Universidad Internacional de Ciencia y Tecnología
Facultad de Ciencias de la Computación y Tecnología
Maestría Académica
Desarrollo De Software Con Énfasis En Software Libre

Tesis de Maestría

Aplicación de Modelos de Madurez de Software Libre, en las organizaciones de software en Panamá.

Tutor: Mgtr. Donna A. Roper C. D.

Estudiante: José de Jesús Jiménez Puello

Número de Cédula: 3-96-654

Cohorte: 032016

Tesis Para Optar al Grado de Maestría

Panamá, diciembre 2020

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Página

AGRADECIMIENTO

A todos los desarrolladores de software que amablemente respondieron la encuesta. A la profesora tutora, Magíster Donna Roper por su tiempo y dedicación.

A todas aquellas personas que en una u otra forma, contribuyeron a llevar a feliz término esta investigación y culminar con esta otra aventura académica de mi vida profesional.

A todos ustedes muchas gracias.

DEDICATORIA

A la salud, que sin ella nada es posible. A la vida por las aventuras y proyectos que me ha permitido realizar. Al futuro que me permita alcanzar los sueños pendientes. A mi esposa Magdilia Castillo Gómez, por su apoyo en la montaña rusa del matrimonio. Porque la vida es un viaje en silencio o con ruido, donde todo pasa y se olvida, como algún día también seremos olvidados.

RESUMEN

La investigación realizada es del tipo investigación por encuestas con diseño no experimental. La data fue obtenida a través de una encuesta de tipo digital, durante el bienio 2018-2019. El objetivo principal fue determinar si los desarrolladores de software en Panamá, aplican los modelos de madurez de software libre o de código abierto, en las organizaciones donde laboran, qué software libre utilizan para el desarrollo de sus aplicaciones y qué aplicaciones desarrollan estas organizaciones con software libre. Los resultados demuestran que las organizaciones de desarrollo de software en Panamá casi no utilizan modelos de madurez de software abierto o de código libre. Que existe un desconocimiento de la existencia de modelos de madurez enfocados en software libre y de su utilidad para la selección de software libre. Que la principal razón para utilizar software libre para el desarrollo de productos es por su cero costo y que el tipo de aplicaciones que más desarrollan son las relacionadas a páginas Web y aplicaciones Web.

Palabras claves: Software libre, modelo de madurez, desarrolladores de software.

ABSTRACT

The research carried out is of the survey research type with a non-experimental design. The data was obtained through a digital survey, during the 2018-2019 biennium. The main objective was to determine if software developers in Panama apply the free or open source software maturity models in the organizations where they work, what free software they use to develop their applications and what applications these organizations develop with software free. The results show that software development organizations in Panama hardly use open source or free source maturity models. That there is a lack of knowledge of the existence of maturity models focused on free software and their usefulness for the selection of free software. That the main reason for using free software for product development is because of its zero cost and that the type of applications that are most developed are those related to Web pages and Web applications.

Keywords: Free software, maturity model, software developers.

INTRODUCCIÓN

El software libre nace como un movimiento de libertad. Esa libertada ha permitido que se desarrolle y emerja como un pilar en el desarrollo de software tanto privativo como libre. Al desarrollar con software libre o abierto esto implica que se ponga en duda la continuidad del producto desarrollado desde la perspectiva de calidad. Es decir, si el software es maduro, si en un futuro el software se podrá mejorar de forma continua o por el contrario, que se vuelva obsoleto.

Para garantizar la madurez del software libre se han desarrollado modelos de madurez de software libre (Open Source Maturity Model de Capgemini, Open Source Maturity Model de Navica, Qualification and Selection of Open Source Software, entre otros), que tienen como finalidad evaluar la madurez del software basados en indicadores o características de la madurez del software.

En la presente investigación, se determinó la aplicación de los modelos de madurez de software libre por parte de los desarrolladores de software. Además, el tipo de software libre que utilizan para el desarrollo de sus aplicaciones y las aplicaciones que desarrollan estas organizaciones con software libre.

El desarrollo de la investigación está organizado en los capítulos siguientes: Capítulo 1. Contextualización del problema de investigación. Capítulo 2. Estado del arte. Capítulo 3. Modelos de madurez de software libre. Capítulo 4. Glosario de términos. Capítulo 5. Metodología. Capítulo 6. Resultados y discusión. Capítulo 7. Conclusiones y recomendaciones. 8. Referencias bibliográficas y un Anexo.

CAPÍTULO 1.

CONTEXTUALIZACIÓN DEL PROBLEMA DE INVESTIGACIÓN

CONTEXTUALIZACIÓN DEL PROBLEMA DE INVESTIGACIÓN

1. CONTEXTUALIZACIÓN

El desarrollo de aplicaciones con software libre o abierto, muchas veces implica que se ponga en duda la continuidad del producto desarrollado desde la perspectiva de calidad. Es decir, si la aplicación o software será seguro, si tendrá actualizaciones, soporte técnico, escalabilidad, entre otros factores. Sin embargo su aplicación, aparte de ser más económica, es tan segura como el del software comercial. La madurez del software es primordial en esta selección del software. Es en esta línea, que se plantea esta novedosa investigación.

1.1. Antecedentes y justificación

El uso de software libre o el software de código abierto se ha incrementado en Panamá, al igual que otras latitudes del mundo. En lo particular considero que su uso debe incrementarse porque nos facilita el desarrollo de software o como usuarios entre todos a muy bajo o cero costo. Además, del aporte que todos podemos realizar para la mejora del software libre, y que se haga más robusto en cuanto a competitividad y usabilidad. Debe estar al servicio de la sociedad en todos sus aspectos: educativo, social, científico, humanístico, entre otros. Sobre todo, a disposición de las comunidades más vulnerables que no pueden pagar este servicio, que se ha convertido en un servicio vital dentro de la sociedad de hoy, la sociedad del conocimiento y de la información.

Durante los últimos diez años el software libre o de código abierto libre ha ganado gran popularidad, tanto es así que el número de proyectos de código libre ha aumentado por lo que se puede decir, que ha alcanzado la madurez durante los últimos años Tiwarin, (2010), Singh y Singh (2015). Debido a ello, se han

desarrollado modelos de madurez de código libre, en los que las empresas se apoyan para seleccionar el software a utilizar.

El software libre o de código abierto es cada vez más utilizado por la industria de software para crear soluciones tecnológicas ya sea libre o privativa. Esa libertad que ofrece en su uso el software libre ha permitido su crecimiento y adopción tanto por pequeñas como grandes empresas, sin olvidar a los emprendedores. Eclipse.org., (2017) señala que:

> El software de código abierto se ha convertido en un proveedor dominante de tecnología para la industria del software. El software de código abierto ha demostrado ser una fuente segura y confiable de software de calidad de producción para muchas industrias. La razón por la que el software de código abierto ha sido tan exitoso es que brinda a los usuarios la flexibilidad de elegir la tecnología más adecuada para sus necesidades sin estar atados a una solución de proveedor en particular (p.11).

Vaughan-Nichols (2015) presentó un estudio donde se determinó que el 78% de las empresas utilizan software de código abierto. Aunado a lo anterior Infosys Digital (2016), señala que casi todas las empresas de Fortune 500, utilizan software de código abierto para crear aplicaciones empresariales simples o complejas.

Las organizaciones que utilizan el software libre necesitan de modelos de madurez de códigos libre a la hora de seleccionar un software para el desarrollo de sus productos, como herramienta de evaluación de aspectos de madurez, durabilidad de escalabilidad, soporte y otros factores que permitan establecer la funcionalidad, confiabilidad y desarrollo del producto a futuro. Estos modelos son metodologías para la selección del software para garantizar su calidad o soporte para el desarrollo de productos.

Por lo tanto, se hace imperativo debido al auge en el uso de software libre o de código abierto y a su presencia de uso en el desarrollo de software en la actualidad, determinar si las organizaciones de software aplican modelos de madurez de software libre, en la selección del software a la hora de ser utilizado para el desarrollo de productos.

Esta investigación, es un referente para conocer qué hacen las organizaciones panameñas en base al uso del software libre para desarrollar productos. Porque una organización que aplica modelos de madurez de software libre, establece buenas prácticas que redundarán en una mejor calidad del producto desarrollado, ya que, el uso de estos modelos permite mejorar las prácticas en el desarrollo de software apoyados en software libre.

En este sentido, nuestra investigación constituye una investigación novedosa y pionera y las interrogantes son ¿Los desarrolladores de software en Panamá, aplican modelos de madurez de software libre para desarrollar sus productos, los conocen, qué software aplican, qué aplicaciones desarrollan?

1.2. HIPÓTESIS DE TRABAJO

Los desarrolladores de software, en Panamá, utilizan modelos de madurez de software libre, para evaluar el software que van a utilizar para el desarrollo de los productos.

1.3. OBJETIVOS

1.3.1. Objetivo General

Determinar si los desarrolladores de software en Panamá, aplican los modelos de madurez de software libre o de código abierto, qué software libre utilizan para el desarrollo de sus aplicaciones y qué aplicaciones desarrollan estas organizaciones con software libre.

1.3.2. Objetivos específicos

1. Determinar si los desarrolladores de software, encuestados, aplican modelos de madurez para el desarrollo de sus productos.

2 Determinar el uso de software libre para el desarrollo de software en las organizaciones desarrolladoras de software.

3. Conocer qué aplicaciones utilizan los desarrolladores de software en las organizaciones con software libre, y qué software libre utilizan para el desarrollo de esas aplicaciones.

4. Conocer qué metodologías, métodos o modelos utilizan los desarrolladores de software, para la selección de software libre utilizan.

5. Identificar qué características de calidad son consideradas, por los desarrolladores de software, para la selección de software libre a utilizar.

6. Comprobar el uso de modelos de madurez en la selección de software libre, por parte de los desarrolladores de software en las organizaciones que laboran.

CAPÍTULO 2.

ESTADO DEL ARTE

2. ESTADO DEL ARTE

2.1. INTRODUCCIÓN

El estado del arte que aquí se presenta ha sido sometido al proceso de revisión sistemática, como manda una investigación de este nivel. La revisión sistemática es un mecanismo que permite identificar, describir, documentar, reflexionar, discernir, evaluar y concluir, sobre un tema específico. Por medio de esta revisión, se puede determinar la existencia o ausencia de estudios científicos en un área del saber. Como señala Kitchenham (2004) la ventaja de la revisión sistemática es que permite determinar si los resultados obtenidos son consistentes y pueden ser aplicados. Es un pilar imprescindible a la hora de realizar una investigación, ya que, la misma ayuda a develar el conocimiento en particular del hecho a investigar.

Por eso, en toda investigación con carácter científico esta revisión es primordial para conocer y poder validar el problema planteado. Nos permite determinar si existen estudios previos o no, qué tipo de estudios se han hecho, identificar los aspectos ausentes en la investigación y sobre todo, platearnos unos buenos objetivos basados en el problema a investigar. El problema para ser abordado debe partir de una necesidad que requiere la solución de un problema. Por este motivo para poder realizar esta investigación se realizó una revisión sistemática sobre el tema. Dicha revisión arrojó que son muy pocos los estudios encaminados al asunto de la madurez de software libre.

Con esta revisión las fuentes primarias que se seleccionaron son las bases de datos electrónicas del área de Informática, que tienen un alto impacto en cuanto a la publicación científica relacionada al área de Informática o de Ingeniería de Software como IEEE Digital Library, ACM Digital Library, Springer Link y ScienceDirect, destacándose los trabajos enumerados en la Tabla 1.

Tabla 1. Bases de datos utilizadas en la revisión sistemática realizada.

Bases de datos	Autor y año de publicación	Título
IEEE Xplorer Digital Library 4 resultados recuperados.	Etiel Petrinja; Ranga Nambakam; Alberto Sillitti 2009.	Introducing the OpenSource Maturity Model.
	Marcus Ciolkowski; Martín Soto 2008.	Towards a Process Maturity Model for Open Source Software
	Martin Soto; Marcus Ciolkowski 2009.	he QualOSS open source assessment model measuring the performance of open source communities
	Ruediger Glott; Arne-Kristian Groven; Kirsten Haaland; Anna Tannenberg, 2010.	Quality Models for Free/Libre Open Source Software Towards the “Silver Bullet”?
ACM digital library.	0 resultados	0 resultados
Science direct 3 resultados recuperados.	Umm- e-Laila, Adnan Zahoor, Khalid Mehboob, Sarfaraz Natha, 2017.	Comparison of open source maturity models
	Yoshitaka Kuwata, Kentaro Takeda, HiroshiMiura, 2010	A Study on Maturity Model of Open Source Software Community to Estimate the Quality of Products
Springer link 6 resultados recuperados.	Lee, ES.,2012.	A Study on Maturity Level for Open Source Software
	Adewumi A1, Misra , Omoregbe , Crawford , Soto, 2016.	A systematic literature review of open source software quality assessment models
	Martín Soto and Marcus Ciolkowski, 2009.	The QualOSS Process Evaluation: Initial Experiences with Assessing Open Source Processes
	Etiel PetrinjaAlberto SillittiGiancarlo Succi Etiel Petrinja Alberto Sillitti Giancarlo Succi, 2010.	Comparing OpenBRR, QSOS, and OMM Assessment Models
	Raja, Uzma; Wheeler, Brennan; and Rajagopalan, Balaji, 2009.	Open Source Software Capability Maturity Model: A Conceptual Framework"
	Etiel Petrinjaand Giancarlo Succi, 2012.	AssessingtheOpenSourceDevelopmentProcessesUsingOMM

2.2. ESTUDIOS SOBRE MODELOS DE MADUREZ DE SOFTWARE LIBRE O DE CÓDIGO ABIERTO

2.2.1 Cantidad de publicaciones

La Figura 1 muestra la evolución publicada sobre el tema de modelos de madurez de software libre o abierto a través de los años 2008-2020 Los años 2009 y 2010 son los que presentan mayor publicación sobre el tema. Lo que acentúa la necesidad de investigar sobre estos modelos de madurez de software libre o de código abierto y su aplicación.

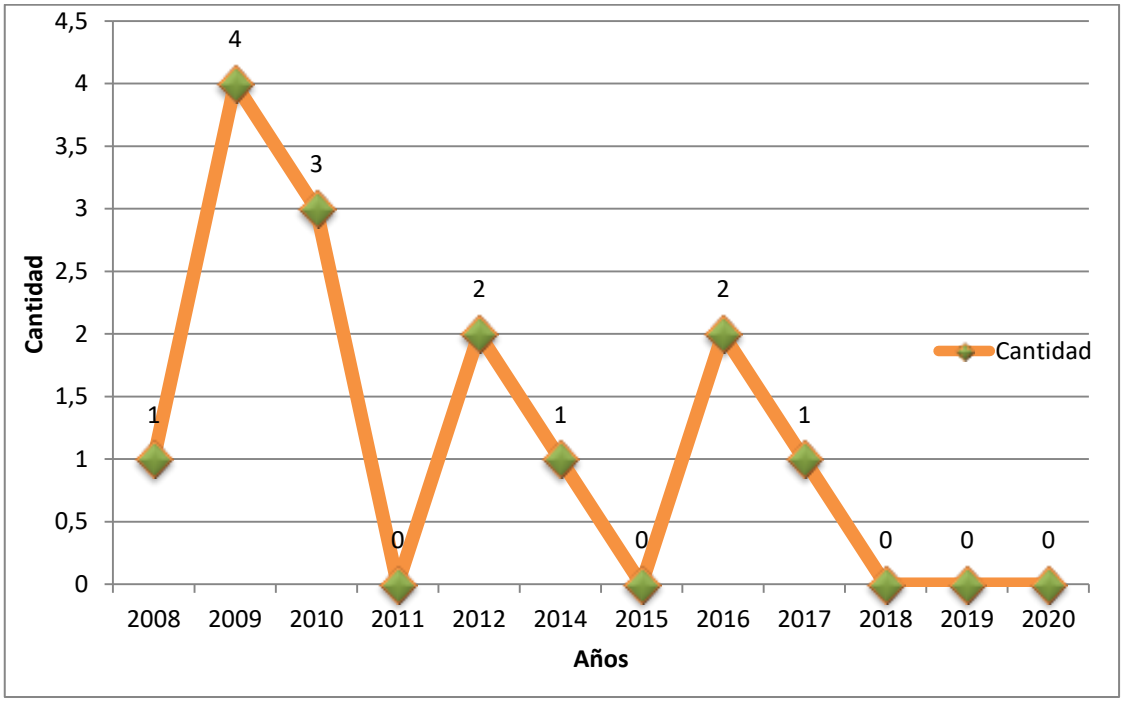

Figura 1. Publicaciones sobre los modelos de madurez de software libre o de código abierto relevantes en esta revisión.

2.2.2. Principales Estudios

A finales de la década del 2000 debido al desarrollo de software libres o de código abierto surge un modelo de madurez para proyectos de software abiertos propuesto por Ciolkowski y Soto (2008). Soto y Ciolkowski (2009) presentan una versión inicial del procedimiento de evaluación y lecciones aprendidas como resultado de su

aplicación a un grupo de proyectos de software libre o abierto. La evaluación se basa en un marco de evaluación de procesos específicamente dirigido a proyectos de software libre o abierto. (Soto y Ciolkowski, 2009) Definen un enfoque que toma en cuenta la calidad del producto, así como la madurez del proceso y la sostenibilidad de la comunidad OSS (Software Open Source) subyacente. Por su parte Díaz, *et al*. (2009) presentan una metodología para evaluación de herramientas de software libre o abierto para prueba de software, donde presenta un modelo de evaluación basado en los modelos de evaluación de madurez de software abierto, donde seleccionan los aspectos más importantes según su criterio y confecciona una serie de plantillas basada en esos criterios para la evaluación y selección de herramientas y no basándola en su madurez.

Raja *et al*. (2009) proponen un marco de modelo de madurez para el código abierto y describe las áreas claves del proceso para diferentes niveles de madurez según sea relevante para el dominio del software libre o abierto. Petrinja *et al.*(2009) presentan un modelo de madurez para software libre, que consiste en un modelo para evaluar el proceso de desarrollo de software de código abierto o libre. El modelo se basa en el conocimiento adquirido por los desarrolladores, usuarios e investigadores de código abierto o libre. Busca incrementar la calidad de los productos de software libre o abierto.

Glott *et al.*(2010) realizan una comparación de modelos de calidad de software libre o de código abierto (FLOSS), para determinar si proporcionan la "bala de plata". Para ello, evalúa un software con el modelo OpenBRR, que es un modelo de primera generación y con el QualOSS, modelo de segunda generación. Los resultados obtenidos demuestran un avance importante en el modelo de segunda generación en cuanto a la calidad, pero que probablemente la "bala de plata" todavía no se ha encontrado. Stol y Babar (2010), realizan una revisión de los métodos para evaluación de software de código abierto y comparan los diferentes métodos. Identificó 20 métodos para evaluación de software de código abierto. Además, propusieron un marco para la comparación de métodos de evaluación de software de código abierto denominado (FOCOSEM). Posteriormente se da un giro hacia la calidad, así Petrinja *et al*, (2010) presenta un estudio que consistió en investigar la calidad y la facilidad de

uso de tres modelos de evaluación de software libre o abierto: Open Business Readiness Rating (OpenBRR), Qualification and Selection of Open Source Software (QSOS) y QualiPSo OpenSource Maturity Model (OMM). El estudio identificó los aspectos positivos y negativos de cada uno de ellos. Los modelos se utilizaron para evaluar dos proyectos de Software Libre. Sin embargo, los tres modelos contienen algunas preguntas y respuestas propuestas que no son claras para los evaluadores, por lo tanto, deben reescribirse o explicarse mejor según los autores.

Lee (2012) plantea niveles de madurez al software de código abierto o libre, a través de la evaluación de la eficiencia de los niveles de madurez del software para determinar su confiabilidad. Mientras que Petrinja y Succi (2012) establecen un modelo de evaluación de mejora similar al CMMI enfocado en software libre o código abierto denominado: Modelo de Madurez para código abierto (OMM) de QualiPSo. El modelo se centra en el proceso de desarrollo lo que lo diferencia de los modelos de evaluación existentes que se centran en la evaluación del producto.

Kuwata *et al.* (2014) proponen la evaluación basado en el modelo de madurez de la comunidad de desarrollo de software abierto. (Kuwataa *et al*. (2014) Establecen una propuesta de un método de evaluación que considere la madurez de la comunidad de desarrollo de software abierto como mecanismo para evaluar el software abierto para desarrollo. Adewumi *et al.* (2016) realizan una revisión sistemática de modelos de evaluación de calidad de software libre. En esa revisión indica Adewumi et al. (2016) que el concepto de mantenibilidad es considerado esencial y primordial como característica de calidad que debe tener el producto, sin embargo, la facilidad de uso también es considerada como un atributo importante en la categoría de calidad de uso.

Laila-e *et al.* (2017) comparan diferentes modelos de madurez de software de código abierto para que sean utilizados, por los usuarios, para la selección de componentes de software de código abierto.

Como se ha demostrado en el estado de arte, los esfuerzos de los autores se centran en el desarrollo de modelos o marcos de referencias a través de la madurez del software.

Pero hay una ausencia en establecer si los desarrolladores y por ende las organizaciones de desarrollo de software, conocen estos modelos, si tienen formación en ellos y lo más importante si los aplican en la selección de software. La importancia de los modelos de madurez subyace en la calidad de software, eje principal y motivacional que debe tener toda organización que desarrolla software.

Queda demostrado con el estado del arte producto de la revisión sistemática realizada, que no se encontraron estudios, que identifiquen si las organizaciones aplican modelos de madurez enfocados al software libre o abierto, mucho menos en Panamá, por lo que constituye un campo inédito para explorar. De ahí el interés en esta línea de investigación.

Capítulo 3.

Modelos de Madurez de Software Libre

3. MODELOS DE MADUREZ DE SOFTWARE LIBRE

3.1. LA MADUREZ DEL SOFTWARE

3.1.1. Madurez

La madurez del software la define Etheredge (2009) como la eficacia alcanzada por el software en su evolución en el tiempo. Por consiguiente, definimos la madurez como la eficacia y la efectividad que logra el software, mediante la mejora continua de sus características de calidad. Chacón y Tesisi (2004) señalan que el problema de la inmadurez de software se debe a la falta de habilidad para administrar sus procesos de desarrollo de software, por parte de las organizaciones que desarrollan y administran software. Es decir, no hay madurez del software.

Mark *et al*. (1993) señalan que la madurez es un proceso específico para definir, claramente, la gestión, medición y control del crecimiento progresivo de una determinada organización. De igual forma De Freitas (2018) define la madurez como el estado de efectividad en que una organización gestiona los procesos, programas o proyectos de forma efectiva a través de su capacidad y competencia.

Por su parte el Software Engineering Institute-SEI (2010), definen un modelo de madurez y capacidad como un modelo que contiene los elementos esenciales de procesos eficaces para una o más áreas de interés y describe un camino de mejora evolutivo desde procesos inmaduros y ad hoc hasta procesos maduros y disciplinados con una mejora en la eficacia y en la calidad.

Luego de las diferentes definiciones e interpretaciones de madurez, la defino como un modelo que establece guías o buenas prácticas para una gestión certera de los procesos de software desde su fase inicial (inmadura) hasta llevarlos de forma planificada a su fase madura, a través de la medición de los procesos de forma que se realicen los ajustes necesarios para que los proyectos estén respaldado por un proceso de calidad.

3.1.2. Modelos de madurez

A este respecto ElFadl *et al.* (2018) señala que la madurez es uno de los factores clave de calidad que las personas consideran al elegir entre proyectos de código abierto. En ese sentido Taki y Elalaoui (2018) se refieren a que el uso de software de código abierto implica aspectos como; la garantía, el ciclo de vida y la sostenibilidad de la organización de desarrollo lo que conlleva a considerar esos riesgos.

Por otro lado, Soto y Ciolkowski (2009) indican que el incremento en el uso del software de código abierto ha provocado un reto para la evaluación de procesos de software, dado que, los modelos orientados a la madurez parecen muy difíciles de aplicar al desarrollo de software abierto. Sin embargo, considero que es necesario aplicar estos modelos de madurez de software para garantizar la calidad del producto. Es por ello, que es imprescindible la evaluación de los modelos de madurez de software libre.

El uso de proyectos de software de código abierto está aumentando en el entorno corporativo, creando así la necesidad de un marco de evaluación para estos proyectos Raja, (2009). Debido a las diferencias significativas en el proceso de desarrollo de código abierto del modelo de desarrollo de software tradicional, el marco del modelo de capacidad de madurez no puede aplicarse directamente al entorno de desarrollo de software abierto.

Se han desarrollado modelos de madurez enfocados al desarrollo de software abierto o libre. Los modelos de madurez de software libre se han desarrollado a partir del año 2003. La mayoría de estos modelo se han basados en modelos de calidad como el ISO 9126 o el CMMI Adewumi *et al*, (2013). A continuación se presenta una revisión de los principales modelos de madurez de software abierto o libre.

3.2. PRINCIPALES MODELOS DE MADUREZ DE SOFTWARE LIBRE O DE CÓDIGO ABIERTO

Este acápite hace referencia a los modelo de madurez de software abierto que han sido desarrollados. Algunos de estos modelos han dejado de utilizarse o no se han actualizados. Otros modelos han sido aplicados a organizaciones de software. Los modelos que se describen son aquellos que están accesibles su información.

3.2.1. Maturity model of OSS community

3.2.1.1. Origen

Modelo propuesto por el equipo de Yoshitaka Kuwata en el 2014. Es un modelo de madurez de software abierto basado en la comunidad de usuarios y toma como referencia el modelo de capacidad y madurez (CMM) Kuwata *et at.* (2014).

3.2.1. 2. Enfoque

El modelo está desarrollado tomando como enfoque un modelo de calidad en el desarrollo del software por medio de las comunidades.

3.2.1.3. Metodología

La metodología se basa en aplicar los niveles de madurez para determinar si la organización produce un software maduro y de calidad. Como todo modelo de madurez basado en el CMM debe aplicar un método para evaluar la aplicación del modelo.

3.2.1.4. Filosofía

Se basa en la calidad del software, la continuidad en el desarrollo, mantenimiento del software y la capacidad que tiene el software para su expansión. Esto último muy importante para poder fortalecer el software con la comunidad de apoyo.

3.2.1.5. Estructura

La estructura del modelo está basada en el CMM. Dispone de 5 niveles.

1. Inicial (La comunidad no se organiza o se constituye en este nivel)

2. Gestionado (Se realiza proyecto)

3. Definido (La comunidad establece los procesos de desarrollo y las reglas para administrar los proyectos y productos)

4. Cuantitativamente gestionado (La comunidad realiza medidas de forma cuantitativa para determinar el estado de los proyectos y productos)

5. Optimizado (La comunidad mide de forma cuantitativa el estado de los proyectos para establecer la mejora continua de los procesos realizados)

3.2.2 Capgemini Open Source Maturity Model

3.2.2.1. Origen

Este modelo es presentado en el año 2003, cuyo objetivo es su aplicación en asesoría a los clientes. Está basado en 7 pasos o indicadores de evaluación que determinan si el software abierto es adecuado para sus fines. Se basa en dos aspectos principales: "indicadores de producto" Laila-e *et al.* (2016), para comprobar la madurez de los productos OSS, y los "Indicadores de aplicación", para comprobar si el producto cumple con los requisitos del usuario y si se anticipa a las necesidades.

3.2.2.2. Enfoque

El enfoque del modelo “Capgemini Open Source Maturity Model” está basado en la evaluación de la madurez de un software o producto de código abierto, de forma que los requisitos del negocio estén alineados al software. Inclusive las necesidades del usuario.

3.2.2.3. Filosofía

Su filosofía consiste en fomentar la relación colaborativa entre el consultor y el cliente. Esta relación es de suma importancia para el éxito de la organización. La asesoría del cliente por parte del consultor es, en muchos casos, decisivos para optar por un producto. Contar con una buena relación consultor-cliente es positivo para mantener una empresa a flote y que los clientes confíen en ella.

3.2.2.4. Metodología

Cada categoría es analizada por lo menos por dos evaluadores que asignan un puntaje entre 1 (menos importante) y 5 (más importante) que determinarán la madurez del producto. Los indicadores se organizan en cuatros grupos. Cada grupo está formado por otros indicadores. Los grupos los constituyen los indicadores de productos, integración, uso y aceptación. Existen 12 criterios que distinguen al producto y 15 criterios formulados para los requisitos del usuario (Tabla 2). A cada indicador se le asigna un puntaje de 1 a 5.

3.2.2.5. Estructura

La Tabla 2 muestra la estructura del modelo. Se establecen los grupos y sus indicadores respectivos. Se destaca la usabilidad como el primer indicador de aplicación. Indicador éste, importante a la horade evaluar y seleccionar un software. El cliente necesita comprender de forma rápida el software, que le sea de utilidad y le evite pérdida de tiempo, que se traduce en pérdida de dinero también.

Tabla 2. Estructura del modelo Capgemini Open Source Maturity.

Grupo	Indicadores	Sub indicadores
Producto	• Producto	• Edad • Licenciamiento • Estructura de recursos humanos • Puntos de ventas • Desarrollo de la comunidad
	• Integración	• Modularidad • Colaboración con otros productos
	• Uso	• Estándares • Soportes
	• Aceptación	• Facilidad de despliegue • Comunidad de usuario • Penetración del mercado
Indicadores de aplicación	• Usabilidad • Interconexión • Rendimiento • Fiabilidad • Seguridad • Tecnología probada • Independencia del vendedor • Plataforma independiente • Soporte • Reporte • Administración • Asesoría • Formación • Programación • Implementación	

3.2.3. OSMM (Open Source Maturity Model de Navica)

3.2.3.1. Origen

El modelo tiene su origen en el año 2004. El OSMM de Navica, fue creado por Bernard Golden, CEO de Navica. Fue desarrollado para que las organizaciones evalúen productos de código abierto y determinen si el producto cumple con los requisitos establecidos por la organización Laila-e *et al*. (2016).

Este modelo evalúa la madurez de un producto en tres fases: evalúa los productos vitales, define un factor de ponderación para cada uno y calcula el puntaje general de

madurez del producto basado en seis elementos claves: la solución de software, del soporte que ofrece, la documentación, las propuesta de capacitación, la capacidad de integración del producto y qué servicio ofrece y las organizaciones asociadas.

3.2.3.2. Enfoque

Se basa en que el usuario adopte el software a través de los resultados que indicarán el grado o no de madurez del producto.

3.2.3.3. Metodología

La metodología se basa, principalmente, en cuatro fases:

1. Seleccionar el software que se va a evaluar.
2. Ponderar cada una de las características.

3. Aplicar las plantillas correspondientes a las categorías y asignarles un puntaje a cada uno de los aspectos que integran las categorías.

4. Multiplicar la puntuación de cada categoría por su ponderación para producir una puntuación final de 0 a 100.

3.2.3.4. Filosofía

El modelo de madurez de Navica es sencillo de adaptar y abarca los pasos o requisitos necesarios para definir la madurez de un producto.

3.2.3.5. Estructura

La estructura del modelo OSMM toma en cuenta elementos claves como son el producto (software), la documentación de apoyo, la formación o el entrenamiento, el servicio profesional que tenga y la integración de los productos.

Este modelo, como se observa en la Tabla 3, está organizado por seis categorías y por fases. La fase 1, que es la más compleja, se divide en cuatro aspectos que incluyen los requisitos de la organización, la identificación de los recursos disponibles, la evaluación de su madurez y la asignación de una puntuación de madurez. Por otro lado, en la fase 2 se asignan peso a cada uno de los elementos clave de acuerdo con la importancia del elemento clave.

En la fase tres se realizan los cálculos de la madurez del software. La madurez total del producto se calcula multiplicando la puntuación de cada elemento clave, con las ponderaciones y se suman todas para obtener el resultado final.

Tabla 3. Estructura del modelo OSMM.

Categorías	**Fase 1**				**Fase 2**	**Fase 3**
	Define requisitos	**Localizar recursos**	**Evaluar modelos de madurez**	**Asignar puntaje a los elementos**	**Asignar pesos a los factores**	**Calcular el puntaje de madurez del producto**
Producto software						
Soporte						
Documentación						
Formación						
Integración del producto						
Servicio profesional						

3.2.4. OSSpal

3.2.4.1. Origen

OSSpal es el sucesor del Business Readiness Rating (BRR) OSSpal (2020), que combina medidas de evaluación cuantitativas y cualitativas para software en diversas categorías. Se origina en el año 2016. Está integrado por siete categorías: funcionalidad, características operacionales del software, soporte y servicio,

documentación, atributos de la tecnología del software, comunidad y adopción y desarrollo de procesos.

Además, cada categoría está determinada por una escala de media que la pondera. Busca que las empresas, agencias gubernamentales y otras organizaciones dispongan de software libre y de código abierto de alta calidad que satisfaga sus necesidades.

3.2.4.2. Enfoque

Se enfoca en la evaluación cuantitativa y las opiniones de los usuarios.

3.2.4.3. Filosofía

Su filosofía es facilitar a las organizaciones una herramienta para la selección de software de calidad.

3.2.4.4. Metodología

La metodología está formada por cuatro fases que se describen a continuación:

Fase 1: identificar lista de componentes de software a analizar.

Fase 2: Establecer ponderaciones para las categorías y para las medidas.

Fase 3: Reunir los datos para cada medida utilizada en cada categoría y calcular su ponderación en un rango entre 1 y 5.

Fase 4: Sumar el resultado obtenido para cada categoría como en la ponderación de los factores para obtener el puntaje final.

3.2.4.5. Estructura

La estructura del modelo OSSpal **e**stá integrada por categorías, fases y una escala para cuantificar y valorar las categorías **(**Figura 2).

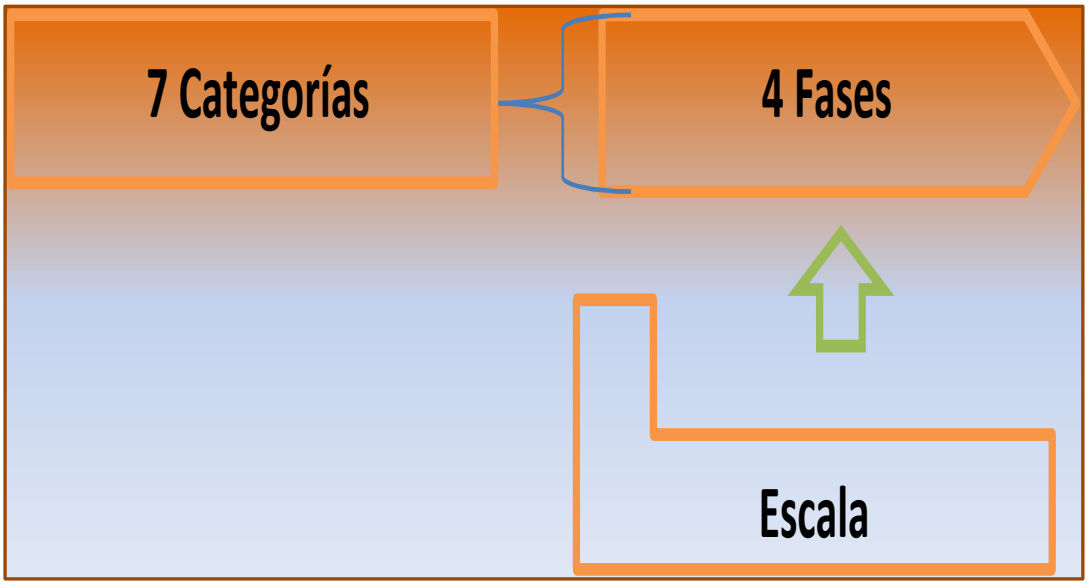

Figura 2. Estructura de OSPAL. Fuente: el autor.

3.2.5. Qualification and Selection of Open Source software methodology (QSOS)

3.2.5.1. Origen

Se origina en el año 2004, cuando se publica su primera versión 1.0, desde entonces se han realizado revisiones y publicados sus versiones. En la actualidad la última versión es la 2.0 publicada el 27/02/2013.

3.2.5.2. Enfoque

El enfoque tiene como referente la calidad del producto, la madurez del proceso y la sostenibilidad de la comunidad de software libre.

3.2.5.3. Filosofía

La metodología es abierta. Cualquier individuo puede aplicar el método para la evaluación y selección de software.

3.2.5.4. Metodología

Está integrada por cuatros pasos independientes con lo cual abarcan el proceso de evaluación y selección del software. Los 4 pasos son:

1. Definido: se clasifica el software, la licencia y la comunidad a la que pertenece.

2. Evaluación: Se evalúa el software en 3 ejes: cobertura funcional, riesgo del usuario y los riesgos de los desarrolladores.

3. Calificación: Se definen los filtros en base a los puntos que se evaluarán. Sirve para eliminar el software que no satisface las necesidades del usuario.

4. Selección: Aplicación del paso 3, con los datos dados por los primeros pasos, formulando consultas, comparaciones y la selección del producto.

A continuación se expone la estructura del modelo.

3.2.5.5. Estructura

En la Figura 3 se muestra la estructura del modelo QSOS. Estos cuatro componentes, del modelo, se interrelacionan y la evalaución del software cubre la cobertura funcional, el riesgo del usuario y los riesgos de los desarrolladores.

Figura 3. Estructura del QSOS. Fuente:Toledo, *et al.* (2013).

3.2.6. Quality of Open Source Software Endeavors (QualOSS)

3.2.6.1. Origen

La metodología nace un 01 de septiembre de 2006, y dura 36 meses para su desarrollo. Metodología desarrollada para determinar los esfuerzos del software libre y de código abierto. QualOSS está constituido por un consorcio integrado por 8 socios de 5 estados miembros de la comunidad europea: Bélgica, Alemania, Francia, España y Países Bajos. Su versión final corresponde a enero del año 2010 (Cetic, 2020). Por lo tanto, es un proyecto europeo que trabaja con los criterios de calidad que se deben considerar en el desarrollo de software libre para generar su propia metodología de desarrollo.

Está constituido o integrado por tres tipos de elementos interrelacionados: características de calidad, métricas e indicadores.

3.2.6.2. Enfoque

Su enfoque se basa en la medición de proyectos de software libre o abierto. Proporciona una herramienta de ayuda a las empresas para la integración de software libre o abierto. Se enfoca en la capacidad de evolución y la robustez.

3.2.6.3. Filosofía

Establecer un marco de referencia para evaluar la adquisición de proyectos de software libre o abierto que sean robustos y que puedan evolucionar en el tiempo.

3.2.6.4. Metodología

La metodología está diseñada para ser flexible e imponer rigidez al realizar la evaluación del software libre o abierto. Se constituye de cuatro pasos interrelacionados:

1. Definido (Define y actualiza las referencias para los otros pasos)

2. Evaluación (Evaluación del software o parte de un software)

3. Calificación (Ponderación de los criterios según el contexto)

4. Selección (Comparación y selección de software, basada en los datos de los pasos anteriores)

3.2.6.5. Estructura

La estructura del modelo QualOSS se detalla en la Figura 4. Nótese que hace mención a la comunidad de miembros para la evolución y robustez. Esta comunidad de miembros es imprescindible para el mejoramiento y actualización del software.

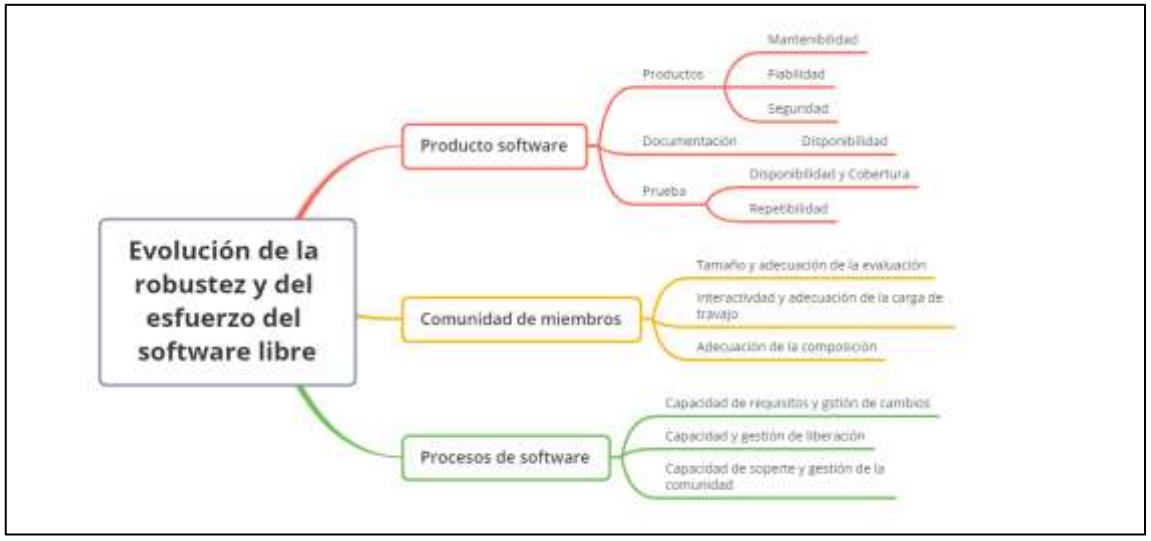

Figura 4. Estructura del modelo QualOSS. Fuente: Leister et al. (2015).

3.2.7. Business Readiness Rating Model (O-BRRM)

3.2.7.1. Origen

El modelo de calificación empresarial (O-BRRM) fue desarrollado por SpikeSource, del Centro de investigación de código abierto en Carnegie MellonWest e Intel

Corporation Wasserman, *et al.* (2006). A diferencia de otros modelos, O-BRR tiene cinco rangos desde uno (1) que representa "Inaceptable" hasta cinco (5) que representa "Excelente".

Este modelo ofrece propuestas a las organizaciones para estandarizar data de evaluación y agruparlos en diferentes categorías.

3.2.7.2. Enfoque

El modelo se enfoca en evaluar las categorías que han demostrado ser las más importantes para una implementación exitosa de software de código abierto en configuraciones específicas. Las categorías son: la funcionalidad, la calidad, el rendimiento, el soporte, el tamaño de la comunidad, la seguridad, entre otros. A su vez las categorías están formadas por subcategorías. El modelo es abierto y flexible, pero normalizado.

3.2.7.3. Filosofía

La filosofía de este modelo, se fundamenta en un modelo confiable y abierto para la evaluación, que permite el intercambio de información entre los responsable de TI (Tecnología de la Información).

El intercambio de información es necesario para robustecer el modelo. El modelo es abierto y personalizable dado que permite que se pueda establecer criterios propios de evaluación. Wasserman (2005).

3.2.7.4. Metodología

La metodología es sencilla. Se evalúa el software. El modelo se fundamenta en 4 pasos que se listan a continuación:

1. Se crea una lista con el software a evaluar.

2. Se clasifica y se ponderan los criterios de selección.

3. Se recopilan datos para cada criterio seleccionado

4. Se calculan y se publican los resultados.

3.2.7.5. Estructura

La estructura del modelo Business Readiness Rating Model **(**BRRM) se resume en la Figura 5. Se puede observar en esta figura, que el modelo está compuesto por siete categorías diferentes, de las cuales tres categorías: software operacional, soporte y tecnología de software, están subdivididas en subcategorías. Dos de las categorías tienen una sola subcategoría y dos no tienen subdivisión.

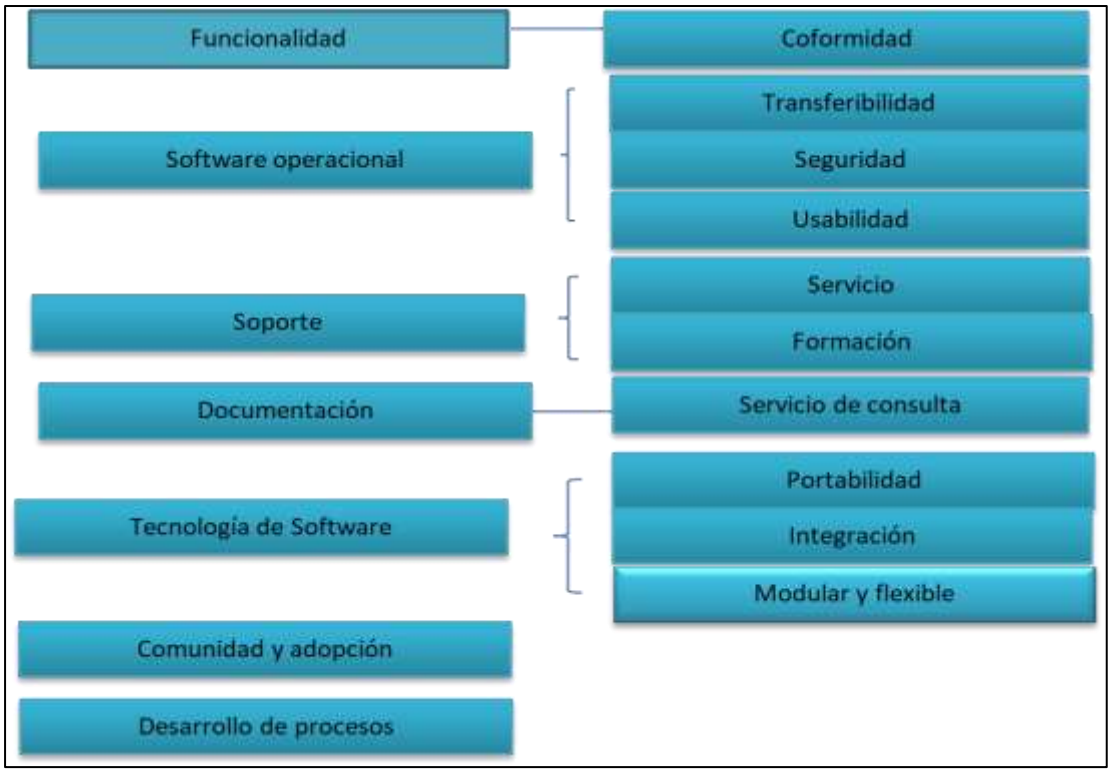

Figura 5. Estructura del Modelo O-BRRM. Fuente: Toledo, *et al*. (2013).

3.2.8. Observatorio de calidad de software para software abierto (Software Quality Observatory forOpen software - SQO-OSS)

3.2.8.1. Origen

Este modelo se origina en el año 2008. Establece un marco para la evaluación para el monitoreo continuo de la calidad del software. Por lo tanto, su principal objetivo son las características de calidad como la confiabilidad y la seguridad. Ambos, elementos claves del software.

3.2.8.2. Enfoque

Se basa en un enfoque holístico para la evaluación de software, inicialmente dirigido al software de código abierto AUEB, (2008). SQO-OSS utiliza diversas fuentes de indicadores de calidad para generar un grupo de métricas que se apliquen de forma automática.

Este enfoque considera las características de calidad como son, la mantenibilidad, confiabilidad y seguridad de los proyectos, código fuente y la comunidad (Samoladas, *et al*. 2008).

3.2.8.3. Filosofía

Parte del hecho que la calidad del software libre está directamente relacionado a dos factores críticos: el código y la comunidad de usuarios.

3.2.8.4. Metodología

La metodología está organizada en dos fases:

Fase 1: Definición del modelo de evaluación

1. Definición de los criterios del modelo (atributos y sub atributos).
2. Definición de métricas.

Fase 2: Definición del método de agregación.

1. Definición de las categorías de evaluación.
2. Definición de perfiles de esas categorías.

3.2.8.5. Estructura

La estructura de ete modleo está focalizada en dos aspectos, la calida ddel producto y la calidad de la comunidad. En la Figura 6 se observa la estructura del modelo Software Quality Observatory for Open software - SQO-OSS. Como se observa en esta figura, en la calidad del producto se contempla la madurez.

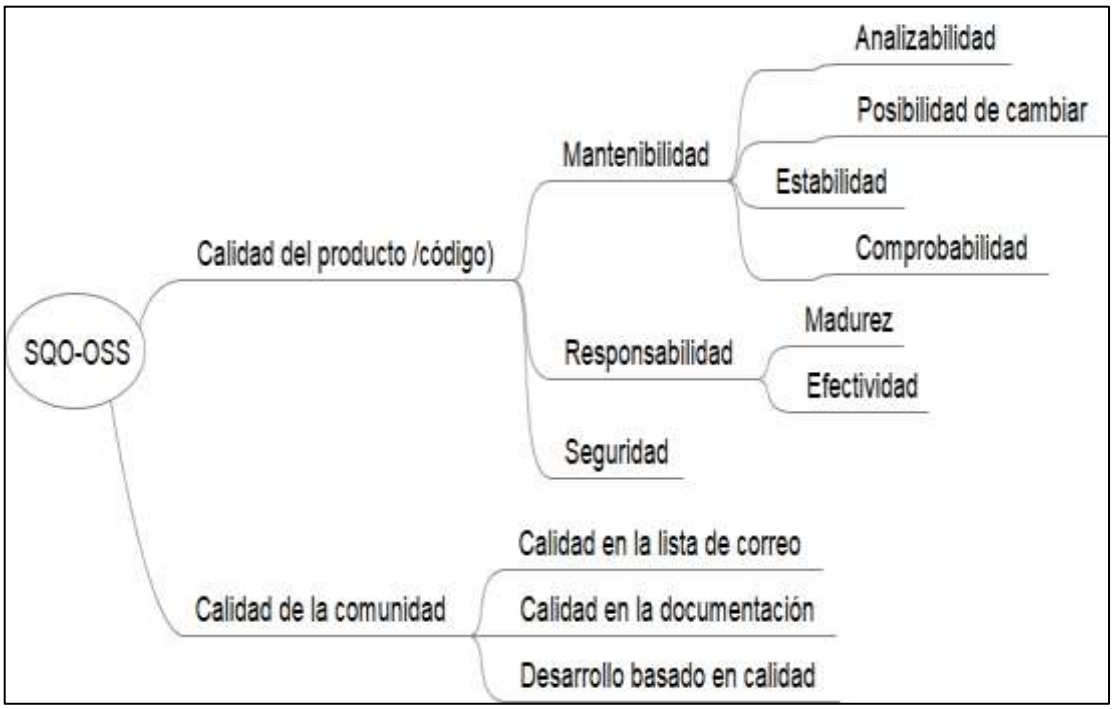

Figura 6. Estructura del modelo SQO-OSS. Fuente: AUEB, (2008).

3.2.9. QualiPSo Open Maturity Model (OMM)

3.2.9.1. Origen

Publicado en el año 2008, es un modelo de proceso para el desarrollo de software. Está organizado como un modelo evolutivo, inspirado en CMMI (capability Maaturity

Model Integration), pero enfocado en las características de desarrollo de software libre o abierto. OMM también se organiza en niveles, cada nivel se basa e incluye elementos confiables en el nivel inferior (QualiPSo, 2020). En general, es un modelo similar a CMM (Sillitti, (2020).

3.2.9.2. Enfoque

El modelo QualiPSo se desarrolló con la intención de ser una herramienta de evaluación de la calidad de los proyectos de software libre, para uso de la comunidad de desarrollo de software, integradores y usuarios finales.

3.2.9.3. Filosofía

Considera que las prácticas pueden ayudar a identificar las características relevantes de un proyecto de software libre o abierto que conduce a un producto confiable.

3.2.9.4. Metodología

El modelo OMM, se basa en el modelo CMMI, por lo tanto, se organiza en niveles de madurez del proceso. Los niveles de madurez incluyen lo que denominan: elementos de confianza específicos (TWEs). Los niveles de madurez están divididos en: Básico, Intermedio o Avanzado. Solo cuando se logra completar todos los niveles se logra la madurez de un nivel. Para subir de un nivel de madurez al otro, se logra al cumplir las prácticas obligatorias del nivel anterior.

3.2.9.5. Estructura

La estructura del modelo QualiPSo se puede observar en la Figura 7. Como se ve está formado por tres niveles: básico, intermedio y avanzado.

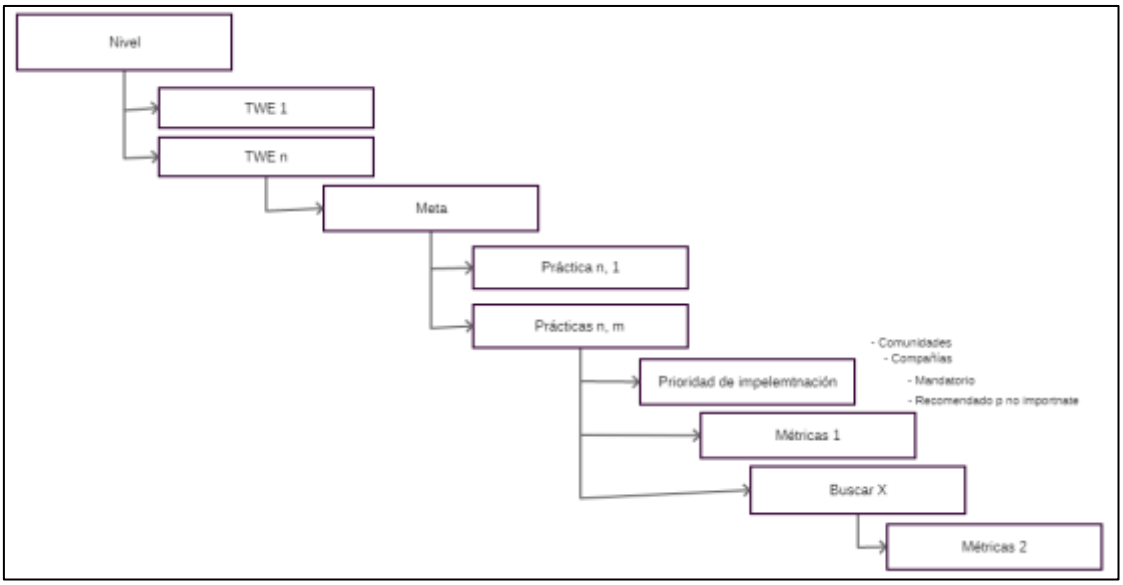

Figura 7. Estructura del modelo QualiPSo (OMM**)**. Fuente: QualiPso, (2020).

3.2.10. Modelo E-OSS (easiest open source)

3.2.10.1. Origen

Desarrollado por SIAD-Laboratory en el año 2015 Laila-e, *et al*. (2016). Es un modelo de reciente creación donde se toma en cuenta la madurez del software.

3.2.10.2. Enfoque

Un modelo relativamente joven, cuyo enfoque está basado en la interoperabilidad del software (Laila-e, *et al*., 2016). Es decir, el intercambio de datos que se da por los distintos programas a través de un formato de intercambio que permite usar los mismos protocolos.

3.2.10.3. Filosofía

Su filosofía se basa en la selección de software para reúso, tomando como base la madurez del software, como ya se señaló, para interoperar como empresas. Interoperar es compartir e integrar con otras empresas, lo que permite el intercambio

de ideas, de información de productos, de modelos y muchas otras utilidades. Se destaca en este modelo la importancia de un software maduro.

3.2.10.4. Metodología

Este modelo E-OSS (easiest open source) está fundamentado en una estructura de cuatro fases: definición, identificación, calificación y selección Belaissaoui y Ait Houaichy (2015).

- La fase 1 se refiere a la definición: se aplica para verificar y analizar los componentes en sus aspectos funcionales, técnicos y estratégicos.

- La fase 2 implica la identificación: se utiliza para generar una hoja de datos de elementos claves basada en la característica general del componente que se describe en cuatro grupos.
- La La fase 3 de calificación: se utiliza para asignar una calificación a los diferentes criterios para obtener una puntuación general para seleecionar los componentes más acordes.

- La fase 4 se basa en la selección. se utiliza para determinar la suma total de la puntuación de todos los criterios, con la finalidad de identificar las diferentes soluciones maduras. Es decir, son soluciones que han sido recomendadas por el largo periodo que han recorrido, han sido probadas y las organizaciones le tienen fe o están satisfechas con ellas.

El modelo E-OSS **(**easiest open source**)** está fundamentado en 4 pilares (Figura 8) que conforman las cuatro fases, antes descritas: definición, identificación y calificación.

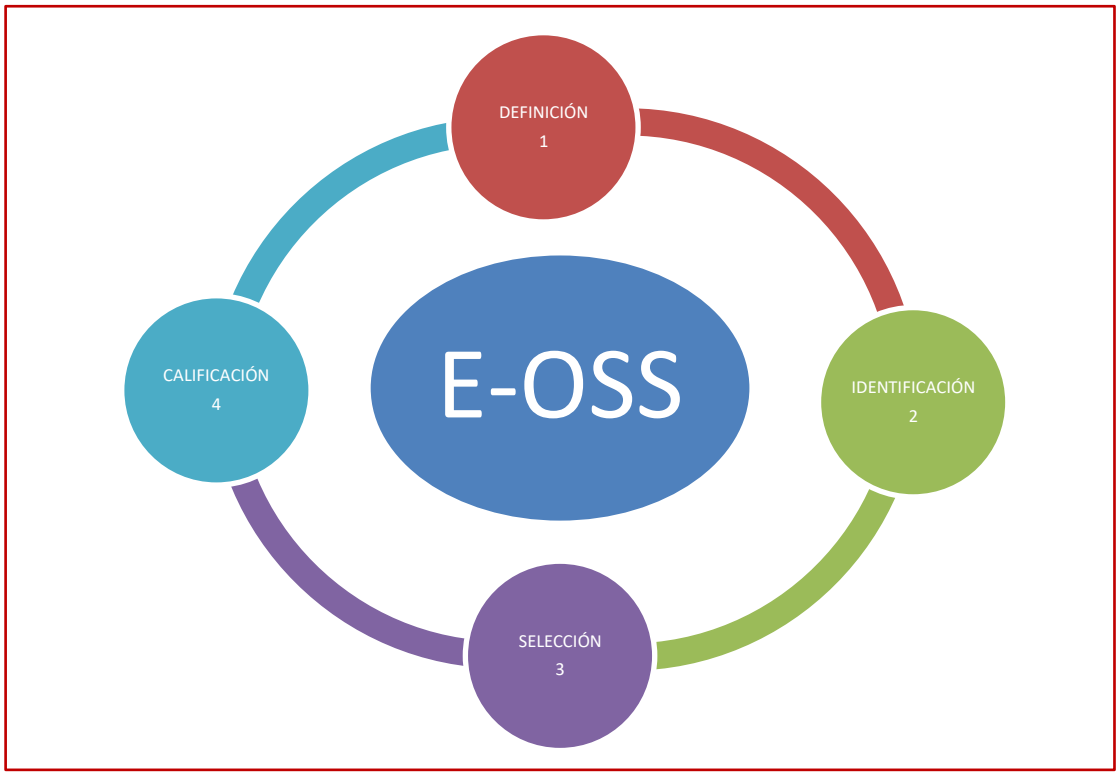

Figura 8. Estructura del modelo E-OSS. Fuente: Belaissaoui y Ait Houaichy, (2015).

CAPÍTULO 4.

GLOSARIO DE TÉRMINOS

GLOSARIO DE TÉRMINOS

1. **Aplicación Web**: Tipo de software que se codifica en un lenguaje que pueda ser soportado y ejecutado por los navegadores de Internet o por una intranet o red local.

2. **Desarrollador de software**: Un desarrollador es (al que con frecuencia también se conoce como analista-programador), es un especialista en informática que es capaz de concebir y elaborar sistemas informáticos (paquetes de software), así como de implementarlos y ponerlos a punto, utilizando uno o varios lenguajes de programación.

3. **Estado del arte**: proviene originalmente del campo de la investigación técnica, científica e industrial y significa, en pocas palabras, la situación de una determinada tecnología.

4. **IEEE:** Instituto de Ingenieros Eléctricos y Electrónicos.

5. **Ingeniería de software**: Disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de los programas informáticos **(**software**).**

6. **Madurez**: Estado de efectividad en que una organización gestiona los procesos, programas o proyectos de forma efectiva a través de su capacidad y competencia.

7. **Metodología**: Conjunto de métodos que se siguen en una investigación científica, un estudio o una exposición doctrinal.

8. **Modelo de madurez y capacidad;** Como un modelo que contiene los elementos esenciales de procesos eficaces para una o más áreas de interés y describe un camino de mejora evolutivo desde procesos inmaduros y ad hoc hasta procesos maduros y disciplinados con una mejora en la eficacia y en la calidad.

9. **Probema:** Cuestión que se plantea para hallar un dato desconocido a partir de otros datos conocidos, o para determinar el método que hay que seguir para obtener un resultado dado.

10. **Revisión sistemática**: Resúmenes claros y estructurados de la información disponible orientada a responder una pregunta clínica específica.

11. **Software libre**: Es el software que respeta la libertad de los usuarios y la comunidad. A grandes rasgos, significa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software

12. **Software privativo**: Tipo de programas informáticas o aplicaciones en el que el usuario no puede acceder al código fuente o tiene un acceso restringido y, por tanto, se ve limitado en sus posibilidades de uso, modificación y redistribución

13. **Software**: Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.

14. **Usabilidad**: Medida en la cual un producto puede ser usado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso especificado.

Capítulo 5.

Metodología

5. METODOLOGÍA

5.1. METODOLOGÍA IMPLEMENTADA

Para llevar a cabo la ejecución de esta investigación novedosa y pionera en nuestro país, se realizó en primera instancia, la planificación de la misma, que incluyó la revisión sistemática sobre el tema objeto de estudio, que conforma el estado del arte, ya presentado.

A continuación se presenta la metodología aplicada. La planificación del trabajo además de la revisión sistemática, incluyó dos fases: exploratoria y de ejecución.

5.1.1. Fase de exploración:

Se realizaron las siguientes actividades en esta fase:

1. Elaboración de los borradores (encuesta previa) en base a los indicadores que aplican los modelos de madurez y de otros elementos que se consideran necesarios indagar.

2. Validación del cuestionario que sirvió para detectar la comprensión del instrumento y/o la mejora del mismo. También nos permitió obtener una idea de los resultados esperados.

3. Búsqueda y localización de las instituciones, organizaciones o empresas desarrolladoras de software y desarrolladores de software independientes.

4. Selección de la muestra para el estudio de la población localizada.

Para la elaboración de la encuesta se utilizaron las recomendaciones de Cardona (2002), Bernardo y Calderero (2000), León y Montero (1997). El instrumento

utilizado para la recogida de los datos fue un cuestionario (Anexo 1) a través de la aplicación Monkey Suvrvey.

5.1.2. Fase de ejecución:

Una vez afinada la encuesta como resultado de la fase exploratoria, se procedió a aplicar la encuesta a los desarrolladores de software que se habían podido contactar. Esta aplicación de la encuesta se realizó vía Web, por medio del correo electrónico o Redes sociales (WhatsApp), anexando el enlace de la misma.

En síntesis, la metodología que se siguió abarca los siguientes aspectos claves (Figura 9).

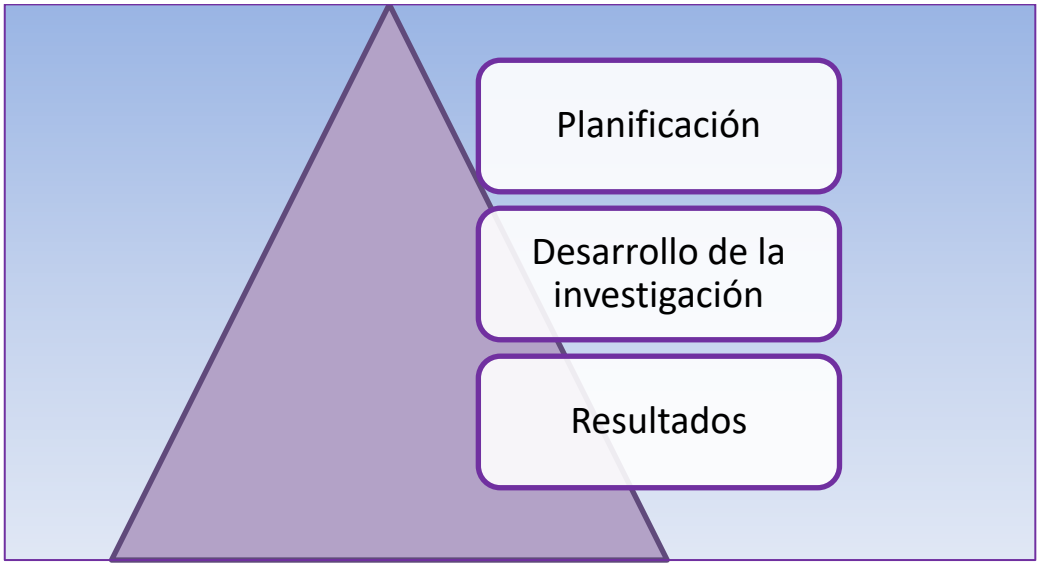

Figura 9. Aspectos básicos de la metodología de la investigación.

5.1.3. Operacionalización de variables e indicadores

A continuación se presenta la operacionalización de las variables del estudio y sus inidcadores (Tabla 4). Operacionalizar una variable es tener claro cómo se va a observar y a medir cada característica del estudio, por esto es importante establecer la relación variables e indicadores.

Para este estudio la variable dependiente sería el desarrollador del software y las variables independientes serían los modelos de madurez de software libre.

Tabla 4. Operacionalización de variables e indicadores.

Variable	Definición conceptual	Dimensiones	Indicadores	Técnicas o intrumentos
Modelos de madurez de software libre o de código abierto	Capacidad para determinar la calidad de un software libre.	Calidad de software	Atributos de los modelos de madurez de software libre	Encuesta
Software libre que utilizan los desarrolladores para el desarrollo de aplicaciones	Software o leguaje de programación para el desarrollo de software	Programación	Tipo de software	Encuesta
Aplicaciones que desarrollan las organizaciones de desarrollo de software con software libre	Desarrollo de aplicaciones o software	Programación	Tipo de aplicación	Encuesta

5.2. FASE DE PLANIFICACIÓN, ORGANIZACIÓN, ANÁLISIS DEL ESTUDIO

5.2.1. Planificación

Se decantó, una vez realizado el análisis previo para seleccionar la técnica más adecuada a utilizar en el estudio, por la encuesta (investigación por encuesta), por tratarse de un estudio sin antecedentes conocidos en nuestro país. Una encuesta es el conjunto de pasos que hay que seguir para llegar a una estimación de algún parámetro Abad y Servín, (2004), la misma se desarrolla para cumplir con los objetivo que se quieren medir en un estudio. Está reforzado por lo expuesto por Cardona (2002): la investigación por encuesta resulta especialmente valiosa cuando se investiga un área por primera vez, se guía por los pasos del método científico. Para Buendía, Colas y Hernández (1998) la encuesta como método de investigación, es capaz de dar respuesta a problemas tanto en términos descriptivos como de relación de variables, tras la recogida de información sistemática, según un diseño previamente establecido que asegure el rigor de la información obtenida.

Por las razones antes expuestas, en este estudio la encuesta fue el instrumento de evaluación que aplicaba, dado el objetivo general planteado en esta investigación y porque se trata de un estudio que por primera vez se realiza en Panamá. Además permitir recoger información valiosa y con los datos obtenidos abrir camino a nuevas investigaciones.

5.2.2. Pasos en la planificación y organización de la investigación

5.2.2.1. Evaluación del contexto

Como en toda investigación, lo primero que se realizó fue la evaluación del contexto panameño, con respecto a los modelos de madurez de software libre, que involucró el planteamiento del problema y los objetivos trazados en el estudio, las condiciones, la factibilidad de realizar la investigación en Panamá, los interrogantes por resolver. La búsqueda de la población adecuada para obtener datos válidos a partir de la muestra que se seleccionara.

5.2.2.2. Evaluación de insumos y del proceso

Otros aspectos importantes en todo estudio científico. Se analizaron los recursos necesarios para llevar a cabo el estudio: humano, económico, disposición de bibliografía, el apoyo de los futuros encuestados, el tiempo requerido para elaborar y aplicar el cuestionario. Toda la logística que conlleva planificar la investigación, tipo de instrumento, elaboración del instrumento de evaluación y ejecutar su aplicación, contactar los desarrolladores de software para colocar la encuesta, el envío por correo electrónico o redes sociales, el tiempo de espera para obtener las encuestas cumplimentadas por parte de los encuestados.

La evaluación del proceso consistió en considerar toda la implementación de los procedimientos para alcanzar los objetivos propuestos: Elaboración del cuestionario previo, prueba piloto, validación, reestructuración, cuestionario definitivo y su aplicación. Los correos de “recorderis” para que, por favor, respondieran la encuesta.

Esto último, trascendental para la recogida de la data significativa para los resultados. Por último, la organización de los resultados y su evaluación.

5.2.3. Organización

5.2.3.1. Validación y confiabilidad

Para validar la encuesta se realizó una pre-prueba piloto y una prueba piloto: Una vez elaborado el cuestionario, se le dio a leer a algunos compañeros del área y a una doctora en Didáctica para conocer el grado de comprensión de los ítems (pre-prueba piloto). Con sus comentarios y observaciones se editó el instrumento. Con el instrumento editado se procedió a aplicárselo a los compañeros expertos en la materia (prueba piloto) que se prestaron de voluntarios. Con sus observaciones se estructuró el cuestionario definitivo que fue aplicado a los desarrolladores localizados o participantes del estudio.

La validación fue significativa para modificar algunas preguntas, adecuarlas a los objetivos trazados y ajustar la cantidad de ítems para no hacerla muy larga. Sobre todo, nos permitió darnos cuenta que no conocían sobre modelos de madurez para software libre. Al parecer los voluntarios que participaron, estaban más relacionados con el tema de modelos de madurez para procesos.

El anterior procedimiento obedece al planteamiento de muchos expertos que consideran que se puede asegurar la validez cualitativa a través de juicios o revisión de expertos, más allá de lo que expresan los números. Planteamiento que fue el utilizado en este estudio. Por lo tanto, el instrumento aplicado o de ejecución fue validado a través de expertos, como ya se expuso, razón por la cual esos voluntarios no fueron incluidos en la muestra para evitar sesgo.

Con respecto a la confiabilidad, se obtuvo de la revisión de los expertos que se realizó y de sus observaciones, lo que permitió modificar y ajustar el instrumento o cuestionario de la encuesta aplicada. En síntesis, la confiabilidad de la encuesta aplicada, se obtuvo mediante la aplicación de la prueba piloto.

5.2.4. Análisis de la data

Para el análisis de la data obtenida y que la misma fuese susceptible de ser estudiada, se procedió a su tabulación. Esto facilitó la visualización e interpretación de los resultados. Se organizaron los datos en tablas de valores absolutos, de distribución de frecuencias, de porcentajes y representaciones gráficas.

Para resumir todo lo anterior, la obtención de los resultados conlleva la organización global en un proceso sistemático que se condensa en la Figura 10.

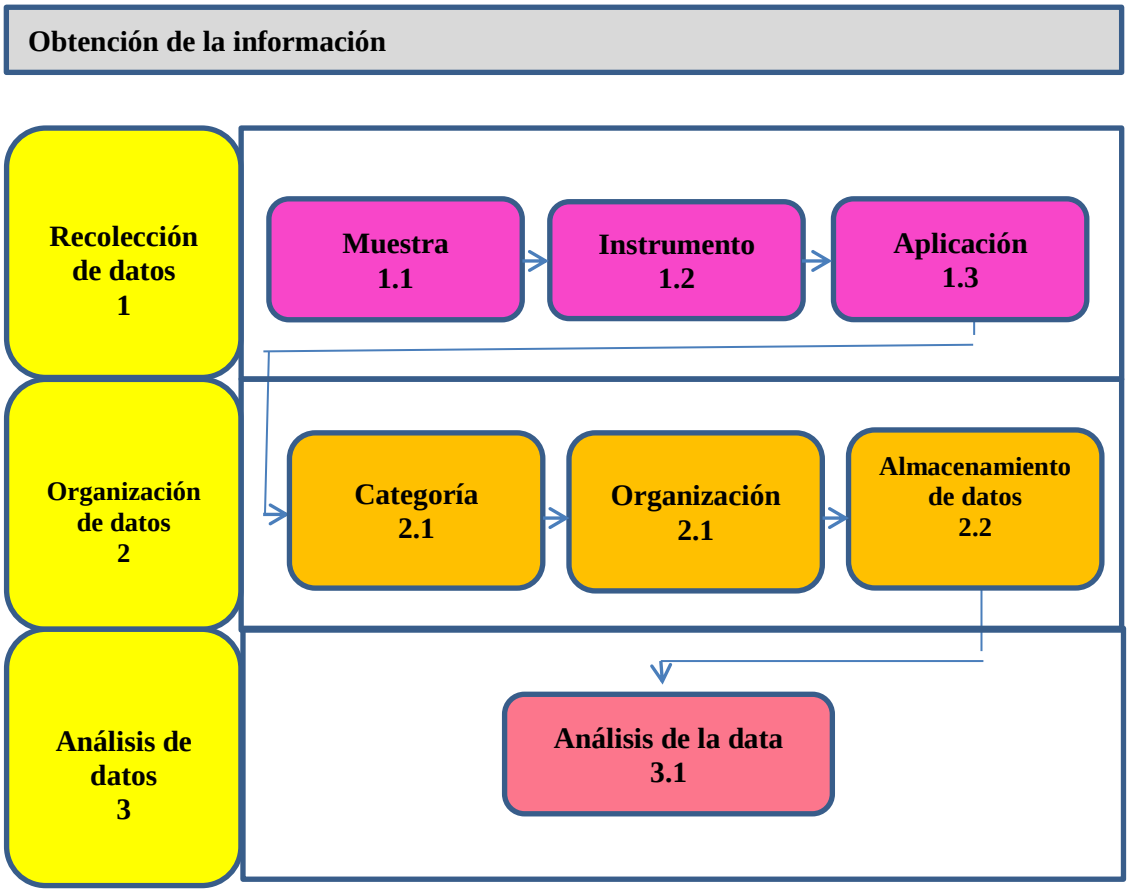

Figura 10. Pasos para obtener y evaluar los resultados en esta investigación.

5.3. MODELOS DE MADUREZ

La selección de los modelos de madurez incluyó una revisión sistemática, como ya se expuso, de los mismos y su estudio y análisis para determinar cuáles podrían ser objeto de estudio. Una vez analizados se procedió a seleccionar los indicadores que

identifiqué, que tenían un punto común en los diferentes modelos y que podrían ser utilizados a la hora de seleccionar un software libre, por parte de los desarrolladores en base a nuestra experiencia.

5.4. DISEÑO E IMPLEMENTACIÓN DEL CUESTIONARIO

5.4.1. Diseño

El estudio realizado consistió en una investigación descriptiva (investigación descriptiva por encuestas) de diseño no-experimental. La investigación desarrollada fue diseñada para recoger la información a través de una encuesta. La data recopilada desde el punto de vista de la estadística es del tipo externa, son datos obtenidos por el autor a través de una encuesta a los desarrolladores de software durante el año 2018-2019. Como toda data que se procesa, cumplió con los procesos establecidos de organización, representación gráfica y el análisis correspondiente.

5.4.2. El cuestionario

Las preguntas del cuestionario (19), que fue aplicado fueron elaboradas para determinar el uso de modelos de madurez para software libre. Eso incluye indicadores orientados a la calidad del software y a las características propias de software libre, que fueron consideradas en base a los diferentes modelos de madurez de software libre revisados en la literatura. Por lo tanto, los indicadores se adaptaron al contexto de esta investigación.

5.4.2.1. Estructura

El cuestionario quedó estructurado en dos apartados que contemplan lo siguiente: **I. Datos Generales,** compuesto por 9 preguntas y el **II. Indicadores de madurez**, estructurado en 10 preguntas. A continuación, se exponen cada una de las partes en que se estructuró el cuestionario.

I. Datos Generales

Los datos generales facilitaron conocer y caracterizar la muestra en estudio como lo**s** años o el tiempo de funcionamiento de la organización, qué tipo de software desarrolla, para qué mercado se desarrolla el software, cuántos años tiene el equipo de desarrollo, qué tipo de formación tienen los desarrolladores**.** La información recopilada permitió categorizar las variables. El cuestionario aplicado se expone a continuación, además se anexa al documento en la sección de Anexo:

I. Datos Generales

1. ¿Cuántos años o tiempo de funcionamiento tiene la organización de desarrollo de software? _____.

2. ¿Qué tipo de software desarrolla: escritorio, aplicaciones Web, páginas Web, móvil, otros? _____.

3. Desarrolla software para el mercado:

Nacional □ Internacional □ Ambos mercados □

4. ¿Cuántos años tiene formado el equipo de desarrollo?_____.

5. ¿Qué formación académica tiene el equipo de desarrollo?

Técnico □

Licenciatura □

Maestría □

Doctorado □

6. ¿Los desarrolladores utilizan software libre para el desarrollo de aplicaciones?

Si □ No □

7. ¿El equipo de desarrollo participa de algún proyecto de software libre?

Si □ No □

8 ¿Qué tipo de software desarrolla: escritorio, aplicaciones web, páginas web, móvil, otros?

_____

9. ¿Mencione que software libre utilizan para desarrollar las aplicaciones?

_____.

El segundo apartado del cuestionario es específico para los niveles de madurez y quedó estructurado así:

II. Nivel de indicadores de madurez.

10. ¿El equipo de desarrollo tiene conocimientos en modelos de madurez de software?

Si □ ¿Por qué si? _____.

No □ ¿Por qué no?_____.

11. ¿La organización implementa modelos de madurez en el desarrollo de software?

Si □ No □

12. ¿Realizan una evaluación del software libre que utilizan para el desarrollo de los productos?

Si □ No □

13. Pregunta 13: La evaluación del software libre lo realiza mediante:

Una metodología □

Una herramienta □

Un modelo de Madurez □

No se realiza evaluación del software □

14. ¿Por qué utiliza software libre para el desarrollo de productos?

_____.

15. ¿Qué indicadores o características importantes considera en la selección de software libre? _____.

16. ¿Para la selección del software libre es importante su madurez?

Si □ ¿Por qué si? _____.

No □ ¿Por qué no?_____.

17. ¿Conoce o a escuchado de algún modelo de madurez para software libre?

Si □ No □

18. ¿De los siguientes indicadores o características de madurez de software**,** cuáles considera para la selección de software libre?

Licencia	□
Comunidad de usuario	□
Soporte detención de errores	□
Gestión del proyecto	□
Aseguramiento de la calidad	□
Modificación de código fuente	□
Rendimiento	□
Usabilidad	□
Escalabilidad	□
Funcionalidad	□
Modo de distribución	□
Versión	□
Todas la anteriores	□

19 ¿Aplica un modelo de madurez de software libre para la selección del software?

Si □ ¿Por qué si? _____.

No □ ¿Por qué no?_____.

5.5. POBLACIÓN Y MUESTRA

La población en nuestro contexto se refiere a todos los desarrolladores de software, que su área laboral se encuentre en la ciudad de Panamá, específicamente en el centro de la ciudad. Para tales fines se contactó a los colegas, a los egresados de la Facultad de Informática, Electrónica y Comunicación de la Universidad de Panamá y de la

Universidad Tecnológica de Panamá. Además, se les solicitó a los participantes ser un medio multiplicador de la encuesta entre sus colegas y conocidos que se dedican al desarrollo de software. Así la población en esta investigación fue de 50 desarrolladores de software (N=50).

Aunque se envió la encuesta a todos los desarrolladores localizados (50) al final se obtuvieron 36 encuestas cumplimentadas lo que pasó a ser la muestra (n=36) del estudio. Esta muestra representa el 72% de la población localizada, que estadísticamente es considerada una muestra significativa. Lo que significa que la muestra en estudio es superior a la recomendada, lo que robustece el estudio realizado.

5.6. RESTRICCIONES DEL ESTUDIO

El estudio estuvo dirigido a desarrolladores de software, ya fuesen de empresas, organizaciones o desarrolladores independientes. De esta forma se cumplía con otro precepto de la estadística que es la homogenización de la muestra, lo que disminuye la probabilidad de sesgo porque todos los participantes compartían el común denominador, de ser desarrolladores de software.

Como en todo estudio que se realiza la confiabilidad de los datos es crucial. Por ello la restricción en esta investigación, fue no considerar a aquellas personas que no trabajaran como desarrolladores de software. De esta forma se trabajaba con una muestra homogénea, lo que permitió comparar los resultados. No hubo restricción por sexo, creencia religiosa, rasgos físicos, vestido, cultura, color, nacionalidad, afinidad política o sexual y de ningún otro tipo.

5.7 COMPILACIÓN DE LA DATA Y PROCESAMIENTO INFORMÁTICO DE LA INFORMACIÓN

Una vez recibidas y reunidas todas las encuestas rellenadas, se procedió a levantar los datos en una base de datos en hojas de Excel, con el programa Microsoft Excel versión 2010. Los datos se organizaron en las diferentes hojas de Excel, acorde con

las variables en estudio. A cada variable se le confeccionó una página de Excel, lo que permitió analizar la data, por variable, y poder comparar entre variables.

5.8. ANÁLISIS DE LA DATA E INTERPRETACIÓN DE LOS RESULTADOS

5.8.1. Análisis e interpretación de los resultados

Los análisis estadísticos que se hicieron, como ya se había mencionado, corresponden a estadísticos descriptivos univariantes. La organización básica y tabulación de los datos para su interpretación consistió en la elaboración de tablas de valores absolutos, de porcentajes y representaciones gráficas.

De nada vale realizar un estudio si no se recaban los resultados obtenidos y se leen estos resultados, es decir, la interpretación de los resultados, qué nos dicen los hallazgos encontrados, qué nuevos conocimientos nos aportan o qué nuevas interrogantes nos dejan. A mi consideración, la parte medular o más importante en un estudio. Por lo tanto, con la data que obtuvimos, debidamente organizada, se inició la etapa de analizar los resultados hallados, interpretar lo que los participantes manifestaron, qué significan los resultados encontrados. Así se realizó una descripción detallada del contenido del cuestionario para determinar la existencia o no, de tendencias en las respuestas obtenidas. Se estudiaron cada una de los porcentajes de cada ítem de toda la encuesta.

Los resultados obtenidos se disponen en apartados acorde con la estructura de la encuesta aplicada.

La Figura 11 resume el planteamiento general, en el diseño de la investigación. Se exponen los principales componentes de la fase procedimental realizados.

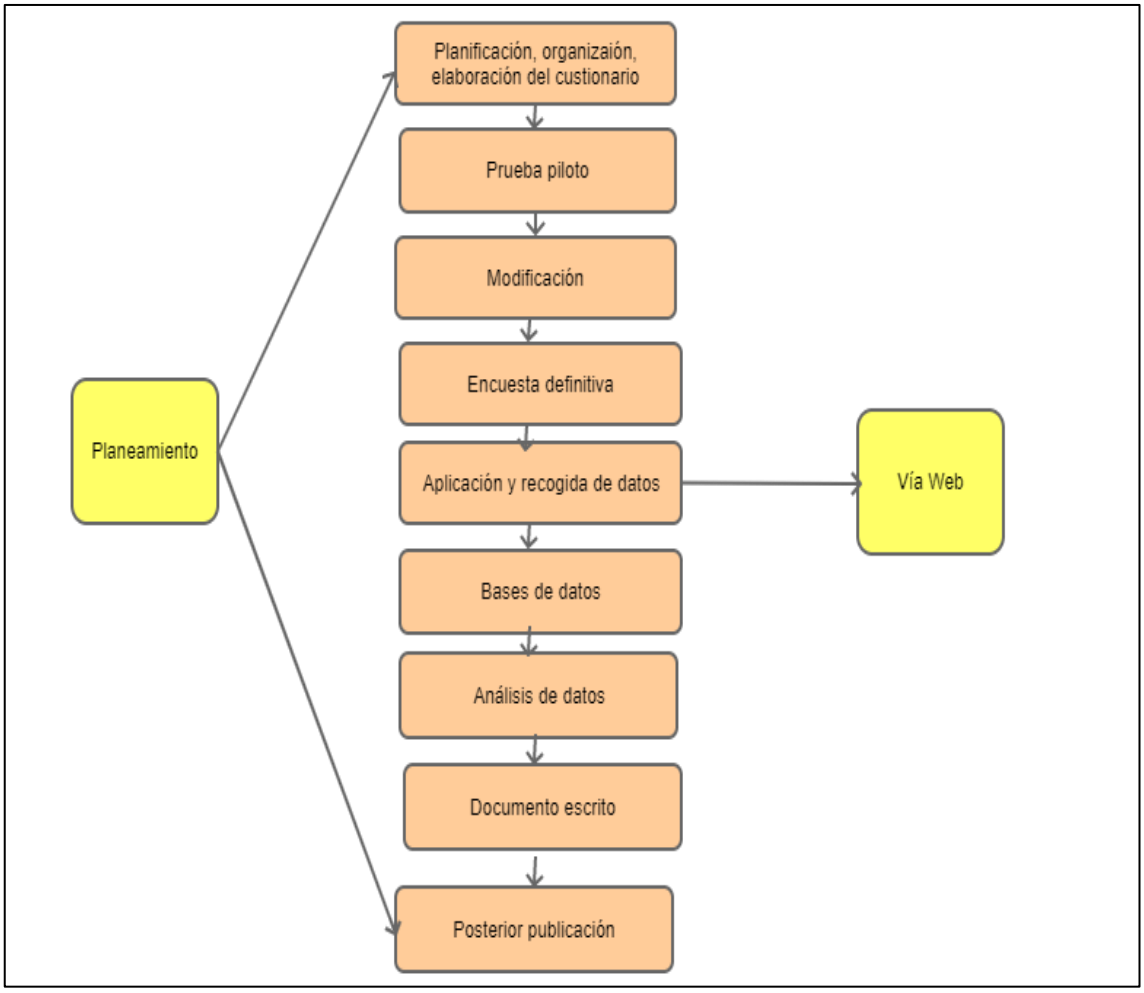

Figura 11. Diseño del plan de trabajo para la realización de la investigación.

A continuación se analizan los resultados obtenidos en la presente investigación, por apartado, como está estructurado en la encuesta y para mayor comprensión de los mismos.

Capítulo 6.

Resultados y Discusión

6. RESULTADOS Y DISCUSIÓN

Se presentan los resultados obtenidos en esta investigación y su discusión, por apartados, tal como quedó conformada la encuesta.

6.1. RESULTADOS DEL APARTADO I: DATOS GENERALES

Pregunta 1: ¿Cuántos años o tiempo de funcionamiento tiene la organización de desarrollo de software?

En la Figura 12 se muestran las respuestas obtenidas a esta pregunta. Se destaca de las mismas, que las organizaciones se concentran en mayor cantidad entre los 3 (6 organizaciones) y 4 años (5 organizaciones). No obstante, las organizaciones que más años tienen en funcionamiento se agrupan entre los 10 y 20 años: 3 organizaciones con 20 años, 1 con 18 años y 2 con 10 y 15 años respectivamente.

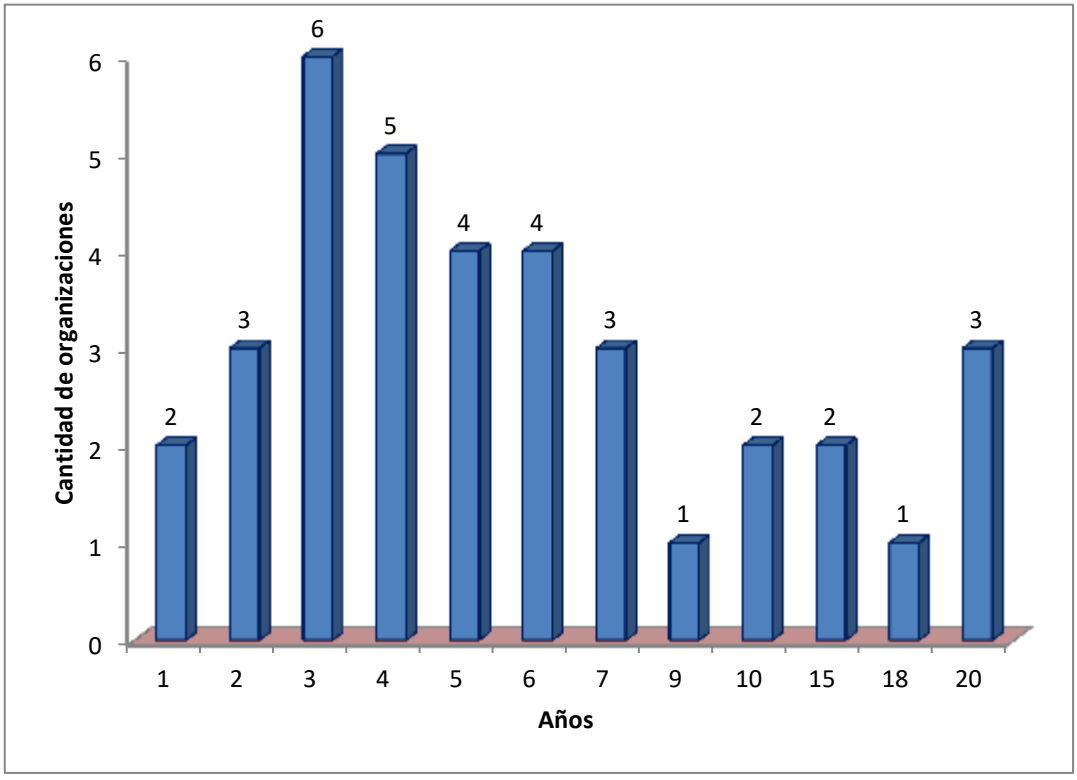

Figura 12. Años o tiempo de funcionamiento de la organización.

Como se deja ver en la Figura 12, lo que se observan son organizaciones bastante nuevas, pero tal vez, eso no implica que sus miembros tengan una amplia experiencia en el campo de desarrollo de software. Un total de 28 organizaciones con menos de 10 años de vida corporativa, situación ésta que es muy positiva para el desarrollo del software en Panamá por la oportunidad que representan, en la aplicación de modelos de madurez de software libre a futuro.

Pregunta 2: **¿Qué tipo de software desarrolla: escritorio, aplicaciones web, páginas Web, móvil, otros?**

En cuanto a la pregunta sobre el tipo de software que desarrollan (Figura 13), las organizaciones encuestadas desarrollan aplicaciones Web en un 35,8%, seguida por las páginas Web con 28,3%. Las aplicaciones de escritorio se mantienen en un 20,8%, mientras que las móviles que se esperaba que debieran tener un porcentaje mayor se mantienen con un bajo 15,1%. Esto quiere decir que contrariamente a lo que se pensaba, no son las aplicaciones móviles las de mayor desarrollo. Las aplicaciones móviles aún les falta por despegar a pesar de la irrupción de los teléfonos inteligentes y la popularidad de los mismos.

Es decir, hay una falsa creencia que los desarrolladores de software están focalizados en esta "modalidad", lo que de acuerdo a estos resultados no es cierto, ya que, vemos que es en la Web donde concentran su atención. Hubo cero respuestas para "otros" lo que significa que estas 4 aplicaciones (escritorio, Web, páginas Web y móvil) son las que se desarrollan principalmente en Panamá, por no decir que son las únicas, de acuerdo a este estudio.

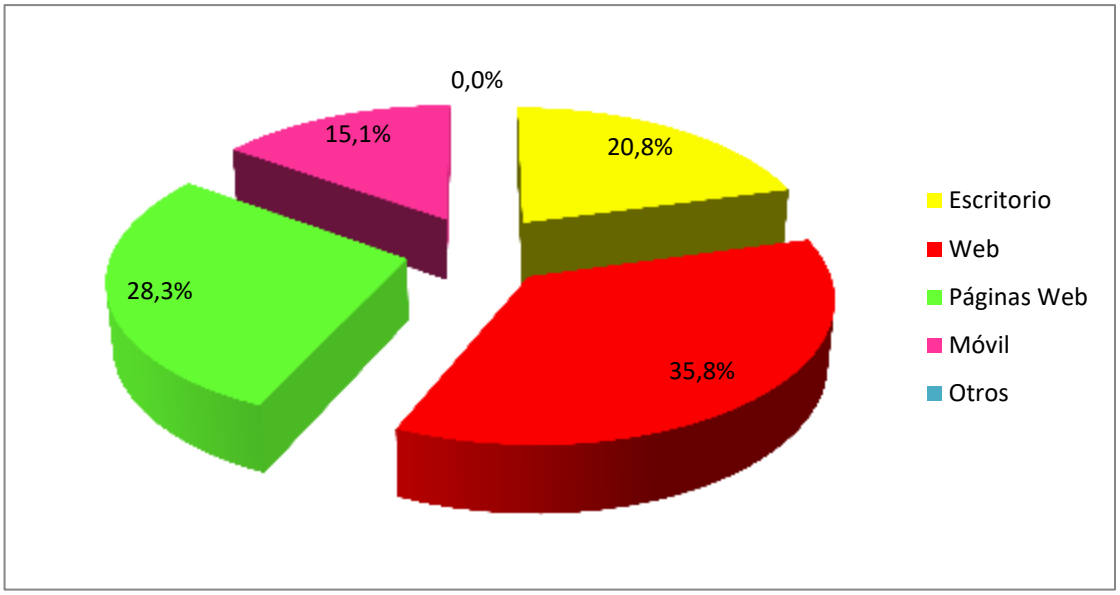

Figura 13. Tipo de software que se desarrolla.

Pregunta 3: **¿Desarrolla software para el mercado?**

Nacional □ **Internacional** □ **Ambos mercados** □

Las organizaciones (Figura 14) que desarrollan software para el mercado nacional alcanzan un 52,8%. De forma mayoritaria el desarrollo de software se concentra para el mercado local, mientras que solo un 25% se dedica al mercado internacional y un 22,2% a ambos mercados.

El mercado local tiene un mayor predominio, quizás porque conocen la idiosincrasia del país y el manejo comercial es más directo. Además, se trata de empresas bastantes jóvenes, como ya se vio, por lo que quizás aún no se aventuran, muchas, a cruzar fronteras. O bien están conformes con dedicarse al mercado local. Por otro lado, podría ser que no sean lo suficientemente grandes para incursionar al mercado internacional. En resumen, se dan los tres tipos de mercados, aunque en menor escala los dedicados al exterior.

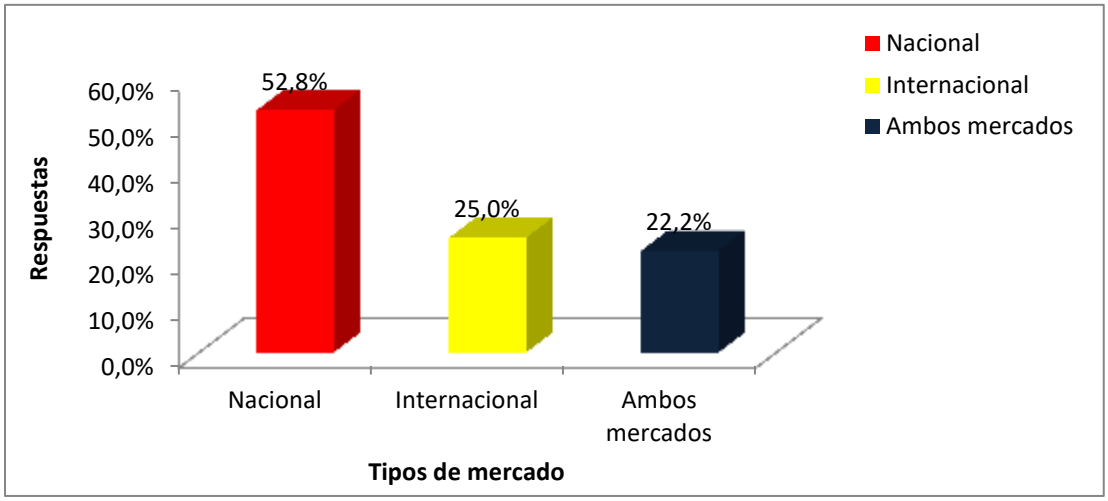

Figura 14. Mercado de software.

Pregunta 4: **¿Cuántos años tiene formado el equipo de desarrollo?**

Los años que tienen de formado el equipo de desarrollo, como se puede apreciar en la Figura 15, un 2,8% están entre los 9 y 16 años (8 organizaciones). Cuatro están entre el 5,6% y el 8,3%. Con mayor cantidad de años de formación rondan los 4 años (16,7%), 5 años (13,9%) y 3 años (11,1%). La mayoría son organizaciones relativamente nuevas.

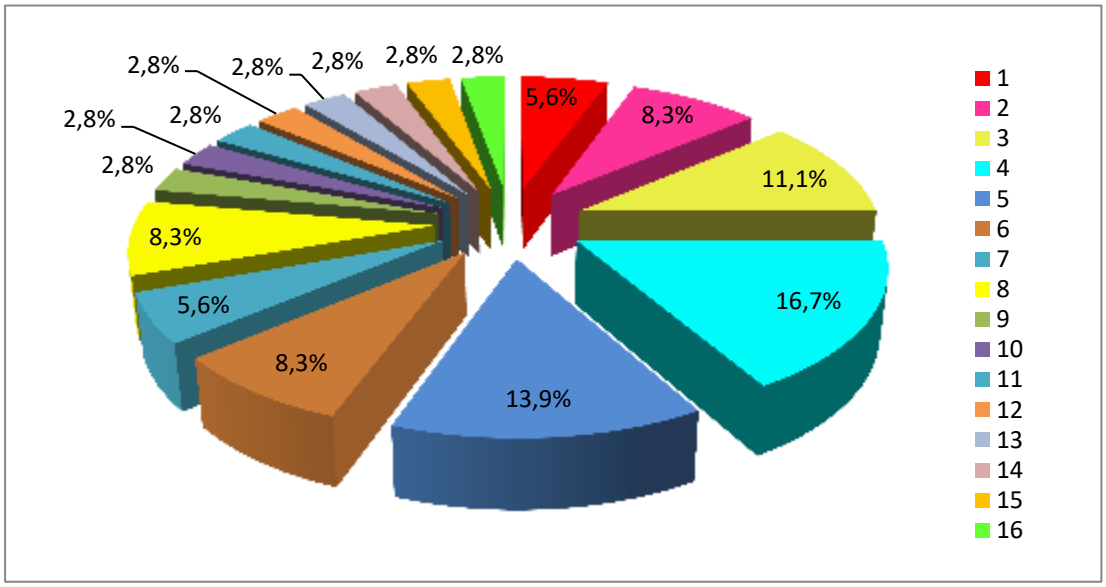

Figura 15. Años que tienen de conformado el equipo de desarrollo.

Pregunta 5. ¿Qué formación académica tiene el equipo de desarrollo?

Técnico □

Licenciatura □

Maestría □

Doctorado □

Los equipos de desarrollo (Figura 16) en su mayoría están integrados por personal con licenciatura (66,7%) y con grado de técnico (16,7%). Personal con maestría un 11,1% y en menor porcentaje (5,6%) con doctorados. Esto puede deberse a que el personal de maestría o doctorado ocupen otras posiciones en la organización o no estén involucrados en desarrollo de forma directa. O bien hay poco personal formado con grado de doctor en desarrollo de software y sobre todo, que estén como desarrolladores de software. También puede darse el caso que las organizaciones no consideren disponer de personal con estos estudios porque no los consideran necesarios en este tipo de trabajo.

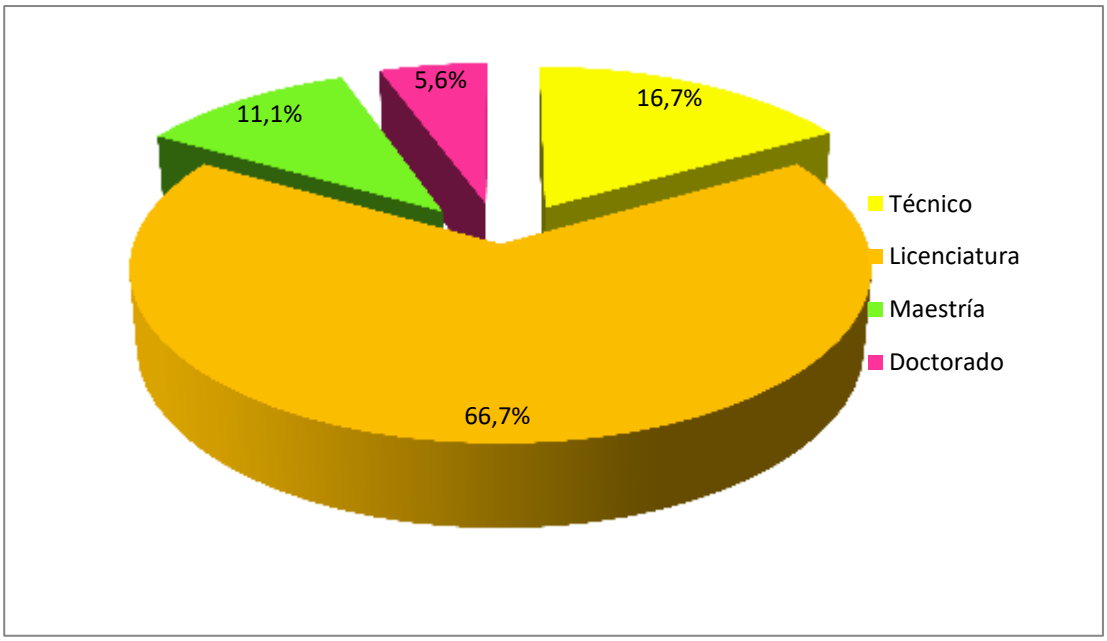

Figura 16. Formación académica de los desarrolladores.

Pregunta 6: **¿Los desarrolladores utilizan software libre para el desarrollo de aplicaciones?**

En esta pregunta (Figura 17) el 91,7% respondieron que sí utilizan software libre para el desarrollo de aplicaciones. Solo el 8,3% respondió que no. Estos resultados demuestran que una cantidad importante de desarrolladores utilizan las ventajas del software libre a la hora de desarrollar software. No obstante, es importante saber cómo se decantan por un determinado software libre. Qué aspectos consideran, si realizan alguna evaluación o aplican algún modelo de madurez.

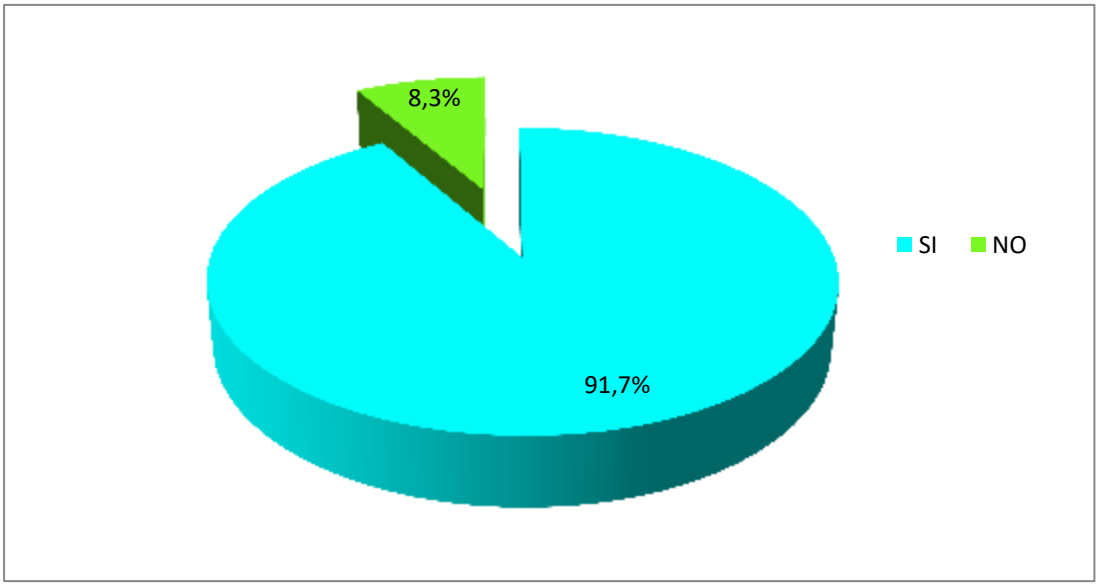

Figura 17. Uso de software libre por los desarrolladores.

Pregunta 7: ¿El equipo de desarrollo participa de algún proyecto de software libre?

Solo once participantes, que representan el 30,6%, indicaron que sí (Figura 18). Mientras que los otros 25 participantes (69,4%) dijeron que no. Como se observa son pocos los que participan de este tipo de proyectos según los datos obtenidos.

Esta respuesta sorprendió un poco, por las respuestas obtenidas en la pregunta anterior donde el 91, 7% dijo que sí utilizan software libre para el desarrollo de las aplicaciones. Esto daba margen a pensar que un porcentaje mayor al 30,6% participaran en algún proyecto de software libre lo que no es así. De hecho se hace necesario desarrollar más proyectos de este tipo donde puedan involucrarse y aportar al grupo de software libre.

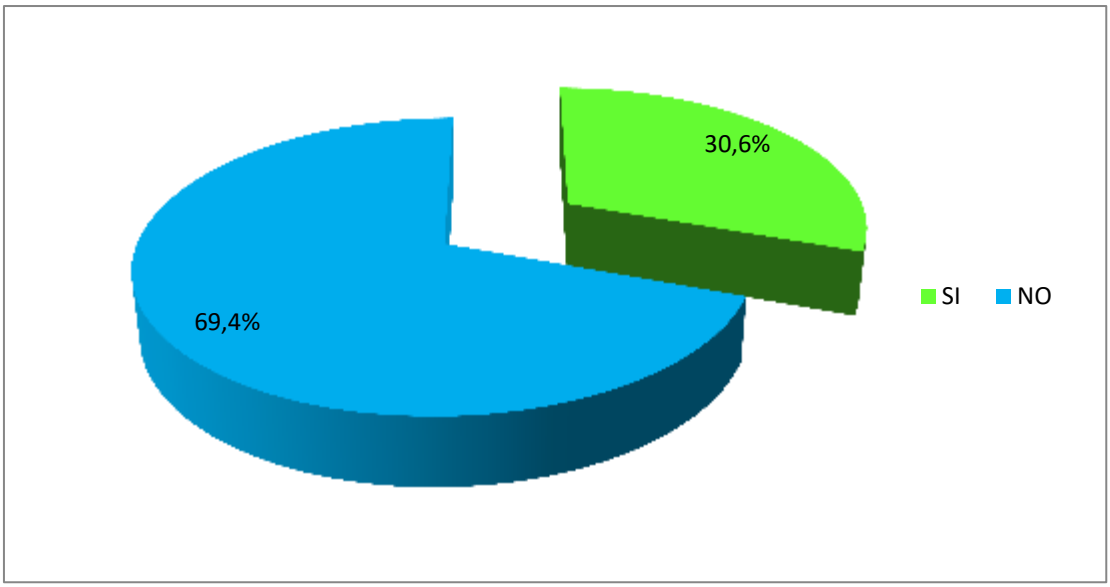

Figura 18. Participación de los desarrolladores de algún proyecto de software libre.

Pregunta 8: ¿En qué tipo de aplicaciones utiliza software libre para el desarrollo?

De acuerdo con estos resultados el 35,8% de los participantes (Figura 19) dicen utilizar software libre para el desarrollo de aplicaciones Web. Un 28,3%, utiliza software libre para el desarrollo de páginas Web. Sin embargo, un poco menos lo utilizan para el desarrollo de aplicaciones de escritorio 20,8%, móvil 15,1% y para otros no respondieron.

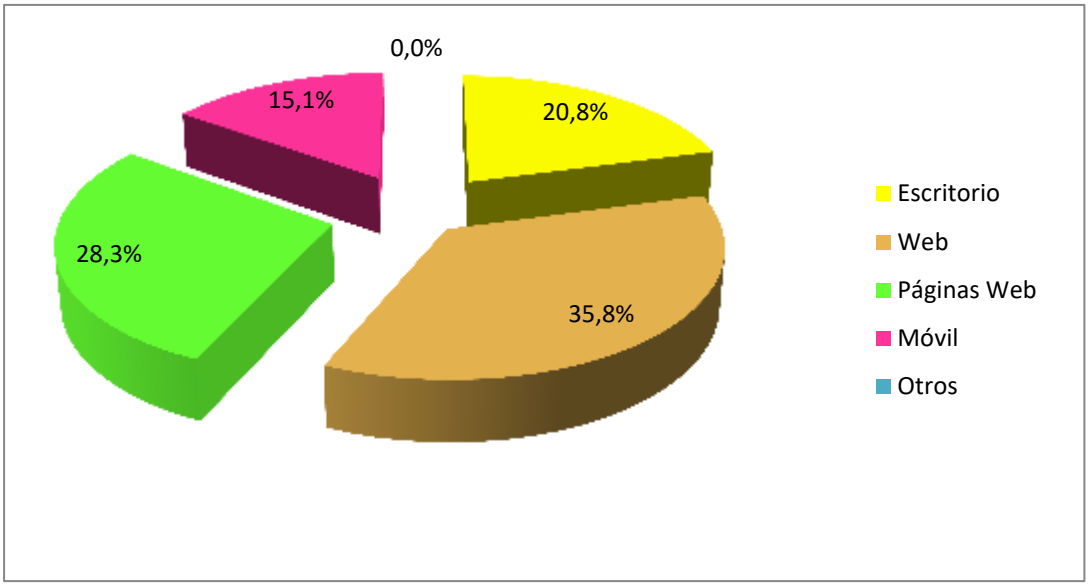

Figura 19. Tipo de aplicaciones donde utilizan software libre.

Pregunta 9: **¿Mencione qué software libre utiliza para desarrollar las aplicaciones?**

Según los resultados obtenidos (Figura 20) para esta pregunta, se detectaron un total de 18 software libre diferentes que se utilizan. El software libre que marca tendencia como herramientas para desarrollo es el Joomla con un 16,2%. Le sigue Wordpress con un 13,5% y Eclipse con un 10,8%. Esto quiere decir que son los tres software libres más utilizados en Panamá, por ende, los que tienen mayor uso. No obstante, seguido están herramientas como: MySQL, PostgresSQL, Laravel, Phyton y Ruby and Rail con un (5,4%) respectivamente. Algunos otros con menor uso, que no significa que no se puedan utilizar o que también tengan sus bondades, pero que son menos explorados.

Es clara la tendencia del uso del software libre para el desarrollo de aplicaciones enfocadas al uso de la Web.

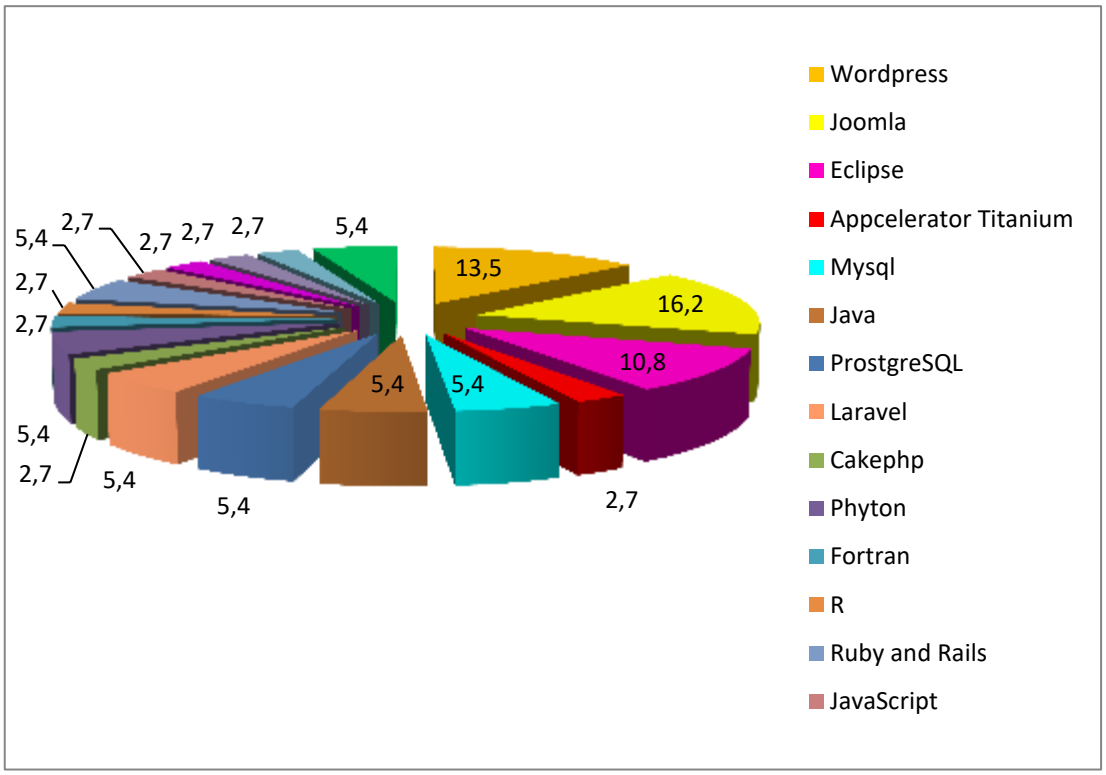

Figura 20. Software libre utilizados para desarrollar las aplicaciones.

6.2. RESULTADOS DEL APARTADO II: NIVEL DE MADUREZ

Igual que se hiso en el apartado anterior, se aborda cada pregunta de la encuesta correspondiente a este apartado.

Pregunta 10: ¿El equipo de desarrollo tiene conocimientos en modelos de madurez de software?

Si □ ¿Por qué si? _____.

No □ ¿Por qué no?_____.

Un 88,9% (32) de los desarrolladores de software encuestados no tenía conocimiento sobre modelo de madurez de software (Figura 21). Mientras que un 11,1% (4) si conocen de la existencia de los modelos de madurez de software. Los que respondieron que sí tienen conocimientos de modelo de madurez de software, indicaron que su uso es porque con los modelos: -se puede corroborar si el software a utilizar es lo suficientemente eficaz, -probar la eficiencia del software, -que cuenta

con un equipo o comunidad que lo avala, -que la documentación sea clara y que las mejoras y las ayudas de los usuario les permite contar con un software estable y confiable.

Sin embargo, los que respondieron que no, señalan que no conocían el término, no hay espacio en el cronograma para conocerlos o estudiarlos, o simplemente no lo conocen.

Estos resultados indican que una gran mayoría desconoce la existencia de los modelos y de sus buenas prácticas para la selección de software libre. Por otro lado, estos modelos permiten mejorar el enfoque para la selección de software, lo que redundaría en garantizar un software que cumpla con ciertas especificaciones de funcionalidad, seguridad, respaldo de una comunidad de usuario, actualizaciones, entre otros muy importantes a la hora de implementar software libre para desarrollo de software. Sobre todo, considerando su madurez que está relacionado a diferentes factores o variables, implica que es un software que ha estado en el mercado por mucho tiempo ya ha pasado por un proceso de mejora continuo a lo largo de sus años.

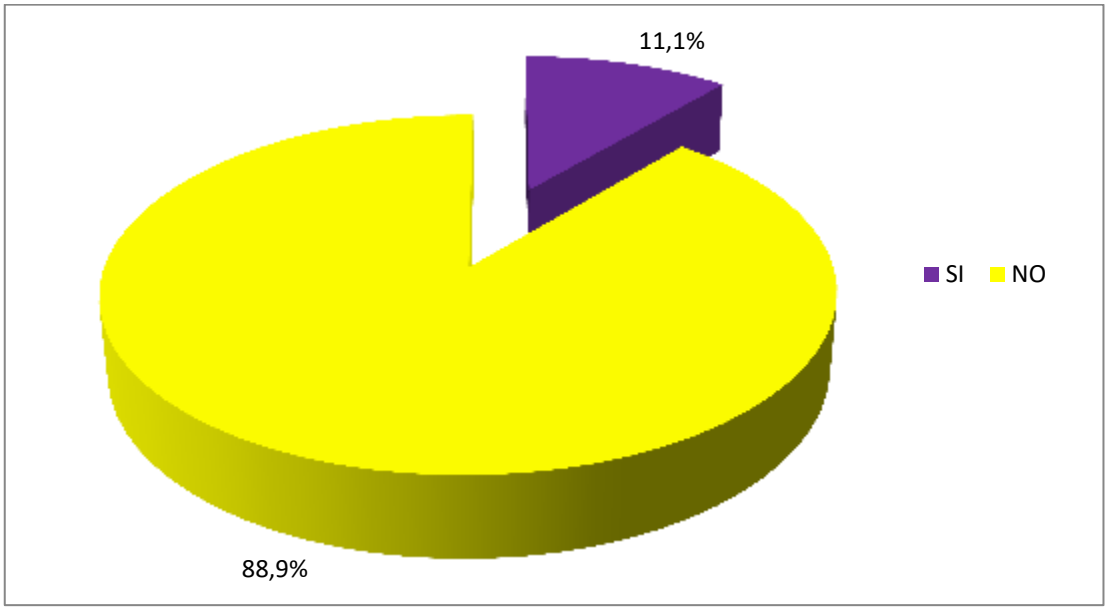

Figura 21. Conocimiento en modelos de madurez de los desarrolladores.

Pregunta 11: ¿La organización implementa modelos de madurez en el desarrollo de software?

Los resultados (Figura 22) indican que el 69,4% (25) de los participantes señalan que la organización no implementa un modelo de madurez. 30,6% (11) indicó que sí. Como vemos un alto porcentaje de las organizaciones no aplican modelos de madurez en el desarrollo de software. Esto puede deberse a la falta de conocimiento de las ventajas de su uso, falta de entrenamiento en su implementación o el factor de rentabilidad económica.

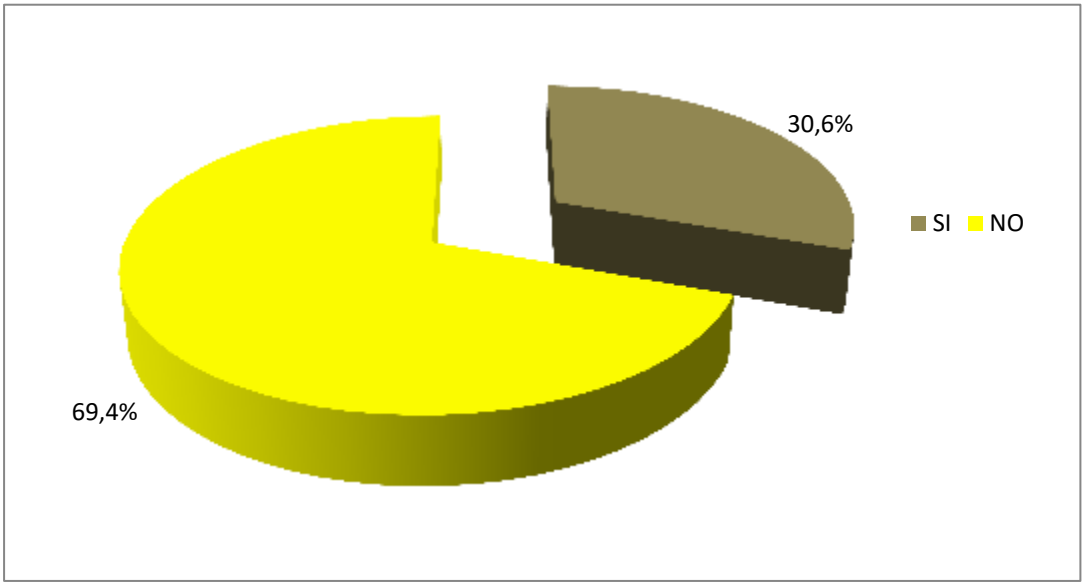

Figura 22. Implementación de modelos de madurez en el desarrollo de software.

Pregunta 12: ¿Realizan una evaluación del software libre que utilizan para el desarrollo de los productos?

Los desarrolladores participantes que indicaron (Figura 23) que nunca realizan evaluación fue un 25,0%. En este caso es posible que la selección del software se deba a varias razones que pueden ser porque está de moda el software, goza de buena reputación o está siendo ampliamente utilizado en el momento. Lo que, consideran que es bueno y que reúne o posee las características para utilizarlo en el desarrollo de sus productos, ya que, otros lo han implementado o simplemente porque es el que usa

la comunidad de desarrolladores y reúne los requisitos para desarrollar los productos. Pero no ha sido, tal vez, evaluado bajo ciertos criterios o características para demostrar su madurez. Por otro lado, un 44,4% lo evalúa de forma ocasional. Es decir, que puede ser de vez en cuando por alguna necesidad técnica o que por otras causas se vean obligados a realizar algún tipo de evaluación.

El 30,6% si evalúa el software. Si al 25% que dijo no evaluar el software se le suma el 44,4% que dijo que lo hace de forma ocasional, asumiendo que ocasional puede ser una sola vez o alguna vez en "x" tiempo, se podría decir que un porcentaje bastante alto de las organizaciones desarrolladores de software en nuestro país, no evalúan el software que utilizan para desarrollar sus productos.

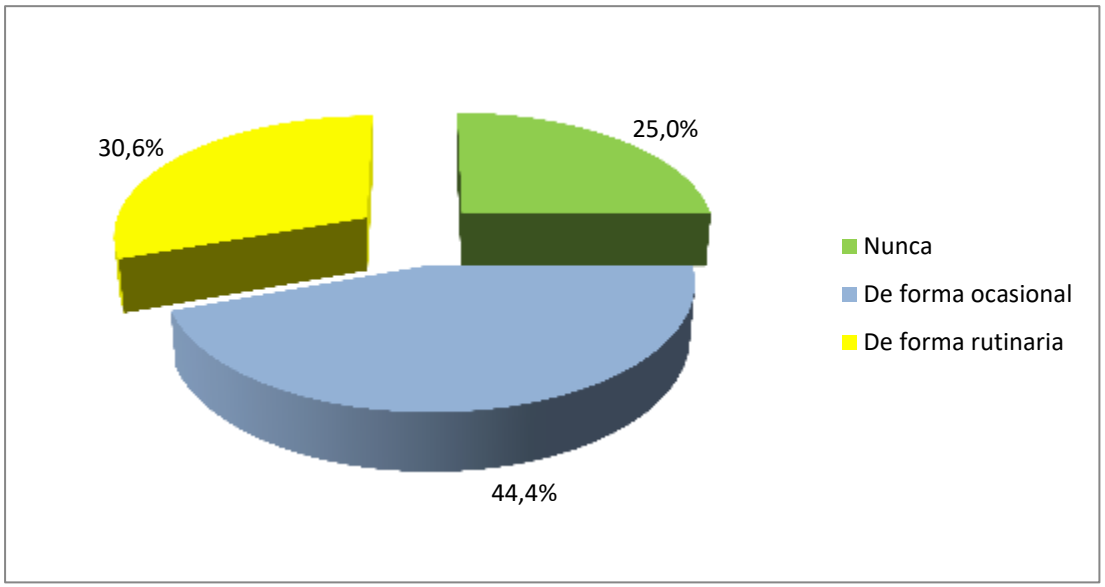

Figura 23. Evaluación del software libre que utilizan para el desarrollo de los productos.

Pregunta 13: **La evaluación del software libre lo realiza mediante:**

Una metodología □

Una herramienta □

Un modelo de Madurez □

No se realiza evaluación del software □

Cabe destacar de estas respuestas (Figura 24), que solo el 5,6% aplican modelo de madurez para software libre, lo que puede deberse a que no hay una cultura en las organizaciones, al menos de este estudio, de aplicar modelos de madurez de software libre para evaluar el software a la hora de desarrollar sus productos. Lo que es impresionante dada las ventajas corporativas que ofrecen. Llama la atención el 30,6% que respondió que no realiza evaluación del software, sobre todo, porque en la pregunta anterior (Pregunta 12) fue un 25 % que dijo que no evaluaban el software libre que utilizaban. No hay una correlación entre ambas preguntas, que era lo esperado. Esto significa que el 25% que dijo que nunca evaluaban el software en realidad es de un 30%, lo que hace pensar que no respondieron con objetividad. Por otro lado, para la evaluación del software un 22,2% aplican una metodología y un 41,7% alguna herramienta.

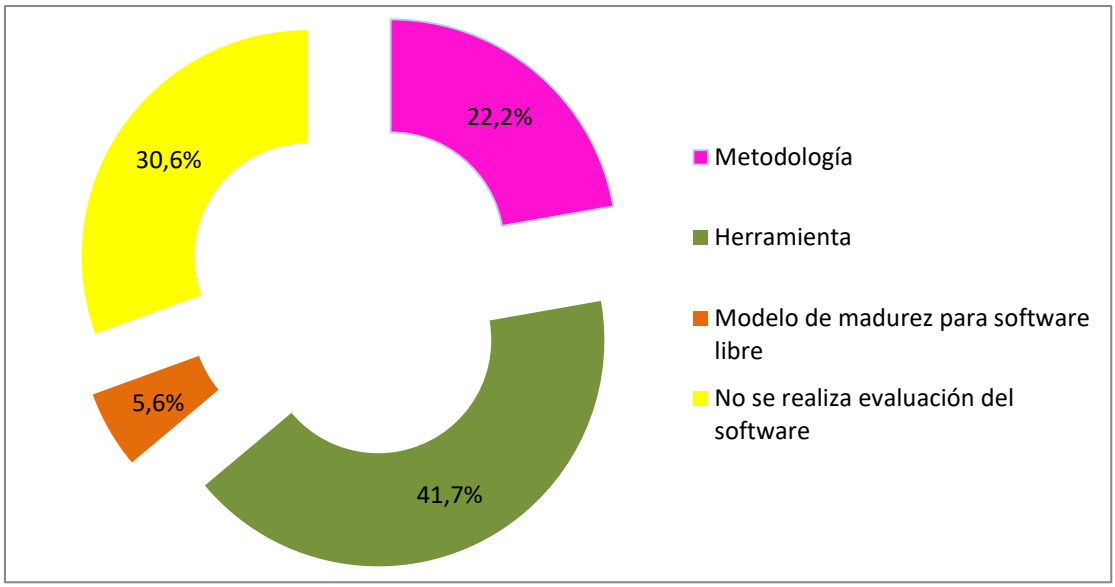

Figura 24. Metodologías para la evaluación de software libre.

A pesar de los resultados, no causa asombro que solo un 5,6% use modelo de madurez para evaluar software libre, ya que, se correlaciona con la Pregunta 10 donde un 88% dijo que no conocía de los modelos de madurez de software. En su mayoría indicaron que utilizan una metodología o una herramientas para su evaluación, en este caso no se preguntó que metodología o qué herramienta. Lo que puede ser una investigación a futuro.

Pregunta 14: **¿Por qué utiliza software libre para el desarrollo de productos?**

Las razones que expusieron los participantes de por qué utilizan software libre para el desarrollo de los productos, son variadas (Figura 25). Es destacable el 41,5% que señalaron el costo de la licencia del software. En este caso la mayoría se refería que el software es gratis, que no debía pagar por su uso entre otros. Un 14% indica que por la comunidad de usuario que apoyo el software, lo que permite intercambiar ideas sobre la instalación, uso, funcionamiento o modificación del software. Por la accesibilidad y por la modificación del código el 9,8%, respectivamente.

El resto indican que por disponer de más actualizaciones, por la documentación y funcionalidad o por lo robusto del software. Este último en un porcentaje muy bajo (2,4%). De esta forma, son dos los aspectos claves para elegir un software libre: cero costos de la licencia y la comunidad de usuario que lo respaldan.

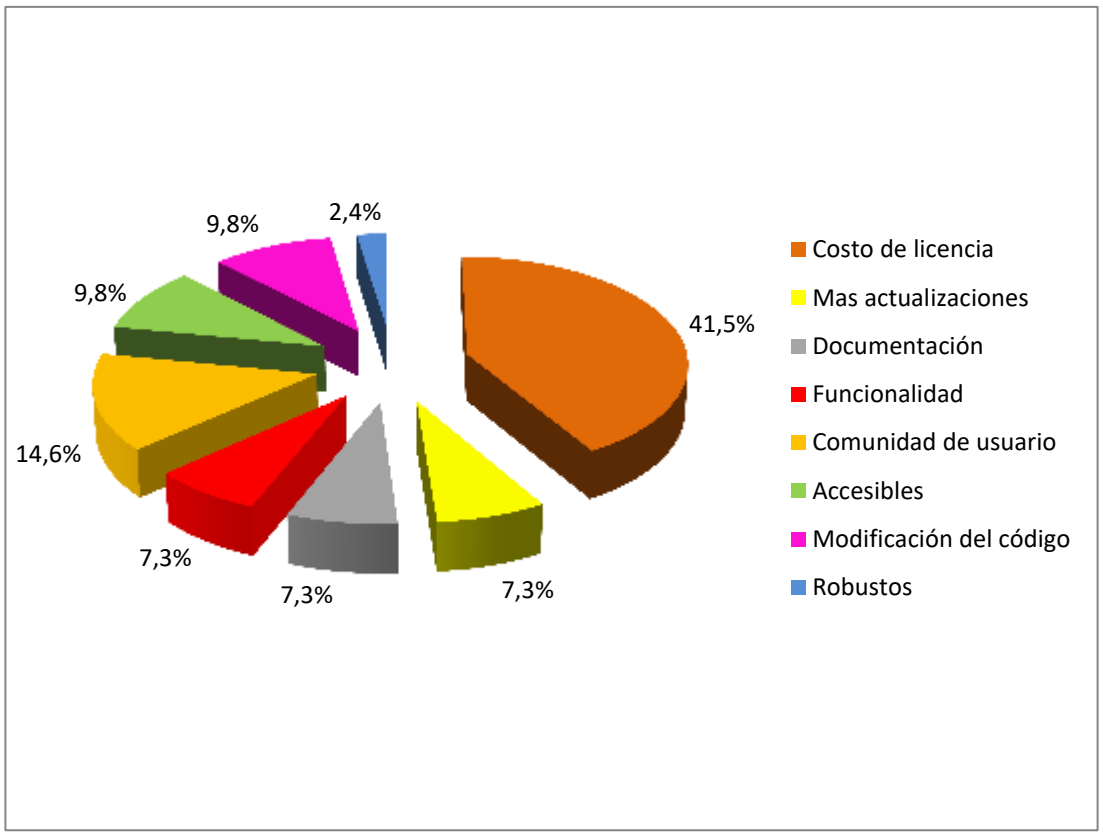

Figura 25.Razones por las que utiliza software libre para el desarrollo de productos.

Pregunta 15. ¿Qué indicadores o características importantes considera en la selección de software libre?

En este caso, son diez las características que los desarrolladores encuestados consideran importantes en la selección de software libre (Figura 26). En primer término (26,7%) está la funcionalidad del software. Un aspecto tan importante como la seguridad solo fue considerado por un 20%. Llama la atención este hecho, ya que, la seguridad de un software previene la vulnerabilidad tanto del mismo software, como de la aplicación que se desarrolla con el software. Si no hay seguridad se expone al usuario a ser objeto de la delincuencia informática. Otra característica es la documentación con un 13,3%, mientras que el rendimiento obtuvo un 8,9%. En menor porcentaje la eficiencia, la mejora en el código fuente y confiabilidad con un 2,2% respetivamente.

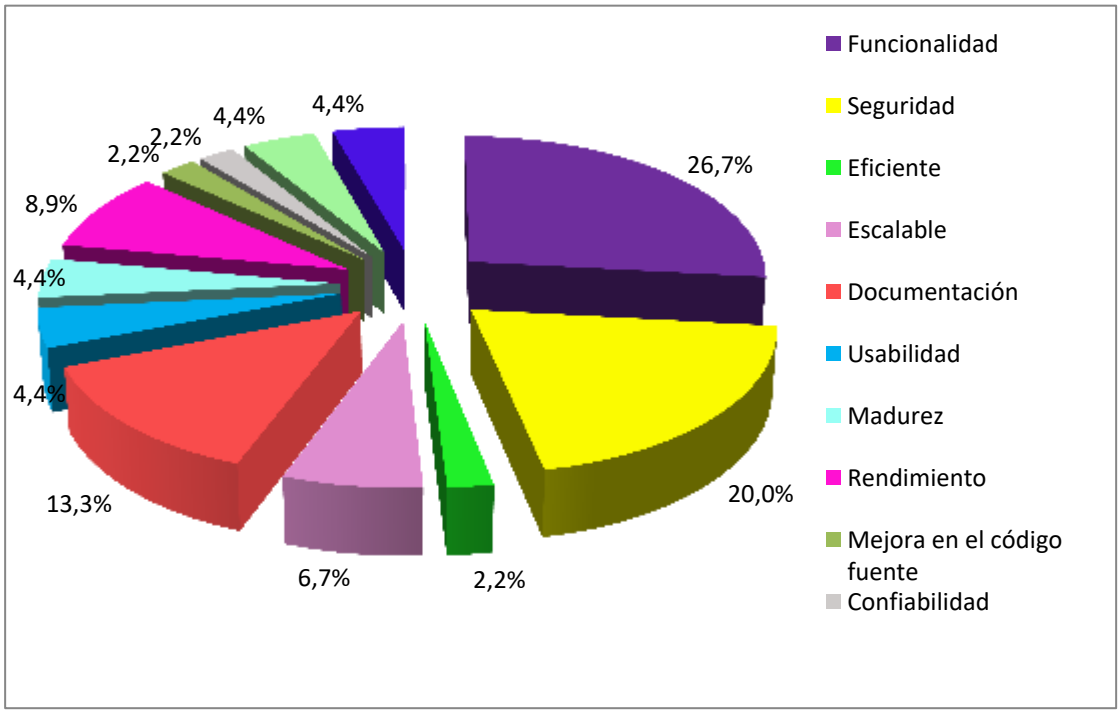

Figura 26. Características señaladas, por los participantes, en la selección de software libre.

Con respecto a la madurez del software, solo un 4,4% la consideró importante en la selección de software libre. Esto correlaciona con las Preguntas 10, 12 y 13, ya antes discutidas. Es probable que no se le da la importancia que tiene esta característica, por

el desconocimiento de la existencia del modelo de madurez del software, porque no lo aplican o usan y por el desconocimiento de su valor en el desarrollo de software. También se puede deber a la falta de formación, por parte de los desarrolladores, de que los modelos te brindan una mayor certeza de los software que vas a utilizar y consideran que la funcionalidad y la documentación son las principales características, sin comprender que la madurez del software involucra todos estos aspectos.

Pregunta 16. ¿Para la selección del software libre es importante su madurez?

Si □ ¿Por qué si? _____.

No □ ¿Por qué no?_____.

Un 91,7% (Figura 27) considera que es importante la madurez del software para su selección. Sin embargo, esto no correlaciona con las respuestas obtenidas en la pregunta anterior (Pregunta 15) donde solo un 4,4 % señaló que la madurez del software es una característica importante en la selección del software libre. En la respuesta de **¿Por qué sí?** Respondieron lo siguiente:

- Que el grado de madurez y seguridad es importante
- Debe existir documentación
- Que funcione de forma correcta
- Por su eficiencia y prolongado periodo de mantenimiento
- Tiene una comunidad que lo sostiene y respalda
- Por su confiabilidad, menos propenso a errores y no está en etapa de desarrollo.

Por otro lado, solo el 8,3% considera que no es importante su madurez y no se obtuvo respuesta de por qué no consideran qué es importante. Esto causa algo de incertidumbre porque consideran que no es importante, pero sin embargo, no saben

por qué no es importante, no dieron razón alguna. Lo que interpreto como falta de conocimiento sobre el tema o del trabajo que se realiza.

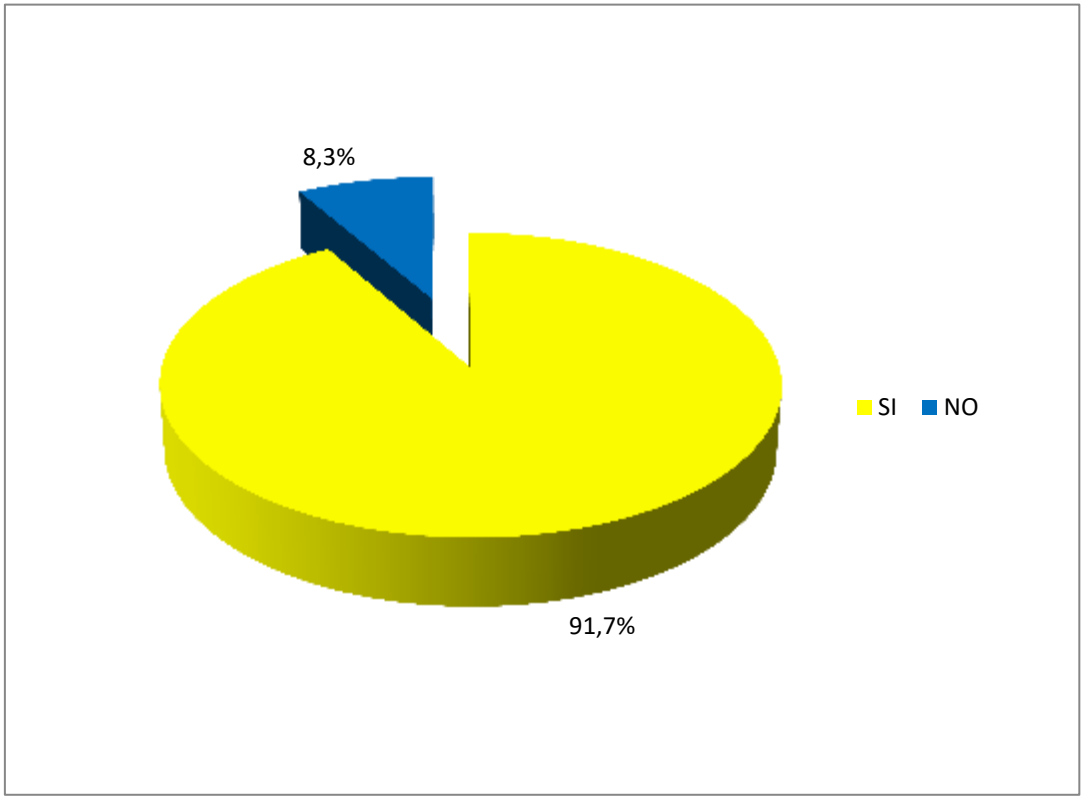

Figura 27. Opinión sobre la importancia de la madurez del software libre.

Pregunta 17: **¿Conoce o ha escuchado de algún modelo de madurez para software libre?**

El 38,9%, indicó que sí ha escuchado sobre modelo de madurez de software libre (Figura 28). Mientras que una gran mayoría (61,1%), señaló que no ha escuchado sobre modelo de madurez para software libre. Esto implica que una mayoría desconoce sobre los diferentes modelos de madurez para software libre, lo que se convierte en un impedimento para poder elegir, correctamente, en la selección de software libre para realizar trabajos. Lo que es una limitante a la hora de aplicar un software o evaluar el software.

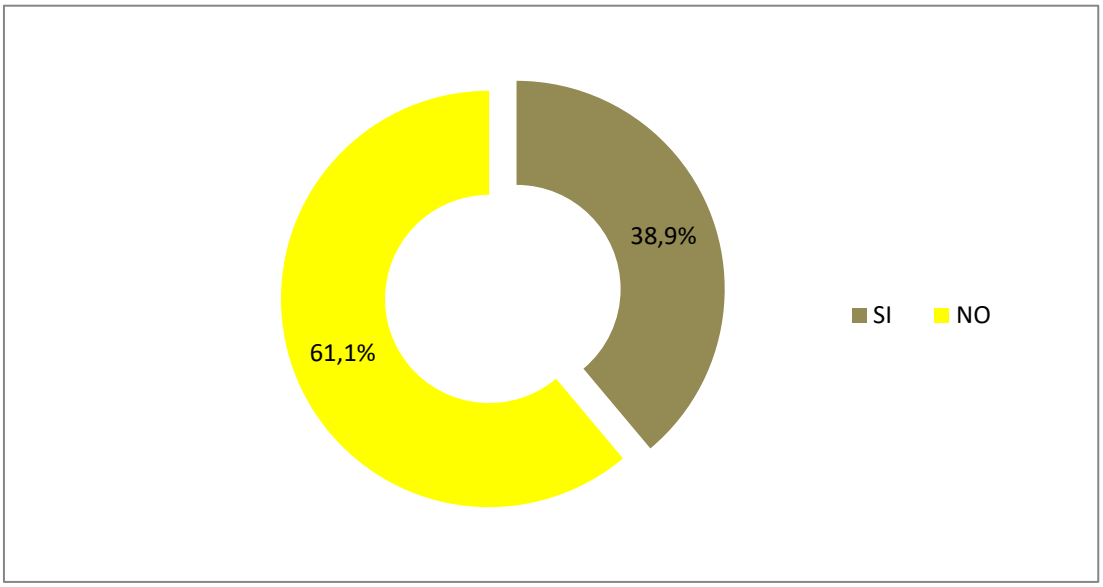

Figura 28.Opinión de los desarrolladores sobre si conoce o ha escuchado de algún modelo de madurez de software libre.

Pregunta 18. ¿De los siguientes indicadores o características de madurez de software, cuáles considera para la selección de software libre?

Licencia	☐
Comunidad de usuario	☐
Soporte detención de errores	☐
Gestión del proyecto	☐
Aseguramiento de la calidad	☐
Modificación de código fuente	☐
Rendimiento	☐
Usabilidad	☐
Escalabilidad	☐
Funcionalidad	☐
Modo de distribución	☐
Versión	☐
Todas la anteriores	☐

La funcionalidad y la comunidad de usuarios (Figura 29) son consideradas para la selección de software (10,5%) por encima de los otros indicadores o característica, soporte y usabilidad (9,3%), rendimiento del software (8,6%) y licencia (8,0%). Las otras características son consideradas en menor grado. El modo de distribución es la que menos consideran (3,7%). Cabe resaltar que son características que se establecen en los modelo de madurez de software libre. Como se indicó en los resultados la funcionalidad y la comunidad de usuarios son esenciales para los encuestados. Esto puede deberse a que la comunidad de usuario es un apoyo fundamental para la solución de problemas en el uso del software. Sin embargo, la usabilidad no está muy lejos como un requisito para su selección. Es decir, si el software presenta usabilidad es un valor agregado para su uso. Algo muy importante que resaltar es que la respuesta todas las anteriores solo es (4,9%) cuando se esperaba fuera la de mayor porcentaje, dado que todas son consideradas de una forma u otra en los modelos de madurez de software libre.

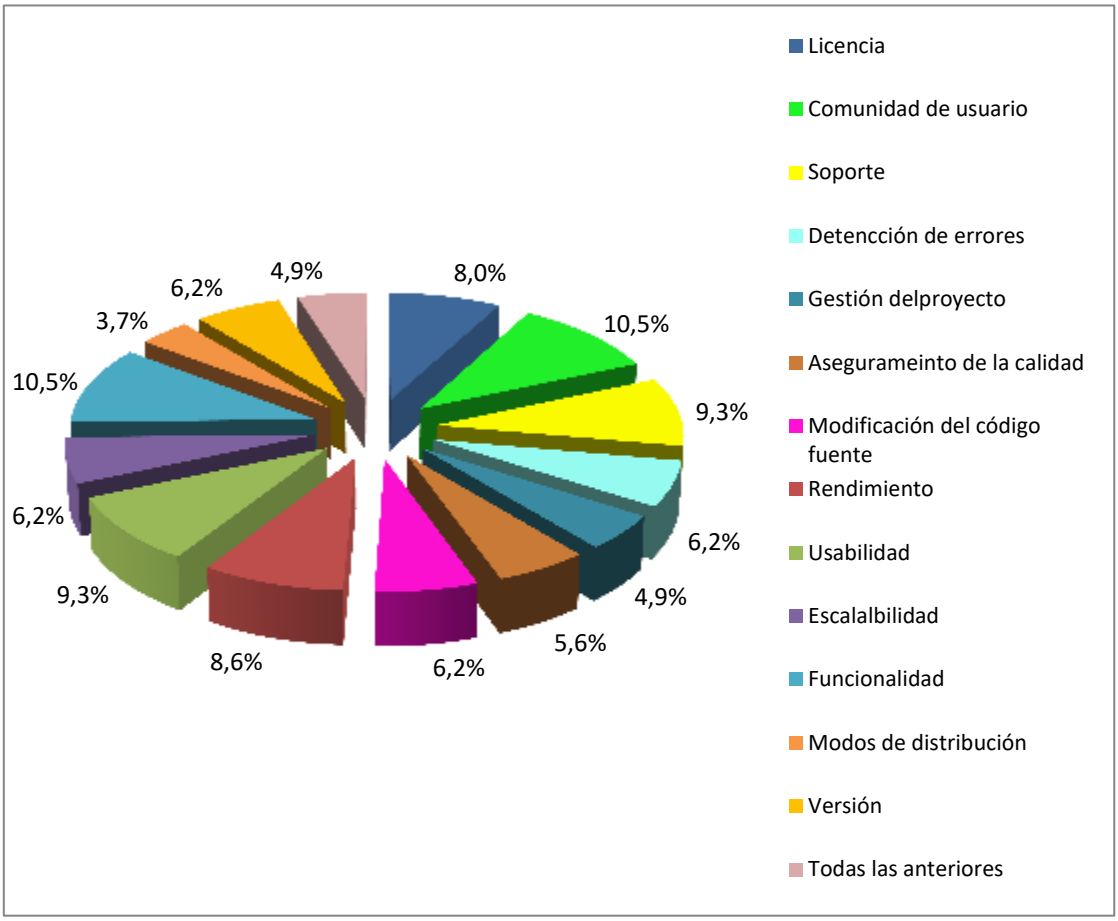

Figura 29. Indicadores o características de madurez de software que consideran.

Pregunta 19: ¿Aplica un modelo de madurez de software libre para la selección del software?

Si □ ¿Por qué si? _____.

No □ ¿Por qué no?_____.

Los resultados obtenidos a esta pregunta se pueden observar en la Figura 30. Como se observa en el gráfico, el 11,1% de los participantes indicó que ha aplicado un modelo de madurez de software libre para la selección del software. A la pregunta ¿Por qué sí? indicaron que para corroborar que el software es eficaz, confiable, eficiente, que dispone de una comunidad organizada de usuarios, que la documentación es clara.

Mientras que un significativo 88,9% indicó que no. De esto se deduce que son pocos los desarrolladores que utilizan un modelo de madurez de software libre, para la selección del software. A la pregunta ¿Por qué no? señalaron que por desconocimiento no aplican un modelo de madurez de software libre para la selección del software. Con estos resultados queda demostrado que la mayoría de los desarrolladores, encuestados en este estudio, no aplican modelos de madurez de software libre para la selección del software; y que los mismos no lo hacen por desconocimiento. Por consiguiente, se hace necesario dar a conocer los modelos de madurez de software entre los desarrolladores de software en ejercicio y a los futuros desarrolladores.

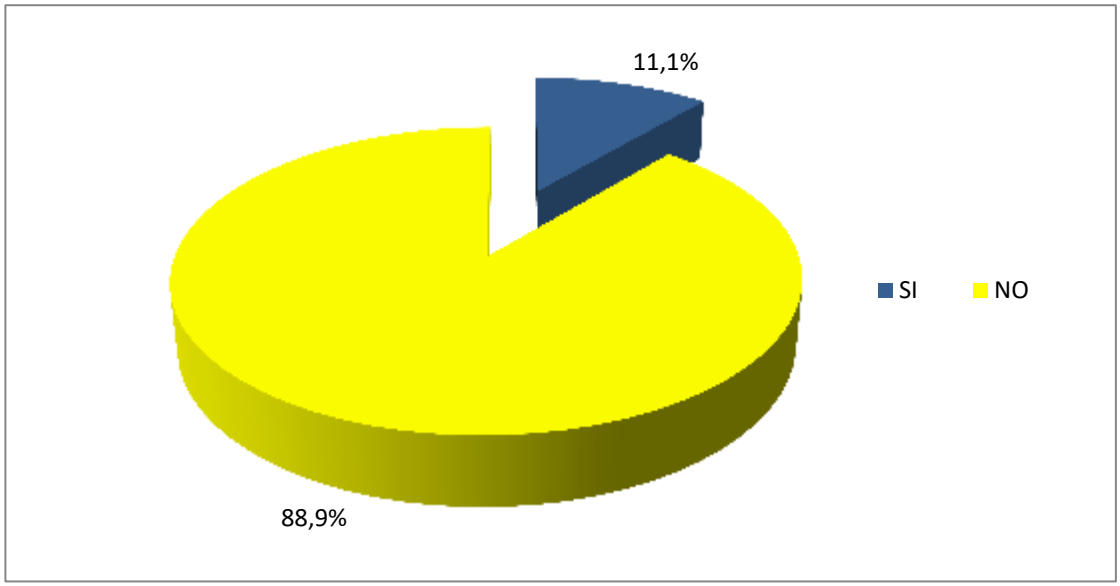

Figura 30. Aplica un modelo de madurez de software libre.

CAPÍTULO 7

CONCLUSIONES Y RECOMENDACIONES

7. CONCLUSIONES Y RECOMENDACIONES

7.1. CONCLUSIONES

Con respecto a la hipótesis planteada, la misma se rechaza. Un alto porcentaje de los desarrolladores de software en Panamá, no evalúan el software que utilizan para desarrollar sus productos o si lo hacen alguna vez, es por una circunstancia "X" que les amerite o les obligue a hacerlo.

Con respecto a los objetivos trazados, basados en los resultados de esta investigación se concluye que:

- Existe un desconocimiento marcado por parte de los desarrolladores de software y por ende de las organizaciones, de la existencia de modelos de madurez enfocados en software libre y de su utilidad para la selección de software libre que cumpla con su madurez, razón por la cual no aplican modelos de madurez de software libe o de código abierto.

- El 94,5% de los desarrolladores de software de este estudio no aplican modelos de madurez de software libre a la hora de desarrollar sus productos en las empresas donde laboran.

- Son muy pocos los desarrolladores de software, que dijeron no usar software libre para el desarrollo de software, en las organizaciones desarrolladoras de software.

- El tipo de aplicaciones que más desarrollan son las relacionadas a páginas Web y aplicaciones Web, observándose una relación directa entre herramientas y tipo de aplicaciones..

- El 63,9% de los desarrolladores de software utilizan una metodología y una herramienta para la evaluación de software libre, sin especificar cuál.

- Las características de calidad que más valoran los desarrolladores de software, para la selección del software a aplicar, son la funcionalidad y la seguridad.

- Los desarrolladores de software, al parecer, tienen claro que es de suma importancia y fundamental la madurez que alcance el software libre. Sin embargo, existe una ausencia en el uso o aplicación de los modelos de madurez de software libre, para el desarrollo de software.

7.2. RECOMENDACIONES

Entre las recomendaciones más destacables, producto de este trabajo, se pueden indicar las siguientes:

- Dar a conocer a las diferentes empresas desarrolladoras de software, los modelos de madurez de software libre o de código abierto, para que puedan evaluar su aplicación en sus productos.

- Realizar seminarios y conferencias de divulgación de los modelos de madurez de software libre. De esta forma se dan a conocer las facilidades que ofrecen.

- Adecuar el contenido dentro de los cursos académicos regulares, correspondientes, con los modelos de madurez de software libre o de código abierto, lo que garantiza que los futuros desarrolladores conozcan sobre esta alternativa.

- Desarrollar e implementar un multi modelo de madurez de software libre.

- Desarrollar proyectos de software libre que involucren al relevo generacional, o el ya en ejercicio, en organizaciones y/o en las universidades.

- Desarrollar líneas de investigación sobre modelos de madurez de software libre, que permitan realizar investigaciones como:

    - Una revisión de metodologías y herramientas para la selección de software libre o de código abierto.

    - Aplicación de software libre o de código abierto para comparar las herramientas utilizadas y los resultados con los modelos de madurez de evaluación de software libre.

    - Determinar, por ejemplo, cuáles son los modelos de madurez de software libre, que utilizan las organizaciones desarrolladoras de software en Panamá.

    - Determinar que metodologías o herramientas utilizan para evaluar los software y qué aspectos abarcan los diferentes modelos de madurez para software libre que utilizan.

Como se puede ver, es mucho lo que se puede hacer en esta línea de investigación. Ha quedado demostrado, con esta tesis, que el tema tiene potencial para una nueva línea de investigación novedosa y actual.

8. REFERENCIAS BIBLIOGRÁFICAS

Abad, A. y Servín, L. (1993). Introducción al muestreo. México: Editorial Limusa.

Adewumi, A., Misra. S., Omoregbe, N., Crawford, B. and Soto A., R. (2016). Systematic literature review of open source software quality assessment models. SpringerPlus, 8:5(1):1936.

AUEB. (2008). Software Quality Observatory for Open Source Software Project Number: IST-2005-33331. Final Activity Report. Recuperado en: https://cordis.europa.eu/docs/projects/files/033/033331/115240691-6_en.pdf

Belaissaoui, M. and Ait Houaich, Y. (2015). Open Source Software Evaluation Model for Smes. International Journal of Advanced Research Trends in Engineering and Technology (IJARTET), (2), 8.

Bernardo, J., y Calderero, J. (2000). Aprendo a Investigar en Educación. Madrid. España. Rialp.

Buendía, L., Colás; P., y Hernández, F: (Eds.). (1998). Método de investigación en Psicopedagogía. Madrid: McGraw-Hill.

Cardona, M.C. (2002). Introducción a los Métodos de Investigación en Educación. Universidad de Alicante. Madrid: Editorial EOS.

Cetic. (2020). Quality of Open Source Software endeavors. Centre of Excellence in Information and Communication Technologies, Recuperado en: https://www.cetic.be/QualOSS-en

Chacón, W. y Tesisi Tarot, Y. (2004). Modelo de capacidad de madurez del software y su influencia en las mejoras de calidad del software. Universidad de San Carlos de Guatemala Facultad de Ingeniería Escuela de Ingeniería en Ciencias y Sistemas Guatemala, Recuperado en: http://biblioteca.usac.edu.gt/tesis/08/08_0213_CS.pdf

Ciolkowski, M and Soto M. (2008). Towards a Process Maturity Model for Open Source Software. 32nd Annual IEEE International Computer Software and Applications Conference, 1213-1214,

De Freitas, V. (2018). Modelo de madurez en sistema de gestión del conocimiento, desde un enfoque holístico. www.revistanegotium.org.ve / núm. 39 (*13*), 5-31.

Díaz Francisco, J., Banchoff T., C. M., Rodríguez. A. S. and Soria. V. (2009). Evaluación de herramientas Free/Open Source para pruebas de software. Recuperado: de https://www.linti.unlp.edu.ar/uploads/docs/evaluacion_de_herramientas_open_source_para_pruebas_de_software.pdf

Eclipse.org. (2017). Open Source Software for Industry 4.0.p. 1-18. Recuperado de: https://iot.eclipse.org/resources/white-papers/Eclipse%20IoT%20White%20Paper%20-%20Open%20Source%20Software%20for%20Industry%204.0.pdf

ElFadl, .R.A., Salah, A and Kamel, A.(2018). Review on Quality Models for Open Source Software and its reflection on Social Coding. International Research Journal Of Electronics And Computer Engineering, 4(*1*), 1-6.

Etheredge, J. (2009). How Do We Measure Maturity In Software? Recuperado en: https://www.simplethread.com/how-do-we-measure-maturity-in-software/

Glott, R., Groven, A.-K., Haaland, K. and Tannenberg, A. (2010). Quality Models for Free/Libre Open Source Software Towards the "Silver Bullet"? In Proceedings of the 36th EUROMICRO Conference on Software Engineering and Advanced Applications, 439–446.

Infosys Digital. (2016). Open Source Software (OSS) Wave of the Future. Recuperado de: http://www.infosysblogs.com/infosysdigital/2016/06/open_source_software_wave_of_future.html

Kitchenham, B. (2004). Procedures for Performing Systematic Reviews", Joint Technical Report Software Engineering Group, Keele University, United Kingdom and Empirical Software Engineering, National ICT Australia Ltd, Australia.

Kuwata, Y., Takedab, K. and Miurac, H. (2014). A study on maturity model of open source software community to estimate the quality of products. 18th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, (*35*), 1711-1717.

Laila-e, U., Zahoor, A., Mehboob, K. and Natha, S. (2017). Comparison of open source maturity models, Procedia Computer Science, (*111*), 348-354.

Lee, ES. (2012). A Study on Maturity Level for Open Source Software. In: J. (Jong Hyuk) Park J., Leung V., Wang CL. and Shon T. (eds) Future Information Technology, Application, and Service. Lecture Notes in Electrical Engineering, (*164*).

Leister, W. Christophersen, N., Tsiavos, P. and Groven, A-K. (2015). NF5780 Compendium Autumn 2015: Open Source, Open Collaboration and Innovation. Recuperado en: https://www.researchgate.net/publication/283045391_INF5780_Compendium_Autumn_2015_Open_Source_Open_Collaboration_and_Innovation

León, O. G. and Montero, I. (1997). Diseño de Investigaciones. 2da. Ed. Madrid: McGraw-Hill.

Mark, P.; Curtis, B.; Chrissis, M. and Webber, S. (1993). Capability maturity model, version 1.1. IEEE Software, 10(*3*), 7-27.

OSSpal (2020). osspal.org. Recuperado en:: https://www.osspal.org.

Petrinja, E. and Succi, G. (2012). Assessing the Open Source Development Processes Using OMM. Advances in Software Engineering, (*2012*), 17.

Petrinja, E., Nambakam, R. and Sillitti, A. (2009).Introducing the OpenSource Maturity Model. ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, 37-41.

Qualipso (2020). The QualiPSo Open Maturity Model Recuperado en: http://qualipso.icmc.usp.br/OMM/

Raja, U., Wheeler, B. and Rajagopalan, B. (2009). Open Source Software Capability Maturity Model: A Conceptual Framework.. AMCIS 2009 Proceedings, 236.

Samoladas I., Gousios G., Spinellis D. and Stamelos I. (2008). The SQO-OSS Quality Model: Measurement Based Open Source Software Evaluation. In: Russo B., Damiani, E., Hissam, S., Lundell, B. and Succi G. (eds) Open Source Development, Communities and Quality. OSS 2008. IFIP – The International Federation for Information Processing, (275).

SEI, (2010). CMMI® para Desarrollo, Versión 1.3CMMI-DEV, V1.3. TECHNICAL REPORT CMU/SEI-2010-TR-033

Sillitti , A. (2010). A. QualiPSo OMM(Open Maturity Model). Free University of Bolzano, Italy. Recuperado en : http://fossa2010.inrialpes.fr/files/2010/10/QualiPSo-OMM-fOSS2010.pdf

Singh, H.J. and Singh, S.(2015), *Open Source Software-Its Impact on Software Industry in India.* InternatIonal Journal of Computer Science and Technology. 6(3), 79-83.

Soto M. and. Ciolkowski, M. (2009). The QualOSS open source assessment model measuring the performance of open source communities. 3rd International Symposium on Empirical Software Engineering and Measurement, 498-501.

Soto M. and Ciolkowski M. (2009). The QualOSS Process Evaluation: Initial Experiences with Assessing Open Source Processes. In: O'Connor R.V., Baddoo N., Cuadrago Gallego J., Rejas Muslera R., Smolander K., Messnarz R. (eds) Software Process Improvement. EuroSPI. Communications in Computer and Information Science, Springer, Berlin, Heidelberg. (42).

Soto, M. and Ciolkowski, M. (2009). The QualOSS Process Evaluation: Initial Experiences with Assessing Open Source Processes R.V. O'Connor et al. (Eds.): EuroSPI, Springer-Verlag Berlin Heidelberg 2009 (42), 105–116.

Stol, K.-J. and Babar, M. A. (2010). A Comparison Framework for Open Source Software Evaluation Methods. 6th International IFIP WG 2.13 Conference on Open Source Systems,(OSS), 389-394.

Taki, F. Z. and El Alaoui, A.E. B. (2018). Maturity Comparison of open source computer vision software using QSOS Mode. Conference: Colloque International sur le Logiciel Libre dans les Pays de Sud. Recuperado en https://www.researchgate.net/publication/323808364_Maturity_Comparison_of_open_source_computer_vision_software_using_QSOS_Model

Tiwari, V. (2010). Some observations on open source software development on software engineering perspectives. *International Journal of Computer Science & Information Technol ogy*. 2 (6), 13-125. Recuperado de: http://airccse.org/journal/jcsit/1210ijcsit11.pdf .

Toledo Tovar, A., Cuellar Rivera, J. V. and Romero Mera, J. R. (2013). Análisis de Modelos de Evaluación de Calidad de Software Libre. Universidad del Cauca. Facultad de Ingeniería Electrónica y Comunicaciones. Recuperado en: ttps://es.slideshare.net/jorarome/anlisis-de-modelos-de-evaluacin-de-calidad-de-software-libre.

Vaughan-Nichols, S.J. (2015). *It's an open-source world: 78 percent of companies run open-source software*. Recuperado de: https://www.zdnet.com/article/its-an-open-source-world-78-percent-of-companies-run-open-source-software/

Wasserman, A,I., Palm M. and Chan, C. (2006). Business Readiness Rating for Open Source. Proceedings of the EFOSS Workshop, Como, Italy.

Wasserman, A., Palm, M. and Chan, C.(2005) Business Readiness Rating Project, BRR Whitepaper RFC 1, http://www.openbrr.org/wiki/images/d/da/BRR_whitepaper_2005RFC1.pdf

ANEXO

Encuesta

Encuesta aplicada en el presente estudio.

Cuestionario para determinar si los desarrolladores de software, aplican los modelos de madurez de software libre o de código abierto en las organizaciones de desarrollo de software, qué software libre utiliza para el desarrollo de sus aplicaciones y qué aplicaciones desarrollan estas organizaciones con software libre.

Ésta es una Investigación de Maestría. Te agradecemos sea sincero/a. Piensa que no te estamos evaluando a ti, es totalmente anónima.

I. Datos Generales

1. ¿Cuántos años o tiempo de funcionamiento tiene la organización de desarrollo de software? _____.

2. ¿Qué tipo de software desarrolla: escritorio, aplicaciones Web, páginas Web, móvil, otros? _____.

3. Desarrolla software para el mercado:

Nacional □ Internacional □ Ambos mercados □

4. ¿Cuántos años tiene formado el equipo de desarrollo?_____.

5. ¿Qué formación académica tiene el equipo de desarrollo?

Técnico □

Licenciatura □

Maestría □

Doctorado □

6. ¿Los desarrolladores utilizan software libre para el desarrollo de aplicaciones?

Si □ No □

7. ¿El equipo de desarrollo participa de algún proyecto de software libre?

Si □ No □

8 ¿Qué tipo de software desarrolla: escritorio, aplicaciones web, páginas web, móvil, otros?

_____

9. ¿Mencione que software libre utilizan para desarrollar las aplicaciones?

_____.

-II. Nivel de indicadores de madurez.

10. ¿El equipo de desarrollo tiene conocimientos en modelos de madurez de software?

Si □ ¿Por qué si? _____.

No □ ¿Por qué no?_____.

11. ¿La organización implementa modelos de madurez en el desarrollo de software?

Si □ No □

12. ¿Realizan una evaluación del software libre que utilizan para el desarrollo de los productos?

Si □ No □

13. Pregunta 13: La evaluación del software libre lo realiza mediante:

Una metodología □

Una herramienta □

Un modelo de Madurez □

No se realiza evaluación del software □

14. ¿Por qué utiliza software libre para el desarrollo de productos?

_____.

15. ¿Qué indicadores o características importantes considera en la selección de software libre?

_____.

16. ¿Para la selección del software libre es importante su madurez?

Si □ ¿Por qué si? _____.

No □ ¿Por qué no?_____.

17. ¿Conoce o a escuchado de algún modelo de madurez para software libre?

Si □ No □

18. ¿De los siguientes indicadores o características de madurez de software**,** cuáles considera para la selección de software libre?

Licencia □

Comunidad de usuario □

Soporte detención de errores □

Gestión del proyecto □

Aseguramiento de la calidad □

Modificación de código fuente □

Rendimiento □

Usabilidad □

Escalabilidad □

Funcionalidad □

Modo de distribución □

Versión □

Todas la anteriores □

19 ¿Aplica un modelo de madurez de software libre para la selección del software?

Si □ ¿Por qué si? _____.

No □ ¿Por qué no?_____.