

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

PROPUESTA DE LINEAMIENTOS PARA LA EVALUACIÓN DE LOS PRODUCTOS DE SOFTWARE PRESENTADOS EN LA FACULTAD DE INGENIERIA EN SISTEMAS

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA
EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN DE LA ESCUELA
POLITÉCNICA NACIONAL**

SAMANTHA MICHELLE DE LA CRUZ BANEGAS

samy_samantha_5@hotmail.com

EDISON FERNANDO TAMAYO ROMERO

eddyferch@hotmail.es

Director: PAMELA CATHERINE FLORES NARANJO, Ph.D.

pamela.flores@epn.edu.ec

CODIRECTOR: EDISON FERNANDO LOZA AGUIRRE, Ph.D.

edison.loza@epn.edu.ec

Quito, 2022

APROBACIÓN DEL DIRECTOR

Como director del trabajo de titulación “Propuesta de lineamientos para la evaluación de los productos de Software presentados en la Facultad de Ingeniería en Sistemas” desarrollado por Samantha Michelle De la Cruz Banegas y Edison Fernando Tamayo Romero, estudiantes de la Facultad de Ingeniería en Sistemas y Ciencias de la Computación, habiendo supervisado la realización de este trabajo y realizado las correcciones correspondientes, doy por aprobada la redacción final del documento escrito para que prosiga con los trámites correspondientes a la sustentación de la defensa oral.

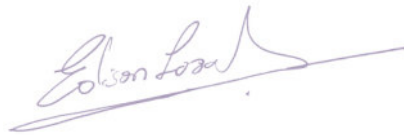


Pamela Catherine Flores Naranjo

DIRECTOR

APROBACIÓN DEL CODIRECTOR

Como director del trabajo de titulación “Propuesta de lineamientos para la evaluación de los productos de Software presentados en la Facultad de Ingeniería en Sistemas” desarrollado por Samantha Michelle De la Cruz Banegas y Edison Fernando Tamayo Romero, estudiantes de la Facultad de Ingeniería en Sistemas y Ciencias de la Computación, habiendo supervisado la realización de este trabajo y realizado las correcciones correspondientes, doy por aprobada la redacción final del documento escrito para que prosiga con los trámites correspondientes a la sustentación de la defensa oral.



Edison Fernando Loza Aguirre

CODIRECTOR

DECLARACIÓN DE AUTORÍA

Yo, Samantha Michelle De la Cruz Banegas, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

A handwritten signature in black ink, appearing to read 'Samantha', with several horizontal lines drawn over it for emphasis or as a flourish.

Samantha Michelle De la Cruz Banegas

DEDICATORIA

A mis padres y a mi hermano, que han pasado junto a mí, al igual que mi familia.

Samantha De la Cruz

AGRADECIMIENTO

A mis padres, hermano, a mi directora de proyecto de titulación y codirector que brindaron todo su apoyo para el desarrollo y culminación de este proyecto.

Samantha De la Cruz

DECLARACIÓN DE AUTORÍA

Yo, Edison Fernando Tamayo Romero, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



Edison Fernando Tamayo Romero

DEDICATORIA

A mis padres, que han sido un ejemplo de determinación y esfuerzo, con sus enseñanzas he podido llegar hasta donde estoy el día de hoy y mirarlos con gran orgullo y amor.

A mi esposa que ha estado a mi lado apoyándome en todo momento y ha sido un verdadero ejemplo de disciplina y amor.

Edison F. Tamayo

AGRADECIMIENTO

A todas las maravillosas personas que han aportado en mi vida sabiduría, conocimiento, amistad y cariño, especialmente a mi esposa, mis padres y hermanos que han sido desde siempre la razón por la cual he buscado ser siempre una mejor persona.

Edison F. Tamayo

Índice de Contenido

LISTA DE FIGURAS	i
LISTA DE TABLAS	ii
LISTA DE ANEXOS	iii
Resumen	V
<i>Abstract</i>	VI
1. Introducción	1
1.1. Pregunta de investigación.....	2
1.2. Objetivo general	2
1.3. Objetivos específicos.....	2
1.4. Alcance.....	2
1.5. Marco teórico.....	3
1.5.1. Evaluación del producto de software.....	3
1.5.2. Estándares utilizados para la evaluación de software	6
1.5.3 Características de software.....	8
1.5.4. Frameworks	11
1.5.5. Pruebas realizadas al evaluar el software	14
1.5.6. Proceso para la evaluación de software.....	16
2. Metodología.....	17
2.1. Revisión sistemática de literatura.....	17
2.2. Análisis documental	24
3. Resultados y discusión	27
3.1 Resultados de la revisión sistemática de literatura.....	27
3.1.1 Estándares más utilizados en los artículos analizados.....	27
3.1.2 Frameworks más utilizados en los artículos analizados	28
3.1.3 Características más utilizadas en los artículos analizados.....	29
3.1.4 Pruebas más utilizadas en los artículos analizados	30
3.2. Análisis documental	31
3.2.1 Análisis de proyectos de titulación de la facultad de ingeniería en sistemas	31
3.2.2 Frameworks utilizados	36
3.2.3 Métricas de evaluación de software	37
3.2.4 Estándares utilizados	38
3.3 Comparación de resultados obtenidos entre papers y proyectos de titulación	38

3.3.1 Herramientas de evaluación de software	38
3.3.2 Frecuencias de elementos encontrados.....	38
3.4 Discusión	41
3.5 Lineamientos para la evaluación de productos de software	45
3.6 Herramientas recomendadas para la evaluación de las características del producto de software recomendadas.....	48
3.7 Estándares recomendados para evaluación de productos de software	54
4.Conclusiones y recomendaciones.....	55
4.1 Conclusiones	55
4.2 Recomendaciones	56
Referencias bibliográficas.....	57
ANEXOS.....	60

LISTA DE FIGURAS

Figura 1.1: Estándares más utilizados para evaluación de software (AENOR, 2005).....	7
Figura 1.2 : Línea de tiempo de frameworks que permiten realizar la evaluación de software	14
Figura 1.3: Proceso para la evaluación de software según ISO 14598	16
Figura 2.1: Descripción del proceso realizado para realizar una Investigación documental	17
Figura 2.2: Combinaciones de la SS1	19
Figura 2.3: Selección de Documentos en SLR.....	21
Figura 2.4: Estructura del código	23
Figura 2.5: Selección de documentos para análisis documental.	25
Figura 2.6: Tipo de información extraída.....	25
Figura 3.1: Gráfica de la relación de números de papers con respecto a los códigos establecidos.....	27
Figura 3.2: Estándares más utilizados en los artículos analizados.....	28
Figura 3.3: Frameworks más utilizados en los artículos analizados	29
Figura 3.4: Características más utilizadas en los artículos analizados	30
Figura 3.5: Pruebas más utilizadas en los artículos analizados	30
Figura 3.6: Tipo de software desarrollado en la FIS	31
Figura 3.7: Tipo de evaluación de software.....	32
Figura 3.8: Tipos de pruebas	32
Figura 3.9: Herramientas utilizadas en pruebas	33
Figura 3.10: Herramientas utilizadas para pruebas funcionales	33
Figura 3.11: Descripción de las encuestas SUS y CSUQ respectivamente las cuales se utilizan para evaluar la usabilidad.	34
Figura 3.12: Herramientas para medir la aceptabilidad	34
Figura 3.13: Descripción de GTmetrix - Jmeter.....	35
Figura 3.14: Herramientas utilizadas para medir la accesibilidad	35
Figura 3.15: Frameworks más utilizados.....	37
Figura 3.16: Métricas de evaluación de software	37
Figura 3.17: Resumen de la discusión de resultados	44
Figura 3.18: Propuesta de lineamientos para la evaluación de productos de software.....	48
Figura 3.19: Interfaz gráfica de usuario JMeter	51

Figura 3.20: Alertas de WAVE sobre el uso de enlaces invisibles y atajos de teclado.	53
Figura 3.21: Resultados del uso de la herramienta WAVE.....	53

LISTA DE TABLAS

Tabla 1.1: Modelos de desarrollo de Software	4
Tabla 2.1: Criterios de búsqueda en SLR	19
Tabla 2.2: Cadena de búsquedas utilizadas en SLR.....	19
Tabla 2.3 : Número de documentos encontrados por cadena de búsqueda.....	20
Tabla 2.4: Tipos de documentos seleccionados.....	21
Tabla 2.5: Categorías para la evaluación de software.....	22
Tabla 2.6: Códigos para determinar las diferentes categorías al momento de evaluar software	23
Tabla 2.7: Criterios de Búsqueda en el Análisis Documental	24
Tabla 2.8: Formato de información extraída de los proyectos de titulación	26
Tabla 3.1: Tipos de evaluación de software con respecto al tipo de software desarrollado	36
Tabla 3.2: Comparación: Papers (ACM - IEEE) vs Proyectos de titulación	39
Tabla 3.3: Comparación entre códigos encontrados más frecuentes en los documentos de investigación y proyectos de titulación la EPN (Escuela Politécnica Nacional).....	40
Tabla 3.4: Comparación entre el total de códigos encontrados en los documentos científicos y proyectos de titulación EPN.....	40
Tabla 3.5: Métricas recomendadas	46
Tabla 3.6: Lineamientos de evaluación de software.....	46
Tabla 3.7: Ejemplo de un caso de prueba.....	49
Tabla 3.8: Cuestionario SUS.....	50
Tabla 3.9: Cuestionario UMUX	50
Tabla 3.10: Cuestionario UMUX-LITE.....	51

LISTA DE ANEXOS

Anexo I: Listado de Artículos de Investigación.....	60
Anexo II: Listado de Proyectos de titulación.....	60
Anexo III: Tabla de Códigos Artículos de Investigación.....	60
Anexo IV: Tabla de Códigos Proyectos de titulación.....	60

RESUMEN

El presente trabajo de titulación tiene la intención de proponer un conjunto de lineamientos para la evaluación de los productos de software desarrollados en los proyectos de titulación que se presentan en la Facultad de Ingeniería en Sistemas de la Escuela Politécnica Nacional. Con este fin, se realizó un estudio cualitativo dividido en dos etapas. En la primera de ellas se realizó una revisión sistemática de literatura de un total de 134 documentos académicos, de esta manera se obtuvo un conjunto de factores como las características para evaluar, los frameworks a utilizar, métricas, procesos, herramientas y pruebas realizadas, este conjunto de factores permitirán establecer un marco comparativo para los resultados de la siguiente etapa. La segunda etapa constó de un análisis documental exhaustivo, en el cual se analizó y evaluó los trabajos de titulación presentados en la FIS con el fin de determinar las tendencias para la evaluación de software y cuáles son las pruebas más comunes que se han realizado dependiendo del proyecto de software desarrollado, siendo las pruebas funcionales las más utilizadas. El análisis de esta documentación juntamente con los resultados de la primera etapa permitió obtener un conjunto de lineamientos de evaluación de software sugeridos, como se puede ver en la sección de resultados y discusión, estos lineamientos de evaluación de software permitirán realizar la evaluación de software aplicable para los proyectos de desarrollo de software venideros. Estos lineamientos de evaluación de software se concentran en la evaluación de cuatro características que prevalecieron coincidentemente tanto en los artículos científicos como en los trabajos de titulación: funcionalidad, usabilidad, accesibilidad y rendimiento.

Palabras clave: Evaluación, Software, Estándares, Pruebas, Metodologías de evaluación, Frameworks.

ABSTRACT

The present work proposes a software evaluation guideline for projects from the Facultad de Ingeniería en Sistemas of the Escuela Politécnica Nacional. We carried out a qualitative study divided in two stages. In the first stage, we carried out a systematic literature review over 134 scientific articles to learn about software evaluating mechanisms. We found frameworks, metrics, processes, tools, quality factors used for the evaluation process. With this information we develop a comparative framework to analyze the results in the next stage. In the second stage, we conducted an exhaustive documentary analysis about software testing techniques in the thesis projects from 2010 to 2020. We found that the most common technique was the functional testing. Additionally, we found that students did not use any standard about testing, but they recurrently used ISO25000 to evaluate software quality factors. As a result of the two stages, we could develop a software evaluation guideline based on the ISO25000 to cover functionality, usability, accessibility, and performance. This guideline also includes a list of recommendations and examples about how to perform software evaluation. We hope this guideline could be applicable for upcoming software development projects.

Keywords: Evaluation, Software, Standards, Tests, Evaluation methodologies

1. INTRODUCCIÓN

La Ingeniería de Software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software, desde las etapas iniciales de la especificación del sistema hasta el mantenimiento de éste después de que se utiliza (Sommerville, 2005). Su principal meta es el desarrollo costeable de sistemas de software. Para lograr esta meta, es necesario cumplir con un ciclo de vida de desarrollo; es decir, pasar por etapas de planificación, implementación, pruebas, documentación, despliegue y mantenimiento de un producto de software.

Un producto de software se refiere a la “unidad lógica de compartición y empaquetado de software que tiene un desarrollo gestionado, un ciclo de vida de mantenimiento y atributos visibles para el cliente” (IBM, 2002). Todo producto de software pasa por una evaluación con el fin de garantizar la calidad de éste, lo cual lleva a las preguntas ¿cuál es la forma adecuada de evaluar un producto de software?, ¿qué procesos se debería seguir para garantizar su calidad?, ¿qué características deben ser evaluadas? Para tener una idea más clara de las respuestas a estas preguntas, en el presente trabajo, analizaremos los artículos científicos publicados en las bibliotecas de ACM e IEEE, al igual que los trabajos de titulación en la Facultad de Ingeniería en Sistemas (FIS) en los últimos años; con el fin de proponer un conjunto de lineamientos para la evaluación de software aplicables a futuros proyectos de desarrollo de software.

En la FIS, desde 1985, se han desarrollado varios proyectos de software como forma de titulación, siendo algunos de estos proyectos con fines académicos, y otros tantos de interés comercial. Estos productos de software deben cumplir con una etapa de evaluación. Por la variedad de aproximaciones de desarrollo, y lineamientos de los directores de los proyectos, hasta la fecha se han utilizado diferentes tipos de pruebas para determinar si el producto desarrollado cumple con los requerimientos y estándares necesarios. Sin embargo, no existe una base de lineamientos de evaluación de software preestablecidos que sirvan como guía a seguir al estudiante en esta etapa de evaluación de software.

Es por esta razón que surge la necesidad de realizar una investigación sobre la evaluación de productos de software, para ello realizaremos una revisión sistemática de literatura (SLR), al igual que un análisis documental, con el fin de responder nuestra pregunta de investigación.

1.1. Pregunta de investigación

¿Qué métodos de evaluación de software, frameworks, estándares, procesos, características de software existen y cuáles se han utilizado en los trabajos de titulación presentados en la Facultad de Ingeniería en Sistemas y cuáles permitirán crear una base de lineamiento para el proceso de evaluación de productos de software?

1.2. Objetivo general

Proponer un conjunto de lineamientos para la evaluación de los productos de software desarrollados en los proyectos de titulación, a partir de un estudio de los proyectos presentados en los últimos 10 años en la Facultad de la Ingeniería en Sistemas de la Escuela Politécnica Nacional.

1.3. Objetivos específicos

- Realizar una revisión de literatura de los métodos existentes para evaluar software.
- Conducir un análisis documental de las prácticas de evaluación del software reportadas en los planes de titulación de la Facultad de Ingeniería en Sistemas en los últimos 10 años.
- Determinar las principales pruebas que se utilizan para la evaluación de software en los diferentes proyectos de titulación.

1.4. Alcance

En la FIS se desarrollan varios productos de software como proyectos de titulación, los cuales deben cumplir con la etapa de evaluación de software, para lo cual se utilizan distintos tipos de pruebas para determinar si el producto desarrollado cumple con los requerimientos y estándares esperados. Sin embargo, no existe una base de lineamientos de evaluación de software preestablecidos que sirvan como guía a seguir en esta etapa de evaluación de software, es por este motivo que hay una diversidad de metodologías utilizadas al momento de la evaluación de software.

El presente trabajo busca determinar los mejores parámetros a tomar en cuenta al momento de realizar una evaluación de un proyecto de titulación de desarrollo en la FIS de la Escuela Politécnica Nacional; mediante la investigación y comparativa de los trabajos publicados en bibliotecas académicas como IEEE y ACM, frente a los trabajos de titulación realizados en la facultad.

Este trabajo no involucra el desarrollo de ninguna herramienta que automatice la recolección y análisis de los trabajos que se estudiarán.

1.5. Marco teórico

Para comprender las implicaciones de evaluar software, es necesario en primer lugar que se definan algunos conceptos ligados a software, un producto de software “es una unidad lógica de compartición y empaquetado de software que tiene un desarrollo gestionado, un ciclo de vida de mantenimiento y atributos visibles para el cliente” (IBM, 2002).

1.5.1. Evaluación del producto de software

La evaluación de un producto de software se refiere a la acción que debe realizar el evaluador para obtener el resultado de una medición o verificación aplicada en un conjunto de componentes de un producto de software o en el producto en su conjunto (AENOR, 2005). Para evaluar un software es necesario tomar en cuenta varios aspectos como: estándares, metodología, frameworks, pruebas a realizar, características a ser evaluadas, quién va a ser el evaluador y qué proceso se va a seguir para realizar la evaluación del producto de software.

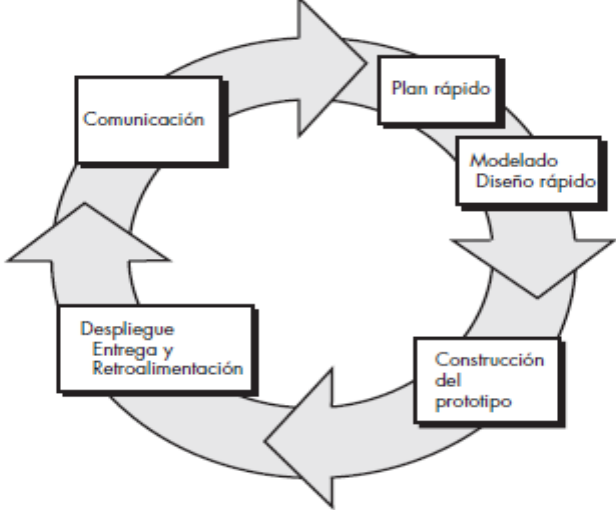
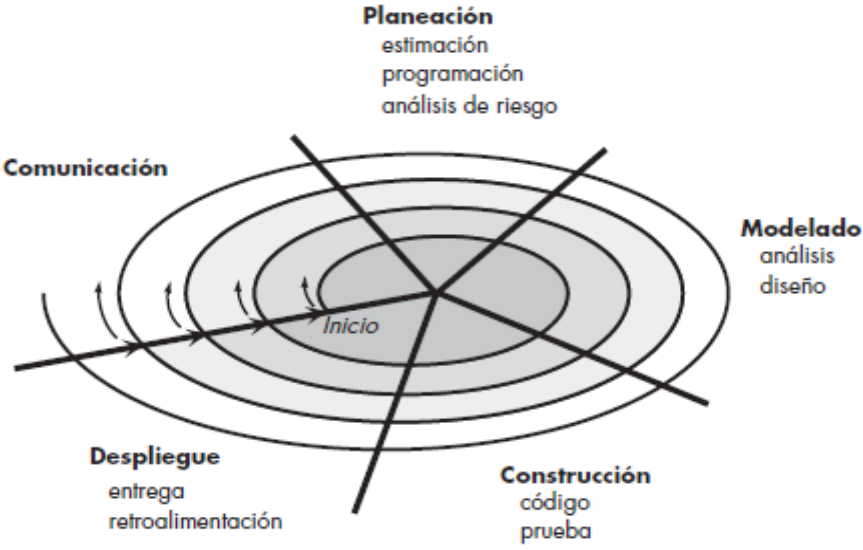
La noción de una prueba de software se refiere al conjunto de procesos con los que se pretende probar en un sistema o aplicación en diferentes momentos para comprobar su correcto funcionamiento (Turrado, 2020). En la norma ISO/IEC/IEEE 29119 se establece que una prueba de software es el conjunto de actividades realizadas para facilitar el descubrimiento y/o evaluación de las propiedades de uno o más elementos de prueba (ISO-29119, 2013), la evaluación del software es parte de las etapas del desarrollo de software, por esa razón se analizarán algunos conceptos y métodos claves referentes al desarrollo de un producto de software. La evaluación del software es parte de las etapas del desarrollo de software, por esa razón se analizarán algunos conceptos y métodos claves referentes al desarrollo de un producto de software.

Se entiende como software al conjunto de programas que, una vez ejecutados, proporcionan las características y funciones que permiten a un sistema desarrollar determinadas tareas (Sommerville, 2005). En este sentido, es necesario que estos sistemas cumplan con ciertos requerimientos que validen el correcto funcionamiento y desempeño del software (Pressman, 2010). Existen diversos modelos para el desarrollo de software entre las más utilizadas tenemos las expuestas en la Tabla 1.1.

La etapa de Pruebas, dentro del proceso de desarrollo de software, se refiere al proceso que intenta proporcionar confianza en el software, cuyo objetivo es convencer a los desarrolladores del sistema y a los clientes que el software es lo suficientemente bueno para su uso operacional. Esta etapa, previa a la etapa de producción, es crucial para garantizar el correcto funcionamiento del software.

Tabla 1.1: Modelos de desarrollo de Software

Modelos de desarrollo de Software		
Modelo	Definición	Etapas
Cascada	Es el enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior	<ol style="list-style-type: none"> 1. Comunicación. 2. Planeación 3. Modelado 4. Construcción 5. Despliegue
<p>Modelo de la cascada</p> <pre> graph LR A[Comunicación inicio del proyecto recabar los requerimientos] --> B[Planeación estimación programación seguimiento] B --> C[Modelado análisis diseño] C --> D[Construcción código pruebas] D --> E[Despliegue entrega asistencia retroalimentación] </pre> <p style="text-align: right;">Recuperado de: (Pressman, 2010)</p>		
Incremental	Es un proceso de desarrollo de software creado en respuesta a las debilidades del modelo tradicional de cascada. Básicamente este modelo de desarrollo, que no es más que un conjunto de tareas agrupadas en pequeñas etapas repetitivas (iteraciones)	<ol style="list-style-type: none"> 1. Comunicación. 2. Planeación 3. Modelado 4. Construcción 5. Despliegue
<p style="text-align: right;">Recuperado de: (Pressman, 2010)</p>		

Modelos de desarrollo de Software (Continuación)		
Evolutivo	Su principal característica es que es flexible y expandible	<ol style="list-style-type: none"> 1. Comunicación 2. Planeación 3. Modelado 4. Construcción del prototipo 5. Despliegue
		
Recuperado de: (Pressman, 2010)		
Espiral	Es una combinación entre el modelo waterfall y un modelo por iteraciones	<ol style="list-style-type: none"> 1. Comunicación 2. Planeación 3. Modelado 4. Construcción del prototipo 5. Despliegue
		
Recuperado de: (Pressman, 2010)		

Modelos de desarrollo de Software (Continuación)		
Rup	Constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.	1.Fase de Inicio 2. Fase de Elaboración 3.Fase de Desarrollo o Construcción 4.Fase de Transición
<p style="text-align: right;">Recuperado de: (Pressman, 2010)</p>		

1.5.2. Estándares utilizados para la evaluación de software

Un estándar es una serie de recomendaciones a seguir para la entrega de un producto. (Nestor, 2017). Entre los estándares más relevantes encontrados en la revisión sistemática de literatura para la evaluación de software tenemos los siguientes:

- **Norma ISO/IEC 9126.** Establecida por la Organización Internacional de Normalización, ha sido utilizada como referencia de evaluación de calidad de software. La primera aparición de la norma ISO/IEC 9126 data en el año de 1991 y para el año del 2001 se publicó una versión actual de esta (INEN-ISO/IEC 9126-1, 2001). De esta norma se destacan varias métricas y medidas que deben ser puntuadas y juzgadas en el proceso de evaluación de software, haciendo uso de los modelos propuestos de calidad interna y externa donde los diferentes atributos del software son caracterizados en seis grupos de características básicas: funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad. Estos atributos permiten definir un marco de evaluación general para la mayoría de los sistemas. Por esta razón la familia de normas ISO/IEC 9126 es aplicable para la definición de requisitos de calidad del software, evaluación de las especificaciones del software durante el desarrollo, descripción de las

características y atributos del software y evaluación del software desarrollado antes de ser entregado.

- **Norma ISO/IEC 25000.** Más conocida como SQuaRE (System and Software Quality Requirements and Evaluation), resulta de la evolución de varias normas anteriores. Es una familia de normas cuyo objetivo es la creación de un framework común para evaluar la calidad del producto software. Su fecha de publicación fue en el 2005 (ISO/IEC, 2005). Esta norma se construyó principalmente de la base de las normas ISO/IEC 9126 e ISO/IEC 14598, las cuales describían las particularidades de un modelo de calidad del producto software y abordaban el proceso de evaluación de productos software respectivamente. De esta manera se busca organizar, enriquecer y unificar los procesos principales en la evaluación de software: especificación de requisitos de calidad del software y evaluación de la calidad del software, soportada por el proceso de medición de calidad del software.
- **Norma ISO/IEC 25010:2011.** Presenta un modelo de calidad cuyo objetivo es medir y evaluar la calidad del software, mediante la implementación de un conjunto de acciones. El uso de modelos y marcos de evaluación de software, se enfocan en un principio de mejores prácticas para el aseguramiento de la aplicabilidad de un multimodelo de calidad de software (ISO/IEC 25010, 2005). Su fecha de publicación fue en el año 2011.

La Figura 1.1 nos muestra un resumen de los estándares más relevantes respecto a evaluación de software.

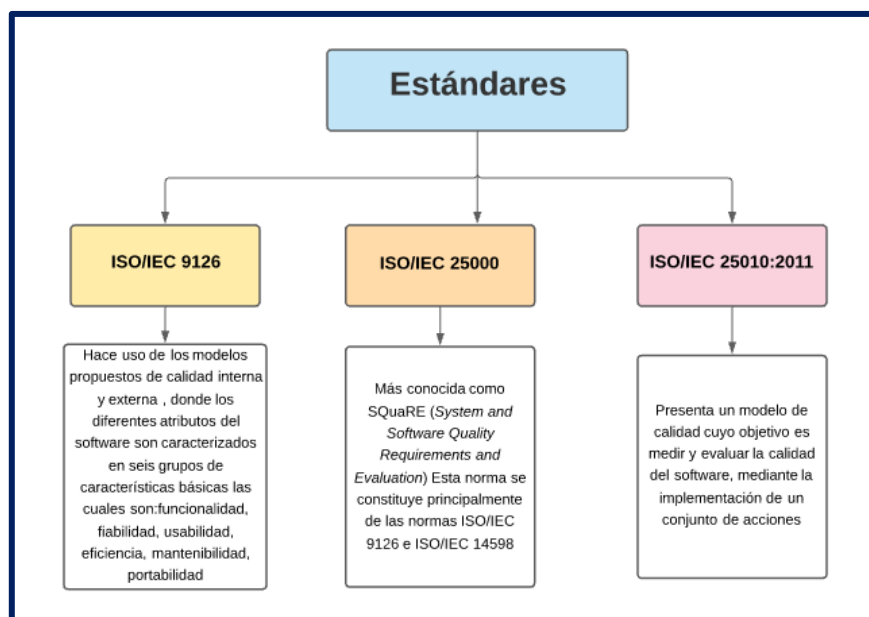


Figura 1.1: Estándares más utilizados para evaluación de software (AENOR, 2005)

1.5.3 Características de software

Podemos destacar las siguientes características que definen a un producto de software estas características se encuentran presente tanto en la revisión sistemática de literatura como en el análisis documental:

- **Funcionalidad.** Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:
 - Completitud funcional. Grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados.
 - Corrección funcional. Capacidad del producto o sistema para proveer resultados correctos con el nivel de precisión requerido.
 - Pertinencia funcional. Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados. (ISO/IEC, 2005).
- **Eficiencia.** Esta característica representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones. Se subdivide a su vez en las siguientes subcaracterísticas:
 - Comportamiento temporal. Los tiempos de respuesta, tiempos de procesamiento y los ratios de throughput de un sistema cuando lleva a cabo sus funciones bajo condiciones determinadas en relación con un banco de pruebas (benchmark) establecido.
 - Utilización de recursos. Las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función bajo condiciones determinadas.
 - Capacidad. Grado en que los límites máximos de un parámetro de un producto o sistema software cumplen con los requisitos (AENOR, 2005).
- **Compatibilidad.** Capacidad de dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno de hardware o software. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:
 - Coexistencia. Capacidad del producto para coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes sin detrimento.
 - Interoperabilidad. Capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada (ISO/IEC 25010, 2005)

- **Usabilidad.** Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:
 - Capacidad para reconocer su adecuación. Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
 - Capacidad de aprendizaje. Capacidad del producto que permite al usuario aprender su aplicación.
 - Capacidad para ser usado. Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
 - Protección contra errores de usuario. Capacidad del sistema para proteger a los usuarios de hacer errores.
 - Estética de la interfaz de usuario. Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.
 - Accesibilidad. Capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades (ISO/IEC, 2005).
- **Confiabilidad.** Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:
 - Madurez. Capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.
 - Disponibilidad. Capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.
 - Tolerancia a fallos. Capacidad del sistema o componente para operar según lo previsto en presencia de fallos hardware o software.
 - Capacidad de recuperación. Capacidad del producto software para recuperar los datos directamente afectados y reestablecer el estado deseado del sistema en caso de interrupción o fallo (AENOR, 2005).
- **Seguridad.** Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:
 - Confidencialidad. Capacidad de protección contra el acceso de datos e información no autorizados, ya sea accidental o deliberadamente.
 - Integridad. Capacidad del sistema o componente para prevenir accesos o modificaciones no autorizados a datos o programas de ordenador.

- No repudio. Capacidad de demostrar las acciones o eventos que han tenido lugar, de manera que dichas acciones o eventos no puedan ser repudiados posteriormente.
- Responsabilidad. Capacidad de rastrear de forma inequívoca las acciones de una entidad.
- Autenticidad. Capacidad de demostrar la identidad de un sujeto o un recurso (ISO/IEC, 2005).
- **Mantenibilidad.** Esta característica representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:
 - Modularidad. Capacidad de un sistema o programa de ordenador (compuesto de componentes discretos) que permite que un cambio en un componente tenga un impacto mínimo en los demás.
 - Reusabilidad. Capacidad de un activo que permite que sea utilizado en más de un sistema software o en la construcción de otros activos.
 - Analizabilidad. Facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias o causas de fallos en el software, o identificar las partes a modificar.
 - Capacidad para ser modificado. Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.
 - Capacidad para ser probado. Facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios (ISO/IEC 25010, 2005).
- **Portabilidad.** Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:
 - Adaptabilidad. Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso.
 - Capacidad para ser instalado. Facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno.

- Capacidad para ser reemplazado. Capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno (ISO/IEC 25010, 2005).

Todas las características listadas han sido definidas por ISO según cada familia de normas a la que pertenecen.

1.5.4. Frameworks

Los principales Frameworks encontrados en los documentos científicos, utilizados para evaluar un producto de software son:

- **CMMI.** Integración de sistemas modelos de madurez de capacidades o Capability Maturity Model Integration (CMMI) es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software (ISACA, 2010).
- **ISO9126-FRAM.** Este marco se basa específicamente en un modelo de calidad externa, para abordar los impactos de las limitaciones de los entornos móviles en la calidad del software de las aplicaciones móviles. Es así como las limitaciones pueden ser catalogadas en dos grupos: las limitaciones de los equipos en si (almacenamiento, interfaz de usuario, autonomía, etc.) y, en segundo lugar, las limitaciones de las redes inalámbricas como desconexión frecuente y anchos de banda diferentes (Moumane, 2015). El análisis con este framework se realiza mediante un proceso que consta de tres pasos: análisis de métricas externas, ejecución de la lista de verificación y determinación de los grados de influencia.
- **Diffserv-FRAM.** Se desarrolló a partir de un proceso de análisis con el fin de aprovechar los beneficios de ISO 9126 y DiffServ (método que intenta garantizar la calidad de servicio en redes de gran tamaño) para ayudar a los evaluadores y diseñadores de calidad a lograr un alto nivel de calidad en entornos móviles, este proceso también se compone de tres pasos: análisis de métricas externas, ejecución de la lista de verificación y determinación de los grados de influencia.
- **Web-CAT.** Es un software gratuito de código abierto de calificación automatizado, altamente personalizable y extensible. Es compatible con prácticamente cualquier modelo de calificación, evaluación y generación de comentarios de programas. Web-CAT se implementa como una aplicación web con una arquitectura de estilo plug-in de modo que también puede servir como una plataforma para proporcionar servicios adicionales de apoyo. Fue diseñado originalmente como un sistema de calificación automatizado de propósito general, pero sus diseñadores decidieron que querían

apoyar las actividades de prueba de software de los estudiantes (Web-Cat-Comunity, 2019).

- **SAMATE.** El Software Assurance Metrics And Tool Evaluation (SAMATE) es un proyecto de NIST enfocado en garantizar el software mediante el desarrollo de métodos para permitir evaluaciones de herramientas de software; midiendo la efectividad de herramientas y técnicas e identificando brechas en herramientas y métodos. El alcance del proyecto SAMATE es amplio: desde sistemas operativos a firewalls, de SCADA a aplicaciones web, de analizadores de seguridad de código fuente a métodos de corrección por construcción (Black, 2007).

El proyecto SAMATE consta de dos partes:

Desarrollo de métricas para la efectividad de herramientas de aseguramiento de seguridad de software (Software Security Assurance, SSA).

Evaluación de métodos y herramientas actuales de SSA para identificar deficiencias que pueden conducir a fallas y vulnerabilidades de los productos de software.

- **UMARA.** La instrumentación interactiva brinda nuevas oportunidades para integrar la evaluación de la usabilidad en el proceso de desarrollo de software, independientemente del tamaño del proyecto. La instrumentación de usabilidad interactiva (IUI) puede acelerar drásticamente el proceso de instrumentación y simplificar los métodos de usabilidad basados en registros (Bateman, 2009).

UMARA no solo habilita IUI para instrumentar software, sino que también admite los tres pasos en una evaluación de usabilidad basada en registros: instrumentación de la aplicación, recopilación de datos de uso y análisis de datos.

- **Multidimensional Weighted Attribute Framework.** También conocida como MWAF es un marco de uso general que se puede adaptar para evaluar diferentes productos de software, por ejemplo: lenguajes de programación, sistemas operativos, metodologías de ingeniería de software, conjuntos de herramientas de desarrollo de software, protocolos de comunicaciones de software, etc.

La idea principal de MWAF es definir los criterios (o dimensiones) más comunes e importantes del sistema que se está evaluando, identificando los atributos que describen cada una de estas dimensiones, y luego evaluando cada dimensión a través de sus atributos contra todos los sistemas potenciales que se seleccionan para evaluar MWAF consta de los siguientes tres componentes principales:

Dimensiones: el marco comprende una serie de dimensiones, cada una de las cuales representa uno de los principales criterios de evaluación de software.

Atributos: son las diferentes características pertenecientes a cada criterio (es decir, dimensión) para describirlo utilizando un conjunto de preguntas definidas.

Parámetros: los valores numéricos que se dan para medir los atributos.

Al aplicar MAAF, a los usuarios expertos se le pedirá que asignen dos calificaciones a cada uno de los atributos evaluados: una ponderación para identificar la importancia del atributo y una tasa para medir su fuerza o eficacia. El peso es un parámetro subjetivo, ya que depende completamente de la opinión del evaluador. Por otro lado, la tasa es un parámetro objetivo porque se mide de acuerdo con el grado de disponibilidad o efectividad de la propiedad examinada representada por el atributo evaluado. En MAAF, los valores dados a los dos parámetros son numéricos y van de 0 a 10. Un valor de 0 implica la ausencia total del atributo medido, mientras que un valor de 10 refleja su máxima disponibilidad y fuerza. Para tener una visión amplia de las conclusiones y hallazgos finales, cada sistema será evaluado por varios usuarios expertos (Far, 2008).

- **DESMET.** Es un marco genérico y formalizado para evaluar métodos y herramientas de Ingeniería de Software (Strooper, 2006). Proporciona un conjunto de nueve métodos de evaluación de software basados en técnicas de medición cualitativas y cuantitativas. Las aplicaciones del marco pueden utilizar experimentos formales, casos de estudio, encuestas de usuarios o una combinación de estos para evaluar formalmente tanto los métodos como las herramientas. Las evaluaciones se llevan a cabo en relación con un entorno particular de procesos, herramientas y procedimientos de desarrollo de software.
- **Usability Metric for User Experience (UMUX).** Este marco fue diseñado para fungir como una escala de usabilidad breve pero efectiva. El contenido de la escala fue diseñado para cubrir de manera más adecuada y uniforme los tres aspectos de usabilidad mencionados en la norma ISO-9241-11 (1998) (Bosley, 2013).

La Figura 1.2 nos muestra cómo se han ido desarrollando los frameworks a través del tiempo.

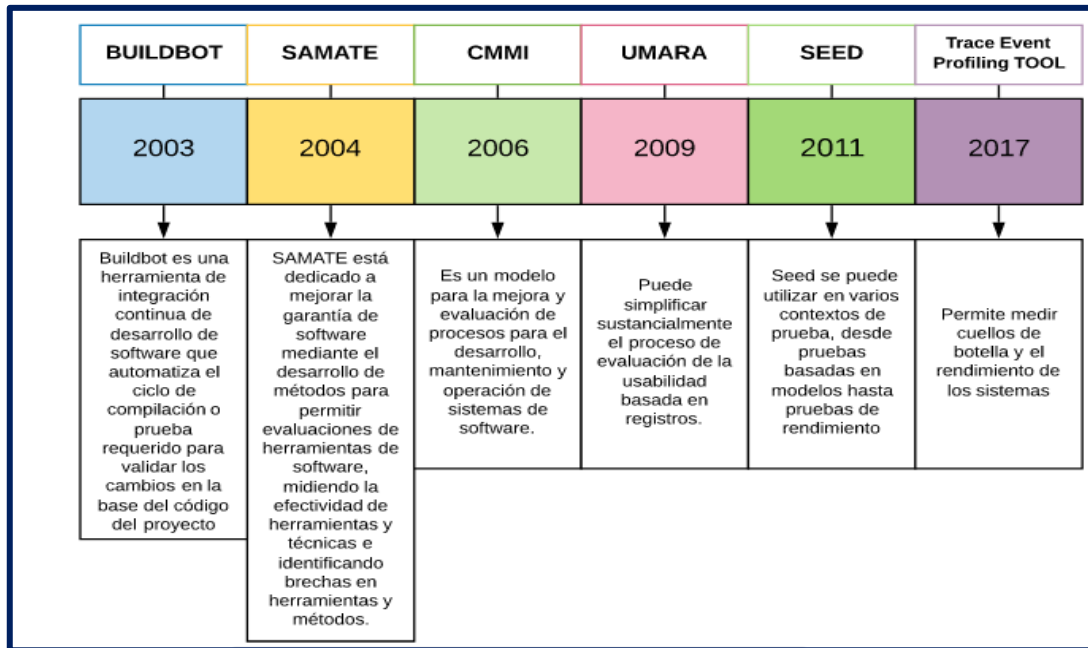


Figura 1.2 : Línea de tiempo de frameworks que permiten realizar la evaluación de software

1.5.5. Pruebas realizadas al evaluar el software

Entre las pruebas realizadas tanto en los proyectos de titulación como en los documentos científicos tenemos las siguientes:

- **Pruebas funcionales.** Son aquellas que están basadas en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software, es decir, en los requisitos funcionales.

Dentro de las pruebas funcionales tenemos:

- Pruebas unitarias que aseguran que cada célula del código desarrollado en un componente brinde los resultados adecuados (Pressman, 2010).
- Pruebas de componentes que se ejecutan de forma independiente para comprobar que el resultado sea el requerido. Su objetivo es verificar las funcionalidades y/o usabilidades de los componentes, aunque no solo se limite a eso (Pressman, 2010).
- Pruebas de integración que es uno de los tipos de prueba funcional más común y se realiza de forma automatizada. Se realizan para probar componentes individuales con el objetivo de verificar cómo los módulos, que trabajan de forma individual, funcionan cuando estén integrados.
- Pruebas de aceptación pertenecen a las últimas etapas previas a la liberación en firme de versiones nuevas a fin de determinar si cumplen con las necesidades y/o requerimientos de las empresas y sus usuarios (IBM, Corp, 2006)

- Prueba alfa: se lleva a cabo, por un cliente, en el lugar de desarrollo. Se usa el software de forma natural con el desarrollador como observador del usuario y registrando los errores y problemas de uso. Las pruebas alfa se llevan a cabo en un entorno controlado.
- Prueba beta: se llevan a cabo por los usuarios finales del software en los lugares de trabajo de los clientes. A diferencia de la prueba alfa, el desarrollador no está presente normalmente. Así, la prueba beta es una aplicación en vivo del software en un entorno que no puede ser controlado por el desarrollador. El cliente registra todos los problemas que encuentra durante la prueba beta e informa a intervalos regulares al desarrollador (IBM, Corp, 2006).
- **Pruebas no funcionales.** Una prueba no funcional es una prueba cuyo objetivo es la verificación de un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema (requisitos no funcionales) como por ejemplo la disponibilidad, accesibilidad, usabilidad, mantenibilidad, seguridad, rendimiento. Podemos clasificar las pruebas no funcionales según el tipo de requisito no funcional que abarcan:
 - La prueba de usabilidad por parte del usuario es una técnica usada en el diseño de interacciones centrado en el usuario para evaluar un producto mediante pruebas con los usuarios mismos. Esto puede ser visto como una práctica de usabilidad irremplazable, dado que entrega información directa de cómo los usuarios reales utilizan el sistema (Morales, 2000).
 - Las pruebas de rendimiento son las pruebas que se realizan con el objetivo de determinar el rendimiento del sistema bajo una carga de trabajo definida. Entre ellas tenemos pruebas de carga y stress (IBM, 2002):

Las pruebas de carga nos permiten evaluar la cantidad de peticiones que puede soportar un sistema.

Las pruebas de stress consisten en probar los límites que un sistema puede soportar.

- Las pruebas de seguridad se podrían definir como el conjunto de actividades que se llevan a cabo para encontrar fallas y vulnerabilidades en aplicaciones web, buscando disminuir el impacto de ataques a ellas y pérdida de información importante.

1.5.6. Proceso para la evaluación de software

Al momento de realizar la revisión sistemática de literatura encontramos que el uso de normas como la ISO 14598 nos permite tener una idea clara de los pasos a seguir al momento de realizar la evaluación de un software. La Figura 1.3 nos muestra el proceso para la evaluación de software y cada una de sus etapas según la norma ISO 14598.

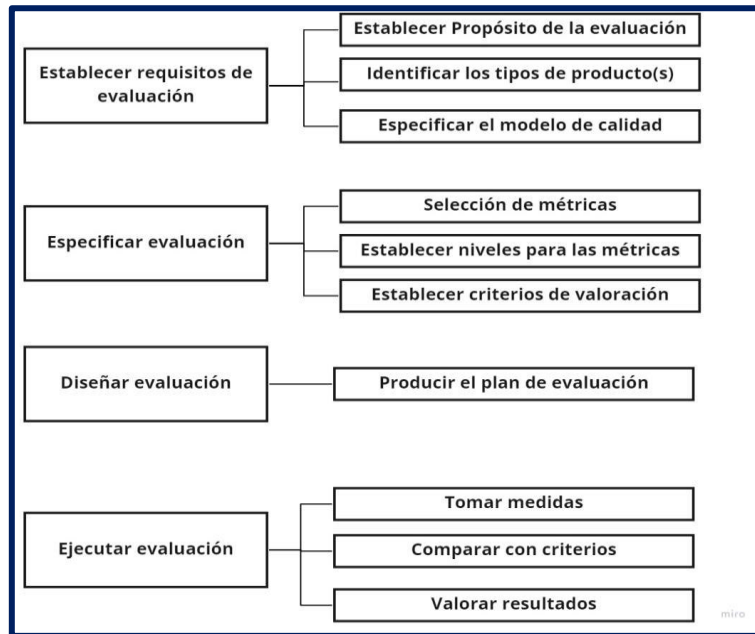


Figura 1.3: Proceso para la evaluación de software según ISO 14598

2. METODOLOGÍA

El desarrollo del presente trabajo se realizó en dos etapas. La primera de ellas consistió en una revisión sistemática de la literatura (SLR por sus siglas en inglés) de un conjunto de artículos científicos. La segunda etapa constó de un análisis documental de los proyectos de titulación realizados en la FIS en los últimos 10 años. Ambas etapas permitirán obtener los lineamientos buscados para la evaluación de software (Figura 2.1).

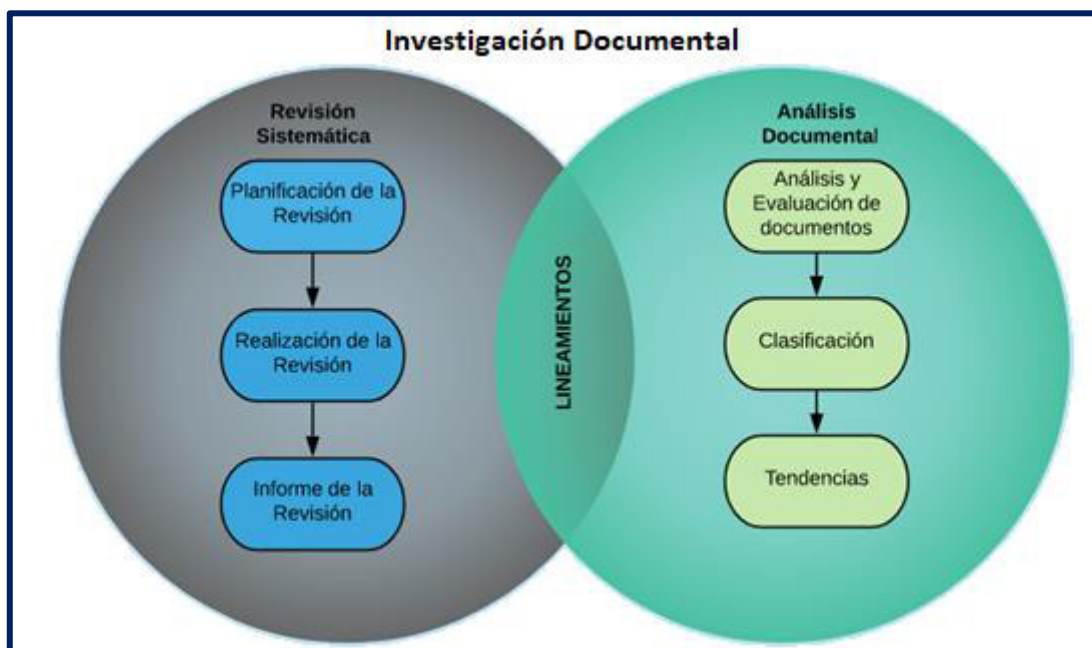


Figura 2.1: Descripción del proceso realizado para realizar una Investigación documental

2.1. Revisión Sistemática de Literatura

La necesidad de una SLR ¹ surge del requisito de los investigadores de resumir toda la información existente sobre algún fenómeno de manera exhaustiva e imparcial. Esto puede servir para sacar una conclusión más general sobre algún fenómeno de lo que es posible a partir de estudios individuales, o como un prelude para futuras actividades de investigación. Una SLR consta de tres fases principales: planificación de la revisión, realización de la revisión e informe de la revisión.

¹ SLR es un medio para identificar, evaluar e interpretar toda la investigación disponible relevante para responder una pregunta de investigación en particular, un área temática o un fenómeno de interés (Kitchenham, 2004)

Planificación de la revisión. En la etapa de planificación de la revisión, se plantea una pregunta de investigación y se define un área de aplicación. Para plantear la pregunta de investigación tomamos en cuenta tres puntos de vista:

Población: Nos referimos al grupo de personas que van a ser afectadas por la intervención (Kitchenham, 2004), que en nuestro caso son estudiantes de la FIS que realizan el desarrollo de un producto de software como proyecto de titulación.

Intervención: La intervención es la metodología / herramienta / tecnología / procedimiento de software que aborda un problema específico (Kitchenham, 2004). En nuestro caso son aquellos métodos de evaluación de software, frameworks, estándares, procesos, características de software que se utilizan para evaluar software.

Resultados: Los resultados deben relacionarse con factores de importancia para la población escogida anteriormente (Kitchenham, 2004). En nuestro caso son aquellos lineamientos que nos permitan realizar la evaluación del producto de software.

Después de haber tomado en cuenta los tres puntos de vista planteados anteriormente, nuestra pregunta de investigación es la siguiente:

¿Qué métodos de evaluación de software, frameworks, estándares, procesos, características de software existen y cuáles se han utilizado para realizar la evaluación de un producto de software?

Mientras que el área al que está enfocada es al desarrollo de productos de software como proyectos de titulación

Realización de la revisión. En esta fase, se definen las fuentes donde se va a realizar la SLR y las respectivas cadenas de búsqueda. Las fuentes definidas para la aplicación de la SLR fueron las bibliotecas digitales tanto de IEEE como de ACM. Dentro de cada biblioteca encontramos artículos, conferencias, artículos de investigación y documentos parte de simposios

Los principales criterios de búsqueda retenidos son aquellos términos que están en relación directa con la evaluación de software, es decir, se buscaron documentos científicos que traten sobre procesos para la evaluación de producto de software, frameworks, características, métodos, estándares para la evaluación de productos de software (Tabla 2.1). En la Tabla 2.2 se muestra la cadena de búsqueda utilizada tanto en el resumen como el título de cada artículo científico.

Tabla 2.1: Criterios de búsqueda en SLR

Criterios de búsqueda			
Palabras Clave	Fuente de Investigación	Tipo	Relevancia
Evaluación de Software	ACM	Artículo	Mayor número de descargas y citas
Metodologías para	IEEE	Conferencia	
Evaluación de software		Artículo de Investigación	
Frameworks para		Simposio	
evaluación de software			
Características de			
evaluación de software			
Modelos de evaluación de software			

Tabla 2.2: Cadena de búsquedas utilizadas en SLR

Cadenas de búsqueda	
SS1	("software evaluation" OR "evaluating software" OR "software assessment") AND ("methodology" OR "method" OR "model" OR "framework" OR "standard" OR "process")

La cadena de búsqueda representa la combinación de las diferentes palabras claves cuyo objetivo es abarcar todos los posibles resultados (Figura 2.2).

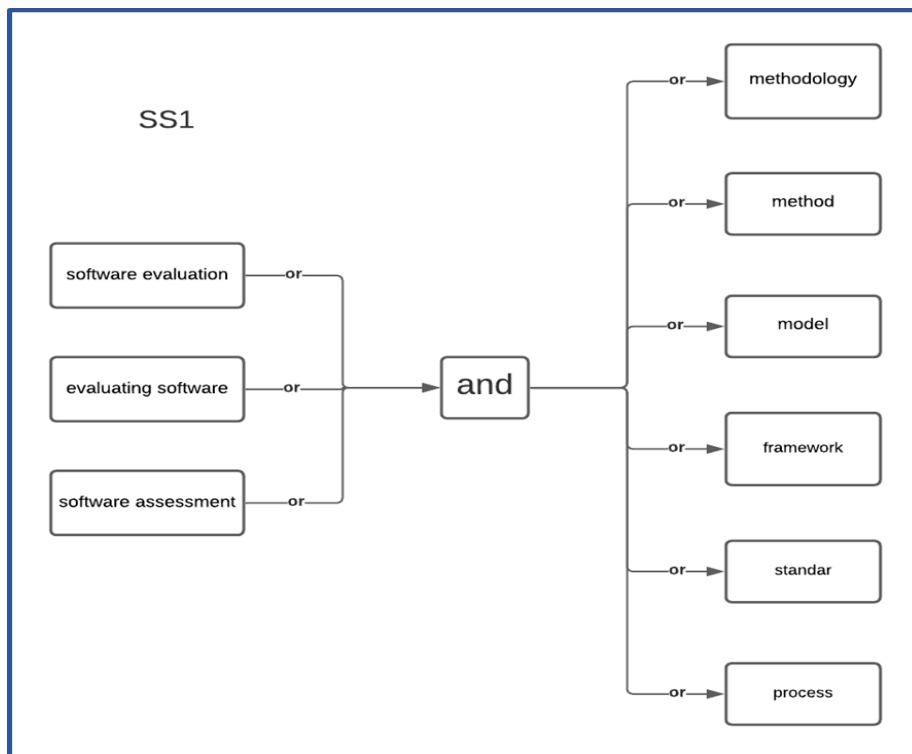


Figura 2.2: Combinaciones de la SS1

La Tabla 2.3 nos muestra el número de documentos encontrados en la cadena de búsqueda las cuales fueron buscadas tanto en el título como en el resumen de cada documento.

Tabla 2.3 : Número de documentos encontrados por cadena de búsqueda

	Cadenas de Búsqueda			
	IEEE		ACM	
	Título	Resumen	Título	Resumen
SS1:	38	159	12	57
Total:	197		69	

A continuación, se revisó detenidamente cada uno de los documentos científicos encontrados para dar paso a la selección de estudios. El proceso de selección de estudios consiste en identificar los documentos científicos relevantes para el objetivo de la SLR (Kitchenham, 2004) para ello definimos los criterios de inclusión y exclusión.

Criterio de inclusión. Han sido seleccionados aquellos documentos científicos que aportan un enfoque relacionado al proceso que se debe seguir para evaluar un software, características que se deben evaluar, metodologías, modelos, frameworks, estándares de evaluación de software. Los documentos científicos se encuentran escritos en inglés y han sido publicados en los últimos 10 años en alguno de los repositorios mencionados en la Tabla 2.1.

Criterio de exclusión. Los documentos que se encuentran duplicados ya que han sido publicados en los dos repositorios tomados en cuenta para esta investigación, así como los documentos que no profundizan en el proceso de evaluación de software o que solo mencionan alguna de las palabras claves buscadas.

La Figura 2.3 nos explica el proceso de selección de documentos. Después de haber realizado la búsqueda en base a la cadena de búsqueda, se obtuvieron un total de 266 documentos. Luego, se procedió a eliminar aquellos documentos que se encuentran duplicados reduciendo el número a un total de 221 documentos. Al aplicar criterios de exclusión e inclusión se redujo a un total de 134 documentos científicos. De estos 134 documentos, 54 corresponden a artículos, 52 a comunicaciones en conferencias, 20 a artículos de investigación y 8 a simposios (Tabla 2.4).

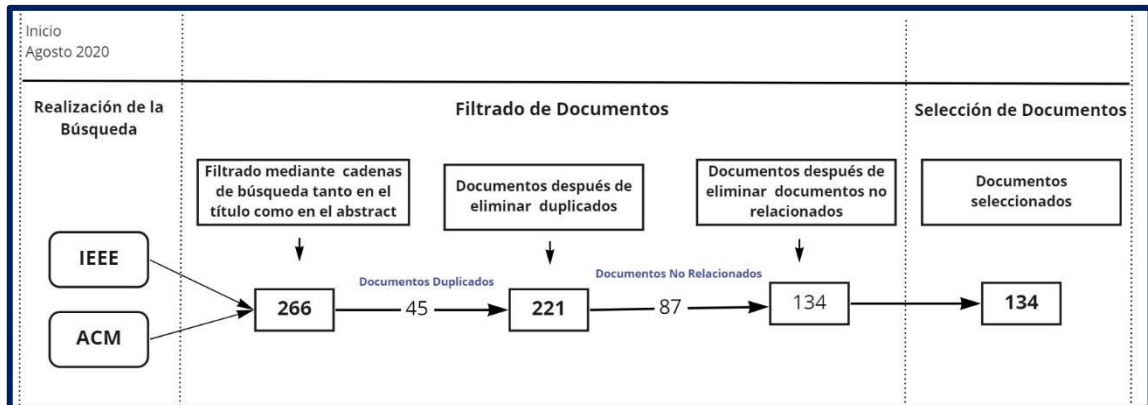


Figura 2.3: Selección de Documentos en SLR

Existen diferencias mínimas entre los términos conferencias y simposios, de igual manera entre los términos artículos y los artículos en investigación. En el caso de conferencia se refiere a una reunión formal en la que los participantes intercambian sus puntos de vista sobre varios temas mientras que en un simposio son expertos en sus campos que presentan o expresan sus opiniones o puntos de vista sobre un tema de discusión elegido. (EIDH, 2021). Por otro lado, el artículo de investigación se caracteriza por reportar los resultados novedosos que son producto de un proceso de investigación realizada, mientras que un artículo en general puede ser la recopilación de varias investigaciones anteriores y su objetivo es enfatizar en un tema en particular.

Tabla 2.4: Tipos de documentos seleccionados

Tipo de documentos seleccionados	
Artículos	54
Conferencias	52
Artículos de investigación	20
Simposios	8
Total:	134

Informe de la revisión. Una vez obtenidos los documentos científicos, se realizaron varias sesiones de discusión que, en conjunto con la lectura a profundidad de los documentos científicos, permitieron abstraer un set de palabras claves, significado y contexto de las mismas, permitieron definir tres categorías principales, dentro de estas categorías se definieron 8 diferentes códigos y a su vez, permiten clasificar los 150 verbatim² recolectados durante el análisis de los 134 documentos científicos, utilizando de la

² Un verbatim “es la reproducción exacta de una sentencia, frase, cita u otra secuencia de texto desde una fuente a otra” (Sensagent Corporation, 2014).

herramienta de Atlas.ti, que facilitó el trabajo de análisis y extracción de la información analizable. Los verbatim obtenidos, fueron clasificados en una grilla de codificación (Tabla 2.5) donde se puede observar las categorías que se definieron como las más relevantes para el estudio.

¿Con que se evaluó? Esta categoría hace referencia a todo artefacto, pudiendo ser frameworks, estándares, métricas o metodologías, que permiten realizar una evaluación de un producto de software.

¿A qué se evaluó? Se refiere a las características que fueron tomadas en cuenta según el tipo de software a evaluar.

¿Cómo se evaluó? Aborda el proceso de evaluación de software, el diseño y aplicación de pruebas y el análisis de los resultados obtenidos en las pruebas aplicadas.

Tabla 2.5: Categorías para la evaluación de software

Categoría	Subcategoría
Artefactos de Evaluación ¿Con qué se evaluó?	Metodologías de evaluación de software
	Frameworks utilizados para la evaluación de software
	Estándares Utilizados
	Métricas de evaluación de software utilizadas
Software Evaluado ¿A qué se evaluó?	Características de software evaluadas
	Tipo de software
Proceso de Evaluación ¿Cómo se evaluó?	Pruebas realizadas
	Proceso que se siguió al evaluar

Cada categoría tiene subcategorías las cuales las explicaremos a continuación:

- Metodologías de evaluación de software: Hace referencia al grupo de mecanismos o procedimientos racionales, empleados para el logro de un objetivo, o serie de objetivos (Pérez, 2021).
- Frameworks utilizados para la evaluación de software: Se refiere aquellos marcos de trabajo que permiten realizar una evaluación de software.
- Estándares Utilizados: Aquella normativa utilizada para evaluar software.
- Métricas de evaluación de software utilizadas: Aquellos indicadores que se usaron para evaluar un software.
- Características de software evaluadas: Propiedades que se evaluaron del software.
- Tipo de software: Clase de producto de software desarrollado.
- Pruebas realizadas: Tipos de pruebas utilizadas para realizar evaluación de software

- Proceso que se siguió al evaluar: Pasos que se realizaron para realizar la evaluación de software.

Los códigos con los que se va a trabajar (Tabla 2.6) son el resultante la unión de las 3 primeras letras del código de la categoría más las 3 primeras letras del código de la subcategoría (Figura 2.4).

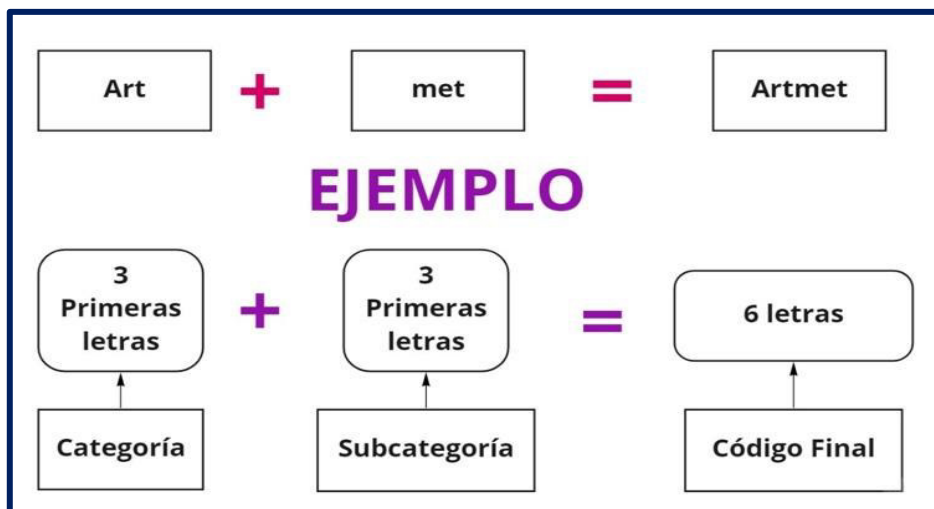


Figura 2.4: Estructura del código

Tabla 2.6: Códigos para determinar las diferentes categorías al momento de evaluar software

Código de categoría	Categoría	Código subcategoría	Sub-Categoría	Código Final
ART	Artefactos de Evaluación ¿Con qué se evaluó?	Met	Metodologías de evaluación de software	ARTmet
		Fra	Frameworks utilizados para la evaluación de software	ARTfra
		Est	Estándares Utilizados	ARTest
		Mtr	Métricas de evaluación de software utilizadas	ARTmtr
SOF	Software Evaluado ¿A qué se evaluó?	Car	Características de software evaluadas	SOFcar
		Tip	Tipo de software	SOFtip
PRO	Proceso de Evaluación ¿Cómo se evaluó?	Pru	Pruebas realizadas	PROpru
		Proc	Proceso que se siguió al evaluar	PROproc

2.2. Análisis documental

El análisis documental ³ será abordado de la misma manera que la SLR, a partir de tres fases principales: análisis y búsqueda de documentos, clasificación y determinación de tendencias.

Análisis y búsqueda de documentos. Para el análisis documental se escogieron los proyectos de titulación desarrollados en la FIS correspondientes a los últimos 10 años, entre el año 2010 y 2020. Para obtener esa información, se utilizó el repositorio digital de la Escuela Politécnica Nacional, específicamente el repositorio de proyectos de titulación de la FIS (<https://bibdigital.epn.edu.ec/>), y la consulta de los trabajos publicados se realizó en marzo del 2021.

Para realizar la búsqueda se utilizaron las palabras clave “implementación” y “desarrollo de software”, limitándose la búsqueda al periodo antes mencionado Tabla 2.7.

Tabla 2.7: Criterios de Búsqueda en el Análisis Documental

Criterios de Búsqueda				
Palabras Clave		Fuente de Investigación	Tipo	Relevancia
Implementación de software	de	Repositorio Digital FIS	Trabajos de Titulación	Año de publicación:
Desarrollo de software			Proyectos integradores	2010- 2020

De esta búsqueda se obtuvo un total de 247 tesis, de los cuales se aplicó criterios de inclusión y exclusión para determinar con que proyectos de titulación se va a trabajar.

Criterio de inclusión. Han sido seleccionados aquellos proyectos de titulación cuyo propósito sea desarrollo de un producto de software además de aquellos proyectos de titulación que hayan sido desarrollados en los últimos 10.

Criterio de exclusión. Los proyectos de titulación que se encuentren fuera del rango de fechas establecido y no estén relacionados con el desarrollo de un producto de software.

³ La investigación documental es una técnica de investigación cualitativa que se encarga de recopilar y seleccionar información a través de la lectura de documentos, libros, revistas, grabaciones, filmaciones, periódicos, bibliografías, etc. Se define como un servicio de información retrospectivo que procura obtener, seleccionar, compilar, organizar, interpretar y analizar información sobre un objeto de estudio a partir de fuentes documentales (Oates., 2006).

La Figura 2.5 muestra el proceso de búsqueda realizado para la obtención de los documentos para el análisis documental a partir de los documentos de los proyectos de titulación de la FIS.

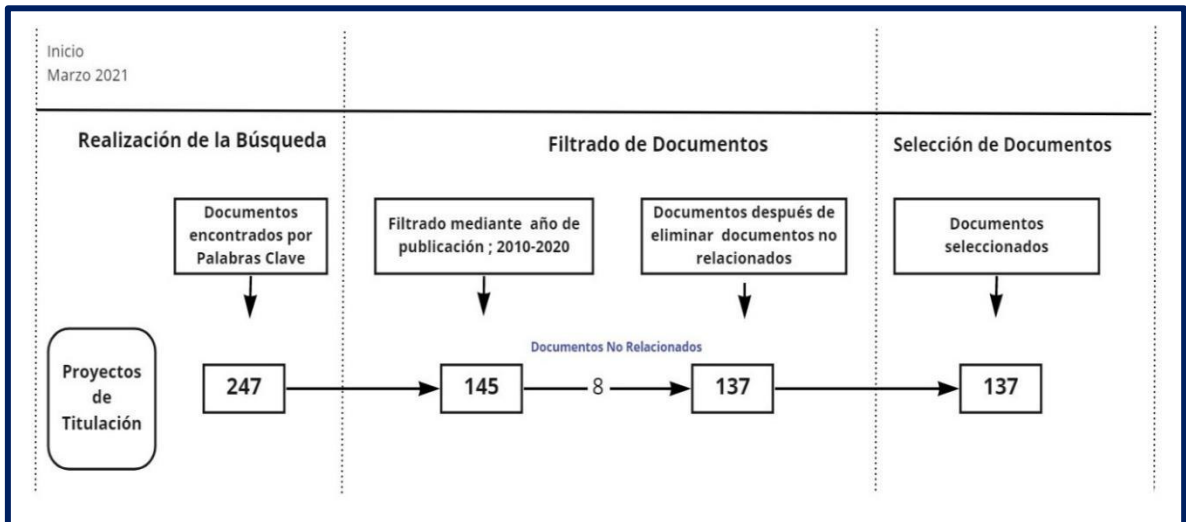


Figura 2.5: Selección de documentos para análisis documental.

Clasificación. En total se analizaron 137 proyectos de titulación para luego clasificar la información obtenida, de acuerdo con las categorías previamente establecidas (Tabla 2.6) y así identificar que tendencias existen en la FIS al momento de realizar la evaluación del producto de software. Al analizar cada proyecto de titulación se extrajo información referente a la etapa de pruebas en el momento del desarrollo del software (Figura 2.6).

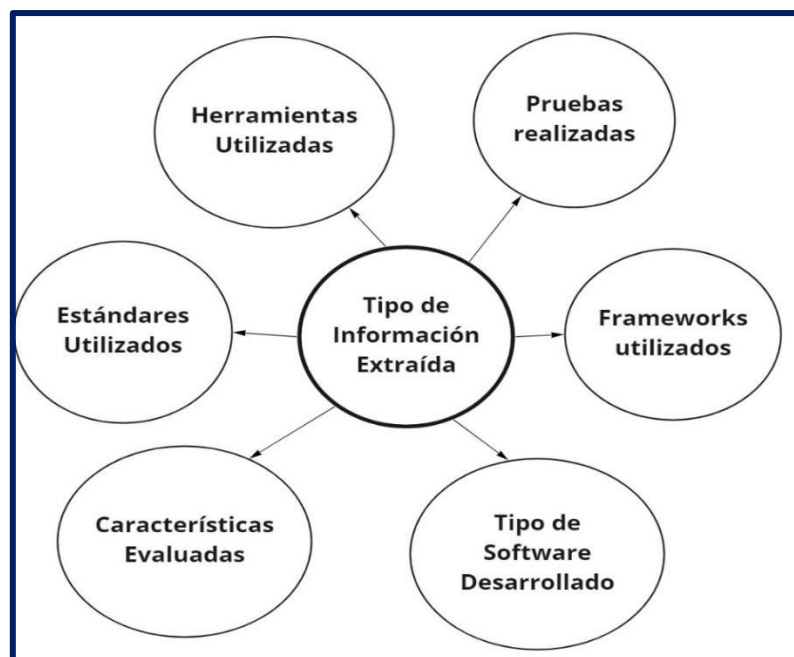


Figura 2.6: Tipo de información extraída

La información extraída se la agrupó en una Hoja de Excel para proceder a analizar cada uno de los datos adquiridos como lo muestra la Tabla 2.8.

Tabla 2.8: Formato de información extraída de los proyectos de titulación

N°	Año	Título	Tipo de Software	Tipo de Evaluación	Tipo de pruebas	Herramienta	Métricas

Ejemplo

Determinación de tendencias. Después de clasificar la información obtenida de cada proyecto de titulación, se analizó la frecuencia con la que se repiten las diferentes categorías o se realizan ciertas actividades al momento de evaluar el software con el fin de determinar cuáles son las tendencias al momento de evaluar software (Tabla 2.5).

3. RESULTADOS Y DISCUSIÓN

3.1 Resultados de la revisión sistemática de literatura

En los 137 documentos científicos se obtuvieron 150 verbatims* codificados de acuerdo con la guía de codificación de la Tabla 2.6. La Figura 3.1 presenta la frecuencia de cada código con respecto al número de artículos analizados.

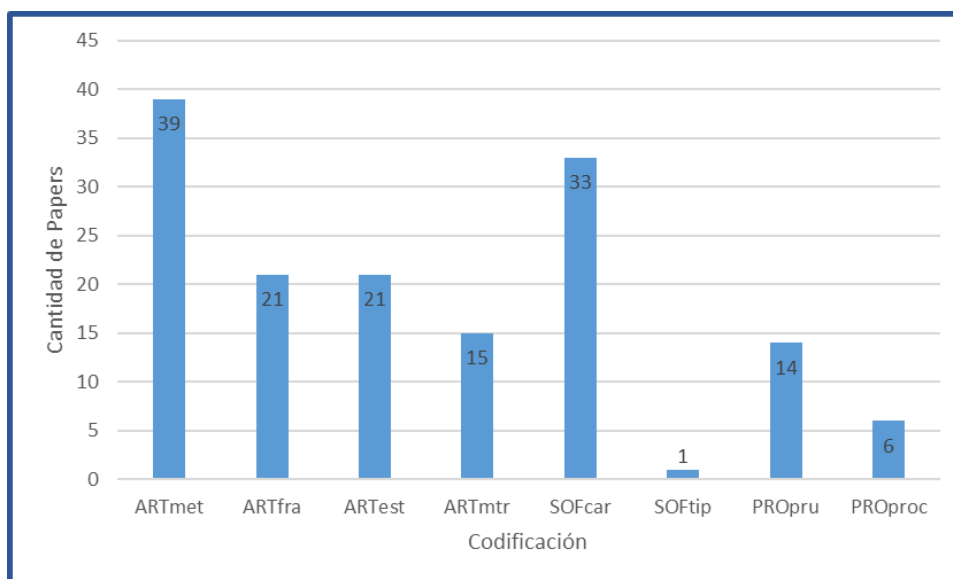


Figura 3.1: Gráfica de la relación de números de papers con respecto a los códigos establecidos

3.1.1 Estándares más utilizados en los artículos analizados

Al momento de realizar la revisión detallada del conjunto de documentos científicos, se evidenciaron varios estándares utilizados al evaluar un producto de software. Los estándares encontrados se muestran en la Figura 3.2.

Observó que el estándar ISO/IEC 9126 es el estándar más utilizado, después de este se encontró a la familia ISO/IEC 25000. En los 10 casos encontrados los autores Paulo Muniz (2015), Tianshu Wang, Deming Zhong (2008), Wang Liwen(2016), JamaiahH et al. (2006) concuerdan que la ISO 9126 nos permite evaluar adecuadamente la calidad de un producto de software debido a que evalúa 6 características básicas: funcionalidad, confiabilidad, usabilidad, mantenibilidad, eficiencia, y portabilidad. Estas características definen un marco general para el proceso de evaluación de software; son adaptables y se pueden utilizar como base para evaluar la mayoría de los sistemas (Barros, 2015).

Mientras que otros autores, como Ahmad Al-Elaimat y Abdel-Rahman Al-Ghuwairi (2015), privilegiaron el uso de ISO/IEC 25010: 2011 como modelo de calidad del software, resaltando que este modelo permite medir y evaluar la calidad del software a través de un conjunto de estandarizados de acciones.

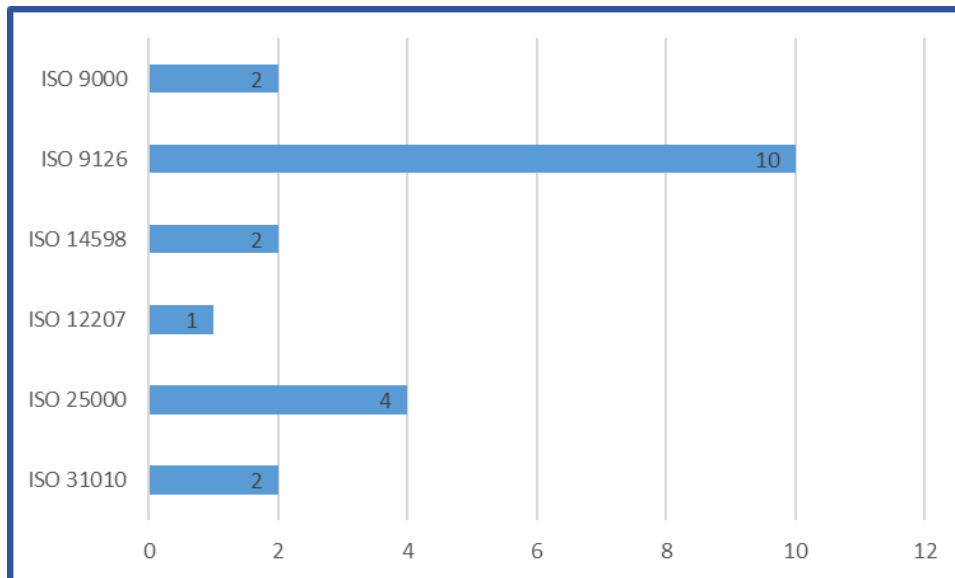


Figura 3.2: Estándares más utilizados en los artículos analizados

De la misma manera otra autora Ana Cristina Fernández Lima et al. (2018) concuerda en el uso de la ISO 25040 la cual nos permite establecer los requisitos generales para evaluar. Por otra parte, Sun-Jan Huang, Wen-Chuan Chen proponen el uso de ISO/IEC 14598 la cual incluye seis partes y define un marco del proceso de evaluación de la calidad del producto de software.

3.1.2 Frameworks más utilizados en los artículos analizados

Como resultado del análisis de los documentos científicos, se logró determinar los frameworks más utilizados para evaluar software, como se puede ver a en la Figura 3.3, se puede observar que CMMI es por mucho el más utilizado de ellos.

Autores como Ahmad Al-Elaimat, Abdel-Rahman (2015), Meena Sharma et al (2015), afirman que CMMI es un framework que permite evaluar el desarrollo de software y sus procesos garantizan la calidad del software. Adicionalmente el framework CMMI se solapa con los objetivos de evaluación de calidad de software de la norma ISO/IEC 25010 (Al-Elaimat, 2015).

Por otra parte, se encontraron frameworks que permiten evaluar características específicas de software como es el caso de Umux y Umara que permite evaluar la usabilidad. Umara

permite a los usuarios instrumentar el software sin necesidad de codificación adicional y proporciona herramientas para la especificación, la recopilación de datos y el análisis de datos. Según Scott Bateman et al.(2009), afirman que Umara puede simplificar sustancialmente el proceso de evaluación de la usabilidad basada en registro.

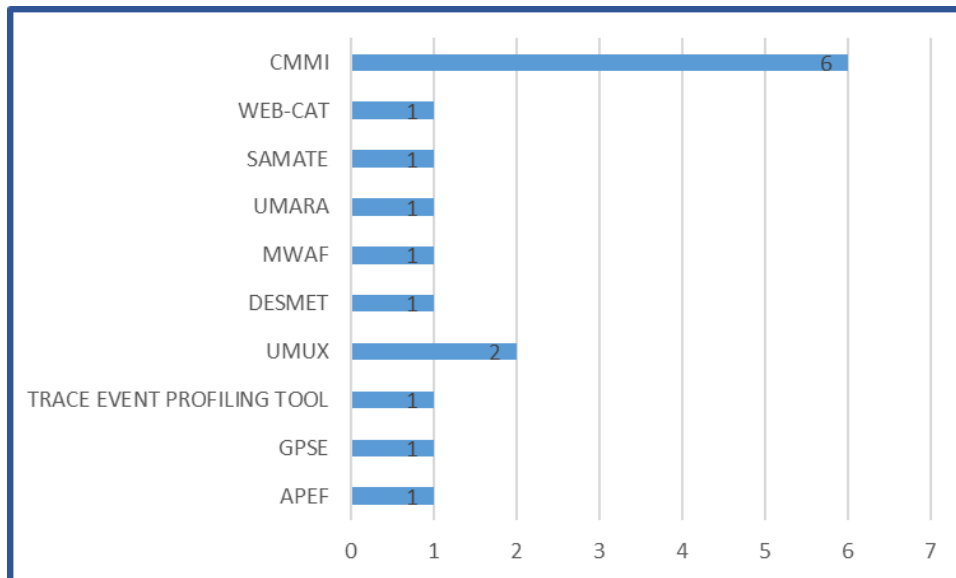


Figura 3.3: Frameworks más utilizados en los artículos analizados

3.1.3 Características más utilizadas en los artículos analizados

En el análisis realizado a los artículos recuperados, se determinó que las características más utilizadas al momento de evaluar software son las listadas en la Figura 3.4. Siendo la usabilidad la característica que más ocasiones aparece en la literatura, la encontramos 13 veces incluso por delante de la seguridad o eficiencia.

Autores como Alfredo Mendoza-González et .al (2018), afirman que se ha demostrado que la evaluación de los usuarios de los productos de software mejora las necesidades de los usuarios en términos de usabilidad, accesibilidad, ergonomía y aprendizaje (Alfredo Mendoza-González, 2018).

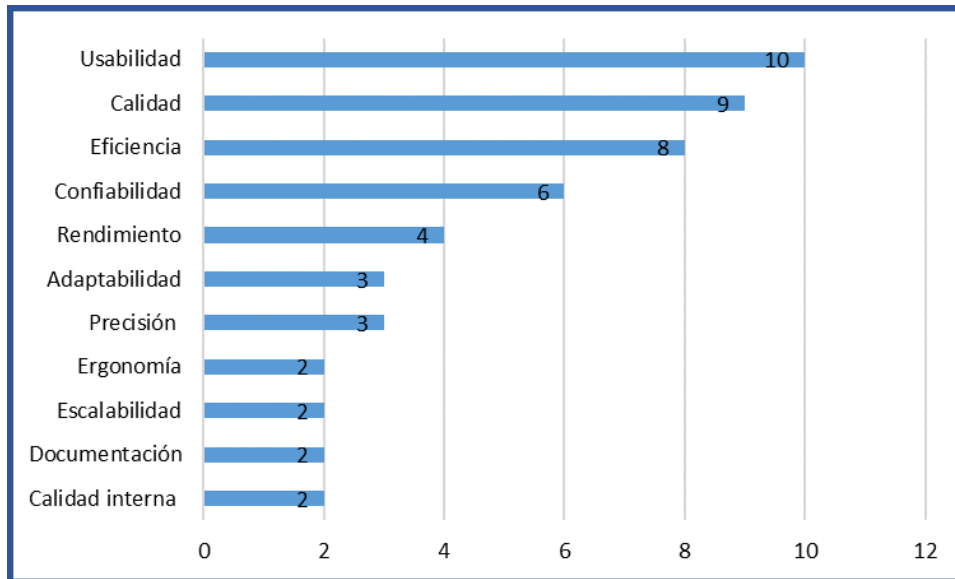


Figura 3.4: Características más utilizadas en los artículos analizados

3.1.4 Pruebas más utilizadas en los artículos analizados

En la Figura 3.5 se puede observar el tipo de pruebas que se han utilizado para evaluar software según los documentos científicos analizados.



Figura 3.5: Pruebas más utilizadas en los artículos analizados

Las pruebas más utilizadas para evaluar software encontradas en los documentos científicos son las encuestas, seguido de las pruebas de rendimiento.

3.2. Análisis documental

3.2.1 Análisis de proyectos de titulación de la Facultad de Ingeniería en Sistemas

Durante los últimos 10 años, en la FIS se han desarrollado diferentes tipos de software como trabajos de titulación. Aplicando los criterios de búsqueda, se han recuperado 137 proyectos de titulación que han desarrollado productos de software para el periodo comprendido entre 2010 y 2020. Siendo el desarrollo web y móvil los tipos de software que más se han desarrollado como proyecto de titulación (Figura 3.6).

Entre los variados productos de software desarrollados en la FIS, se han realizado diferentes tipos de evaluaciones, predominando la evaluación de la funcionalidad del producto de software. La Figura 3.7 muestra el número de veces que se ha realizado un tipo de evaluación de software específico del total de proyectos de titulación analizados.

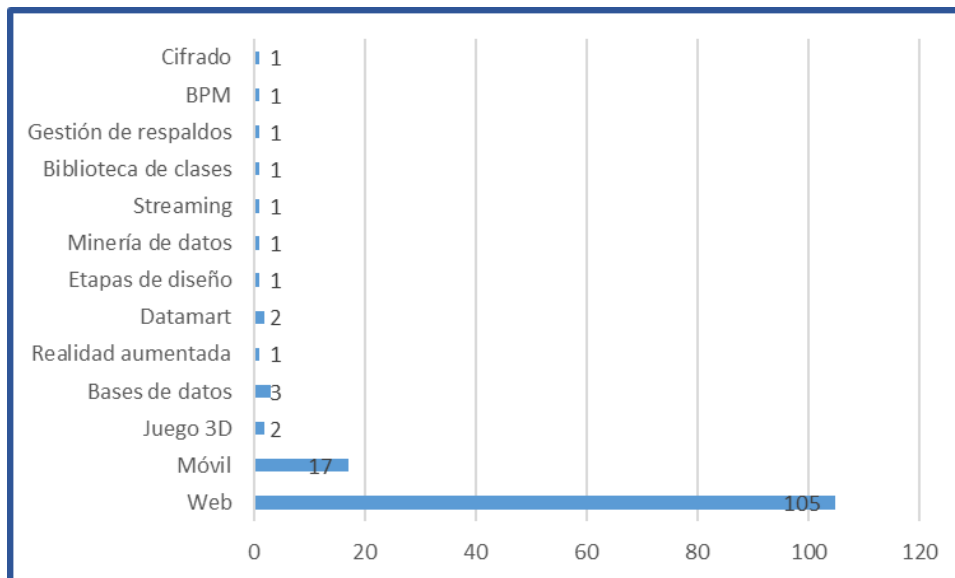


Figura 3.6: Tipo de software desarrollado en la FIS

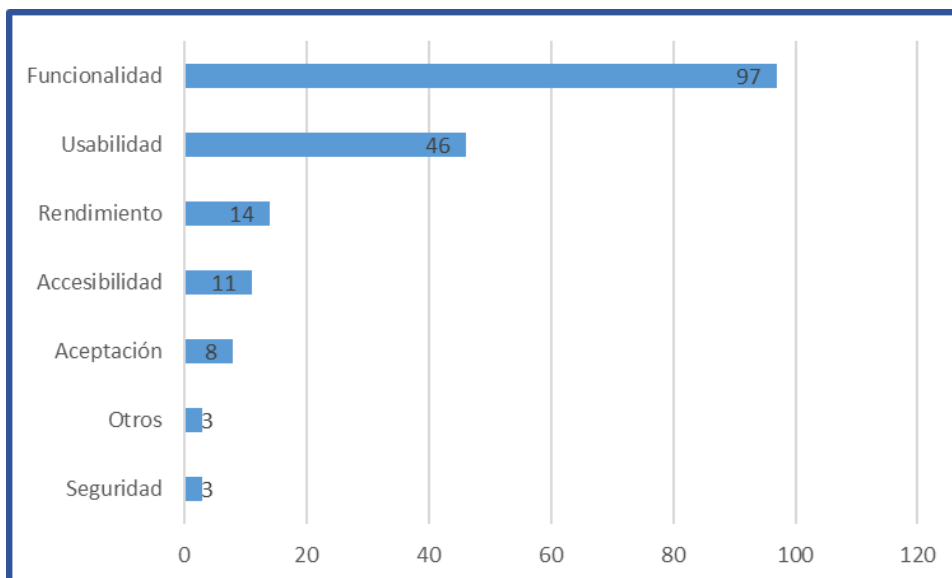


Figura 3.7: Tipo de evaluación de software

En la evaluación de estos productos de software, se han realizado distintos tipos de pruebas con el fin de garantizar el correcto funcionamiento del producto de software. La Figura 3.8 muestra las diferentes pruebas realizadas y el número de veces que han sido efectuadas.

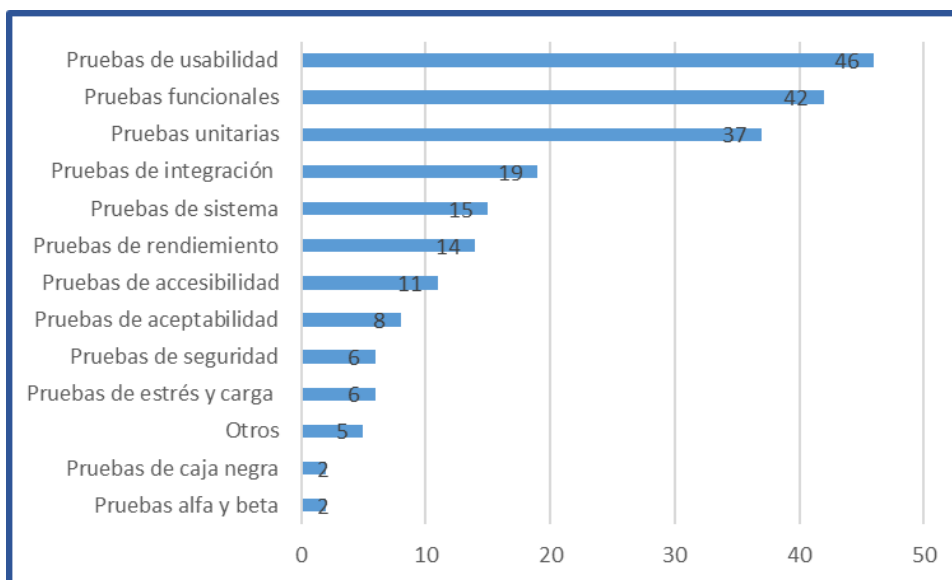


Figura 3.8: Tipos de pruebas

Por otro lado, se encontró que se han utilizado distintas herramientas que permiten la ejecución y automatización de estas pruebas (Figura 3.9). En el caso de evaluar la

funcionalidad del producto en su mayoría se realizan casos de prueba como herramienta para la evaluación del producto de software (Figura 3.10).

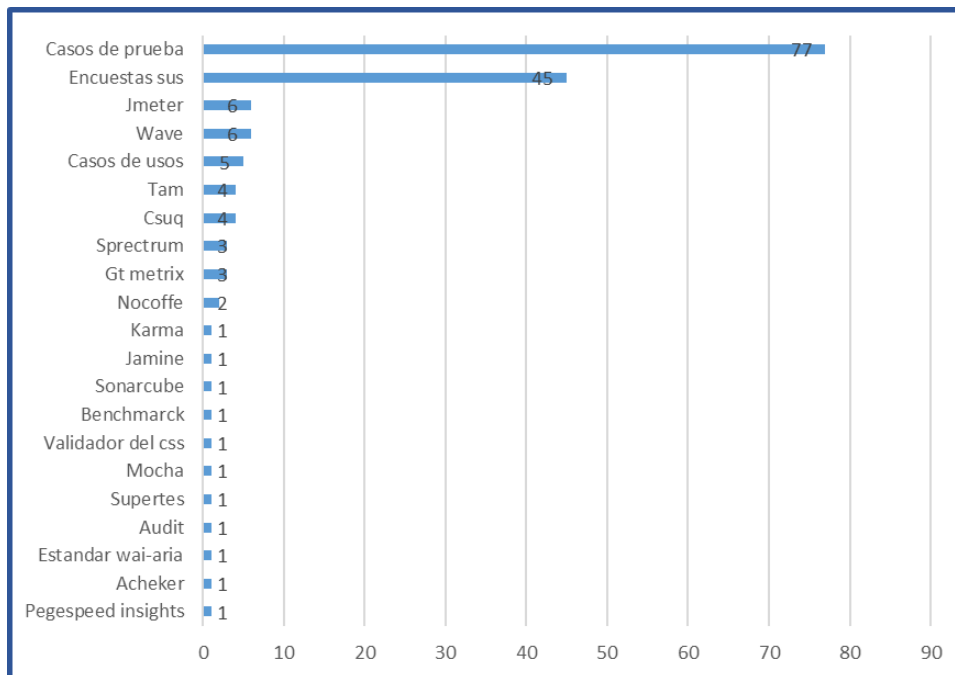


Figura 3.9: Herramientas utilizadas en pruebas

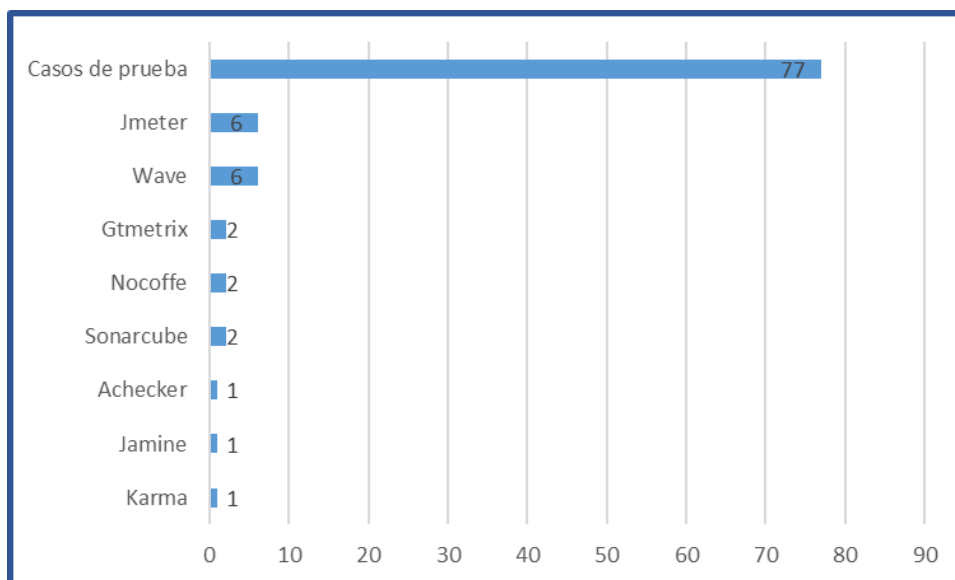


Figura 3.10: Herramientas utilizadas para pruebas funcionales

Cuando se trata de medir la usabilidad, en el 100% de los casos se realizan encuestas las cuales siguen los lineamientos de SUS (System Usability Scale) y CSUQ (Computer System Usability Questionnaire) o una adaptación de ellas.

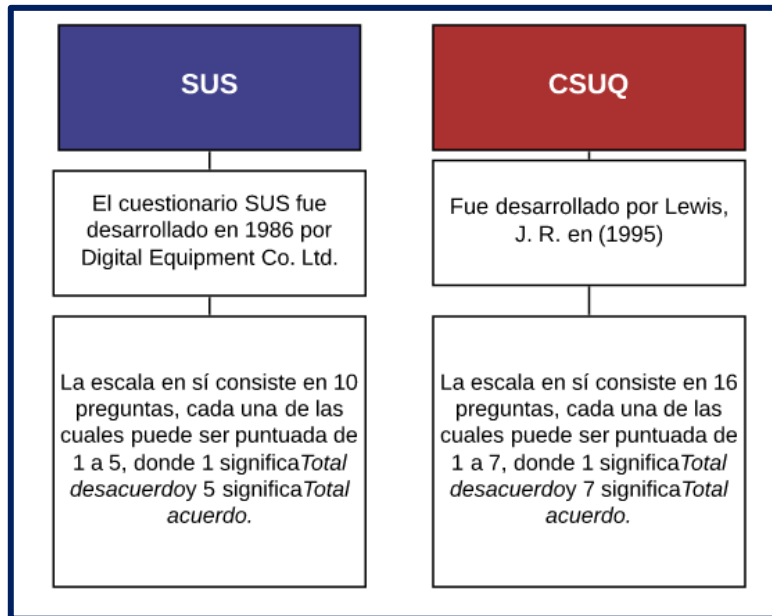


Figura 3.11: Descripción de las encuestas SUS y CSUQ respectivamente las cuales se utilizan para evaluar la usabilidad.

La evaluación de la aceptabilidad se realizó mediante el modelo TAM en su mayoría. La Figura 3.12 muestra el número de veces que se ha utilizados este modelo en los proyectos de titulación.

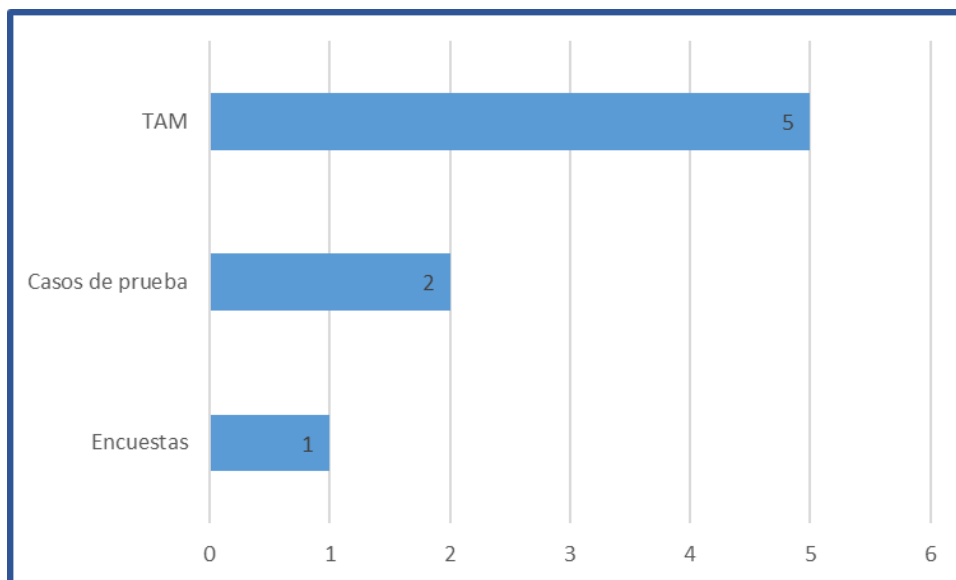


Figura 3.12: Herramientas para medir la aceptabilidad

En la FIS, se han utilizado herramientas como JMeter y GTMetrix para evaluar el rendimiento en los proyectos de titulación. En la Figura 3.13 se describen estas dos herramientas.

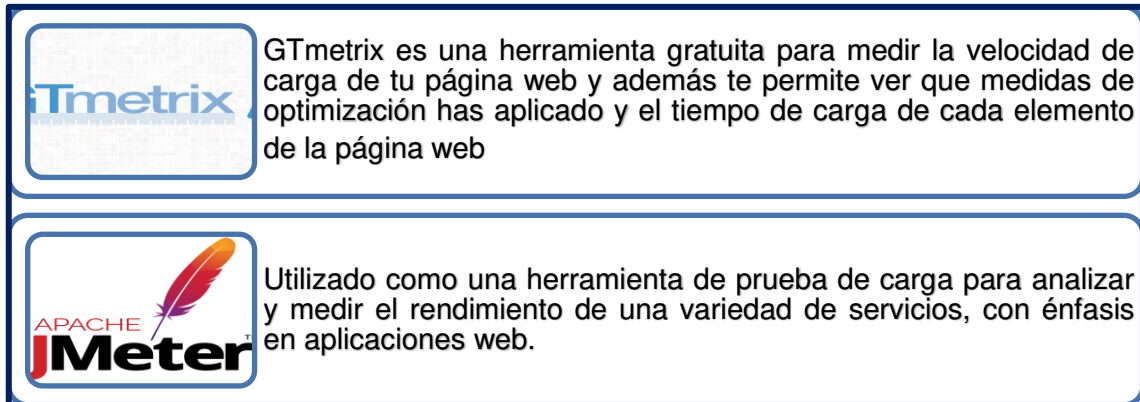


Figura 3.13: Descripción de GTmetrix - Jmeter

Al hablar de accesibilidad, en la FIS se ha venido utilizando en la mayoría de los casos la herramienta WAVE la cual se utiliza para verificar la accesibilidad que existe en las páginas web. WAVE es un conjunto de herramientas de evaluación de software que ayuda a los autores de páginas web a hacer que su contenido sea más accesible para las personas con discapacidades. WAVE puede identificar muchos errores de accesibilidad y de las pautas de accesibilidad al contenido web (WCAG), pero también facilita la evaluación humana del contenido web. La Figura 3.14 representa el número de veces que se ha utilizado la herramienta en los proyectos de titulación.

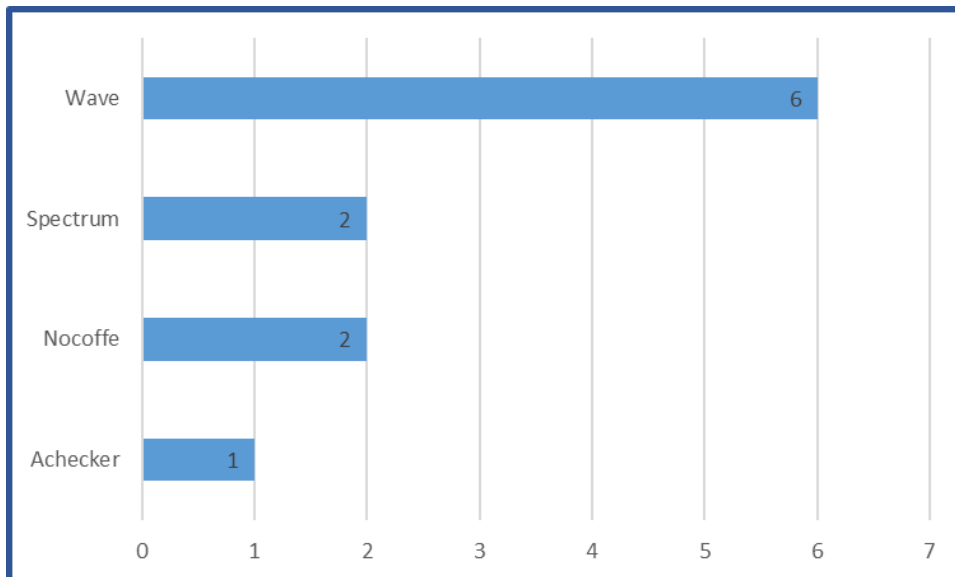


Figura 3.14: Herramientas utilizadas para medir la accesibilidad

A continuación, se realizó un análisis respecto al tipo de evaluación de software con el tipo de software desarrollado. Es importante notar que la funcionalidad ha sido evaluada independiente del software desarrollado (Tabla 3.1).

Tabla 3.1: Tipos de evaluación de software con respecto al tipo de software desarrollado

	Web	Móvil	Juego 3d	Bases de datos	Realidad aumentada	Datamart	Etapa de diseño	Minería de datos	Streaming	Biblioteca de clases	Bpm	Cifrado
Funcionalidad	62,0	40,9	66,7	50,0	100,0	100,0	50,0	100,0	50,0	100,0	50,0	50,0
Usabilidad	16,7	36,4	33,3	0,0	0,0	0,0	50,0	0,0	50,0	0,0	50,0	0,0
Accesibilidad	9,3	9,1	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
Aceptación	4,6	4,5	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
Rendimiento	2,8	9,1	0,0	50,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
Confiabilidad	0,9	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
Seguridad	2,8	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	50,0
Navegabilidad	0,9	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0

	<i>Frecuencia de uso alta (>66%)</i>
	<i>Frecuencia de uso media (entre 33% y 66%)</i>
	<i>Frecuencia de uso baja (<33%)</i>
	<i>Frecuencia de uso nula (0%)</i>

3.2.2 Frameworks utilizados

Entre los frameworks más utilizados están aquellos que permiten evaluar el rendimiento, accesibilidad, y usabilidad del Software. (Figura 3.15). La Figura 3.15 evidencia el número de veces que se ha utilizado la herramienta en los diferentes proyectos de titulación.

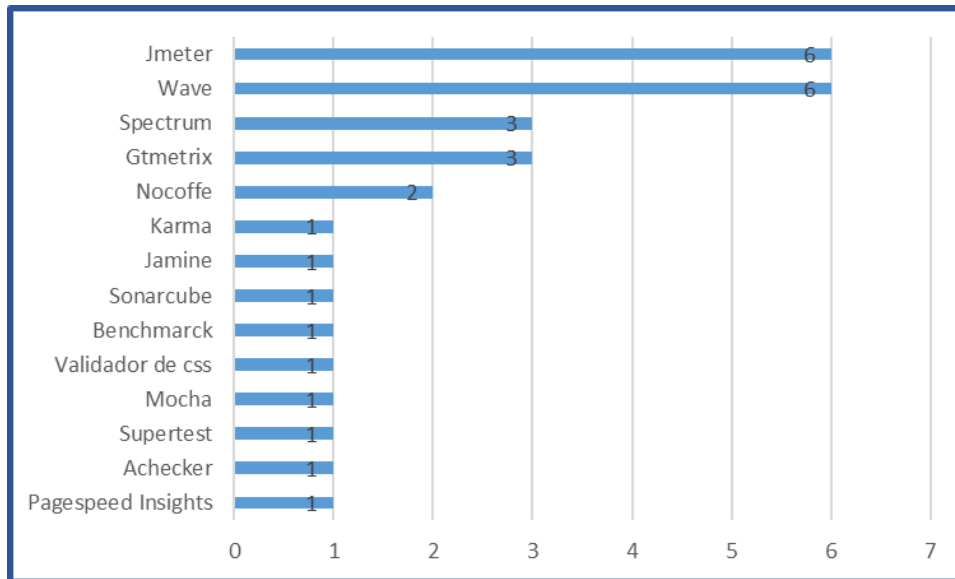


Figura 3.15: Frameworks más utilizados

3.2.3 Métricas de evaluación de software

En la etapa de evaluación de software en la mayoría de los casos no se ha utilizado métricas que permitan realizar la evaluación del producto terminado, sin embargo, entre las métricas más utilizadas se encuentran aquellas que forman parte de la ISO 25000 (Figura 3.16).

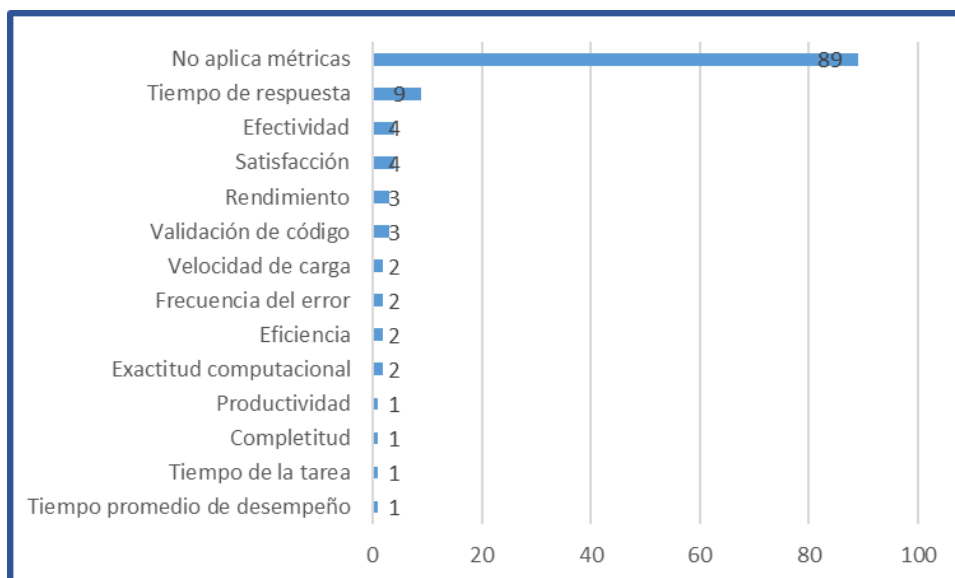


Figura 3.16: Métricas de evaluación de software

3.2.4 Estándares utilizados

Para la evaluación de la calidad del software se ha utilizado el estándar ISO 25000 como un referente de evaluación de software, aunque en tan solo 3 de los 137 analizados. Mientras que los otras no se rigen por ningún estándar.

3.3 Comparación de resultados obtenidos entre papers y proyectos de titulación

3.3.1 Herramientas de evaluación de software

En la Tabla 3.2, se puede observar un despliegue comparativo y detallado de los resultados obtenidos después de la SLR y el análisis documental realizados durante el presente trabajo de titulación. Es posible así observar los resultados por categoría, código e ítem obtenido durante el trabajo investigativo. Además, la Tabla 3.2 permite la comparación directa entre los documentos científicos analizados, y los proyectos de titulación de la facultad. De esta manera, por ejemplo, se puede determinar que mientras en la facultad se utilizan Frameworks como Jmeter, Wave o GTmetrix, los documentos científicos mencionan otros frameworks como CMMI, UMUX, UMARA, entre otros. Es posible entonces identificar diferentes herramientas como frameworks, métricas, tipos de pruebas o las diferentes normas que se pueden utilizar para evaluar un producto de software dependiendo del tamaño del proyecto, de la arquitectura de este o del público al que esté destinado, brindando así un abanico aún más amplio al momento de evaluar un sistema. En color verde se muestran los elementos coincidentes de cada categoría.

3.3.2 Frecuencias de elementos encontrados

La Tabla 3.3 muestra los elementos con las frecuencias de codificación más altas de los resultados obtenidos después de haber realizado la SLR a los documentos académicos y el análisis documental realizado a los proyectos de titulación de la FIS. En esta Tabla es posible visualizar los ítems que resultaron más relevantes según cada grupo de categoría analizada, lo que puede sugerir su relevancia según cada caso. De esta manera por ejemplo se puede determinar que, según los documentos académicos, la norma ISO que mayormente estuvo presente durante el análisis es la 9126, con 10 apariciones durante toda la SLR, mientras que en la facultad la norma ISO que fue mencionada mayoritariamente fue la norma 25000, pero con un total de solo 3 apariciones. El análisis debe hacerse con más profundidad, sin embargo. Puesto que, aunque en los 137 documentos de la facultad solo 3 mencionaron utilizar la Norma ISO25000, esta norma es más actual que la norma ISO9126 a la cual también la cobija. No obstante, para temas

prácticos de evaluación de productos pequeños de software la norma ISO 9126 podría cubrir la evaluación de software de manera adecuada.

Tabla 3.2: Comparación: Papers (ACM - IEEE) vs Proyectos de titulación

Categoría	Código	Artículos Científicos	Proyectos de Titulación
Frameworks utilizados para la evaluación	ARTfra	CMMI UMUX WEBCAT SAMATE UMARA MWAF DESMET	JMeter Wave Spectrum Gtmetrix NoCoffee
Estándares utilizados	ARTest	ISO/IEC 25000 ISO/IEC 9126 SO/IEC 12207 ISO/IEC 14598 ISO/IEC 9000	ISO/IEC 25000
Métricas de evaluación utilizadas	ARTmtr	Efectividad Rendimiento Tiempo de respuesta Validación de Código	Efectividad Rendimiento Tiempo de respuesta Validación de Código Satisfacción
Características de software evaluadas	SOFcar	Accesibilidad Funcionalidad Rendimiento Usabilidad Adaptabilidad Arquitectura Calidad Disponibilidad Documentación Eficiencia Ergonomía Escalabilidad Fiabilidad Interfaz Interoperabilidad Modificabilidad Portabilidad Precisión Privacidad	Accesibilidad Funcionalidad Rendimiento Usabilidad Accesibilidad Aceptabilidad Confiabilidad Seguridad
Pruebas realizadas	PROpru	Prueba de rendimiento Encuesta Abstracción Pruebas unitarias Pruebas de regresión Controles de flujo Pruebas de penetración Complejidad ciclomática Complejidad Mc Cabe Complejidad Halstead Observación	Pruebas de rendimiento Pruebas funcionales Pruebas de aceptación Pruebas de usabilidad Pruebas unitarias Pruebas del sistema Pruebas de accesibilidad Prueba de seguridad Pruebas de integración Pruebas de stress y carga Pruebas de caja negra Pruebas alfa y beta
Proceso que se siguió al evaluar	PROproc	ISO / IEC 14598	ISO / IEC 14598

Tabla 3.3: Comparación entre códigos encontrados más frecuentes en los documentos de investigación y proyectos de titulación la EPN (Escuela Politécnica Nacional)

Categoría	Papers		Proyectos de Titulación	
	Elemento	Número documentos (Total 134)	Elemento	Número documentos (Total 137)
Estándar más utilizado	ISO/IEC 9126	10	ISO/IEC 25000	3
Framework más utilizado	CMMI	6	JMeter/ Wave	7
Herramienta más utilizada	Findbugs	2	Casos de prueba	77
Prueba más utilizada	Encuesta	23	Pruebas funcionales	42
			Pruebas de usabilidad	46
Característica más utilizada	Usabilidad	13	Funcionalidad	97

La Tabla 3.4 nos permite realizar una comparación entre el número total de códigos encontrados tanto en los artículos científicos como en los proyectos de titulación. Existe un mayor número de repeticiones de los códigos en los proyectos de titulación debido a que cada uno de los proyectos de titulación desarrolla un producto de software. Por lo tanto, en esos proyectos, al parecer, se necesita evaluar la calidad de los productos con enfoque en lo que son las características de software, el tipo de pruebas a realizar y los frameworks a utilizar para dicha evaluación de software. Sin embargo, cuando hablamos de estándares encontramos que en los artículos científicos existe un mayor número de estándares encontrados esto es debido a la diversidad de criterios de los investigadores con respecto a la utilización de estándares para evaluar software.

Tabla 3.4: Comparación entre el total de códigos encontrados en los documentos científicos y proyectos de titulación EPN

Categoría	Artículos Científicos	Proyectos de Titulación
	Total de Códigos encontrados	Total de Códigos encontrados
Frameworks utilizados para la evaluación de software	21	29
Estándares utilizados	21	3
Métricas de evaluación de software utilizadas	15	38
Características de software evaluadas	33	182
Pruebas Utilizadas	14	213

3.4 Discusión

Para realizar la evaluación de un producto de software es necesario tomar en cuenta mecanismos estandarizados y repetibles como metodologías, frameworks, características, pruebas, métricas, estándares, procesos. La SLR realizada permitió tener una imagen sobre las principales prácticas utilizadas para evaluar productos de software resultantes de actividades de investigación. Mientras que el análisis documental nos presenta una imagen de como se ha venido realizando el proceso de evaluación de software en la FIS. Si bien, ninguno de los dos resultados refleja la realidad de la evaluación de productos de software que se realiza a nivel de la industria del software, la comparación entre ambos resultados obtenidos permite obtener una serie de resultados que sirven como base para un primer acercamiento para la proposición de un conjunto de lineamientos para evaluar de los productos de software en la Facultad. Los lineamientos de evaluación de software, deben considerar las preferencias por ciertas prácticas “establecidas”, rescatando sus fortalezas, e intentando reforzar o introducir perspectivas poco o nada consideradas hasta la fecha. Llama la atención, por ejemplo, que, al hablar de metodologías para evaluación de productos de software, se observó que en la FIS no se sigue ninguna de ellas mientras que en los documentos científicos si se evidenció la presencia de modelos que permiten evaluar un software como los modelos de Bohem (Bohem, 1981) o de McCall (Miao, 2012). Este resultado es particularmente llamativo, dado que un modelo ofrece la ventaja de definir las características con sus respectivas métricas, lo cual permite abarcar de una mejor manera todas las características para la evaluación de software. Una respuesta a esto puede encontrarse en el hecho de que los modelos suelen ser “pesados” en cuanto a carga de trabajo y, probablemente, los tiempos destinados a la evaluación de los productos de software de la FIS son reducidos y, por tanto, poco compatibles con la demanda de tiempo requerido en los modelos encontrados en los artículos.

En cuanto a los Frameworks más utilizados para evaluar un producto de software en un proyecto de titulación, los más encontrados son aquellos que permiten medir el rendimiento o accesibilidad como es el caso de Jmeter o Wave; mientras que en los documentos científicos se encontró con la utilización de frameworks como CMMI, que se centran más en el proceso de desarrollo. Otro framework para evaluación de software que se encontró en la revisión de papers, y que se podría usar en la FIS es MyFeps el cual es un framework de Calidad de Software (Modelo, Proceso y Herramientas) que brinda apoyo al proceso de evaluación de software en su totalidad. Está basado en normas internacionales (ISO/IEC 9216, ISO/IEC 14598, e ISO/IEC 25000 SQuaRE.) (Ramos, 2015). Sin embargo, tanto

MyFeps como CMMI suelen orientarse más hacia procesos de calidad que se mantendrán en el tiempo y, por tanto, son muy aplicables en la industria del software, pero poco aplicables en proyectos puntuales y relativamente efímeros como los trabajos de titulación. Distinto sería el caso si los proyectos de desarrollo se canalizaran dentro de estructuras como laboratorios o fábricas de software de la FIS, en donde introducir estos frameworks en las prácticas de estas estructuras, se traduciría en mejoras de calidad puntual para los proyectos de titulación que se implementen allí.

Por otro lado, cuando se llevó a cabo el análisis de los documentos científicos, se observó mayor variedad en los estándares utilizados para evaluar productos de software, siendo el más utilizado el estándar ISO/IEC 9126, que en 2005 fue reemplazado por el estándar ISO/IEC 25000. Estos estándares se centran más en el producto, a la vez que define subcaracterísticas para la calidad interna, externa y de uso. Se puede decir que ISO 25000 intenta definir lo más preciso posible cada una de las características y subcaracterísticas, a diferencia del estándar ISO/IEC 9126, que se basa en el modelo de McCall, y que únicamente contempla la calidad interna y externa. No obstante, al igual que en el caso de los frameworks, los estándares suelen ser documentos que proponen una elevada carga de trabajo para las tareas de evaluación de software. Si bien son completos, su uso parece no ser especialmente atractivo para el caso de proyectos de titulación debido a que la utilización de ellos implicaría un aumento en el tiempo de entrega del producto de software. Si bien se observó que existe una gran diferencia entre los estándares y frameworks utilizados para la evaluación de software entre lo que se hace en la facultad con lo que se reporta en los documentos científicos analizados, en cuanto a las métricas usadas para analizar el software hubo una notable mayor coincidencia (Tabla 3.2). En los trabajos de titulación de la FIS, al igual que en los artículos científicos, se utilizaron métricas de Efectividad, Rendimiento, Tiempo de respuesta y Validación de Código. Cabe mencionar que en la FIS también se utilizaron métricas de Satisfacción. El uso de métricas, más ligeras y directas que un modelo o un estándar, refuerza la idea de la preferencia por métodos más rápidos de evaluación de software. Las métricas coincidentes se concentran en la evaluación no funcional del software, dejando solo la Satisfacción como mecanismo de medición de la percepción del usuario por el producto. En ese sentido, la realización de pruebas con usuarios finales, si bien son altamente recomendables, involucran un esfuerzo que aparentemente, y solo considerando hasta ahora las métricas, parece ser poco privilegiado en los trabajos de titulación.

Con respecto a las características a ser evaluadas se observó que para evaluar la usabilidad en la FIS se ha utilizado los cuestionarios SUS y CSUQ (Figura 3.11) mientras

que en los papers se ha utilizado Umux. Umux es una herramienta que permite realizar un cuestionario más corto y alineado a la definición de usabilidad ISO 9241 (efectividad, eficiencia y satisfacción) (Finstad, 2010). Umux-Lite representa una mejora de la versión anterior (Lewis, 2015) y permite evaluar tanto que las capacidades del sistema cumplan con mis requisitos y que sea fácil de usar. Umux-Lite puede ser una alternativa interesante a considerar cuando se trate de medir usabilidad.

De los resultados cabe recalcar que la funcionalidad es una de las características que siempre se ha evaluado en la FIS, sin importar el tipo de software que se haya desarrollado (Tabla 3.1). Para hacer esta evaluación de un producto de software, se han utilizado principalmente casos de prueba (Figura 3.10) con el fin de determinar y validar el funcionamiento correcto del software. Algunas alternativas de evaluación de software se describen a continuación.

De la evaluación de los dos corpus documentales, cuatro características prevalecieron coincidentemente tanto en los artículos científicos como en los trabajos de titulación (Tabla 3.2): funcionalidad, usabilidad, accesibilidad y rendimiento. En consecuencia, estas cuatro características han sido tomadas en cuenta para la proposición de unos lineamientos de evaluación de software básicos para la evaluación de futuros productos de software. Detalles de esta recomendación se muestran en las siguientes secciones.

A continuación, podemos observar un resumen de la discusión planteada anteriormente como lo muestra la Figura 3.17.

Discusión de los resultados obtenidos

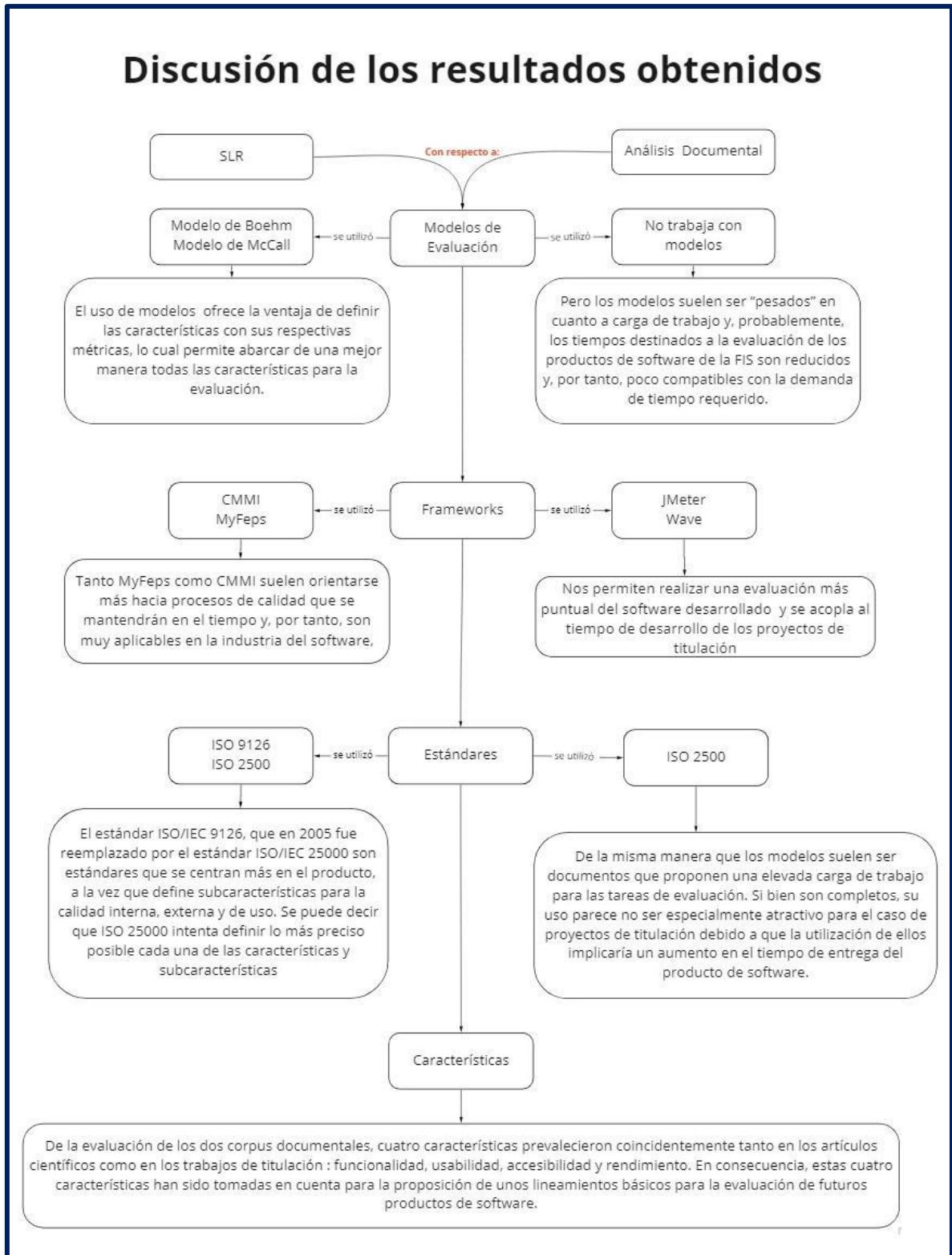


Figura 3.17: Resumen de la discusión de resultados

3.5 Lineamientos para la evaluación de productos de software

Para poder desarrollar un producto de software profesionalmente, es necesario utilizar mediciones que permitan evaluar la calidad del análisis y los modelos de diseño, el código fuente y los casos de prueba que se hayan creado al momento de aplicar la Ingeniería de Software (Pressman, 2010). Para poder conseguir estas evaluaciones de calidad, se deben utilizar medidas técnicas que evalúan la calidad con objetividad, no con subjetividad (Far, 2008). Después de haber realizado un extenso análisis tanto a los documentos científicos, como a los trabajos de titulación y de haber comprendido la importancia de estandarizar un método para realizar la evaluación de un producto de software, se ha desarrollado un grupo de lineamientos de evaluación de software sugeridos para la evaluación de software de futuros proyectos de titulación para al FIS.

Durante el análisis de los documentos, se identificó un total de once características tomadas en cuenta para evaluar software según los documentos científicos, como se muestra en la Figura 3.4, mientras que, por otro lado, en los trabajos de titulación se encontró un total de siete características tomadas en cuenta al momento de evaluar un producto de Software. Como se mencionó en la sección previa, de estos dos resultados, se identificaron cuatro características que han sido tomadas en cuenta para ambos casos, así como la relevancia de estas, por este motivo se tomará en cuenta para la evaluación de los productos de software las siguientes características mínimas: funcionalidad, usabilidad, accesibilidad y rendimiento, dichas características permitirán garantizar la calidad del producto de software. Se ha escogido estas cuatro características debido a que se encuentran presente tanto en los modelos, como en los estándares, al igual que en la evaluación de productos de software en la FIS.

La funcionalidad proporcionará una idea clara de cómo se encuentra el software con respecto a los requisitos previamente establecidos. Es decir, si el producto de software cumple o no con el objetivo para el cual fue desarrollado. Por otro lado, evaluar la usabilidad permitirá conocer si el producto de software desarrollado cumple con las perspectivas del usuario, es fácil de entender y usar por este, lo cual garantizara el uso de este. La accesibilidad, por su lado, garantiza que el software pueda ser utilizado por personas con capacidades especiales en cualquier contexto. Finalmente, el rendimiento ayuda a conocer cómo se encuentra el producto de software y cuál es la capacidad de este. La Tabla 3.5

muestra las métricas identificadas en nuestra revisión y que aplicarse a las cuatro características mencionadas.

Tabla 3.5: Métricas recomendadas

Métricas recomendadas	
Funcionalidad	Compleitud.
	Corrección de errores.
Usabilidad	Satisfacción.
Rendimiento	Utilización de recursos.
	Comportamiento en el tiempo
Accesibilidad	Hay que asegurar que cumplan con WCAG 2.0 nivel AA.

En la Tabla 3.6 se detalla un resumen de lineamientos de evaluación de software recomendados para la evaluación de software en los proyectos de titulación de la FIS, con lo que se pretende dar una sugerencia de homogenización para la etapa de pruebas realizadas durante el desarrollo de nuevos productos de software en los proyectos de titulación de la facultad.

Tabla 3.6: Lineamientos de evaluación de software

Característica evaluada	Métrica	Parámetros	Herramienta	Proceso	Puntuación
Funcionalidad	Compleitud Funcional	$X = A / B$ A = Número de funciones que están incorrectas o que no fueron implementadas B = Número de las funciones establecidas en la especificación de requisitos Dónde: $B > 0$	Casos de prueba	1. Obtener la especificación de los requisitos. 2. Realizar casos de prueba para cada función establecida en la especificación de requisitos. 3. Cuantificar el número de funciones correctas, incorrectas, funciones implementadas, no implementadas. 4. Resolver la ecuación para medir la funcionalidad.	$0.95 < x < 1$ Muy Bueno $0.90 < x < 0.94$ Bueno $0.75 < x < 0.89$ Aceptable $0.50 < x < 0.74$ Regular $x < 0.49$ Deficiente

Característica evaluada	Métrica	Parámetros	Herramienta	Proceso	Puntuación
	Corrección de errores	$X = A/B$ A = Número de fallas corregidas en la fase de diseño/codificación/pruebas B = Número de fallas detectadas en las pruebas Dónde: $B > 0$	Casos de prueba	1. Obtener los resultados de los casos de prueba realizados. 2. Cuantificar el número de fallas detectadas en los casos de prueba 3. Cuantificar el número de fallas corregidas	$0.95 < x < 1$ Muy Bueno $0.90 < x < 0.94$ Bueno $0.75 < x < 0.89$ Aceptable $0.50 < x < 0.74$ Regular $x < 0.49$ Deficiente
Usabilidad	Satisfacción	$X = A/B$ A= Numero de preguntas con respuesta satisfactorias B = Número total de preguntas realizadas en el cuestionario Dónde: $B > 0$	UMUX o UMUX-LITE	1. Realizar la respectiva encuesta a los usuarios del producto de software (Umux/Umux Lite) 2. Cuantificar los resultados obtenidos	$0.95 < x < 1$ Muy Bueno $0.90 < x < 0.94$ Bueno $0.75 < x < 0.89$ Aceptable $0.50 < x < 0.74$ Regular $x < 0.49$ Deficiente
Rendimiento	Utilización de recursos	Utilización del CPU Utilización de memoria Utilización De dispositivos de E/S	Administrador de tareas	Ingresar al administrador de tareas y proceder a: 1. Medir la cantidad de CPU que es usado para realizar una tarea 2. Medir la cantidad de memoria que es usado para realizar una tarea 3. Medir la cantidad de tiempo que los dispositivos de E/S pasan ocupados para realizar la tarea.	
	Rendimiento	Tiempo de Carga pruebas de estrés	Gtmetrix Jmetrix	1. Con la ayuda de los framework mencionados evaluar el rendimiento del producto de software.	
Accesibilidad	WCAG 2.0.	Cumplimiento de WCAG	Wave	1. Utilizar la herramienta Wave.	Cumplimiento de WCAG nivel AA

A continuación, en la Figura 3.18 se describe de manera resumida los lineamientos de evaluación de software propuestos para evaluar un producto de software desarrollado en la FIS.

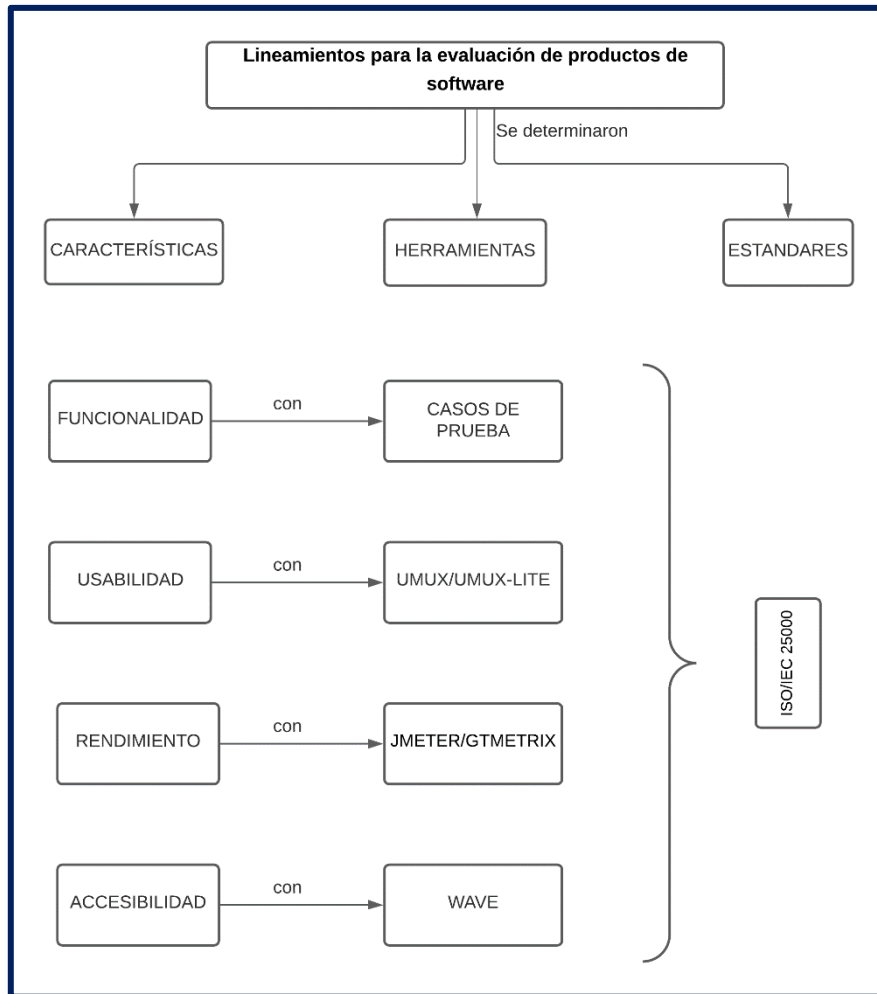


Figura 3.18: Propuesta de lineamientos para la evaluación de productos de software

3.6 Herramientas recomendadas para la evaluación de las características del producto de software recomendadas

Después de haber analizado los proyectos de titulación y determinar qué características han sido las más evaluadas tanto en la literatura como en los trabajos de titulación, aparecen cuatro características que destacan y que se pueden tomar en cuenta para trabajos futuros desde una perspectiva evaluación de software mínima. El objetivo principal de esta aproximación es evaluar el software de una manera rápida pero que nos permita obtener un producto de software de calidad. Las características consideradas son funcionalidad, usabilidad, rendimiento y accesibilidad; debido a que fueron aquellas que coincidieron en los corpus analizados en las dos etapas de este trabajo.

Evaluación de funcionalidad mediante el uso de casos de prueba. Los casos de prueba permiten definir los ítems que necesitan de una validación con el fin de asegurar que el sistema funciona correctamente y que está construido con un alto nivel de calidad.

Es recomendable crear casos de prueba que sean simples y con instrucciones claras ya que el o los evaluadores, al ser diferentes personas al o a los autores, deben poder ejecutar los casos de prueba con facilidad. Por tanto, es necesario usar un lenguaje asertivo por ejemplo “*vaya a la página de inicio*”, “*ingresar usuario*”, “*hacer clic en el botón verde*”. Estas instrucciones facilitan la comprensión de los pasos de prueba y permiten que la ejecución sea más rápida y fluida. Adicionalmente cada caso de prueba debe estar correctamente documentado lo cual permitirá verificar o corregir los problemas encontrados al momento de realizar la evaluación del producto de software.

El objetivo final de un caso de prueba es asegurar que se cumplan los requisitos del cliente y que el producto sea fácil de usar. Por este motivo se deben crear casos de prueba tomando en cuenta la perspectiva del usuario final. A su vez, debería reducir al máximo las subjetividades del evaluador, para reflejar los mismos resultados cada vez que se realice, sin importar quién o cuantas veces se desarrolle la prueba. Un ejemplo de una ficha de caso de prueba se muestra en la Tabla 3.7.

Tabla 3.7: Ejemplo de un caso de prueba

Caso de prueba			
ID Caso de prueba:			
Nombre del caso de prueba:			
Identificador de requerimientos:			
Nombre del evaluador:			
Ambiente	Inicio	Acciones	Finalización
Resultados:			

Evaluación de usabilidad mediante la utilización de Umux. UMUX es una herramienta ligera de evaluación de software cualitativa, que ha sido diseñada para medir la usabilidad general de un sistema. La intención de UMUX es ofrecer una alternativa más corta al cuestionario SUS de 10 ítems (Tabla 3.8), se prefieren cuestionarios más cortos cuando el tiempo en un estudio de usabilidad es limitado o cuando se necesita una medida de usabilidad como parte de una encuesta más grande. Mientras que SUS evalúa la usabilidad y la capacidad de aprendizaje percibidas, UMUX se enfoca en la usabilidad evaluando la efectividad, la eficiencia y la satisfacción. Además, UMUX evalúa la usabilidad en base a la definición oficial y estándar establecido por la norma ISO 9241. El cuestionario de UMUX es simple y contempla cuatro ítems (Tabla 3.9) que enumera dos afirmaciones positivas y

dos negativas. UMUX-Lite (Tabla 3.10) es una versión aún más simple que UMUX con un contenido que tiene solo declaraciones positivas. Esta versión más corta imita muy bien el Modelo de Aceptación de Tecnología (TAM) en contenido, proporcionando una medida de utilidad y usabilidad percibidas

Tabla 3.8: Cuestionario SUS

		Muy en desacuerdo					Muy de acuerdo	
		1	2	3	4	5	6	7
1	Creo que me gustaría utilizar este sistema con frecuencia.							
2	Encontré que el sistema es simple.							
3	Pensé que el sistema era fácil de usar.							
4	Creo que podría usar el sistema sin el apoyo de un técnico.							
5	Encontré que las diversas funciones del sistema estaban bien integradas							
6	Pensé que había mucha coherencia en el sistema.							
7	Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente							
8	Encontré el sistema muy intuitivo.							
9	Me sentí muy seguro usando el sistema.							
10	Podría usar el sistema sin tener que aprender nada nuevo.							

Tabla 3.9: Cuestionario UMUX

		Muy en desacuerdo					Muy de acuerdo	
		1	2	3	4	5	6	7
1	Las capacidades [de este sistema] cumplen con mis requisitos.							
2	Usar [este sistema] es una experiencia frustrante.							
3	[Este sistema] es fácil de usar.							
4	Tengo que dedicar demasiado tiempo a corregir cosas con [este sistema].							

Tabla 3.10: Cuestionario UMUX-LITE

		Muy en desacuerdo					Muy de acuerdo	
		1	2	3	4	5	6	7
1	Las capacidades [de este sistema] cumplen con mis requisitos.							
2	[Este sistema] es fácil de usar.							

Es importante comentar dos cosas acerca de UMUX. El cuestionario que propone tiene una longitud ideal, sin embargo, hay que tomar en cuenta que la puntuación es algo más compleja, especialmente si se está interesado en comparar los resultados con una puntuación SUS. Debido a que UMUX combina preguntas positivas y negativas, algunos investigadores han descubierto que crea una prueba de dos factores. Sin embargo, se puede utilizar UMUX-Lite en su lugar, ya que solo contiene declaraciones positivas.

Evaluación de Rendimiento mediante herramientas como JMeter o Gtmatrix. JMeter es una herramienta que facilita la gestión integral de los procesos de pruebas de rendimiento no funcionales que incluye pruebas de: carga, estrés, las pruebas de picos, de resistencia y las pruebas de volumen.

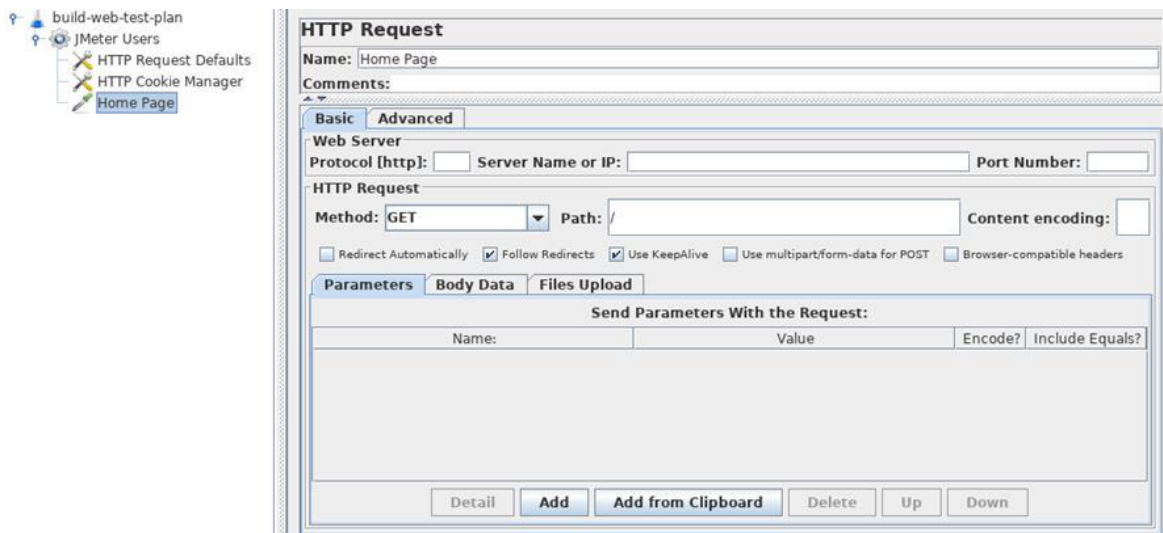


Figura 3.19: Interfaz gráfica de usuario JMeter

Jmeter es una aplicación de código abierto, totalmente basada en Java, que cuenta con una interfaz gráfica de usuario (Figura 3.19) que ha sido diseñada para probar aplicaciones web, aplicaciones FTP, pruebas funcionales, conexiones de bases de datos JDBC

(conectividad de bases de datos Java), servicios web, conexiones TCP genéricas, procesos nativos del sistema operativo.

GTmetrix por su parte es una herramienta de prueba de velocidad que fue desarrollada por GT.net. GTmetrix permite calcular el rendimiento mediante las siguientes pruebas:

Largest Contentful Paint (LCP) que mide la cantidad de tiempo que tarda en cargarse la mayor parte de la página.

Total Blocking Time (TBT) evalúa la cantidad de tiempo que una página se bloquea antes de que un usuario pueda interactuar con ella. El CSS y el JS de bloqueo de renderizado pueden tener un gran impacto en el TBT.

Cumulative Layout Shift (CLS) prueba al desplazamiento de elementos mientras se carga una página. (Jackson, 2021).

Evaluación de accesibilidad mediante el uso de WAVE. La herramienta Web Accessibility Tool o WAVE por sus siglas en inglés, permite evaluar la accesibilidad de una página web, ya sea indicando la URL de la página a analizar o cargando directamente la página en la interfaz de la herramienta. Al igual que otras herramientas similares, WAVE permite elegir un nivel de análisis deseado.

WAVE detecta, por ejemplo, si se está empleando alguna de las siguientes características para mejorar la accesibilidad de una página y desaconseja su uso (Figura 3.20):

Enlaces invisibles en la misma página: aconseja hacerlos visibles para los usuarios con discapacidad motora.

Atajos de teclado: desaconseja su uso debido a posibles conflictos con los atajos de otros programas (por ejemplo, los atajos del propio navegador).

Orden de tabulación indicado: desaconseja su uso debido a irregularidades en su implementación.



Figura 3.20: Alertas de WAVE sobre el uso de enlaces invisibles y atajos de teclado.

Es importante aclarar que WAVE no puede decidir si un contenido web es accesible o no; solo un ser humano/probador puede determinar la verdadera accesibilidad. Al finalizar la prueba de accesibilidad, obtendremos un resumen como el indicado en la Figura 3.21 , donde se especifican los detalles, información y recomendaciones adicionales sobre las correcciones sugeridas para mejorar la accesibilidad de la aplicación.

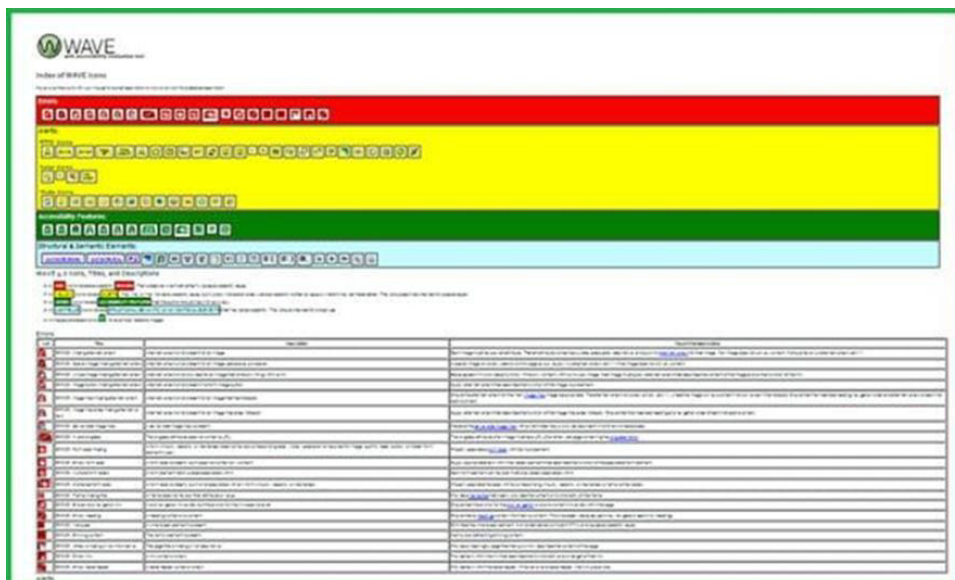


Figura 3.21: Resultados del uso de la herramienta WAVE.

En el Ecuador la accesibilidad informática es un requisito fundamental para todos los sitios web ecuatorianos que presten un servicio público ya que estos deben cumplir con los requisitos de accesibilidad WCAG 2.0 nivel AA.

3.7 Estándares recomendados para evaluación de productos de software

Al momento de realizar la evaluación de software en la FIS, la tendencia es de no usar estándares. Esto puede responder a las limitaciones en cuanto a duración y extensión de los trabajos de titulación. No obstante, en los pocos casos en los cuales se evalúa el software con un estándar, se lo hace guiándose en la norma ISO/IEC 25000, la cual por otro lado fue recurrente en los artículos científicos analizados. De la revisión realizada, esta norma parece constituir una referencia a ser considerada para la evaluación de los productos de software de los trabajos de titulación de la FIS.

La ISO/IEC 25000 es una norma madura que proporciona recomendaciones y directrices para la evaluación de productos de software (Cai, 2016). Es una evolución de las normas ISO/IEC 9126, que describe las particularidades de un modelo de calidad del producto software, y de la ISO/IEC 14598 que abordaba el proceso de evaluación de software. (AENOR, 2005)

La norma ISO/IEC 25000 está conformada por cinco divisiones:

- 2500n División de la gestión de Calidad
- 2501n División del modelo de Calidad
- 2502n División de la medición de Calidad
- 2503n División de requisitos de Calidad
- 2504n División de evaluación de Calidad

Cada división agrupa un conjunto de parámetros y requisitos que permiten garantizar la calidad de software y es por esta razón que la norma es de gran ayuda al momento de realizar una evaluación de software (AENOR, 2005).

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Existe una gran diversidad de formas de evaluar un producto de software, esto se debe a que cada producto de software se adapta a las necesidades para las que fue desarrollado, por lo tanto, dependiendo del tipo de software se realizará su evaluación. La evaluación de software es una de las etapas necesarias para garantizar la calidad de un producto, que cuenta con amplio bagaje de herramientas y frameworks que se han ido desarrollando en función de la evolución del desarrollo del software, pero que al mismo tiempo una de las etapas que aún requiere investigación y estandarización en los proyectos desarrollados en la FIS, aun cuando existen normas como la ISO/IEC 25000 que brinda una idea clara de los aspectos, criterios y métodos para la evaluación de un producto de software, no se ha desarrollado un modelo estándar y específico para tratar la evaluación de software.
- Después de haber analizado los diferentes documentos científicos y proyectos de titulación se determinó un conjunto base de lineamientos de evaluación los cuales facilitarán la evaluación, y brindarán una idea de las herramientas que se puede utilizar para la misma, sin embargo, la decisión de las herramientas siempre dependerá del evaluador.
- Luego de haber analizado los diferentes documentos y proyectos de titulación se induce que las actividades de evaluación aplicadas en los proyectos de titulación de la FIS están condicionados a la duración y plazo de estos. Por tanto, son pocos los proyectos en los cuales se aplican frameworks, modelos o normas; privilegiándose la evaluación de características individuales con el soporte, en ciertos casos, de algunas métricas específicas.
- En el análisis de los documentos, se determinaron varias características tomadas en cuenta para evaluar software. De este resultado, se identificaron cuatro características que han sido tomadas en cuenta recurrentemente en la evaluación de productos de software tanto en los documentos de carácter científico como en los trabajos de titulación de la FIS. Estas características fueron: funcionalidad, usabilidad, accesibilidad y rendimiento. Se escogió en consecuencia estas cuatro

características para proponer una base de lineamientos de evaluación de software mínima para la evaluación de proyectos de titulación.

- La funcionalidad de un producto de software fue la característica más tomada en cuenta en ambos corpus analizados. Este resultado puede sugerir una mayor predilección por asegurar que el software cumpla los requisitos establecidos. Este resultado podría incidir en la aplicación de metodologías ágiles en el desarrollo de proyectos de titulación en la FIS

4.2 Recomendaciones

- Se recomienda realizar un estudio para evaluar el estado de prácticas en el campo profesional de desarrollo de software y ajustar los lineamientos de evaluación de software establecidos, incorporando características y métodos utilizados actualmente en la industria. Esto no solo podría ayudar en la mejora de la calidad de los productos de software producidos, sino que también incidiría en la formación de los estudiantes en estos métodos.
- Se recomienda realizar entrevistas y grupos focales con los docentes de la FIS especializados en los diferentes campos de desarrollo de software, para reunir sus criterios en relación con la presente proposición de lineamientos de evaluación de software.
- Se recomienda explorar mecanismos de evaluación financiera para los proyectos. Esta temática estuvo ausente en los documentos analizados, pero su consideración es importante siempre a nivel de proyectos comerciales.

REFERENCIAS BIBLIOGRÁFICAS

- AENOR, I. J. (2005). *NORMAS ISO 25000*. Obtenido de <https://iso25000.com/index.php/normas-iso-25000/5-iso-iec-25000>
- Al-Elaimat, A. (2015). Procedural Assessment Process of Software Quality Models Using Agility.
- Alfredo Mendoza-González, C. R.-R. (2018). An Approach to Make Software Testing for Users.
- Barros, P. R. (2015). Analyzing softwares in medical education focusing on quality standards. *IEEE 28th International Symposium on Computer*.
- Bateman, S. (2009). Interactive Usability Instrumentation . 6.
- Black, P. E. (2007). Software assurance with samate reference dataset, tool standards, and studies.
- Bohem. (1981). *Software Engineering Economics*. Prentice Hall.
- Bosley, J. J. (2013). Creating a Short 'Usability Metric for User Experience' (UMUX) Scale.
- Cai, J. G. (2016). An Induction to the Development of Software Quality Model Standards . 5.
- EIDH. (2021). <https://es.strephonsays.com/symposium-and-vs-conference-7109>. Obtenido de <https://eventoseidh.com/diferencias-entre-seminario-simposio-y-debate/>
- Far, B. H. (2008). A General Purpose Software Evaluation System . 7.
- Finstad, K. (2010). The Usability Metric for User Experience. *Elsevier*, 5.
- Fonseca, R. (2012). *Técnicas de evaluación del software*. Obtenido de <http://www.grise.upm.es/sites/extras/12/pdf/PruebasEstaticasSP.pdf>
- Hofacker, A. (2008). *Rapid lean construction - quality rating model*. Manchester: s.n.
- IBM. (2002). Obtenido de <https://www.ibm.com/docs/es/rtw/9.0.0?topic=phases-performance-testing>
- IBM. (2002). *IBM CORPORATION*. Obtenido de <https://www.ibm.com/docs/es/license-metric-tool?topic=concepts-products-components-bundles>
- IBM. (2006). *Corp*. Obtenido de https://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/core.base_rup/guidances/concepts/acceptance_testing_12A0F152.html
- INEN-ISO/IEC 9126-1. (2001). Obtenido de https://www.normalizacion.gob.ec/buzon/normas/nte_inen_iso_iec_9126-1.pdf
- ISACA. (2010). *CMMI-ISACA*. Obtenido de <https://cmmiinstitute.com/>

- ISO/IEC. (2005). *Software Engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE*. Obtenido de <https://www.iso.org/standard/35683.html>
- ISO/IEC 25010. (2005). *iso25000*. Obtenido de <https://iso25000.com/index.php/normas-iso-25000/iso-25010>
- ISO-29119. (2013). *Software and systems engineering — Software testing — Part 1: Concepts and definitions*.
- Jackson, B. (2021). *Kinsta*. Obtenido de <https://kinsta.com/es/blog/gtmetrix-herramienta-de-test-de-velocidad/>
- Kitchenham, B. (2004). *Procedures for Performing*. UK: Department of Computer Science.
- Koskela, L. (1992). *Application of the new production philosophy to construction*. Finland: VTT Building Technology.
- Lewis, J. R. (2015). Measuring Perceived Usability: The SUS, UMUX-LITE, and AltUsability. *International Journal of Human–Computer Interaction* .
- Miao, X. (2012 de 2012). Method of Military Software Quality Evaluation Based on Multi-Agent Fuzzy Neural Network.
- Morales, V. M. (20 de Mayo de 2000). *Academia Pragma*. Obtenido de <https://www.pragma.com.co/blog/conoce-que-son-las-pruebas-no-funcionales-de-software>
- Moumane, K. (2015). Using ISO 9126 with QoS DiffServ model for evaluating software quality in mobile environments.
- Nestor, A. (2017). Estándares para la calidad de software. . *TIA*.
- Oates., B. J. (2006). *Researching Information Systems and Computing*. SAGE Publications.
- Pérez, M. (28 de Julio de 2021). *Definición de Metodología*. Obtenido de <https://conceptodefinicion.de/metodologia/>.
- Pressman, R. (2010). *Ingeniería Del Software. 7th ed*. Mexico: McGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V., pp.2,3,10,31.
- Ramos, D. (2015). *Framework basado en el estándar de calidad del software ISO/IEC 25000:2005 (SQuaRE) para la evaluación de la calidad de las ontologías*. Obtenido de <https://www.tesisenred.net/handle/10803/365309#page=2>
- Sensagent Corporation. (2014). Obtenido de <http://diccionario.sensagent.com/Verbatim/es-es/>
- Sommerville, I. (2005). *Ingeniería de Software 7ed*. Madrid: Pearson Educación. S.A.
- Strooper, P. (2006). Evaluating software refactoring tool support . 7.

Turrado, J. (10 de Marzo de 2020). *Campus MVP*. Obtenido de <https://www.campusmvp.es/recursos/post/que-son-las-pruebas-de-software.aspx>
Web-Cat-Community. (2019). *Web-CAT: the Web-based Center for Automated Testing*.
Obtenido de <https://web-cat.org/>

ANEXOS

Anexo I: Listado de Artículos de Investigación

Anexo II: Listado de Proyectos de titulación

Anexo III: Tabla de Códigos Artículos de Investigación

Anexo IV: Tabla de Códigos Proyectos de titulación