



UNIVERSIDAD DEL AZUAY

Facultad de Ciencias de la Administración

Escuela de Ingeniería de Sistemas y Telemática

Indagación Exploratoria para la Actualización del Estado del Arte sobre
Modelos de Calidad de Software

Trabajo de titulación previo a la obtención de Título de Ingeniero en
Sistemas y Telemática

Autor: Ana Victoria Villalta López

Director: Ing. Juan Pablo Carvallo Vega, PhD

Cuenca, Ecuador

2015

DEDICATORIA

A mis padres, Miguel y Anita, por ser la muestra más grande del amor de Dios en mi vida.

A mi amigo y director de tesis, Juan Pablo Carvallo Vega, por el tiempo, las oportunidades y las palabras de aliento.

AGRADECIMIENTOS

A Jehová mi pastor.

“Tú eres el camino, la verdad y la vida”

ÍNDICE GENERAL

DEDICATORIA	II
AGRADECIMIENTOS	III
ÍNDICE GENERAL.....	IV
ÍNDICE DE TABLAS	VII
ÍNDICE DE ILUSTRACIONES.....	VIII
RESUMEN EJECUTIVO	IX
ABSTRACT	X
INTRODUCCION	1
CAPÍTULO 1. INTRODUCCIÓN.....	1
1.1 Antecedentes	0
1.2 Justificación	0
1.3 Objetivos	1
1.4 Metodología	1
1.5 Estructura general del documento.....	2
CAPÍTULO 2. MÉTODO DE TRABAJO	1
2.1 Introducción	2
2.2 Calidad de software.....	3
2.2.1 Tipos de modelos de calidad de software	3
2.2.2 Estándares de modelos de calidad	4
2.2.3 Propiedades de los modelos de calidad.....	6
2.2.4 Relaciones entre factores de calidad.....	7
2.2.5 Relación de las métricas con los factores de calidad.....	9
2.3 Validación empírica en Ingeniería de Software	9

2.4	Revisión sistemática de la literatura	11
2.4.1	Guías para la realización de SLR en Ingeniería de software	13
2.5	Proceso de la Revisión	15
2.5.1	Planificación	15
2.5.2	Desarrollo de la Revisión.....	18
2.5.3	Reporte de la Revisión.....	21
CAPÍTULO 3. ESTUDIO - ESTADO DEL TRABAJO DE TITULACIÓN DEL AÑO 2007 ...		22
3.1	Introducción	23
3.2	Recopilación de técnicas	24
3.3	Análisis de artefactos utilizados	36
CAPÍTULO 4: PLANEACIÓ DE LA REVISIÓN.....		39
4.1	Introducción	40
4.2	Campo a estudiar	41
4.3	Formulación del problema	41
4.4	Preguntas de investigación	41
4.5	Protocolo	43
4.5.1	Antecedentes	43
4.5.2	Preguntas de Investigación.....	43
4.5.3	Estrategia de Búsqueda	44
4.5.4	Criterios de Selección de estudios.....	46
4.5.5	Procedimientos de Selección de estudios.....	47
4.5.6	Listas de verificación y procedimientos de evaluación.....	48
4.5.7	Estrategia de extracción de datos.	49
4.5.8	Síntesis de los datos extraídos.	50
4.5.9	Estrategia de difusión.....	50
CAPÍTULO 5. PUESTA EN MARCHA		51
5.1	Introducción	52
5.2	Generación de una estrategia de búsqueda	53
5.2.1	Prueba Piloto (Primera Etapa).....	55
5.2.2	Segunda Etapa (Refinamiento 1)	58
5.2.3	Tercera Etapa (Refinamiento 2)	69
5.3	Selección de los estudios	80
5.4	Evaluación de calidad del estudio	84
5.4.1	Proporcionar criterios aún más detallados de selección.....	84

5.4.2	Ponderación de la importancia de los estudios individuales.....	85
5.4.3	Orientar a las recomendaciones para futuras investigaciones.....	85
5.5	Extracción de estudios.....	89
5.6	Síntesis de los datos.....	96
CAPITULO 6. ANÁLISIS DE LA REVISIÓN		99
6.1	Introducción	100
6.2	Análisis de la SLR	101
6.3	Otros Datos.....	109
CAPITULO 7: CONCLUSIONES		111
7.1	Conclusiones.....	112
7.2	Lecciones Aprendidas	113
7.3	Trabajos Futuros.....	118
BIBLIOGRAFÍA.....		119

ÍNDICE DE TABLAS

TABLA 2.1	INFORMACIÓN GENERAL DE LAS CARACTERÍSTICAS DE LA METODOLOGÍA DE INVESTIGACIÓN	10
TABLA 2.2	COMPARACIÓN DE GUÍAS PARA REALIZAR SLR	13
TABLA 2.3	CONTRIBUCIÓN A LAS GUÍAS PARA ELABORACIÓN DE PROTOCOLOS DE SLR INCLUIDAS EN [K07] INCLUIDAS EN [CHHK13].....	14
TABLA 2.4	CADENAS UTILIZADAS EN EL REFINAMIENTO 2 SPRINGER. ¡ERROR! MARCADOR NO DEFINIDO.	
TABLA 2.5	CADENAS UTILIZADAS EN EL REFINAMIENTO 2 ACM.....	74
TABLA 2.6	CADENAS UTILIZADAS EN EL REFINAMIENTO 3 ACM..... ¡ERROR! MARCADOR NO DEFINIDO.	
TABLA 3.1	FORMATO COMÚN USADO EN TODAS LAS PROPUESTAS	25
TABLA 3.2	EJEMPLO DEL FORMATO COMÚN UTILIZADO PARA LA UBICACIÓN DE FACTORES DE CALIDAD DE LOS ARTÍCULOS INDIVIDUALES LEÍDOS POR LOS AUTORES UTILIZANDO EL TRABAJO [GD05]	29
TABLA 3.3	EJEMPLO DEL USO DE LA REGLA DE IDENTIFICACIÓN.....	31
TABLA 3.4	EJEMPLO DE APLICACIÓN DE LA REGLA DRAG-AND-DROP.....	31
TABLA 3.5	EXTRACTO DEL CATÁLOGO RESULTANTE DEL PROCESO DE COMPOSICIÓN DE ATRIBUTOS TÉCNICOS 32	
TABLA 3.6	REGLAS DE COMPOSICIÓN APLICADAS ENTRE EL MODELO [ISO12119] Y [PVL97]	35
TABLA 3.7	FORMATO COMÚN PARA LOS ARTÍCULOS INDIVIDUALES LEÍDOS.....	36
TABLA 4.1.	PREGUNTAS DE INVESTIGACIÓN Y SUS OBJETIVOS	42
TABLA 4.2.	TÉRMINOS DE BÚSQUEDA A USAR EN LAS DIFERENTES BASES DE DATOS.....	44
TABLA 4.3.	FUENTES DE BÚSQUEDA DIGITALES SELECCIONADAS	45
TABLA 4.4.	OTRAS FUENTES (REVISTAS) EN LAS CUALES BUSCAR LOS ARTÍCULOS.....	45
TABLA 5.1.	PALABRAS CLAVE IDENTIFICADAS EN RELACION A LAS PREGUNTAS	55
TABLA 5.2.	CADENAS DERIVADAS DE LAS PALABRAS CLAVE DE LA TABLA 5.1	56
TABLA 5.3.	NÚMERO DE ARTÍCULOS ENCONTRADOS POR CADENA DE BÚSQUEDA EN LA PRUEBA PILOTO	57
TABLA 5.4.	TÉRMINOS DE BÚSQUEDA ALTERNATIVOS.....	58
TABLA 5.5.	BASES DE DATOS Y SUS OPERADORES LÓGICOS	59
TABLA 5.6.	CADENAS OBTENIDAS EN LA ETAPA DE REFINAMIENTO 1	63
TABLA 5.7.	NÚMERO DE ARTÍCULOS ENCONTRADOS: PRIMER REFINAMIENTO BASE DE DATOS SPRINGER	66
TABLA 5.8.	NÚMERO DE ARTÍCULOS ENCONTRADOS: PRIMER REFINAMIENTO BASE DE DATOS IEEE	67
TABLA 5.9.	NÚMERO DE ARTÍCULOS ENCONTRADOS: PRIMER REFINAMIENTO BASE DE DATOS ACM.....	68
TABLA 5.10.	CADENAS UTILIZADAS EN EL REFINAMIENTO 2 SPRINGER.....	72
TABLA 5.11.	CADENAS UTILIZADAS EN EL REFINAMIENTO 2 IEEE	73
TABLA 5.12.	CADENAS UTILIZADAS EN EL REFINAMIENTO 2 ACM	74
TABLA 5.13.	CADENAS UTILIZADAS EN EL REFINAMIENTO 3 ACM	74
TABLA 5.14.	NÚMERO DE ARTÍCULOS ENCONTRADOS: SEGUNDO REFINAMIENTO BASE DE DATOS SPRINGER E IEEE	75
TABLA 5.15.	NÚMERO DE ARTÍCULOS ENCONTRADOS: SEGUNDO Y TERCER REFINAMIENTO BASE DE DATOS ACM	76
TABLA 5.16.	DATOS EXTRAÍDOS PARA CADA ARTÍCULO.- SPRINGER.....	81
TABLA 5.17.	DATOS EXTRAÍDOS PARA CADA ARTÍCULO.- IEEE.....	82
TABLA 5.18.	DATOS EXTRAÍDOS PARA CADA ARTÍCULO.- ACM.....	83
TABLA 5.19.	CLASIFICACIÓN DE ARTÍCULOS EN SELECCIONADO, DUDOSO O FALSO.- SPRINGER.....	86
TABLA 5.20.	CLASIFICACIÓN DE ARTÍCULOS EN SELECCIONADO, DUDOSO O FALSO.- IEEE.....	87
TABLA 5.21.	CLASIFICACIÓN DE ARTÍCULOS EN SELECCIONADO, DUDOSO O FALSO.- ACM.....	88
TABLA 5.22.	COLORES DE LOS ARTÍCULOS POR BASE DE DATO	92
TABLA 5.23.	ARTÍCULOS CLASIFICADOS ACORDE A LAS PREGUNTAS DE INVESTIGACIÓN Y POR BASE DE DATOS	93

TABLA 5.24.	ARTÍCULOS A LEER, CLASIFICADOS POR DOS INVESTIGADORES	95
TABLA 5.25.	TABLA FORMATO PARA RESUMIR LOS ARTÍCULOS INDIVIDUALES [MTK09]	97
TABLA 5.26.	TABLA RESUMEN QUE SINTETIZA LOS DATOS DE LOS ARTÍCULOS INDIVIDUALES	98

ÍNDICE DE ILUSTRACIONES

FIGURA 2.1	PROPIEDADES ESTRUCTURALES DE LOS MODELOS DE CALIDAD DE SOFTWARE	8
FIGURA 2.2	INFORMACIÓN GENERAL DE LAS CARACTERÍSTICAS DE LA METODOLOGÍA DE INVESTIGACIÓN.....	12
FIGURA 2.3	62
FIGURA 5.1	PROCESO DE BÚSQUEDA DE ARTÍCULOS EN IEEE	54
FIGURA 5.2	OPCIONES PARA REALIZAR REFINAMIENTOS AUTOMÁTICOS EN SPRINGER.....	60
FIGURA 5.3	OPCIONES DE REFINAMIENTOS AUTOMÁTICOS EN IEEE.....	61
FIGURA 5.4	OPCIONES DE BUSQUEDA AVANZADA DE ACM.....	62
FIGURA 5.5	RESUMEN DEL PROCESO DE REFINAMIENTO PARA LA PALABRA “ELEMENTS” - IEEE.....	77
FIGURA 5.6	INCLUSIÓN DE LOS PROCESO DE “SELECCIÓN” Y “EVALUACIÓN” DE ARTÍCULOS A LA FIGURA 5.1	79
FIGURA 5.7	INCLUSIÓN DEL PROCESO DE “EXTRACCIÓN DE DATOS” A LA FIGURA 5.2	90
FIGURA 6.1	NÚMERO DE ATRIBUTOS POR NIVEL	102
FIGURA 6.2	NUMERO DE TIPOS DE ATRIBUTOS ENCONTRADOS	102
FIGURA 6.3	NOMBRE DADO AL NIVEL 1 POR ARTÍCULO	103
FIGURA 6.4	NOMBRE DADO AL NIVEL 2 POR ARTÍCULO	103
FIGURA 6.5	NOMBRE DADO AL NIVEL 3 POR ARTÍCULO	104
FIGURA 6.6	NOMBRE DADO AL NIVEL 4 POR ARTÍCULO.....	104
FIGURA 6.7	NOMBRE DADO AL NIVEL 5 POR ARTÍCULO.....	105
FIGURA 6.8	TIPOS DE RELACIONES ENTRE LOS DISTINTOS ELEMENTOS.....	106
FIGURA 6.9	TIPOS DE ATRIBUTOS CON PROPIEDAD DE SOLAPAMIENTO.....	106
FIGURA 6.10	APLICACIONES PARA LAS QUE SE HAN DESCRITO LOS MODELOS DE CALIDAD DE SOFTWARE .	107
FIGURA 6.11	ARTÍCULOS QUE DESCRIBEN UN MÉTODO DE CONSTRUCCIÓN	108
FIGURA 6.12	TIPO DE CALIDAD QUE EVALÚAN LOS FACTORES DE CALIDAD.....	108
FIGURA 6.13	PORCENTAJE DE INVESTIGACIÓN POR AÑOS.....	109
FIGURA 6.14	PORCENTAJE DE ARTÍCULOS QUE CONTIENEN MÉTRICAS DE CALIDAD DE SOFTWARE.....	110
FIGURA 7.1	RESULTADO DE LA BÚSQUEDA APLICANDO UNA SOLA CADENA EN IEEE.....	115
FIGURA 7.2	BÚSQUEDA EN ACM DIGITAL LIBRARY	116
FIGURA 7.3	BÚSQUEDA EN IEEE UTILIZANDO LA MISMA CADENA QUE EN ACM DIGITAL LIBRARY.....	117
FIGURA 7.4	BÚSQUEDA EN IEEE.....	117

RESUMEN EJECUTIVO

La valoración de la calidad de un producto de software juega un papel fundamental tanto en la evaluación y selección de un componente de software (ej., un sistema legado o un componente a ser adquirido, ya sea de índole comercial, COTS; o libre, FOSS), como al evaluar la calidad de un producto resultante de un proceso de construcción a la medida (sea este construido por una unidad interna de la organización o por una empresa externa subcontratada para este propósito). Los Modelos de Calidad del Software o *Quality Models* (QM) [ISO 8402] son artefactos específicamente diseñados y construidos para este propósito.

El presente trabajo pretende realizar una revisión sistemática de la literatura o *Systematic Literature Review* (SLR), con el objeto de identificar los estudios relevantes que reúnan los últimos aportes en cuanto a QMs, considerando sus elementos estructurales, propiedades, métodos de construcción y aplicaciones para las cuales han sido utilizados.

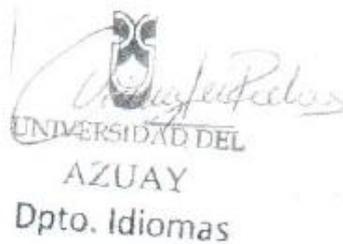
Se revisaron tres bases de datos digitales y tres revistas que presentaban las publicaciones entre el 2007 y 2014. A partir de una población de 49.935 documentos de la base de datos del IEEE, se identificaron 20 documentos relevantes que proporcionan evidencia empírica sobre el tema. Los resultados se han tabulado en tablas específicamente diseñadas para analizar los elementos requeridos y responder a las preguntas propuestas para esta investigación. Se destacó el aporte significativo que, en materia de métricas de calidad, ha existido en estos últimos años.

ABSTRACT

ABSTRACT

Evaluating the quality of a software product plays a key role in the assessment and selection of a software component, as well as in assessing the quality of a product that results from a made-to measure construction process. Quality Models (QMs) are devices specifically designed for this purpose.

This paper aims to perform a Systematic Literature Review (SLR) in order to identify the studies that include the latest contributions in regard to QMs, considering its structural elements, properties, construction methods and applications for which they have been used.



Translated by,
Lic. Lourdes Crespo

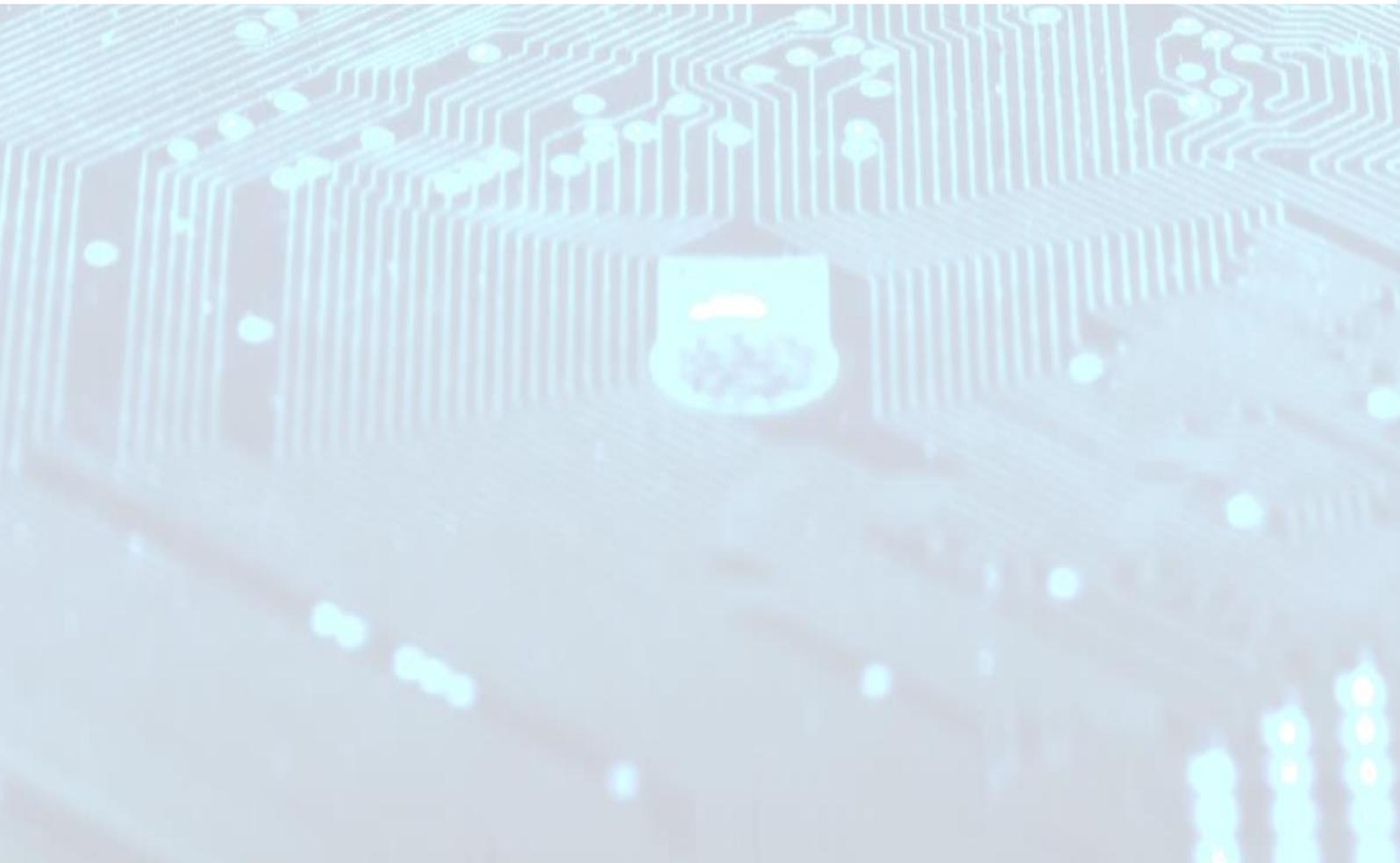
INTRODUCCION

Considerando a la Calidad de Software un área de la Ingeniería, que requiere su comprensión para la construcción de un producto software que cumpla con los requerimientos del usuario final, se realizó un estado del arte actualizado que reúne los modelos de calidad que los investigadores han incorporado como aporte a la ciencia en los últimos años. Para esto se exploró la literatura y se observó que existe un gran número de artículos que abordan el tema de modelos de calidad del software. Estos se encuentran dispersos en diversas bases de datos bibliográficas. Aunque su estudio podría aportar información relevante para la construcción del estado del arte en esta materia, la selección y clasificación de artículos, así como la extracción y tabulación de información demandan una importante cantidad de tiempo y recursos.

Debido a ello y como contribución, tanto a la academia como a la industria, se realizó una indagación en los avances reportados en esta materia usando la técnica de SLR. Los resultados del proceso se tabularon en cuadros específicamente diseñadas para categorizar aspectos importantes del estudio que incluyeron elementos, propiedades, métodos de construcción y aplicaciones de los Modelos de Calidad del Software (MC). El protocolo de estudio se describió detalladamente con el objeto de que sea replicable en otros ámbitos de la ingeniería de software.



CAPITULO 1. INTRODUCCIÓN



1.1 Antecedentes

En el año 2007, como aporte a la investigación, se realizó un sondeo de artículos selectos en relación a los Atributos de Calidad utilizados para la Evaluación de Sistemas Basados en Componentes [DO07]. La revisión de la literatura base se llevó a cabo sin una técnica de investigación formal, sin embargo, como resultado se obtuvo un catálogo de atributos de calidad, técnicos y no - técnicos, estructurados a modo de un modelo de calidad del software genérico. A más de las relaciones jerárquicas existentes entre los atributos identificados, el resultado permitió identificar casos de sinonimia existentes entre ellos.

Este trabajo propone extender este estudio, incorporando una técnica formal de investigación empírica, que permite identificar artículos relevantes y actualizar el estado del arte sobre modelos de calidad de software desde el año 2007 al 2014.

1.2 Justificación

Al cursar la materia de Calidad de Software, en la carrera de Ingeniería de Sistemas, se comprende la importancia que tiene construir productos software que satisfagan los requerimientos y estándares de calidad impuestos por la industria. Para lograrlo se debe entender a los modelos de calidad de software como artefactos diseñados para medir, de manera objetiva, la calidad del producto resultante del proceso de ingeniería.

Existe un gran número de artículos que abordan el tema de modelos de calidad del software; su estudio podría aportar información para la construcción del estado del arte en esta materia, pero su selección y clasificación, así como la extracción y tabulación de información, demandan gran cantidad de tiempo y recursos; por esta razón, utilizando la técnica SLR, se planteó construir un reporte que engloba atributos descritos en artículos seleccionados, siguiendo un protocolo riguroso de análisis.

1.3 Objetivos

Objetivo general

Realizar una indagación exploratoria del estado del arte sobre modelos de calidad de software en el periodo comprendido entre los años 2007 y 2014.

Objetivos específicos

- 1) Investigar los fundamentos teóricos y revisar los artefactos construidos en torno al estado del arte, sobre los modelos de calidad de software, que fueron realizados en el año 2007, e identificar posibles elementos a ser reutilizados en este estudio.
- 2) Actualizar el estado del arte utilizando la técnica de SLR, que requiere formar preguntas de investigación y planificar la aplicación de la técnica por medio de la redacción de un protocolo que incorpore estas preguntas durante todo el proceso.
- 3) Desarrollar el protocolo y reportar los resultados.
- 4) Analizar, organizar y distribuir los artículos en tablas diseñadas para su síntesis.

1.4 Metodología

La técnica de SLR analiza artículos científicos por medio de estrategias que permiten limitar los sesgos en el proceso de recopilación, valoración crítica y síntesis de los estudios relevantes sobre un tema [Good96]. De esta manera se actualiza la información distribuyéndola en grupos y haciendo más fácil su búsqueda al categorizarla por temas. Debido a su efectividad, es actualmente utilizada en todas las áreas del conocimiento que requieran una evaluación de artículos científicos.

Con el aporte de este trabajo se pretende que los estudiantes y profesionales del área de Ingeniería de Sistemas dispongan de un documento resumen del estado del arte en relación a los modelos de calidad del software. Adicionalmente, el documento sirve como guía para

estudiantes, profesores e investigadores que a futuro pretendan realizar un SLR de manera formal y rigurosa.

1.5 Estructura general del documento

Esta tesis está estructurada por un capítulo introductorio además de seis capítulos adicionales.

El segundo capítulo provee los conceptos básicos a cerca de los Modelos de Calidad de Software, Investigación Empírica y la técnica de Revisión Sistemática de la Literatura.

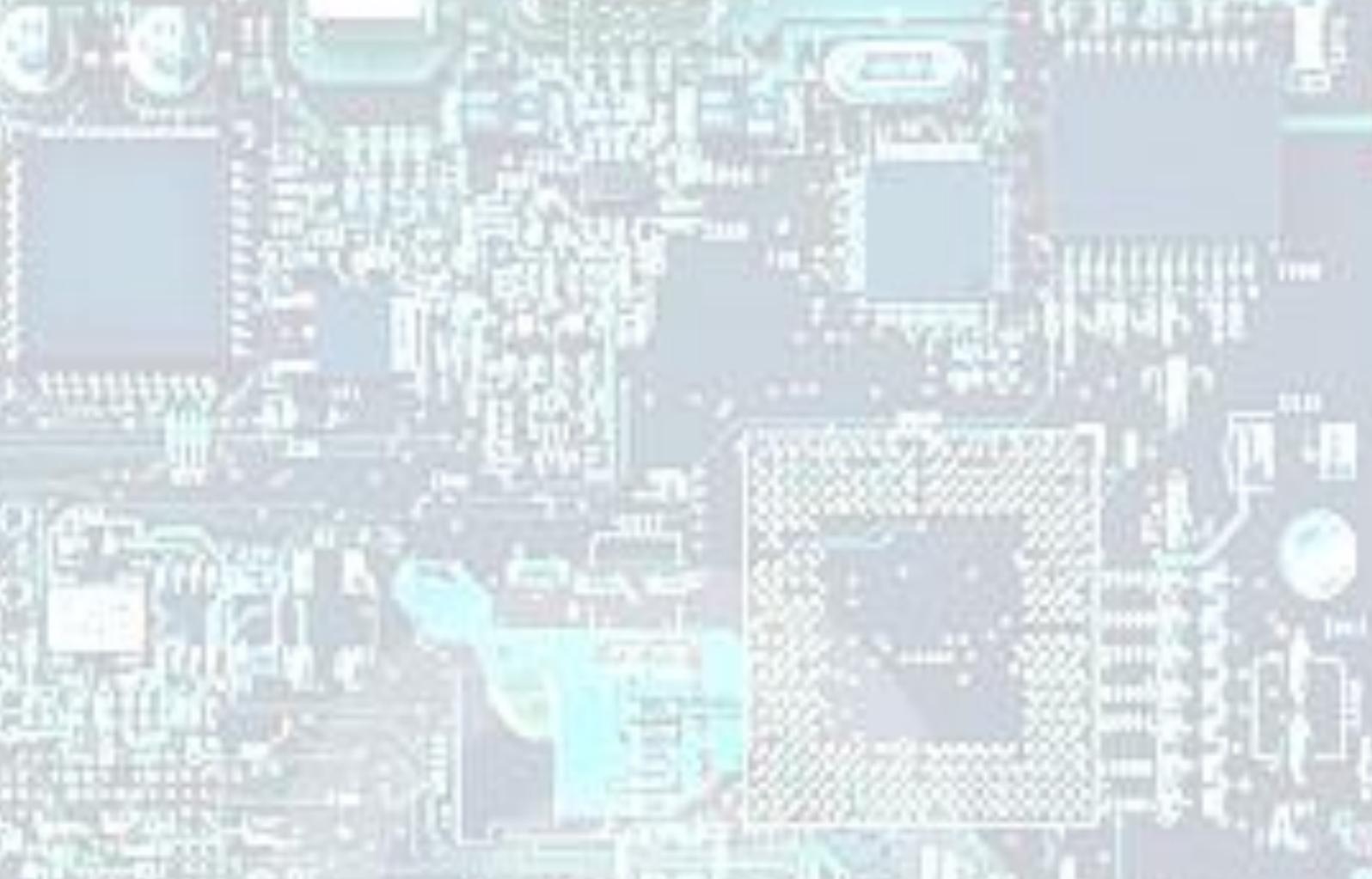
En el tercer capítulo se presenta un resumen de los artefactos construidos en el trabajo del año 2007 y se identifican los elementos a ser reutilizados en este estudio.

El cuarto capítulo detalla la planificación de la revisión sistemática, comienza con la identificación de las preguntas de investigación y la definición del protocolo a ser utilizado en este estudio.

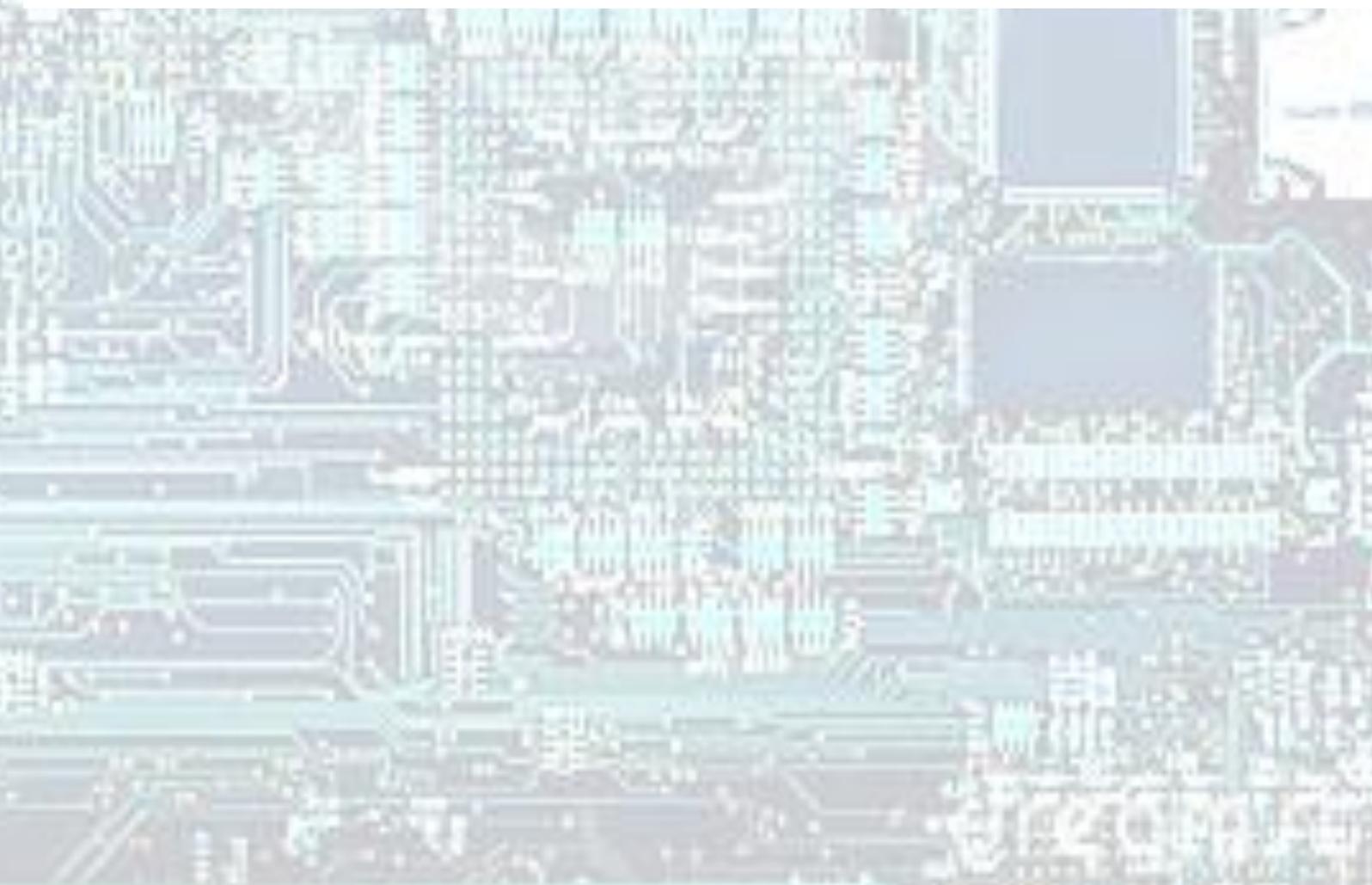
En el quinto capítulo se desarrolla la técnica de SLR; se detallan las estrategias de búsqueda de estudios primarios; el proceso utilizado para su selección y la evaluación de calidad, y la síntesis de la información encontrada.

El sexto capítulo analiza el trabajo realizado, detallando los refinamientos y los pasos seguidos, hasta conseguir los artículos relevantes del estudio.

En el último capítulo se describe las conclusiones, recomendaciones y trabajos futuros acerca de este tema de estudio.



CAPÍTULO 2. FUNDAMENTOS TEORICOS



2.1 Introducción

Tradicionalmente, la investigación en la ingeniería de software ha sido conducida de una manera "semiformal", lo cual ha generado la transferencia de tecnología inmadura a la industria [A14].

La Ingeniería de Software Empírica ha cobrado fuerza en los últimos años [RH08] [K07], la misma ha adoptado técnicas provenientes de otras ramas, como de la psicología o medicina, con el objeto de aportar formalismo a la investigación en este ámbito de la ciencia. La ingeniería de software empírica hace uso de estrategias y métodos empíricos usados en **estudios primarios** – aquellos que permiten levantar evidencia en contacto directo con los actores o fenómenos que la generan; por ejemplo experimentos, casi-experimentos, estudio de casos, estudios *survey*, etnografía, e investigación-acción – o en **estudios secundarios** – aquellos que se basan en la evidencia levantada en los estudios primarios, entre los que se describen revisiones sistemáticas de la literatura y mapeos de estudios.

Los SLR son estudios secundarios que buscan identificar, revisar y sintetizar el conocimiento resultante de estudios primarios que han sido documentados en la literatura, o al menos en una muestra significativa de ella. Una de las aplicaciones de este tipo de estudio es la identificación del estado del arte en un área específica del conocimiento. Es por ello que todo investigador, que necesita establecer una línea base sobre la cual proponer contribuciones, en algún momento de su carrera deberá involucrarse en un SLR.

En este capítulo se presentan los fundamentos teóricos que ayudarán a la comprensión de la investigación y además se ofrecen conceptos básicos acerca los modelos de calidad de software, ya que constituyen el campo de estudio.

2.2 Calidad de software

En [ISO 8402] se define la calidad como “la suma de todos aquellos aspectos o características de un producto o servicio que influyen en su capacidad para satisfacer las necesidades, expresadas o implícitas”.

Según este estándar, un modelo de calidad puede definirse como “el conjunto de factores de calidad, y de relaciones entre ellos, que proporciona una base para la especificación de requisitos de calidad y para la evaluación de la calidad de los componentes software”. Los modelos de calidad se estructuran generalmente como una jerarquía (ya sea un árbol o un grafo dirigido), donde los factores de calidad más genéricos como eficiencia o usabilidad, se descomponen en otros más particulares como tiempo de respuesta o facilidad de aprendizaje, probablemente en diversos niveles de descomposición [CFQ10].

2.2.1 Tipos de modelos de calidad de software

Según [CFQ10] la literatura define tres tipos básicos de modelos de calidad de software:

Fijos.- Existe un catálogo base para la evaluación de calidad, este supone la existencia de todos los factores de calidad posibles y se usará un subconjunto de ellos para evaluar cada proyecto. Este catálogo se estructura en una jerarquía multinivel que se descompone en factores de alto nivel; a su vez, estos se descomponen en criterios, y estos en métricas. Los modelos de calidad fijos son reusables, comparables y rígidos, tales como los modelos de McCall -1997, Boehm -1978, Keller – 1990 o FURPS – 1987.

A la medida.- No existen catálogos de factores de partida; los mismos deben ser creados para el proyecto, se parte de la identificación de los objetivos que constituyen factores más abstractos para luego descomponerlos en factores concretos hasta que estos sean medibles.

Proveen adaptabilidad pero el coste de construcción es muy alto y son desechables.

Existen métodos para crear modelos a la medida tales como *Goal Question Metric* – GQM, 1992 o IEEE 1061, 1998.

Mixtos.- Combinan las ventajas de los dos modelos anteriores, poseen factores abstractos que son reutilizados en varios proyectos y que pueden ser refinados y convertidos en operables para un proyecto en particular. Existen modelos mixtos tales como ADEQUATE, 1999; Gilb,1998 e ISO/IEC 9129-1, 2001.

2.2.2 Estándares de modelos de calidad

Se han propuesto diversos estándares de modelos de calidad de software, entre los que se puede citar los encontrados en [CFQ10]:

IEEE 1061 – (1998).- El objetivo es la definición de métricas de software, propone la construcción de modelos de calidad a la medida. No fija ningún factor, pero sí, una clasificación de los factores que debe tener un modelo desde un nivel más alto y abstracto, hasta llegar a descomponerse en sub factores y estos en métricas.

ISO/IEC 9126.- Propone la construcción de modelos mixtos, ya que parte de una jerarquía de factores clasificados como: “características” (6), “subcaracterísticas” (27) y “atributos”. Este estándar distingue la calidad interna, que evalúa la calidad del software mediante factores medibles durante el desarrollo; la calidad externa, que mide la calidad del software mediante su comportamiento en un sistema del cual forme parte; e introduce el concepto de calidad en uso, que es la calidad del software desde el punto de vista del usuario una vez la versión final del sistema se encuentre instalada y en operación.

Este estándar, en el 2001 fue sustituido por una nueva versión que consta de cuatro partes:

- **9126-1.-** Posee un modelo de calidad común utilizado para medir la calidad interna y externa, y otro para medir la calidad en uso.
- **9126-2.-** Presenta posibles métricas externas para atributos de calidad externos.
- **9126-3.-** Indica posibles métricas para medir atributos de calidad internos.
- **9129-4.-** Señala posibles métricas para evaluar atributos de calidad en uso.

[ISO/IEC 25000] SQUARE.- Estándar creado con el objetivo de profundizar en la especificación de requisitos, y en la evaluación de la calidad del producto software, basado en la ISO/IEC 9126 e ISO/IEC 14598. Este se compone por:

- **ISO/IEC 2500n:** división de gestión de calidad.
 - **ISO/IEC 25000.-** Contiene el modelo de arquitectura SQUARE.
 - **ISO/IEC 25001.-** Establece los requisitos y orientaciones para evaluar y especificar los requisitos del producto software.
- **ISO/IEC 2501n:** división para el modelo de calidad.
 - **ISO/IEC 25010.-** Describe un modelo de calidad para el producto software y para la calidad en uso.
 - **ISO/IEC 25012.-** Define un modelo general para la calidad de los datos.
- **ISO/IEC 2502n:** división para la medición de la calidad.
 - **ISO/IEC 25020.-** Presenta un modelo de referencia común, para los elementos de medición de la calidad, y una guía para que los usuarios seleccionen/desarrollen y apliquen medidas propuestas por normas ISO.

- **ISO/IEC 25021.-** Especifica un conjunto recomendado de métricas base y derivadas, que pueden ser usadas a lo largo del ciclo de vida en el desarrollo del software.
- **ISO/IEC 25022.-** Define métricas para la medición de la calidad en uso del producto.
- **ISO/IEC 25023.-** Define métricas para la medición de la calidad de productos y sistemas software.
- **ISO/IEC 25024.-** Define métricas para la medición de la calidad de datos.

- **ISO/IEC 2503n:** División para los requisitos de calidad.
 - **ISO/IEC 25030.-** Proporciona un conjunto de recomendaciones, para realizar la especificación de los requisitos de calidad del producto software.

- **ISO/IEC 2504n:** División para la evaluación de calidad.
 - **ISO/IEC 25040.-** Propone un modelo de referencia general para la evaluación de calidad.
 - **ISO/IEC 25041.-** Describe los requisitos y recomendaciones para la implementación práctica de la evaluación del producto software, desde el punto de vista de los desarrolladores, adquirentes y evaluadores independientes.
 - **ISO/IEC 25042.-** Define un módulo de evaluación, la documentación, estructura y contenido que se debe utilizar a la hora de determinar uno de estos módulos.
 - **ISO/IEC 25045.-** Establece un módulo para la evaluación de la sub característica "Recuperabilidad"

2.2.3 Propiedades de los modelos de calidad

En [CFQ10], se han descrito las siguientes propiedades para los modelos de calidad del software, (Figura 2.1):

- **Número de Capas.-** Es el nivel de detalle para describir el dominio de software. Mientras más niveles, más descomposición y descripción detallada del componente a evaluar.

- **Tipos de elementos del modelo.-** Se distinguen los elementos de alto nivel, para el propósito de clasificación; y los de bajo nivel, para la descripción detallada y evaluación de características observables de los componentes. Existe una relación entre el número de capas y los tipos de elemento.

- **Propósito del modelo.-** Al construir un modelo es necesario considerar al menos dos dimensiones: la “específico/general” y la “reutilizable/descartable”.
 - **Específico.-** Dispone de modelos a la medida para evaluar un producto software, tiene mayor cantidad de información disponible y su estructura es más compleja.
 - **General.-** Carece de información de un producto software, es menos compleja en su estructura y usualmente posee modelos fijos.
 - **Reutilizable/Descartable.-** Categoriza los modelos de calidad respecto a su nivel de reusabilidad. La reusabilidad de un modelo de calidad se puede reducir al hacerlo más específico y se incrementa al hacerlo más general.

- **Separación entre los Elementos Internos y Externos.-** Los factores externos son los que pueden ser directamente percibidos por los usuarios y afectan a su trabajo; están relacionados usualmente con las características de funcionalidad y usabilidad. Los factores internos se refieren a las características constructivas de los componentes, que son accesibles por los fabricantes.

2.2.4 Relaciones entre factores de calidad.

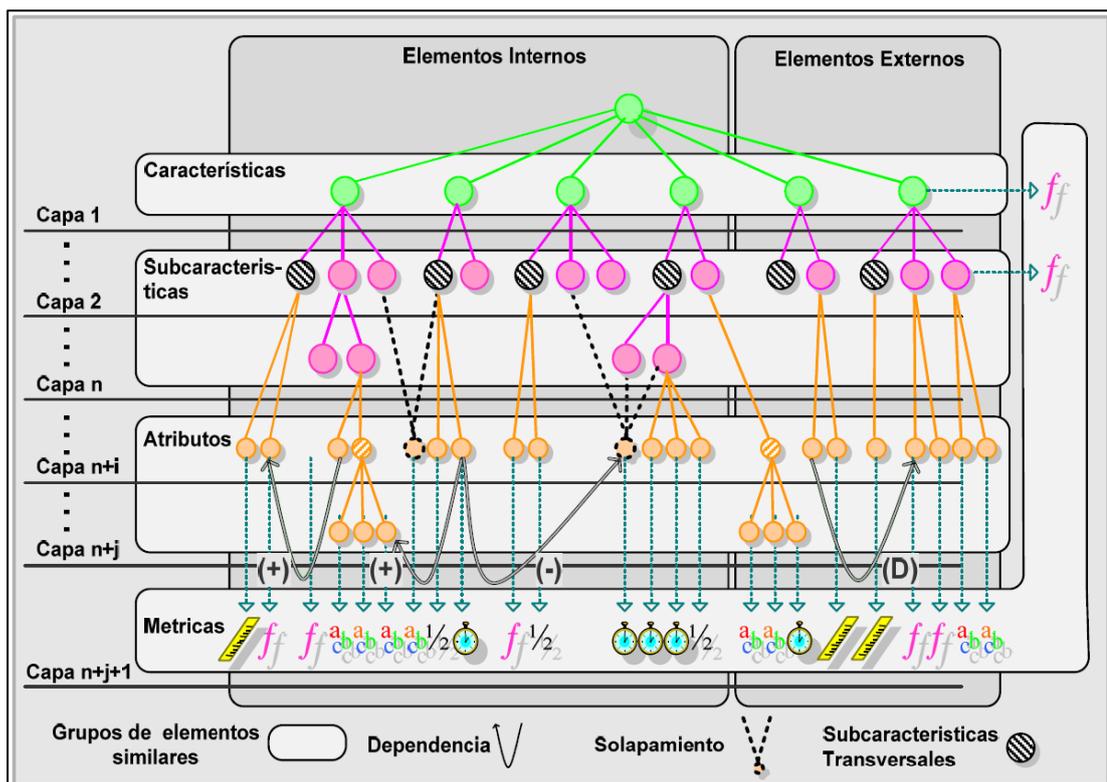
A más de la relación jerárquica entre elementos de distintos niveles, en [CFQ10] se han propuesto varias maneras en que los factores de calidad pueden relacionarse, tales como: (Figura 2.1)

- **Solapamiento.-** Un factor de calidad participa en la descomposición jerárquica de varios niveles superiores, dicho factor puede evaluarse

con métricas diferentes para cada uno de los factores que descompone.

- **Transversalidad.**- Se basa en el concepto de solapamiento, donde cambia la métrica y la definición para cada uno de los factores que descompone, por ejemplo, las seis sub características de “cumplimiento” asociadas a cada una de las características incluidas en el modelo de calidad del estándar ISO/IEC 9126-1 – 2001.
- **Dependencia.**- Es un factor de calidad que se relaciona con otros factores, generalmente del mismo nivel. El número de dependencias puede ser muy elevado, pero como indican en [E04] muchas de ellas pueden ser irrelevantes.

Figura 2.1 Propiedades Estructurales de los Modelos de Calidad de Software



Fuente.- [CFQ10]

2.2.5 Relación de las métricas con los factores de calidad.

Ciertas propuestas de modelos de calidad existentes, incluyen métricas asociadas a un nivel más detallado de descomposición, aunque en algunos casos, como se indica en el estándar [IEEE-1061] según [CFQ10], requieren que las métricas sean aplicadas a los niveles más altos o abstractos de la jerarquía.

2.3 Validación empírica en Ingeniería de Software

En los últimos años, se han propuesto diversas técnicas para la realización de estudios primarios en ingeniería de software, entre las que se encuentran las descritas en [RH08] [EA06]:

- **Experimentos.-** Se caracterizan por medir los efectos de la manipulación de una variable sobre otra, considerando sujetos de estudio asignados al azar [RH08].
- **Casi-experimentos.-** Son experimentos de campo (ambiente industrial), donde los sujetos no se asignan aleatoriamente, ya que precisan determinados conocimientos o habilidades. Los casi – experimentos, permiten obtener datos estadísticos, pero esta característica no se puede generalizar [EA06].
- **Estudio de casos.-** Es una técnica que comprende y explica fenómenos en un entorno de trabajo determinado, debido a que los objetos de estudio son difíciles de comprender aisladamente. Permite el conocimiento detallado en las cadenas causa – efecto y por su naturaleza, deben realizarse en ambientes industriales. Son difíciles de acceder y no son cuantificables por lo que exigen un análisis cualitativo [RH08] [EA06].
- **Estudio Survey.-** Provee un método exhaustivo para el levantamiento de información; sirve para describir, comparar, o explicar conocimientos o actitudes sobre grandes poblaciones; investiga la

naturaleza de una población amplia y permite probar teorías cuando hay poco control sobre las variables [EA06].

De acuerdo a su objetivo los distintos métodos primarios de validación empírica en ingeniería de software, se categorizan como: "exploratorio", "descriptivo", "explicativo" o "mejora" [RH08]. Cada uno de ellos está asociado a distintas características, (Tabla 2.1):

- **Exploratorio.-** Es un método que permite descubrir lo que sucede en un cierto ámbito de estudio; genera ideas e hipótesis para nuevas investigaciones y busca nuevas percepciones.
- **Descriptivo.-** Se investiga la relación existente entre dos o más variables en base a una hipótesis; retrata una situación o fenómeno, describiendo sus elementos fundamentales.
- **Explicativo.-** Busca explicaciones de una situación o un problema de la realidad, mediante el planteamiento de una teoría que establece relaciones causa - efecto.
- **Mejora.-** Es un método de mejora progresiva de un determinado aspecto, de un fenómeno estudiado.

Tabla 2.1. Información general de las características de la metodología de investigación

Metodología	Objetivo Primario	Datos Primarios	Diseño
Survey	Descriptivo	Cuantitativa	Fijo
Casos de Estudio	Exploratorio	Cualitativa	Flexible
Experimento	Exploratorio	Cuantitativa	Fijo
Investigación Acción	Mejora	Cualitativa	Flexible

Elaborado por: Ana Villalta

Fuente: [RH08]

En cuanto a técnicas secundarias para la validación empírica en la ingeniería de software se han descrito al menos dos:

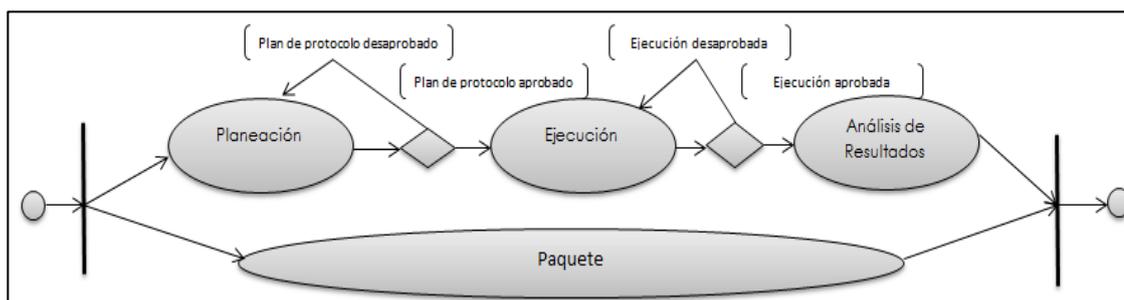
- **Revisión sistemática de la literatura (SLR).**- En las SLR se plantea un conjunto específico de preguntas de investigación y un proceso riguroso de refinamiento, que permite, recopilar, clasificar y sintetizar varios artículos. El objetivo es que estos artículos respondan a las preguntas planteadas, siguiendo un protocolo previamente definido [K07].
- **Mapping.**- El mapeo es un tipo de revisión sistemática, que se diferencia de una SLR normal en el tipo de preguntas de investigación. Las SLR plantean preguntas específicas, mientras que los mapeos abordan una pregunta abierta. Esta técnica se utiliza para realizar un esquema de clasificación y de esta manera abordar un campo de estudio [A14].

2.4 Revisión sistemática de la literatura

La técnica de investigación empírica SLR, es un estudio secundario que se fundamenta en estudios primarios, y evalúa toda la literatura que la investigación aporta, realizando una síntesis de ella como parte del proceso. Una SLR es rigurosa al centrarse en un protocolo previamente definido, por lo que cualquier otro investigador puede seguirlo explícitamente [K07].

Una SLR sigue un proceso que está basado en una pregunta de investigación, que pretende profundizar una temática establecida. Posteriormente, se desarrolla un protocolo que servirá de guía en todo el progreso de la revisión, hasta conseguir una síntesis de los datos extraídos, finalmente, se obtendrá un reporte que documentará los resultados del trabajo.

Figura 2.2 Información general de las características de la metodología de investigación



Elaborado por: Ana Villalta

Fuente: [BGCH05]

Existen varios trabajos que han sido descritos como aporte a la realización de una SLR ej., [K07] [MMA10] [BBHRS06], cuyo objetivo principal ha sido el de identificar y describir guías que sirvan de apoyo en el proceso de la exploración.

En [K07], se describe el proceso a seguir en una revisión sistemática, distribuyendo todas las etapas de una SLR en tres fases, la planificación de la revisión, el desarrollo del trabajo y el reporte del mismo. Estas fases son descritas en el documento con un alto nivel de detalle, observando que varias de ellas requieren iteración, además, se provee un anexo que ejemplifica un protocolo en un SLR basado en las guías propuestas. De esta manera se obtiene un aporte importante para los investigadores en Ingeniería de Software, reiterando, que las guías médicas y libros de texto sociológicos están ampliamente de acuerdo sobre las principales etapas de este tipo de estudio.

En [MMA10], los autores proponen cinco fases a seguir para la realización de una SLR: la identificación del campo de estudio y período a analizar; la selección de fuentes de información; la realización de la búsqueda que requiere la definiciones de criterios tales como, las bases de datos donde se realizará la búsqueda y la sintaxis a utilizar; la depuración de resultados y por último, el análisis de los resultados.

En [BBHRS06], se analizan las recomendaciones incluidas en [K07] para realizar su indagación. El trabajo se puede considerar un ejemplo de cómo desarrollar las fases de una revisión sistemática, su lectura completa, permite observar una SLR desarrollada desde el punto de vista de los autores.

2.4.1 Guías para la realización de SLR en Ingeniería de software

Algunos autores como [K07] [MMA10], han propuesto un conjunto de guías específicas para la realización de SLR's, las cuales se resumen en la Tabla 2.2, en donde se puede observar que todas las fases descritas en [MMA10] se encuentran incluidas en [K07], por lo que pueden ser consideradas un subconjunto de ese estudio. En adición, algunos autores [BGCH05] [BBHRS06] ya han realizado estudios utilizando estas guías, por lo que pueden ser calificados como ejemplos de aplicación.

Tabla 2.2. Comparación de guías para realizar SLR

Fases para una SLR [K07]	Propuesta metodológica para una SLR [MMA10]
Planificación de la revisión	
Identificación de la necesidad de una revisión	Identificación del campo de estudio y período a analizar
Puesta en marcha de una revisión	
Especificar la pregunta(s) de investigación	
Desarrollo de un protocolo de revisión	
Evaluar el protocolo de revisión	
Desarrollo de la revisión	
Identificación de la investigación	Identificación del campo de estudio y período a analizar
Selección de los estudios primarios	Selección de las fuentes de información
	Realización de la búsqueda
Evaluación de la calidad del estudio	Gestión y depuración de los resultados
La extracción de datos y monitoreo	
Síntesis de los datos	
Reporte de la revisión	
Mecanismos de difusión	
Formatear el informe principal	
Evaluar el Informe	Análisis de los resultados

Fuente: [K07], [MMA10]

Elaborado: Ana Villalta

Tabla 2.3. Contribución a las guías para elaboración de protocolos de SLR incluidas en [K07] incluidas en [CHHK13]

Guías para la elaboración del protocolo SLR [K07]	Recomendaciones incluidas en [CHHK13] en relación a las guías para el desarrollo del protocolo propuesto en [K07]
Antecedentes	
Las preguntas de investigación que la revisión se pretende responder	1. Definición de las preguntas de investigación Deben estar al alcance correcto Deben contar con retroalimentación de expertos
La estrategia que se utilizará para buscar los estudios primarios, incluidos los términos de búsqueda y los recursos que se debe buscar. Los recursos incluyen bibliotecas digitales, revistas específicas y actas de congresos. Un estudio inicial de mapeo puede ayudar a determinar una estrategia adecuada.	2. Estrategia de búsqueda Adaptar al estrategia de búsqueda para cada base de datos Plan de manipular la cadena de búsqueda para cada base de datos Las bases de datos se superponen, por lo que definen una estrategia para la identificación y eliminación de resultados duplicados No hay que subestimar el esfuerzo necesario para combinar referencias exportadas de diferentes bases de datos
Estudio de los criterios de selección. Criterios de selección del estudio se utilizan para determinar qué estudios debe incluirse , o excluirse de una revisión sistemática	3. Selección de estudios Examen de los expertos a los criterios de inclusión/exclusión
Estudio de los procedimientos de selección. El protocolo debe describir como se aplicarán los criterios de selección, por ejemplo, cuantos asesores evaluarán cada estudio primario prospectivo y como se va a resolver los desacuerdos entre los evaluadores	Colaboración en la revisión y selección de los trabajos Estrategias para la gestión de equipos y resolución de conflictos
Listas de verificación y procedimientos de evaluación de la calidad del estudio. Los investigadores deben desarrollar listas de control de calidad para evaluar los estudios individuales. El propósito de la evaluación de la calidad guiará el desarrollo de la lista de verificación.	5. Evaluación de la calidad de los estudios Definir los criterios adecuados de evaluación de la calidad Aplicar los criterios para eliminar los estudios de baja calidad Tener múltiples autores independientemente para evaluar la calidad
Estrategia de extracción de datos. Esto define como se obtendrá la información requerida de cada estudio primario. Si los datos requieren manipulación o suposiciones e inferencias a realizar, el protocolo debe especificar un proceso de validación adecuado.	4. Extracción de datos Tener un experto que revise el formulario de extracción de datos Revisión por varios autores Registro y análisis de los datos extraídos
Síntesis de los datos extraídos. Esto define la estrategia de síntesis. Debería aclararse si se pretende un meta - análisis formal y si es así que técnicas se utilizarán	
Estrategia de difusión	
Calendario del proyecto	
Evaluar el protocolo de revisión	

Fuente: [K07], [CHHK13]

Elaborado: Ana Villalta

2.5 Proceso de la Revisión

En las siguientes secciones se describirá el proceso que se tiene que llevar a cabo para la elaboración de una revisión sistemática, siguiendo las guías de B. Kitchenham [K07].

2.5.1 Planificación

Antes de empezar a desarrollar una SLR se debe elaborar un esquema que constituirá el protocolo a seguir durante toda la investigación. En esta sección se describen los conceptos básicos para una planificación correcta de una Revisión Sistemática de la Literatura en la Ingeniería en Software.

- 1) **Necesidad de una SLR.-** En este punto se aclaran las razones por las que se debe realizar la investigación. Generalmente las revisiones sistemáticas son utilizadas para resumir una gran cantidad de aporte a la investigación que se ha registrado en un rango definido de años. El resultado permite establecer el estado del arte de una rama en particular o generar una base teórica importante que servirá para realizar estudios posteriores.

- 2) **Puesta en marcha.-** En algunas ocasiones, a nivel industrial, es necesario redactar un documento de puesta en marcha para la revisión sistemática. El documento incluye un resumen que describe el alcance del proyecto, previo a la aprobación por parte del personal que requiere que se realice la investigación. Este debe incluir:
 - Título del proyecto
 - Antecedentes
 - Preguntas de repaso
 - Asesor / Steering Group Membership (investigadores, profesionales, miembros, responsables de políticas, etc.)
 - Métodos de la revisión
 - Cronograma del Proyecto

- Estrategia de Difusión
- Infraestructura de Apoyo
- Presupuesto
- Referencias

En el caso de las revisiones sistemáticas, que actualmente existen en la Ingeniería de Software Empírica, el paso anterior es reemplazado por la elaboración de un protocolo que incorpora la puesta en marcha y que debe ser revisado por el tutor de la SLR.

3) Preguntas de investigación

Se realiza una lista de preguntas que denotan de manera clara las interrogantes que se quieren resolver en la revisión de los artículos. Este punto constituye la base para la SLR, ya que será la guía con la que se elaborarán las tablas resúmenes. Se recomienda que al realizar las preguntas se tome en cuenta diferentes aspectos, tales como:

- **Población.-** Personas a las que está dirigida la investigación. Por ejemplo: probadores, gerentes, etc. Se puede dirigir también a una categoría de ingeniero de software, por ejemplo, un novato o experimentado ingeniero; un área de aplicación, por ejemplo, sistemas informáticos, sistemas de mando y control; un grupo de la industria, tal como, empresas de telecomunicaciones, o pequeñas empresas de TI.
- **Intervención.-** Es la metodología de software / herramienta / tecnología / procedimiento que se ocupa de un tema específico.
- **Resultados.-** Los resultados de las preguntas de investigación deben apuntar a mejorar algún aspecto de la rama que en la que se realice la intervención o a responder factores que resulten importantes para un grupo de interesados. Entre estos factores se incluye mejorar costos, tiempo y otros aspectos relevantes.

4) Elaboración de un Protocolo.

El protocolo para la revisión sistemática constituye el índice a seguir durante todo el desarrollo de la investigación y define todos los pasos que serán ejecutados a lo largo del trabajo. De esta manera reduce el sesgo para el investigador. El protocolo consta de los siguientes puntos:

- **Antecedentes.-** Describen las referencias de algún trabajo o hecho que constituirá el punto de partida hacia la actual investigación.
- **Preguntas de investigación.-** Deberán ser planteadas con la ayuda de un experto en la materia a estudiar ya que serán la base de toda la investigación.
- **Estrategia de búsqueda de estudios primarios.-** Incluye los términos de búsqueda y los recursos en los que se debe buscar, tales como: bibliotecas digitales, revistas específicas, y resúmenes de congresos.
- **Criterios de selección de estudios.-** Son elaborados para describir qué estudios primarios se utilizarán en la investigación y cuáles de ellos serán descartados.
- **Procedimientos de selección de estudios.-** Constituyen la aplicación de los criterios anteriores. Especifican quiénes son los evaluadores de cada trabajo y cómo se va a resolver los desacuerdos entre ellos.
- **Listas de verificación y procedimientos de evaluación de la calidad del estudio.-** Se deben realizar listas que permitan verificar si cada estudio individual es de calidad para la investigación.
- **Estrategia de extracción de datos.-** El investigador debe detallar cómo va a obtener los estudios precisos para su trabajo.
- **Síntesis de los datos extraídos.-** Para sintetizar los datos lo ideal es que los investigadores se apoyen en técnicas estadísticas formales. De no ser el caso se puede optar por una síntesis informal que responda al objetivo de la revisión.
- **Estrategia de difusión.-** En esta etapa se describe la forma en que los investigadores piensan transmitir los resultados de la SLR.
- **Calendario del proyecto.-** Se establece un cronograma que servirá de apoyo para los investigadores a cargo de la SLR.

2.5.2 Desarrollo de la Revisión

Una vez planificada la SLR se continúa con la fase de desarrollo que incluye los siguientes pasos:

1) Generación de una estrategia de búsqueda.

Es conveniente que para la generación de una estrategia de búsqueda se consulte con algún experto o se consiga un estudio de las bases de datos en las que se pretende realizar la búsqueda. Generalmente, las bibliotecas digitales disponen de información importante sobre el uso de operadores lógicos y sobre la forma en la que se debe incorporar las cadenas de búsqueda en el motor.

Las estrategias de búsqueda permiten evaluar el volumen de los estudios relevantes, realizar pruebas piloto utilizando diversas combinaciones de términos derivados de la pregunta de investigación y generar cadenas de búsqueda óptimas.

Algunas estrategias para generar las cadenas de búsqueda son:

- Romper la pregunta en partes, es decir, población, intervención y resultados.
- Elaborar una lista de sinónimos, abreviaturas y palabras alternativas.
- Construir cadenas de búsqueda utilizando operadores lógicos.

Una vez definidas las estrategias se debe seleccionar las bases de datos o repositorios sobre los cuales se realizarán las búsquedas. Estas pueden ser de tipo automático (usualmente en base a bibliotecas digitales especializadas) o manuales (en base a listas de referencias de los estudios primarios, revistas no indexadas, literatura gris, registros de investigación, Internet, entre otros).

Para la Gestión Bibliográfica y recuperación de documentos, es importante utilizar un esquema diseñado para ubicar todos los datos necesarios de la bibliografía recuperada.

Lecciones aprendidas para la búsqueda:

Para que la búsqueda se realice correctamente no se debe olvidar la importancia de seleccionar y justificar una estrategia, así como de buscar fuentes electrónicas diferentes. Al realizar la búsqueda se recomienda utilizar sinónimos de las palabras de búsqueda. Además es esencial tomar en cuenta que los motores de búsqueda no están totalmente acoplados para realizar revisiones sistemáticas. Sin embargo expertos en el área señalan varios motores que aportan a la investigación de Ingeniería en Software como:

- IEEEExplore
- Biblioteca Digital ACM:
- Google scholar (scholar.google.com)
- Biblioteca Citeseer (citeseer.ist.psu.edu)
- Inspec (www.iee.org/Publish/INSPEC/)
- ScienceDirect (www.sciencedirect.com)
- ElCompendex(www.engineeringvillage2.org/Controller/Servlet/AthensService).
- SpringerLink (acceder a revistas como empírica Ingeniería de Software y actas de congresos)
SCOPUS (que presume de ser la más grande base de datos de resúmenes y citas).

2) Selección de los estudios.

La bibliografía recuperada en el paso anterior es la que se considera importante. Sin embargo, se debe buscar un método de selección que ayude a refinar los estudios de tal manera que obtengamos artículos con relevancia actual. Se debe llevar a la práctica los criterios de selección incorporados en el protocolo, estos criterios pueden ser refinados en esta etapa.

Para la selección de los estudios es importante que se tome en cuenta la selección por título, resumen y/o conclusiones. Luego se debe emplear criterios de inclusión/exclusión como: idioma, revistas, autores, participantes o sujetos, diseño de la investigación, métodos de muestreo y fecha de publicación, los criterios de inclusión/exclusión deben basarse en las preguntas de investigación. Se sugiere mantener un listado con la literatura excluida.

Cuando un solo investigador realiza la revisión sistemática es muy importante que se revalúen los estudios con la opinión del asesor, comprobando si existe coherencia con los criterios de inclusión exclusión señalados.

3) Evaluación de la Calidad de Estudio

Se debe desarrollar instrumentos de calidad a modo de listas, que permitan controlar si cada estudio individual es de calidad para la investigación. Cada lista debe contar con escalas numéricas que permitan la evaluación de los artículos

Una vez definida la lista de verificación es importante ponerla en práctica. La puesta en marcha de esta lista nos puede servir para ayudar a la selección de estudios primarios o al análisis de datos y síntesis.

4) Extracción de datos

En esta etapa se elaboran los formularios que permitirán obtener todos los datos de los artículos seleccionados y de los investigadores que obtienen los estudios.

Los formularios deben estar diseñados para recopilar toda la información necesaria, para hacer frente a las preguntas de revisión y los criterios de calidad de estudio. Deben contener principalmente las preguntas de investigación, nombre del crítico, la fecha de extracción de los datos, título,

autores, revistas, detalles de la publicación y espacio para notas adicionales. Si existe más de una publicación con los mismos datos se debe verificar la publicación más completa.

Es conveniente que la selección definitiva de los estudios se realice entre dos o más investigadores. Los desacuerdos que se presenten se pueden resolver mediante un consenso o por un arbitraje.

5) Síntesis de los datos

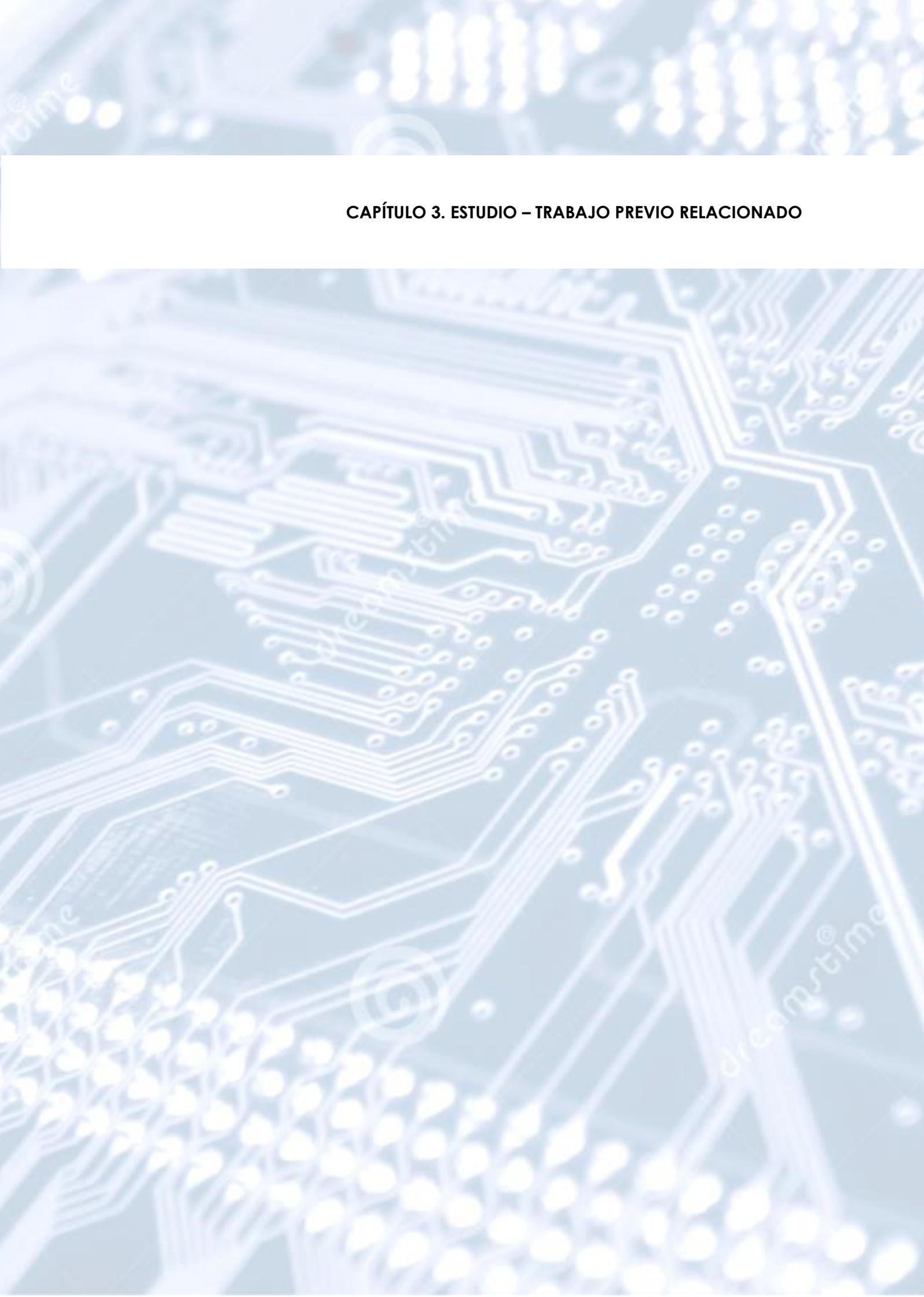
Lo ideal en este paso es realizar un meta-análisis. Se pueden utilizar también otras técnicas estadísticas para realizar la síntesis de los datos o proponer métodos informales que respondan al objetivo del estudio.

2.5.3 Reporte de la Revisión

En esta fase se considera la difusión de los resultados a las partes interesadas y se debe especificar la estrategia de difusión en la etapa de realización del protocolo. Algunas formas de difusión son:

- Los diarios y revistas
- Comunicados de Prensa
- Los folletos de resumen cortos
- Posters
- Páginas Web
- La comunicación directa a los afectados

Los formatos para la presentación de la SLR, pueden ser reportados por medio de un informe técnico o en una sección de una tesis; en un artículo de revista o conferencia.



CAPÍTULO 3. ESTUDIO – TRABAJO PREVIO RELACIONADO

3.1 Introducción

Con el objetivo de lograr la actualización del estado del arte y de reforzar los conceptos estudiados en el capítulo 2, se analiza un trabajo realizado en el año 2007 [DO07]. El enfoque de este trabajo relaciona tanto la calidad del software como la investigación empírica.

Los autores plantearon una indagación exploratoria, que concluyó con la construcción de un catálogo de atributos de calidad de software. Para realizar esta construcción se examinaron varios artículos científicos utilizando la técnica de mapeo de la literatura, que es un tipo de revisión sistemática no rigurosa y se usa para realizar un esquema de clasificación en un campo de estudio [A14].

Este capítulo examina y resume las actividades realizadas en [DO07]. Además, detalla el proceso utilizado para la construcción del catálogo resultante, a partir de la descripción de propiedades y atributos incluidos en cada uno de los artículos estudiados.

El capítulo concluye con la selección de ciertos artefactos que se consideran apropiados para incorporarlos en el presente estudio, así como la asociación de nuevos campos a dichos elementos.

3.2 Recopilación de técnicas

Tal como se mencionó en la introducción de este capítulo, en el 2007 se realizó un mapeo informal de la literatura. El mismo concluyó con un catálogo de atributos de calidad del software que contiene 448 atributos técnicos y 255 atributos no técnicos.

Para la realización de este catálogo se siguieron los siguientes pasos:

- 1)** Construcción de catálogos individuales para cada artículo leído
 - a. Convertir cada propuesta en un formato común
 - b. Identificación de los Tipos de atributos de Calidad
 - c. Clasificación de Atributos Técnicos en Funcionales y No-Funcionales

- 2)** Construcción de un catálogo unificado de atributos técnicos
 - a. Inserción de los atributos de calidad dentro del Estándar ISO/IEC 9126 y aplicación de Reglas de Composición
 - b. Comparación de la matriz de atributos Técnicos frente al modelo ISO-EXTENDIDO y otros patrones de calidad.
 - c. Consolidación del Catálogo Unificado

- 3)** Construcción de un catálogo unificado de atributos no técnicos
 - a. Elaboración de un diseño de tabla en Excel para la ubicación de los artículos.

Las siguientes secciones describen cada uno de los pasos listados:

Construcción de catálogos individuales para cada artículo leído

Para la construcción de un catálogo común se plantearon los siguientes pasos:

a. Convertir cada propuesta en un formato común

El contenido relevante de cada artículo revisado fue categorizado en una tabla (Tabla 3.1), diseñada e implementada utilizando como herramienta de soporte el programa Excel. La tabla 3.1 constituyó el "Formato Común" utilizado en cada propuesta.

Tabla 3.1. Formato común usado en todas las propuestas

Título del Artículo									
Autores. [NombreNemotécnico]									
No.	Levels	[CKV09]	Type	If Technical, which type	References	Descriptions	Metrics	Metrics Description	Interpretation of the metric

Fuente: [DO07]

Elaborado: Ana Villalta

La tabla está dividida en campos. Cada uno de ellos reúne información precisa que cumple con el objetivo de estudio, de esta manera:

- **El título del artículo.**- Campo en donde se ubica el nombre del artículo leído.
- **El nombre de los autores.**- Nombre asignado de manera nemotécnica compuesto por las iniciales de los apellidos de los autores más el año de publicación del trabajo analizado. Los autores del trabajo de titulación del año 2007 proponen un ejemplo. (Marco **T**orchiano, Letizia **J**accheri [TJ+02]).

- **Nº.-** Total de factores encontrados.
- **Niveles.-** Número de niveles que componen la jerarquía.
- **Listado de Factores de Calidad.-** Nombres que han sido asignados en cada artículo para cada factor. El título de esta columna es el mismo nombre nemotécnico utilizado para asignar el nombre de los autores.
- **Tipo.-** Tipo de cada factor identificado: Técnicos (T), No-Técnicos (NT), Otros (O). Calidad en uso (CU) Métricas(M) y Atributos Técnicos – No Técnicos (NT-T)
- **Si es técnico, Qué tipo.-** Combinaciones para cada tipo de factor Técnico esto es, Funcionales (F), No funcionales (NF) o Funcionales & No funcionales (F-NF)
- **Referencia.-** Referencias en donde se señala el número de la página donde se ha encontrado la información para cada artículo
- **Descripción.-** Columna en la que se ubican las descripciones encontradas para cada factor de calidad
- **Métricas.-** Métricas ordenadas por niveles
- **Descripción de las métricas.-** Descripciones encontradas para cada métrica.
- **Interpretación de la métrica.-** Fórmula correspondiente para cada una de las métricas, en el caso de que ésta se encuentre en el artículo leído.

Para profundizar en el conocimiento de los campos que forman la parte fundamental de este catálogo, tales como el "Tipo" y "Si es técnico, qué Tipo", se estudiaron más a fondo ciertos conceptos:

Tipos de factores

- **Técnicos (T).-** Se refiere a características de calidad del componente a evaluar. Se podrían subclasificar en funcionales y no-funcionales.
- **No-Técnicos (NT).-** Son atributos que incluyen aspectos relacionados al costo, vendedor, aspectos contractuales y características en relación al producto.
- **Otros (O).-** Son atributos que no pueden ser categorizados como técnicos o no técnicos en un producto o componente software.

- **Calidad en uso (CU).**- Son atributos que hacen referencia a aspectos que pueden ser evaluados por los usuarios una vez que el producto haya sido implementado.
- **Métricas (M).**- Nivel en el que el componente o producto software incorpora cierto atributo, las métricas se componen de un valor de unidad, una unidad de medida y una escala; adicionalmente deben tener asignado un procedimiento de medición y una forma específica en la que los resultados de la medición deben ser interpretados.

Combinaciones en los tipos de atributos:

- **Atributos Técnicos – No Técnicos (T-NT).**- Son atributos que por su naturaleza se han considerado útiles para la evaluación de aspectos técnicos y también algunos aspectos no técnicos. Por esta razón se han incluido tanto en la matriz de atributos Técnicos como en la de atributos No Técnicos
- **Atributos Funcionales y No Funcionales.**- Son aquellos atributos identificados como Técnicos, y pueden ser considerados de utilidad para evaluar subcaracterísticas de tipo funcional pero también no funcionales.

b. Identificación de los Tipos de atributos de Calidad

Una vez clasificados los factores de calidad a ser exportados, se examinaron para determinar el tipo al que pertenecen. Esta actividad se realizó con el propósito de facilitar el conteo de tipos de atributos en el "catálogo unificado resumen", que será detallado más adelante.

La nomenclatura utilizada para esta actividad es la siguiente:

- Técnico (T)
- No técnico (NT)
- Otros (O)
- Calidad en Uso (CU)

- Métrica (*M*)).

c. Clasificación de Atributos Técnicos en Funcionales y No-Funcionales

Una vez determinado el tipo al que pertenece cada factor de calidad, se analizó, para cada uno de los atributos técnicos, el subtipo al que pertenece. Al igual que el caso anterior, esta actividad se realizó con el propósito de facilitar el conteo de subtipos de atributos en el "catálogo unificado resumen". La nomenclatura utilizada para esta actividad es la siguiente:

- Funcionales (*F*)
- No-Funcionales (*NF*)

A los factores que no encajaban con las clasificaciones mencionadas se los definieron como:

- No Definidos (*ND*).

Tabla 3.2. Ejemplo del formato común utilizado para la ubicación de factores de calidad de los artículos individuales leídos por los autores utilizando el trabajo [GD05]

A Process Based Model for Measuring Process Quality Attributes A. Selcuk Guceglioglu and Onur Demirors [GD05]									
No.	Levels	[GD05]	Type	If Technical, which type	References	Descriptions	Metrics	Metrics Description	Interpretation of the metric
1	1	Functionality	T	F	Selcuk pag 5 Fig. 1. Measurement categories and metrics of the model	is defined for evaluating the capability of the process to provide functionality properties			
2	1 1	Suitability Metric	T	F	Selcuk pag 4 3.1 Measurement Structure of the Model	Suitability metrics are used for ensuring that business process activities are complete and adequate for performing the tasks.			
3	1 1 1	Functional Adequacy	M		Selcuk pag 5 Fig. 1. Measurement categories and metrics of the model		$X=1-A/B$ A= Number of activities in which problems about functional adequacy are detected in evaluation, B= Number	Count the number of activities that are not functionally adequate, and compare with the number of activities	$0 \leq X \leq 1$ The closer to 1, the more functional adequacy of the business process
4	1 1 2	Functional Completeness	M		Selcuk pag 5 Fig. 1. Measurement categories and metrics of the model				
5	1 2	IT Based Functionality Metrics	O		Selcuk pag 4 3.1 Measurement Structure of the Model	Examine the Information Technology usages in the process activities			
6	1 2 1	IT Usage	M		Selcuk pag 5 Fig. 1. Measurement categories and metrics of the model				
7	1 3	Accuracy Metrics	T	F	Selcuk pag 4 3.1 Measurement Structure of the Model	Accuracy metrics investigate the capability of the process to achieve correct or agreeable results			
8	1 3 1	Functional Accuracy	M		Selcuk pag 5 Fig. 1. Measurement categories and metrics of the model				
9	1 4	Interoperability Metrics	T	F	Selcuk pag 4 3.1 Measurement Structure of the Model	investigate the capability of the process interactions with other processes and problems experienced during the interactions			
10	1 4 1	Data Exchangeability	M		Selcuk pag 5 Fig. 1. Measurement categories and metrics of the model				
11	1 5	Security Metrics	T	F	Selcuk pag 5 3.1 Measurement Structure of the Model	it investigates the protection of the information and its access			
12	1 5 1	Access Auditability	M		Selcuk pag 5 Fig. 1. Measurement categories and metrics of the model				
13	2	Reliability	T	NF	Selcuk pag 5 3.1 Measurement Structure of the Model	is used for evaluating the capability of the process to provide reliability			
14	2 1	Maturity Metrics	T	NF	Selcuk pag 5 3.1 Measurement Structure of the Model	investigate the failures that may happen in the process activities and failure avoidance mechanisms employed for preventing from			
15	2 1 1	Failure	T	NF	Selcuk pag 5 Fig. 1. Measurement categories and metrics of the model				
16	2 1 2	Failure Avoidance	T	NF	Selcuk pag 5 Fig. 1. Measurement categories and metrics of the model		X= Number of failure avoidance mechanisms	Count the number of mechanisms that will provide failure avoidance	The higher value of X, the more failure avoidance of the business process
17	2 2	Recoverability Metrics	T	NF	Selcuk pag 5 3.1 Measurement Structure of the Model	investigate the capability of the process to continue with minimum data lost when abnormal events occur.			
18	2 3	Restorability	T	NF	Selcuk pag 5 3.1 Measurement Structure of the Model	provide re-establishing an adequate level of performance and recovering the data in case of a failure.			

Fuente: [DO07]

Una vez tabulados los datos de los artículos en tablas individuales, se elaboraron catálogos de calidad unificados consolidando los datos incluidos en cada tabla. Se consideraron los atributos Técnicos y No Técnicos que fueron clasificados, utilizando tablas específicas para cada caso.

1) Construcción de un catálogo unificado de atributos técnicos

Para construir el catalogo unificado de atributos técnicos, se siguieron los siguientes pasos:

a. Inserción de atributos de calidad dentro del Estándar ISO/IEC 9126 y aplicación de Reglas de Composición

Los atributos técnicos fueron analizados y comparados con el estándar de calidad ISO/IEC 9126-1, con el objeto de categorizarlos en relación a sus 6 características y 27 subcaracterísticas.

Se diseñó una tabla en Excel que incluye los siguientes campos:

- **Código Origen.-** Corresponde al número de nivel de cada atributo en la tabla individual del artículo correspondiente.
- **Origen.-** Corresponde a la nomenclatura definida para cada artículo leído, tal nomenclatura puede ser observada en la misma ubicación en donde se ha incorporado los nombres de los autores en sus tablas individuales.
- **Niveles.-** Corresponde al número de nivel jerárquico al que actualmente pertenece el atributo.
- **Código Consolidado.-** Corresponde al número de nivel de la nueva tabla.
- **Atributos.-** Corresponde al nombre de la característica o subcaracterística definida en el estandar ISO/IEC 9126-1
- **Descripción.-** Corresponde a las descripciones encontradas para cada factor.

Para la construcción del catálogo unificado los investigadores utilizaron las reglas de composición descritas en [C07]. De esta manera se consiguió agrupar dentro del catálogo a los factores semánticamente similares.

Tabla 3.3. Ejemplo del uso de la regla de identificación

No.	Código Origen	Origen	Levels	Código Consolidado	Attributes
1	1 1 2 1 1 1 1	ISO/IEC 9126-1 [BTV03] [GD05] [ISO12119] [Kon98] [Kun03] [Lux] [SB03]	1	1	Functionality Functionality Functionality Statements on Functionality Functional requirements Functionality Functional Calculuses Functionality
2	1.1 1.1 1.1 1.2	ISO/IEC 9126-1 [BTV03] [GD05] [SB03] [SJG03]	1 1	1.1	Suitability Suitability 1 Suitability Metric Suitability Functionality description
3	1.1.1	[SB03]	1 1 1	1.1.1	Technological suitability
4	1.1.1.1	[SB03]	1 1 1 1	1.1.1.1	Methodological suitability
5	1.1.1.2	[SB03]	1 1 1 2	1.1.1.2	Technological infrastructure suitability
6	1.1.1.3	[SB03]	1 1 1 3	1.1.1.3	Appropriateness of development environment

Tabla 3.4. Ejemplo de aplicación de la regla Drag-and-drop

3.2 3.1 3.3	ISO/IEC 9126-1 [BTV03] [GD05] [SB03]	3 1	3.1	Understandability 1 Understandability Understandability Metrics Understandability
3.3.1	[SB03]	3 1 1	3.1.1	Contextualization
3.3.1.1	[SB03]	3 1 1 1	3.1.1.1	Commonality
3.3.1.2	[SB03]	3 1 1 2	3.1.1.2	Variability
3.3.1.3	[SB03]	3 1 1 3	3.1.1.3	Dependencies

Fuente: [DO07]

Tabla 3.5. Extracto del catálogo resultante del proceso de composición de atributos Técnicos

				ISO/IEC 3126-1	TJ-02	[PVL97]	[OVM97]	[ISO12119]	[MoT02]	[Kun03]	[Kun00]	GD05	[Kon96]	[WJ01]	[SG03]	BTV03	[Luz]	SB03	SJG03	TOTAL
No.	Código Origen	Origen	Atributos																	
1		ISO/IEC 3126-1	Functionality																	
1		[BTV03]	Functionality																	
1		[GD05]	Functionality																	
2		[ISO12119]	Statements on Functionality																	
1		[Kon96]	Functional requirements	1			1		1		1	1			1	1	1			8
1		[Kun03]	Functionality																	
1		[Luz]	Functional Calculuses																	
1		[SB03]	Functionality																	
1.1		ISO/IEC 3126-1	Suitability																	
1.1		[BTV03]	Suitability																	
2		[GD05]	1 Suitability Metric	1								1			1		1	1	1	5
1.1		[SB03]	Suitability																	
1.2		[SJG03]	Functionality description																	
3	1.1.1	[SB03]	Technological suitability															1		1
4	1.1.1.1	[SB03]	Methodological suitability															1		1
5	1.1.1.2	[SB03]	Technological infrastructure suitability															1		1
6	1.1.1.3	[SB03]	Appropriateness of development environment															1		1
1.2		ISO/IEC 3126-1	Accuracy																	
1.2.1		[BTV03]	Accuracy																	
11	1.2.2	[BTV03]	2 Computational Accuracy	1								1			3	1	1			7
1.3		[GD05]	Accuracy Metrics																	
2.2.2		[Luz]	Requirement of precision of the system																	
1.2		[SB03]	Accuracy																	
9	1.1.5	[SB03]	Correctness															1		1
10	1.1.6	[SB03]	Consistency															1		1
14	3.3	[WJ01]	Reports and queries are what the user wants											1						1

Fuente: [DO07]

b. Comparación de la matriz de atributos Técnicos frente a al modelo ISO-EXTENDIDO y otros factores de calidad

En esta fase, los factores de calidad, incluidos en el catálogo unificado, fueron comparados con el modelo ISO-EXTENDIDO y factores de calidad descritos en [Car05], logrando una re categorización jerárquica de atributos de calidad aún más específica.

c. Consolidación del Catálogo Unificado

Una vez analizado el modelo ISO-Extendido y los patrones de calidad propuestos en [Car05], los investigadores elaboraron un catálogo denominado "Consolidación Final Técnicos". Este catálogo reúne los campos descritos en la sección "a) Inserción de atributos de calidad dentro del Estándar ISO/IEC 9126 y aplicación de Reglas de Composición" de este capítulo.

En el catálogo mencionado se crearon nuevas subcaracterísticas que detallan con mayor precisión ciertos atributos. Se distinguen los siguientes atributos:

- **Atributos Eliminados.**- Atributos que no responden a la categorización de componentes COST que definen los autores
- **Atributos de calidad en uso.**- Atributos que responden a conceptos de eficacia, productividad y satisfacción con los que un usuario específico puede lograr sus objetivos, en un ambiente particular en el que se desarrolla. [ISO14598-1]
- **Atributos que son métricas.**- Las métricas se componen de un valor de unidad, una unidad de medida y una escala. Adicionalmente deben tener asignado un procedimiento de medición y una forma específica en la que los resultados de la medición deben ser interpretados.
- **Atributos generales.**- Atributos que no pueden ser categorizados en ningún tipo de jerarquía. Estos atributos no presentan conceptos bien definidos que permitan categorizarlos
- **Atributos de calidad del Sistema.**- Atributos que logran seguridad, disponibilidad, rapidez e integración con el propio sistema y otros sistemas existentes.
- **Atributos semánticamente ambiguos.**- Atributos que no pueden ser evaluados, medidos ni analizados.

Se incorporaron a la matriz los elementos que responden al concepto de **Solapamiento de Atributos**. Se vio la necesidad de crear subcaracterísticas adicionales a los modelos fijos para añadirlos a la matriz. Esto permitió la incorporación de varios atributos que no pudieron ser clasificados dentro de la categorización normal.

Además se encontraron los denominados "Atributos semánticamente heterogéneos". Este tipo de atributos tienen el mismo nombre pero hacen referencia a aspectos de calidad distintos.

2) **Construcción de un catálogo unificado de atributos no técnicos**

Los atributos clasificados como -No Técnicos- fueron analizados pero no se contrastaron con ningún modelo. Se recopilaron varios atributos. Para su clasificación se elaboró un diseño de tabla en Excel con varios campos separados en diferentes columnas. El resultado presentó el siguiente formato:

- **Código Origen.-** Corresponde al número de nivel de cada atributo en la tabla individual del artículo correspondiente.
- **Origen.-** Corresponde a la nomenclatura definida para cada artículo leído, tal nomenclatura puede ser observada en la misma ubicación en donde se ha incorporado los nombres de los autores en sus tablas individuales.
- **Niveles.-** Corresponde al número de nivel jerárquico al que actualmente pertenece el atributo.
- **Código Consolidado.-** Corresponde al número de nivel de la nueva tabla.
- **Atributos.-** Corresponde al nombre de la característica o subcaracterística definida en la norma ISO/IEC 9126-1
- **Descripción.-** Corresponde a las descripciones encontradas para cada factor.

Se aplicaron las reglas de composición y se trabajó con los atributos Técnicos, tomando al artículo [ISO12119] como base.

Tabla 3.6. Reglas de Composición aplicadas entre el modelo [ISO12119] y [PVL97]

Extended ISO/IEC 9126-1	IDENTIFICATION	FO NESTING	DO NESTING	DRAG-AND-DROP	ABSTRACTION	SUBTOTAL
Identifications and Indications				3	2	
Identification of the product description						
Product identification				1		
Name						
Version						
Variant						
Supplier	1			10		
Name						
Address						
Work Task						
Required System						
Hardware						
Processing Unit including co-processors						
Main memory size						
Types and sizes of peripheral storage						
Extension cards						
Input and output equipment						
Network environment						
Software						
System software						
Other software						
Items to be delivered						
Installation						
Support	1					
Maintenance						
Rules Applied	2	0	0	14	2	18
% Application	0%	0%	0%	0%	0%	0%
GRAN TOTAL	2	0	0	14	2	18
% Application	11%	0%	0%	78%	11%	100%

Fuente: Tesis del año 2007

3.3 Análisis de artefactos utilizados

A continuación se analizan los artefactos utilizados en el trabajo [DO07] y se define cuáles de ellos serán incorporadas en el presente estudio.

Luego de estudiar las tablas empleadas para reunir los factores de calidad y una vez verificado su diseño, se decidió utilizar en este trabajo el denominado "Formato Común usado en todas las propuestas" (ver Tabla 3.7).

Tabla 3.7. Formato común para los artículos individuales leídos

Título del Artículo									
Autores. [NombreNemotécnico]									
No.	Levels	[CKV09]	Type	If Technical, which type	References	Descriptions	Metrics	Metrics Description	Interpretation of the metric

Fuente: Tesis del año 2007

Elaborado: Ana Villalta

A más de indagar en el documento propuesto por los autores se han observado las tablas implementadas. Para este estudio es de mucha utilidad la tabla denominada "Tabla Resumen", la cual consta de:

- **Nombre Nemotécnico.**- Campo en donde se toma el nombre nemotécnico de cada tabla, formato individual.
- **Título de Artículo.**- Nombre que se la ha dado al artículo leído.
- **Autores.**- Nombre asignado de manera nemotécnica compuesto por las iniciales de los apellidos de los autores, más el año de publicación del trabajo analizado.
- **Número de Niveles.**- Niveles que corresponden a las tablas formato individual.
- **Número de Atributos por nivel.**- Sumatoria del número de atributos.

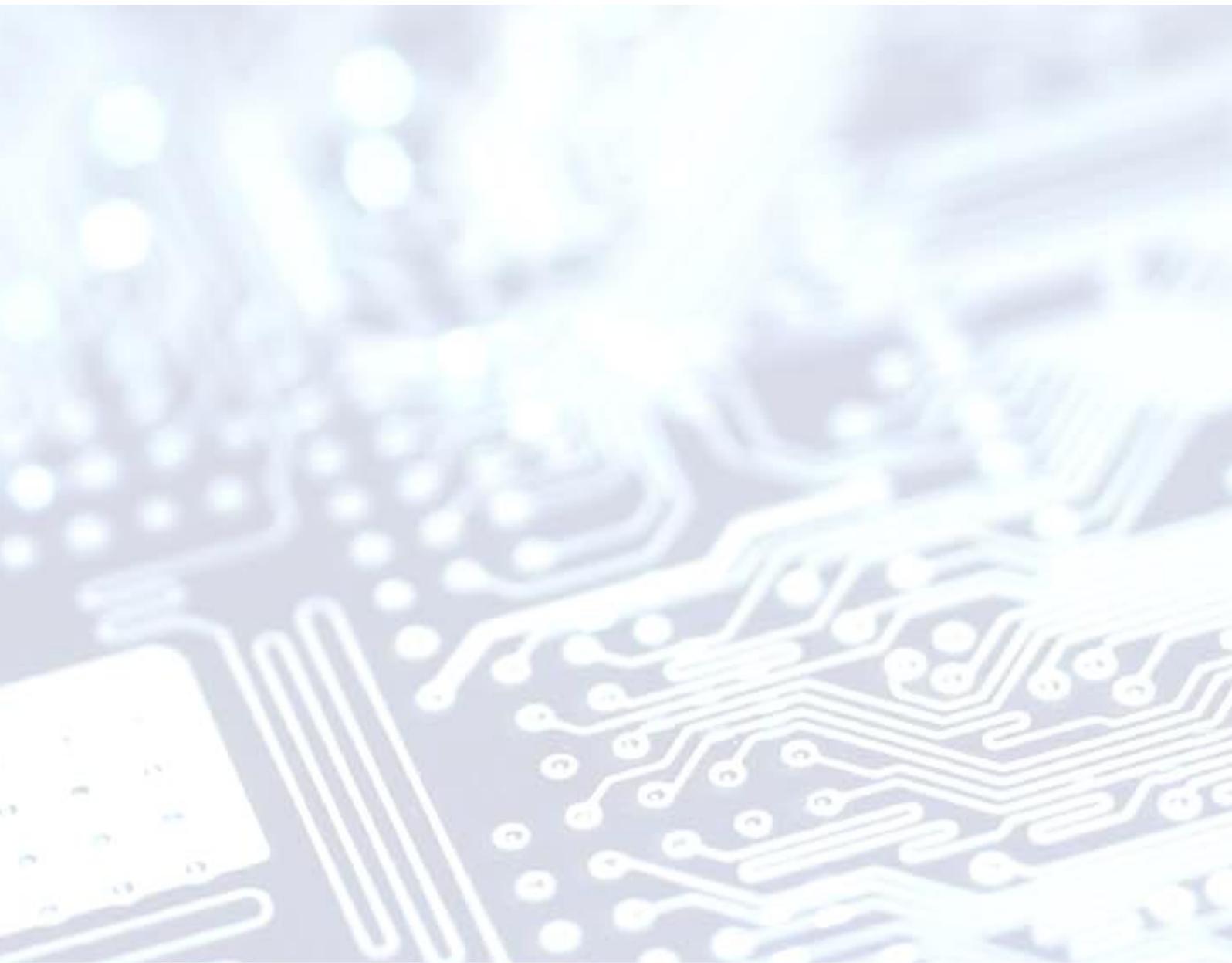
- **Total de Atributos.-** Sumatoria de todos los atributos, por niveles y por artículo.
- **Tipos de Atributos Identificados.-** Número de atributos por tipo y por artículo.
 - Atributos Técnicos Funcionales (TF)
 - Atributos Técnicos No Funcionales (TNF)
 - Atributos No Técnicos (NT)
 - Atributos Categorizados como Otros (O)
 - Atributos que son Métricas (M)
 - Atributos Técnicos No Definidos (T-ND)
 - Atributos de Calidad en el Uso (CU)
- **Solapamiento en QF.-** Número de atributos que tienen overlapping categorizándolos por tipo y por artículo
 - Atributos T- NT & No Funcionales
 - Atributos T- NT & Funcionales
 - Atributos Técnicos F&NF
 - Atributos T-NT F&NF
- **Total de Atributos.-** Sumatoria de todos los atributos de calidad incluyendo los que tienen overlapping.
- **Total de Atributos por cada nivel.-** Sumatoria de los atributos por cada nivel.
- **Total de Atributos por tipo.-** Sumatoria de los atributos por cada tipo.
- **Total de Atributos con Solapamiento.-** Sumatoria de los atributos que incorporan la propiedad de solapamiento.

Se tomaron todos los campos de la "Tabla Resumen" y se añadieron 7 campos más para cumplir el propósito de la investigación:

- **Incluye métricas (Si/No).-** En este campo se incorpora la palabra "Si", en el caso de que en el artículo leído se encuentren métricas de calidad, caso contrario, la palabra "No".
- **Métricas aplicadas a.-** Se define a que factor de calidad se han aplicado las métricas encontradas, (características, sub

características), dentro de la jerarquía establecida por los autores de los artículos.

- **Incluye método de construcción (Si/No).**- En este campo, se incorpora la palabra "Si", en el caso de que en el artículo leído se encuentre un método de construcción, caso contrario, la palabra "No".
- **Pasos del método de construcción.**- Se listan los pasos que los autores de los artículos han propuesto para la construcción del método.
- **Basado en qué modelo.**- Se detalla en qué modelo se han basado los autores de los artículos para describir su modelo actual.
- **Tipos de relaciones entre elementos.**- Se lista el tipo de relación entre los elementos, ya sea, jerárquica, de soporte, en capas, etc.
- **Tipo de modelo.**- Se describe si el modelo es fijo, a la medida o mixto.
- **Aplicación para la que se usa el modelo.**- Se describe en qué etapa de construcción del software se puede aplicar el modelo propuesto.



CAPÍTULO 4. PLANEACIÓN DE LA REVISIÓN



4.1 Introducción

Tal como se vio en la sección 2.4, las Revisiones Sistemáticas de Literatura buscan responder a preguntas de investigación específicas, siguiendo un proceso riguroso de refinamiento, recopilación, clasificación y síntesis de artículos relevantes, establecido en un protocolo predefinido.

En este estudio el protocolo sigue las guías propuestas en [K07], las cuales fueron descritas en el capítulo 2. Se puede observar que una SLR se estructura en tres fases, siendo la primera de ellas la planificación de la revisión.

Este capítulo se enfoca en la planificación. Fundamenta las razones por las cuales es necesario realizar la indagación exploratoria usando la técnica de SLR, plantea las preguntas de investigación que servirán de base para la revisión y define un protocolo que constituirá el índice a seguir durante todo el desarrollo de la investigación y que busca reducir el sesgo por parte del investigador.

4.2 Campo a estudiar

El campo en el que se desarrolla la investigación es la Calidad de Software, específicamente, los modelos de calidad de software.

4.3 Formulación del problema

Se observó que existe mucha documentación que trata sobre modelos de calidad de software. Ésta se encuentra en varios artículos contenidos en diversas fuentes de investigación que pueden o no ser relevantes para un determinado estudio. Los investigadores que la revisen deben contar con el tiempo suficiente para discernir qué trabajos aportan a sus áreas de interés particular.

Para solucionar este problema se buscarán artículos que respondan a preguntas de investigación que pueden ser importantes para los interesados en la Calidad de Software, tomando como guía las líneas base de una técnica rigurosa de investigación empírica (SLR).

4.4 Preguntas de investigación

Para determinar las preguntas de investigación, como primer punto se definió el grupo de personas a las que se quiere llegar con esta intervención, el área propuesta dentro de la Ingeniería en Software y los resultados que se esperan conseguir. Para evaluar todas estas posibilidades se puso en práctica los conceptos de población, intervención y resultados explicados en la sección 2.5.1. En esta se recomienda, antes de establecer las preguntas de investigación, definir los conceptos mencionados (ver Tabla 4.1).

Población.- La población a la que se quiere llegar con esta intervención es la de estudiantes y profesionales de Ingeniería en sistemas de la Universidad del Azuay.

Intervención.- El área de estudio es la de Calidad de Software, se especifican sus modelos y los principales elementos y propiedades que los definen.

Resultados.- Se busca mejorar la calidad de consulta de los artículos, tomando en cuenta que las derivaciones están categorizadas por preguntas de investigación en tablas de estudio. Otro de los propósitos es minimizar el tiempo de búsqueda.

Tabla 4.1. Preguntas de Investigación y sus objetivos

PREGUNTAS	MOTIVACION PRINCIPAL
<p>P1. ¿Cuáles son los elementos que estructuran los modelos de calidad de software?</p>	<p>Diferenciar los tipos de estructuras, así como los catálogos de factores que poseen los distintos modelos de calidad de software, distinguiendo el desglose de sus niveles tanto como los aspectos de formalización en la distribución de los elementos.</p>
<p>P2. ¿Cómo se relacionan los distintos elementos entre sí?</p>	<p>Estudiar la distribución de los elementos de calidad de software, distinguir las métricas para cada modelo, además de a qué factor de calidad pueden ser aplicadas.</p>
<p>P3. ¿Cuáles son las principales propiedades de los modelos de calidad del software que han sido descritas en la literatura?</p>	<p>Obtener artículos en los que se distinga la propiedad de solapamiento. Además de distinguir a qué tipo y subtipo pertenece cada factor estudiado.</p>
<p>P4. ¿Cuáles son las aplicaciones para las cuales han sido propuestos los modelos de calidad de software?</p>	<p>Evaluar la importancia de la aplicación de la calidad del software por medio de la definición de los diferentes aspectos a los que da soporte.</p>
<p>P5. ¿Qué métodos se han utilizado para su construcción?</p>	<p>Distinguir los métodos aplicados para construir los modelos de calidad de software mixtos y a la medida.</p>
<p>P6. ¿Qué tipo de calidad evalúan interna, externa, en uso y qué elementos se relacionan a estos tipos?</p>	<p>Clasificar los distintos modelos de calidad de acuerdo a los factores que evalúan.</p>

Fuente: Trabajo de Titulación

Elaborado: Ana Villalta

4.5 Protocolo

Una vez definido el enfoque del estudio se elaboró un protocolo que ayudará a que el trabajo no se distancie del objetivo planteado.

Los conceptos de cada uno de los aspectos descritos en esta sección se pueden encontrar en la sección 2.5.1.

Se siguieron, paso a paso, las guías planteadas para esta etapa, las cuales se detallan a continuación.

4.5.1 Antecedentes.

En el año 2007 se realizó un catálogo de atributos de calidad [DO07]. Se identificó 448 atributos técnicos y 255 atributos no técnicos utilizando la técnica de mapeo de artículos, la cual se aplicó de manera semiformal. Se planteó actualizar esta construcción utilizando SLR como técnica de investigación empírica, formalizando la búsqueda de literatura. El objetivo fue valorar las publicaciones correspondientes al período 2007-2014, en el área de Calidad de Software, con un enfoque en sus modelos y en los distintos elementos y propiedades que estos poseen.

4.5.2 Preguntas de investigación.

P1. ¿Cuáles son los elementos que estructuran los modelos de calidad de software?

P2. ¿Cómo se relacionan los distintos elementos entre sí?

P3. ¿Cuáles son las principales propiedades de los modelos de calidad del software que han sido descritas en la literatura?

P4. ¿Cuáles son las aplicaciones para las cuales han sido propuestos los modelos de calidad de software?

P5. ¿Qué métodos se han utilizado para su construcción?

P6. ¿Qué tipo de calidad evalúan, interna, externa, en uso y qué elementos se relacionan a estos tipos?

4.5.3 Estrategia de Búsqueda

Para llegar a los resultados requeridos y obtener los artículos adecuados, se definieron estrategias a seguir. Éstas fueron:

1) Identificar las palabras para formar cadenas de búsqueda (términos).

Para comenzar la estrategia, se realizó una selección de términos alternativos partiendo de palabras clave o términos importantes obtenidos de la desintegración de las preguntas de investigación (ver Tabla 4.2).

Tabla 4.2. Términos de búsqueda a usar en las diferentes Bases de Datos

Términos Importantes	Términos Alternativos
Models	
Quality	
Software	
Product	
Elements	Characteristic OR Attributes OR Factors
Structure	Hierarchy OR Decomposition OR Levels OR Catalog
Properties	
Applications	Use
Metrics	

Fuente: Trabajo de Titulación

Elaborado: Ana Villalta

2) Definir los recursos en los cuales buscar (Bases de datos, revistas, actas de congreso, etc.)

El proceso de búsqueda se realizó de manera automática en bases de datos digitales específicas. Se consultó con el Ing. Juan Pablo Carvallo Vega, PhD (recurso primario), como experto en el área para comenzar con este proceso. Las bases seleccionadas fueron (ver Tabla 4.3):

Tabla 4.3. Fuentes de búsqueda digitales seleccionadas

Fuente	Responsable
ACM Digital Library	Ana Villalta
IEEE	Ana Villalta
Springer	Ana Villalta
Elsiver	Ana Villalta
ISI Web of Knowledge	Ana Villalta

Fuente: Trabajo de Titulación

Elaborado: Ana Villalta

Se describe a continuación otros recursos de búsqueda (ver Tabla 4.4). Para llegar a clasificar otras fuentes se utilizaron dos estrategias.

- Mapeo en las diferentes bases de datos.
- Consulta con el experto en el área.

El proceso de búsqueda en estas "Otras fuentes" se realizó de forma automática

Tabla 4.4. Otras fuentes (revistas) en las cuales buscar los artículos

Fuente	Responsable
Software Quality Journal	Ana Villalta
Empirical Software Engineering	Ana Villalta
Innovations in Systems and Software Engineering	Ana Villalta

Elaborado: Ana Villalta

3) Establecer los pasos a seguir para que se dé una correcta iteración entre los términos definidos y los recursos a utilizar.

Descritos los términos a utilizar y establecidos los recursos (BD), se definieron las acciones a realizar por el investigador. Estas fueron:

La construcción de una prueba piloto que sirvió como línea base para:

- El refinamiento de cadenas posteriores
- Estudio de cómo trabajan los motores de búsqueda de las diferentes bases de datos
- Estudio de los diferentes tipos de refinamientos que ofrecen las bases de datos

La aplicación de refinamientos necesarios para conseguir que el número y la calidad de artículos resultasen los óptimos para el estudio. Los refinamientos fueron ejecutados en base a la experiencia obtenida al realizar la prueba piloto

4.5.4 Criterios de Selección de estudios.

Los criterios sirvieron para definir qué artículos serían seleccionados y cuáles no. Para esto se elaboró una lista a seguir en la etapa de desarrollo de la SLR y se puede observar a continuación:

Criterios de inclusión.- Todos los artículos publicados entre el 2007 y el 2014 que contengan:

- En el título del documento los términos incluidos en las búsquedas
- Si no se encuentran los términos en el título del documento estos serán buscados en los metadatos.
- Artículos que respondan a los refinamientos automáticos que se escoja para cada base de datos.

Criterios de exclusión. Quedan excluidos los documentos que:

- Previo no haber respondido en el título del documento a los términos buscados no respondan en el resumen.

- Tras haber sido seleccionados por un investigador, no respondan a la selección del segundo investigador.
- En su lectura no respondan a las preguntas de búsqueda.
- Si se encuentran artículos repetidos en las distintas bases de datos seleccionadas, solo se tomará de uno de ellos.

4.5.5 Procedimientos de Selección de estudios.

En esta etapa se delimitó la iteración con las diferentes bases de datos y con los artículos seleccionados. Además se definió la manera de sintetizar los resultados. Los procedimientos utilizados fueron:

Criterios de selección:

- Se utilizarán tres de las cinco bases de datos descritas. Estas serán: Springer, IEEE, ACM.
- En SPRINGER se seleccionarán los artículos que respondan a las preguntas de investigación en el título del documento.
- En IEEE se seleccionarán los artículos que respondan a las preguntas de investigación en su resumen y conclusiones, aun si su título no llevase las palabras clave.
- En ACM se dará lectura a todos los resúmenes de los artículos tomando en cuenta aquellos que también respondan a la selección en el título del documento.

Los resultados deberán ser tabulados de la siguiente manera:

- Actividad de investigación que ha habido en los años definidos.
- Artículos que contengan factores de calidad.
- Artículos que contengan métricas de calidad.
- Artículos que contengan la propiedad de solapamiento.
- Artículos que contengan métodos de construcción de modelos de calidad además de contar con factores de calidad o métricas.

Proceso para la selección de estudios

- La primera selección de estudios será realizada por un solo investigador (Ana Villalta).
- Los estudios rechazados serán evaluados por un segundo investigador, al igual que la segunda selección de estudios (Ing. Juan Pablo Carvallo Vega).
- La segunda selección de estudios se realizará sobre los artículos seleccionados en una primera instancia.

4.5.6 Listas de verificación y procedimientos de evaluación.

Al elaborar una lista de verificación, todo el proceso llevado a cabo en la SLR podrá ser evaluado. En esta sección se definió la puntuación para cada criterio de dicha lista.

Esta verificación no será llevada a cabo, pero quedará establecida en el protocolo.

Los artículos seleccionados se evaluarán utilizando los siguientes criterios.

- Criterio 1: ¿Cada artículo seleccionado atiende a los criterios de inclusión – exclusión?
- Criterio 2: ¿En la búsqueda realizada que tan altas son las posibilidades de haber cubierto toda la literatura relevante?
- Criterio 3: ¿Existe un revisor que evalúe la calidad de los estudios incluidos?
- Criterio 4: ¿Los estudios atienden a las preguntas de investigación?

Las preguntas de los diferentes criterios se calificarán de la siguiente manera.

- Criterio 1: S (sí), los criterios de inclusión están presentes en el artículo; P (parcialmente), los criterios de inclusión no son claramente definidos; N (no), los criterios de inclusión no están presentes en el artículo.

- Criterio 2: S, los autores han buscado ya sea 4 o más bibliotecas digitales e incluyó estrategias de búsqueda adicionales y referencia todas las revistas que abordan el tema de interés; P, los autores han buscado 3 o 4 bibliotecas digitales e incluyeron estrategias de búsqueda adicionales y referencia todas las revistas que abordan el tema de interés; N, los autores buscan hasta en 2 bibliotecas digitales o un conjunto muy restringido de revistas.
- Criterio 3: S, existe un segundo revisor; N, los artículos han sido seleccionados por un solo revisor.
- Criterio 4: S, cada trabajo responde al menos a una pregunta de investigación ampliamente; P, el trabajo responde a una pregunta de investigación resumidamente; N, no se responde a ninguna pregunta de investigación.

El procedimiento de puntuación es S = 1, P = 0,5 y N o Desconocido = 0.

4.5.7 Estrategia de extracción de datos.

Los datos extraídos de cada artículo serán:

- Título del documento
- La fuente (la conferencia o revista)
- El año en que se publicó el documento
- Clasificación del trabajo
- El autor (s)
- Resumen

Cada trabajo será leído y clasificado entre:

- Positivo
- Negativo
- Falso

4.5.8 Síntesis de los datos extraídos.

Para la síntesis de los datos no se utilizará un meta – análisis debido a los tiempos planteados para realizar la revisión sistemática. Se utilizarán técnicas informales con el propósito de mostrar resultados.

4.5.9 Estrategia de difusión.

Los resultados de esta investigación serán presentados como tesis de pre grado y se espera que el trabajo sea de interés para los estudiantes de ingeniería de sistemas de la Universidad del Azuay, así como para docentes que soliciten su revisión.

Se planteó realizar una primera versión de un artículo en el que se redacten las lecciones aprendidas de la experiencia al realizar una SLR, como investigadora novata. (Anexo: Lecciones aprendidas)



CAPÍTULO 5. PUESTA EN MARCHA



5.1 Introducción

En el capítulo 4, se describió la planeación de la SLR y se detalló el protocolo a seguir. El siguiente paso fue desarrollar la SLR que constituyó la puesta en marcha de la investigación.

Para lograr el correcto desarrollo se plantearon ciertas estrategias de búsqueda que fueron guiadas por una prueba piloto en diferentes bases de datos digitales. Las estrategias sirvieron para evaluar el volumen de los estudios relevantes, utilizar varias combinaciones de términos y generar cadenas de búsqueda óptimas [K07].

A continuación se detalla cómo se seleccionó y evaluó la literatura, tomando en cuenta la selección por título, resumen y/o conclusiones de los artículos. Este proceso estuvo guiado por los criterios de inclusión y exclusión definidos en la planeación. Posteriormente se extrajeron los datos necesarios para lograr la síntesis de contenidos.

A pesar de que el desarrollo es una etapa de naturaleza iterativa, las siguientes secciones se describen siguiendo un orden específico.

5.2 Generación de una estrategia de búsqueda

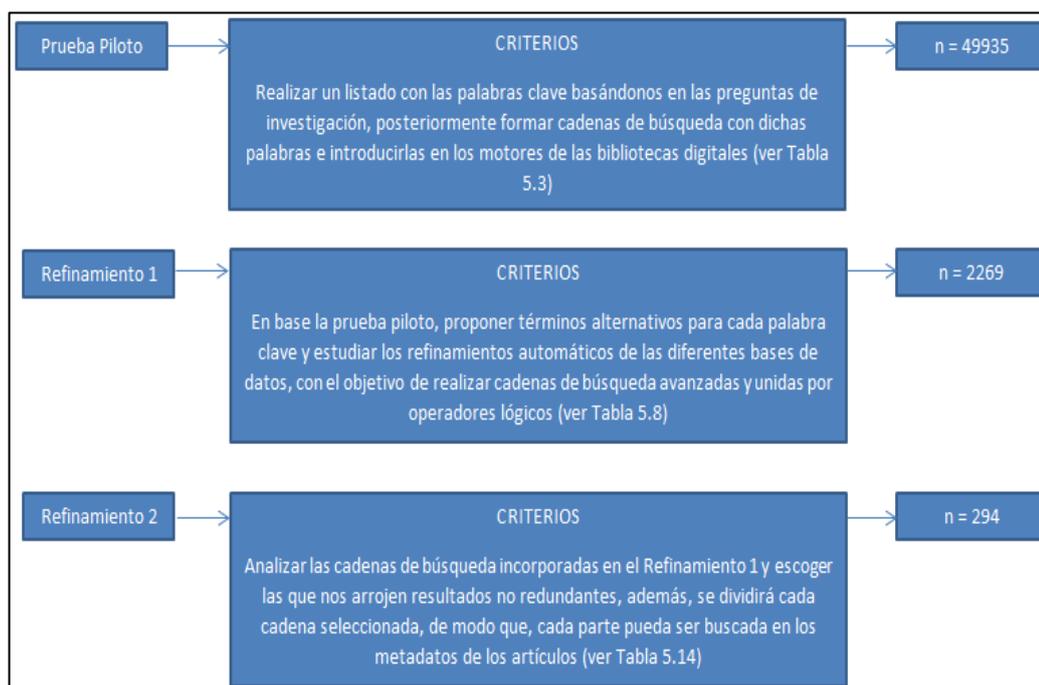
Como se especificó en la introducción, para desarrollar las estrategias de búsqueda se utilizó la planeación de la estrategia descrita en el apartado 4.4.3 del capítulo 4. Se buscó en tres bibliotecas digitales: IEEE, SPRINGER y ACM. Las búsquedas fueron automáticas, mientras que la clasificación de los artículos se realizó de forma manual.

La Figura 5.1 muestra las etapas de búsqueda y el número de artículos encontrados en la base de datos IEEE, que es de donde se sintetizaron los artículos. Sin embargo, en todas las bases de datos se realizaron las mismas actividades y se aplicaron los mismos criterios:

Criterios de inclusión: Todos los artículos publicados entre el 2007 al 2014 que contengan:

- En el título del documento los términos incluidos en las búsquedas.
- Si no se encuentran los términos en el título del documento, estos serán buscados en los metadatos.
- Artículos que respondan a los refinamientos automáticos que se escoja para cada Base de datos.

Figura 5.1 Proceso de Búsqueda de artículos en IEEE



Fuente: Trabajo de Titulación

Elaborado: Ana Villalta

Las siguientes secciones detallan la prueba piloto, que sirvió como aporte principal para la generación de nuevas estrategias, y dos refinamientos realizados a partir de esta prueba piloto. Se tomó como base la experiencia práctica obtenida al interactuar con las bases de datos registradas.

Al ser una SLR un proceso iterativo, se amplió el protocolo y se definieron dos tácticas de búsqueda que fueron utilizadas en el estudio:

- Búsquedas de prueba utilizando diversas combinaciones de términos, derivados de la pregunta de investigación.
- Consulta con expertos.

Al concluir esta etapa se obtuvo el refinamiento de la cantidad y calidad de la literatura que sirvió como base para las posteriores secciones.

5.2.1 Prueba Piloto (Primera Etapa)

Se definió una subestrategia denominada "Estrategia para la selección de información: Primera Etapa", la cual contempló las siguientes actividades:

a. Realizar un listado de las principales bases de datos que incorporan artículos relacionados con el objetivo a tratar. Se identificaron las siguientes bases de datos bibliográficas para el estudio:

- **IEEE:** <http://www.ieee.org/index.html>
- **Springer:** <http://link.springer.com/>
- **ACM:** <http://dl.acm.org/>

b. Realizar un listado con palabras clave acorde a las preguntas de investigación. Para la identificación de las palabras clave (términos) se consideraron los sustantivos incluidos en cada pregunta de investigación. Las palabras clave identificadas se presentan en la Tabla 5.1:

Tabla 5.1. Palabras clave identificadas en relación a las preguntas

Pregunta de Investigación	Palabras Clave
P1	Modelos de calidad"
P2	Elementos
P3	Propiedades
P4	Aplicaciones
P1	Características
P1	Sub-características
P1	Atributos
P1	Software
P3	Métricas
P1	Producto

c. Armar cadenas de búsqueda con las palabras clave obtenidas.

A continuación se detallan las cadenas formadas por cada palabra definida en el punto anterior (ver Tabla 5.2).

Tabla 5.2. Cadenas derivadas de las palabras clave de la Tabla 5.1

Palabras	Cadenas
ELEMETS	Cadena1.Software quality models elements Cadena2.Software quality elements Cadena3.Software product quality elements
PROPERTIES	Cadena1.Software quality models properties Cadena2.Software quality properties Cadena3.Software product quality properties
APPLICATIONS	Cadena1.Software quality models applications Cadena2.Software quality applications Cadena3.Software product quality applications
CHARACTERISTICS	Cadena1.Software quality models characteristics Cadena2.Software quality characteristics Cadena3.Software product quality characteristics
SUBCHARACTERISTICS	Cadena1.Software quality models subcharacteristics Cadena2.Software quality subcharacteristics Cadena3.Software product quality subcharacteristics
ATRIBUTES	Cadena1.Software quality models attributes Cadena2.Software quality attributes Cadena3.Software product quality attributes
METRICS	Cadena1.Software quality models metrics Cadena2.Software quality metrics Cadena3.Software product quality metrics

Se obtuvo una tabla piloto que recopila el número de artículos por frase y por base de datos (ver Tabla 5.3).

Tabla 5.3. Número de artículos encontrados por cadena de búsqueda en la Prueba Piloto

Palabra2	Palabra3	Palabra4	IEEE	ACM	Springe	Total
Software	Elementos		598	34532 991	160.829	196.950
Software	Elementos		1248	37243 1212	188294	227997
Producto	Software	Elementos	206	21593 846	104985	127630
Software	Propiedades		729	32227 772	161013	194741
Software	Propiedades		1560	34824 944	190150	227478
Producto	Software	Propiedades	165	19066 642	103715	123588
Software	Aplicaciones		5218		248170	
Software	Aplicaciones		14019		300480	
Producto	Software	Aplicaciones	1425		156898	
Software	Métodos		3987		297798	
Software	Métodos		9355		378828	
Producto	Software	Métodos	1162		188245	
Software	Características		974	35748 772	196956	234450
Software	Características		2209	38261 970	242343	283783
Producto	Software	Características	316	21516 657	124157	146646
Software	Subcaracterísticas		10	40 1	108	159
Software	Subcaracterísticas		12	40 1	110	163
Producto	Software	Subcaracterísticas	7	38 1	102	148
Software	Atributos		658	20464 571	110097	131790
Software	Atributos		1319	21503 671	129284	152777
Producto	Software	Atributos	238	13274 510	73172	87194
Software	Métricas		1262	13789 332	56193	71576
Software	Métricas		2709	14871 423	62771	80774
Producto	Software	Métricas	549	8274 289	33303	42415

Fuente: Trabajo de titulación (Anexo Refinamientos, Prueba Piloto)

Elaborado: Ana Villalta

Como resultado se obtuvieron 21 cadenas de búsqueda ingresadas en tres bases de datos distintas. Esta prueba sirvió como aporte para refinamientos posteriores debido a que la cantidad de literatura encontrada fue excesiva y la estructura de las cadenas fue muy general.

5.2.2 Segunda Etapa (Refinamiento 1)

Se tomó la prueba piloto como base y se planteó una segunda sub estrategia que se denominó "Estrategias para la selección de información: Segunda Etapa". Se contemplaron las siguientes actividades:

- a. Realizar un refinamiento de las palabras clave proponiendo términos alternativos. (ver tabla5.4).

Tabla 5.4. Términos de búsqueda alternativos

Términos Importantes	Términos Alternativos
Models	
Quality	
Software	
Product	
Elements	Characteristic OR Attributes OR Factors
Structure	Hierarchy OR Decomposition OR Levels OR Catalog
Properties	
Applications	Use
Metrics	

Fuente: Trabajo de titulación

Elaborado: Ana Villalta

b. Estudiar los tipos de refinamientos (filtros) automáticos y operadores lógicos que proporcionan las distintas bases de datos elegidas. Para lograr obtener esta información se exploró las bibliotecas digitales, (IEEE, SPRINGER, ACM), y se extrajeron los conceptos de cada operador lógico, como se detalla a continuación (ver Tabla 5.5):

Tabla 5.5. Bases de datos y sus operadores lógicos

Base de datos	Operadores
Springer	<ul style="list-style-type: none"> • OR. Las referencias seleccionadas contendrán por lo menos un término de los introducidos. • NOT. Excluye los términos después del NOT ninguna referencia seleccionada podrá contenerlos. • AND. Cada referencia seleccionada va a contener todos los términos combinados con AND • NEAR. El término de búsqueda de la izquierda es del plazo de diez palabras de la palabra a la derecha del operador NEAR • ONNEAR. Los términos de búsqueda deben estar cerca uno del otro en el texto y en el orden en que se ha introducido en el cuadro de búsqueda. • EI *. Es usado como un comodín para buscar palabras que contengan los caracteres antes del *, es mejor su uso cuando por lo menos hay tres caracteres antes del comodín. • EI ?. Es utilizado como sustituto de una sola letra. • Varios operadores incluidos en una misma frase son interpretados con el orden de preferencia NOT, OR, AND. Si se quiere evaluar una frase completa sin operadores, esta debe ser usada entre comillas
IEEE	<ul style="list-style-type: none"> • AND. Busca registros que cumplan las dos condiciones que une el operador. • OR. Busca registros que cumplan una o ambas condiciones que une el operador. • NOT. Busca registros que no cumplan la condición especificada.
ACM	<ul style="list-style-type: none"> • Iguales a los de Springer.

En adición a los operadores descritos en la tabla 5.5, las bases de datos proveen opciones de filtrado que permiten refinar las búsquedas y limitar los resultados generados. Las figuras 5.2 y 5.3 presentan las opciones de refinamiento de las bases de datos Springer e IEEE. Para el caso de ACM se utilizaron las opciones de búsquedas avanzadas, ver figura 5.4.

Figura 5.2 Opciones para Realizar Refinamientos Automáticos en SPRINGER

Content Type	
Chapter	161
Article	45
Book	1
Reference Work Entry	1
Discipline	
see all	
Computer Science	21
Business & Management	26
Engineering	21
Mathematics	3
Physics	2
Subdiscipline	
see all	
SWE	155
Database Management & Information Retrieval	89
Communication Networks	54
Theoretical Computer Science	48
Information Systems and Applications	46

Published In	
see all	
Software Quality Journal	21
Software Quality. Model-Based Approaches for Advanced Software and Systems Engineering	16
Current Trends in Web Engineering	8
Empirical Software Engineering	8
Product-Focused Software Process Improvement	7
Language	
English	

Fuente: Base de datos SPRINGER

Figura 5.3 Opciones de refinamientos automáticos en IEEE

Publisher		
Return Results from		
<input type="checkbox"/> IEEE(2,585,002)	<input type="checkbox"/> Wiley-IEEE Press(11,091)	
<input type="checkbox"/> AIP(253,858)	<input type="checkbox"/> IBM(6,022)	
<input type="checkbox"/> IET(194,385)	<input type="checkbox"/> VDE(3,260)	
<input type="checkbox"/> AVS(35,231)		
Content Types		
<input type="checkbox"/> Conference Publications (1,961,660)	<input type="checkbox"/> Early Access Articles (5,996)	
<input type="checkbox"/> Journals & Magazines (1,105,566)	<input type="checkbox"/> Standards (4,568)	
<input type="checkbox"/> Books & eBooks (11,193)	<input type="checkbox"/> Education & Learning (273)	
Topics		
<input type="checkbox"/> Computing & Processing (Hardware/Software) (1,443,864)	<input type="checkbox"/> Fields, Waves & Electromagnetics (679,875)	<input type="checkbox"/> Transportation (198,591)
<input type="checkbox"/> Components, Circuits, Devices & Systems (1,421,258)	<input type="checkbox"/> Photonics & Electro-Optics (627,646)	<input type="checkbox"/> Geoscience (197,800)
<input type="checkbox"/> Communication, Networking & Broadcasting (995,348)	<input type="checkbox"/> General Topics for Engineers (Math, Science & Engineering) (607,848)	<input type="checkbox"/> Nuclear Engineering (191,117)
<input type="checkbox"/> Engineered Materials, Dielectrics & Plasmas (870,104)	<input type="checkbox"/> Bioengineering (577,860)	<input type="checkbox"/> Engineering Profession (157,717)
<input type="checkbox"/> Signal Processing & Analysis (867,771)	<input type="checkbox"/> Robotics & Control Systems (436,450)	
<input type="checkbox"/> Power, Energy, & Industry Applications (805,958)	<input type="checkbox"/> Aerospace (399,055)	
Publication Year		
<input type="radio"/> Search latest content update (12/10/2011)		
<input type="radio"/> Specify Year Range From: <input type="text" value="All"/> To: <input type="text" value="Present"/>		
<input checked="" type="radio"/> All Available Years		

Fuente: Base de datos IEEE

Figura 5.4 Opciones de búsqueda avanzada de ACM.

The image shows a screenshot of the ACM advanced search interface. It is organized into several sections, each with a yellow background header:

- Words or Phrases:** Includes a dropdown menu for "Find [any field] with" and three input fields for "all of this text (and)", "any of this text (or)", and "none of this text (not)".
- Names:** Includes a dropdown menu for "Find [any field] with names" and a radio button group for "using" with options "all", "any", and "none of the names".
- Keywords:** Includes an input field for "Find author's keywords" and a radio button group for "using" with options "all", "any", and "none of the keywords".
- Affiliations:** Includes an input field for "Find company or school" and a radio button group for "using" with options "all", "any", and "none of the affiliations".
- Publication:** Includes an input field for "Find publication" and a radio button group for "using" with options "all", "any", and "none of the names". It also includes an input field for "Find publisher" and a radio button group for "using" with options "any" and "none of the names". Below this are two dropdown menus for "Published since [year]" and "Published before [year]".
- In publication types:** A row of checkboxes for "Journal", "Proceeding", "Transaction", "Magazine", and "Newsletter".
- Conference:** Includes an input field for "Find sponsor names" and a radio button group for "using" with options "all", "any", and "none of the names". It also includes an input field for "Find location" and a radio button group for "using" with options "any" and "none of the locations". Below this is an input field for "Find year (yyyy)" and a radio button group for "using" with options "any" and "none of the years".
- Identification codes:** Includes an input field for "Find ISBN/ISSN" and an input field for "Find DOI".
- Computing Classification System (CCS):** Includes an input field for "Find node" and an input field for "Find subject/houn". Below this is a checkbox for "Look at primary category only".
- Required components:** Includes a radio button group for "Results must have" with options "Full Text", "Abstract", and "Review".

A blue "SEARCH" button is located in the bottom right corner of the form.

- c. Estudiar las reglas de los diferentes operadores lógicos que, en conjunto con las palabras clave, puedan convertirse en instrucciones interpretables para los buscadores de las bases de datos.

La estructura de las cadenas de búsqueda fue distinta para cada Base de datos debido a que cada motor trabaja de forma diferente y a que los resultados surgieron alterados al ingresar exactamente la misma cadena en cada Biblioteca.

Se realizaron varias pruebas de búsqueda. En las pruebas que no arrojaron resultados con frases precisas, se efectuaron búsquedas en los metadatos (título, resumen). También se ejecutaron pruebas utilizando las búsquedas avanzadas que ofrecen las bibliotecas seleccionadas hasta lograr un refinamiento adecuado para continuar con la siguiente etapa. Las cadenas obtenidas se detallan a continuación (ver Tabla 5.6):

Tabla 5.6. Cadenas obtenidas en la etapa de Refinamiento 1

Base de datos	Cadenas
SPRINGER	<p>Elements</p> <p>“Software quality models”</p> <p>“Software quality models” AND elements</p> <p>“Software quality” AND elements</p> <p>“Software product quality” AND elements</p> <p>“Software quality models” and (attributes or characteristics or factors)</p> <p>“Software quality” and (characteristics or attributes or factors)</p> <p>“Software product quality” and (characteristics or attributes or factors)</p> <p>Properties</p> <p>8. “Software quality models” and properties</p> <p>9. “Software quality” and properties</p> <p>10. “Software product quality” and properties</p> <p>Applications</p> <p>11. “Software quality models” and applications</p> <p>12. “Software quality” and applications</p> <p>13. “Software product quality” and applications</p> <p>Metrics</p> <p>14. “Software quality models” and metrics</p> <p>15. “Software quality” and metrics</p> <p>16. “Software product quality” and metrics</p>
IEEE	<p>Elements</p> <p>“Software quality models”</p> <p>(Software quality models) AND elements</p> <p>(Software quality) AND elements</p> <p>(Software product quality) AND elements</p> <p>“Document Title”: Software quality models and (attributes or characteristics or factors)</p> <p>“Document Title”: Software quality and (attributes or characteristics or factors)</p> <p>“Document Title”: Software product quality and (attributes or characteristics or factors)</p>

	Properties
	(Software quality models) and "Document Title": Properties
	(Software quality) and "Document Title": Properties
	(Software product quality) and "Document Title": Properties
	Applications
	("Document Title": Software quality models") and "Document Title": Applications
	("Document Title": Software quality") and "Document Title": Applications
	("Abstract": Software product quality") and "Document Title": Applications
	Metrics
	("Abstract": Software quality models) and "Document Title": metrics
	("Document Title": Software quality) and "Document Title": metrics
	("Abstract": Software product quality) and "Document Title": metrics
ACM	Elements
	"Software quality models"
	"Software quality models" AND elements
	"Software quality" AND elements
	"Software product quality" AND elements
	"Software quality models" and (attributes or characteristics or factors)
	"Software quality" and (attributes or characteristics or factors)
	"Software product quality" and (attributes or characteristics or factors)
	Properties
	"Software quality models" and Properties
	"Software quality" and Properties
	"Software product quality" and Properties
	Applications
	"Software quality models" and Applications
	"Software quality" and Applications

	"Software product quality" and Applications
	Metrics
	"Software quality models and metrics
	"Software quality and metrics
	"Software product quality and metrics

Como resultado se obtuvo tablas separadas por cada base de datos. En las tablas se muestra el total de artículos encontrados por frase y los tipos de refinamientos realizados (ver Tabla 5.7, 5.8 y 5.9).

Tabla 5.7. Número de Artículos encontrados: Primer Refinamiento base de datos SPRINGER

Palabra1	Palabra2	Palabra3	Palabra4	Total	Content Type	Discipline	SubDiscipline	Published In	Language	Año
"Software quality models"				206	Chapter, Article	Computer Science			English	2007-2014
"Software quality models"	Elements			128	Chapter, Article	Computer Science			English	2007-2014
"Software quality"	Elements			515	Article	Computer Science			English	2007-2014
"Software product quality"	Elements			193	Chapter, Article	Computer Science			English	2007-2014
"Software quality models"	Attributes	Characteristics	Factors	43	Article	Computer Science			English	2007-2014
"Software quality"	Attributes	Characteristics	Factors	726	Article	Computer Science			English	2007-2014
"Software product quality"	Attributes	Characteristics	Factors	43	Article	Computer Science			English	2007-2014
"Software quality models"	Properties			25	Article	Computer Science			English	2007-2014
"Software quality"	Properties			486	Article	Computer Science			English	2007-2014
"Software product quality"	Properties			28	Article	Computer Science			English	2007-2014
"Software quality models"	Applications			42	Article	Computer Science			English	2007-2014
"Software quality"	Applications			735	Article	Computer Science			English	2007-2014
"Software product quality"	Applications			42	Article	Computer Science			English	2007-2014
"Software quality models"	Metrics			31	Article	Computer Science			English	2007-2014
"Software quality"	Metrics			488	Article	Computer Science			English	2007-2014
"Software product quality"	Metrics			35	Article	Computer Science			English	2007-2014

Fuente: Trabajo de Titulación (Anexo Refinamientos, Refinamiento de Cadenas-Etapa1)

Elaborado: Ana Villalta

Tabla 5.8. Número de Artículos encontrados: Primer Refinamiento base de datos IEEE

IEEE									
Palabra1	Palabra2	Palabra3	Palabra4	Total	Content Type	Language	Año	En	En
Software quality models				37	Conference Publications, Journals and Magazines	English	2007-2014	Metadata Only	
Software quality models	Elements			388	Conference Publications, Journals and Magazines	English	2007-2014	Metadata Only	
Software quality	Elements			719	Conference Publications, Journals and Magazines	English	2007-2014	Metadata Only	
Software product quality	Elements			100	Conference Publications, Journals and Magazines	English	2007-2014	Metadata Only	
Software quality models	Attributes	Characteristics	Factors	223	Conference Publications, Journals and Magazines	English	2007-2014	Metadata Only	Document Title
Software quality	Attributes	Characteristics	Factors	411	Conference Publications, Journals and Magazines	English	2007-2014	Metadata Only	Document Title
Software product quality	Attributes	Characteristics	Factors	88	Conference Publications, Journals and Magazines	English	2007-2014	Metadata Only	Document Title
Software quality models	Properties			31	Conference Publications, Journals and Magazines	English	2007-2014	Metadata Only	Document Title
Software quality	Properties			59	Conference Publications, Journals and Magazines	English	2007-2014	Metadata Only	Document Title
Software product quality	Properties			6	Conference Publications, Journals and Magazines	English	2007-2014	Metadata Only	Document Title
Software quality models	Applications			12	Conference Publications, Journals and Magazines	English	2007-2014	Document Title	Document Title
Software quality	Applications			28	Conference Publications, Journals and Magazines	English	2007-2014	Document Title	Document Title
Software product quality	Applications			51	Conference Publications, Journals and Magazines	English	2007-2014	Abstract	Document Title
Software quality models	Metrics			47	Conference Publications, Journals and Magazines	English	2007-2014	Abstract	Document Title
Software quality	Metrics			28	Conference Publications, Journals and Magazines	English	2007-2014	Document Title	Document Title
Software product quality	Metrics			41	Conference Publications, Journals and Magazines	English	2007-2014	Abstract	Document Title

Fuente: Trabajo de Titulación (Anexo Refinamientos, Refinamiento de Cadenas-Etapa1)

Elaborado: Ana Villalta

Tabla 5.9. Número de Artículos encontrados: Primer Refinamiento base de datos ACM

ACM								
Palabra1	Palabra2	Palabra3	Palabra4	Total	Año	Word or Phrases	Publication types	Results must have
Software quality models				247	2007-2014			
Software quality models	Elements			120	2007-2014			
Software quality	Elements			3604	2007-2014			
Software product quality	Elements			269	2007-2014			
Software quality models	Attributes	Characteristics	Factors	222	2007-2014			
Software quality	Attributes	Characteristics	Factors	5580	2007-2014			
Software product quality	Attributes	Characteristics	Factors	417	2007-2014			
Software quality models	Properties			115	2007-2014			
Software quality	Properties			3220	2007-2014			
Software product quality	Properties			241	2007-2014			
Software quality models	Applications			214	2007-2014			
Software quality	Applications			5969	2007-2014			
Software product quality	Applications			398	2007-2014			
Software quality models	Metrics			188	2007-2014			
Software quality	Metrics			3611	2007-2014			
Software product quality	Metrics			308	2007-2014			

Fuente: Trabajo de Titulación (Anexo Refinamientos, Refinamiento de Cadenas-Etapa1)

Elaborado: Ana Villalta

5.2.3 Tercera Etapa (Refinamiento 2)

Este trabajo pretende documentar los datos clave de toda la literatura encontrada pero propone la lectura completa únicamente de 50 artículos de la base de datos digital IEEE. Se esperaba alcanzar este número de artículos en la segunda etapa de refinamiento, sin embargo no se consiguió una reducción sustancial por lo que se planteó una tercera etapa.

En esta etapa se discriminaron varias cadenas de búsqueda utilizadas en la segunda etapa (Refinamiento 1). El detalle de este proceso se muestra a continuación:

- **SPRINGER**

En las diferentes bases de datos se utilizaron búsquedas avanzadas. En este tipo de búsqueda las comillas dobles "" indican al gestor que los resultados deben llevar la frase exacta que se encuentra entre dichas comillas y que las palabras que se encuentran luego las comillas "", pueden ser cualquiera. Por lo tanto se determinó que las cadenas:

“Software quality” AND elements

“Software quality” and (characteristics or attributes or factors)

“Software quality” and properties

“Software quality” and applications

“Software quality” and metrics

Engloban a las cadenas restantes y que las referencias arrojadas contienen las palabras que se determinen después del operador AND.

Se descartaron las siguientes cadenas:

“Software quality models”

“Software quality models” AND elements

“Software product quality” AND elements

“Software quality models” and (attributes or characteristics or factors)

“Software product quality” and (characteristics or attributes or factors)

“Software quality models” and properties

“Software product quality” and properties
“Software quality models” and applications
“Software product quality” and applications
“Software quality models” and metrics
“Software product quality” and metrics

- **IEEE**

En esta base de datos se utilizó la búsqueda avanzada y se incluyó el operador AND en todos los casos de cadenas formadas. No se incorporaron las comillas dobles "" ya que los resultados arrojados, sin ser discriminados por frase exacta, lanzaron un número importante y no excesivo de artículos significativos que podrían ser utilizados en una segunda lectura, a futuro. Las "" serán utilizadas en otra etapa de refinamiento.

Por otro lado se realizaron pruebas buscando parte de las cadenas en el “Document Title” o “Abstract” de tal forma que las derivaciones reflejen un mayor número de artículos a ser estudiados. Las búsquedas que devolvieron el mayor número de artículos englobando al resto de cadenas fueron:

Software quality AND elements
Software quality and (characteristics or attributes or factors)
Software quality and properties
Software quality and applications
Software quality and metrics

Se descartaron las cadenas:

“Software quality models”
Software quality models AND elements
Software product quality AND elements
Software quality models and (attributes or characteristics or factors)
Software product quality and (characteristics or attributes or factors)

Software quality models and properties
Software product quality and properties
Software quality models and applications
Software product quality and applications
Software quality models and metrics
Software product quality and metrics

- **ACM**

Se utilizaron los mismos criterios que en la base de datos SPRINGER, de esta manera las cadenas seleccionadas para la segunda etapa de refinamiento fueron:

“Software quality” AND elements
“Software quality” and (characteristics or attributes or factors)
“Software quality” and properties
“Software quality” and applications
“Software quality” and metrics

Se descartaron las cadenas:

“Software quality models”
“Software quality models” AND elements
“Software product quality” AND elements
“Software quality models” and (attributes or characteristics or factors)
“Software product quality” and (characteristics or attributes or factors)
“Software quality models” and properties
“Software product quality” and properties
“Software quality models” and applications
“Software product quality” and applications
“Software quality models” and metrics
“Software product quality” and metrics

Una vez limitado el número de cadenas, se procedió a refinar las cadenas seleccionadas utilizando los siguientes criterios:

- **SPRINGER**

En esta base de datos el refinamiento se realizó por revista. Por esta razón se dividió la búsqueda por cada cadena en tres etapas haciendo referencia a las tres revistas definidas en el protocolo.

- Software Quality Journal
- Empirical Software Engineering
- Innovations in Systems And Software Engineering

También se realizó el refinamiento de este grupo de cadenas por criterios comunes tales como: el tipo de Contenido, disciplina, subdisciplina, lugar de publicación, lenguaje y año (ver Tabla 5.14).

Por otra parte se incorporaron las "" para restringir cada cadena tal como se muestra a continuación (ver Tabla 5.10):

Tabla 5.10. Cadenas utilizadas en el Refinamiento 2 SPRINGER

Palabra	Cadenas
Elements	"Software quality" AND elements "Software quality models" and (characteristics or attributes or factors)
Properties	"Software quality" and properties
Applications	"Software quality" and applications
Metrics	"Software quality" and metrics

- **IEEE**

Los refinamientos automáticos comunes que fueron realizados para esta base de datos se pueden observar en la Tabla 5.14 y las nuevas cadenas utilizadas fueron (ver Tabla 5.11):

Tabla 5.11. Cadenas utilizadas en el Refinamiento 2 IEEE

Palabra	Cadena
Elements	"Software quality" AND elements ("Document Title":"Software quality") AND (Attributes OR Characteristics OR Factors)
Properties	((("Software quality") AND "Document Title": properties)
Applications	("Document Title": "Software quality") AND "Document Title": Applications)
Metrics	((("Document Title":"Software quality") AND "Document Title":Metrics)

- **ACM**

Para esta etapa, en la base de datos ACM se realizaron dos refinamientos, debido a que la cantidad de artículos que se redujeron fue grande pero no suficiente.

El refinamiento se realizó en la estructura de las cadenas de búsqueda. En las Tabla 5.15 se puede observar estos refinamientos, y las cadenas utilizadas se muestran a continuación (ver Tabla 5.12):

Tabla 5.12. Cadenas utilizadas en el Refinamiento 2 ACM

Palabra	Cadena
Elements	("Software quality" and elements) and (PublishedAs:journal OR PublishedAs:magazine OR PublishedAs:newsletter) "Software quality" and (attributes or characteristics or factors)
Properties	((("Software quality" and properties)) and (PublishedAs:journal OR PublishedAs:magazine OR PublishedAs:newsletter))
Applications	((("Software quality" and applications)) and (PublishedAs:journal OR PublishedAs:magazine OR PublishedAs:newsletter))
Metrics	((("Software quality" and metrics)) and (PublishedAs:journal OR PublishedAs:magazine OR PublishedAs:newsletter))

ACM: Tercer Refinamiento

Tabla 5.13. Cadenas utilizadas en el Refinamiento 3 ACM

Palabra	Cadenas
Elements	(Title:"Software quality" and elements) and (PublishedAs:journal OR PublishedAs:magazine OR PublishedAs:newsletter) ((Title:"Software quality" and (attributes or characteristics or factors))) and (PublishedAs:journal OR PublishedAs:magazine OR PublishedAs:newsletter)
Properties	((Title:"Software quality" and properties)) and (PublishedAs:journal OR PublishedAs:magazine OR PublishedAs:newsletter)
Applications	((Title:"Software quality" and applications)) and (PublishedAs:journal OR PublishedAs:magazine OR PublishedAs:newsletter)
Metrics	((Title:"Software quality" and metrics)) and (PublishedAs:journal OR PublishedAs:magazine OR PublishedAs:newsletter)

Tabla 5.14. Número de Artículos encontrados: Segundo Refinamiento base de datos SPRINGER e IEEE

SPRINGER										
Palabra1	Palabra2	Palabra3	Palabra4	Total	Content Type	Discipline	SubDiscipline	Published In	Language	Año
"Software quality"	Elements			143	Article	Computer Science	SWE	Software Quality Journal	English	2007-2014
"Software quality"	Elements			79	Article	Computer Science	SWE	Empirical Software Engineerig	English	2007-2014
"Software quality"	Elements			15	Article	Computer Science	SWE	Innovations in Systems and Software Engineering	English	2007-2014
"Software quality"	Attributes	Characteristics	Factors	210	Article	Computer Science	SWE	Software Quality Journal	English	2007-2014
"Software quality"	Attributes	Characteristics	Factors	134	Article	Computer Science	SWE	Empirical Software Engineerig	English	2007-2014
"Software quality"	Attributes	Characteristics	Factors	21	Article	Computer Science	SWE	Innovations in Systems and Software Engineering	English	2007-2014
"Software quality"	Properties			132	Article	Computer Science	SWE	Software Quality Journal	English	2007-2014
"Software quality"	Properties			80	Article	Computer Science	SWE	Empirical Software Engineerig	English	2007-2014
"Software quality"	Properties			16	Article	Computer Science	SWE	Requirements Engineering	English	2007-2014
"Software quality"	Applications			208	Article	Computer Science	SWE	Software Quality Journal	English	2007-2014
"Software quality"	Applications			122	Article	Computer Science	SWE	Empirical Software Engineerig	English	2007-2014
"Software quality"	Applications			25	Article	Computer Science	SWE	Innovations in Systems and Software Engineering	English	2007-2014
"Software quality"	Metrics			156	Article	Computer Science	SWE	Software Quality Journal	English	2007-2014
"Software quality"	Metrics			105	Article	Computer Science	SWE	Empirical Software Engineerig	English	2007-2014
"Software quality"	Metrics			16	Article	Computer Science	SWE	Innovations in Systems and Software Engineering	English	2007-2014
TOTAL				1462						

IEEE									
Palabra1	Palabra2	Palabra3	Palabra4	Total	Content Type	Language	Año	En	En
"Software quality"	Elements			148	Conference Publications, Journals and Magazines	English	2007-2014	Metadata Only	Metadata Only
"Software quality"	Attributes	Characteristics	Factors	91	Conference Publications, Journals and Magazines	English	2007-2014	Document Title	Metadata Only
"Software quality"	Properties			23	Conference Publications, Journals and Magazines	English	2007-2014	Metadata Only	Document Title
"Software quality"	Applications			13	Conference Publications	English	2007-2014	Document Title	Document Title
"Software quality"	Metrics			19	Conference Publications, Journals and Magazines	English	2007-2014	Document Title	Document Title
TOTAL				294					

Fuente: Trabajo de Titulación (Anexo Refinamientos, Refinamiento de Cadenas-Etapa2)

Elaborado: Ana Villalta

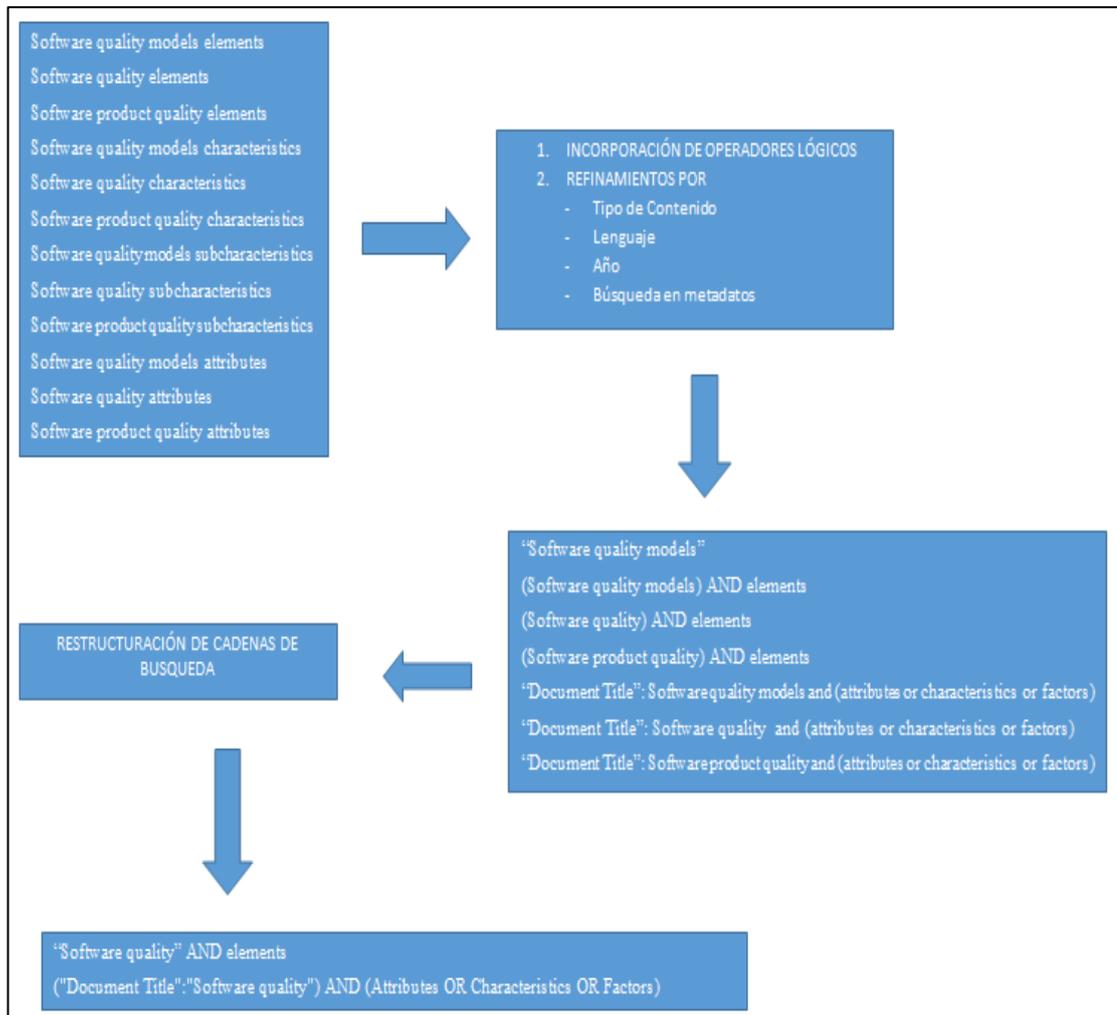
Tabla 5.15. Número de Artículos encontrados: Segundo y Tercer Refinamiento base de datos ACM

ACM								
Palabra1	Palabra2	Palabra3	Palabra4	Total	Año	Word or Phrases	Publication types	Results must have
Software quality	Elements			1714	2007-2014		Journals, Magazines, Conferences, SIGs	
Software quality	Attributes	Characteristics	Factors	2448	2007-2014		Journals, Magazines, Conferences, SIGs	
Software quality	Properties			1516	2007-2014		Journals, Magazines, Conferences, SIGs	
Software quality	Applications			2579	2007-2014		Journals, Magazines, Conferences, SIGs	
Software quality	Metrics			1578	2007-2014		Journals, Magazines, Conferences, SIGs	
TOTAL				9835				
ACM								
Palabra1	Palabra2	Palabra3	Palabra4	Total	Año	Word or Phrases	Publication types	Results must have
Software quality	Elements			45	2007-2014		Journals, Magazines, Conferences, SIGs	
Software quality	Attributes	Characteristics	Factors	76	2007-2014		Journals, Magazines, Conferences, SIGs	
Software quality	Properties			37	2007-2014		Journals, Magazines, Conferences, SIGs	
Software quality	Applications			72	2007-2014		Journals, Magazines, Conferences, SIGs	
Software quality	Metrics			62	2007-2014		Journals, Magazines, Conferences, SIGs	
TOTAL				292				

Fuente: Trabajo de Titulación (Anexo Refinamientos, Refinamiento de Cadenas-Etapa2)

Elaborado: Ana Villalta

Figura 5.5 Resumen del proceso de refinamiento para la palabra “Elements”.- IEEE



Para seleccionar los estudios de los 294 artículos obtenidos del último refinamiento, se eliminaron los redundantes arrojados al utilizar varias cadenas de búsqueda, en una misma base de datos. El resultado fue 283 artículos no redundantes entre sí. (Anexo, Selección de estudios IEEE)

Se dio lectura de los resúmenes y conclusiones de cada uno de los trabajos y 164 de ellos fueron evaluados como "Seleccionados", quedando los restantes como "Dudosos" o "Falsos". Los artículos clasificados como Dudosos pueden ser evaluados con mayor rigurosidad en etapas posteriores. El documento con mayor detalle se puede encontrar en el Anexo, Evaluación de estudios, IEEE y un resumen de este proceso se puede observar en la figura 5.6.

Las siguientes secciones (5.3, 5.4), describen con detalle las etapas de Selección y Evaluación de estudios.

Figura 5.6 Inclusión de los proceso de “Selección” y “Evaluación” de artículos a la figura 5.1



5.3 Selección de los estudios.

Para iniciar con la selección de los estudios, se exportaron los datos necesarios de cada uno de los artículos obtenidos en el último refinamiento, Esto se realizó para cada base de datos seleccionada (IEEE, SPRINGER, ACM).

Se utilizó la herramienta Excel para la creación de formularios que incluyen como campos los datos que se listan a continuación.

Los datos extraídos de cada trabajo fueron:

- Título del documento
- El autor(s)
- La fuente (la conferencia o revista)
- El año en que se publicó el documento
- Clasificación del trabajo
- Url

Los resultados de este proceso se pueden observar en el anexo, "Selección de estudios". Las tabla 5.16, 5.17 y 5.18 presenta un extracto de esta información.

Tabla 5.16. Datos extraídos para cada artículo.- SPRINGER

TITULO DEL DOCUMENTO	AUTORES	FUENTE	AÑO	CLASIFICACIÓN	URL
In search for a widely applicable and accepted software quality model for software quality engineering	Marc-Alexis CôtéWitold SurynElli Georgiadou	Software Quality Journal	2007	Article	http://link.springer.com/article/10.1007/s11219-007-9029-0
Integrating fuzzy theory and hierarchy concepts to evaluate software quality	Che-Wei ChangCheng-Ru WuHung-Lung Lin	Software Quality Journal	2008	Article	http://link.springer.com/article/10.1007/s11219-007-9035-2
An analysis of software quality management at AWE plc.	Michael ElliottRay DawsonJanet Edwards	Software Quality Journal	2007	Article	http://link.springer.com/article/10.1007/s11219-007-9027-2
A fuzzy regression and optimization approach for setting target levels in software quality function deployment	Zeynep SenerE. Ertugrul Karsak	Software Quality Journal	2010	Article	http://link.springer.com/article/10.1007/s11219-010-9095-6
Should we try to measure software quality attributes directly?	John Moses	Software Quality Journal	2009	Article	http://link.springer.com/article/10.1007/s11219-008-9071-6
Software quality analysis by combining multiple projects and learners	Taghi M. KhoshgoftarPierre ReboursNaeem Seliya	Software Quality Journal	2009	Article	http://link.springer.com/article/10.1007/s11219-008-9058-3
An experience-based framework for evaluating alignment of software quality goals	Panagiota ChatzipetrouLefteris AngelisSebastian BarneyClaes Wohlin	Software Quality Journal	2014	Article	http://link.springer.com/article/10.1007/s11219-014-9251-5
Quality specification and metrication, results from a case-study in a mission-critical software domain	Jos J. M. TrienekensRob J. KustersDennis C. Brussel	Software Quality Journal	2010	Article	http://link.springer.com/article/10.1007/s11219-010-9101-z
Introduction of static quality analysis in small- and medium-sized software enterprises: experiences from technology transfer	Mario GleirscherDmitriy GolubitskiyMaximilian IrlbeckStefan Wagner	Software Quality Journal	2014	Article	http://link.springer.com/article/10.1007/s11219-013-9217-z
An evolutionary cultural-change approach to successful software process improvement	Michael ElliottRay DawsonJanet Edwards	Software Quality Journal	2009	Article	http://link.springer.com/article/10.1007/s11219-008-9070-7
In this issue	Margaret Ross	Software Quality Journal	2009	Article	http://link.springer.com/article/10.1007/s11219-009-9073-z
Risk management capability model for the development of medical device software	Fergal Mc CafferyJohn BurtonIta Richardson	Software Quality Journal	2010	Article	http://link.springer.com/article/10.1007/s11219-009-9086-7
Prioritizing code-smells correction tasks using chemical reaction optimization	Ali OuniMarouane KessentiniSlim BechikhHouari Sahraoui	Software Quality Journal	2014	Article	http://link.springer.com/article/10.1007/s11219-014-9233-7
A comparative analysis of CMMI software project management by Brazilian, Indian and Chinese companies	Saulo Barbarã de OliveiraRogerio ValleCláudio Fernando Mahler	Software Quality Journal	2010	Article	http://link.springer.com/article/10.1007/s11219-009-9087-6
A test case refactoring approach for pattern-based software development	Peng-Hua ChuNien-Lin HsuehHong-Hsiang ChenChien-Hung Liu	Software Quality Journal	2012	Article	http://link.springer.com/article/10.1007/s11219-011-9143-x
Static correspondence and correlation between field defects and warnings reported by a bug finding tool	Cesar CoutoJoão Eduardo MontandonChristofer SilvaMarco Tulio Valente	Software Quality Journal	2013	Article	http://link.springer.com/article/10.1007/s11219-011-9172-5
A software product certification model	Petra HeckMartijn KlabbersMarko van Eekelen	Software Quality Journal	2010	Article	http://link.springer.com/article/10.1007/s11219-009-9080-0
Analyzing the co-evolution of comments and source code	Beat FluriMichael WärschEmanuel GigerHarald C. Gall	Software Quality Journal	2009	Article	http://link.springer.com/article/10.1007/s11219-009-9075-x
What you like in design use to correct bad-smells	Marouane KessentiniRim MahaouachiKhaled Ghedira	Software Quality Journal	2013	Article	http://link.springer.com/article/10.1007/s11219-012-9187-6
Design pattern evolutions in QVT	Jing DongYajing ZhaoYongtao Sun	Software Quality Journal	2010	Article	http://link.springer.com/article/10.1007/s11219-009-9093-8
Measuring size, complexity, and coupling of hypergraph abstractions of software: An information-theory approach	Edward B. AllenSampath GottipatiRajiv Govindarajan	Software Quality Journal	2007	Article	http://link.springer.com/article/10.1007/s11219-006-9010-3
Exploring fault types, detection activities, and failure severity in an evolving safety-critical software system	Maggie HamillKaterina Goseva-Popstojanova	Software Quality Journal	2014	Article	http://link.springer.com/article/10.1007/s11219-014-9235-5
A declarative approach for Java code instrumentation	Tian ZhangXiaomei ZhengYan ZhangJianhua ZhaoXuandong Li	Software Quality Journal	2013	Article	http://link.springer.com/article/10.1007/s11219-013-9220-4
Knowledge-oriented software engineering process in a multi-cultural context	Hannu JaakkolaAnneli HeimbürgerPetri Linna	Software Quality Journal	2010	Article	http://link.springer.com/article/10.1007/s11219-009-9091-x
Requirements and constructors for tailoring software processes: a systematic literature review	Tomás Martínez-RuizJürgen MünchFelix García-Mario Piattini	Software Quality Journal	2012	Article	http://link.springer.com/article/10.1007/s11219-011-9147-6

Fuente: Trabajo de titulación (Anexo Selección de Estudios) **Elaborado:** Ana Villalta

Tabla 5.17. Datos extraídos para cada artículo.- IEEE

TITULO DEL DOCUMENTO	AUTORES	FUENTE	ANO	RESUMEN	URL
Enhancing ISO/IEC 25021 quality measure elements for wider application within ISO 25000 series	St-Louis, D.; Suryan, W.	IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society	2012	This article presents an innovative approach to the determination of a core set of Quality Measure Elements (QME), and a comparative analysis aimed at enhancing applicability of ISO/IEC 25021, which are critical to the ISO/IEC 25000 SQUARE series of standards. The research presents the methodology, allowing for moving from an outdated and extensive set of measures, to a core list of base measures, which can be further diversified through contexts of use, parameters, and considerably simplify the development of derived measures.	http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6389400
Knowledge for Software Quality Control and Measurement	Haitao Sun	Business Computing and Global Informatization (BCGIN), 2011 International Conference on	2011	Software quality is turning out to be the driving force of competition for high-technology software products. Not only users but also software enterprises are attaching important to the influences which software quality brings more and more. Since measuring quality is the key to developing high-quality software [6], it is also disputed about how to control and measure quality during software development life cycle. In this paper, I show when and how to approach quality control and measurement. And which are the relations between each other. Confirmedly, they both are essential elements of succeeding in developing high-technology and high-quality software product for enterprise.	http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6003925
An Improved Fuzzy Synthesis Evaluation Algorithm for Software Quality	Jianli Dong	Information Management, Innovation Management and Industrial Engineering, 2009 International Conference on	2009	Based on the practice of software quality evaluated by the fuzzy synthesis evaluation algorithm, it is chiefly analyzed that the reason a lot of available information has been lost by calculating method of 'taking big and taking small value' in the fuzzy algorithm model, and shortage and limitation to be resulted in the simplification and the absolutization of evaluation result in the evaluation process are especially discussed. The operation 'taking big and taking small value' in original algorithm model is changed into matrix multiplication operation with original algorithm structure, and the normal distribution function is introduced for the measurement of the metric elements, these are used to establish a new mathematical model to suit modern software quality evaluation requirements. It is very useful and effective to apply the new model to software quality computing evaluation. And that it is fruitful to improve the accuracy, practicability and reliability as well as flexibility of software quality evaluation.	http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5370523
Multilayer Matter-Element Extension Synthesis Evaluation of Software Quality	Jianli Dong; Ningguo Shi	Biomedical Engineering and Computer Science (ICBECS), 2010 International Conference on	2010	According to the needs and characteristics of software quality synthesis evaluation, a new multilayer matter-element extension evaluation algorithm is introduced in the reference. This paper will continue to study and discuss the application of the new algorithm. Practice application has shown that the metric results of the multilayer matter-element extension synthesis evaluation algorithm is more accurate, practical and reliable than original algorithm for software quality metrics.	http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5462341
The influence of team relationships on software quality	Wong, B.; Bhatti, M.	Software Quality, 2009. WOSQ '09. ICSE Workshop on	2009	Implementing software development successfully in an organization is possibly one of the most challenging issues that the industry faces today. Project failures and the lack of software quality have become a major concern for researchers and practitioners. For many software development projects as well as for many quality assurance departments, teamwork is extremely important. Therefore, nearly all organizations pay attention to teamwork which is essential for smooth operation of a project. It is believed that teamwork is crucial for the success or failure of a project. In addition to the importance of teamwork, team issues affecting its' performance also needs attention. Among these team issues, people issues are scarce in the literature. The research outlined in this paper investigated the factors which influence teamwork. It identified the importance of trust and how trust influences IT project teams and the role of project managers and team leaders in improving trust within teams. It also includes individual behaviors that affect trust within a team. A study of six project managers from two Australian organizations was conducted to investigate the factors which influence teamwork to result in project success and software quality. The results showed that trust is a central element of the many factors and that without trust in the team, teamwork will be negatively affected resulting in project failure and poor quality.	http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5071550
A Fuzzy Synthetic Evaluation Method for Software Quality	Xiaojing Liu; Jihong Pang	e-Business and Information System Security (EBISS), 2010 2nd International Conference on	2010	The software quality evaluation problem is becoming increasingly important for software providers and their users. Existing most methods suitable for software quality evaluation are provided based on multiple-attribute decision-making (MADM) model. In this paper, an integration method of a fuzzy matter element (FME) and analytic hierarchy process (AHP) is used to consider quantitative and qualitative factors in evaluating the software quality. The fuzzy synthetic evaluation method is proposed to build the multiple software quality characteristics evaluation model based on the theory of fuzzy mathematics and matter element theory. Then, the AHP method is applied to calculate the weight of each evaluation index. Finally, a case study show that the new method is applicable for both theoretical and practical purposes.	http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5473544

Fuente: Trabajo de Titulación (Anexo, Selección de Estudios)

Elaborado: Ana Villalta

Tabla 5.18. Datos extraídos para cada artículo.- ACM

TITULO DEL DOCUMENTO	AUTORES	FUENTES	AÑO	URL
Software Quality Evaluation and Forecast Based on unascertained-SVR Model	Zhang, Yapeng and Zhang, Shihu	Int. J. Wire. Mob. Comput.	2014	http://dx.doi.org/10.1504/IJWMC.2014.059716
A Hybrid Heuristic Approach to Optimize Rule-based Software Quality Estimation Models	Azar, D. and Harmanani, H. and Korkmaz, R.	Inf. Softw. Technol.	2009	http://dx.doi.org/10.1016/j.infsof.2009.05.003
The Use of Mathematics in Software Quality Assurance	Parnas, David Lorge	Front. Comput. Sci China	2012	http://dx.doi.org/10.1007/s11704-012-2904-2
Integrating Fuzzy Theory and Hierarchy Concepts to Evaluate Software Quality	Chang, Che-Wei and Wu, Cheng-Ru and Lin, Hung-Lung	Software Quality Control	2008	http://dx.doi.org/10.1007/s11219-007-9035-2
Fuzzy Cognitive Map Based Approach for Software Quality Risk Analysis	Bhatia, Nitin and Kapoor, Namarta	SIGSOFT Softw. Eng. Notes	2011	http://doi.acm.org/10.1145/2047414.2047422
Why Interaction Between Metrics Should Be Considered in the Development of Software Quality Models: A Preliminary Study	Goyal, Rinkaj and Chandra, Pravin and Singh, Yogesh	SIGSOFT Softw. Eng. Notes	2014	http://doi.acm.org/10.1145/2632434.2659853
A Fuzzy Regression and Optimization Approach for Setting Target Levels in Software Quality Function Deployment	Sener, Zeynep and Karsak, E. Ertugrul	Software Quality Control	2010	http://dx.doi.org/10.1007/s11219-010-9095-6
Evaluating Software Quality Using Software Metrics for a Software Product	Sudhaman, Parthasarathy	Int. J. Inf. Syst. Chang. Manage.	2011	http://dx.doi.org/10.1504/IJISCM.2011.041514
Tests for Consistent Measurement of External Subjective Software Quality Attributes	Moses, John and Farrow, Malcolm	Empirical Softw. Engg.	2008	http://dx.doi.org/10.1007/s10664-007-9058-0
Should We Try to Measure Software Quality Attributes Directly?	Moses, John	Software Quality Control	2009	http://dx.doi.org/10.1007/s11219-008-9071-6
A Fuzzy Classifier Approach to Estimating Software Quality	Pizzi, Nick J.	Inf. Sci.	2013	http://dx.doi.org/10.1016/j.ins.2013.04.027
In Search for a Widely Applicable and Accepted Software Quality Model for Software Quality Engineering	C\^{\o}{\e}, Marc-Alexis and Suryan, Witold and Georgiadou, Elli	Software Quality Control	2007	http://dx.doi.org/10.1007/s11219-007-9029-0
The Software Quality Economics Model for Software Project Optimization	Lazic, Ljubomir and Kola\{s}inac, Amel and Avdic, D\{v}zenan	W. Trans. on Comp.	2009	http://dl.acm.org/citation.cfm?id=1534462.1534465
A Quantitative Evaluation of Aspect-oriented Software Quality Model (AOSQUAMO)	Kumar, Avadhesh and Grover, P. S. and Kumar, Rajesh	SIGSOFT Softw. Eng. Notes	2009	http://doi.acm.org/10.1145/1598732.1598736
A Conceptual Bayesian Net Model for Integrated Software Quality Prediction	Radli\{n}ski, \LUkasz	Ann. UMCS, Inf.	2011	http://dx.doi.org/10.2478/v10065-011-0032-5
Relating Software Quality Models and Process Methods to User Value	Sheriff, Mohamed A and Georgiadou, Elli	Int. J. Hum. Cap. Inf. Technol. Prof.	2013	http://dx.doi.org/jhcitp.2013040103
Measuring Critical Factors of Software Quality Management: Development and Validation of an Instrument	Vitharana, Padmal and Mone, Mark A.	Inf. Resour. Manage. J.	2008	http://dx.doi.org/10.4018/irmj.2008040102
Empirical Study of Open Source Software Selection for Adoption, Based on Software Quality Characteristics	Sarrab, Mohamed and Rehman, Osama M. Hussain	Adv. Eng. Softw.	2014	http://dx.doi.org/10.1016/j.advengsoft.2013.12.001
Determinants of Software Quality: A Survey of Information Systems Project Managers	Gorla, Narasimhaiah and Lin, Shang-Che	Inf. Softw. Technol.	2010	http://dx.doi.org/10.1016/j.infsof.2009.11.012

Fuente: Trabajo de Titulación (Anexo Selección de Estudios)

Elaborado: Ana Villalta

5.4 Evaluación de calidad del estudio

Para que los estudios de tipo SLR resulten de calidad, se deben utilizar criterios para la selección de artículos que permitan descartar aquellos que no resultan relevantes para el estudio.

En esta sección se amplían y desarrollan los criterios de selección de estudios descritos en el capítulo 4 (Protocolo). Se incluye información que apoya el desarrollo de la sección 5.3 de este capítulo y se profundiza en la evaluación de datos. Las estrategias que se utilizaron en esta etapa fueron:

1. Proporcionar criterios aún más detallados de selección
2. Ponderación de la importancia de los estudios individuales
3. Orientar a las recomendaciones para futuras investigaciones

Las siguientes secciones describen cada una de ellas.

5.4.1 Proporcionar criterios aún más detallados de selección.

Para refinar los resultados se plantearon los siguientes criterios de selección y evaluación:

- En SPINGER se seleccionaron los artículos que respondieron a las preguntas de investigación en el título del documento.
- En IEEE se seleccionaron los artículos que respondieron a las preguntas de investigación en su resumen y conclusiones, aun si su título no contenía las palabras clave.
- En ACM se dio lectura a todos los resúmenes de los artículos tomando en cuenta aquellos que también respondieron a la selección en el título del documento.
- Para cada base de datos se creó una columna más en los formularios que se denominó notas adicionales. Tras haber aplicado los criterios descritos para cada Base de Datos y evaluado cada uno de los estudios se colocó una palabra descriptiva cuyo fin fue indicar si el artículo sería "Seleccionado", "Dudoso" o "Falso".

5.4.2 Ponderación de la importancia de los estudios individuales.

Los artículos clasificados como dudosos o falsos podrán servir como base para futuros estudios que amplíen esta SLR y evalúen una mayor cantidad de trabajos con mayor profundidad, de esta forma:

- Los estudios clasificados como dudosos podrán ser llevados a estudios seleccionados o falsos.
- Los estudios falsos podrán ser llevados a falsos positivos o falsos negativos
- Los falsos negativos podrán ser llevados a positivos, si fuera el caso.

5.4.3 Orientar a las recomendaciones para futuras investigaciones.

Debido a que una SLR es un trabajo extenso, se propuso una recomendación para trabajos futuros:

- Se dejó abierta la investigación en las bases de datos SPRINGER y ACM. En una futura investigación se puede comenzar la lectura de los artículos clasificados como Seleccionados y continuar a la etapa de extracción de datos hasta terminar el estudio propuesto.

En esta etapa se reevaluaron los estudios extraídos en la etapa anterior "Selección de estudios" por un solo investigador, la autora de esta tesis. El documento con los resultados se puede observar en el anexo "Evaluación de Estudios" mientras que, un extracto de los formularios para evaluación de los artículos seleccionados se puede observar en las tablas 5.19, 5.20, 5.21:

Tabla 5.19. Clasificación de artículos en Seleccionado, Dudoso o Falso.- SPRINGER

TITULO DEL DOCUMENTO	AUTORES	FUENTE	AÑO	CLASIFICACION	URL	NOTAS ADICIONALES
In search for a widely applicable and accepted software quality model for software quality engineering	Marc-Alexis CĂ'itĂ@Witold Suryelli Georgiadou	Software Quality Journal	2007	Article	http://link.springer.com/article/10.1007/s11219-007-90	SELECCIONADO
Integrating fuzzy theory and hierarchy concepts to evaluate software quality	Che-Wei ChangCheng-Ru WuHung-Lung Lin	Software Quality Journal	2008	Article	http://link.springer.com/article/10.1007/s11219-007-90	DUDOSO
An analysis of software quality management at AWE plc.	Michael ElliottRay DawsonJanet Edwards	Software Quality Journal	2007	Article	http://link.springer.com/article/10.1007/s11219-007-90	DUDOSO
A fuzzy regression and optimization approach for setting target levels in software quality function deployment	Zeynep SenerE. Ertugrul Karsak	Software Quality Journal	2010	Article	http://link.springer.com/article/10.1007/s11219-010-90	FALSO
Should we try to measure software quality attributes directly?	John Moses	Software Quality Journal	2009	Article	http://link.springer.com/article/10.1007/s11219-008-90	DUDOSO
Software quality analysis by combining multiple projects and learners	Taghi M. KhoshgoftaarPierre ReboursNaeem Seliya	Software Quality Journal	2009	Article	http://link.springer.com/article/10.1007/s11219-008-90	FALSO
An experience-based framework for evaluating alignment of software quality goals	Panagiota ChatzipetrouEfteris AngelisSebastian BarneyClaes Wohlin	Software Quality Journal	2014	Article	http://link.springer.com/article/10.1007/s11219-014-92	FALSO
Quality specification and metrication, results from a case-study in a mission-critical software domain	Jos J. M. TrienekensRob J. KustersDennis C. Brussel	Software Quality Journal	2010	Article	http://link.springer.com/article/10.1007/s11219-010-91	FALSO
Introduction of static quality analysis in small- and medium-sized software enterprises: experiences from technology transfer	Mario GleirscherDmitriy GolubitskiyMaximilian IrlbeckStefan Wagner	Software Quality Journal	2014	Article	http://link.springer.com/article/10.1007/s11219-013-92	DUDOSO
An evolutionary cultural-change approach to successful software process improvement	Michael ElliottRay DawsonJanet Edwards	Software Quality Journal	2009	Article	http://link.springer.com/article/10.1007/s11219-008-90	FALSO
In this issue	Margaret Ross	Software Quality Journal	2009	Article	http://link.springer.com/article/10.1007/s11219-009-90	DUDOSO
Risk management capability model for the development of medical device software	Fergal Mc CafferyJohn BurtonIta Richardson	Software Quality Journal	2010	Article	http://link.springer.com/article/10.1007/s11219-009-90	FALSO
Prioritizing code-smells correction tasks using chemical reaction optimization	Ali OuniMarouane KessentiniSlim BechikhHouari Sahraoui	Software Quality Journal	2014	Article	http://link.springer.com/article/10.1007/s11219-014-92	FALSO
A comparative analysis of CMMI software project management by Brazilian, Indian and Chinese companies	Saulo BarbarĂ; de OliveiraRogerio ValleClĂ;udio Fernando Mahler	Software Quality Journal	2010	Article	http://link.springer.com/article/10.1007/s11219-009-90	FALSO
A test case refactoring approach for pattern-based software development	Peng-Hua ChuNien-Lin HsuehHong-Hsiang ChenChien-Hung Liu	Software Quality Journal	2012	Article	http://link.springer.com/article/10.1007/s11219-011-91	DUDOSO
Static correspondence and correlation between field defects and warnings reported by a bug finding tool	Cesar CoutoJoĂ;o Eduardo MontandonChristofer SilvaMarco Tullio Valente	Software Quality Journal	2013	Article	http://link.springer.com/article/10.1007/s11219-011-91	FALSO

Fuente: Trabajo de Titulaci3n (Anexo Evaluaci3n de Estudios)

Elaborado: Ana Villalta

Tabla 5.20. Clasificación de artículos en Seleccionado, Dudoso o Falso.- IEEE

TITULO DEL DOCUMENTO	AUTORES	FUENTE	AÑO	RESUMEN	URL	NOTAS ADICIONALES
Enhancing ISO/IEC 25021 quality measure elements for wider application within ISO 25000 series	St-Louis, D.; Suryn, W.	IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society	2012	This article presents an innovative approach to the determination of a core set of Quality Measure Elements (QME), and a comparative analysis aimed at enhancing applicability of ISO/IEC 25021, which are critical to the ISO/IEC 25000 SQuaRE series of standards. The research presents the methodology, allowing for moving from an outdated and extensive set of measures, to a core list of base measures, which can be further diversified through contexts of use, parameters, and considerably simplify the development of derived measures.	http://ieeexplore.ieee.org/stamp/stamp.jsp?arn	SELECCIONADO
Knowledge for Software Quality Control and Measurement	Haitao Sun	Business Computing and Global Informatization (BCGIN), 2011 International Conference on	2011	Software quality is turning out to be the driving force of competition for high-technology software products. Not only users but also software enterprises are attaching importance to the influences which software quality brings more and more. Since measuring quality is the key to developing high-quality software [6], it is also disputed about how to control and measure quality during software development life cycle. In this paper, I show when and how to approach quality control and measurement. And which are the relations between each other. Confirmedly, they both are essential elements of succeeding in developing high-technology and high-quality software product for enterprise.	http://ieeexplore.ieee.org/stamp/stamp.jsp?arn	SELECCIONADO
An Improved Fuzzy Synthesis Evaluation Algorithm for Software Quality	Jianli Dong	Information Management, Innovation Management and Industrial Engineering, 2009 International Conference on	2009	Based on the practice of software quality evaluated by the fuzzy synthesis evaluation algorithm, it is chiefly analyzed that the reason a lot of available information has been lost by calculating method of 'taking big and taking small value' in the fuzzy algorithm model, and shortage and limitation to be resulted in the simplification and the absolutization of evaluation result in the evaluation process are especially discussed. The operation 'taking big and taking small value' in original algorithm model is changed into matrix multiplication operation with original algorithm structure, and the normal distribution function is introduced for the measurement of the metric elements, these are used to establish a new mathematical model to suit modern software quality evaluation requirements. It is very useful and effective to apply the new model to software quality computing evaluation. And that it is fruitful to improve the accuracy, practicability and reliability as well as flexibility of software quality evaluation.	http://ieeexplore.ieee.org/stamp/stamp.jsp?arn	SELECCIONADO
Multilayer Matter-Element Extension Synthesis Evaluation of Software Quality	Jianli Dong; Ningguo Shi	Biomedical Engineering and Computer Science (ICBECS), 2010 International Conference on	2010	According to the needs and characteristics of software quality synthesis evaluation, a new multilayer matter-element extension evaluation algorithm is introduced in the reference. This paper will continue to study and discuss the application of the new algorithm. Practice application has shown that the metric results of the multilayer matter-element extension synthesis evaluation algorithm is more accurate, practical and reliable than original algorithm for software quality metrics.	http://ieeexplore.ieee.org/stamp/stamp.jsp?arn	SELECCIONADO

Fuente: Trabajo de Titulación (Anexo Evaluación de Estudios)

Elaborado: Ana Villalta

Tabla 5.21. Clasificación de artículos en Seleccionado, Dudoso o Falso.- ACM

TITULO DEL DOCUMENTO	AUTORES	FUENTE	AÑO	URL	RESUMEN	NOTAS ADICIONALES
Software Quality Evaluation and Forecast Based on unascertained-SVR Model	Zhang, Yapeng and Zhang, Shihu	Int. J. Wire. Mob. Comput.	2014	http://dx.doi.org/10.1504/IJWMC.2014.059716	The software quality is a comprehensive fuzzy evaluation problem which might be divided into different layers based on its attributes. An unascertained-SVR measure for evaluating and forecasting the quality of software is established on the basis of analysing the factors affecting the quality risks of software system by applying the unascertained measure theory. Therefore, a new system is found to evaluate software quality based on the knowledge of software project. Then, the quality of software is evaluated and predicted by introducing a new mathematical model - Support Vector Regression SVR model. SVR is one of the best events on dealing with small samples, avoiding the defects of neural network that is easy to fall into local minimum and lower accuracy rate. Finally, the practical application shows that the method overcomes the defect that the variable set by experts, then makes the evaluation results objective and scientific.	SELECCIONADO
A Hybrid Heuristic Approach to Optimize Rule-based Software Quality Estimation Models	Azar, D. and Harmanani, H. and Korkmaz, R.	Inf. Softw. Technol.	2009	http://dx.doi.org/10.1016/j.infsof.2009.05.001	Software quality is defined as the degree to which a software component or system meets specified requirements and specifications. Assessing software quality in the early stages of design and development is crucial as it helps reduce effort, time and money. However, the task is difficult since most software quality characteristics (such as maintainability, reliability and reusability) cannot be directly and objectively measured before the software product is deployed and used for a certain period of time. Nonetheless, these software quality characteristics can be predicted from other measurable software quality attributes such as complexity and inheritance. Many metrics have been proposed for this purpose. In this context, we speak of estimating software quality characteristics from measurable attributes. For this purpose, software quality estimation models have been widely used. These take different forms: statistical models, rule-based models and decision trees. However, data used to build such models is scarce in the domain of software quality. As a result, the accuracy of the built estimation models deteriorates when they are used to predict the quality of new software components. In this paper, we propose a search-based software engineering approach to improve the prediction accuracy of software quality estimation models by adapting them to new unseen software products. The method has been implemented and favorable result comparisons are reported in this work.	SELECCIONADO
The Use of Mathematics in Software Quality Assurance	Parnas, David Lorge	Front. Comput. Sci China	2012	http://dx.doi.org/10.1007/s11704-012-2904-2	The use of mathematics for documenting, inspecting, and testing software is explained and illustrated. Three measures of software quality are described and discussed. Then three distinct complementary approaches to software quality assurance are presented. A case study, the testing and inspection of a safety-critical system, is discussed in detail.	SELECCIONADO
Integrating Fuzzy Theory and Hierarchy Concepts to Evaluate Software Quality	Chang, Che-Wei and Wu, Cheng-Ru and Lin, Hung-Lung	Software Quality Control	2008	http://dx.doi.org/10.1007/s11219-007-9035-1	This study proposes a software quality evaluation model and its computing algorithm. Existing software quality evaluation models examine multiple characteristics and are characterized by factorial fuzziness. The relevant criteria of this model are derived from the international norm ISO. The main objective of this paper is to propose a novel Analytic Hierarchy Process (AHP) approach for addressing uncertainty and imprecision in service evaluation during pre-negotiation stages, where comparative judgments of decision makers are represented as fuzzy triangular numbers. A new fuzzy prioritization method, which derives crisp priorities from consistent and inconsistent fuzzy comparison matrices, is proposed. The Fuzzy Analytic Hierarchy Process (FAHP)-based decision-making method can provide decision makers or buyers with a valuable guideline for evaluating software quality. Importantly, the proposed model can aid users and developers in assessing software quality, making it highly applicable for academic and commercial purposes.	SELECCIONADO

Fuente: Trabajo de Titulación (Anexo Evaluación de Estudios)

Elaborado: Ana Villalta

5.5 Extracción de estudios

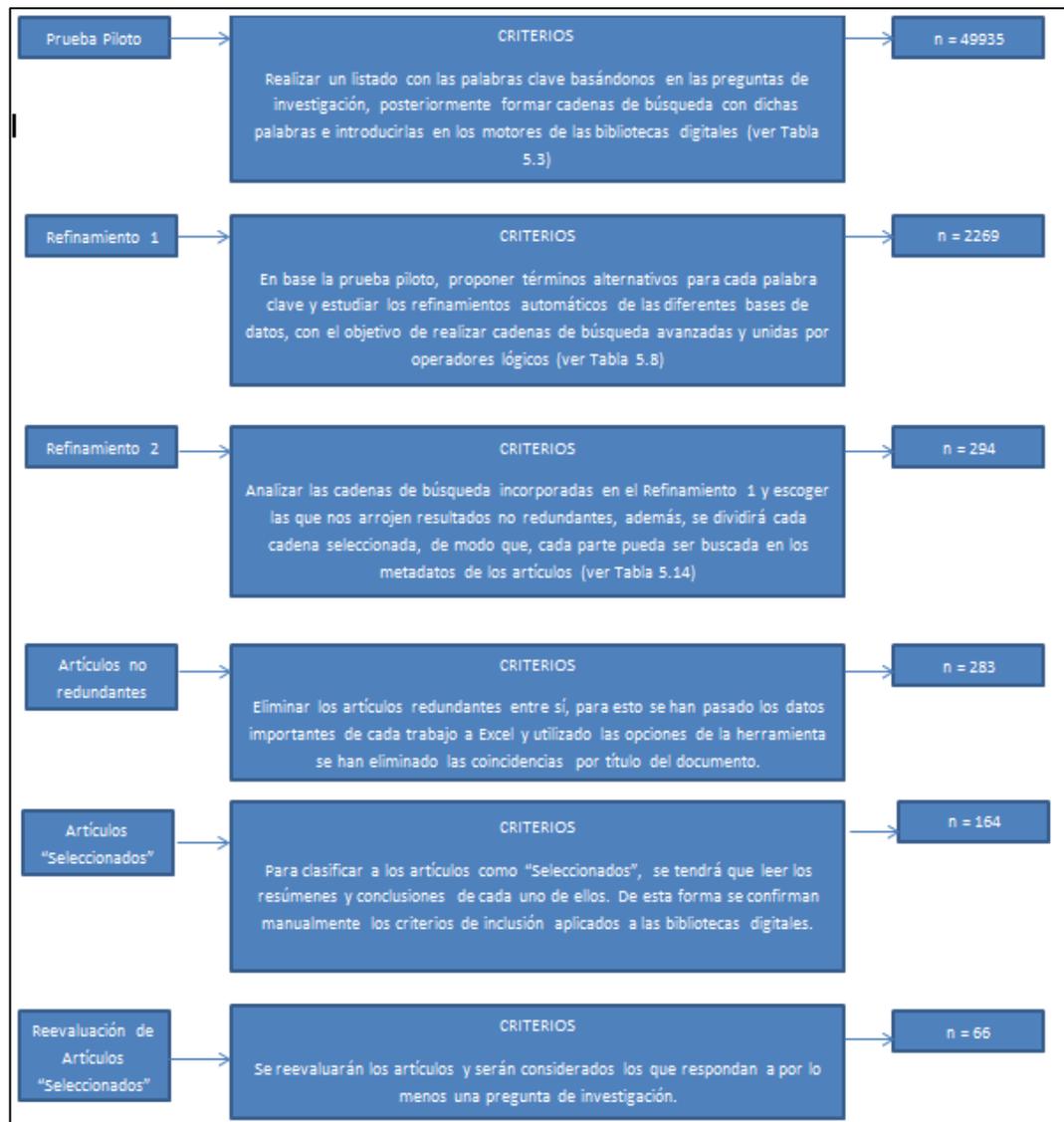
En esta sección se describe el proceso realizado para la extracción y selección definitiva de los estudios.

Se utilizó un formulario con las preguntas de investigación y cada artículo "Seleccionado" fue reevaluado, de acuerdo al Criterio 4 de la "Lista de verificación y control". Este puede ser observado en el protocolo del capítulo dos de este documento.

- Criterio 4: ¿Los estudios atienden a las preguntas de investigación?

En la figura 5.7 se puede observar que 66 estudios respondieron a las preguntas de investigación. Para mayor detalle se adjunta el Anexo "Extracción de estudios, Preguntas de clasificación".

Figura 5.7 Inclusión del proceso de “Extracción de datos” a la figura 5.6



De los 66 artículos mencionados, 50 de ellos fueron escogidos por un investigador para su lectura. Estos fueron puestos a consideración de un segundo investigador (Ing. Juan Pablo Carvallo) y fueron seleccionados 41 artículos en consenso, para lograrlo se llevaron a cabo dos etapas:

1) Primera etapa.

El objeto de la primera etapa fue seleccionar 50 artículos a los cuales se quiso dar lectura, para conseguirlo se desarrollaron las siguientes actividades:

- Creación de formularios con las preguntas de investigación descritas en el protocolo.
- Distribución de cada estudio leído y clasificado como Seleccionado en cada una de las preguntas descritas en el formulario. Se aplicó el criterio 4 de la lista de verificación y control ver sección 4.5.6.
- Selección de 50 artículos considerados pertinentes para el estudio de la Base de Datos IEEE (La autora de esta tesis). Estos fueron puestos en discusión en una segunda etapa en la que se reconsideró su lectura según la opinión de un segundo investigador (Ing. Juan Pablo Carvallo Vega)

El formulario obtenido incluyó la primera selección de los 50 artículos elegidos de la Base de Datos digital IEEE y señalados con letra amarilla para facilitar su identificación, un extracto de este proceso se puede observar en la Tabla 5.23 y el documento completo se encuentra en el anexo "Extracción de datos", Preguntas de Clasificación. Los artículos de las diferentes bases de datos pueden ser diferenciados según se muestra en la Tabla 5.22:

Tabla 5.22. Colores de los artículos por Base de Datos

IEEE
SPRINGER
ACM

Fuente: Trabajo de Titulación

Elaborado: Ana Villalta

Tabla 5.23. Artículos clasificados acorde a las Preguntas de Investigación y por Base de Datos

P1. ¿Cuáles son los elementos que estructuran los modelos de calidad de software?	P2. ¿Cómo se relacionan los distintos elementos entre sí?	P3. ¿Cuáles son las principales propiedades de los modelos de calidad del software que han sido descritas en la literatura?	P4. ¿Cuáles son las aplicaciones para las cuales han sido propuestos los modelos de calidad de software?	P6. ¿Qué tipo de calidad evalúan interna, externa, en uso y que elementos se relacionan a estos tipos?
Enhancing ISO/IEC 25021 quality measure elements for wider application within ISO 25000 series(mejora de la norma ISO 25021)		The Utilization of Metrics Usability to Evaluate the Software Quality(analiza la ISO/9126)	Providing a Software Quality Framework for Testing of Mobile Applications	JISBD01 - An Analysis of the Software Components Quality in Use using Bayesian Networks(calidad en uso)
Code4Thought Project: Employing the ISO/IEC-9126 Standard for Software Engineering-Product Quality Assessment(Ampliación de la ISO / IEC-9126 calidad de producto)		An experience report for software quality evaluation in highly iterative development methodology using traditional metrics(Metricas de fiabilidad para desarrollo iterativo)	Evaluating scalable matching tools: A quality-oriented approach(metodo de evaluacion basado en modelo de calidad para herramientas de correlacion escalables)	Knowledge for Software Quality Control and Measurement(calidad de software en el desarrollo)
A Knowledge Discovery Case Study of Software Quality Prediction: ISBSG Database(calidad basada en el modelo MCLP)		A Quantitative Evaluation of Software Quality Enhancement by Refactoring Using Dependency Oriented Complexity Metrics(ev alua metricas de complejidad)	Can requirements dependency network be used as early indicator of software integration bugs?(complejidad y cohesión de acoplamiento como indicadores de calidad en redes)	Software quality in use characteristic mining from customer reviews
A survey-based study of the mapping of system properties to ISO/IEC 9126 maintainability characteristics(medición de calidad técnica del producto software)		Using security metrics in software quality assurance process(ev alua metricas de seguridad)	An Online Monitoring Approach for Web services(Monitoreo Online de calidad Web impulsado por un modelo de calidad)	Evaluating software quality in use using user reviews mining
Guideline-based approach for achieving non-functional attributes of software (calidad por medio de atributos no funcionales)		Software Quality, Metrics, Process Improvement, and CMMI: An Interview with Dick Fairley	Software Quality Model for Consumer Electronics Product(ISO 9126 para el software de la electronica de consumo)	Software quality measurement for reusability(cambios en el software en la fase de uso)
Object-Oriented Software Maintainability Measurement in the Past Decade(revisa estudios existentes en la medición de la mantenibilidad en la última década)		Taguchi Smaller-the-Best Software Quality Metrics	A Visualization Technique for Metrics-Based Hierarchical Quality Models(Técnica de visualización para el modelo de calidad jerárquico basado en métricas y redes neuronales)	
Research Development on the Systematic Application of Software Quality Management(compara ISO 9001 y CMMI para sinergia entre los dos)		Taguchi Larger-the-Best Software Quality M	Evaluating the Impact of Adaptive Maintenance Process on Open Source Software Quality(calidad de software en código abierto OSS)	
Integration of DEMATEL and ANP methods for calculate the weight of characteristics software quality based model ISO 9126 (evalua los pesos locales para las características de la norma ISO y los pesos globales para cada subcaracterística, representando su nivel de importancia)		Enabling the Evolution of J2EE Applications through Reverse Engineering and Quality Assurance(Técnicas y métricas para la calidad de aplicaciones J2EE)	Reducing the number of security vulnerabilities in web applications by improving software quality	
A Mobile Software Quality Model(modelo de calidad para para desarrollo de aplicaciones móviles)		Towards a set of Measures for Evaluating Software Agent Autonomy(Medidas para evaluar un Agent-oriented software asegurando la fiabilidad y funcionalidad)	The influence of team relationships on software quality	
An improved analytic hierarchy process model on Software Quality Evaluation(nuevo metodo de evaluacion de calidad de software)		Assessing quality of web based systems(modelo de evaluacion para sistemas basados en la web)	Towards data warehouse quality through integrated requirements analysis(calidad empresarial, calidad técnica y de información para reducir el riesgo en un almacén de datos)	
SQUALE T0 Software QUALity Enhancement(mejoras del modelo Qualixo basado en métricas de software)		Method of software quality management based on functional point metrics(Estimación de parámetros de coste y tiempo de desarrollo mediante puntos funcionales)	An Improved Fuzzy Synthesis Evaluation Algorithm for Software Quality	
Analysis of Software Evolvability in Quality Models(Adaptar modelos de calidad para analizar la capacidad de evolución del Software)		Novel sensitive object-oriented cohesion metric(Evaluación de la calidad de diseño con la métrica de cohesión)	How to improve software development process using mathematical models for quality prediction and elements of Six Sigma methodology(Técnicas de predicción de calidad de software)	

Fuente: Trabajo de Titulación (Anexo Extracción de datos)

Elaborado: Ana Villalta

2) **Segunda etapa.**

La lectura de los primeros 50 artículos, seleccionados en la primera etapa de esta sección, fue puesta en consideración por un segundo investigador (Ing. Juan Pablo Carvallo Vega).

De esta revisión se obtuvo 41 artículos, resultado de un consenso entre los 2 investigadores. Pueden ser observados con más detalle en el anexo , "Extracción de estudios" Artículos a leer.

Para esta etapa se seleccionó una sola base de datos (IEEE). Junto al segundo investigador, se escogieron los estudios pertinentes del formulario Preguntas de Clasificación. Estos fueron seleccionados para una lectura completa. Los estudios se señalaron con letra naranja para su distinción y un extracto de la lista obtenida se muestra a continuación (ver Tabla 5.24).

Tabla 5.24. Artículos a leer, clasificados por dos investigadores

P1. ¿Cuáles son los elementos que estructuran los modelos de calidad de software?	P2. ¿Cómo se relacionan los distintos elementos entre sí?	P3. ¿Cuáles son las principales propiedades de los modelos de calidad del software que han sido descritas en la literatura?	P4. ¿Cuáles son las aplicaciones para las cuales han sido propuestos los modelos de calidad de software?	P6. ¿Qué tipo de calidad evalúan interna, externa, en uso y que elementos se relacionan a estos tipos?
Enhancing ISO/IEC 25021 quality measure elements for wider application within ISO 25000 series(mejora de la norma ISO 25021)		The Utilization of Metrics Usability to Evaluate the Software Quality	Providing a Software Quality Framework for Testing of Mobile Applications	JISBD01 - An Analysis of the Software Components Quality in Use using Bayesian Networks
Code4Thought Project: Employing the ISO/IEC-9126 Standard for Software Engineering-Product Quality Assessment(Ampliación de la ISO / IEC-9126 calidad de producto)		An experience report for software quality evaluation in highly iterative development methodology using traditional metrics	Evaluating scalable matching tools: A quality-oriented approach	Knowledge for Software Quality Control and Measurement
A Knowledge Discovery Case Study of Software Quality Prediction: ISBSG Database(calidad basada en el modelo MCLP)		A Quantitative Evaluation of Software Quality Enhancement by Refactoring Using Dependency Oriented Complexity Metrics	Can requirements dependency network be used as early indicator of software integration bugs?	Software quality in use characteristic mining from customer reviews
A survey-based study of the mapping of system properties to ISO/IEC 9126 maintainability characteristics		Using security metrics in software quality assurance process	An Online Monitoring Approach for Web services(Monitoreo Online de calidad Web impulsado por un modelo de calidad)	Evaluating software quality in use using user reviews mining
Guideline-based approach for achieving non-functional attributes of software		Software Quality, Metrics, Process Improvement, and CMMI: An Interview with Dick Fairley	Software Quality Model for Consumer Electronics Product	Software quality measurement for reusability
Object-Oriented Software Maintainability Measurement in the Past Decade		Taguchi Smaller-the-Best Software Quality Metrics	A Visualization Technique for Metrics-Based Hierarchical Quality Models	
Research Development on the Systematic Application of Software Quality Management		Taguchi Larger-the-Best Software Quality M	Evaluating the Impact of Adaptive Maintenance Process on Open Source Software Quality{	
Integration of DEMATEL and ANP methods for calculate the weight of characteristics software quality based model ISO 9126		Enabling the Evolution of J2EE Applications through Reverse Engineering and Quality Assurance	Reducing the number of security vulnerabilities in web applications by improving software quality	
A Mobile Software Quality Model		Towards a set of Measures for Evaluating Software Agent Autonomy	The influence of team relationships on software quality	
An improved analytic hierarchy process model on Software Quality Evaluation		Assessing quality of web based systems	Towards data warehouse quality through integrated requirements analysis	
SQUALE \rightarrow Software QUALity Enhancement		Method of software quality management based on functional point metrics	An Improved Fuzzy Synthesis Evaluation Algorithm for Software Quality	
Analysis of Software Evolvability in Quality Models		Novel sensitive object-oriented cohesion metric	How to improve software development process using mathematical models for quality prediction and elements of Six Sigma methodology	
User-Oriented Measurement of Software Flexibility		Software Quality Metrics and their Impact on Embedded Software	Does distributed development affect software quality? An empirical case study of Windows Vista	
An Approach Based on DEVS for Evaluating Quality Attributes		Quantitative Evaluation of Software Quality Metrics in Open-Source Projects	On understanding software quality evolution from a defect perspective: A case study on an open source software system	

5.6 Síntesis de los datos

Tras la lectura completa de los 41 artículos seleccionados por los dos investigadores se determinó que 20 de ellos respondieron completamente a los criterios de selección.

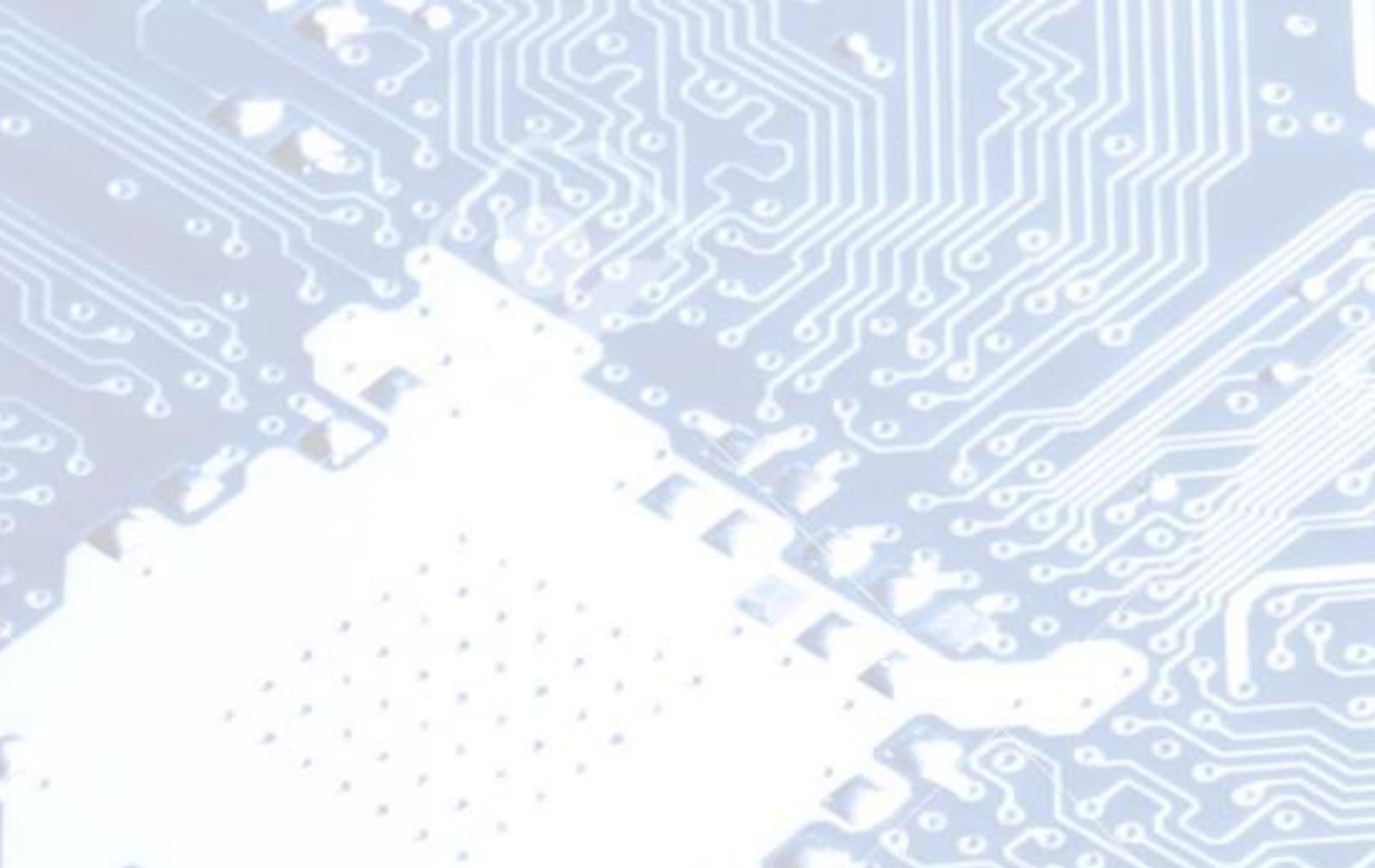
Para sintetizar la información de los artículos individuales se utilizaron las tablas denominadas "Formato Individual" (ver Tabla 5.25) y para resumir los estudios individuales se utilizó la denominada "Tabla resumen" (ver Tabla 5.26). Estas tablas fueron analizadas en el capítulo 3, en donde se dedica una sección para describirlas.

El detalle de esta etapa se puede encontrar en el anexo "Síntesis de datos" y los conceptos base para realizarla están descritos en el capítulo 2 apartado 2.5.2.

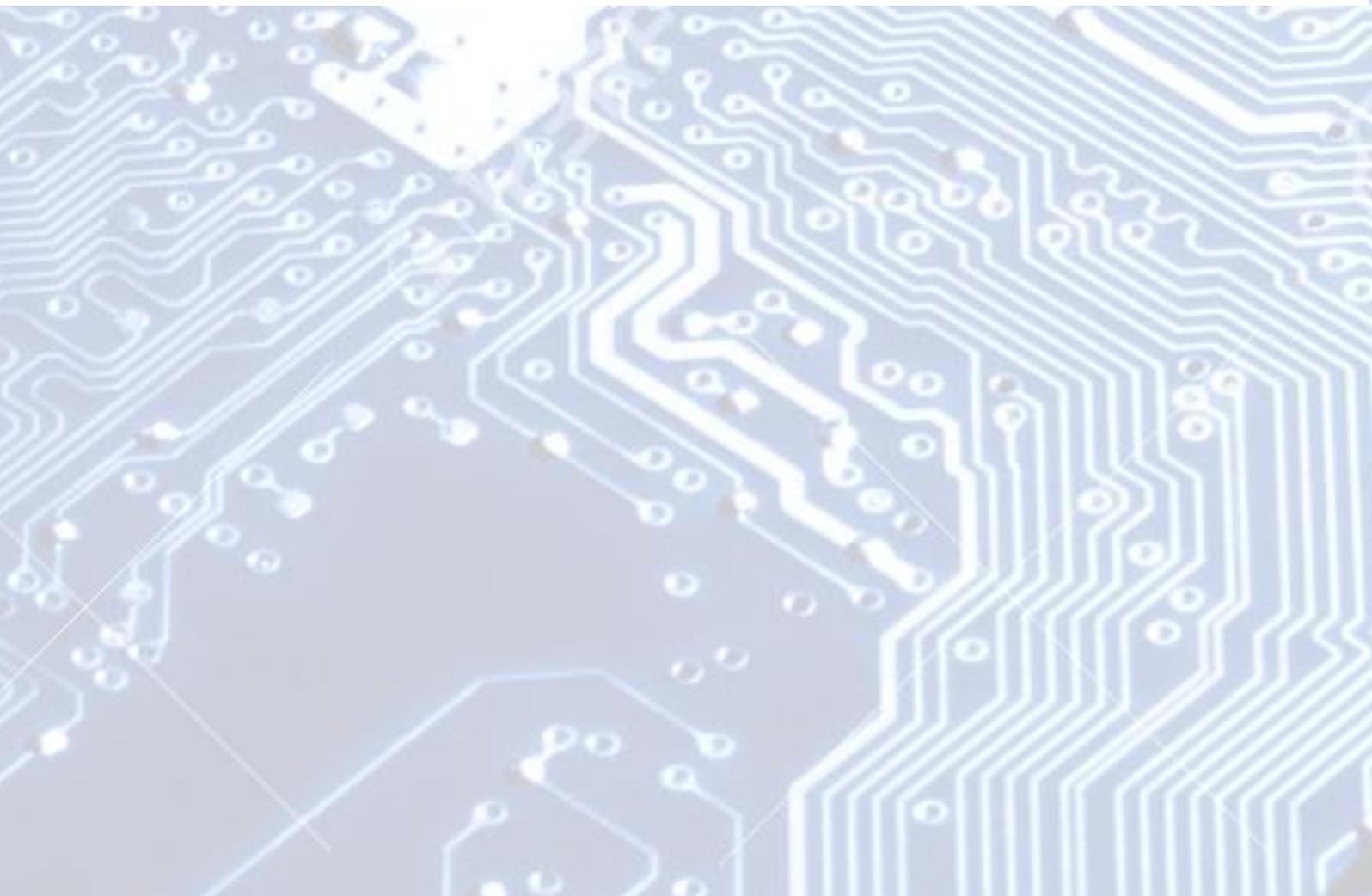
Tabla 5.25. Tabla formato para resumir los artículos individuales [MTK09]

An Economical Approach to Usability Testing									
Mueller, C.J.; Tamir, D.; Komogortsev, O.V.; Feldman, L. [MTK09]									
No.	Levels	[CKV09]	Type	If Technical, which type	References	Descriptions	Metrics	Metrics Description	Interpretation of the metric
1	1	Effort-based Usability	NT		Pag.1 (2. Effort-based Usability)				E = (Emental Ephysical)
2	1 1	E-mental	NT			Amount of mental effort to complete the task			Emental = (Eyemental Eothermental)
3	1 1 1						Eeye – mental	Amount of mental effort necessary to move and focus the eyes to complete the task	
4	1 1 2						Eother – mental	Amount of unspecified mental effort necessary to complete the task	
5	1 2	E-physical	NT			Amount of physical effort to complete the task			Ephysical = (Emanual_physical Eye_physical Eother_physical)
6	1 2 1						Emanual - physical	Amount of manual effort to complete the task. Manual effort includes, but is not limited to, the movement of fingers, hands, arms, etc.	
7	1 2 2						E-eye-physical	Amount of eye physical effort to complete the task	
8	1 2 3						E-other-physical	Amount of unspecified physical effort to complete the task.	

Elaborado: Ana Villalta



CAPÍTULO 6. ANÁLISIS DE LA REVISIÓN



6.1 Introducción

Este capítulo busca sintetizar las respuestas a las preguntas de investigación planteadas en el capítulo 4, "Planeación de la Revisión". Las respuestas serán mostradas por medio de gráficos estadísticos que surgieron a partir de la culminación de la última etapa del capítulo 5, "Desarrollo de la SLR", sección 5.6, "Síntesis de los datos".

En la etapa de síntesis de los datos, por medio de la "Tabla Resumen" (ver Tabla 5.26), se observó que el proceso de SLR ha sido conducido de manera correcta y que los 20 artículos leídos apoyan a la investigación realizada.

Las próximas secciones darán énfasis a los datos recopilados en las tablas y analizarán el proceso de SLR llevado a cabo.

6.2 Análisis de la SLR

Como se dijo en la introducción, en esta sección, se respondió a las preguntas de investigación definidas en la etapa de planeación de la revisión.

Los datos mostrados, corresponden a la base de datos IEEE, que es la biblioteca digital con la que se trabajó en toda la investigación, hasta concluir el proceso.

Mediante gráficos estadísticos se dio respuesta a las siguientes preguntas:

P1. ¿Cuáles son los elementos que estructuran los modelos de calidad de software?

P2. ¿Cómo se relacionan los distintos elementos entre sí?

P3. ¿Cuáles son las principales propiedades de los modelos de calidad del software que han sido descritas en la literatura?

P4. ¿Cuáles son las aplicaciones para las cuales han sido propuestos los modelos de calidad de software?

P5. ¿Qué métodos se han utilizado para su construcción?

P6. ¿Qué tipo de calidad evalúan, interna, externa, en uso y qué elementos se relacionan a estos tipos?

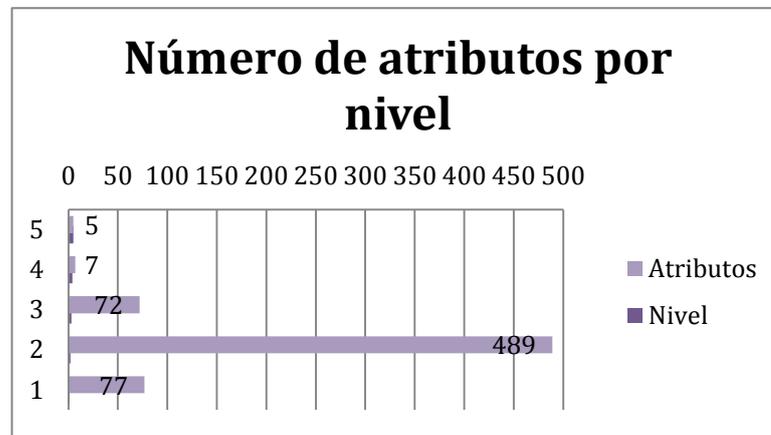
El detalle de este proceso se detalla a continuación:

P1.- ¿Cuáles son los elementos que estructuran los modelos de calidad de software?

Respuesta.- De la lectura de los 20 artículos, fue posible obtener hasta 5 niveles de jerarquía con un total de 650 atributos de calidad mostrados en la "Tabla resumen", siendo el "Nivel 1" de la fila "Nombre de cada nivel", el nivel más abstracto de la estructura.

Dicho esto, 77 atributos pertenecieron al nivel 1; 489 al nivel 2, siendo el nivel con mayor cantidad de atributos; 72 al nivel 3; 7 al nivel 4 y finalmente 5 al nivel 5 (ver Figura 6.1).

Figura 6.1 Número de atributos por nivel

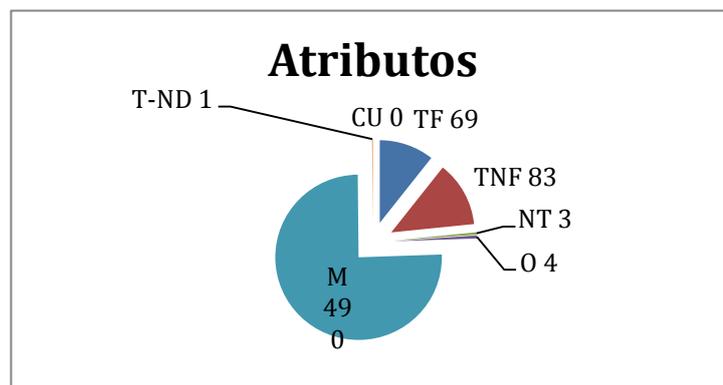


Fuente: Anexo Análisis de la SLR

Elaborado: Ana Villalta

De los atributos mostrados, el 75% fueron del tipo métricas, mientras que un 11% fueron de tipo "Técnico funcionales"; un 13% de tipo "Técnico no funcionales" y un 1% de tipo otros (ver Figura 6.2)

Figura 6.2 Numero de tipos de atributos encontrados

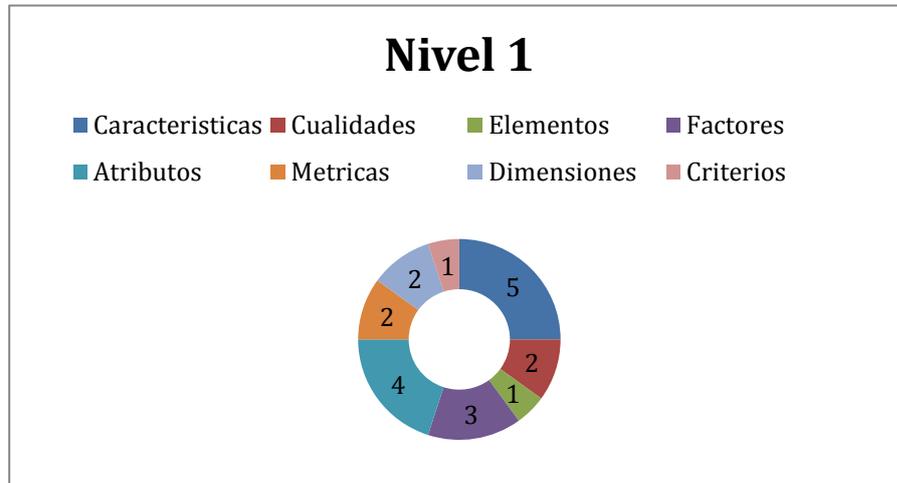


Fuente: Anexo Análisis de la SLR

Elaborado: Ana Villalta

De los 20 artículos, todos describieron factores de calidad en un primer nivel, y los autores los nombraron de la siguiente manera (ver Figura 6.3):

Figura 6.3 Nombre dado al nivel 1 por artículo

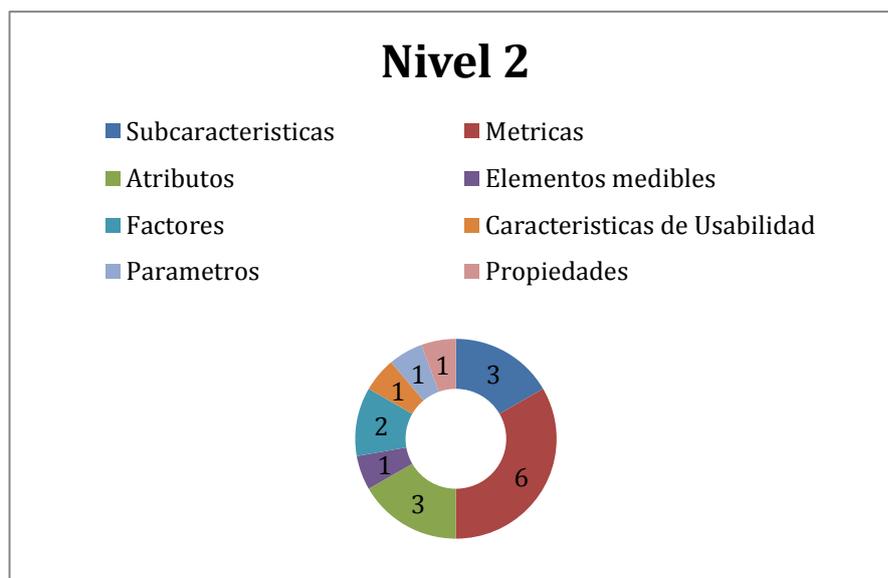


Fuente: Anexo Análisis de la SLR

Elaborado: Ana Villalta

Dieciocho artículos presentaron factores de calidad en un segundo nivel, y los autores los nombraron de la siguiente manera (ver Figura 6.4):

Figura 6.4 Nombre dado al nivel 2 por artículo

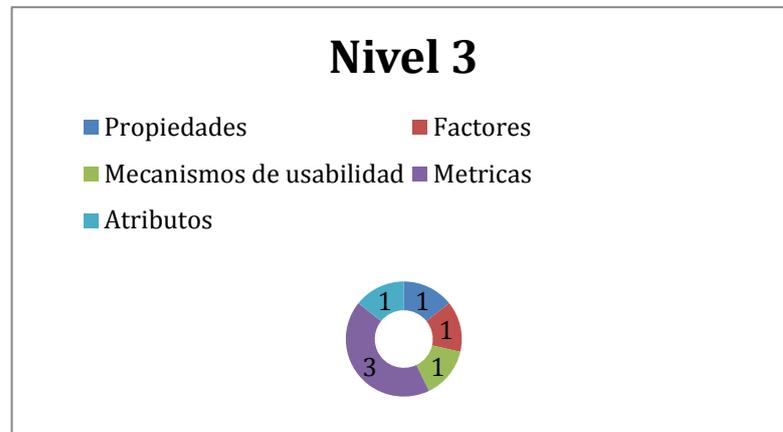


Fuente: Anexo Análisis de la SLR

Elaborado: Ana Villalta

Siete artículos presentaron factores de calidad en un tercer nivel, y los autores los nombraron de la siguiente manera (ver Figura 6.5):

Figura 6.5 Nombre dado al nivel 3 por artículo

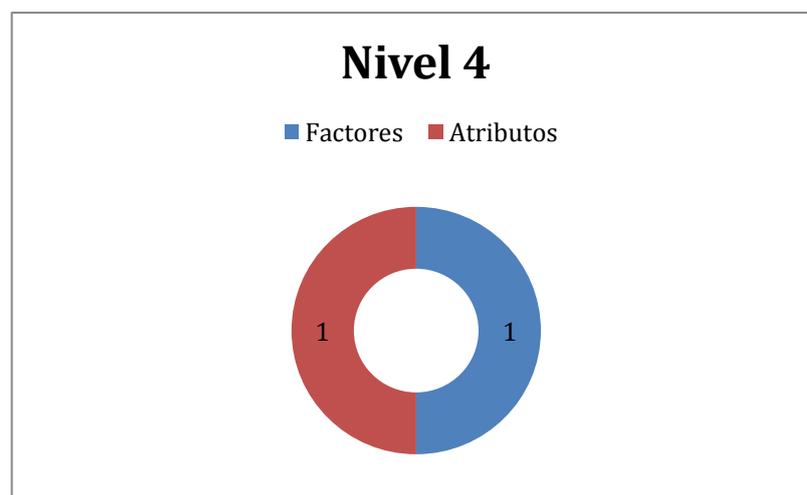


Fuente: Anexo Análisis de la SLR

Elaborado: Ana Villalta

Dos artículos presentaron factores de calidad en un cuarto nivel, y los autores los nombraron de la siguiente manera (ver Figura 6.6):

Figura 6.6 Nombre dado al nivel 4 por artículo



Fuente: Anexo Análisis de la SLR

Elaborado: Ana Villalta

Un artículo presentó factores de calidad en un quinto nivel, a los cuales los autores lo nombraron como "Factores" (ver Figura 6.7):

Figura 6.7 Nombre dado al nivel 5 por artículo



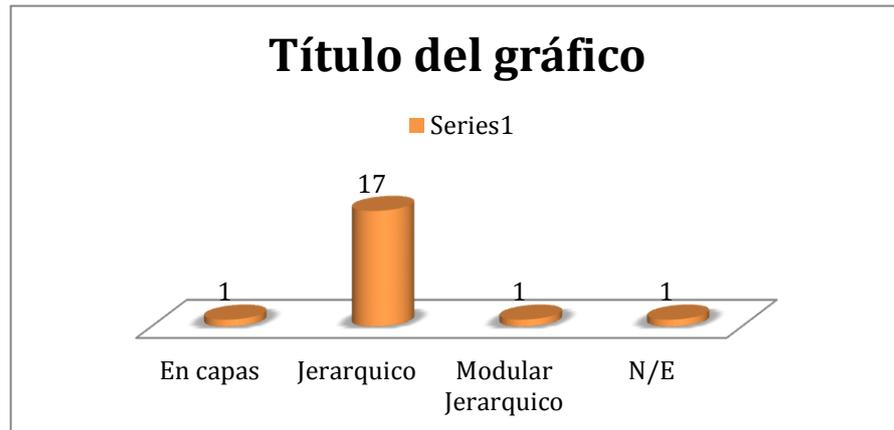
Fuente: Anexo Análisis de la SLR

Elaborado: Ana Villalta

P2.- ¿Cómo se relacionan los distintos elementos entre sí?

Respuesta.- De los 20 artículos, 17 de ellos identificaron a la relación entre los elementos como jerárquica; 1 como modular jerárquica, 1 como en capas y 1 no específica (ver Figura 6.8)

Figura 6.8 Tipos de relaciones entre los distintos elementos



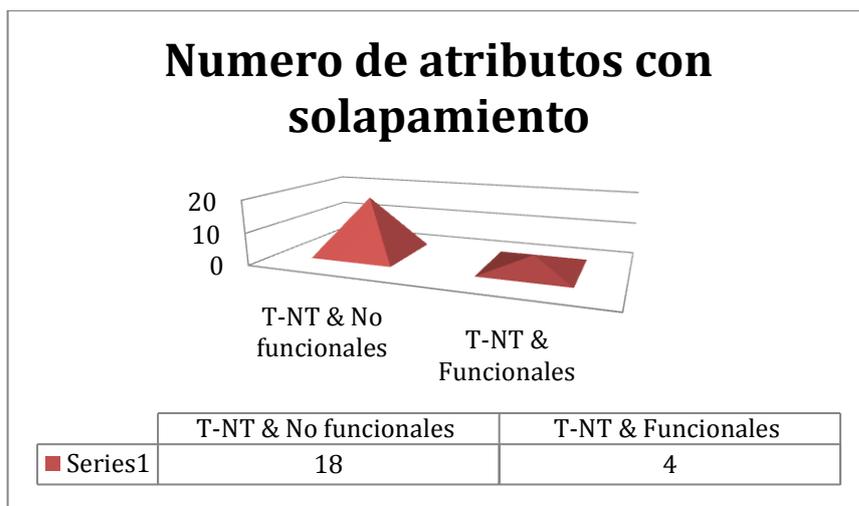
Fuente: Anexo Análisis de la SLR

Elaborado: Ana Villalta

P3.- ¿Cuáles son las principales propiedades de los modelos de calidad del software que han sido descritas en la literatura?

Respuesta.- Dos artículos presentaron la propiedad de solapamiento, los cuales se detallan en la “Tabla resumen”, fila “Overlapping of QF”. De estos se agruparon 18 factores “Técnicos - No técnicos y No funcionales” y 4 “Técnicos - No técnicos y Funcionales” (ver Figura 6.9)

Figura 6.9 Tipos de atributos con propiedad de solapamiento



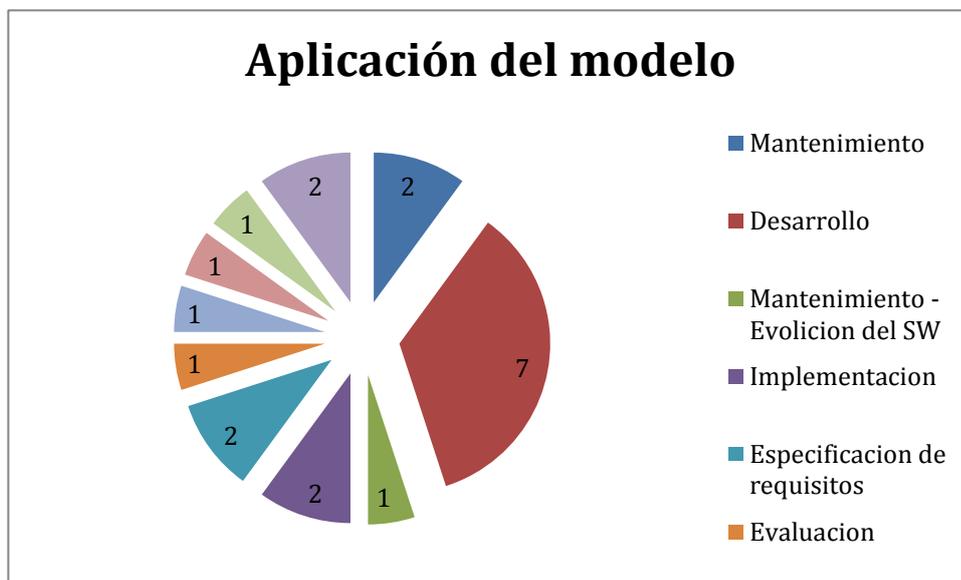
Fuente: Anexo Análisis de la SLR

Elaborado: Ana Villalta

P4. ¿Cuáles son las aplicaciones para las cuales han sido propuestos los modelos de calidad de software?

Respuesta.- Para la etapa de desarrollo han sido propuestos 7 de 20 modelos de calidad. El resto de modelos han sido propuestos para las siguientes aplicaciones (ver Figura 6.10):

Figura 6.10 Aplicaciones para las que se han descrito los modelos de calidad de software



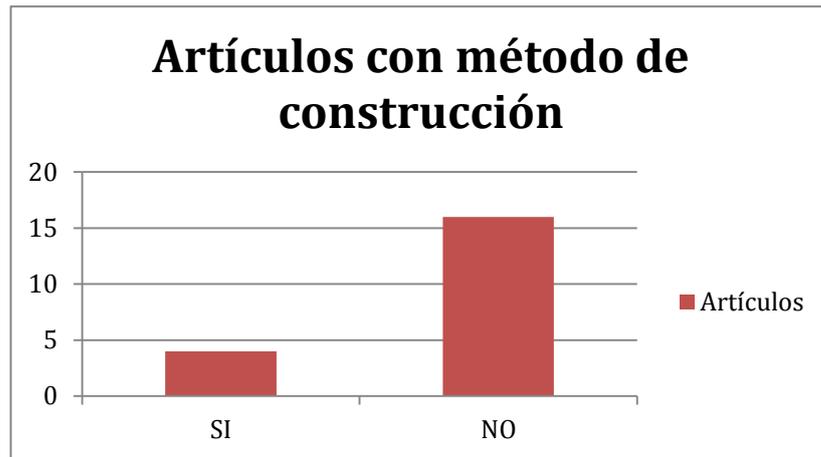
Fuente: Anexo Análisis de la SLR

Elaborado: Ana Villalta

P5. ¿Qué métodos se han utilizado para su construcción?

Respuesta.- En base a la tabla resumen se obtuvo que 16 artículos no presentan métodos de construcción mientras que, 4 de ellos sí. Los pasos de los métodos propuestos pueden ser observados en la "Tabla Resumen", columna pasos del método de construcción (ver Figura 6.11).

Figura 6.11 Artículos que describen un método de construcción



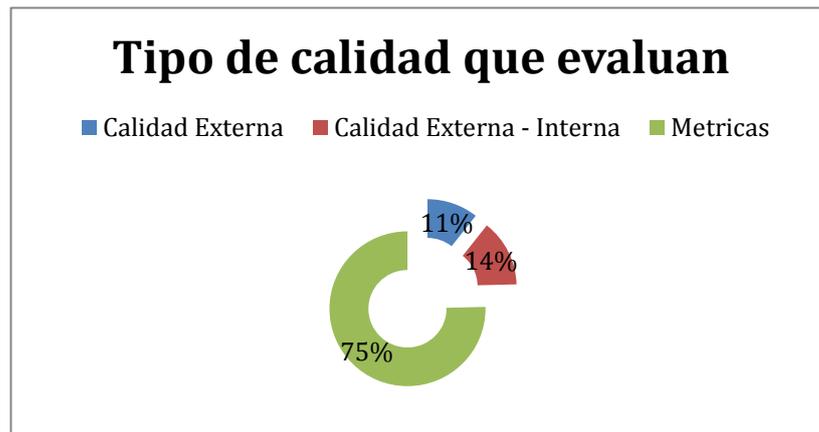
Fuente: Anexo Análisis de la SLR

Elaborado: Ana Villalta

P6. ¿Qué tipo de calidad evalúan, interna, externa, en uso y qué elementos se relacionan a estos tipos?

Respuesta.- Se han diferenciado 69 atributos funcionales que evalúan la calidad externa; 91 atributos no funcionales que están divididos entre evaluar la calidad interna y externa y 490 métricas (ver Figura 6.12).

Figura 6.12 Tipo de calidad que evalúan los factores de calidad



Fuente: Anexo Análisis de la SLR

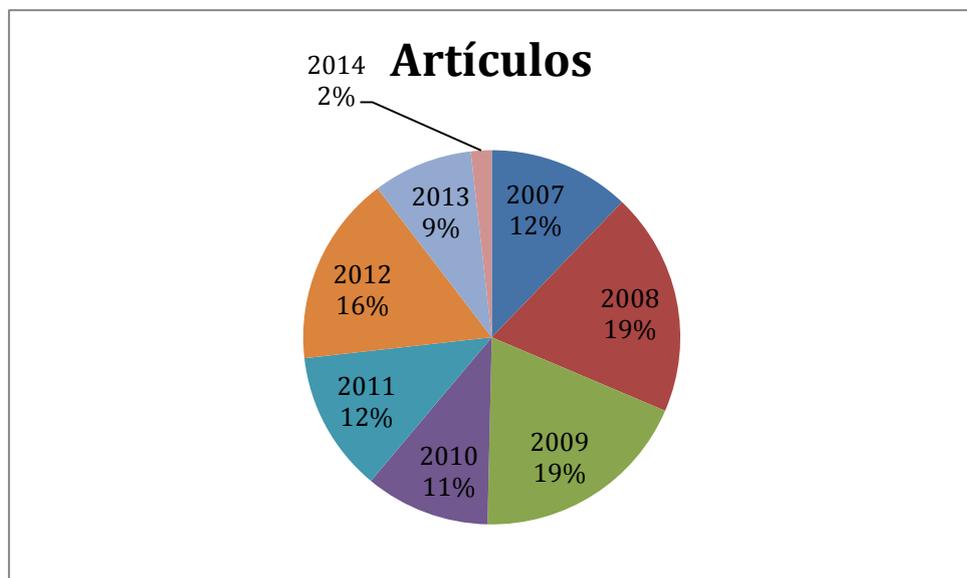
Elaborado: Ana Villalta

6.3 Otros Datos

Actividad de investigación entre el 2007 al 2014 en IEEE.

En los artículos obtenidos de la última etapa de refinamiento en la base de datos IEEE (294 artículos), se observó que la mayor parte de actividad de investigación se dio en los años 2008 y 2009, con un 19% en cada caso. Mientras que en el año 2014 existió solamente un 2% (ver Figura 6.13).

Figura 6.13 Porcentaje de investigación por años



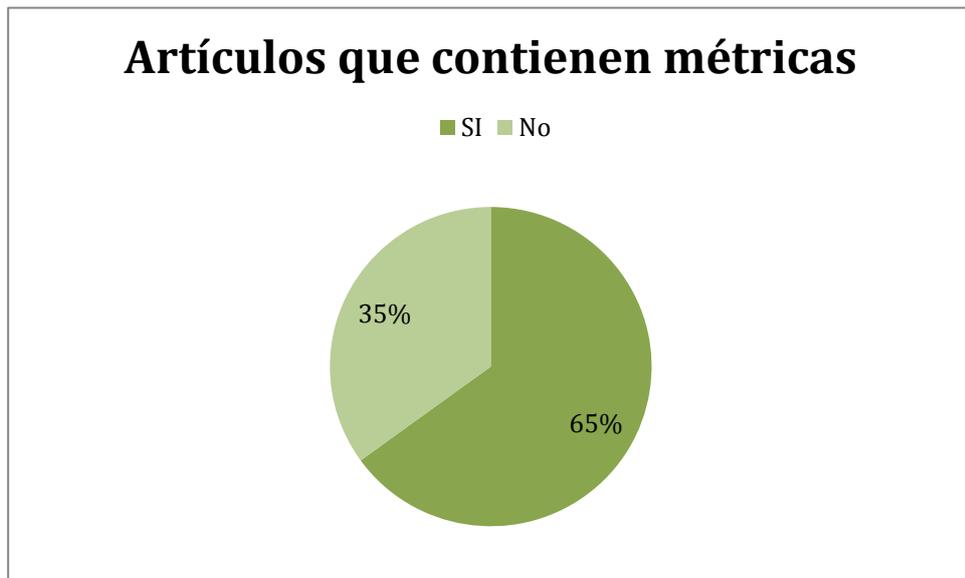
Fuente: Anexo Análisis de la SLR

Elaborado: Ana Villalta

Artículos que contengan métricas de calidad

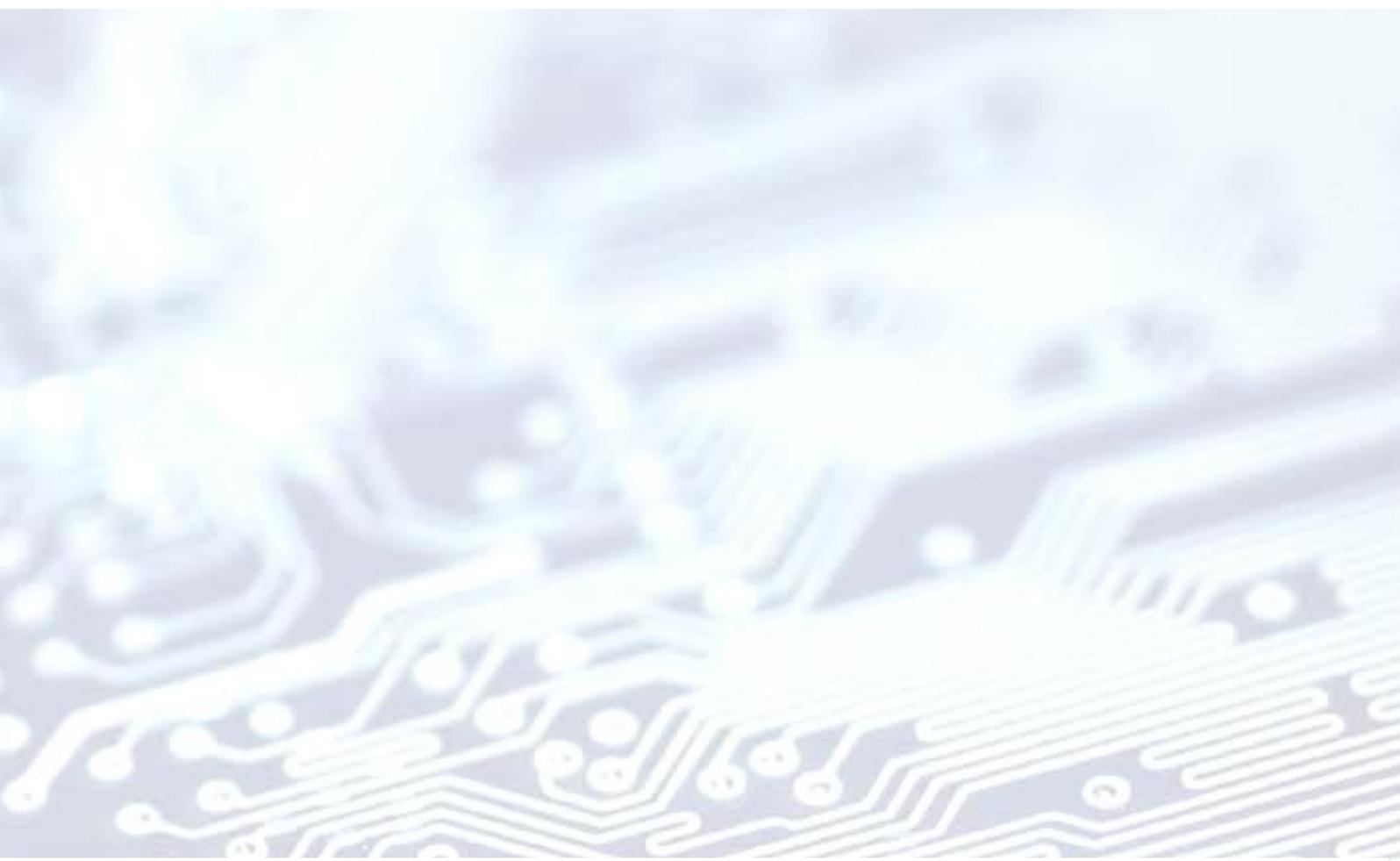
Se pudo observar, mediante los datos de la tabla resumen, que de los 20 artículos leídos 13 contienen métricas; mientras que, 7 de ellos no. El resultado fue un porcentaje de 65% y 35% respectivamente.

Figura 6.14 Porcentaje de artículos que contienen métricas de calidad de software



Fuente: Anexo Análisis de la SLR

Elaborado: Ana Villalta



CAPÍTULO 7. CONCLUSIONES



7.1 Conclusiones

El objetivo del software en cualquier rama es satisfacer la necesidad del usuario. Es aquí donde los Modelos de Calidad juegan un papel fundamental a la hora de construir un producto que, además de cumplir los requerimientos del interesado, plasme los estándares que la industria exige en todas las etapas a las que se enfrenta el programa.

El objetivo de este trabajo fue identificar la literatura que reúna los últimos aportes en cuanto a modelos de calidad de software, tomando en cuenta sus elementos, propiedades y métodos de construcción. Para cumplir este objetivo se usó la técnica de Revisión Sistemática de la Literatura. Para la aplicación de la técnica se estudiaron varias guías y, luego de su análisis, se adoptaron las guías base de Kitchenham [K07].

Se elaboró un protocolo a seguir durante toda la investigación, el cual puede ser replicable en cualquier otra intervención relacionada con el tema estudiado. En el protocolo se planteó preguntas de investigación y para lograr responderlas se elaboró tablas individuales que recolectan los datos más importantes artículo por artículo, para concluir con una tabla resumen que responde a manera de síntesis a cada una de las preguntas establecidas.

Se revisaron 3 bases de datos digitales y 3 revistas que contenían las publicaciones comprendidas entre el año 2007 y 2014. A partir de una población de 49935 documentos de la base de datos IEEE, se seleccionaron 164 artículos de los cuales se leyó su resumen y conclusiones para posteriormente ubicar a 66 de ellos en el formulario de preguntas de investigación. A 41 de los 66 artículos se les dio lectura completa y se identificó 20 documentos que proporcionaron evidencia empírica del tema tratado.

Los artículos leídos reúnen los distintos elementos de los modelos de calidad estudiados, los tipos a los que pertenecen estos elementos, e incorporan un aporte sustancial en métricas de calidad.

La aplicación de la Investigación empírica en el área de Ingeniería en Software permitió que se redacte un -paso a paso- de lo que constituye una Revisión Sistemática, logrando profundizar los conceptos de la materia de Calidad de Software.

7.2 Lecciones Aprendidas

Una importante motivación de esta investigación fue describir, desde la experiencia de los investigadores, las lecciones que podrían ayudar a las personas que no han incursionado en la investigación y quieren conducirse hacia una SLR siguiendo las guías de B. Kitchenham:

Lección 1. Analizar las guías adecuadas.

Problema: Al querer realizar una primera revisión sistemática de manera independiente, los recursos que se suelen utilizar como guía son otros trabajos de SLR similares. En un primer intento y, al conocer estos recursos, lo que se crea es un mapeo informal de la literatura que conduce a un sinfín de incumplimientos. Una vez puesta en marcha la investigación y, verificando que no se puede llevar a cabo un estudio de esta manera, se suele recurrir a guías que cuentan con fases que hablan sobre la estructura de una SLR. Elegir la correcta pueda ser una cuestión de azar. Esto puede llevar a que el investigador escoja la guía equivocada. Además, la mayoría de estas se encuentran enfocadas al área de la salud y llevan a un intento erróneo de querer imitarla en la Ingeniería de Software.

Solución: La solución que se propone es el análisis de guías base que proyecten a la conducción de una SLR formal, mediante la conformación de un protocolo. Esto es abordable, utilizando las Guidelines de Bárbara Kitchenham en donde se describe las bases que ayudaron a realizar un protocolo correcto en el área de Ingeniería en Software (Capítulo 2, Planificación de la revisión, Protocolo [K07]).

Lección 2. Consultar con un experto en el área al realizar las preguntas de investigación.

Problema: Como estudiantes de ingeniería existen muchos enfoques que pueden resultar interesantes. Dar lectura a la materia relacionada con el tema es de gran ayuda pero, no resuelve problemas que un experto en el campo ha vivido gracias a las experiencias y la práctica. De esta lectura pueden surgir infinidad de preguntas que, aunque sean interesantes, no suelen contar con el enfoque y alcance necesario.

Solución: Las preguntas deben ser hechas y estudiadas desde el principio con un experto. Además se debe buscar términos alternativos y sinónimos posibles para cada palabra clave que surja de cada pregunta de investigación

Lección 3. Realizar una sola cadena de búsqueda

Problema: Al realizar varias cadenas de búsqueda en una prueba piloto se observa que la duplicidad de los artículos puede ser elevada. En esta investigación, el total de los resultados en IEEE, sin aplicar ningún tipo de refinamiento, fue de 49935 trabajos relacionados. Las cadenas utilizadas fueron:

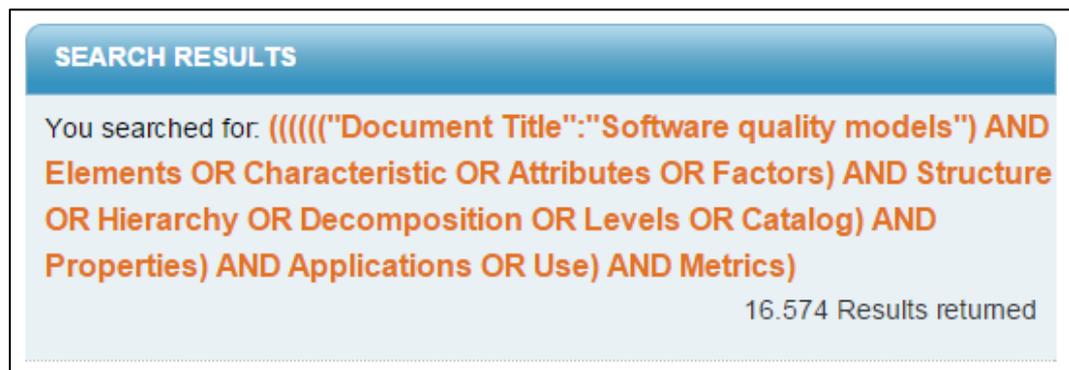
- “Software quality models”
- “Software quality models” AND elements
- “Software quality models” and attributes
- “Software quality models” and characteristics
- “Software quality models” and factors
- “Software quality models” and properties
- “Software quality models” and applications
- “Software quality models” and use
- “Software quality models” and metrics

Solución: Lo que se propone es realizar una sola cadena, utilizando operadores lógicos que permitan cubrir todas las palabras clave seleccionadas. Tomando en cuenta lo dicho, se establece un ejemplo de lo que según el análisis realizado debería hacerse.

La cadena que se propone incorporar en la búsqueda debe ser usada enlazando las seis listas nombradas con el operador AND. El total de artículos relacionados sin aplicar ningún tipo de refinamiento se resume en 16.574 trabajos (ver Figura 7.1).

- "Software quality models"
- Elements OR Characteristic OR Attributes OR Factors
- Structure OR Hierarchy OR Decomposition OR Levels OR Catalog
- Properties
- Applications OR Use
- Metrics

Figura 7.1 Resultado de la búsqueda aplicando una sola cadena en IEEE



Fuente: IEEE

Lección 4. Realizar un estudio de cada base de datos digital.

Problema: Incorporar la misma estructura de cadena de búsqueda en todas las bases de datos digitales seleccionadas puede ocasionar una pérdida importante de artículos.

Solución: Es importante realizar un estudio de cómo cada base de datos estructura y ordena las búsquedas, ya que los resultados obtenidos no serán los precisos si no se comprende cómo cada una de ellas trabaja y las diferentes opciones presentan.

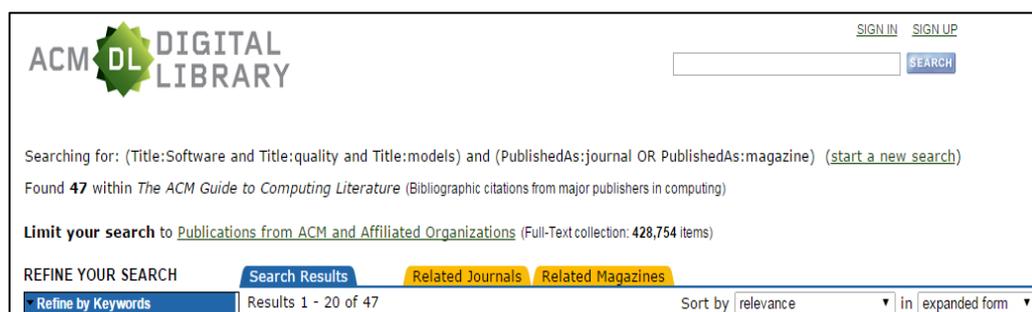
Se recomienda utilizar las estructuras de búsqueda avanzada. Generalmente las bases digitales cuentan con una explicación clara de cómo hacer uso de esta opción. A continuación se presenta un ejemplo de lo que se considera, se debe hacer.

Presumiendo que se quiera buscar literatura relacionada con modelos de calidad de software y que dichas palabras clave se encuentren en el título del documento en journals o revistas, la cadena de prueba que se propone ingresar en ACM Digital Library es:

- (Title:Software and Title:quality and Title:models) and (PublishedAs:journal OR PublishedAs:magazine)

Han resultado 47 artículos que responden a las especificaciones antes mencionadas (ver Figura 7.2).

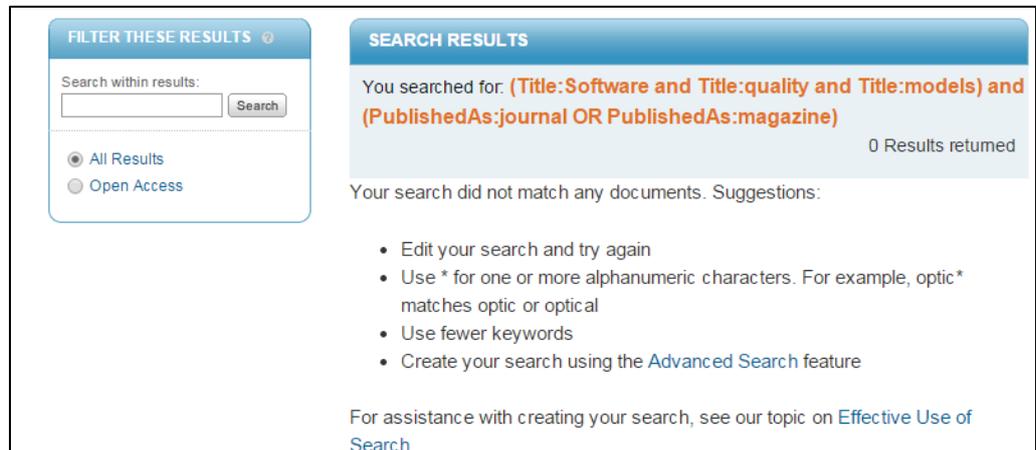
Figura 7.2 Búsqueda en ACM Digital Library



Fuente: ACM

Ahora si ingresamos la misma cadena en IEEE se puede observar que el número de artículos que lanza la búsqueda es cero (ver Figura 7.3)

Figura 7.3 Búsqueda en IEEE utilizando la misma cadena que en ACM Digital Library



Fuente: IEEE

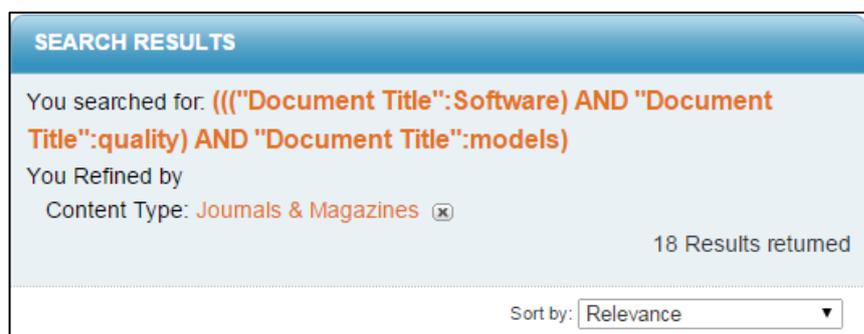
Elaborado: Ana Villalta

Al ingresar en IEEE la cadena:

- (((("Document Title":Software) AND "Document Title":quality) AND "Document Title":models)

Seleccionando la opción de Journals y Magazines en la búsqueda avanzada se obtienen los siguientes resultados (ver Figura 7.4):

Figura 7.4 Búsqueda en IEEE



Fuente: IEEE

7.3 Trabajos Futuros

Se dejó abierta la investigación en las bases de datos SPRINGER y ACM. En una futura investigación se puede comenzar la lectura de los artículos clasificados como Seleccionados y continuar a la etapa de extracción de datos hasta terminar el estudio.

Otra opción que se propuso en el capítulo 5 es la ponderación de la importancia de los estudios individuales. En donde se partiría desde la etapa selección de los estudios, de tal manera que:

- Los artículos clasificados como dudosos o falsos podrán servir como base para próximos estudios que continúen esta SLR y evalúen una mayor cantidad de trabajos con mayor profundidad.
- Los estudios clasificados como dudosos podrán ser llevados a estudios seleccionados o falsos.
- Los estudios falsos podrán ser llevados a falsos positivos o falsos negativos
- Los falsos negativos podrán ser llevados a positivos, si fuera el caso.

BIBLIOGRAFÍA

[RH08] Runeson, P., Host, M. (2008, diciembre 19). Guidelines for conducting and reporting case study research in software engineering. *Empir Software Eng (2009)*, 1, 34. 2014, julio, De Springer Base de datos

[K07] Kitchenham, B. (2007, julio 7). Guidelines for performing Systematic Literature Reviews in Software Engineering. *EBSE Technical Report*, 1, 65. 2014, julio, De SPRINGER Base de datos.

[EA06] Easterbrokk, S., Aranda, J. (2006). Case Studies for Software Engineers. *28th International Conference on Software Engineering*, 1, 68. 2014, diciembre, De IEEE Base de datos.

[BGCH05]Biolchini, J., Gomes P., Cruz, A., & Horta, G. (2005, mayo). Systematic Review in Software Engineering. Technical Report, 1, 31. 2014, Julio, De Google Académico Base de datos.

[MMA10]Medina, C., Marín, J., & Alfalla, R. (2010). Una propuesta metodológica para la realización de búsquedas sistemáticas de la bibliografía. *Working Papers on Operations Management*, 1, 18. 2014, Julio, De Google Académico Base de datos.

[BBHRS06]Beecham, S., Baddoo, N., Hall, T., Robinson, H., & Sharp, H. (2006, Septiembre). Protocol for a Systematic Literature Review of Motivation in Software Engineering. *Information and Software Technology*, 50, 87. 2014, Septiembre, De Google Académico Base de datos.

[CHHK13] Carver, J., Hassler, E., Hernandez, E., & Kraft, N. (2013). Identifying Barriers to the Systematic Literature Review Process. *Empirical Software Engineering and Measurement*, ND, 11. 2014, Julio, De IEEE Base de datos.

[CFQ10] Carvallo, J., Franch, X., & Quer, C . (2010). CALIDAD DE COMPONENTES SOFTWARE . En *Calidad del Producto y Proceso Software*(30). España: RA-MA Editorial.

[A14] Ayala, C. (2014). Empirical Software Engineering. Octubre 27, 2014, de Group of Software and Service Engineering Sitio web: <https://dl.dropboxusercontent.com/u/589179/Material%20Ecuador/CursoEmpiricalSoftwareEngineering-ClaudiaEcuador2014-Session1-Basics.pdf>

[ISO 8402] INTERNATIONAL STANDARDS ORGANIZATION (1986). ISO International Standard 8402: Quality Management and Quality Assurance Vocabulary.

[ISO 25000] Norma ISO 25000. (2014). Normas ISO 25000. 2015, de ISO 25000 Sitio web: <http://iso25000.com/index.php/normas-iso-25000>

[Good96] Goodman C. Literature Searching and Evidence Interpretation for Assessing Health Care Practices. SBU, Stockholm, 1996

[DO07] Diaz, A., Ortega, M. (2007). Catálogo de Atributos de Calidad para la Evaluación de Sistemas Basados en Componentes. 2014, Julio, de Universidad del Azuay Sitio web: dspace.uazuay.edu.ec

[CKV09] Correia, J., Kanellopoulos, Y., & Visser, J. (2009, Septiembre 20-26). A survey-based study of the mapping of system properties to ISO/IEC 9126 maintainability characteristics. *Software Maintenance, 2009. ICSM 2009. IEEE International Conference on*, 1, 10. 2014, Septiembre, De IEEE Base de datos.

[OBL12] Orenyi, B., Basri, S., & Low Tan, J. (2012, Noviembre 26-28). Object-Oriented Software Maintainability Measurement in the Past Decade. *Advanced Computer Science Applications and Technologies (ACSAT), 2012 International Conference on*, 1, 6. 2014, Septiembre, De IEEE Base de datos.

[FKW12] Franke, D., Kowalewski, S., & Weise, C. (2012, agosto 27-29). A Mobile Software Quality Model. *Quality Software (QSIC), 2012 12th International Conference on*, 1, 4. 2014, Septiembre, De IEEE Base de datos.

[BC09] Breivold, H., Crnkovic, I. (2009, Agosto 27-29). Analysis of Software Evolvability in Quality Models. *Software Engineering and Advanced Applications, 2009. SEAA '09. 35th Euromicro Conference on*, 1, 4. 2014, Septiembre, De IEEE Base de datos.

[SLHF09] Siwei, P., Limin, S., Hui, L., & Feng, L. (2009, abril 2). User-Oriented Measurement of Software Flexibility. *Computer Science and Information Engineering, 2009 WRI World Congress on*, 7, 5. 2014, Septiembre, De IEEE Base de datos.

[BGL10] Bogado, V., Gonnet, S., & Leone, H. (2010, Noviembre 15-19). An Approach Based on DEVS for Evaluating Quality Attributes. *Chilean Computer Science Society (SCCC), 2010 XXIX International Conference of the*, 1, 9. 2014, Septiembre, De IEEE Base de datos.

[MTKF09] Mueller, C., Tamir, D., Komogortsev, O., & Feldman, L. (2009, julio 20-24). An Economical Approach to Usability Testing. *Computer Software and Applications Conference, 2009. COMPSAC '09. 33rd Annual IEEE International*, 1, 6. 2014, Septiembre, De IEEE Base de datos.

[AW09] Alsultanny, Y., Wohaishi, A. (2009, diciembre 28-30). Requirements of Software Quality Assurance Model. *Environmental and Computer Science, 2009. ICECS '09. Second International Conference on*, 1, 5. 2014, Septiembre, De IEEE Base de datos.

[JMS07] Juristo, N., Moreno, A., & Sanchez-Segura, M.-I. (2007, noviembre). Guidelines for Eliciting Usability Functionalities. *Software Engineering, IEEE Transactions on*, 33, 15. 2014, septiembre, De IEEE Base de datos.

[BF09] Bombardieri, M., Fontana, F. (2009, mayo 16). Software aging assessment through a specialization of the SQuaRE quality model. *Software Quality, 2009. WOSQ '09. ICSE Workshop on*, 1, 6. 2014, Septiembre, De IEEE Base de datos.

[ASAM12] Abdi, A., Souzani, A., Amirfakhri, M., Moghadam, A. (2012, noviembre 6-8). Using security metrics in software quality assurance process. *Telecommunications (IST), 2012 Sixth International Symposium on*, 1, 4. 2014, Septiembre, De IEEE Base de datos.

[P09] Perin, F. (2009, octubre). Enabling the Evolution of J2EE Applications through Reverse Engineering and Quality Assurance. *Reverse Engineering, 2009. WCRE '09. 16th Working Conference on*, 1, 4. 2014, septiembre, De IEEE Base de datos.

[AFMS09]Alonso, F., Fuertes, J., Martinez, L., & Soza, H. (2009, noviembre). Towards a set of Measures for Evaluating Software Agent Autonomy. *Artificial Intelligence, 2009. MICA I 2009. Eighth Mexican International Conference on, 1, 6*. 2014, Septiembre, De IEEE Base de datos.

[ORCDW08]Oliveira, M., Redin, R., Carro, L., da Cunha Lamb, L., Wagner, F. (2008, abril). Software Quality Metrics and their Impact on Embedded Software. *Model-based Methodologies for Pervasive and Embedded Software, 2008. MOMPES 2008. 5th International Workshop on, 1, 10*. 2014, septiembre, De IEEE Base de datos.

[BLL09] Barkmann, H., Lincke, R., Lowe, W. (2009, mayo). Quantitative Evaluation of Software Quality Metrics in Open-Source Projects. *Advanced Information Networking and Applications Workshops, 2009. WAINA '09. International Conference on, 1, 6*. 2014, septiembre, De IEEE Base de datos.

[CK09]Chanwook Kim., Keun Lee. (2009, agosto). Software Quality Model for Consumer Electronics Product. *Quality Software, 2009. QSIC '09. 9th International Conference on, 1, 6*. 2014, septiembre, De IEEE Base de datos.

[CA12] Chotjaratwanich, U., Arpnikanondt, C. (2012, diciembre). A Visualization Technique for Metrics-Based Hierarchical Quality Models. *Software Engineering Conference (APSEC), 2012 19th Asia-Pacific, 1, 4*. 2014, septiembre, De IEEE Base de datos.

[KKMS07]Kozlov, D., Koskinen, J., Markkula, J., & Sakkinen, M. (2007, septiembre). Evaluating the Impact of Adaptive Maintenance Process on Open Source Software Quality. *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on, 1, 10*. 2014, septiembre, De IEEE Base de datos.

[MSI11] Munawar, M., Salim, N., & Ibrahim, R. (2011, diciembre). Towards Data Warehouse Quality through Integrated Requirements Analysis. *Advanced Computer Science and Information System (ICACISIS), 2011 International Conference on, 1, 6*. 2014, Septiembre, De IEEE Base de datos.

[KJYQ13] Ke Li., Junchao Xiao., Yongji Wang., & Qing Wang. (2013, julio). Analysis of the Key Factors for Software Quality in Crowdsourcing Development: An Empirical Study on TopCoder.com. *Computer Software and Applications Conference (COMPSAC), 2013 IEEE 37th Annual, 1, 6*. 2014, septiembre, De IEEE Base de datos.