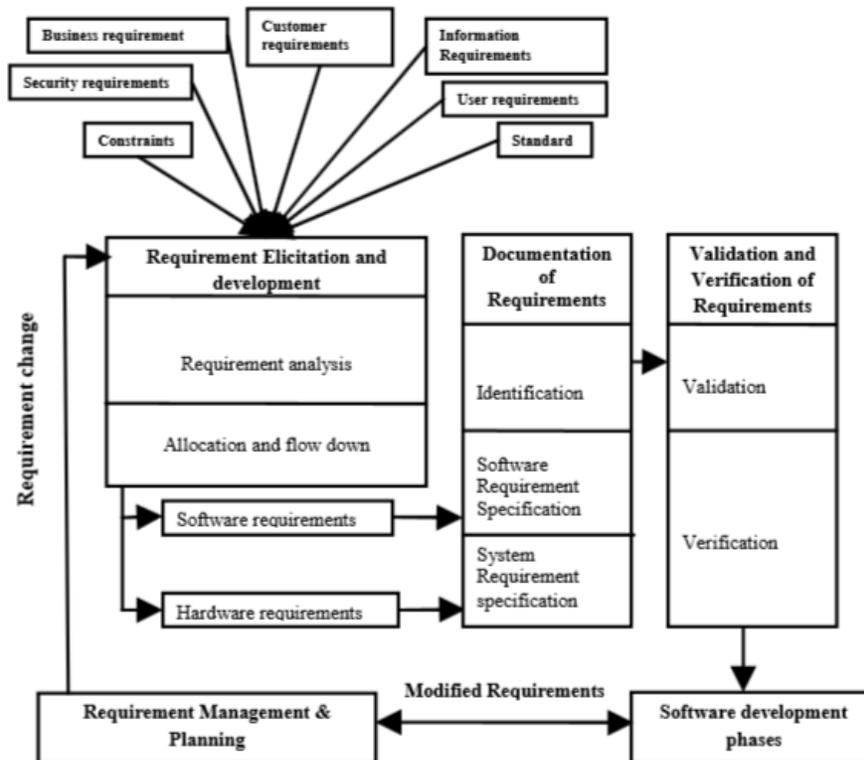
El documento de requisitos es la herramienta para el registro y la comunicación de los requisitos. Dentro de las actividades relacionadas con la documentación, se encuentran la gestión de requisitos para los aspectos como el control de versiones de cambios en los requisitos.

### **Validación de Requisitos**

En esta actividad se comprueban los documentos de requisitos con el fin de detectar ambigüedades o conflictos que no fueron detectados en el análisis para comprobar que los requisitos siguen las normas de calidad establecidas; es decir esta actividad combina la validación y verificación en una sola actividad.

#### **2.1.4.4.Otros Modelos**

(Pandey & Surman, 2010), proponen un modelo de proceso de ingeniería de requisitos, que se compone principalmente de cuatro fases: elicitación y desarrollo de requisitos, documentación de requisitos, validación y verificación de requisitos, y finalmente gestión y planificación de requisitos (ver figura 9).



**Figura 9.** Modelo de Procesos de Ingeniería de Requisitos

Fuente: (Pandey & Surman, 2010)

### **Elicitación y desarrollo de requisitos**

Esta fase se centra principalmente en examinar y elicitar los requisitos así como los objetivos del sistema considerando varios puntos de apreciación (por ejemplo, cliente, usuarios, restricciones, entorno operativo del sistema, comercio, marketing y estándares, etc.). Normalmente, la especificación de los requisitos del sistema comienza con la observación y la entrevista de personas.

Sin embargo, el analista del sistema puede malinterpretar las necesidades del usuario. Es por eso que esta fase además de la recopilación de requisitos, se encarga de establecer los estándares y las restricciones, mismas que desempeñan un papel importante en el desarrollo de sistemas.

### **Documentación de requisitos**

Después de recopilar los requisitos, se prepara un documento formal que contiene una descripción completa del comportamiento externo del sistema de software. El proceso de desarrollo de requisitos es la actividad de determinar qué funcionalidad del sistema realizará el software.

Los requisitos no funcionales se combinan junto con los requisitos funcionales en la especificación de requisitos del software con la ayuda del flujo hacia abajo, la asignación y la derivación. Se establecerá una especificación de requisitos de software para cada subsistema de software, elemento de configuración de software o componente que forma parte de esta fase. La documentación de requisitos incluye la identificación de requisitos y la especificación de requisitos.

### **Verificación y validación de requisitos**

Cuando todos los requisitos están descritos y especificados en el SRS, todos los involucrados tienen que ponerse de acuerdo sobre su naturaleza. Uno debe asegurarse de que se establecen los requisitos correctos (validación) y estos requisitos se establecen correctamente (verificación). Las actividades de validación y verificación incluyen la validación de los requisitos del sistema contra los requisitos sin procesar y la verificación de la exactitud de la documentación de los requisitos del sistema.

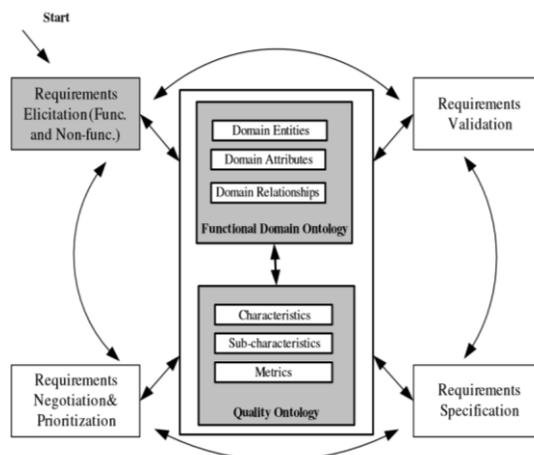
En la verificación y validación de requisitos, el mecanismo de seguimiento de requisitos puede generar un seguimiento de auditoría entre los requisitos de software y el código finalmente probado. El resultado es un documento formal que incluye una línea de base de los requisitos de software acordados.

### **Gestión y planificación de requisitos**

Esta fase controla y rastrea los cambios de los requisitos acordados, las relaciones entre los requisitos y las dependencias entre los documentos de requisitos y otros documentos producidos durante el proceso de ingeniería de sistemas y software. La gestión de requisitos es una actividad continua que puede realizarse después del desarrollo y durante el mantenimiento porque los requisitos pueden continuar cambiando la trazabilidad de los requisitos es una parte de la gestión de requisitos, donde se describe el requisito además de hacer un seguimiento del ciclo de vida del mismo y sus relaciones con otros artefactos de desarrollo tanto en dirección hacia adelante como hacia atrás.

(Taiseera Hazeem , Falcone Sampaio, Dabhi, & Loucopoulos , 2006), proponen un modelo de procesos de ingeniería de requisitos enfocado en la elicitación de requisitos y una herramienta para capacitar a los analistas de requisitos con un repositorio de conocimientos que ayuda en el proceso de formular las preguntas correctas y capturar la especificación precisa de requisitos durante las entrevistas de obtención.

El enfoque se basa en la aplicación de ontologías para sustentar la actividad de elicitación (Ver Figura 10). En particular, la ontología ayuda a garantizar que todos los requisitos de calidad relevantes relacionados con un dominio se aborden durante las entrevistas (reduciendo así los problemas), y también ayuda en el proceso de creación de criterios de calidad estandarizados que pueden ser uniformes, utilizado por diferentes analistas durante las entrevistas (reduciendo así los problemas de comprensión).



**Figura 10.** Proceso de IR basado en Ontologías

Fuente: (Taiseera Hazeem , Falcone Sampaio, Dabhi, & Loucopoulos , 2006)

La propuesta consiste en desarrollar un método de obtención de requisitos impulsado por la ontología, guiado por un modelo de calidad estándar (ISO / IEC 9126). El modelo de calidad se codifica como ontología de calidad y se automatiza mediante una herramienta de obtención de requisitos, que ayuda a abordar los factores de calidad durante las entrevistas de obtención este procesos se encuentra detallado en la Figura 10.

Como herramienta para sustentar su propuesta utilizaron “Protégé”, que es una herramienta de elicitación guiada por ontología de calidad que respalda el proceso al recuperar de la base de conocimientos de ontología y las características de calidad relevantes adjuntas al dominio.

### 2.1.5. Métodos de Investigación

Los métodos de investigación, según lo expuesto por (Ouhbi, Idri, Fernandez-Aleman, & Toval, 2013), (Condori-Fernandez, et al., 2009) y (Tonella, Torchiano, Du Bois, & Systä, 2007) se pueden clasificar en estudio de caso, encuesta, experimento, teoría y revisión. Los más relevantes para la elaboración de este trabajo se detallan en las siguientes secciones.

### **2.1.5.1. Estudio de caso**

Es una herramienta de investigación que permite medir y registrar la conducta de los involucrados en el fenómeno que se desea estudiar mediante la obtención de datos que pueden ser cualitativos como cuantitativos (Martínez Carazo, 2006).

### **2.1.5.2. Encuestas**

De acuerdo con (Besrou, AB Rahim, & Dominic, 2016), la encuesta es un instrumento para recopilar información y datos de personas para describir su experiencia, conocimiento, actitud y perspectiva.

### **2.1.5.3. Revisión Sistemática de Literatura**

La revisión sistemática de literatura (SLR) es un estudio secundario que proporciona una manera de sintetizar los estudios primarios disponibles en literatura empírica sobre un tema en particular (Kitchenham & Charters, 2007). La SLR se distingue de una revisión tradicional, ya que sigue un procedimiento riguroso para la búsqueda y selección de estudios para la revisión.

Es un proceso metódico y meticuloso de recopilar y compilar estudios empíricos de calidad publicados, basados en un protocolo sistémico para reducir el sesgo y proporcionar transparencia al proceso. El proceso está documentado formalmente y por lo tanto es repetible (Bano, Zowghi, & Ikram, 2014).

### **2.1.5.4. Mapeo Sistemático de Literatura**

Es un estudio que tienen como objetivo encontrar y clasificar los estudios en un área específica. Utilizan el mismo procedimiento de búsqueda y selección que el SLR convencional, pero se centran en la clasificación de los estudios resultantes para proporcionar un mapa de la investigación en esa área (Bano, Zowghi, & Ikram, 2014).

### **2.1.5.5. Estudio Terciario**

Se denomina estudio terciario a un SLR que agrega los datos y la información de una serie de SLR existentes sobre un tema específico o área de investigación. Un estudio terciario tiene como objetivo proporcionar información sobre las SLR publicadas sobre un tema específico y tabular la información sobre las mismas, por ejemplo el número de SLR publicadas, su calidad y el enfoque de estas SLR dentro del tema de interés (Kitchenham, y otros, 2010).

## **2.2. ESTADO DEL ARTE**

En base a la problemática en torno a la Ingeniería de Requisitos identificada; como estudio preliminar se intentó conocer la factibilidad de la creación de este modelo, para lo cual se realizó una revisión inicial de literatura.

Este estudio se basó en algunas actividades de las guías propuestas por (Kitchenham & Charters, 2007). El protocolo reestructurado propone la definición de: (1) Planteamiento del objetivo de búsqueda, (2) conformación del grupo de control (GC), (3) Construcción de la cadena de búsqueda, (4) selección de estudios primarios.

### **2.2.1. Planteamiento del objetivo de búsqueda**

Con el propósito de determinar el estado del arte para la presente investigación, se propuso en primera instancia: Construir una estructura que provea una vista panorámica de los problemas en torno a la IR, sus causas y las características de las soluciones propuestas para dichos problemas. Con el fin de estructurar el objetivo de búsqueda como alcanzable se definieron los siguientes criterios de inclusión y exclusión:

Criterios de inclusión. El artículo:

- Describe investigaciones que se han realizado en torno a la Ingeniería de requisitos.

- Menciona explícitamente los problemas inherentes a la Ingeniería de Requisitos.
- Menciona las técnicas aplicadas para medir la calidad de los requisitos elicitados.
- Detalla propuestas realizadas en miras a tratar los problemas que se presentan en la IR.

Criterios de exclusión. Serán excluidos:

- Estudios previos al 2014.
- Short papers y resúmenes de conferencias.
- Artículos que solo reporten detalles meramente técnicos de soluciones a la problemática de la IR.
- Artículos escritos en un idioma diferente al inglés.
- Artículos no accesibles en texto completo.

### **2.2.2. Conformación del grupo de control (GC)**

El grupo de control (GC) de literatura es un conjunto de estudios que cumplen con los criterios definidos en la sección anterior. El GC se conforma con el propósito de ser la fuente de la cual se seleccionan los términos que servirán como base para la conformación de la cadena de búsqueda. Para conformar el GC se tuvo la participación de dos investigadores, los cuales propusieron estudios conocidos y de buena fuente como candidatos para conformar el GC. Por medio de una validación cruzada se seleccionaron 4 artículos de los propuestos por los investigadores para el GC. Tabla 2 muestra los estudios que integran el grupo de control así como las palabras clave identificadas en cada estudio.

**Tabla 2***Estudios del Grupo de Control*

No.	TÍTULO	PALABRAS CLAVE
EC1	A Review of Practice and Problems in Requirements Engineering in Small and Medium Software Enterprises in Thailand	Software Engineering, Requirements Engineering, SMEs, Process Improvement
EC2	A Survey on Current Requirement Process Practices in Software Companies and Requirement Process Problems	SRS, Requirement Process, Software Requirement Engineering, Requirement Elicitation, Requirement Specifications
EC3	Towards Building Knowledge on Causes of Critical Requirements Engineering Problems	NaPiRE; Knowledge Building; Requirements Engineering; Problems; Causes; Causal Analysis
EC4	Towards Guidelines for Preventing Critical Requirements Engineering Problems	Guidelines; problem prevention; defect prevention; requirements engineering

### 2.2.3. Construcción y afinación de la cadena de búsqueda

La creación de la cadena de búsqueda parte de un análisis profundo de los artículos del grupo de control. En este proceso los investigadores encontraron términos similares entre estudios y términos particulares de cada artículo, todos consecuentes con el objetivo del SMS. A paso seguido se realizó una validación cruzada entre los investigadores para clasificar los términos por afinidad, de lo que se identificaron 5 contextos relacionados con el objetivo del SMS: Entorno, Problema, Propuesta, Propósito y Herramienta.

Para conformar la cadena de búsqueda se utilizaron los términos identificados; dichos términos se relacionaron a través de los siguientes operadores: OR y AND para combinar o agregar diferentes contextos. La cadena de búsqueda fue probada iterativamente en la base SCOPUS. En primera instancia se conformó la siguiente cadena:

ALL ({requirements engineering} OR {elicitation}) AND ALL ({problems} OR {RE problems}) AND ALL ({causes} OR {impact}) AND ALL ({model} OR {elicitation} OR {practices}) AND ALL ({review} OR {survey})

Sin embargo, esta cadena retornaba un número exagerado de investigaciones de los cuales solamente 2 eran estudios del GC; por lo cual fue necesario realizar varias pruebas con diferentes combinaciones de palabras claves, agregando y quitando términos, hasta que se encontró la cadena de búsqueda ideal:

ALL ({requirements engineering}) AND ALL ({quality}) OR {issues}) AND ALL ({problems}) AND ALL ({review} OR {survey}) AND ALL ({guidelines} OR {practices})

#### **2.2.4. Proceso de Selección de Estudios Primarios y Extracción de Datos**

A partir de la conformación de la cadena de búsqueda se obtuvieron 211 estudios en la base digital SCOPUS, entre los cuales aparecen los 4 estudios que conforman el GC.

Para estudiar la factibilidad del objetivo principal de la búsqueda fueron aplicados filtros en los estudios obtenidos en SCOPUS. Los filtros aplicados corresponden a los criterios de inclusión y exclusión antes establecidos en el primer punto de esta sección.

Como resultado de esta actividad se obtuvieron 36 estudios (denominados estudios seleccionados). A paso seguido se accedió al texto completo de los estudios, los cuales fueron revisados por al menos dos investigadores y a través de una validación cruzada seleccionaron finalmente 6 estudios, que fueron denominados estudios primarios.

Con el propósito de establecer el estado del arte, la información más relevante para la investigación contenida en los estudios primarios fue extraída.

## **2.2.5. Determinación de las características del estado del arte**

### **2.2.5.1. Resumen de los estudios primarios**

#### **EP1 (Liebel, Tichy, Knauss, Ljungkrantz, & Stieglbauer, 2018)**

En este documento se presenta un estudio de caso en dos compañías, un fabricante de equipo original (OEM) y un proveedor, dentro del dominio automotriz. Se estudiaron los desafíos actuales en la ingeniería de requisitos (RE) con respecto a la estructura de la organización y la comunicación mediante la recopilación de datos a través de 14 entrevistas con personas de las áreas de ingeniería de software integrado, ingeniería de sistemas e ingeniería de software de aplicación. Se extrajeron siete desafíos clave como la falta de conocimiento del producto, recursos insuficientes para entender y mantener los requisitos, falta de comunicación y canales de retroalimentación, entre otros. Finalmente se propuso como solución a estos problemas el aplicar un enfoque con metodologías ágiles y un manejo de procesos más estricto como revisiones formales. Sin embargo, este trabajo no ofrece una solución novedosa como tal, solo establece recomendaciones en base a los resultados obtenidos de las encuestas. Además, los autores mencionan explícitamente que los problemas encontrados pueden llegar a diferir en otros dominios diferentes al automotriz por lo que sus resultados no son aplicables a toda la industria del software.

#### **EP2 (Ambreen, Ikram, Usman, & Niazi, 2016)**

Este estudio presenta un estado del arte de la ingeniería de requisitos, junto con varias tendencias y oportunidades de investigación empírica de IR, se llevó a cabo un mapeo sistemático para sintetizar el trabajo empírico realizado en IR. Se utilizaron cuatro bases de datos principales IEEE, ScienceDirect, SpringerLink y ACM e identificaron 270 estudios primarios hasta el año 2012. Un análisis de los datos extraídos de estudios primarios mostró que el trabajo de investigación empírica en IR está en aumento desde el año 2000. La elicitación de requisitos con

el 22% del total de estudios, análisis de requisitos con 19% y proceso de IR con 17% son las principales áreas de enfoque de la investigación empírica de IR. Los requisitos no funcionales se encontraron como el área emergente más investigada. La mayoría de los estudios (50%) utilizaron un método de investigación de estudio de caso seguido de experimentos (28%), mientras que los informes de experiencia son pocos (6%). Una tendencia común en casi todas las subáreas de IR es proponer nuevas guías, técnicas y procesos. Sin embargo, todos éstos hallazgos no representan la realidad que se vive actualmente ya que solo considera estudios hasta 2012, es decir, estudios de más de 6 años de antigüedad.

### **EP3 (Méndez Fernández, Tießler, Kalinowski, Felderer, & Kuhrmann, 2017)**

En este trabajo, los autores proponen y validan un enfoque basado en la evidencia para evaluar riesgos en la IR utilizando datos de empresas sobre problemas, causas y efectos.

Se utilizaron datos de encuestas realizadas a 228 compañías y se construyó una red probabilística que respalda el pronóstico de fenómenos de IR específicos del contexto. Se implementó este enfoque utilizando hojas de cálculo para respaldar una evaluación de riesgos ligera y se realizó una primera validación entrevistando a 6 profesionales de diferentes compañías. Los resultados de la validación mostraron potencial de mejora, pero también confirman que la propuesta de solución es bastante madura e idónea para su transferencia a la práctica y su aplicación en una evaluación a gran escala. Esta propuesta es de gran apoyo para tratar los problemas relacionados a la ingeniería de requisitos pero aún es muy temprano y el enfoque no ha sido probado lo suficiente en la práctica como para constituirse en una verdadera solución para tratar los problemas en la IR.

**EP4 (Méndez Fernández, y otros, 2017)**

En este artículo, se informa sobre el análisis cualitativo realizado con los datos obtenidos de 228 empresas que trabajan en 10 países en varios dominios y se muestra qué problemas contemporáneos enfrentan los profesionales en torno a la ingeniería de requisitos. Con este fin, se analizaron 21 problemas derivados de la literatura con respecto a su relevancia y criticidad en la dependencia de su contexto, y se complementó este panorama con un análisis de causa-efecto que muestra las causas y los efectos que rodean los problemas más críticos. Los resultados obtenidos permiten comprender de mejor manera qué problemas existen y cómo se manifiestan en entornos prácticos. Por lo tanto, los autores brindaron un primer paso para fundamentar las contribuciones a IR sobre observaciones empíricas que, hasta ahora, estaban dominadas solo por la sabiduría convencional. Sin embargo, aún no se ha hecho un uso adecuado de los resultados del análisis de causa-efecto, ya que estos podrían servir para establecer un modelo de IR con recomendaciones prácticas sobre cómo mejorar el IR en respuesta a ciertas situaciones específicas del contexto.

**EP5 (Kalinowski, y otros, 2017)**

En este estudio se propuso un nuevo enfoque DCA (Análisis causal de defectos) que utiliza datos obtenidos de 74 organizaciones brasileñas sobre las causas de los problemas de IR y se construyó una red bayesiana. Este enfoque DCA utilizó la inferencia diagnóstica de la red bayesiana para admitir sesiones DCA. Se evaluó el enfoque aplicando un modelo para la transferencia de tecnología a la industria y se realizaron tres evaluaciones consecutivas: (i) en la academia, (ii) con representantes de la industria del Centro de proyectos Fraunhofer en la UFBA, y (iii) en un estudio de caso industrial en el Banco Nacional de Desarrollo de Brasil (BNDES). Se recibieron comentarios positivos en las tres evaluaciones y los datos de las organizaciones se consideraron útiles para determinar las causas principales. Sin embargo, no se propusieron mejoras

al proceso de IR ni mucho menos soluciones para tratar los problemas ya que el estudio se enfocó meramente en determinar causas a través de enfoque DCA.

#### **EP6 (Ilyas & Zahra, 2016)**

Este trabajo exhibe las prácticas actuales del proceso de requisitos en las empresas. Se llevó a cabo una encuesta de diez empresas con la ayuda de un cuestionario. Luego, los resultados del cuestionario se compararon y examinaron para descubrir qué estándares estaban siguiendo las empresas y se realizó un esfuerzo para evaluar los criterios actuales de las empresas paquistaníes con respecto a proceso de ingeniería de requisitos. Se determinó además que los problemas organizativos tienen un gran impacto en el proceso de requisitos en comparación con los problemas técnicos. La falta de habilidades, la poca participación de los usuarios previstos, la comprensión deficiente del usuario, el desconocimiento del dominio y los recursos inadecuados se determinaron como factores que tienen un profundo impacto en el desarrollo de software exitoso. Sin embargo, más allá de conocer cuáles son los problemas de las empresas y sus prácticas de IR, no se propuso ningún tipo de solución o enfoque que permita mejorar el proceso de IR.

#### **2.2.5.2. Características del estado del arte**

Se identificó que en muchos casos los autores discrepan entre si un término debería ser considerado como un problema o causa, lo cual dificulta establecer claramente cuáles son los problemas y causas que inciden sobre la IR. Los métodos de investigación empleados en los estudios primarios son los estudios de caso, surveys y revisiones literarias. En lo referente a la contribución presentada por los estudios, varios de éstos presentan guías, herramientas y modelos. Sin embargo, ninguna de estas contribuciones ha sido probada lo suficiente como para constituirse en una solución definitiva a los problemas que se presentan en el proceso de ingeniería de requisitos.

## 2.3. METODOLOGÍA DE LA INVESTIGACIÓN

### 2.3.1. Metodología Design Science

Design Science es una metodología que comprende un conjunto de técnicas y perspectivas sintéticas y analíticas (que complementan las perspectivas positivista, interpretativa y crítica) para realizar investigaciones en el área de ciencias de la computación (Vaishnavi, Kuechler, & Petter, 2004).

La metodología design science implica: la creación de nuevos conocimientos a través del diseño de artefactos novedosos o innovadores (cosas o procesos) y el análisis del uso del artefacto y / o performance con reflexión y abstracción (Vaishnavi, Kuechler, & Petter, 2004) (Peffer, Tuunanen, Rothernberger, & Chatterjee, 2007). Los artefactos creados en el proceso de investigación en ciencias del diseño incluyen, constructos, modelos, métodos e instancias (Winter, 2008).

Dado que el propósito de este trabajo es la creación de un modelo, design science es la metodología más adecuada para llevar a cabo esta tarea. El proceso se describe en la Figura 11 y se detalla a continuación.



*Figura 11.* Design Science

#### 2.3.1.1. Identificación del problema y motivación

En esta fase se define el problema de investigación y se justifica el valor de la solución. Justificar el valor de la solución cumple dos objetivos motiva al investigador y a la audiencia a

perseguir la solución y aceptar los resultados, además ayuda a entender el razonamiento asociado con la comprensión del problema por parte del investigador. Los recursos necesarios para el desarrollo de esta actividad incluyen el conocimiento del estado del problema y la importancia de su solución.

Esta fase se llevó a cabo a través de una revisión de literatura básica en la base digital IEEE Xplore que permitió identificar claramente el problema, sus causas y sus consecuencias.

#### **2.3.1.2. Definición de los objetivos de la propuesta**

En esta fase se deducen los objetivos de una solución a partir de la definición del problema y el conocimiento de lo que es posible y factible. Los objetivos pueden ser cuantitativos, como los términos en los que una solución deseable sería mejor que aquellas ya existentes, o cualitativos como una descripción de como se espera que un nuevo artefacto apoye las soluciones a problemas que hasta ahora no se han abordado.

Los recursos necesarios para esto incluyen el conocimiento del estado de los problemas y las soluciones actuales si las hay y su eficacia.

Previo a la definición de los objetivos de la propuesta, se llevó a cabo una revisión de literatura preliminar basada en las guías de revisión sistemática de literatura propuestas por (Kitchenham & Charters, 2007).

#### **2.3.1.3. Desarrollo del artefacto**

En esta fase se crea el artefacto. Dicho artefacto es potencialmente un constructo, modelo, método o ejemplificación (cada uno definido ampliamente). Esta actividad incluye determinar la funcionalidad deseada del artefacto y su arquitectura para luego crear el artefacto real. Los recursos necesarios para pasar de los objetivos al diseño y desarrollo incluyen el conocimiento de la teoría que se puede aplicar en una solución.

En esta fase se llevó a cabo un estudio terciario de literatura, cuyos resultados permitieron identificar problemas y causas presentes en el proceso de Ingeniería de requisitos actual. Con la finalidad de dar solución a estos problemas se estructuró un modelo de procesos de ingeniería de requisitos alternativo.

#### **2.3.1.4.Demostración del artefacto**

Demostrar el uso del artefacto para dar solución a una o más instancias del problema. Esto puede involucrar su uso en experimentación, simulación, estudio de caso, prueba u otra actividad apropiada. Los recursos requeridos para la demostración incluyen el conocimiento efectivo de cómo utilizar el artefacto para resolver el problema. Para cumplir con esta fase se llevó a cabo un entrenamiento a los estudiantes de la asignatura de Desarrollo de Sistemas de la Universidad de las Fuerzas Armadas ESPE, con el objetivo de explicar el proceso que conlleva el modelo de ingeniería de requisitos estructurado de modo que se pueda aplicar a un caso de estudio posterior.

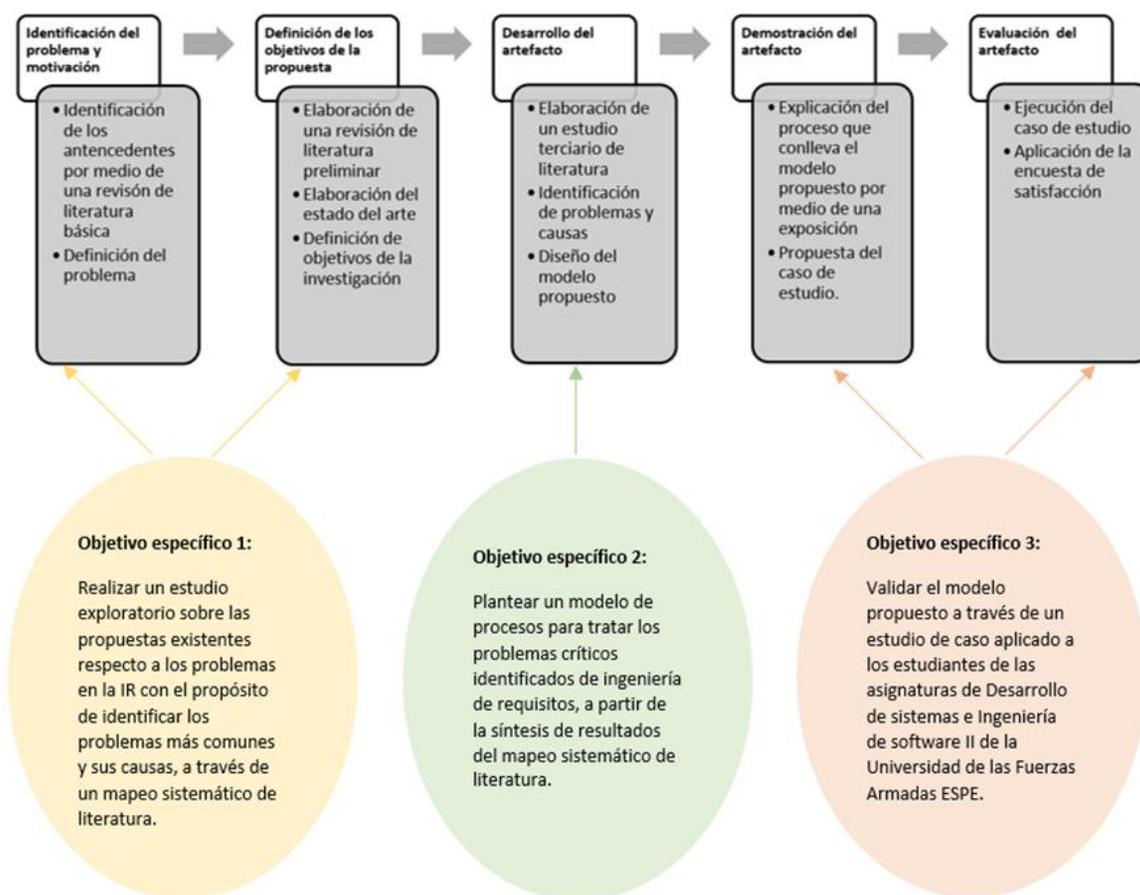
#### **2.3.1.5.Evaluación del artefacto**

Observar y medir que tan bien el artefacto se desempeña en la solución del problema. Esta actividad implica comparar los objetivos de una solución con los resultados reales observados del uso del artefacto en la demostración. Requiere conocimiento de métricas relevantes y técnicas de análisis. Se podría incluir medidas de rendimiento cuantitativas como los presupuesto o ítems producido, los resultados de encuesta de satisfacción, comentarios de los clientes o simulaciones.

Una vez cumplida la demostración del artefacto los estudiantes aplicaron el modelo, resultando de esto un documento de especificación de requisitos, el cual fue comparado con el documento de requisitos de referencia elaborado previamente. Para esto, se aplicó la métrica utilizada en (Sevilla, Collazos, Zapata, & Giraldo , 2016), la cual permitió medir la calidad del documento de especificación de requisitos resultante.

### 2.3.2. Design Science aplicado

En la Figura 12 se muestran las fases y actividades de la metodología aplicadas en la presente investigación.



**Figura 12.** Design Science aplicado a la investigación

## **CAPÍTULO III**

### **MAPEO SISTEMÁTICO DE LITERATURA**

En este capítulo se detalla el proceso del Mapeo Sistemático de Literatura, el cual se realizó como una extensión de la revisión de Literatura descrita en el capítulo anterior, con el fin de dar respuesta a las preguntas de investigación planteadas en torno a los problemas de la Ingeniería de Requisitos.

#### **3.1. PROPÓSITO DE LA INVESTIGACIÓN**

Construir una estructura que provea una vista panorámica de los problemas en torno a la IR, sus causas y las características de las soluciones propuestas para dichos problemas.

#### **3.2. DEFINICIÓN DE LAS PREGUNTAS DE INVESTIGACIÓN**

El SMS propuesto en esta investigación se enfoca en dar respuesta a las preguntas de investigación que se describen a continuación:

RQ1: ¿Cuál es la frecuencia de los estudios realizados cuyo enfoque es la IR?

RQ2: ¿Qué problemas y causas son más habituales dentro de la IR?

RQ3: ¿Cuáles son los métodos de investigación empleados en los trabajos que se han llevado a cabo en torno a la IR?

RQ4: ¿Cuál es la tendencia en torno a las propuestas realizadas con miras a tratar los problemas vinculados a la IR?

#### **3.3. DEFINICIÓN DE CRITERIOS DE INCLUSIÓN Y EXCLUSIÓN**

Los criterios de inclusión que se tomaron en cuenta para la presente SMS son los siguientes:

- Artículos en los cuales se reporten estudios de revisión de literatura sobre la frecuencia en la que se han presentado estudios en torno a la IR.

- Artículos en los cuales se reporten estudios de revisión de literatura sobre la problemática en torno a la Ingeniería de requisitos y sus posibles causas.
- Artículos en los cuales se reporten estudios de revisión de literatura sobre calidad de requisitos en Ingeniería de Software.
- Artículos en los cuales se reporten estudios de revisión de literatura sobre buenas prácticas en la ingeniería de requisitos.
- Artículos en los cuales se reporten estudios de revisión de literatura que sean actuales (a partir del año 2013) y hayan sido publicados en una revista o congreso de alto impacto.

Los criterios de exclusión que se tomarán en cuenta para la presente SMS serán los siguientes:

- Artículos en los cuales se reporten estudios sobre ingeniería de requisitos, pero que no indique explícitamente que el método de investigación fue una revisión de literatura.
- Artículos en los que se reporten revisiones de literatura sobre cualquier campo de la Ingeniería de Requisitos, pero que hagan énfasis únicamente a las herramientas utilizadas en el proceso.

### **3.4. DEFINICIÓN DE LA ESTRATEGIA DE BÚSQUEDA**

La estrategia de búsqueda constituye el procedimiento a través del cual se determina el conjunto de estudios que constituyen la fuente fiable de la cual se obtiene la información que da respuesta a las preguntas de investigación. Para el caso de esta investigación, la estrategia planteada consta de tres actividades: (1) Configuración del GC, (2) elaboración de la cadena de búsqueda y, (3) búsqueda de estudios candidatos.

### 3.4.1. Configuración del GC

El grupo de control (GC) de literatura es un conjunto de estudios que cumplen con los criterios definidos en la sección 3.3. El GC se conforma con el propósito de ser la fuente de la cual se seleccionan los términos que servirán como base para la conformación de la cadena de búsqueda. Los estudios del GC son seleccionados en consenso por los participantes de la investigación (Zhang, Babar, & Tell, 2011). Para conformar el GC se tuvo la participación de tres investigadores, los cuales propusieron estudios conocidos y de buena fuente como candidatos para conformar el GC. Por medio de una validación cruzada se seleccionaron 6 artículos de los propuestos por los investigadores para el GC. La Tabla 3 describe los artículos que forman parte del grupo de control.

**Tabla 3.**

*Artículos que conforman el Grupo de Control*

No.	Título	Palabras clave
EC1	A Review of Practice and Problems in Requirements Engineering in Small and Medium Enterprises in Thailand	Software Engineering, Requirements Engineering, SMEs, Process Improvement
EC2	A study on the software requirements elicitation issues - Its causes and effects	Software requirements engineering, software requirements elicitation, elicitation issues, systematic literature review, cause-and-effect
EC3	A Survey on Current Requirement Process Practices in Software Companies and Requirement Process Problems	SRS, Requirement Process, Software Requirement Engineering, Requirement Elicitation, Requirement Specifications
EC4	State-of-art practices to detect Inconsistencies and Ambiguities from Software Requirements	Inconsistency, Ambiguity, Requirement Engineering

**CONTINÚA**

No.	Título	Palabras clave
EC5	Towards Building Knowledge on Causes of Critical Requirements Engineering Problems	NaPiRE; Knowledge Building; Requirements Engineering; Problems; Causes; Causal Analysis
EC6	Towards Guidelines for Preventing Critical Requirements Engineering Problems	Guidelines; problem prevention; defect prevention; requirements engineering

### 3.4.2. Construcción de la cadena de búsqueda

La construcción de la cadena de búsqueda inicia con el análisis detallado de los artículos que conforman el GC, con el objetivo de identificar términos significativos relativos a la investigación. Cada investigador identificó términos comunes entre artículos y términos particulares de cada artículo, todos consecuentes con el objetivo del SMS. A paso seguido se realizó una validación cruzada entre los investigadores para clasificar los términos por afinidad, de lo que se identificaron 5 contextos relacionados con el objetivo del SMS: Herramienta, Propuesta, Propósito, Problema y Entorno. Seguidamente, se construyó una matriz que conjugó la actividad manual de los investigadores con el uso de herramientas informáticas, en este caso el software CATMA, que permite el análisis cualitativo de datos. La matriz incluye: (1) el total de repeticiones de un término por artículo y la suma total entre todos los artículos, (2) el número de artículos en que aparece un mismo término y, (3) el número de investigadores que llegaron a la conclusión de que el término era de especial relevancia (Ver Figura 13).

CONTEXTO	PALABRA	CONTEO INDIVIDUAL						TOTAL	# ARTICULOS EN QUE SE REPITE	# INVESTIGADORES QUE VEN
Entorno	elicitation	9	71	1	4		6	91	5	2
Entorno	RE process	1	2	1	4		4	12	5	3
Entorno	requirements engineering	56	22	3	4	5	4	94	6	3
Problema	impact	3	3	7	7	1	4	25	6	2
Problema	problems	24	23	16	6	65	57	191	6	3
Propuesta	practices	12	16	19	2	3	2	54	6	2
Propósito	impact	3	3	7	7	1	4	25	6	2
Recursos	review	3	13		2	1	1	20	5	2
Recursos	survey	6	5	14	1	38	17	81	6	2

**Figura 13.** Detalle Contextos

Gracias a la matriz se pudo identificar los términos de mayor relevancia que integrarían la cadena de búsqueda, dichos términos se relacionaron a través de los siguientes operadores: OR y AND para combinar o agregar diferentes contextos. La cadena de búsqueda fue probada iterativamente en la base Scopus, tras cada ajuste hasta que se encontró la cadena de búsqueda ideal, entendiéndose como cadena ideal aquella que cumple con las tres condiciones siguientes: (i) obtiene un número de estudios razonable (cientos no miles), (ii) los títulos de los estudios que obtiene están muy relacionados con el objetivo de la búsqueda y, (iii) la mayoría de los estudios del GC son parte del conjunto de estudios que obtiene la cadena. Desde esta perspectiva la cadena de búsqueda que fue considerada la ideal obtiene 289 estudios dentro de los cuales aparecen 4 de los estudios de control (Ver Tabla 4).

**Tabla 4.**  
*Cadena de Búsqueda (Estudios recuperados)*

CADENA DE BÚSQUEDA	#RESULTADOS	# ARTICULOS GC	OBSERVACIONES
TITLE-ABS-KEY ( {requirements engineering} ) AND TITLE-ABS ( {problems} OR {issues} ) AND ALL ( {quality} ) AND ALL ( {practices} OR {guidelines} ) AND ALL ( {survey} OR {review} )	211	6	Se añadieron las palabras más relevantes de cada contexto

**CONTINÚA**

CADENA DE BÚSQUEDA	#RESULTADOS	# ARTICULOS GC	OBSERVACIONES
ALL ( {requirements engineering} ) AND ALL ( {requirements problems} OR {issues} ) AND ALL ( {quality} ) AND ALL ( {practices}OR {guidelines} ) AND ALL ( {survey} OR {review} )	2020	4	Se cambió todo a ALL
ALL ( {requirements engineering} ) AND ALL ( {problems} ) AND ALL ( {quality} ) AND ALL ( {practices} OR {guidelines} ) AND ALL ( {survey} OR {review} )	1284	6	Se eliminó issues
ALL ( {requirements engineering} ) AND ALL ( {problems} ) AND ALL ( {quality} ) AND ALL ( {requirements practices} OR {guidelines} ) AND ALL ( {survey} OR {review} )	674	4	Se puso requirements practices en lugar de practices
ALL ( {requirements engineering} ) AND ALL ( {problems} ) AND ALL ( {quality} ) AND ALL ( {practices} ) AND ALL ( {survey}OR {review} )	1013	6	Se eliminó guidelines
ALL ( {requirements engineering} ) AND ALL ( {problems} ) AND ALL ( {quality} ) AND ALL ( {practices} ) AND ALL ( {survey questionnaire} OR {review} )	841	5	Se puso survey questionnaire
ALL ( {requirements engineering} ) AND ALL ( {problems} ) AND ALL ( {quality} ) AND ALL ( {practices} ) AND ALL ( {surveys} OR {review} )	907	6	Cambio de survey por surveys
ALL( ({requirements engineering}) AND ({problems} ) AND ({quality}) AND ({practices}) AND ({surveys}))	289	4	Se borró review

### 3.4.3. Búsqueda de estudios candidatos

La búsqueda de los estudios candidatos se realizó en 4 bases digitales: Scopus, IEEE, SpringerLink y Science Direct. Sin embargo, tras probar reiteradamente la cadena de búsqueda en las bases digitales antes mencionadas, se llegó a la conclusión de que existen demasiados estudios sobre ingeniería de requisitos, motivo por el cual se tomó la decisión de realizar un estudio terciario y para este efecto se añadieron a la cadena de búsqueda los siguientes términos: “Systematic

Mapping Study” y “Systematic Literature Review” quedando como resultado la siguiente cadena de búsqueda:

**ALL( (requirements engineering) AND (problems) ) AND (quality) AND (practices) AND (surveys) AND (Systematic Mapping Study) OR(Systematic Literature Review))**

Con el ajuste realizado a la cadena de búsqueda se obtuvieron un total de 1104 estudios candidatos de las 4 bases digitales (Ver Tabla 5).

**Tabla 5.**  
*Estudios resultantes por base digital*

Base Digital	# Estudios resultantes
Scopus	71
IEEE	621
SpringerLink	223
Science Direct	189
Total	1104

### 3.5. DEPURACIÓN DE ESTUDIOS CANDIDATOS

En las bases digitales se encontraron estudios duplicados por lo que fue necesario realizar una depuración de estudios candidatos para identificar los estudios que se repetían dentro de la misma base así como entre las diferentes bases digitales (Ver Tabla 6). Esta depuración dio como resultado un total de 19 estudios duplicados.

**Tabla 6.**  
*Estudios duplicados por base digital.*

Base digital	# Estudios repetidos
Scopus	17
IEEE	8
SpringerLink	3
Science Direct	10

### 3.6. SELECCIÓN DE ESTUDIOS

La ejecución de esta fase se llevó a cabo mediante la revisión a nivel de título y resumen de cada uno de los estudios candidatos, para lo cual se contó con la participación de dos investigadores los cuales una vez revisados estos parámetros debían determinar por separado si el artículo era aprobado o rechazado.

Con el resultado de esta revisión se realizó una validación cruzada que permitió llegar a un acuerdo entre discrepancias presentadas por los revisores, lo que dio como resultado un total de 73 estudios seleccionados (Ver Tabla 7).

**Tabla 7.**  
*Estudios Seleccionados*

<b>Total de estudios candidatos</b>	<b>1104</b>
<b>Estudios seleccionados</b>	73
<b>Estudios descartados</b>	1031
<b>% Acuerdo</b>	82.19%
<b>% Desacuerdo</b>	17.80

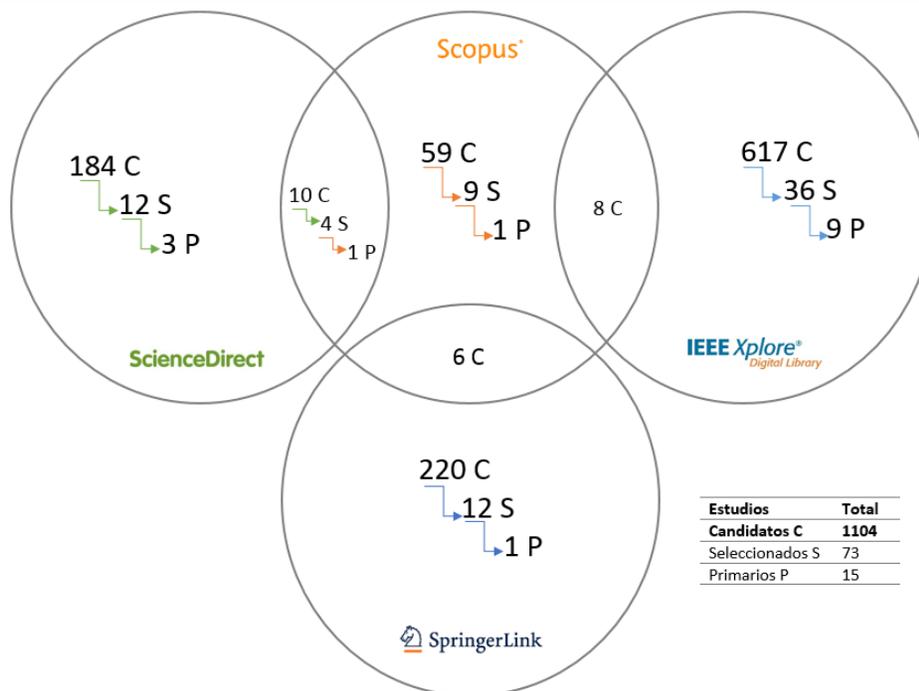
#### 3.6.1. Selección de Estudios Primarios

Para la selección de estudios primarios, en primera instancia se obtuvieron los 74 artículos (estudios seleccionados), los cuales fueron revisados a profundidad por los investigadores para determinar si estos cumplían con los criterios de inclusión y exclusión detallados en la sección 3.3 de este capítulo.

Como resultado de este proceso se obtuvieron 15 estudios primarios entre las 4 bases digitales analizadas. Cabe mencionar que en algunos casos un mismo estudio se encontraba presente en más de una base digital (Ver Tabla 8).

**Tabla 8.**  
*Estudios Primarios*

Estudio Primario	Base Digital			
	Scopus	IEEE	Science Direct	SpringerLink
EP1		X		
EP2		X		
EP3		X		
EP4		X		
EP5		X		
EP6		X		
EP7		X		
EP8		X		
EP9	X		X	
EP10			X	
EP11				X
EP12	X			
EP13			X	
EP14			X	
EP15		X		



**Figura 14.** Relación Estudios Primarios

### 3.7. EXTRACCIÓN DE RESULTADOS

En esta sección se presentan los resultados obtenidos del estudio terciario a partir de la extracción de la información más relevante de los estudios primarios, mismos que permitieron dar respuesta a las preguntas de investigación previamente definidas.

**RQ1: ¿Cuál es la frecuencia de los estudios realizados cuyo enfoque es la Ingeniería de Requisitos?**

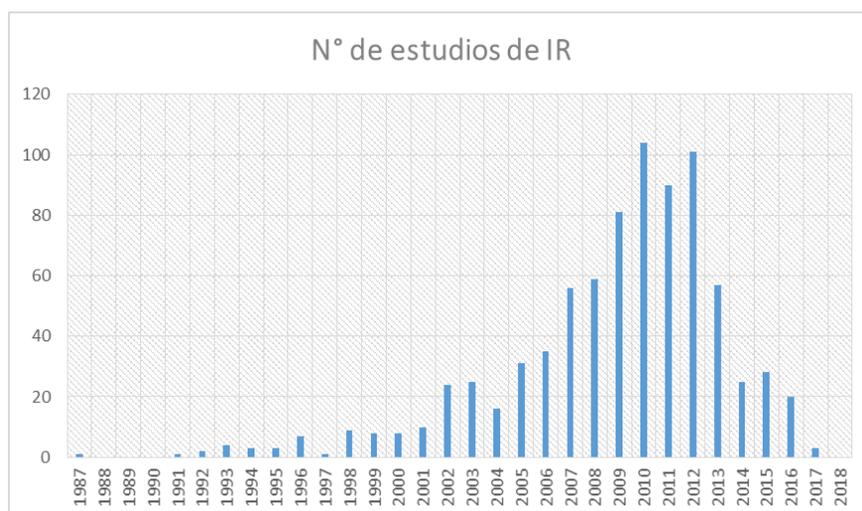
De acuerdo con los resultados de nuestro estudio se han identificado 15 estudios primarios (EP) que corresponden a Revisiones de Literatura enfocadas en los problemas más comunes de la Ingeniería de Requisitos, así como en propuestas para tratar dichos problemas.

En las 15 revisiones de literatura resultantes de nuestra investigación, se identificó un total de 1047 estudios primarios; de los cuales 823 corresponden a investigaciones en torno a la

ingeniería de requisitos que van desde el año 1987 hasta el año 2017 (Ver Tabla 9), donde se puede evidenciar que el trabajo en torno a la IR no fue significativo sino hasta el año 2000 donde se da un incremento en cuanto al número de estudios publicados, por otro lado el año que más reporta estudios es el 2010 con 105 estudios publicados (Ver Figura 15).

**Tabla 9.**  
*Estudios de Ingeniería de Requisitos.*

Revisiones de literatura	Nº de Eps	EPs de IR
EP2	81	78
EP3	66	66
EP4	71	13
EP6	85	59
EP7	22	21
EP8	87	17
EP9	54	54
EP10	104	95
EP11	270	251
EP12	28	25
EP13	18	6
EP14	21	21
EP15	140	117
<b>Total</b>	<b>1047</b>	<b>823</b>



**Figura 15.** Distribución Anual de estudios de IR.

## RQ2: ¿Qué problemas y causas son más habituales dentro de la Ingeniería de Requisitos?

En las 15 Revisiones de Literatura (estudios primarios) resultantes de nuestro estudio, un total de 64 problemas específicos de la Ingeniería de Requisitos se han identificado como se presenta en la Tabla 10.

**Tabla 10.**  
*Problemas Relacionados con la IR*

N°	Problemas	Estudios Primario
1	Pobre identificación de Stakeholders.	EP1, EP6, EP2
2	Pobre entendimiento de requisitos.	EP1, EP2, EP13 EP6.
3	Falta de aseguramiento de la satisfacción de los Stakeholders con los requisitos.	EP1, EP6, EP2
4	Recolección de requisitos incompletos y ambiguos.	EP1, EP2, EP10, EP13, EP14
5	Requisitos incorrectos.	EP1, EP2
6	Pobre modelamiento de requerimientos funcionales.	EP1
7	Trazabilidad de requisitos deficiente.	EP1, EP2, EP14

**CONTINÚA**

<b>N°</b>	<b>Problemas</b>	<b>Estudios Primario</b>
8	Especificación pobremente definida.	EP1, EP13
9	Falta de consideración de problemas sociales y organizacionales.	EP1
10	Requisitos sin priorizar	EP1, EP2
11	Poca comunicación durante la elicitación de requisitos.	EP1, EP2
12	Preferencias del usuario poco claras.	EP1, EP10, EP2
13	Falta de documentación de requisitos funcionales y no-funcionales.	EP1, EP10, EP2, EP14
14	Poca comprensión de las restricciones de dominio.	EP1, EP2, EP13
15	Falta de documentación de la relación entre requisitos y Stakeholders.	EP1, EP2
16	Ambigüedad en la definición del alcance del sistema.	EP1, EP10, EP2, EP12, EP13, EP14
17	Requisitos funcionales, no-funcionales y restricciones de dominio no definido.	EP1, EP2
18	Mal manejo de los riesgos de seguridad en los requisitos.	EP1
19	Comunicación deficiente.	EP2, EP4, EP7, EP6, EP10, EP14
20	Falta de colaboración y coordinación.	EP4, EP6, EP10
21	Falta de gestión del intercambio de conocimientos	EP4, EP6, EP13
22	Falta de gestión de Requisitos.	EP4, EP2
23	Problemas de gestión de equipo.	EP4, EP13
24	Cambios organizacionales	EP4, EP2
25	Falta de Confianza	EP4, EP6
26	Falta de reuniones presenciales.	EP4, EP6
27	Falta de herramientas colaborativas y tecnología.	EP4, EP6, EP10, EP2
28	Falta de negociaciones y discusiones.	EP4
29	Falta de habilidad y capacitación.	EP4, EP6, EP10
30	Falta de un proceso de Ingeniería de Requisitos.	EP4, EP2
31	Alto costo de mano de obra.	EP4, EP12, EP14
32	Infraestructura técnica inadecuada.	EP4
33	Falta de disponibilidad del cliente.	EP7, EP10
34	Arquitectura de Software inapropiada.	EP7, EP10
35	Dificultad de estimación de tiempo y presupuesto del proyecto	EP7, EP6, EP10, EP2, EP12, EP14
36	Falta de consideración de requisitos no-funcionales	EP7, EP2

**CONTINÚA**

<b>N°</b>	<b>Problemas</b>	<b>Estudios Primario</b>
37	Cambio de requisitos.	EP7, EP2
38	Zona horaria diferente	EP6
39	Falta de canales de comunicación informales y síncronos.	EP6, EP2
40	Falta de cohesión del equipo.	EP6, EP13
41	Falta de conciencia global	EP6
42	Falta de rapidez al comunicar cambios en los requisitos.	EP6, EP2
43	Problemas colaborativos socio-culturales.	EP6, EP2
44	Problemas de comunicación socio-culturales.	EP6, EP2
45	Falta de confidencialidad.	EP6
46	Falta de gestión de conflictos.	EP6
47	Distribución de trabajo inadecuada	EP6
48	Falta de experiencia.	EP6
49	Falta de liderazgo.	EP6
50	Procesos mal definidos.	EP6, EP2, EP13
51	Conflictos entre el cliente y el proveedor.	EP6, EP12
52	Diferencias entre los modelos de proceso utilizados.	EP6, EP2
53	Diferentes terminologías y notaciones.	EP6, EP13
54	Dificultad para extraer información relevante de los Stakeholders.	EP2, EP6, EP13
55	Alto número de usuarios como de requisitos.	EP6, EP2
56	Herramientas no utilizadas debido a la falta de seguridad y desempeño.	EP6, EP2
57	Falta de motivación por parte del equipo de trabajo.	EP10
58	Malas prácticas de Ingeniería de Requisitos.	EP10, EP2
59	Falta de entendimiento entre Stakeholders.	EP2
60	Falta de mecanismos para el intercambio de conocimiento.	EP2
61	Problemas legales.	EP2
62	Falta de involucramiento de usuarios.	EP2, EP14
63	Formato de historias de usuario deficiente	EP12
64	Dificultades en la priorización de requisitos	EP12

Con el propósito de identificar la relación entre las causas y los problemas, llevamos a cabo una categorización de los problemas identificados en los estudios primarios (Ver Tabla 11). La

categorización se basó en las propuestas que se encontraron en dichos estudios (Nicolás, Carrillo de Gea, Nicolás, Fernández Aleman, & Toval, 2018), (Kumari S & S Pillai, 2013 ), (Anu, Hu, C Carverc, S Waliaa, & Bradshaw, 2018).

- Problemas relacionados con la comunicación PRC.
- Gestión del Conocimiento GDC.
- Problemas relacionados con el equipo del proyecto PRE.
- Herramientas y Técnicas HYT.
- Problemas relacionados con los Requisitos PRR.
- Problemas relacionados con el Proyecto PRP.

**Tabla 11**  
*Categorías de Problemas de Requisitos*

Categoría	Problema	Estudio Primario
<b>PRC</b>	- Poca comunicación durante la elicitación de requisitos.	EP1, EP2.
	- Comunicación deficiente.	EP2, EP4, EP7, EP6, EP10, EP14
	- Falta de canales de comunicación informales y síncronos.	EP2, EP6
	- Falta de rapidez al comunicar cambios en los requisitos.	EP2, EP6
	- Problemas de comunicación socio-culturales.	EP2, EP6
	- Falta de entendimiento entre Stakeholders.	EP2
	- Zona horaria diferente	EP6
	- Falta de colaboración y coordinación.	EP4, EP6, EP10
	- Falta de reuniones presenciales.	EP4, EP6
	- Falta de negociaciones y discusiones.	EP4
	- Diferentes terminologías y notaciones.	EP6, EP13
	- Preferencias del usuario poco claras.	EP1, EP10, EP2
	- Falta de disponibilidad del cliente.	EP7, EP10
	- Falta de confidencialidad.	EP6
<b>GDC</b>	- Falta de gestión del intercambio de conocimientos.	EP4, EP6, EP13
	- Falta de mecanismos para el intercambio de conocimiento.	EP2

**CONTINÚA**

Categoría	Problema	Estudio Primario
<b>PRE</b>	- Falta de Confianza.	EP4, EP6
	- Falta de habilidad y capacitación.	EP4, EP6, EP10
	- Falta de cohesión del equipo.	EP6, EP13
	- Problemas colaborativos socio-culturales.	EP2, EP6
	- Falta de experiencia.	EP6
	- Falta de liderazgo.	EP6
	- Conflictos entre el cliente y el proveedor.	EP6, EP12
	- Falta de motivación por parte del equipo de trabajo.	EP10
	- Pobre entendimiento de requisitos.	EP1, EP2, EP13 EP6.
	- Problemas de gestión de equipo.	EP4, EP13
	- Falta de involucramiento de usuarios.	EP2, EP14
	- Falta de aseguramiento de la satisfacción de los Stakeholders con los requisitos.	EP1, EP6, EP2, EP13
	- Pobre Identificación de Stakeholders.	EP1, EP6, EP2
	- Dificultad para extraer información relevante de los Stakeholders.	EP2, EP6, EP13
<b>HYT</b>	- Falta de herramientas colaborativas y tecnología.	EP2, EP4, EP6, EP10
	- Herramientas no utilizadas debido a la falta de seguridad y desempeño.	EP2, EP6
<b>PRR</b>	- Requisitos incorrectos.	EP1, EP2
	- Trazabilidad de requisitos deficiente	EP1, EP2, EP14
	- Requisitos sin priorizar.	EP1, EP2
	- Falta de gestión de Requisitos.	EP2, EP4
	- Falta de consideración de requisitos no-funcionales	EP2, EP7
	- Cambio de requisitos.	EP2, EP7
	- Recolección de requisitos incompletos y ambiguos.	EP1, EP2, EP10, EP13, EP14
	- Pobre modelamiento de requerimientos funcionales.	EP1
	- Requisitos funcionales, no-funcionales y restricciones de dominio no definido.	EP1, EP2
	- Malas prácticas de Ingeniería de Requisitos.	EP10, EP2

*CONTINÚA*

	- Dificultades en la priorización de requisitos	EP12
<b>PRR</b>	- Falta de documentación de requisitos funcionales y no-funcionales.	EP1, EP2, EP10, EP14
	- Falta de documentación de la relación entre requisitos y Stakeholders	EP1, EP2
	- Especificación pobremente definida.	EP1, EP13
	- Formato de historias de usuario deficiente	EP12
	- Poca comprensión de las restricciones de dominio.	EP1, EP2, EP13
	- Ambigüedad en la definición del alcance del sistema.	EP1, EP10, EP2, EP12, EP13, EP14
	- Mal manejo de los riesgos de seguridad en los requisitos.	EP1
	- Falta de conciencia global	EP6
	- Alto costo de mano de obra.	EP4, EP12, EP14
	- Dificultad de estimación de tiempo y presupuesto del proyecto	EP7, EP6, EP10, EP2, EP12, EP14
	- Falta de gestión de conflictos.	EP6
	- Alto número de usuarios como de requisitos.	EP6, EP2
<b>PRP</b>	- Falta de consideración de problemas sociales y organizacionales.	EP1
	- Cambios organizacionales	EP2, EP4
	- Falta de un proceso de Ingeniería de Requisitos.	EP4, EP2
	- Infraestructura técnica inadecuada.	EP4
	- Arquitectura de Software inapropiada.	EP7, EP10
	- Diferencias entre los modelos de proceso utilizados.	EP6, EP2
	- Problemas legales.	EP2
	- Distribución de trabajo inadecuada.	EP6
	- Procesos mal definidos.	EP6, EP2, EP13

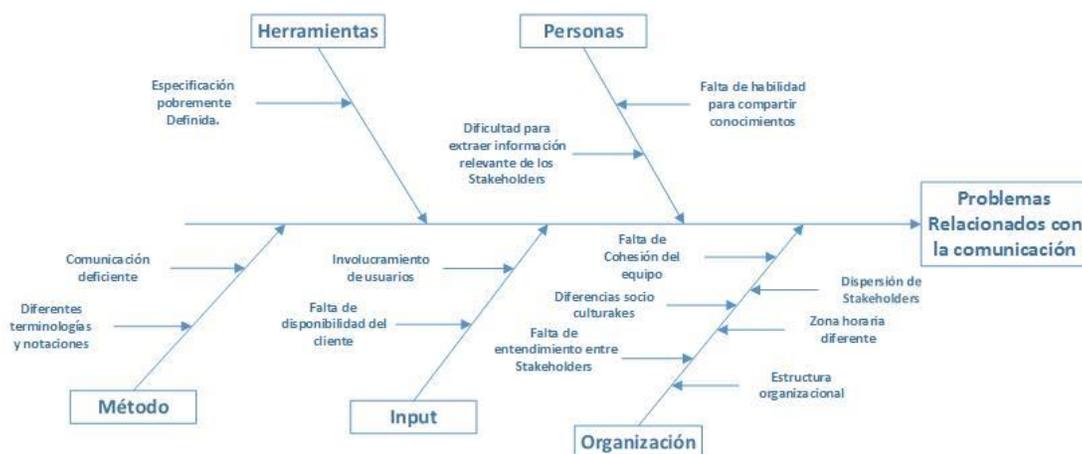
Por otro lado, se pudo identificar un total de 36 causas específicas de la Ingeniería de Requisitos en las revisiones de literatura resultantes (estudios primarios), como se presenta en la Tabla 12.

**Tabla 12.**  
*Causas Identificadas*

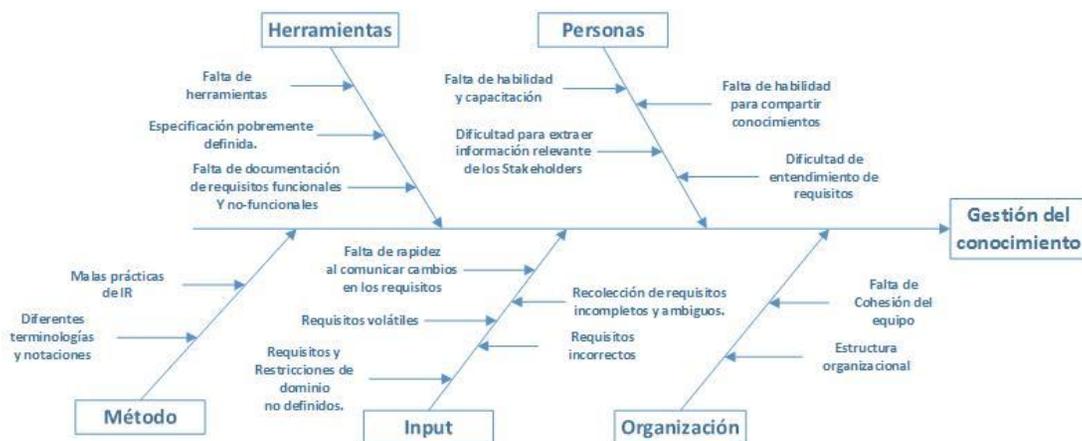
N°	Causas	Estudio Primario
1	Dispersión de los Stakeholders.	EP6, EP2
2	Zona horaria diferente	EP6
3	Falta de Cohesión en el equipo	EP6
4	Falta de Habilidad para compartir conocimiento	EP6, EP2
5	Diferencias Socio-Culturales	EP6, EP2, EP9
6	Estructura Organizacional	EP6, EP2
7	Falta de definición de errores	EP6
8	Falta de Coordinación	EP6
9	Conflictos entre el cliente y el proveedor.	EP6, EP14
10	Falta de Herramientas	EP6, EP2
11	Requisitos Volátiles	EP7,EP10,EP2, EP14
12	Mala Planificación y diseño	EP7, EP2
13	Involucramiento de usuarios	EP8
14	Falta de consideración de problemas sociales y organizacionales.	EP9, EP2
15	Falta de habilidad y capacitación.	EP9, EP13
16	Malas prácticas de Ingeniería de Requisitos.	EP10, EP12
17	Toma de Decisiones incorrectas en las fases iniciales del proyecto.	EP10
18	Dificultad de entendimiento de requisitos	EP2, EP14, EP13
19	Falta de entendimiento entre Stakeholders.	EP2, EP13
20	Falta de rapidez al comunicar cambios en los requisitos.	EP2
21	Cambios organizacionales	EP2
22	Ambigüedad en la definición del alcance del sistema.	EP2, EP13
23	Falta de documentación de requisitos funcionales y no-funcionales.	EP2, EP14
24	Dificultad para extraer información relevante de los Stakeholders.	EP2, EP13
25	Comunicación deficiente	EP2, EP13
26	Falta de disponibilidad del cliente.	EP14
27	Arquitectura de Software inapropiada.	EP14
28	Dificultad de estimación de tiempo y presupuesto del proyecto	EP14, EP12
29	Limitaciones contractuales	EP14
30	Especificación pobremente definida.	EP13
31	Diferentes terminologías y notaciones	EP13
32	Requisitos y restricciones de dominio no definido.	EP13
33	Diferencias entre los modelos de proceso utilizados.	EP13
34	Pobre Identificación de Stakeholders.	EP13
35	Recolección de requisitos incompletos y ambiguos.	EP15
36	Requisitos Incorrectos	EP15

Se puede observar cierta discrepancia entre los diferentes autores ya que en las tablas anteriores, algunos problemas son considerados también como causas.

Los diagramas de causa efecto fueron introducidos para proporcionar apoyo visual en sesiones de análisis causal con conocimiento de las causas comunes de problemas recopilados de experiencias anteriores (Ishikawa, 1991). Además, han demostrado ser un instrumento útil en una prueba de concepto, un estudio experimental y una experiencia industrial (Kalinowski, y otros, 2015). Dada su funcionalidad, hemos identificado la relación entre las causas y las categorías de los problemas antes mencionadas como se muestra en los siguientes diagramas.



**Figura 16.** PRC – Causas



**Figura 17.** GDC – Causas

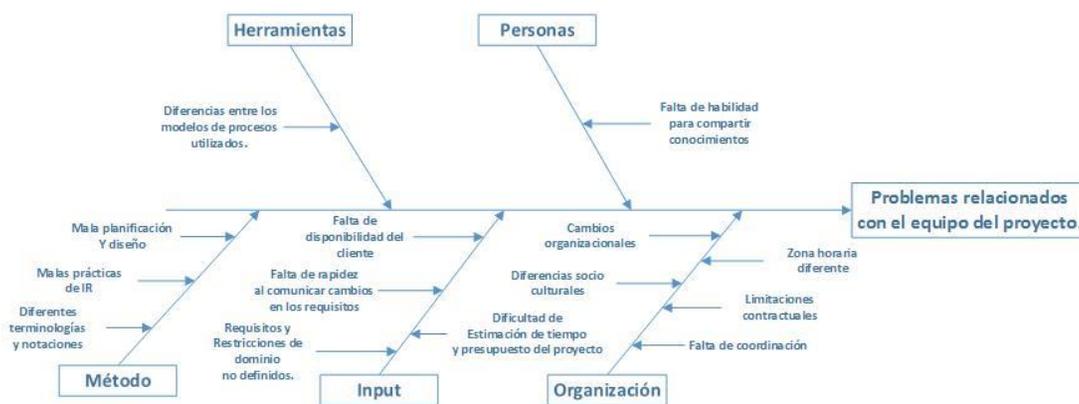


Figura 18. PRE – Causas

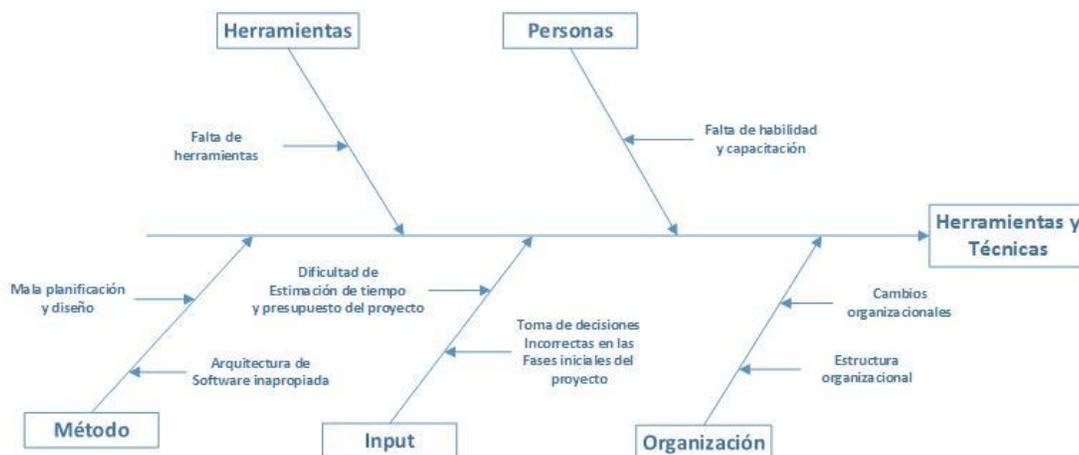
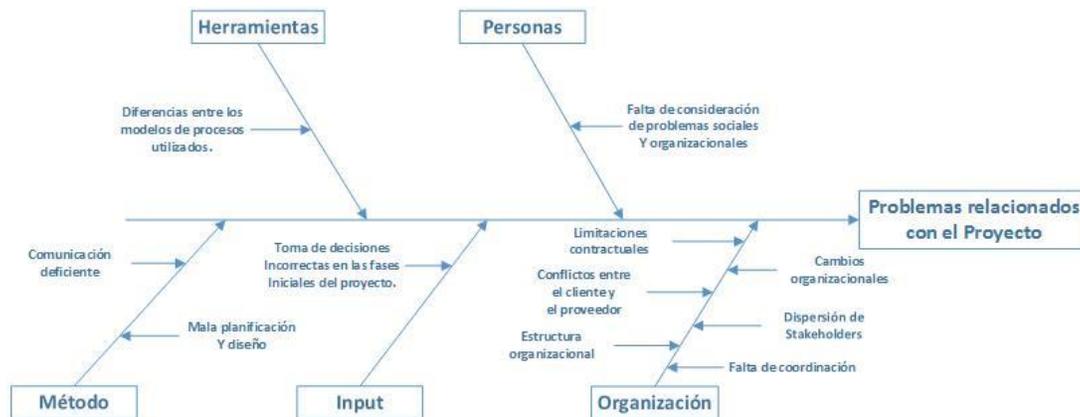


Figura 19. HYT – Causas



Figura 20. PRR – Causas



**Figura 21.** PRP – Causas

**RQ3: ¿Cuáles son los métodos de investigación empleados en los trabajos que se han llevado a cabo en torno a la Ingeniería de Requisitos?**

(Besrou, AB Rahim, & Dominic, 2016), investigaron aún más los desafíos críticos identificados en su revisión de literatura a través de un estudio cuantitativo aplicado a pequeñas y medianas empresas de desarrollo de Software; para lo cual se recolectaron 217 encuestas y posterior a su análisis se identificaron 12 desafíos críticos en la IR.

(Dermeval, y otros, 2014), identificaron dentro de sus resultados los siguientes métodos de investigación, resumidos en el la Figura 22.



**Figura 22.** Métodos de Investigación Identificados en el EP3

Los autores clasificaron los métodos de investigación basados en las categorías: Caso de estudio, experimento controlado, cuasi experimento, escenarios ilustrativos y no aplicable; donde un escenario ilustrativo se refiere a artículos que solo evalúan sus contribuciones con pequeños ejemplos, mientras que no aplicable se refiere a artículos que no presentan ningún tipo de métodos de investigación.

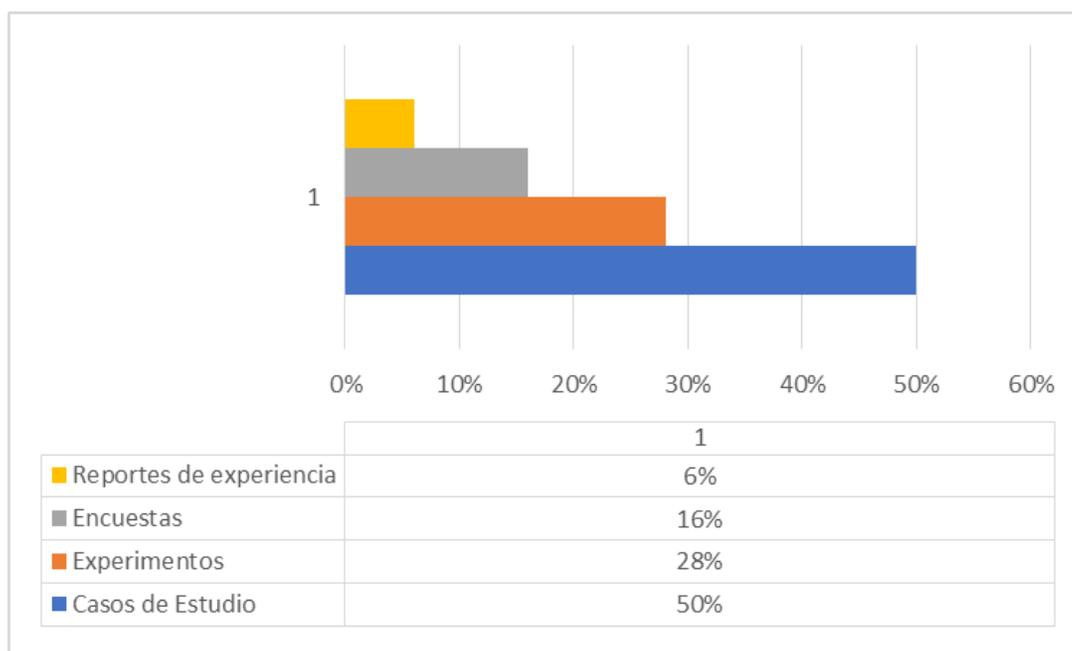
Los escenarios ilustrativos (54.55%; 36 artículos) constituyen la mayoría de los artículos, seguidos de experimentos controlados (21.21%; 14 artículos), estudios de caso (16.67%; 11 artículos), no aplicable (6.06%; 4 artículos) y cuasi-experimento (1,52%; 1 artículo).

(Besrou, AB Rahim, & Dominic, 2016), llevaron a cabo un estudio cuantitativo a través de una encuesta aplicada a Pymes de la industria de Software y obtuvieron como resultado que 12 técnicas de las 15 identificadas en su revisión de literatura, son las que mayores beneficios presentan.

(Bano & Zowghi, 2013), en su estudio identificaron los siguientes métodos de investigación: encuesta, casos de estudio, experimentos y estudios de campo. Los porcentajes de los métodos de investigación utilizados en los estudios identificados por los autores son: encuesta 53%, casos de estudio 23%, experimentos 13% y estudios de campo 8%, Grounded Theory 1%, reporte de experiencia 1%, action research 1%.

(Daneva, Damian, Marchetto, & Pastor, 2014), realizaron una revisión exhaustiva de 17 artículos de los cuales 5 estudios de alta calidad fueron seleccionados para su análisis, en los cuales fueron empleados los siguientes métodos de investigación: estudio de campo, encuesta, experimentos, Grounded Theory, action research, reporte de experiencia y casos de estudio. Siendo los experimentos y los casos de estudio los métodos empleados más de una vez (3 veces cada uno) dentro de las investigaciones.

En el estudio realizado por (Ambreen, Ikram, Usman, & Niazi, 2018), se puede evidenciar que la mayoría de estudios utilizaron casos de estudio como método de investigación (50%), seguido de experimentos (28%), encuestas (16%), mientras que los reportes de experiencia son pocos (6%) (Ver Figura 23).



**Figura 23.** Métodos de Investigación Identificados en el EP11

(Heikkila, Lassenius, Damian , & Paasivara, 2015), realizaron un estudio de mapeo sistemático sobre la Ingeniería de Requisitos con el objetivo de explorar la literatura científica que se ha desarrollado en torno a la temática, en su estudio analizaron 28 artículos los mismo que utilizaron los siguientes métodos de investigación (Ver Tabla 13).

**Tabla 13.**

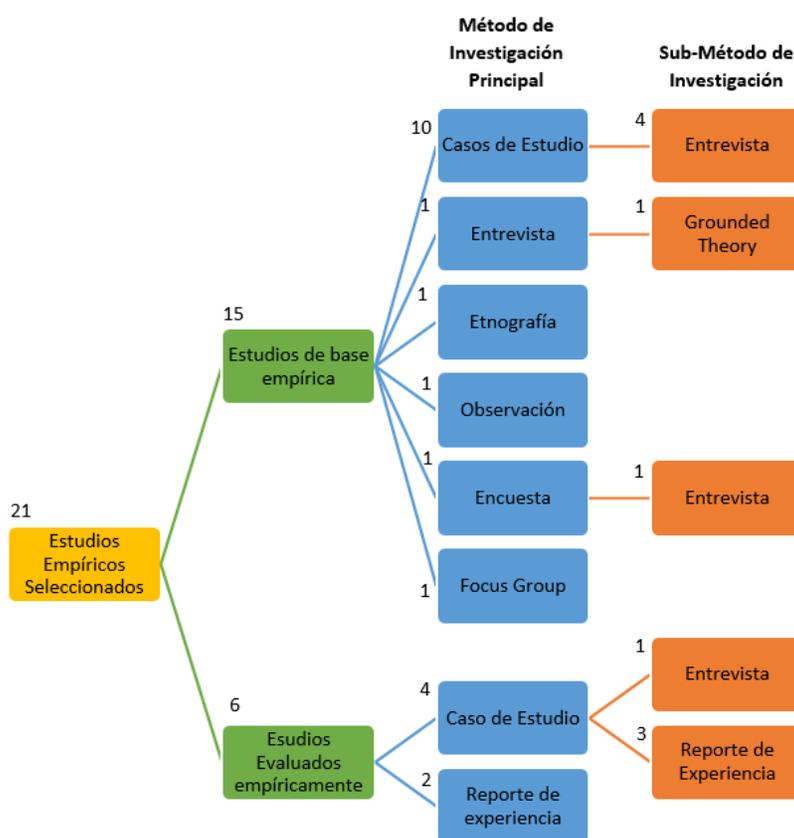
*Resumen de Métodos de Investigación identificados en el EP12*

Método de Investigación	N° de artículos.
Caso de Estudio Múltiple	6
Caso de Estudio Único	5
Reporte de Experiencia	3

El método más utilizado en la literatura identificado por los autores es el caso de estudio múltiple ( $N = 6 \approx 21\%$ ), seguido por caso de estudio único ( $N = 5 \approx 18\%$ ). También se lograron

identificar tres artículos cuyo método de investigación corresponde a reportes de experiencia ( $\approx$  11).

(Inayat, Salwah Salim, Marczak, Daneva, & Shamshirband, 2015), llevaron a cabo una revisión sistemática de literatura y seleccionaron 21 artículos los cuales analizan la Ingeniería de requisitos ágil. La Figura 24 presenta los métodos de investigación que fueron adoptados por los 21 estudios.



**Figura 24.** Métodos de Investigación identificados en el EP14

Los autores encontraron que la mayoría de estudios son de naturaleza exploratoria. De los 21 estudios 6 son “evaluados empíricamente”, basados en la evaluación de métodos o herramientas sin ninguna investigación experimental; mientras que 12 estudios utilizaron métodos de

investigación cualitativa y son considerados de base empírica, es decir son de naturaleza exploratoria.

De los 15 estudios empíricos, las entrevistas y los experimentos se utilizaron como sub-métodos en un 40% (6 de 15). El 60% de estudios del grupo no mencionan haber utilizado ningún sub-método.

Por otro lado, los autores encontraron que los estudios “evaluados empíricamente” corresponden a un 18% y están basados en nuevas ideas tales como herramientas y métodos.

(Pacheco , García , & Reyes, 2018), analizaron 140 estudios en los cuales identificaron, que el 92% son empíricos (en su mayoría casos de estudio), el 5% son teóricos y un pequeño número de estudios (3%) se encuentran en la categoría de otros. Además, el 55.8% de los estudios empíricos provino de experiencias académicas, el 38.8% estuvo relacionado con experiencias industriales y el 5.4% proporcionó experiencias académicas e industriales al mismo tiempo.

#### **RQ4: ¿Cuál es la tendencia en torno a las propuestas realizadas con miras a tratar los problemas vinculados a la Ingeniería de Requisitos?**

Con base en los estudios resultantes de la Revisión Sistemática de Literatura, la información referente a los tipos de propuestas realizadas se extrajo de 10 estudios primarios.

En la revisión de literatura realizada por (Kumari S & S Pillai, 2013 ) se identifican algunas prácticas recomendables para la Ingeniería de Requisitos todas estas con miras a tratar los problemas identificados en dicho estudio, cada práctica se encuentra clasificada de acuerdo a distintos factores (Ver Figura 25).



**Figura 25.** Prácticas de Ingeniería de Requisitos identificadas en el EP2

Todas estas propuestas se las puede clasificar como lineamientos para mejorar el proceso de Ingeniería de Requisitos.

(Dermeval, y otros, 2014), realizan una revisión sistemática con el objetivo de comprender mejor cómo las ontologías pueden ayudar en el proceso de Ingeniería de requisitos, en su estudio encuentra que las ontologías se han utilizado en la ingeniería de requisitos para minimizar y resolver diferentes tipos de problemas. Por ejemplo, para mejorar la integridad de la especificación de requisitos y para ayudar a eliminar la ambigüedad de su significado, para realizar un análisis de consistencia en los requisitos, para ayudar a modelar explícitamente el conocimiento del dominio, para administrar el conocimiento de los requisitos y los cambios de requisitos, etc.

Por otro lado, en este estudio los autores identifican los tipos de contribuciones existentes para la ingeniería de requisitos impulsada por ontologías entre los cuales se encuentran métodos, modelos herramientas y procesos (Ver Tabla 14).

**Tabla 14.**  
*Tipos de Contribución de IR relacionadas con Ontologías encontrados en el EP3*

Tipo de contribución	Nº de artículos	%
<b>Método</b>	58	87.9 %
<b>Modelo</b>	50	75.8 %
<b>Herramienta</b>	38	57.6 %
<b>Proceso</b>	2	3 %

La contribución predominante que identificaron fue el Método (87.9%; 58 artículos), seguido del Modelo (75.8%; 50 artículos), Herramienta (57.6%; 38 artículos) y Proceso (3%; 2 artículos).

(Besrou, AB Rahim, & Dominic, 2016), realizan una revisión de literatura con el objetivo de investigar las técnicas de requisitos más aceptadas que tienen un impacto positivo en la Ingeniería de Requisitos (Ver Tabla 15).

**Tabla 15.**  
*Técnicas de requisitos más aceptadas encontradas en el EP5*

NO	TÉCNICA DE REQUISITOS	NIVEL
1	Entrevista	4.39
2	Observación	1.89
3	Lluvia de ideas	4.02
4	Lenguaje Natural Estructurado	4.08
5	JAD (Joint Application Design)	4.35
6	Revisión por pares	4.17
7	Etnografía	2.18
8	Diagrama de actividades	4.09

**CONTINÚA**

NO	TÉCNICA DE REQUISITOS	NIVEL
9	Especificación de requisitos de software	4.05
10	Checklist	4.11
11	Casos de Uso	4.17
12	Laddering	2.04
13	Caso de mal uso	4.06

La Tabla 15 muestra las puntuaciones alcanzadas para comprender mejor el nivel de aceptación de las técnicas de requisitos, los autores utilizaron una encuesta para medir la importancia de las técnicas de requisitos, de acuerdo a la escala de Likert; donde 1 = totalmente en desacuerdo, 2 = en desacuerdo, 3 = neutral, 4 = de acuerdo y 5 = muy de acuerdo. Una técnica con una calificación de 4.0 o superior se considera una técnica aceptada.

En el estudio se encontró que, de las 13 técnicas, 10 de estas se pueden considerar como aceptadas, siendo la entrevista la técnica con mayor puntuación de todas (4.39). Lo que demuestra que esta técnica es la más preferida por los profesionales de software.

El objetivo principal de identificar estas técnicas es proporcionar a las compañías de software particularmente a las compañías con menos experiencia, un grupo de técnicas efectivas que buscan reducir los desafíos de los requisitos.

(Nicolás, Carrillo de Gea, Nicolás, Fernández Aleman, & Toval, 2018), en su estudio establecen una metodología que se basa en los procedimientos de análisis cuantitativo y el método de Grounded Theory (GTM), cuyo objetivo es categorizar el conocimiento en la literatura sobre riesgos y salvaguardas para Ingeniería de Requisitos Global.

Los resultados de esta investigación se compilan en un repositorio público, si bien esta investigación aborda los riesgos y las salvaguardas recopiladas de la literatura científica, es de

naturaleza muy práctica lo que la hace útil para cualquier empresa involucrada en el desarrollo de software tanto global como tradicional.

(Bano & Zowghi, 2013), realizan una revisión sistemática de literatura sobre la participación de los usuarios durante las actividades de Ingeniería de requisitos y exploran su relación con el éxito del sistema.

Los autores ofrecen un checklist creado a partir de factores identificados en la literatura que podrían utilizarse para la participación efectiva de los usuarios en las actividades de Ingeniería de Requisitos. Para la correcta participación de los usuarios en la Ingeniería de Requisitos se debe dar respuesta al siguiente checklist (Tabla 16).

**Tabla 16.**

*Checklist aplicable para la correcta participación de usuarios encontrada en el EP8*

PREGUNTAS	SUB-PREGUNTAS
<b>1. ¿Cuál es la perspectiva para la participación de los usuarios?</b>	1.1 ¿Por qué involucrar a los usuarios? 1.2 ¿Cómo involucrar a los usuarios?
<b>2. ¿A qué nivel o grado deben ser involucrados los usuarios?</b>	NA
<b>3. ¿Quién debe estar involucrado y quién debe participar?</b>	3.1 ¿Quién será afectado? 3.2 ¿Es posible identificar solo 2 o 3 categorías de usuarios principales? 3.3 ¿Cuáles son las características de las personas en cada categoría de usuario? 3.4 ¿Cuáles son las características de las tareas desempeñadas por cada categoría de usuario? 3.5 ¿Qué es lo que les gusta y disgusta sobre su trabajo a los diferentes usuarios? 3.6 ¿Cómo es probable que los diferentes usuarios reaccionen al sistema?

**CONTINÚA**

PREGUNTAS	SUB-PREGUNTAS
4. ¿Qué método y técnica debe ser utilizado?	NA

(Curcio , Navarro , Malucelli, & Reinehr , 2018), realizaron un estudio de mapeo sistemático, en el cual analizaron 104 artículos de los cuales se pudo Identificar los tipos de contribuciones existentes para la ingeniería de requisitos (Ver Tabla 17).

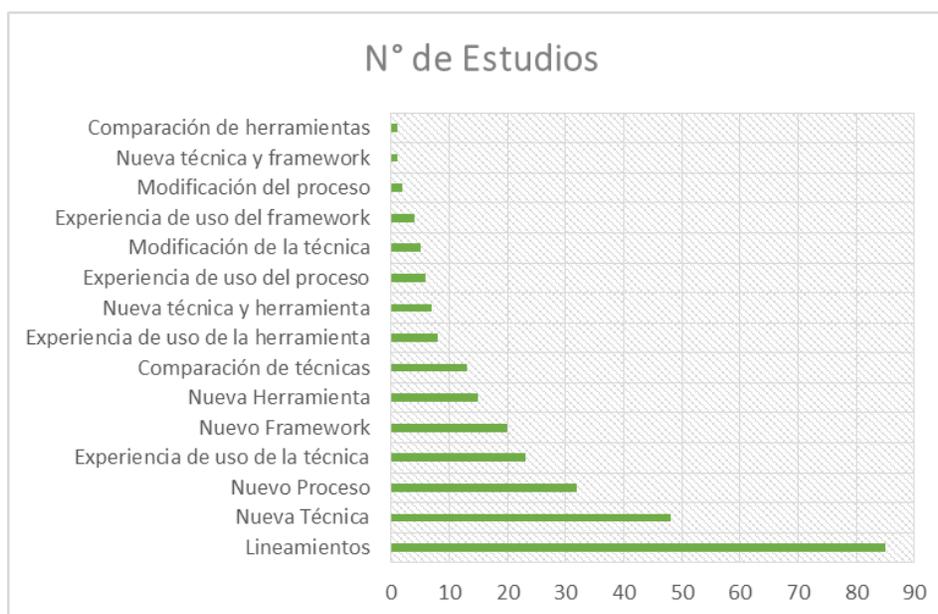
**Tabla 17.**

*Propuestas de Ingeniería de Requisitos identificadas en el EP10*

Propuesta	N° de artículos
Frameworks	6
Método	2
Lineamientos	7
Técnicas	2
Modelo	3
Herramientas	5
Proceso de Negocio basado en Ontologías	1

Las propuestas más predominantes que identificaron los autores corresponden a los lineamientos, seguidos de frameworks, herramientas y modelos. Por otro lado se encontró que las contribuciones menos predominantes son los métodos, técnicas y un proceso de negocio basado en ontologías. Es importante aclarar que no en todos los artículos se encontró una propuesta de IR.

(Ambreen, Ikram, Usman, & Niazi, 2018), en su estudio encontraron diversos tipos de propuestas en la investigación en Ingeniería de Requisitos (Ver Figura 26).



**Figura 26.** Propuestas de Ingeniería de Requisitos identificadas en el EP11

La Figura 26 muestra varios tipos de propuestas que se han investigado en torno a la Ingeniería de Requisitos. Estos resultados indican que las guías, técnicas y procesos son los más investigados en los estudios empíricos de IR. El 31 % de los estudios propusieron guías para aplicarlas en las diversas áreas de IR. Se puede ver que hay más interés en las nuevas propuestas tales como guías, técnicas, proceso, etc., mientras que se presta poca atención al uso de propuestas existentes.

(Heikkila, Lassenius, Damian , & Paasivara, 2015), no identifican como tal las propuestas, sin embargo los autores clasifican los estudios en una categoría denominada “tipo de artículo”, en la cual se incluyen estudios según el método de investigación utilizado, así como el tipo de propuesta; en donde se pudo identificar métodos que corresponden a 8 artículos del total de 26 que fueron considerados para este estudio de mapeo.

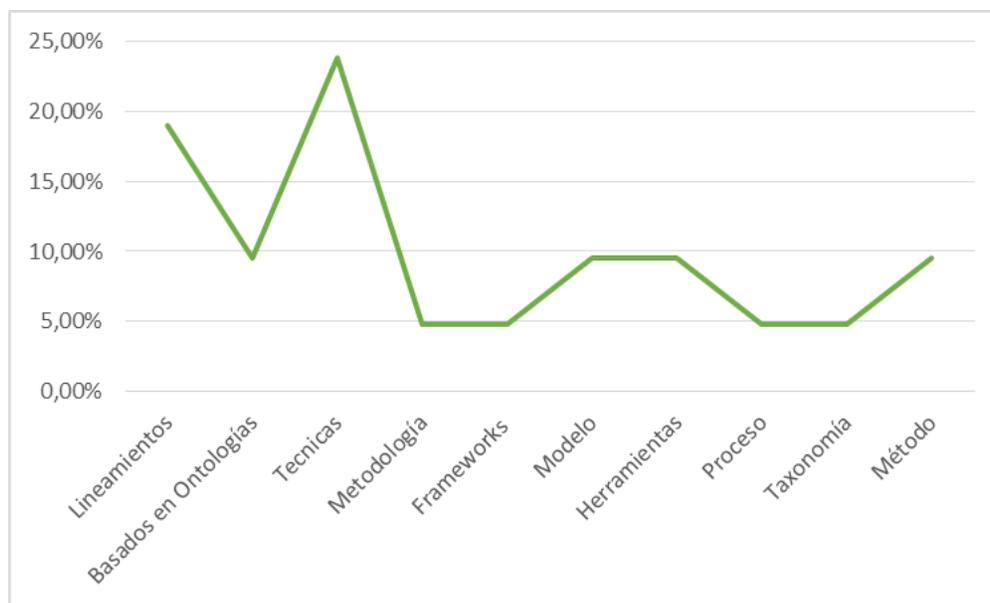
(Anu, Hu, C Carverc, S Waliaa, & Bradshaw, 2018), realizan una investigación en torno a los errores humanos que tienen lugar en la Ingeniería de Requisitos, lo cual dio como resultado el desarrollo de una taxonomía teórica de errores humanos en la IR. Esta propuesta se basa en la taxonomía estándar del error humano de la psicología cognitiva. Las principales contribuciones de esta investigación son:

- Ilustración de la capacidad y el valor de aplicar la investigación de errores humanos a un problema de ingeniería de software a través de una estrecha interacción entre expertos en ingeniería de software y expertos en psicología cognitiva.
- Desarrollo de una taxonomía de error humano para organizar los errores humanos comunes cometidos durante la ingeniería de requisitos en un marco sistemático; y
- Un análisis de la literatura que describe los errores humanos para identificar la falta de cohesión en la investigación de ingeniería de requisitos en relación con el uso de la información de errores humanos.

(Inayat, Salwah Salim, Marczak, Daneva, & Shamshirband, 2015), en su estudio encontraron propuestas basadas en nuevas ideas tales como herramientas y métodos. Estas incluyen herramientas para modelar requisitos no funcionales, una técnica de mapeo mental para facilitar la IR, un modelo para la priorización de requisitos, una técnica de bosquejamiento de metas para IR y modelado de requisitos a través de un razonamiento para -consistente. Generalizando podemos encontrar un total de 2 herramientas, 2 técnicas y 2 modelos.

Con base en los estudios primarios identificados, se han logrado identificar 10 tipos de contribuciones (Ver Figura 27). En primer lugar se encuentran las técnicas presentes en el 24% de los estudios primarios, en segundo lugar están los lineamientos que representan el 19%; mientras

tanto las propuestas basadas en ontologías, modelos, métodos y herramientas se encuentran presentes en casi un 10% cada una. Finalmente las propuestas menos encontradas en la literatura corresponden a las metodologías, frameworks, procesos y taxonomía con 5% cada una.



**Figura 27.** Tendencia de Propuestas identificada en la literatura.

### 3.8. SÍNTESIS DE RESULTADOS DE LA SMS

Los problemas más reportados en la literatura son comunicación deficiente, dificultad de estimación de tiempo y presupuesto, ambigüedad en la definición del alcance del sistema y recolección de requisitos incompletos y ambiguos. Sin embargo, la frecuencia con que son reportados no implica que la magnitud del impacto sobre la calidad del producto software resultante sea mayor que aquellos menos citados. Además de esto, encontramos que dentro de la categorización realizada anteriormente, los problemas relacionados con los requisitos son aquellos que se ven influenciados por un número mayor de causas.

Con respecto a las causas identificadas en la literatura encontramos que requisitos volátiles,

diferencias socio culturales y dificultad de entendimiento de requisitos son las más citadas, sin embargo se debe considerar que una causas puede originar más de un problema específico, razón por la cual hemos establecido la relación entre las causas y las categorías de los problemas definidas anteriormente, lo cual nos permitió identificar que las causas más comunes entre problemas son: diferentes terminologías y notaciones, dispersión de Stakeholders, falta de habilidad para compartir conocimiento, estructura organizacional, falta de coordinación, falta de herramientas, mala planificación y diseño, falta de habilidad y capacitación, falta de rapidez al comunicar cambios en los requisitos, cambios organizacionales, dificultad para extraer información relevante de los Stakeholders, comunicación deficiente, falta de disponibilidad del cliente, especificación pobremente definida y requisitos y restricciones de dominio no definido.

Para contrarrestar estas causas y evitar los problemas que afectan a la Ingeniería de Requisitos, se han propuesto varias herramientas, lineamientos, modelos y técnicas con miras a tratar los problemas vinculados a la Ingeniería de Requisitos. Destacándose de estas últimas la entrevista, JAD (Joint Application Development), Casos de uso, revisión por pares, checklist, lluvia de ideas, lenguaje natural estructurado, Diagrama de actividades y especificación de requisitos de Software.

Pese a los muchos esfuerzos que se han llevado a cabo para lidiar con los problemas relacionados a la Ingeniería de Requisitos utilizando técnicas que de acuerdo con la literatura son las más adecuadas, aún no se ha llegado a una solución definitiva, lo cual nos ha motivado a llevar a cabo un proceso de snowballing (Jalali & Wohlin, 2012) con el fin de identificar artículos relevantes adicionales a través de la lista de referencias de los estudios primarios así como información adicional respecto al área de interés. (Inayat, Salwah Salim, Marczak, Daneva, & Shamshirband, 2015), y (Daneva, y otros, 2013) aseguran que hay una urgente necesidad de que

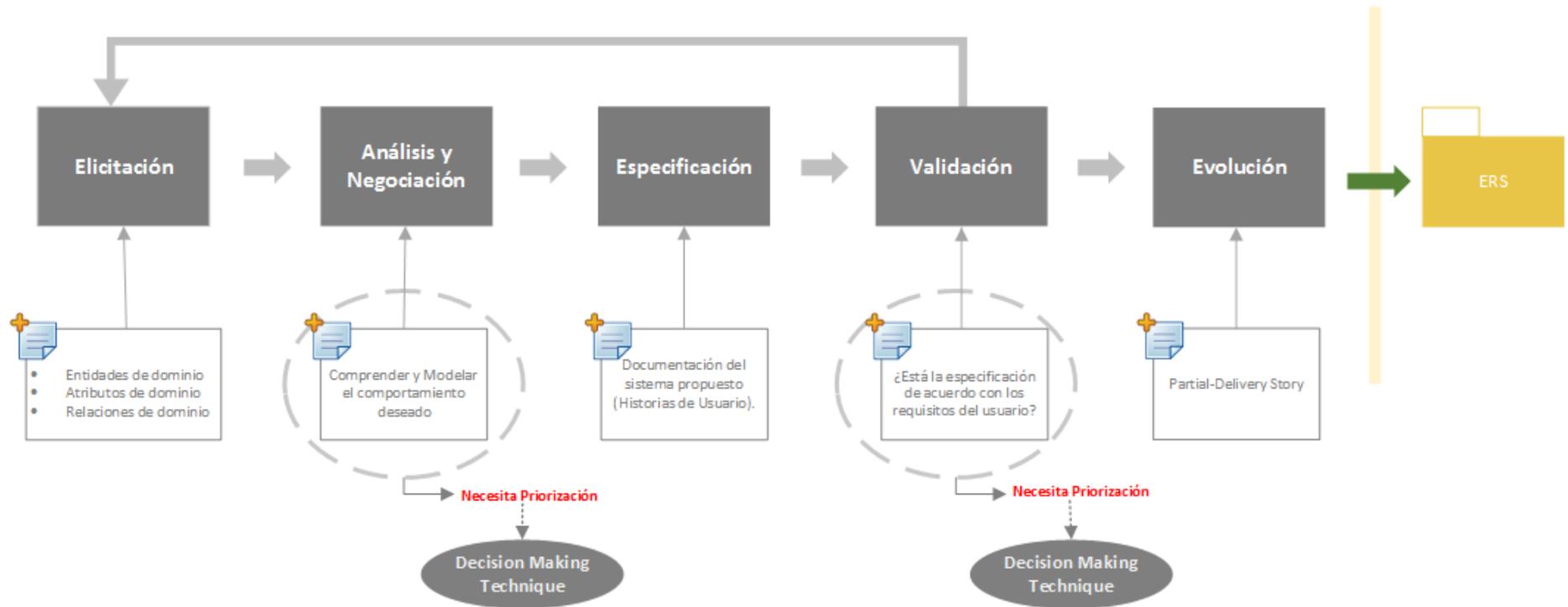
los requisitos sean apropiadamente definidos, clarificados y priorizados. Este proceso permitió identificar que la priorización es una actividad crucial a la hora de seleccionar los requisitos correctos de un set de requisitos candidatos, de modo que todos los intereses clave, restricciones técnicas y preferencias de los Stakeholders sean cumplidas (Gambo , Soriyan, & Ikono, 2015).

## **CAPITULO IV**

### **MODELO PROPUESTO**

#### **4.1. PROPUESTA DEL MODELO**

El modelo de procesos de Ingeniería de Requisitos que se propone en este trabajo de investigación corresponde a una adaptación del modelo de Swebok (Bourque & Fairley, 2014) con un enfoque ágil, de la mano de una serie de modificaciones provenientes de la síntesis de resultados del estudio terciario realizado (ver Figura 1). El proceso de selección de las características añadidas al modelo de Swebok consistió en abordar las causas más comunes relacionadas con los diversos problemas de IR encontrados en la literatura, las cuales son: Falta de coordinación del equipo de trabajo, falta de herramientas para apoyar el proceso de IR, mala planificación y diseño del proyecto, dificultad para extraer información relevante de los stakeholders, comunicación deficiente entre los stakeholders y el equipo de trabajo, utilización de diferentes terminologías y notaciones para referirse al mismo concepto y finalmente requisitos y restricciones de dominio no definidos. Como resultado se obtuvo un modelo que consta de 5 fases (ver Figura 28).



*Figura 28.* Modelo de Procesos de IR propuesto

Es preciso mencionar que, con respecto al modelo Swebok, el modelo propuesto añade la fase de evolución de requisitos y ciertas actividades adicionales a cada una de las fases anteriores. Cada una de las fases del modelo propuesto así como las actividades inherentes a cada una de ellas se detalla a continuación.

#### **4.1.1. Elicitación de Requisitos**

La elicitación de requisitos es la fase inicial del proceso de Ingeniería de Requisitos cuyo objetivo es recolectar todos los requisitos relevantes a través de interacciones directas o indirectas entre analistas de requisitos y los Stakeholders (Spoletini & Ferrari, 2017).

En esta fase se obtiene toda la información sobre el alcance del sistema, se planifican y ejecutan las reuniones con los usuarios para la obtención de los requisitos, se identifica los objetivos del sistema a desarrollar, se identifican los requisitos funcionales del sistema y finalmente se identifica los requisitos no funcionales del sistema (Bourque & Fairley, 2014).

A modo de aporte a esta fase, el estudio terciario permitió identificar diversas técnicas para realizar la elicitación, las cuales han sido ampliamente probadas y aceptadas por la industria del Software, tales como: JAD (Joint Application Development), brainstorming, entrevistas, casos de uso, análisis etnográfico, checklist, entre otras.

#### **4.1.2. Análisis y Negociación**

De acuerdo a Swebok (Bourque & Fairley, 2014), en esta fase se detectan y resuelven los conflictos existentes entre los requisitos elicitados, se descubren los límites del software y cómo éste debe interactuar en el entorno organizacional y operacional.

Las actividades inherentes a esta fase según Swebok (2014) son:

1. Clasificación de requisitos para determinar problemas asociados (Incompletos, incorrectos, incompatibles, innecesarios, entre otros).
2. Modelado conceptual para entender la naturaleza del problema así como para bosquejar una posible solución.
3. Elaboración de lista de requisitos candidatos.
4. Negociación y priorización.
5. Elaboración de lista de requisitos priorizados.

Las actividades 4 y 5 fueron añadidas como parte de la propuesta ya que durante la fase de análisis es necesario llevar a cabo una priorización de requisitos. Esta afirmación se ve respaldada por Gambo & otros (2015), quienes mencionan que la cantidad de conocimiento así como las perspectivas humanas durante la fase de análisis, crean un problema de toma de decisiones sobre los requisitos que pueden facilitarse mediante la priorización de los mismos.

Daneva & otros (2013), indican que la priorización de requisitos es un proceso de toma de decisiones, en el que se consideran diversas técnicas que facilitan la ejecución del mismo, tales como lluvia de ideas, técnica de grupo nominal, técnica Delphi, abogado del diablo e indagación dialéctica (Lunenburg, 2011).

Con esto se espera que una vez realizada la fase de análisis y negociación se obtenga un listado de requisitos priorizados, proveniente de los usuarios con amplio conocimiento del dominio del sistema.

### **4.1.3. Especificación de Requisitos**

Esta fase tiene como propósito estructurar un documento de requisitos para su posterior revisión, evaluación y aprobación.

El modelo propuesto considera la elaboración de un documento de especificación de requisitos a modo de historias de usuario, siguiendo los estándares definidos por cada organización.

Esta parte de la propuesta se inspiró en los hallazgos del SMS, respecto a que las organizaciones que realizan proyectos de software a gran escala en una gran variedad de dominios, han estado reconsiderando su especificación de documentos desde una perspectiva ágil y han estado adoptando el enfoque de historias de usuario para sus requisitos (Daneva, y otros, 2013); mientras que, aquellos que no han adoptado las historias de usuario para sus requisitos de negocio, han rediseñado sus prácticas de especificación de requisitos para “ser más ágiles” (Grewal & Maurer, 2007), (Kendall , Mark, Squires, & Halverson, 2010), (Christou, Ponis, & Palaiologou, 2010), (Gary, y otros, 2011). Todo esto con el fin de llegar a un mejor entendimiento de los requisitos.

### **4.1.4. Validación**

Es preciso validar los requisitos especificados, para garantizar que el ingeniero de software tenga claro todos los requisitos que componen el software a ser desarrollado. En esta fase también se ha propuesto la integración de un proceso de priorización, en vista de que se precisa verificar que el documento de especificación concuerda con los requisitos del usuario (Gambo , Soriyan, & Ikono, 2015).

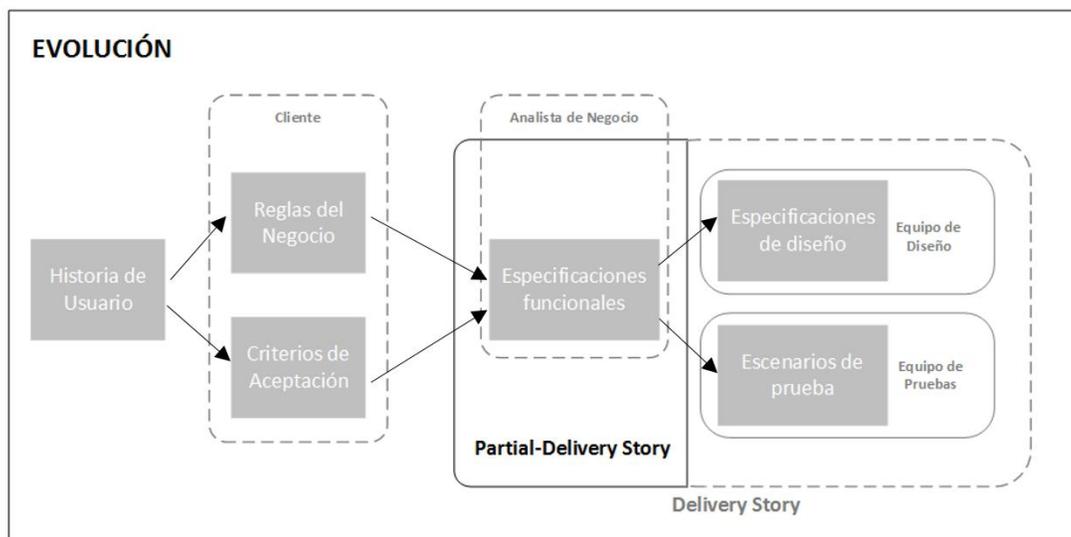
Es preciso validar los requisitos especificados para garantizar que el ingeniero de software haya comprendido todos los requisitos que componen el software a ser desarrollado. Además, se debe verificar que la especificación de requisitos cumpla con todas las expectativas e intereses de los Stakeholders así como con cualquier restricción técnica o estándares de la organización (Gambo, Soriyan, & Ikono, 2015).

De acuerdo al modelo propuesto, durante esta fase también se debe llevar a cabo una priorización de requisitos de modo que se pueda llegar a un consenso entre las discrepancias que puedan surgir de las expectativas de los stakeholders y los requisitos, para lo cual se puede utilizar las mismas herramientas antes mencionadas en la fase de análisis y negociación.

Adicionalmente, el modelo establece que a partir de esta fase se puede regresar a la fase de Elicitación para asegurar el desarrollo eficiente de cada una de las fases antes de continuar con la fase de evolución.

#### **4.1.5. Evolución**

En esta fase se incorporó el concepto “delivery story” (Daneva, y otros, 2013), el cual provee al equipo del proyecto de un recurso más detallado con lo que se necesita hacer para entregar una historia de usuario, ya que muchas veces es difícil llegar al desarrollo, partiendo de la comprensión de las reglas del negocio. Las delivery stories se basan en las historias de usuario, de la mano de la especificación funcional, diseño de alto nivel y escenarios de prueba (ver Figura 29).



**Figura 29.** Fase Evolución.

En primer lugar, el o los analistas de negocios obtienen las historias de usuario del documento validado de especificación de requisitos para reunirse nuevamente con el cliente quien define reglas de negocio junto con criterios de aceptación para cada regla. Los analistas de negocio recogen esta información y la traducen en especificaciones funcionales. En este punto, el equipo de diseño toma parte y crea especificaciones de diseño a partir de las especificaciones funcionales, mientras que el equipo de pruebas las utiliza para crear escenarios de prueba.

El modelo propuesto considera la extensión de la historia de usuario a una especificación funcional, puesto que el diseño de alto nivel y los escenarios de prueba serán desarrollados eventualmente por otros equipos ajenos al de ingeniería de requisitos. Por consiguiente, a esta extensión de la historia de usuario se la ha denominado “Partial-Delivery Story”.

El documento de cada partial-delivery story incluye los siguientes apartados:

- Identificador de delivery story.
- Identificador de historia de usuario.
- Nombre del requisito funcional
- Actor que corresponde al usuario que ejecutará la funcionalidad del requisito.
- Especificaciones funcionales detalladas que debe cumplir el requisito.
- Precondiciones que corresponden a las entradas necesarias para que se cumpla la funcionalidad del requisito a implementar.
- Proceso detallado en pasos.
- Salida.- Resultado esperado de la ejecución del proceso
- Validación.- Criterios de aceptación tomadas de la historia de usuario con los cambios si son necesarios
- Tiempo requerido de implementación
- Prioridad

Las partial-delivery stories son creadas con una estrecha colaboración con los clientes, lo que permite minimizar la brecha en la comprensión que podría existir entre los clientes y los desarrolladores. Si bien esta fase se enfoca en hacer que los requisitos sean más explícitos, se pueden plantear solicitudes de aclaración en caso de ser necesarias.

Una vez finalizado el proceso de IR que conlleva el Modelo propuesto, el resultado final es un documento de especificación de requisitos de software, donde los requisitos son lo suficientemente concretos como para ya ser entregados al equipo de desarrollo a la vez que se minimiza la incertidumbre.

## 4.2.VALIDACIÓN DEL MODELO

La ingeniería de requisitos representa una fase fundamental del proceso desarrollo de software. Su objetivo es recopilar buenos requisitos de los stakeholders de la manera correcta. La ingeniería de requisitos es un ejercicio complejo que considera las demandas de productos desde una gran cantidad de puntos de vista, roles, responsabilidades y objetivos. Por lo tanto, se hace necesario aplicar buenas prácticas de ingeniería de requisitos en cada fase del proceso de desarrollo de software (Pandey, Surman, & Ramani, 2010).

El modelo de procesos de ingeniería de requisitos propuesto nace de una investigación profunda sobre el estado del arte de la IR y está enfocado a afrontar los problemas más comunes a los que esta se enfrenta y cuyo objetivo es producir requisitos de calidad para el desarrollo de software. La Figura 30 ilustra el proceso que se siguió para llevar a cabo el estudio de caso.



**Figura 30.** Proceso de realización de estudios de caso.

Fuente: (Genero Bocco, Cruz Lemus, & Piattini Velthuis, 2014)

### 4.2.1. Diseño y planificación del estudio caso.

A continuación se describen los puntos específicos que se tomaron en cuenta para el diseño del estudio de caso de la presente investigación.

### **Objetivo del Estudio de Caso**

Aplicar el modelo de procesos de Ingeniería de Requisitos propuesto, medir la calidad de los requisitos resultantes y compararlo con el modelo tradicional.

### **Unidad de análisis.**

Modelo de Ingeniería de Requisitos mismo que tiene dos alternativas para este caso: El modelo tradicional utilizado comúnmente por los estudiantes y El modelo propuesto en esta investigación.

### **Preguntas de investigación.**

RQ5: ¿Es posible estructurar un modelo de procesos alternativo de IR?

RQ6: ¿Cuáles son los beneficios obtenidos al aplicar el modelo de procesos alternativo de IR con respecto al tradicional?

### **Participantes.**

Para la ejecución del experimento se seleccionaron alumnos y docentes de dos asignaturas que son parte de la malla de la carrera de Ingeniería de Sistemas de la Universidad de las Fuerzas Armadas – ESPE. A continuación se detallan las características de estos sujetos.

- Asignatura: Desarrollo de Sistemas
- Nivel de estudios: 5to nivel.
- Número de Participantes
  - o Desarrollo de sistemas: 17 estudiantes.
- Género de los sujetos experimentales: Indistinto.

- Rol de los estudiantes: Ingenieros de requisitos.
- Rol del docente: Cliente del sistema (Owner).

Cabe mencionar que se seleccionaron estudiantes de estas asignaturas ya que se asume que estos cuentan con el conocimiento necesario para llevar a cabo el proceso de Ingeniería de requisitos.

### **Recursos para la aplicación del estudio de caso**

Para llevar a cabo el experimento se preparó un kit de trabajo para distribuirlo entre los sujetos del estudio de caso, el mismo que contiene el documento de Documento de Requisitos de Software de Referencia (DRSR) para los docentes que se muestra en la Tabla 18 y Tabla 19 y plantillas de historias de usuario y de partial delivery stories para los estudiantes que se muestra en la Tabla 20 y Tabla 21.

**Tabla 18.**  
*DRSR Sistema A*

<b>Documento de Requisitos del Sistema A</b>
<b>Requisitos Funcionales</b>
<b>RF01.</b> El Sistema debe permitir inscribir a una Concesionaria, incorporando su: Nombre, Dirección, Teléfono y Ruc.
<b>RF02.</b> El Sistema debe permitir modificar datos de una Concesionaria, esto es: Nombre, Dirección, Teléfono y Ruc.
<b>RF03.</b> El Sistema debe permitir dar de baja una Concesionaria, a través del Nombre.
<b>RF04.</b> El sistema debe permitir inscribir un modelo de moto en el catálogo de motos, ingresando: código, nombre modelo, cilindraje, color y precio.
<b>RF05.</b> El sistema debe permitir modificar datos del catálogo para una moto, ingresando su código.
<b>RF06.</b> El sistema debe permitir dar de baja del catálogo una moto a través del código.
<b>RF07.</b> El sistema debe permitir imprimir el catálogo de motos.
<b>RF08.</b> El sistema debe permitir cargar un pedido de motos, incorporando: Número de pedido, Fecha de pedido, Concesionaria (cliente) y de cada moto pedida: código, color, cantidad y Monto de Anticipo.

**CONTINÚA**

---

**Documento de Requisitos del Sistema A**

---

**RF09.** El sistema debe permitir crear un recibo por el anticipo de cada pedido, detallando número de recibo, Numero de pedido, fecha y Monto de Anticipo y a la vez pasar dicho pedido al estado de “fabricación”.

**RF10.** El sistema debe permitir acceder un pedido de fabricación a través del número de pedido para marcarlo como “realizado”.

**RF11.** El sistema deber permitir emitir la factura de venta para los pedidos que figuran como “realizado”, registrando fecha de venta, número de factura, comprador (Concesionaria), el monto total de venta y de cada moto: código, cantidad y precio.

**RF12.** El sistema debe permitir imprimir una planilla de entregas, con los siguientes datos de los pedidos “realizados” y facturados para entregar: nro de pedido, domicilio, saldo restante y además lugar para completar nro de cheque, fecha de entrega del pedido y datos de la persona que recibe.

**RF13.** El sistema debe permitir acceder un pedido de fabricación a través del número de pedido para marcarlo como “entregado”, adicionando la fecha de entrega y nombre de la persona que recibió y nro de cheque.

**RF14.** El sistema debe permitir emitir un listado por pantalla con los pedidos que tienen su estado en “fabricación” detallando para cada pedido: Número de Pedido, Fecha Pedido, Cliente y por cada moto del pedido: código, color, cantidad, Monto de Anticipo.

**RF15.** El sistema debe permitir emitir un listado por pantalla con los pedidos que tienen su estado “realizado” detallando para cada pedido: Número de Pedido, Fecha Pedido, Cliente y por cada moto del pedido: código, color, cantidad, Monto de Anticipo.

**RF16.** El sistema debe permitir mostrar para un determinado mes los pedidos entregados en un mes a una concesionaria en particular, detallando: N° Pedido y Monto.

**RF17.** El sistema debe permitir mostrar las ventas registradas entre dos fechas, detallando Número de Venta, Fecha de la Venta, Comprador y Número de Pedido.

**RF18.** El sistema debe permitir mostrar el o los modelos de moto más vendida en un determinado periodo de tiempo, detallando código, nombre de modelo, cilindradas y la cantidad vendida.

**RF19.** El sistema debe permitir administrar tres tipos de cuentas de usuarios, estos son: Encargado de Ventas, Encargado de Fábrica y Gerente.

**RF20.** El sistema debe permitir al usuario Encargado de Ventas registrar y actualizar los pedidos, realizar facturación y todas las consultas antes indicadas.

**RF21.** El sistema debe permitir al usuario Encargado de Fabrica consultar pedidos pendientes de entrega (accediendo a todos los datos de los mismos) y marcar como “realizados” las notas de pedido fabricadas.

**RF22.** El sistema debe permitir al usuario Gerente tener acceso a todas las funcionalidades del sistema.

---

**Tabla 19.**  
*DRSR Sistema B*

<b>Documento de Requisitos del Sistema</b>
<b>Requisitos Funcionales</b>
<b>RF01.</b> El Sistema debe permitir agregar clientes, incorporando su: Ruc, Nombre, Dirección, Teléfono y correo electrónico.
<b>RF02.</b> El Sistema debe permitir modificar datos de clientes esto es: Ruc, Nombre, Dirección, Teléfono y correo electrónico.
<b>RF03.</b> El Sistema debe permitir agregar productos incorporando su: Código, Nombre, Precio de envío, condición “frágil/ no frágil.
<b>RF04.</b> El sistema debe permitir modificar datos de productos, ingresando su código.
<b>RF05.</b> El sistema debe permitir dar de baja un producto a través del código.
<b>RF06.</b> El sistema debe permitir ingresar las zonas de envío (Quito, Machala, Guayaquil, Provincia de Loja) ingresando su: código y nombre.
<b>RF07.</b> El sistema debe permitir dar de baja las zonas de envío, a través del código
<b>RF08.</b> El sistema debe permitir ingresar los datos del transportista, ingresando su: código, nombre, numero de camión, teléfono
<b>RF09.</b> El sistema debe permitir consultar todos los productos existentes para añadirlos en la guía de remisión.
<b>RF10.</b> El sistema debe permitir la emisión una guía de remisión para el envío de productos, detallando número de guía, Remitente, Destinatario, Dirección del destinatario, Teléfono, producto(s), zona, transportista, nota (“cancelado” o “pendiente de cobro”), estado (“entregado”, “pendiente de entrega”).
<b>RF11.</b> El sistema debe permitir cambiar la nota de una guía de remisión una vez que el producto(s) ha sido entregado.
<b>RF12.</b> El sistema debe permitir cambiar la nota de una guía de remisión una vez que esta sea cancelada.
<b>RF13.</b> El sistema debe permitir mostrar un listado por pantalla con las guías de remisión que tienen su estado “cancelando” detallando para cada guía de remisión: Número de Pedido, Fecha Pedido, Remitente, Destinatario, Producto, Detalle de Producto, precio.
<b>RF14.</b> El sistema debe permitir mostrar un listado por pantalla con las guías que tienen su estado en “pendiente de cobro” detallando para cada para cada guía de remisión: Número de Pedido, Fecha Pedido, Remitente, Destinatario, Producto, Detalle de Producto, precio.
<b>RF15.</b> El sistema debe permitir mostrar para un determinado mes las guías de remisión de productos entregados en un mes.
<b>RF16.</b> El sistema debe permitir mostrar el total del valor recaudado por el servicio de transporte de cada transportista por mes.

**CONTINÚA**

---

**Documento de Requisitos del Sistema**

---

**RF17.** El sistema debe permitir administrar tres tipos de cuentas de usuarios, estos son: Administrador y el encargado de logística.

**RF18.** El sistema debe permitir al usuario Encargado de logística registrar, modificar y eliminar zonas de envío, productos y transportistas.

**RF19.** El sistema debe permitir al usuario administrador registrar las guías de remisión y hacer todas las consultas antes mencionadas.

---

**Tabla 20.**

*Plantilla Historia de Usuario*

<b>Id. Historia de usuario</b>	HUS-XXX
<b>Prioridad</b>	Alta-Media-Baja
<b>Nombre del Requisito</b>	Nombre del requisito funcional a ser implementado
<b>Rol</b>	Usuario que ejecutará la funcionalidad del requisito.
<b>Característica /Funcionalidad</b>	Descripción de la funcionalidad a modo de dialogo o monologo.
<b>Validación</b>	Criterios de aceptación.

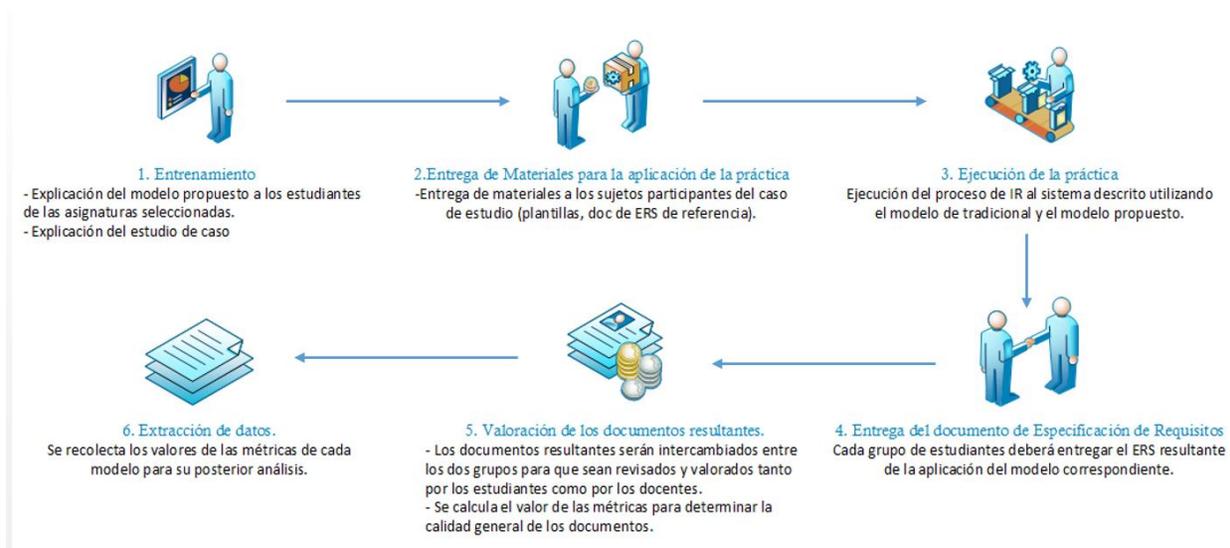
**Tabla 21.**

*Plantilla Partial Delivery Story*

<b>Id. Partial-Delivery Story</b>	PDS-XXX
<b>Id. Historia de Usuario</b>	HUS-XXX
<b>Nombre del requisito</b>	Nombre del requisito funcional a ser implementado
<b>Actor</b>	Usuario que ejecutará la funcionalidad del requisito.
<b>Especificaciones funcionales</b>	A partir de las reglas de negocio y criterios de aceptación provenientes de la historia de usuario en concordancia con el usuario, se describe de forma detallada las especificaciones que debe cumplir el requisito.
<b>Precondiciones</b>	Entradas necesarias para que se cumpla la funcionalidad del requisito a implementar.
<b>Proceso</b>	Describir el proceso a partir de las especificaciones detalladas y de las precondiciones que se deben tener en cuenta para el requisito.
<b>Salida</b>	Resultado esperado de la ejecución del proceso (pantallas de alerta o confirmación en caso de que se requiera).
<b>Validación</b>	Criterios de aceptación tomadas de la historia de usuario con los cambios si son necesarios.
<b>Tiempo de implementación</b>	Tiempo requerido para la implementación de la delivery story (especificaciones funcionales) en días.
<b>Prioridad</b>	Alta-Media-Baja

## Esquema de actividades

La figura a continuación contiene el proceso de aplicación del estudio de caso.



**Figura 31.** Proceso de aplicación del estudio de caso

### 4.2.2. Preparación y recolección de datos

#### Método

Método para recolección de datos independiente (tercer grado).

De acuerdo con (Genero Bocco, Cruz Lemus, & Piattini Velthuis, 2014), este método se refiere al análisis independiente de artefactos de trabajo donde se utilizan datos previamente disponibles. En este grupo se enmarcan el estudio de documentos tales como informes de fallos de una organización así como especificaciones de requisitos, siendo este último el artefacto que se va a considerar para este estudio.

**Variable a estudiar**

Calidad del documento de especificación de requisitos de software resultante.

**Métricas**

Se aplicaron cuatro métricas diferentes para evaluar la calidad general de los documentos de especificación de requisitos de software las cuales fueron aplicadas por (Lloyd, Rosson, & Arthur, 2002) (Sevilla G. , Zapata , Torres, & Collazo, 2014) (Lloyd, 2001) (Sevilla, Zapata, Collazos , & Giraldo, 2016) y se explican a continuación.

- *Nivel de adecuación general del documento (NAD).*- Mide la calidad del documento desde la perspectiva de los estudiantes de la asignatura así como de los investigadores. Para obtener este valor, los documentos resultantes se intercambiaron entre cada grupo conformado por los estudiantes de las asignatura seleccionada para el caso de estudio; donde cada grupo de estudiantes evaluó el documento de ERS producido por el otro grupo y al mismo tiempo los investigadores hicieron su propia valoración para cada documento. Finalmente estos valores se promedian para obtener el NAD.
- *Porcentaje de requisitos evolucionados (RE).*- Mide la madurez del documento de especificación de requisitos de software. Para obtener este valor, todos los requisitos identificados se clasificaron como Original (O), Original con evolución (OE), Evolucionado (E) o Ambiguo (A). La Tabla 22 describe cada una de estas clasificaciones.

**Tabla 22.**  
*Tipos de Requisitos para la métrica de evolución*

Tipo de requisito	Descripción
<b>O: Requisito Original</b>	<ul style="list-style-type: none"> <li>- Requisito original completo del documento de especificación de requisitos de referencia.</li> <li>- Requisito original parcial del documento de especificación de requisitos de referencia.</li> </ul>
<b>OE: Requisito original con evolución</b>	<ul style="list-style-type: none"> <li>- Requisito original con nuevas adiciones</li> <li>- Requisito original parcial con nuevas adiciones.</li> <li>- Requisito original con cambios.</li> </ul>
<b>E: Requisito evolucionado</b>	- Requisito completamente nuevo no mencionado en el documento de especificación de requisitos de referencia.
<b>A: Requisito ambiguo</b>	- Requisito tan ambiguo que no puede ser calificado dentro de los otros tipos.

- Una vez clasificados los requisitos se deben calcular el valor de la métrica aplicando la siguiente fórmula:

$$\text{Evolución del documento} = \frac{OE_{TOTAL} + E_{TOTAL}}{REQ_{TOTAL}}$$

Donde,

OE = Requisito original con evolución

E = Requisito evolucionado

REQ<sub>TOTAL</sub> = Total de requisitos identificados.

- *Porcentaje de requisito sin defecto (RSD).*- Permite identificar la claridad y la integridad de los requisitos del documento de ERS. Para calcular este valor se debía determinar si un requisito presentaba algún tipo de defecto como ambigüedad, incompletitud, inconsistencia, entre otros. Una vez identificados los defectos se aplica la siguiente formula:

$$\text{Porcentaje de requisitos sin defectos} = 1 - \left[ \frac{REQ_{CON DEFECTOS}}{REQ_{TOTAL}} \right]$$

Donde,

$REQ_{CON DEFECTOS}$  = Total de requisitos con uno o más defectos

$REQ_{TOTAL}$  = Total de requisitos identificados.

- *Porcentaje de requisitos soportados (RS).*- Esta métrica es la que mejor representa un proceso de ingeniería de requisitos exitoso. Mide el porcentaje de requisitos que estando en el documento de Requisitos de Software de referencia (DRSR), también se encuentran en el documento de ERS resultante. Para calcular este valor se aplica la siguiente formula:

$$\text{Porcentaje de requisitos soportados} = \frac{REQ_{SOPORTADOS}}{REQ_{ORIGINALES}}$$

Donde,

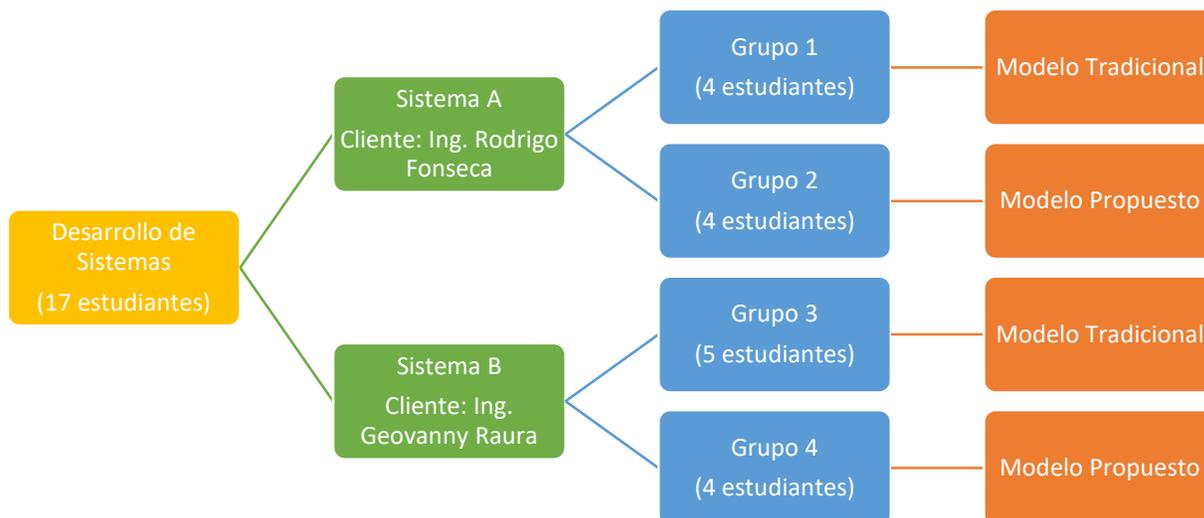
$REQ_{SOPORTADOS}$  = Número de requisitos que se encuentran tanto en el DRSR así como en el ERS resultante.

$REQ_{ORIGINALES}$  = Total de requisitos que se encuentran en el ERS de referencia.

Cada métrica produce un valor decimal positivo de 0 a 1. Un promedio igualmente ponderado de todas las métricas tomadas en conjunto produce un resultado numérico que refleja la calidad general de los documentos, que se utiliza para la presentación y el análisis de los resultados, como lo indica la siguiente fórmula:

$$\text{Calidad General del documento resultante} = \frac{NAD + RE + RSD + RS}{4}$$

Para la recolección de datos se dividió el curso de Desarrollo de Sistemas a cargo del Ingeniero Geovanny Raura en 3 grupos de 4 estudiantes y 1 grupo de 5 estudiantes dando un total de 17 estudiantes. A continuación se eligieron 2 grupos de forma aleatoria para lleven a cabo el proceso de Ingeniería de Requisitos aplicando el modelo propuesto mientras que los otros 2 grupos debían ejecutar el mismo proceso aplicando el modelo tradicional. Para aplicar los modelos designados a los distintos grupo se establecieron dos sistemas para los cuales se definió un documento de requisitos de referencia (Ver Sección 4.2.1 Recursos para la aplicación del estudio de caso) que fue entregado a dos ingenieros quienes debían hacer el rol de cliente del sistema. La Figura 32 muestra la configuración del estudio de caso.



**Figura 32.** Configuración de los grupos de trabajo.

La aplicación del estudio de caso se llevó a cabo siguiendo el cronograma de actividades que se detalla en la Figura 33.

ACTIVIDADES	MARZO					ABRIL																		
	I	J	V	S	D	L	M	I	J	V	S	D	L	M	I	J	V	S	D	L	M	I	J	V
	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Introducción	█																							
<b>Fase 1: Elicitación de requisitos</b>	█	█	█	█	█																			
Ejecución Fase 1	█	█	█	█	█																			
Entrega de avances Fase 1						█	█	█	█	█														
<b>Fase 2: Análisis y Negociación de requisitos</b>						█	█	█	█	█														
Ejecución Fase 2						█	█	█	█	█														
Entrega de avances Fase 2											█	█	█	█	█									
<b>Fase 3: Especificación de requisitos</b>											█	█	█	█	█									
Ejecución Fase 3											█	█	█	█	█									
Entrega de avances Fase 3																█	█	█	█	█				
<b>Fase 4: Validación de requisitos</b>																█	█	█	█	█				
Ejecución Fase 4																█	█	█	█	█				
Entrega de avances Fase 4																								
<b>Fase 5: Evolución de requisitos</b>																								
Ejecución Fase 5																								
Entrega de avances Fase 5																								
Valoración del documento de ERS																								█

**Figura 33.** Cronograma de Aplicación del Estudio de Caso.

### 4.2.3. Análisis y presentación de datos

Para la obtención de datos se analizaron los documentos de especificación de requisitos resultante de cada grupo de estudiantes y se calcularon los valores correspondientes a cada métrica para obtener la calidad general de los documentos resultantes. A continuación se detallan los valores obtenidos para cada métrica.

**Nivel de adecuación general del documento (NAD).** Una vez realizado el proceso de Ingeniería de Requisitos se solicitó a los estudiantes que se intercambiaran los documentos de especificación de requisitos para que realizaran una valoración cruzada. A continuación cada investigador realizó la respectiva valoración de cada uno de los cuatro documentos. Estas valoraciones se realizaron en la escala del 1 al 20 y se calcularon los porcentajes correspondientes esta información se encuentra detallada en la Tabla 23.

**Tabla 23.**

*Nivel de adecuación del documento (NAD)*

GRUPO	SISTEMA	MODELO	ALUMNOS	INV 1	INV 2	PROM	RESULTADO
1	A	Tradicional	16,5	14,4	15,4	15,43	<b>77%</b>
2	A	Propuesto	14	14	15	14,33	<b>72%</b>
3	B	Tradicional	18	16	16	16,67	<b>83%</b>
4	B	Propuesto	18,25	17,5	18,5	18,08	<b>90%</b>

**Porcentaje de requisitos evolucionados (RE).** Para obtener este valor, todos los requisitos identificados en cada uno de los documentos de especificación de requisitos resultantes se clasificaron como Original (O), Original con evolución (OE), Evolucionado (E) o Ambiguo (A). A continuación se aplicó la formula descrita en la sección 4.2.2 de este capítulo. Las Tablas a

continuación describen como se obtuvo el valor del porcentaje de requisitos evolucionados (RE) para cada ERS resultante.

**Tabla 24.**  
*Requisitos Evolucionados Grupo 1*

REQUISITO	O	OE	E	A
1	X			
2		X		
3		X		
4				X
5	X			
6			X	
7	X			
8	X			
9			X	
10	X			
11		X		
12		X		
13			X	
14			X	
15	X			
16		X		
17	X			
18	X			
19	X			
20	X			
<b>TOTAL</b>	<b>10</b>	<b>5</b>	<b>4</b>	<b>1</b>

Una vez clasificados los requisitos del ERS resultante del grupo 1 (G1) se calculó el valor de la métrica aplicando la fórmula:

$$\text{Evolución del documento}_{G1} = \frac{5 + 4}{20}$$

$$\text{Evolución del documento}_{G1} = 45 \%$$

**Tabla 25.**  
*Requisitos Evolucionados Grupo 2*

REQUISITO	O	OE	E	A
1			X	
2			X	
3			X	
4			X	
5	X			
6	X			
7		X		
8	X			
9	X			
10			X	
11			X	
12	X			
13	X			
14		X		
15				X
16			X	
17			X	
18			X	
19			X	
20	X			
21	X			
22			X	
23	X			
<b>TOTAL</b>	<b>9</b>	<b>2</b>	<b>11</b>	<b>1</b>

Una vez clasificados los requisitos del ERS resultante del grupo 2 (G2) se calculó el valor de la métrica aplicando la fórmula:

$$\text{Evolución del documento}_{G2} = \frac{11 + 2}{23}$$

$$\text{Evolución del documento}_{G2} = 57\%$$

**Tabla 26.**  
*Requisitos Evolucionados Grupo 3*

REQUISITO	O	OE	E	A
1	X			
2	X			
3	X			
4			X	
5			X	
6	X			
7		X		
8		X		
9	X			
10	X			
11				X
12			X	
13		X		
14	X			
15			X	
16			X	
17			X	
18	X			
19			X	
20		X		
21			X	
<b>TOTAL</b>	<b>8</b>	<b>4</b>	<b>8</b>	<b>1</b>

Una vez clasificados los requisitos del ERS resultante del grupo 3 (G3) se calculó el valor de la métrica aplicando la fórmula:

$$\text{Evolución del documento}_{G3} = \frac{4 + 8}{21}$$

$$\text{Evolución del documento}_{G3} = 57\%$$

**Tabla 27.**  
*Requisitos Evolucionados Grupo 4*

REQUISITO	O	OE	E	A
1		X		
2		X		
3		X		
4		X		
5		X		
6		X		
7			X	
<b>TOTAL</b>	<b>0</b>	<b>6</b>	<b>1</b>	<b>0</b>

Una vez clasificados los requisitos del ERS resultante del grupo 4 (G4) se calculó el valor de la métrica aplicando la fórmula:

$$\text{Evolución del documento}_{G4} = \frac{6 + 1}{7} = 100\%$$

**Porcentaje de requisitos sin defectos (RSD).** Para obtener este valor, se determinó si algún requisito presentaba algún tipo de defecto (ambiguo, incompleto, entre otros) todo esto se realizó para cada documento de ERS. A continuación se aplicó la fórmula descrita en la sección 4.2.2. Las tablas a continuación presentan la información de los requisitos con defecto encontrados.

**Tabla 28.**  
*Requisitos con Defectos encontrados en la ERS del Grupo 1*

REQUISITO	DEFECTO
1	NO
2	SI
3	SI
4	NO
5	SI
6	SI

**CONTINÚA**

REQUISITO	DEFECTO
7	NO
8	SI
9	SI
10	SI
11	SI
12	SI
13	NO
14	SI
15	SI
16	SI
17	NO
18	NO
19	NO
20	NO
<b>TOTAL DEFECTOS</b>	<b>12</b>

Una vez identificados el número de requisitos con defecto de la ERS resultante del grupo 1 (G1) se calculó el valor de la métrica aplicando la fórmula:

$$\text{Porcentaje de requisitos sin defectos}_{G1} = 1 - \left[ \frac{12}{20} \right] = 40\%$$

**Tabla 29.**

*Requisitos con Defectos encontrados en la ERS del Grupo 2*

REQUISITO	DEFECTO
1	NO
2	NO
3	SI
4	SI
5	SI
6	SI
7	SI
8	SI
9	SI
10	NO

**CONTINÚA**

REQUISITO	DEFECTO
11	NO
12	NO
13	NO
14	NO
15	NO
16	NO
17	SI
18	NO
19	SI
20	SI
21	SI
22	NO
23	NO
<b>TOTAL DEFECTOS</b>	<b>11</b>

Una vez identificados el número de requisitos con defecto de la ERS resultante del grupo 2 (G2) se calculó el valor de la métrica aplicando la fórmula:

$$\text{Porcentaje de requisitos sin defectos}_{G2} = 1 - \left[ \frac{11}{23} \right] = 52\%$$

**Tabla 30.**

*Requisitos con Defectos encontrados en la ERS del Grupo 3*

REQUISITO	DEFECTO
1	SI
2	SI
3	NO
4	SI
5	NO
6	NO
7	NO
8	NO
9	SI
10	SI
11	SI

**CONTINÚA**

REQUISITO	DEFECTO
12	NO
13	NO
14	NO
15	NO
16	NO
17	SI
18	NO
19	NO
20	NO
21	SI
<b>TOTAL DEFECTOS</b>	<b>8</b>

Una vez identificados el número de requisitos con defecto de la ERS resultante del grupo 3 (G3) se calculó el valor de la métrica aplicando la fórmula:

$$\text{Porcentaje de requisitos sin defectos}_{G2} = 1 - \left[ \frac{8}{21} \right] = 62\%$$

**Tabla 31.**

*Requisitos con Defectos encontrados en la ERS del Grupo 4*

REQUISITO	DEFECTO
1	NO
2	NO
3	NO
4	NO
5	SI
6	NO
7	SI
<b>TOTAL DEFECTOS</b>	<b>2</b>

$$\text{Porcentaje de requisitos sin defectos}_{G2} = 1 - \left[ \frac{2}{7} \right] = 71\%$$

**Porcentaje de requisitos soportados (RS).** Para calcular este valor se determinó el número de requisitos que se encuentran tanto en los documentos de requisitos de referencia (Ver apartado 4.2.1. Recursos para la aplicación del estudio de caso), así como en los documentos de especificación de requisitos resultantes. A continuación la Tabla 32 muestra el porcentaje de requisitos soportados del Sistema A, obtenidos de los documentos de especificación de requisitos de los grupos 1 (modelo tradicional) y 2 (modelo propuesto).

**Tabla 32.**  
*Porcentaje de Requisitos Soportados Sistema A (G1 y G2)*

<b>DRSR SISTEMA A</b>	<b>REQUISITOS G1</b>	<b>REQUISITOS G2</b>
RF01. El Sistema debe permitir inscribir a una Concesionaria, incorporando su: Nombre, Dirección, Teléfono y Ruc.	X	X
RF02. El Sistema debe permitir modificar datos de una Concesionaria, esto es: Nombre, Dirección, Teléfono y Ruc.	X	X
RF03. El Sistema debe permitir dar de baja una Concesionaria, a través del Nombre.		X
RF04. El sistema debe permitir inscribir un modelo de moto en el catálogo de motos, ingresando: código, nombre modelo, cilindraje, color y precio.	X	X
RF05. El sistema debe permitir modificar datos del catálogo para una moto, ingresando su código.	X	X
RF06. El sistema debe permitir dar de baja del catálogo una moto a través del código.	X	X
RF07. El sistema debe permitir imprimir el catálogo de motos.	X	X
RF08. El sistema debe permitir cargar un pedido de motos, incorporando: Número de pedido, Fecha de pedido, Concesionaria (cliente) y de cada moto pedida: código, color, cantidad y Monto de Anticipo.	X	X
RF09. El sistema debe permitir crear un recibo por el anticipo de cada pedido, detallando número de recibo, Numero de pedido, fecha y Monto de Anticipo y a la vez pasar dicho pedido al estado de “fabricación”.		

**CONTINÚA**

DRSR SISTEMA A	REQUISITOS G1	REQUISITOS G2
RF09. El sistema debe permitir crear un recibo por el anticipo de cada pedido, detallando número de recibo, Numero de pedido, fecha y Monto de Anticipo y a la vez pasar dicho pedido al estado de “fabricación”.		
RF10. El sistema debe permitir acceder un pedido de fabricación a través del número de pedido para marcarlo como “realizado”.		
RF11. El sistema deber permitir emitir la factura de venta para los pedidos que figuran como “realizado”, registrando fecha de venta, número de factura, comprador (Concesionaria), el monto total de venta y de cada moto: código, cantidad y precio.	X	
RF12. El sistema debe permitir imprimir una planilla de entregas, con los siguientes datos de los pedidos “realizados” y facturados para entregar: nro de pedido, domicilio, saldo restante y además lugar para completar nro de cheque, fecha de entrega del pedido y datos de la persona que recibe.		X
RF13. El sistema debe permitir acceder un pedido de fabricación a través del número de pedido para marcarlo como “entregado”, adicionando la fecha de entrega y nombre de la persona que recibió y nro de cheque.	X	
RF14. El sistema debe permitir emitir un listado por pantalla con los pedidos que tienen su estado en “fabricación” detallando para cada pedido: Número de Pedido, Fecha Pedido, Cliente y por cada moto del pedido: código, color, cantidad, Monto de Anticipo.	X	X
RF15. El sistema debe permitir emitir un listado por pantalla con los pedidos que tienen su estado “realizado” detallando para cada pedido: Número de Pedido, Fecha Pedido, Cliente y por cada moto del pedido: código, color, cantidad, Monto de Anticipo.		X
RF16. El sistema debe permitir mostrar para un determinado mes los pedidos entregados en un mes a una concesionaria en particular, detallando: N° Pedido y Monto.		
RF17. El sistema debe permitir mostrar las ventas registradas entre dos fechas, detallando Número de Venta, Fecha de la Venta, Comprador y Número de Pedido.		
RF18. El sistema debe permitir mostrar el o los modelos de moto más vendida en un determinado periodo de tiempo, detallando código, nombre de modelo, cilindradas y la cantidad vendida.		

**CONTINÚA**

<b>DRSR SISTEMA A</b>	<b>REQUISITOS G1</b>	<b>REQUISITOS G2</b>
RF19. El sistema debe permitir administrar tres tipos de cuentas de usuarios, estos son: Encargado de Ventas, Encargado de Fábrica y Gerente.	X	
RF20. El sistema debe permitir al usuario Encargado de Ventas registrar y actualizar los pedidos, realizar facturación y todas las consultas antes indicadas.	X	
RF21. El sistema debe permitir al usuario Encargado de Fabrica consultar pedidos pendientes de entrega (accediendo a todos los datos de los mismos) y marcar como “realizados” las notas de pedido fabricadas.		
RF22. El sistema debe permitir al usuario Gerente tener acceso a todas las funcionalidades del sistema.		
<b>TOTAL</b>	<b>12</b>	<b>11</b>
<b>REQUISITOS SOPORTADOS</b>	<b>55%</b>	<b>50%</b>

La Tabla 33 muestra el porcentaje de requisitos soportados del Sistema B, obtenidos de los documentos de ERS de los grupos 3 (modelo tradicional) y 4 (modelo propuesto).

**Tabla 33.**  
*Porcentaje de Requisitos Soportados Sistema B (G3 y G4)*

<b>DRSR SISTEMA A</b>	<b>REQUISITOS G3</b>	<b>REQUISITOS G4</b>
RF01. El Sistema debe permitir agregar clientes, incorporando su: Ruc, Nombre, Dirección, Teléfono y correo electrónico.	X	X
RF02. El Sistema debe permitir modificar datos de clientes esto es: Ruc, Nombre, Dirección, Teléfono y correo electrónico.	X	X
RF03. El Sistema debe permitir agregar productos incorporando su: Código, Nombre, Precio de envío, condición “frágil/ no frágil.	X	X
RF04. El sistema debe permitir modificar datos de productos, ingresando su código.	X	X
RF05. El sistema debe permitir dar de baja un producto a través del código.	X	X
RF06. El sistema debe permitir ingresar las zonas de envío (Quito, Machala, Guayaquil, Provincia de Loja) ingresando su: código y nombre.	X	X
RF07. El sistema debe permitir dar de baja las zonas de envío, a través del código	X	X

**CONTINÚA**

<b>DRSR SISTEMA A</b>	<b>REQUISITOS G3</b>	<b>REQUISITOS G4</b>
RF08. El sistema debe permitir ingresar los datos del transportista, ingresando su: código, nombre, numero de camión, teléfono	X	X
RF09. El sistema debe permitir consultar todos los productos existentes para añadirlos en la guía de remisión.	X	X
RF10. El sistema debe permitir la emisión una guía de remisión para el envío de productos, detallando número de guía, Remitente, Destinatario, Dirección del destinatario, Teléfono, producto(s), zona, transportista, nota (“cancelado” o “pendiente de cobro”), estado (“entregado”, “pendiente de entrega”).	X	X
RF11. El sistema debe permitir cambiar la nota de una guía de remisión una vez que el producto(s) ha sido entregado.		
RF12. El sistema debe permitir cambiar la nota de una guía de remisión una vez que esta sea cancelada.		
RF13. El sistema debe permitir mostrar un listado por pantalla con las guías de remisión que tienen su estado “cancelando” detallando para cada guía de remisión: Número de Pedido, Fecha Pedido, Remitente, Destinatario, Producto, Detalle de Producto, precio.	X	X
RF14. El sistema debe permitir mostrar un listado por pantalla con las guías que tienen su estado en “pendiente de cobro” detallando para cada para cada guía de remisión: Número de Pedido, Fecha Pedido, Remitente, Destinatario, Producto, Detalle de Producto, precio.	X	X
RF15. El sistema debe permitir mostrar para un determinado mes las guías de remisión de productos entregados en un mes.		
RF16. El sistema debe permitir mostrar el total del valor recaudado por el servicio de transporte de cada transportista por mes.		
RF17. El sistema debe permitir administrar tres tipos de cuentas de usuarios, estos son: Administrador y el encargado de logística.		X
RF18. El sistema debe permitir al usuario Encargado de logística registrar, modificar y eliminar zonas de envío, productos y transportistas.	X	X
RF19. El sistema debe permitir al usuario administrador registrar las guías de remisión y hacer todas las consultas antes mencionadas.	X	X
<b>TOTAL</b>	<b>14</b>	<b>15</b>
<b>REQUISITOS SOPORTADOS</b>	<b>74%</b>	<b>79%</b>

Con los valores resultantes de cada métrica se obtuvo la calidad general de los documentos de ERS resultantes aplicando la siguiente fórmula:

$$\text{Calidad General del documento resultante} = \frac{NAD + RE + RSD + RS}{4}$$

En la Tabla 34 se presentan los valores obtenidos por cada documento de especificación de requisitos producido por cada grupo para ambos modelos y sistemas.

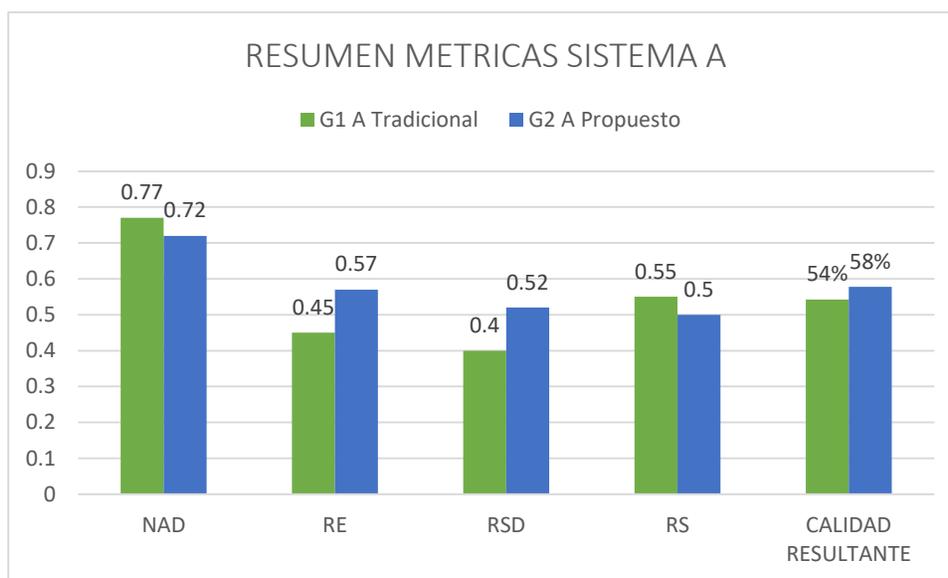
**Tabla 34.**

*Calidad General de los Documentos de Especificación de Requisitos resultantes.*

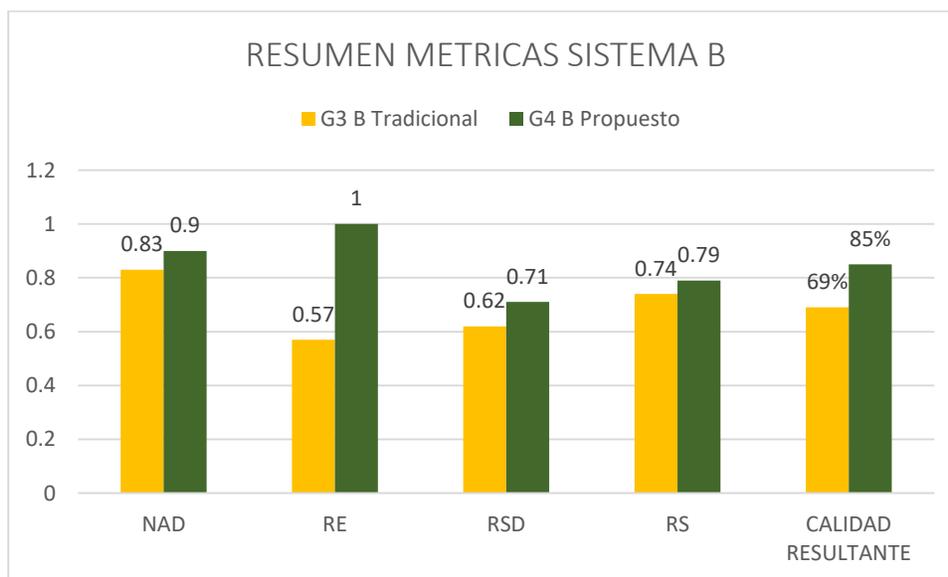
GRUPO	SISTEMA	MODELO	NAD	RE	RSD	RS	VALOR TOTAL
G1	A	Tradicional	0,77	0,45	0,4	0,55	54%
G2	A	Propuesto	0,72	0,57	0,52	0,5	58%
G3	B	Tradicional	0,83	0,57	0,62	0,74	69%
G4	B	Propuesto	0,9	1	0,71	0,79	85%

Una vez procesados los resultados obtenidos, se hace evidente que tanto para el sistema A como para el sistema B, la calidad del documento de especificación de requisitos de software resultante de aplicar el modelo propuesto es superior en comparación a aquellos que aplicaron el modelo tradicional, siendo un 4% superior en el caso del sistema A y un 16% superior en el caso del sistema B. De este análisis se concluye que el modelo propuesto contribuye a la mejora de la calidad de los documentos de especificación de requisitos de software resultantes apegados al marco común del estudio de caso principal.

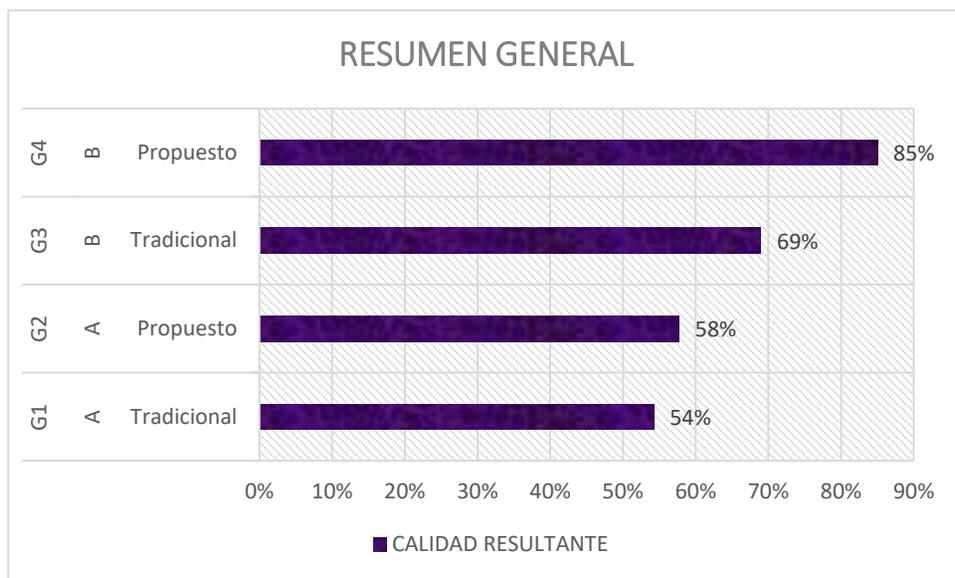
A modo de resumen se presentan los siguientes gráficos (Figura 34, Figura 35, Figura 36) donde se detallan los valores obtenidos para cada métrica según cada sistema y modelo de Ingeniería de Requisitos aplicado.



**Figura 34.** Métricas de calidad de los documentos del sistema A.



**Figura 35.** Métricas de calidad de los documentos del sistema B.



**Figura 36.** Resumen general de la calidad resultante según el modelo aplicado.

## **CAPÍTULO V**

### **CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS**

#### **5.1. CONCLUSIONES**

Existe una gran cantidad de estudios en torno a la Ingeniería de Requisitos y el interés por esta temática no ha hecho más que aumentar en los años recientes, de hecho, nuestros hallazgos muestran que un gran número de estos estudios son revisiones de literatura. Sin embargo, el número de estudios terciarios que permitan agrupar todo ese conocimiento generado por las revisiones anteriores es muy limitado, y poco o nada se hace más allá de agrupar el conocimiento sin una aplicación real del mismo.

Se identificaron varios problemas inherentes al proceso de Ingeniería de Requisitos como la falta de comunicación, falta de retroalimentación, falta de colaboración del cliente, entre otros; y pese a que algunos son citados con mayor frecuencia que otros dentro de la literatura, la magnitud del impacto sobre la calidad del software resultante en caso de presentarse uno de estos problemas no es directamente proporcional al número de citas.

En relación a las causas identificadas, se debe tomar en consideración que una causa en particular puede estar relacionada a más de un problema, motivo por el cual se organizaron los problemas identificados dentro de 6 categorías y se estableció la relación existente entre categorías de problemas y causas a través de diagramas de causa-efecto, de modo que se pudo obtener una visión clara de las causas más críticas de los problemas que se presentan en el proceso de IR.

Se identificó que dentro de la literatura existe una discrepancia entre los investigadores sobre qué debería ser considerado como un problema y qué debería ser considerado como causa, haciendo evidente la necesidad de llegar a un consenso que permita identificar claramente cuáles son los problemas y causas que afectan de forma crítica a la IR..

El modelo propuesto presenta varios beneficios en comparación al modelo tradicional como son: mejor coordinación del equipo de trabajo, reducción en la dificultad para extraer información relevante de los stakeholders, comunicación más eficiente entre los stakeholders y el equipo de trabajo, utilización de terminologías y notaciones comunes, documentación con enfoque ágil (historias de usuario), reducción de las discrepancias que puedan surgir entre los stakeholders y los requisitos, y finalmente mejora en la calidad general del documento de especificación de requisitos resultante.

El proceso de Ingeniería de Requisitos, independientemente del modelo que se aplique, debe estar claramente definido y las diferentes técnicas que han de utilizarse para cada fase deben ser seleccionadas en base a recomendaciones y buenas prácticas en la industria a fin de garantizar que los productos generados durante su ejecución sean de la mejor calidad posible.

## **5.2. RECOMENDACIONES**

Se recomienda utilizar ontologías como sustento en las actividades de elicitación de requisitos ya que estas promueven un vocabulario de dominio compartido que se puede utilizar para evitar ambigüedades; así como para desarrollar un lenguaje común que se puede aplicar en varios proyectos.

Si se cuenta con un equipo de desarrollo completo, se recomienda que cuando se estén llevando a cabo la fase de Evolución se aplique todo el proceso que conlleva la Delivery Story.

Al momento de diseñar el estudio de caso se recomienda planificar todas las actividades que conllevará el mismo, de modo que todo el proceso pueda ser controlado y los resultados sean más fiables.

Aplicar el modelo de Ingeniería de Requisitos propuesto en un proceso de desarrollo de software más amplio, donde los requisitos continuamente están cambiando, esto para comprobar la efectividad de la aplicación del modelo en diversos escenarios y dominios de proyectos de Software.

## **5.3. TRABAJOS FUTUROS**

El concepto de Delivery Story puede estudiarse y ampliarse aún más ya que al ser tan novedoso se puede obtener un margen de mejora para el mismo.

El modelo propuesto puede ser evaluado en mayor profundidad si se considera como parte del análisis de resultados el producto software generado al completar todo el ciclo de desarrollo y no únicamente el documento de requisitos resultante del proceso de Ingeniería de Requisitos.

## REFERENCIAS BIBLIOGRÁFICAS

- Ambreen, T., Ikram, N., Usman, M., & Niazi, M. (2018). Empirical research in requirements engineering: trends and opportunities. *Requirements Engineering Journal*, 63-95.
- Anu, V., Hu, W., C Carverc, J., S Waliaa, G., & Bradshaw, G. (2018). Development of a human error taxonomy for software requirements: A systematic literature review . *Information and Software Technology*, 112-124.
- Bano, M., & Zowghi, D. (2013). Users' involvement in requirements engineering and system success. *2013 3rd International Workshop on Empirical Requirements Engineering (EmpiRE)*. Rio de Janeiro.
- Bano, M., Zowghi, D., & Ikram, N. (2014). Systematic reviews in requirements engineering: A tertiary study. *2014 IEEE 4th International Workshop on Empirical Requirements Engineering (EmpiRE)* (págs. 9-16). IEEE: Karlskrona.
- Barmi, Z., Ebrahimi, A., & Feldt, R. (2011). Alignment of requirements specification and testing: A systematic mapping study. *2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops* (págs. 476-485). Washington DC: IEEE Computer Society.
- Besrou, S., AB Rahim, L. B., & Dominic, P. (2016). A Quantitative Study to Identify Critical Requirement Engineering Challenges in the Context of Small and Medium Software Enterprise. *2016 3rd International Conference On Computer And Information Sciences (ICCOINS)*. Kuala Lumpur.

- Besrou, S., AB Rahim, L. B., & Dominic, P. (2016). Investigating Requirement Engineering Techniques in The Context of Small and Medium Software Enterprises. *2016 3rd International Conference On Computer And Information Sciences (ICCOINS)*. Kuala Lumpur.
- Boehm, B. W. (1989). *Software Risk Managment*. IEEE Computer Society Press.
- Bourque, P., & Fairley, R. E. (2014). *Guide to Software Engineering Body of knowledge (SWEBOK(R)): Version 3.0*. Los Alamitos: IEEE Computer Society Press.
- Bourque, P., Dupuis, R., & Abran , A. (1999). *The Guide to the Software Engineering Body of Knowledge*. IEEE.
- Christou, I., Ponis, S., & Palaiologou, E. (2010). Using the Agile Unified Process in Banking. *IEEE Software*, 72-79.
- Condori-Fernandez, N., Daneva, M., Sikkel, K., Wieringa, R., Dieste, O., & Pastor, O. (2009). A systematic mapping study on empirical evaluation of software requirements specifications techniques. *3rd International Symposium on Empirical Software Engineering and Measurement* (págs. 502-505). Washington DC: IEEE Computer Society.
- Cubillos Rodríguez, M. C., & Rozo Rodríguez, D. (2009). El concepto de calidad: Historia, evolución e importancia para la competitividad. *Revista De La Universidad De La Salle*, 80-99.
- Curcio , K., Navarro , T., Malucelli, A., & Reinehr , S. (2018). Requirements engineering: A systematic mapping study in agile software development. *The Journal of Systems and Software*, 32-50.

- Daneva, M., Damian, D., Marchetto, A., & Pastor, O. (2014). Empirical Research Methodologies and Studies in Requirements Engineering: How far did we come? *Journal of Systems and Software*, 1-9.
- Daneva, M., van der Veen, E., Amrit, C., Ghaisas, S., Sikkel, K., Kumar, R., . . . Wieringa, R. (2013). Agile requirements prioritization in large-scale outsourced system projects: An empirical study. *The journal of Systems and Software*, 1333-1353.
- Dermeval, D., Vilela, J., Bittencourt, I., Castro, J., Isotani, S., & Brito, P. (2014). A Systematic Review on the Use of Ontologies in Requirements Engineering. *2014 Brazilian Symposium on Software Engineering*. Maceio.
- Fernández Méndez, D., & Wagner, S. (2014). Naming the pain in requirements engineering: A design for a global family of surveys and first results from Germany. *Information and Software Technology*, 57, 616-643.
- Fernández Sanz, L., & Bernad Silva, P. (2014). Risk management in software development projects in Spain: a state of art. *Facultad de Ingeniería Universidad de Antioquia*(70), 233-243.
- Fitzgerald, C., Letier, E., & Finkelstein, A. (2011). Early Failure Prediction in Feature Request Managment Systems. *2011 IEEE 19th International Requirements Engineering Conference*, (págs. 229-238). Trento.
- Fotrousi, F., Fricker, S. A., & Fiedler, M. (2014). Quality requirements elicitation based on inquiry of quality-impact relationships. *2014 IEEE 22nd International Requirements Engineering Conference (RE)*. Karlskrona.

- Frese, R., & Sauter, V. (27 de Noviembre de 2014). Improving Your Odds for Software Project Success. *IEEE Engineering Management Review*, págs. 125-131.
- G. Martins, L. E., & Gorscheck, T. (2016). Requirements engineering for safety-critical systems: A systematic literature review. *Information and Software Technology*, 71-89.
- Gambo , I. P., Soriyan, A. H., & Ikono, R. N. (2015). A Proposed Process Model for Requirements Engineering using Delphi Techniques for Prioritisation. *Information Technology and Computer Science*, 73-80.
- Gary, K., Enquibahrie, A., Ibanez, L., Cheng, P., Yaniv , Z., Cleary, K., . . . Heidenreich, J. (2011). Agile Methods for Open Source Safety-Critical Software. *Software: Practice and Experience*, 945-962.
- Genero Bocco, M., Cruz Lemus, J. A., & Piattini Velthuis, M. G. (2014). *Métodos de Investigación en ingeniería del software*. Madrid: Ra-Ma.
- Grewal, H., & Maurer, F. (2007). Scaling Agile Methodologies for Developing a Production Accounting System for the Oil & Gas Industry. *Agile 2007* (págs. 309-315). Washington DC: .
- Group, S. (2015). *Chaos Report*.
- Hatcliff, J., Wassyn, A., Kelly, T., Comar, C., & Jones, P. (2014). Certifiably safe software-dependent systems: challenges and directions. *FOSE 2014 Proceedings of the on Future of Software Engineering* (págs. 182-200). Hyderabad: ACM.

- Heikkila, V., Lassenius, C., Damian , D., & Paasivara, M. (2015). A Mapping Study on Requirements Engineering in Agile Software Development. *2015 41st Euromicro Conference on Software Engineering and Advanced Applications*. Funchal.
- Heimdal , M. P. (2007). Safety and Software Intensive Systems: Challenges Old and New. *Future of Software Engineering (FOSE '07)*. Minneapolis: IEEE.
- Huang, F., & Sun, T. (2013). Software Project Metrics and Quality Managment. *2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing* (págs. 615-618). Beijing: IEEE.
- IEEE. (2014). IEEE Standar for Sofware Quality Assurance Processes. New York: IEEE.
- Ilyas, F., & Zahra, K. (2016). A Survey on Current Requirement Process Practices in Software Companies and Requirement Process Problems. *International Conference on Computational Science and Computational Intelligence*. LasVegas.
- Inayat, I., Salwah Salim, S., Marczak, S., Daneva, M., & Shamshirband, S. (2015). A systematic literature review on agile requirements engineering. *Computers in Human Behavior*, 915-929.
- Ishikawa, K. (1991). *Guide to Quality Control*. Asian Productivity Organization.
- ISO. (2007). *ISO 25000*.
- Jalali, S., & Wohlin, C. (2012). Systematic Literature Studies: Database Searches vs. Bacward Snowbaling. *Proceedings of the 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. Lund .

Juran, J. M. (1990). *Juran y la planificación de la calidad*. Madrid: Díaz de Santos.

Kalinowski, M., Curty, P., Paes, A., Ferreira, A., Spinola, R., Méndez Fernández Daniel, . . .

Wagner, S. (2017). Supporting defect causal analysis in practice with cross-company data on causes of requirements engineering problems. *Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*,. Buenos Aires.

Kalinowski, M., Oliveira Spínola, R., Conte, T., Prikladnicki, R., Méndez Fernández, D., &

Wagner, S. (2015). Towards Building Knowledge on Causes of Critical Requirements Engineering Problems. *27th International Conference on Software Engineering and Knowledge Engineering (SEKE 2015)*. Pittsburgh.

Kalinowski, M., Spínola, R., Conte , T., Prikladnicki, R., Méndez Fernández, D., & Wagner, S.

(2015). Towards Building Knowledge on Causes of Critical Requirements Engineering Problems. *International Conference on Software Engineering and Knowledge Engineering*. Pittsburgh.

Kendall , R. P., Mark, A., Squires, S. E., & Halverson, C. A. (2010). Condor: Case Study of a

Large-Scale, Physics-Based Code Development Project. *Computing in Science & Engineering* , 22-27.

Khankaew, S., & Riddle, S. (2014 IEEE 4th International Workshop on Empirical Requirements

Engineering (EmpiRE)). A review of practice and problems in requirements engineering in small and medium software enterprises in Thailand. *Empirical Requirements Engineering*. Karlskrona.

- Kitchenham, B. A., & Charters, S. M. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Durham: EBSE .
- Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O. P., Turner , M., Niazi, M., & Linkman, S. (2010). Systematic literature reviews in software engineering – A tertiary study. *Information and Software Technology*, 792-805.
- Kpodjedo, S., Ricca, F., Galinier, P., Guéhuéneuc, Y.-G., & Antoniol, G. (2011). Desing Evaluation Metrics for Defect Prediction in Objet Oriented Systems. *Empirical Software Engineering*(10664), 141-175.
- Kumari S, N., & S Pillai, A. (2013 ). A study on the software requirements elicitation issues - its causes and effects. *2013 Third World Congress on Information and Communication Technologies (WICT 2013)*. Hanoi.
- Laporte, C. Y., & April, A. (2018). *Software Quality Requirements*. Wiley-IEEE Press.
- Lewis, W. E. (2004). *Software Testing and Continuos Quality Improvement*. Boca Raton: Auerbach Publications.
- Liebel, G., Tichy, M., Knauss, E., Ljungkrantz, O., & Stieglbauer, G. (2018). Organisation and communication problems in automotive requirements engineering. *Requirements Engineering*, 23, 145-167.
- Lloyd, W. J. (11 de Julio de 2001). *Tools and Techniques for Effective Distributed Requirements Engineering: An Empirical Study*. Blacksburg, Virginia, Estados Unidos.

- Lloyd, W. J., Rosson, M. B., & Arthur, J. D. (2002). Effectiveness of elicitation techniques in distributed requirements engineering. *Proceedings IEEE Joint International Conference on Requirements Engineering*. Essen: IEEE.
- Londoño, L. F., Anaya , R., & Tabares, M. S. (2008). ANÁLISIS DE LA INGENIERÍA DE REQUISITOS ORIENTADA POR ASPECTOS SEGÚN LA INDUSTRIA DEL SOFTWARE. *EIA*, 43-52.
- Lunenburg, F. C. (2011). Decision Making in Organizations. *International Journal of Managment, Business, and Administration*, 1-9.
- Mafra, P., Kalinowski, M., Méndez Fernández, D., Felderer, M., & Wagner, S. (2016). Towards Guidelines for Preventing Critical Requirements Engineering Problems. *2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. Limasol.
- Martínez Carazo, P. (2006). Estrategia metodológica de la investigación científica. *Pensamiento & Gestión* .
- McCall, J., Richards, P., & Walters , G. (1977). Factors in Software Quality.
- Méndez Fernández, D., Tießler, M., Kalinowski, M., Felderer, M., & Kuhrmann, M. (2017). On Evidence-based Risk Management in Requirements Engineering. *Software Quality: Methods and Tools for Better Software and Systems. SWQD 2018*. Vienna.
- Méndez Fernández, D., Wagner, S., Kalinowski, M., Felderer, M., Mafra, P., Vetrò, A., . . . Wieringa, R. (2017). Naming the pain in requirements engineering Contemporary problems, causes, and effects in practice. *Empir Software Eng*, 22, 2298-2338.

- Nadir, S., Streitferdt, D., & Burggraf, C. (2016). Industrial software developments effort estimation. *2016 International Conference on Computational Science and Computational Intelligence*. Las Vegas.
- Nagappan, N., Ball, T., & Zeller, A. (2006). Mining Metrics to Predict Component Failures. *28th International Conference On Software Engineering*. Shanghai.
- Nagappan, N., Murphy, B., & Basili, V. (2008). The Influence of Organizational Structure on Software Quality: An Empirical Case Study. *30th International Conference on Software Engineering*. Leipzig.
- Nakai, H., Tsuda, N., Honda, K., Washizaki, H., & Fukazawa, Y. (2016). A SQuaRE-based software quality evaluation framework and its case study. *TENCON*. Singapore: IEEE.
- Nicolás, J., Carrillo de Gea, J. M., Nicolás, B., Fernández Aleman, J. L., & Toval, A. (2018). On the Risks and Safeguards for Requirements Engineering in Global Software Development: Systematic Literature Review and Quantitative Assessment . *IEEE Access*, 59628 - 59656.
- Nolan, A., Abrahão, S., & Clements, P. (2011). Managing requirements uncertainty in engine control systems development. *Requirements Engineering Conference*. Trento.
- Olaperi, Y. S., & Sanjay, M. (2015). An Empirical Evaluation of Software Quality Assurance Practices and Challenges in a Developing Country. *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*. Liverpool.

- Oliveros, A., Napolillo, F., & López Infesta, F. (2016). Requirements in Web applications development. *IEEE CACIDI 2016 - IEEE Conference on Computer Sciences*. Buenos Aires Argentina.
- Ouhbi, S., Idri, A., Fernandez-Aleman, J. L., & Toval, A. (2013). Software Quality Requirements: A Systematic Mapping Study. *2013 20th Asia-Pacific Software Engineering Conference (APSEC)*. Bangkok.
- Pacheco , C., García , I., & Reyes, M. (2018). Requirements elicitation techniques: a systematic literature review based on the maturity of the techniques. *The Institution of Engineering and Technology* , 365-378 .
- Pandey, D., & Surman, U. (2010). An Effective Requirement Engineering Process Model for Software Development and Requirements Management . *International Conference on Advances in Recent Technologies in Communication and Computing* (págs. 287-291). Kottayam: IEEE.
- Pandey, D., Surman, U., & Ramani, A. K. (2010). An Effective Requirement Engineering Process Model for Software Development and Requirements Management. *2010 International Conference on Advances in Recent Technologies in Communication and Computing* (págs. 287-291). Kottayam: IEEE.
- Peffer, K., Tuunanen, T., Rothernberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 45-77.

- Petersen, K., Feldt, R., Mujtaba, M., & Mattsson, M. (2008). Systematic mapping studies in software engineering. *12th International Conference on Evaluation and Assessment in Software Engineering*. (págs. 71-80). Bari: Blekinge Institute of Technology.
- Pohl, K. (2010). *Requirements Engineering: Fundamentals, Principles, and Techniques*. New York: Springer-Verlag Berlin Heidelberg.
- Rahman, A. A., & Nurdatillah, H. (2015). Defect Management Life Cycle Process for Software Quality Improvement. *3rd International Conference on Artificial Intelligence, Modelling and Simulation AIMS*. Kota Kinabalu.
- Reeves, C. A., & Bednar, D. A. (1994). DEFINING QUALITY: ALTERNATIVES AND IMPLICATIONS. *Academy of Management Review*, 419-445.
- Rempel, P., & Mäder, P. (2016). Preventing Defects: The Impact of Requirements Traceability Completeness on Software Quality. *IEEE Transactions on Software Engineering*.
- Sawyer, P., Sommerville, I., & Viller, S. (1997). Requirements Process Improvement through The Phased Introduction of Good Practice. *Software Process e. Software Process Improvement and Practice*, 19-34.
- Sevilla, G. A., Collazos, C. A., Zapata, S. G., & Giraldo, F. D. (2016). Modelo de Procesos de Ingeniería de Requisitos de Software Distribuida. *CONAIIISI 2016*. At Salta.
- Sevilla, G. A., Zapata, S. G., Collazos, C. A., & Giraldo, F. D. (2016). Modelo de Proceso de Ingeniería de Requisitos de Software Distribuida. *CONAIIISI*. At Salta.

- Sevilla, G., Zapata, S., Torres, E., & Collazo, C. (2014). Using Wikis as collaborative strategy to support software requirements elicitation. *2014 9th Computing Colombian Conference (9CCC)* (págs. 54-61). Pereira: IEEE.
- Solemon, B., Sahibuddin, S., & Abd, A. A. (2008). Requirements Engineering Problems in 63 Software Companies in Malaysia. *2008 International Symposium on Information Technology*. Kuala Lumpur.
- Spoletini, P., & Ferrari, A. (2017). Requirements Elicitation: A Look at the future through the lenses of the past. *IEEE 25th International Requirements Engineering Conference (RE)* (págs. 476-477). Lisboa: IEEE.
- Sukoco, A., Marzuki, & Cucus, A. (2012). Concept of quality measurement system software based on standard ISO 9126 and ISO 19011. *2012 2nd International Conference on Uncertainty Reasoning and Knowledge Engineering* (págs. 105-108). Jalarta: IEEE.
- Sun, Y., & Shinde, K. (2016). Template-Based Code Generation Framework for Data-Driven Software Development. *2016 4th Intl Conf on Applied Computing and Information Technology/3rd Intl Conf on Computational Science/Intelligence and Applied Informatics/1st Intl Conf on Big Data, Cloud Computing, Data Science & Engineering (ACIT-CSII-BCD)*. Las Vegas.
- Suranto, B. (2015). Software Prototypes: Enhancing The Quality of Requirements Engineering Process. *2015 International Symposium on Technology Managment and Emerging Technologies (ISTMET)*. Langkawi.

- Taiseera Hazeem , A. B., Falcone Sampaio, P. R., Dabhi, D., & Loucopoulos , P. (2006). Performing Requirements Elicitation Activities Supported by Quality Ontologies. *18th International Conference on Software Engineering and Knowledge Engineering, SEKE 2006*, (págs. 343-348). San Francisco .
- Tonella, P., Torchiano, M., Du Bois, B., & Systä, T. (2007). Empirical studies in reverse engineering: state of the art and future trends. *Empirical Software Engineering*, 551-571.
- Vaishnavi, V., Kuechler, B., & Petter, S. (2004). *Design Science Research in Information Systems*.
- Winter, R. (2008). Design Science research in Europe. *European Journal of Information Systems*, 470-475.
- Zhang, H., Babar, M. A., & Tell, P. (2011). Identifying Relevant Studies in Software Engineering. *Information and Software Technology*, 625-637.