ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

Desarrollo de un modelo de calidad externa de software para mejorar la eficiencia y portabilidad del producto final en aplicaciones móviles en la Empresa ADS - Software en el 2020

Vega Molina, Sthalin Paul

Centro de estudios de posgrado

Maestría de Ingeniería en Software

Trabajo de titulación, previo a la obtención del título de magister en ingeniería en software

Ing. Quiña Mera, José Antonio, Msc

Abril, 2021

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y TRANSFERENCIA DE TECNOLOGÍA

CENTRO DE POSGRADOS

CERTIFICACIÓN

Certifico que el trabajo de titulación, "**Desarrollo de un modelo de calidad externa de software para mejorar la eficiencia y portabilidad del producto final en aplicaciones móviles en la Empresa ADS - Software en el 2020**" fue realizado por el señor **Vega Molina, Sthalin Paul** el mismo que ha sido revisado y analizado en su totalidad, por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Latacunga, 31 de marzo del 2021

Firma:

.....

Ing. Quiña Mera, José Antonio Msc.

Director

C.C.: 100232238-4

URKUND

Urkund Analysis Result

Analysed Document:	ProyectoTitulacion_Urkund.pdf
(D100575331)Submitted:	4/4/2021 2:30:00 AM
Submitted By:	paul.milcape@gmail.com
Significance:	5 %

Sources included in the report:

articulo para ponencia en ECE2I-Calidad.docx (D54760845)
https://www.ctr.unican.es/asignaturas/MC_OO/Doc/OO_08_I2_Proceso.pdf
https://estandarsw.wordpress.com/category/iso/iso-9126/
https://silo.tips/download/81-conceptos-de-calidad
https://universidades.cr/carreras/desarrollo-de-software
https://openwebinars.net/blog/react-native-que-es-para-que-sirve/
https://sites.google.com/site/iunesricomputacion/proseso-del-desarrollo-de-software
https://sites.google.com/site/moduloevaluacionred/modelo-de-calidad-boehm
http://www.angelfire.com/my/jimena/ingsoft/calidad.htm
https://ingenieriasoft.webcindario.com/control-de-calidad-del-software/garantia-de-calidad-del-software.html
https://repositorio.pucese.edu.ec/bitstream/123456789/1904/1/BAUTISTA%20ANGULO% 20JENNIFFER%20VER%C3%93NICA.pdf
https://eprints.ucm.es/33446/1/TFG%20Matias%20Sanchez-Carrasco%20Garcia.pdf
http://repositorio.utn.edu.ec/bitstream/123456789/7457/1/PG%20533%20TESIS.pdf
https://docplayer.es/92853527-Escuela-politecnica-nacional.html
https://repositorio.espe.edu.ec/bitstream/21000/5915/1/T-ESPEL-0955.pdf
https://core.ac.uk/download/pdf/296420657.pdf
http://sedici.unlp.edu.ar/bitstream/handle/10915/82406/Documento_completo.pdf-PDFA.pdf%3Fsequence%3D1%26isAllowed%3Dy
Instances where selected sources appear:

39

Firma:

.....

Ing. Quiña Mera, José Antonio Msc.

Director

C.C.: 100232238-4

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y TRANSFERENCIA DE TECNOLOGÍA

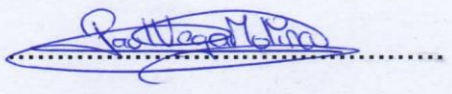
CENTRO DE POSGRADOS

Responsabilidad de auditoría

Yo **Vega Molina, Sthalin Paul**, con cédula de ciudadanía **N° 050331193-8**, declaro que el contenido, ideas y criterios del trabajo de titulación*:* **Desarrollo de un modelo de calidad externa de software para mejorar la eficiencia y portabilidad del producto final en aplicaciones móviles en la Empresa ADS - Software en el 2020** es de mí autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 31 de marzo del 2021

Firma:



Ing. Vega Molina, Sthalin Paul

C.C.: 050331193-8

VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y TRANSFERENCIA DE TECNOLOGÍA

CENTRO DE POSGRADOS

Autorización de publicación

Yo **Vega Molina, Sthalin Paul,** con cédula de ciudadanía **N° 050331193-8**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Desarrollo de un modelo de calidad externa de software para mejorar la eficiencia y portabilidad del producto final en aplicaciones móviles en la Empresa ADS - Software en el 2020** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi/nuestra responsabilidad.

Latacunga, 31 de marzo del 2021

Firma:

.....

Ing. Vega Molina, Sthalin Paul

C.C.: 050331193-8

Dedicatoria

Dedico este trabajo en primer lugar a Dios, por darme fuerzas para seguir adelante y no decaer ante los obstáculos presentados hasta poder culminar uno más de mis sueños.

A mis queridos padres quienes, con su cariño, amor, comprensión y su paciencia, han sido mi apoyo y guía constante en toda mi vida, sin ellos nada de esto hubiera sido posible.

A mi hermano y primos, por ser mis mejores amigos, confidentes y estar a mi lado apoyándome con sus locuras, sonrisas y cariño hasta culminar un objetivo más en mi vida.

A mis tíos, quienes siempre me han hecho reflexionar y ver la vida de una forma distinta sin darme por vencido.

A mis abuelitas, que siempre con sus palabras de aliento no han dejado de apoyarme incondicionalmente.

A mi novia quien, con su amor, cariño, comprensión y paciencia ha estado a mi lado, a pesar de muchos obstáculos sin darme nunca la espalda y estando en los momentos difíciles de la vida.

A mis hijos que están en el cielo viéndome cumplir mis metas.

Agradecimientos

La agradezco a Dios por protegerme, cuidarme y guiarme en el transcurso de la vida, siempre dándome fuerza para seguir adelante.

A mis padres por brindarme su amor incondicional, por no dejar que me rinda nunca, por ser mi fuerza, mi motivación y mi razón de seguir adelante.

A mi novia, por ser siempre comprensiva y estar junto a mí en cada momento de mi vida, brindándome su amor incondicional.

Mi agradecimiento a la empresa ADS – SOFTWARE, por permitirme realizar mi proyecto de titulación en sus instalaciones facilitándome información importante, y principalmente al Ing. Jorge Anchatuña, quien me ha brindado la oportunidad de crecer personal y profesionalmente, siendo mentor y fuente de inspiración para quienes laboramos junto a él.

A la Universidad de las Fuerzas Armadas ESPE-L, por abrirme las puertas de sus instalaciones para de este modo seguirme formando académicamente.

Al Ingeniero Antonio Quiña que, gracias a su apoyo y guía por medio de su experiencia en el desarrollo del proyecto de titulación, me permitió culminarlo.

Al Ingeniero Javier Montaluisa, por darme la oportunidad de seguirme preparando académicamente, y mostrándome su apoyo desde el inicio de la maestría.

Tabla de contenidos

Índice de tablas

Índice de figuras

Resumen

La evaluación de la calidad del producto es una práctica importante en el desarrollo de software, sin embargo, la falta de conocimiento, capacitación e investigación de normas y estándares dirigido a la calidad externa dificulta la evaluación de la calidad en productos finales de software. En el presente proyecto de titulación se desarrolla un modelo de calidad externa implementada en el caso de estudio de la Empresa ADS – Software, el cual servirá como guía e instrumento para evaluar el producto de software. La evaluación de calidad se la realiza por medio de un conjunto estructurado de características, subcaracteristicas, métricas y atributos de calidad específicos para su medición. Para el desarrollo de la investigación se empezó con la construcción del marco teórico y un estudio de los modelos y estándares orientados a evaluar la calidad externa, donde se analiza el modelo de referencia, ciclo de vida de la calidad y estructura de las normas ISO/IEC 25000 que servirán como guía, además la caracterización tecnológica para el desarrollo de la aplicación móvil mediante arquitectura de microservicios, lenguajes de consulta GraphQL y el framework React Native. Siendo validado las características de eficiencia y portabilidad por medio de un análisis comparativo entre dos aplicaciones móviles, una ya existente y otra que se desarrolla por medio de la caracterización tecnológica, siguiendo la metodología que maneja la Empresa ADS – Software.

Palabras clave:

- **MODELO DE CALIDAD**
- **CALIDAD EXTERNA DE SOFTWARE**
- **ISO/IEC 25000**
- **GRAPHQL**

Abstract

Product quality assessment is an important practice in software development. However, the lack of knowledge, training and policies and standards research aimed at external quality makes it difficult to evaluate the quality of final software products. This currently degree project develops an external quality model implemented in the study case of the “ADS - Software Company”, which will be used as a guide and instrument to evaluate the software product. Quality assessment is performed by means of a structured set of metrics characteristics, sub-characteristics and specific quality attributes to be measured. For the development of this research, it began with the construction of the theoretical framework and the study of the models and standards aimed at assessing the external quality, in which is analyzed the reference model, quality life cycle and structure of the ISO/IEC 25000 standards that will serve as a guide. In addition, the technological characterizations for a mobile application development through microservices architecture, GraphQL query languages and the React Native framework. Efficiency and portability characteristics were validated by means of a comparative analysis between two mobile applications, one already existing and another one that is developed through the technological characterization, following the methodology used by ADS - Software.

Keywords:

- **QUALITY MODEL**
- **EXTERNAL SOFTWARE QUALITY**
- **ISO/IEC 25000**
- **GRAPHQL**

CAPÍTULO I

1. Introducción

1.1. Introducción del Capítulo

En este capítulo se describe el inicio del proyecto de investigación, donde se resalta los problemas que ocasiona la falta de conocimiento y estudio de normas y estándares enfocados en la calidad externa del producto de software aplicado en la Empresa ADS – Software. La evaluación de la calidad únicamente se centra en la funcionalidad cuya característica no es suficiente para evaluarla correctamente. Por lo que se propone el desarrollo del modelo de calidad externa de software, mismo que servirá como una herramienta a fin de ejecutar la evaluación de los productos de software. Validándolo mediante el desarrollo de una aplicación móvil aplicando el modelo de calidad externa de software y el proceso de evaluación para las características de eficiencia, portabilidad, subcaracteristicas de comportamiento del tiempo, utilización de recursos, capacidad, instalabilidad, adaptabilidad, atributos y métricas. A fin de verificar la mejoría en la calidad de los productos de software mediante la implementación del modelo propuesto.

1.2. Planteamiento del Problema

La carencia en conocimiento, capacitación e investigación de normas, estándares y criterios hacia la calidad externa por parte de los desarrolladores dificultan la evaluación de eficiencia y portabilidad del producto de software, lo cual ocasiona problemas como: la pérdida de tiempo, retrasos en los proyectos y malestar con los usuarios al momento de su implementación.

En ADS – Software existe varias aplicaciones y una de ellas es el registro de actividades, órdenes de trabajo, solicitud de permisos, vacaciones, viáticos fuera de la oficina, sin embargo, no la utilizan debido a la baja eficiencia y portabilidad tecnológica

al momento de adaptarse a los dispositivos móviles actuales y sus diferentes sistemas operativos.

Ejecutando dicha aplicación se han encontrado problemas tales como: la falta de capacidad al momento de trabajar con un gran volumen de conexiones y transacciones por parte de los usuarios, complejidad en su instalación y su poca adaptabilidad a las diferentes plataformas, también se observó que la evaluación en la empresa, únicamente se centra en la funcionalidad, sin contemplar normas, estándares, características, subcaracteristicas y métricas de calidad externa que se refiere a la totalidad de peculiaridades que contiene el software, las cuales son evaluadas cuando el producto se encuentra en ejecución.

En base a las limitaciones antes descritas del producto software se plantea el siguiente problema: ¿Cómo mejorar la eficiencia y portabilidad del producto final en aplicaciones móviles en la Empresa ADS - Software en el 2020?

1.3. Antecedentes

El desarrollo de software es un área de la década de 1940, ha evolucionado hasta transformarse en la disciplina de la Ingeniería de Software que se ocupa de cómo crear y maximizar su calidad, la misma puede referirse a cuán mantenible es el software, su estabilidad, velocidad, usabilidad, comprobabilidad, legibilidad, tamaño, costo, seguridad y número de fallas, sin embargo, es un tema tratado más recientemente. En los inicios del desarrollo del software, la calidad solamente se centraba en la funcionalidad del producto, a medida que ha evolucionado se han incorporado más características y subcaracteristicas que han permitido realizar la evaluación de estas de forma más optima. Desde hace tiempo la calidad del software ha progresado de tal modo que ha surgido medidas de calidad como también al momento de realizar su medición se ha hecho difícil de forma manual.

Las empresas que desarrollan software en su gran mayoría no utilizan un modelo o estándar para evaluar la calidad de los productos, aun existiendo numerosas propuestas de modelos y estándares. Además, se torna complejo la selección de un determinado modelo o estándar, debido al nicho de mercado que cada organización ofrece a la sociedad (Alena González Reyes, 2016). Empresas como ADS – Software evalúa los productos finales en aplicaciones móviles únicamente centrándose en la funcionalidad, este proceso no es suficiente para valorar la calidad. Si se desea tener una mejor evaluación de la calidad en productos finales es necesario diseñar un modelo de calidad externa.

La propuesta de este modelo de calidad para productos finales en aplicaciones móviles tiene como punto de partida un estudio de los modelos y estándares orientados a evaluar la calidad externa, con el propósito de identificar los aspectos más importantes que caracterizan a cada uno de ellos, entre los que se encuentran: características y subcaracterísticas de calidad, relación de las métricas, tipo de proyecto al que se aplica y clasificación del modelo.

1.4. Objetivo General

Desarrollar un modelo de calidad externa de software para mejorar la eficiencia y portabilidad del producto final en aplicaciones móviles en la Empresa ADS - Software en el 2020.

1.5. Objetivos Específicos

- Establecer un Marco Teórico que fundamente el desarrollo y evaluación del modelo de calidad externa de software.
- Desarrollador el modelo de calidad externa de software.
- Implementar el modelo de calidad externa de software.
- Validar el modelo de calidad externa de software y analizar sus resultados.

1.6. Justificación e Importancia

En la actualidad la Empresa ADS – Software está compuesta por sus diferentes departamentos: comerciales, técnicos, cobranzas y desarrollo, los mismo que usan la tecnología en todo el transcurso de sus actividades, al ser ellos el principal cliente y el principal tester cuando se desarrolla software es más fácil corregir los incidentes que se presentan, sin embargo, se han presentado inconformidades por parte del personal en temas de eficiencia y portabilidad cuando se utilizan las aplicaciones móviles desarrollados. Se hace evidente la necesidad de implementar un modelo de calidad externa de software que se enfoque a evaluar el producto final en tales aplicaciones:

Es ante esta perspectiva, por lo cual se decide realizar el presente proyecto cuyos fines inmediatos consisten tanto en la investigación, diseño e implementación de un modelo de calidad externa de software para el producto final, mismo que será validado mediante un caso práctico.

El presente proyecto contribuirá de primera mano con la evaluación de la calidad externa en las aplicaciones móviles hasta tal punto de mejorar la eficiencia y portabilidad de estas, que actualmente no se las evalúa con ninguna herramienta.

Los beneficiaros del proyecto serán: en primer punto el personal de la empresa y seguidamente todos los clientes que adquieran este tipo de solución tecnológica para satisfacer sus necesidades.

1.7. Hipótesis

Si se desarrolla un modelo de calidad externa de software entonces se mejora la eficiencia y portabilidad del producto final en aplicaciones móviles en la Empresa Ads - Software en el 2020.

1.8. Variable de la Investigación

- **Variable Independiente**

  Desarrollar un modelo de calidad externa de software

- **Conceptualización de la variable independiente**

Un modelo de calidad externa de software es una herramienta que contiene un conjunto de características, subcaracteristicas y medidas, que consideran criterios de calidad externa que permiten evaluar al producto final de software, mediante fórmulas, criterios de aceptación y ponderaciones establecidas.

- **Variable Dependiente**

Mejorar la eficiencia y portabilidad del producto final en aplicaciones móviles en la Empresa Ads - Software en el 2020.

- **Operacionalización de la variable dependiente (Indicadores)**

  Comportamiento del tiempo

  Utilización de recursos

  Capacidad

  Instalabilidad

  Adaptabilidad

CAPÍTULO II

2. Marco teórico

2.1. Introducción

En este capítulo se estructura los siguientes temas; antecedentes históricos que aproximan la comprensión y evolución de los modelos de calidad externa en el desarrollo de software. Antecedentes conceptuales y referenciales determinando la conceptualización del modelo de calidad externa de software. Caracterización tecnológica para el desarrollo de la aplicación móvil utilizando arquitectura de microservicios. GraphQL usada como lenguaje de consultas de datos de API's. React Native usado para desarrollar aplicaciones nativas para IOS y Android. Los antecedentes contextuales justificarán la existencia del problema y la factibilidad del desarrollo e implementación del modelo de calidad externa de software de productos finales en aplicaciones móviles mediante encuestas orientadas al personal de la Empresa ADS – Software.

2.2. Antecedentes Históricos

En esta sección se analizará la evolución de las técnicas de Administración Total de Calidad nombradas (TQM) por sus siglas en inglés Total Quality Management y del Sistema de Administración de Calidad conocidas como (QMS) por sus siglas en ingles Quality Management System, siendo estas técnicas usadas frecuentemente para mejorar la calidad del desarrollo de software. La filosofía TQM plantea que las empresas tengan como objetivo mantenerse en el mercado para satisfacer las necesidades de sus clientes externos, promoviendo la estabilidad de la sociedad y proporcionando la satisfacción y desarrollo a los mismos miembros de la propia organización. Para Hauser y Clausing (1988), el principio más importante es enfocarse en los clientes, con la meta de satisfacer sus expectativas y por medio de esta primicia obtener la calidad total, lo

que refleja al interior de la organización en sus intentos por atender estos requerimientos (Jorge BenzaquenDe las Casas, 2016).

Primera Etapa: Cronológica (1977 - 1988)

Modelo McCall (1977 – 1984)

Es uno de los modelos pioneros en la evaluación de la calidad de software combina once criterios entorno a las operaciones del producto, las revisiones y las transiciones. La idea de McCall es evaluar las relaciones entre los factores de calidad externos y los criterios de calidad del producto mediante: la exactitud, provee el efecto correcto; confiablidad, sin errores; eficiencia, utilización de recursos; integridad, proteger los datos; usabilidad, facilidad de comprensión; mantenibilidad destreza de modificación; testeabilidad, habilidad de ser probada; flexibilidad, quitar funcionalidades; portabilidad, ejecución en varias plataformas; reusabildiad poder emplear métodos e interoperabilidad, facilidad de intercambiar datos (Callejas Cuervo, Alarcón Aldana, & Álvarez Carreño, 2017).

Modelo GQM o Goal Question Metric (1984 – 1986)

Se enfoca en proporcionar una forma que permita definir métricas para medir el avance de los resultados de algún proyecto, a partir de la aplicación de varias preguntas relacionadas con el mismo proyecto, permitiendo alcanzar metas previamente planteadas (Callejas Cuervo, Alarcón Aldana, & Álvarez Carreño, 2017). Se caracteriza por su aplicabilidad en todo el ciclo de vida del producto, procesos, y recursos; alineándose fácilmente con el ambiente organizacional. Puede ser utilizado por lo miembros individuales de un equipo de proyecto, mejorando la calidad, confiabilidad, reduciendo costos, riesgos y mejorando tiempos de ejecución del proyecto (Ladino, Evaluación de la Calidad de la Tecnología Educativa, 2020).

Modelo Boehm (1986 – 1987)

Se fundamenta en que la solución tecnología y/o empresarial debería ejecutar lo cual el cliente desea que realice, por lo tanto, se espera que el software sea adaptable, mantenible, escalable y que utilice los recursos de manera correcta y eficiente, además sea fácil de usar y aprender para los usuarios finales. La estructura presenta tres niveles de características: (RED, 2020).

- Nivel Alto; utilidad, usabilidad, confiabilidad, eficiencia en el producto mantenimiento facilidad de modificarlo, entenderlo y retestearlo (RED, 2020).
- Nivel Intermedio; portabilidad, Fiabilidad, Eficiencia, Usabilidad, Capacidad de prueba, Flexibilidad (RED, 2020).
- Nivel Bajo; portabilidad, independencia de dispositivos, autocontención de confiabilidad, exactitud, completitud, consistencia, robustez integridad (RED, 2020).

Modelo FURPS (1987 – 1988)

El modelo de calidad de software examina cinco propiedades, mismo que toma su nombre por sus siglas en ingles FURPS (funcionalidad, usabilidad, confibalidad, desempeño y soporte). Tiene entre sus ventajas realizar la calidad de un producto asignando prioridades y definiendo atributos que pueden ser medidos (Ladino, Evaluación de la Calidad de la Tecnología Educativa, 2020).

Segunda Etapa: Cronológica (1988 – 1998)

Modelo Gilb (1988 – 1991)

Utiliza el termino de “Atributos” los cuales hacen referencia a la medida de calidad que los sistemas (productos) poseen para ser evaluados. Los atributos definidos

para establecer la calidad de un sistema son los siguientes: capacidad, busca medir la capacidad del sistema para ejecutar tareas; disponibilidad, mide la capacidad del sistema para realizar un trabajo de forma útil; adaptabilidad, hace referencia a la medida de la capacidad que tiene el sistema para sufrir modificaciones; utilizabilidad, mide la facilidad con la cual las personas serán capaces y estarán motivadas para hacer uso del sistema (S/A, Modelos de evaluación de los R.E.D. Grupo 8 UDES W, 2020).

Modelo ISO 9126-1 (1991 – 1997)

Basado en el modelo McCall dirigido a desarrolladores, aseguradores de calidad, evaluadores, analistas y cualquier otro involucrado en el proceso de construcción de software (Callejas Cuervo, Alarcón Aldana, & Álvarez Carreño, 2017). La misma que fue publicada en 1992, a fin de ejecutar el proceso de evaluación de la calidad (GARCIA, 2019).

Se encuentra fraccionado en cuatro partes: el modelo, calidad interna, calidad externa y en uso. Califica la calidad del producto de software en un grupo estructurado de propiedades y subcaracterísticas divididas en: funcionalidad, usabilidad, eficiencia, mantenibilidad y portabilidad (Enrique Morales, 2020).

Modelo SQAE o Software Quality Assessment Exercise (1997 – 1998)

Este modelo, basado en Boehm, McCall, Dromey e ISO 9126, está orientado principalmente a realizar evaluación por terceros que no están directamente involucrado con el desarrollo (Callejas Cuervo, Alarcón Aldana, & Álvarez Carreño, 2017). Es una herramienta que provee un método de evaluación de calidad del software que sea confiable además asociado al riesgo, se enfoca en los diferentes riesgos relacionados con las áreas de calidad que pueden resultar evitables o mitigables haciendo elecciones correctas y acertadas. Trabaja en cuatro áreas básicas: mantenibilidad, evolución,

portabilidad y descripción, presentando siete factores que ayudan a medir la calidad y el servicio de cada concepto fundamental, estos son: consistencia, modularidad, auto descripción, diseño simple, control de anomalías, independencia y documentación (César Augusto López Zapata, 2020).

Tercera Etapa: Cronológica (1998 – Actualidad)

Modelo WebQEM (1998 – 2005)

Es una metodología de evaluación de calidad de sitios web (Web-site Quality Evaluation Method), diseñada para la evaluación siguiendo seis fases: planificación y programación de la evaluación de calidad, definición y especificación de requerimientos de calidad, definición e implementación de la evaluación elemental, definición e implementación de la evaluación global, análisis de resultados, conclusión y documentación, validación de métricas (Callejas Cuervo, Alarcón Aldana, & Álvarez Carreño, 2017).

Modelo ISO 25000 SQuaRE (2005 – Actualidad)

Surge SQuaRe (System Quality Requirements and Evaluation), o bien ISO 25000, que introduce nuevas formas para analizar los conceptos requeridos a la calidad y considera que los objetivos deben estas alineados, con relación a la especificación de todos los requerimientos que demanda un producto software (Estayno, 2012).

Este modelo es la evolución de las normas: 9126, la cual se refiere al modelo de calidad y 14598 que se refiere al proceso de evaluación, esta se categoriza en cinco familias de las normas: 2500n la cual hace referencia a la segmentación de gestión; la 2501n a la segmentación del modelo, la 2502n a la segmentación de la medición, la 2503n la segmentación de requisitos y 25040 a la segmentación de la evaluación todas enfocadas a la calidad (S/A, ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2020).

2.3. Antecedentes Conceptuales y Referenciales

2.3.1. Caracterización del Proceso de Desarrollo de Software

Desarrollo de Software

Es una disciplina que estudia los componentes necesarios para la creación, gestión, mantenimiento y testeo de software computacionales. El software puede entenderse como la programación lógica que todo sistema computacional necesita para funcionar apropiadamente y permitir al usuario disfrutar de aspectos como una interfaz amigable y las funciones que el programa realice. Este concepto se opone al de hardware, que representa todos los componentes físicos de un sistema virtual (Global, 2020).

Proceso de Desarrollo de Software

Es la descripción de una secuencia de actividades que deben ser seguidas por un equipo de trabajo para generar un conjunto coherente de productos de software. El objetivo es la formalización de las actividades relacionadas con el desarrollo del software de un sistema informático (Drake, 2020).

El proceso de desarrollo de software define el qué, quién, cuándo y cómo del desarrollo de software, teniendo cuatro actividades fundamentales que son comunes para todos los procesos de desarrollo de software.

- Especificación del software**;** se debe definir la funcionalidad y restricciones operacionales que debe cumplir el software.
- Diseño e Implementación del software; se diseña y construye el software de acuerdo con la especificación.
- Validación del software; el software debe validarse, para asegurar que cumpla con lo que quiere el cliente.

- Evolución del software; el software debe evolucionar, para adaptarse a las necesidades del cliente.

Además de estas actividades fundamentales, Pressman menciona un conjunto de “actividades protectoras”, que se aplican a lo largo de todo el proceso del software. Las actividades protectoras son: (Morales, Ojeda, Borges, & Rodriguez, 2020).

- Seguimiento y control de proyecto de software
- Revisiones técnicas formales
- Garantía de calidad del software
- Gestión de configuración del software
- Preparación y producción de documentos
- Gestión de reutilización
- Mediciones
- Gestión de riesgos

Garantía de Calidad del Software (SQA)

Es una acción de defensa que se aplica en todo el proceso del desarrollo de software. Antes del siglo veinte, la garantía de calidad era responsabilidad única de la persona que implementaba la solución tecnológica. La primera función de control y de garantía de calidad formal fue introducida por los laboratorios Bell en 1916 y se extendió rápidamente por todo el mundo de las manufacturas. Hoy en día, cada compañía tiene un mecanismo que testifica la calidad, durante la década de los años 90, se usó ampliamente SQA como tácticas de mercado que ponían de manifiesto la calidad ofrecida por las compañías (Diez, 2013).

Garantía de Calidad

Consiste en la auditoria y las funciones de información de la gestión, el objetivo es facilitar la gestión de la información sobre la calidad del producto, por lo que se va adquiriendo un enfoque más profundo y seguro de la calidad. La garantía de calidad del software comprende una gran variedad de tareas, asociadas como dos constitutivos diferentes: los ingenieros de software; que ejecutan un trabajo técnico y un conjunto de SQA por sus siglas en inglés (Software Quality Assurance), que tiene la responsabilidad de la planificación de garantía de calidad (Ferreyra Moreno, 2016).

Control de Calidad

Se basa en las siguientes acciones:

- Uso de métodos y herramientas de análisis, diseño, codificación y prueba.
- Técnicas formales aplicadas durante cada paso de la Ingeniería en Software.
- Táctica de prueba, las cuales completan diferentes escalas.
- Inspección de la documentación y las mejoras realizadas.
- Instrucciones que afirmen un arreglo en la utilización de los estándares de desarrollo.
- Métricas a fin de evaluar y medir la calidad.

2.3.2. Caracterización del Modelo de Calidad Externa

Calidad

La calidad según W. Edwards Deming es traducir las necesidades futuras de los usuarios en características medibles, solo así un producto puede ser diseñado y fabricado para dar satisfacción a un precio que el cliente pagará; la calidad puede estar definida solamente en términos del agente.

Calidad de Software

Hace referencia al nivel de manejo de las propiedades con la que debería llevar a cabo un sistema software a lo largo de su ciclo de vida, las mismas que avalan que el usuario cuente con un sistema confiable, lo que incrementa su gusto ante la funcionalidad y eficiencia que ofrece (Callejas Cuervo, Alarcón Aldana, & Álvarez Carreño, 2017).

Modelos de Calidad de Software

Un modelo de calidad es un grupo de buenas prácticas asociadas a las técnicas de administración y desarrollo de soluciones tecnológicas y/o empresariales. Implicando una idealización a fin de conseguir un efecto estratégico, cumpliendo con las metas propuestas, en cuanto a la calidad del producto o servicio.

La aplicación de modelos de calidad favorece a la mejora continua, establece procesos estándares con insumos y resultados medibles, reduce costos y promueve la eficiencia. Las empresas se ven las beneficiadas al poder ofrecer a sus clientes productos de mayor calidad y seguridad cumplidos en los tiempos previstos (Llaneza, Dapozo, Greiner, & Estayno, 2013).

En el maro del desarrollo de software, los modelos de calidad acceden a realizar los procesos de evaluación de forma cuantitativa o cualitativa, con el objetivo de ayudar a las organizaciones a plantear estrategias a fin de mejorar las etapas de esta (Callejas Cuervo, Alarcón Aldana, & Álvarez Carreño, 2017).

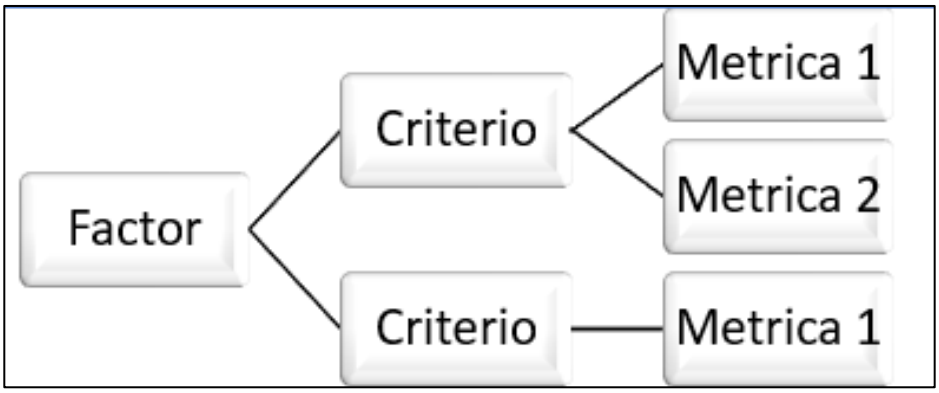
Estructura y Enfoque de los Modelos de Calidad de Software

Se catalogan según el enfoque de evaluación, así sea a grado de procesos, productos o calidad en uso.

Como se muestra en la figura 1; los modelos de calidad de software son estructurados por: diversos factores de calidad que contienen a su vez criterios que a través de las métricas son evaluados, permitiendo la reducción de la subjetividad en la asignación de un valor sean estos cuantitativos y cualitativos.

Figura 1

Estructura de Calidad



Nota. El gráfico representa la estructura general de los modelos de calidad. Tomado de (Callejas Cuervo, Alarcón Aldana, & Álvarez Carreño, 2017).

Calidad del proceso

Debería ser programada a partir del principio del proyecto y después de cada fase del desarrollo de software realizando una inspección y rastreo de los puntos de calidad a fin de reducir los peligros y proporcionar soporte constante; asegurándose de este modo un alto grado de cumplimiento de los componentes de la calidad, teniendo presente que en cada fase se debe ejecutar la verificación de los componentes y criterios con el fin de disminuir los riesgos de una deficiencia y disminución de la calidad (Callejas Cuervo, Alarcón Aldana, & Álvarez Carreño, 2017).

Calidad del producto

El fundamental propósito es evaluar y detallar el cumplimiento de criterios del producto, para lo que se utilizan medidas de calidad externa o interna. Por tal motivo, varias reglas y estándares han determinado la calidad en tres diferentes tipos: externa, interna y en uso, enfocados en comprobar el cumplimiento de las propiedades las mismas que permitan satisfacer los requerimientos establecidos por el usuario en las fases iniciales del desarrollo (Callejas Cuervo, Alarcón Aldana, & Álvarez Carreño, 2017).

Calidad en uso

Es importante resaltar que, aunque en diferentes escenarios se utilizan los términos usabilidad y calidad en uso, con el mismo propósito y de forma intercambiable tienen significados distintos, principalmente porque el concepto de calidad en uso es más amplio y abarca más elementos que la usabilidad y esta última es una de las características de calidad de un producto software. La norma ISO/IEC 9126, la define como un grupo de atributos involucrados con la aprobación por parte del cliente, basándose en la efectividad, productividad, estabilidad y satisfacción (Callejas Cuervo, Alarcón Aldana, & Álvarez Carreño, 2017).

Modelos a Nivel de Productos

La norma ISO/IEC 9126 cataloga la calidad del software en un grupo estructurado de propiedades y subcaracteristicas, siendo estas:

Funcionalidad, es un grupo de atributos relativos a la realidad de un conjunto de funcionalidades y sus características concretas. Las funcionalidades son esas que satisfacen las necesidades del usuario de forma implícita y explícita (Enrique Morales, 2020).

- Idoneidad
- Exactitud
- Interoperabilidad
- Seguridad
- Cumplimiento de normas

Fiabilidad, es un grupo de atributos implicados con la capacidad de conservar su grado de prestación a lo largo de un tiempo (Enrique Morales, 2020).

- Madurez
- Recuperabilidad
- Tolerancia a fallos

Usabilidad, es un grupo de atributos implicados con el esfuerzo a fin de su utilización por un conjunto de usuarios (Enrique Morales, 2020).

- Aprendizaje
- Comprensión
- Operatividad
- Atractividad

Eficiencia, es un grupo de atributos involucrados con el grado del software y la cantidad de recursos (Enrique Morales, 2020).

- Comportamiento en el tiempo
- Comportamiento de recursos

Mantenibilidad, es un grupo de atributos involucrados con la disposición de desarrollar, modificar o corregir errores (Enrique Morales, 2020).

- Estabilidad

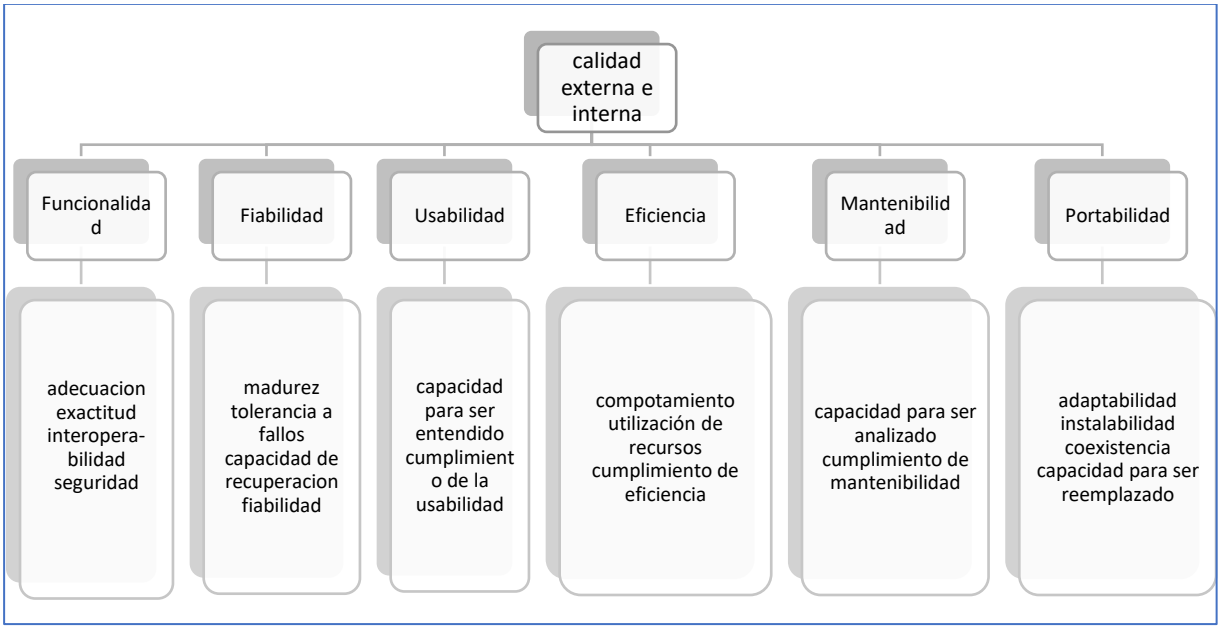
- Facilidad de análisis
- Facilidad de cambio
- Facilidad de pruebas

Portabilidad, es un grupo de atributos involucrados con la capacidad de trasladar una solución tecnológica a diferentes plataformas (Enrique Morales, 2020).

- Capacidad de instalación
- Capacidad de reemplazamiento
- Adaptabilidad
- Co – Existencia

Figura 2

Modelo de Calidad Norma 9126-1



Nota. El gráfico representa la división de la calidad externa e interna de características y subcaracteristicas pertenecientes al modelo de calidad. Tomado de (Enrique Morales, 2020).

El modelo ISO/IEC 25000 denomina SQUARE, proporciona una guía a fin de usar las nuevas series y estándares, denominados Requisitos y Evaluación de Calidad (ISO 25000 calidad de software y datos, 2020).

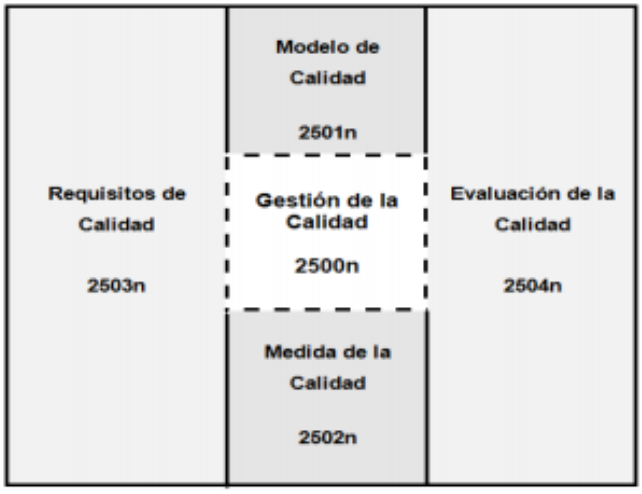
Su primordial objetivo es dirigir el proceso de evaluación de calidad, instituyendo criterios para la explicación de requisitos y sus métricas a fin de ser medidas (ISO 25000 calidad de software y datos, 2020).

División de la Norma ISO/IEC 25000

La norma ISO/IEC 25000 es una familia de normas, como se observa en la figura 3 las cuales están formadas por cinco de ellas, siendo estas:

Figura 3

División de la Norma



Nota. El gráfico representa la división de las normas ISO/IEC de la familia 25000. Tomado de (ISO 25000 calidad de software y datos, 2020).

ISO/IEC 2500n: Gestión de calidad (ISO 25000 calidad de software y datos, 2020)

Establecen clausulas y modelos a fin de implementarlos en lo demás estándares, siendo estos: 25000; estable las actividades y 25001; establece la organización y gestión de estas (ISO 25000 calidad de software y datos, 2020):

ISO/IEC 2501n: Modelo de calidad (ISO 25000 calidad de software y datos, 2020)

Se encuentra catalogadas por las normas: 25010; la cual detalla las características, subcaracteristicas y atributos de la calidad externa, interna y en uso (ISO 25000 calidad de software y datos, 2020).

ISO/IEC 2502n: Medición de calidad (ISO 25000 calidad de software y datos, 2020)

Estable un modelo de referencia y actividades a fin de ejecutar su implementación aplicando las operaciones matemáticas de las métricas con el objetivo de establecer la calidad del software, los mismos que encuentran catalogados es las normas: 25020, que ofrece las actividades a realizar por medio del modelo de referencia; 25021, las primitivas; 25022, establece las métricas de calidad en uso y 25023, la cual establece las métricas de la calidad del producto refiriéndose a la calidad externa e interna (ISO 25000 calidad de software y datos, 2020).

ISO/IEC 2503n: Requisitos de calidad (ISO 25000 calidad de software y datos, 2020)

Se encuentra catalogado por la norma: 25030, la cual facilita la descripción de los requerimientos del producto de software en la etapa de evaluación o en el desarrollo (ISO 25000 calidad de software y datos, 2020).

ISO/IEC 2504n: Evaluación de calidad (ISO 25000 calidad de software y datos, 2020)

Facilita los requerimientos, recomendaciones y actividades a fin de ejecutar la evaluación de los productos de software, implementándolas en roles de: desarrolladores, compradores o evaluadores, misma que contiene las normas: 25040;

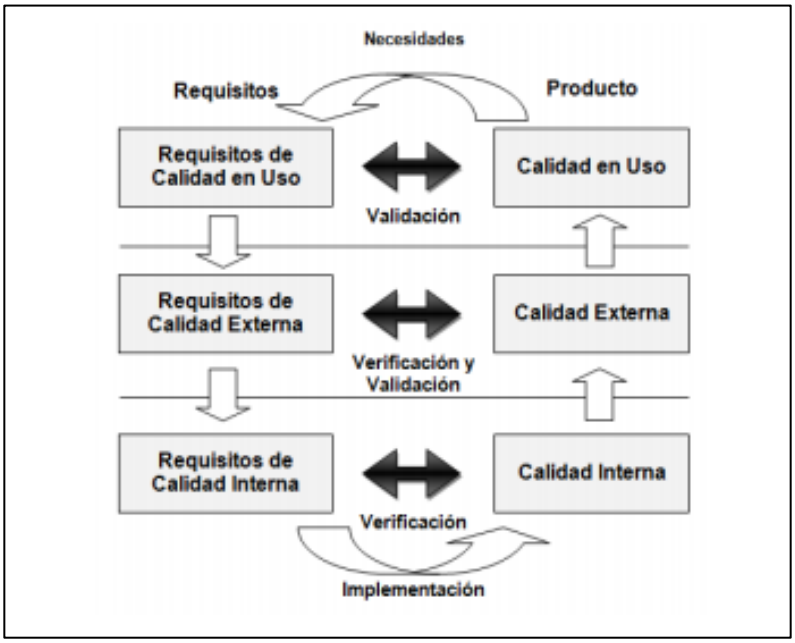
refiriéndose al proceso de evaluación y 25041; refiriéndose a las actividades de evaluación para los diferentes roles (ISO 25000 calidad de software y datos, 2020).

Ciclo de Vida de la Calidad del Producto Software (ISO 25000 calidad de software y datos, 2020)

Demanda un proceso equivalente al proceso de desarrollo de software, como se puede observar en la figura 4, se inicia desde una necesidad, dando como resultado los requerimientos donde en cada proceso se realiza verificación y validación, y finalmente la implementación (ISO 25000 calidad de software y datos, 2020).

Figura 4

Ciclo de Vida de la Calidad del Producto Software



Nota. Tomado de (ISO 25000 calidad de software y datos, 2020).

Como se muestra en la figura 4, esta se maneja en tres etapas del producto, siendo estas:

Interna, en el momento que el producto de software está en el proceso de desarrollo de software.

Externa, en el momento que el producto de software se pone en fase de funcionamiento.

Uso, en el momento que el producto de software es utilizado por el usuario.

Cabe mencionar que cada una de las etapas contiene sus requisitos de calidad, las mismas que manifiestan la necesidad de la solución tecnológica, misma que convendrá ser implementada y posteriormente validada (ISO 25000 calidad de software y datos, 2020). A continuación, se describe los tipos de requisitos:

Los requisitos de calidad en uso detallan el grado de calidad solicitado a partir de la perspectiva del cliente. Como se visualiza en la figura 4, la especificación de requisitos de calidad en uso ayuda a establecer los requisitos de calidad externa (ISO 25000 calidad de software y datos, 2020).

Los requisitos de calidad externa se manipulan a fin de ejecutar la validación y verificación del producto de software, establece los requisitos de calidad interna y conjuntamente anuncia si alcanzará la calidad en uso esperada (ISO 25000 calidad de software y datos, 2020).

Los requisitos de calidad interna se manipulan a fin de verificar la calidad del producto durante las diversas fases del desarrollo, teniendo la posibilidad de conceptualizar tácticas y criterios (ISO 25000 calidad de software y datos, 2020).

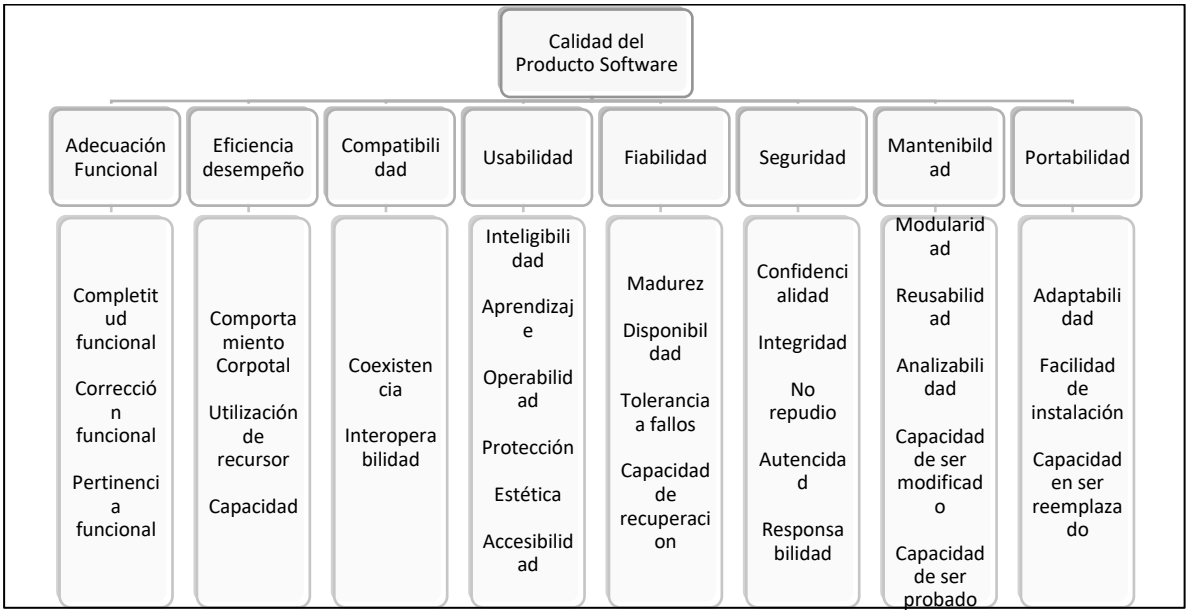
Modelo de calidad del Producto de Softwar ISO/IEC 25010

La norma ISO/IEC 25010 figura como la parte fundamental a fin de ejecutar la evaluación de la calidad del producto de software. Donde se establece las características al momento de valorar las propiedades determinadas (ISO 25000 calidad de software y datos, 2020).

La calidad se puede definir como el resultado por el cual el producto satisface las necesidades del usuario proporcionando un valor. Como se muestra en la figura 5, el modelo de calidad representa las características básicas: adecuación funcional, eficiencia de desempeño, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad y portabilidad (ISO 25000 calidad de software y datos, 2020).

Figura 5

Modelo de Calidad Norma ISO/IEC 25010



Nota. El gráfico representa como se encuentra distribuido la calidad del producto software, en función de características y subcaracteristicas. Tomado de (INEN, Instituto Ecuatoriano de Normalización NTE INEN-ISO/IEC 25000, 2014).

Adecuación Funcional, simboliza la capacidad de la solución tecnológica y/o empresarial a fin de cumplir con los requerimientos del cliente; dividiéndolas en las consecutivas subcaracteristicas (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021):

- Completitud funcional, facilita un grupo de funcionalidades (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).
- Exactitud funcional, facilita los resultados de forma concreta y correcta (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).

Fiabilidad, habilidad de la solución tecnológica a fin de ejecutar funciones determinadas bajo situaciones y lapsos de tiempo (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).

- Madurez, habilidad del producto a fin de complacer las necesidades de fiabilidad (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).
- Disponibilidad, destreza de la solución tecnológica a fin de encontrarse alcanzable y funcional (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).
- Tolerancia a fallo, habilidad de la solución tecnológica a fin de manejar fallos presentados mediante su ejecución (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).

- Recuperabilidad, destreza de la solución tecnológica a fin de recobrar el curso del sistema y restaurar la información en cuestiones de fallos o interrupciones del sistema (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).

Eficiencia en el Desempeño, habilidad del producto de software a fin de facilitar el rendimiento de los recursos bajo situaciones determinadas, dividiéndolas en las subsiguientes (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021):

- Comportamiento Temporal, destreza de un producto de software a fin de facilitar los tiempos de procesamiento y respuesta mostrados mediante su ejecución (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).
- Utilización de Recursos, destreza de un producto de software a fin de manejar las cantidades y los recursos necesarios (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).
- Capacidad, destreza de un producto de software con el fin de efectuar los requisitos (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).

Facilidad de Uso, habilidad del producto de software a fin de ser fácil de usar, aprender y entender (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).

- Capacidad de reconocer su adecuación, destreza del producto de software a fin de comprobar si el sistema cumple con las necesidades del cliente (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).
- Capacidad para ser entendido, destreza del producto de software a fin de evidenciar si el desarrollo consigue obtener los objetivos (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).

- Operativa, destreza del producto de software a fin de permitir al cliente controlarlo de manera fácil y operativa (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).
- Protección contra errores del usuario, destreza del producto de software a fin de resguardarla frente a la manipulación errónea por parte de los clientes (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).
- Estética de la interfaz del usuario, destreza de la interfaz a fin de complacer visual y funcionalmente al cliente (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).
- Accesibilidad técnica, destreza del producto de software a fin de permitir su uso a clientes con alguna discapacidad (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).

Seguridad, habilidad a fin de resguardar la información de modo que solo personal autorizado tenga accesos (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).

- Confidencialidad, destreza a fin de salvaguardar la información y el acceso del personal no autorizado (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).
- Integridad, destreza del producto de software a fin de impedir al personal no autorizado el uso de datos e información (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).
- No repudio, destreza a fin de comprobar eventos suscitados (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).

- Responsabilidad, destreza a fin de llevar la trazabilidad de los eventos que se ejecutaron por cada ente (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).
- Autenticidad, destreza a fin de comprobar la veracidad del personal que ingresa (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).

Compatibilidad, habilidad de varios productos de software a fin de ejecutar sus procedimientos con el objetivo de conmutar información (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).

- Coexistencia, destreza del producto de software a fin de compartir recursos con diferentes programadas cada uno de forma independiente (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).
- Interoperatividad, destreza de varios productos de software con el objetivo de permutar información y monopolizar (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).

Mantenibilidad, habilidad del producto de software con el objetivo de poder ser rectificado y/o mejorado dependiendo los requisitos del cliente (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).

- Modularidad, destreza del producto de software a fin de cuando se mejore las funcionalidades no perturbe a las demás (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).
- Reusabilidad, habilidad de utilizar información, funciones, datos en la construcción de otro producto de software (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).

- Capacidad de ser analizado, habilidad de considerar el impacto de una mejora o reconstrucción en el producto de software (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).
- Capacidad de ser modificado, destreza del producto de software a fin de admitir modificaciones sin producir perjuicios o disminuyendo la calidad (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).
- Capacidad de ser probado, habilidad de ejecutar pruebas unitarias, de componentes o del sistema con el objetivo de cumplir las necesidades del cliente (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).

Portabilidad, habilidad del producto de software a fin de trasladarlo entre varios entornos sin perjudicar su funcionalidad y calidad, la misma que se encuentra dividida en (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021):

- Adaptabilidad, destreza del producto de software a fin de adaptarse a varios entornos (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).
- Capacidad de ser instalado, destreza del producto de software a fin de instalarlo y/o desinstalarlo de forma fácil y rápida (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).
- Capacidad de ser reemplazado, destreza del producto de software a fin de ejecutarlo en vez de otro y desempeñando el mismo rol (25000 I. , ISO 25000 CALIDAD DE SOFTWARE Y DATOS, 2021).

Métrica para la Calidad Interna, Externa y en Uso ISO/IEC 25020 y 25023

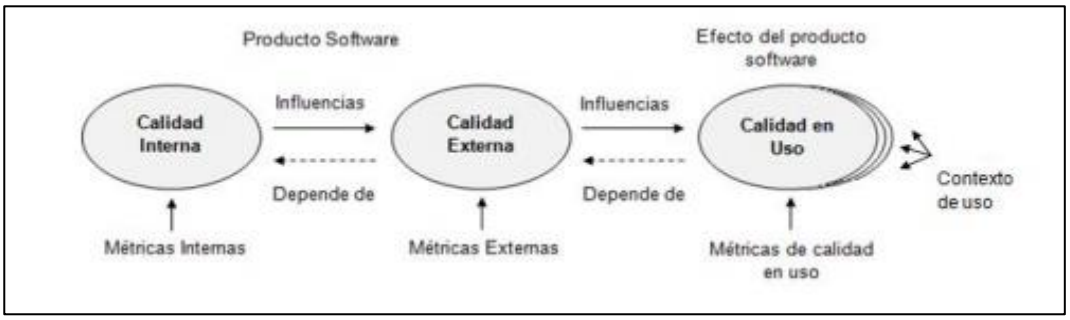
Proporcionan un grupo de métricas, empleadas en el modelo de calidad ISO/IEC 25010 (INTECO, 2020). La norma que concreta el proceso de evaluación de la calidad

del producto es la ISO/IEC 2504n, misma que facilita técnicas a fin de ejecutar la evaluación y valoración.

En la figura 6, se puede evidenciar la interacción existente en medio de las etapas de calidad juntamente con sus métricas y predominación existente entre ellas.

Figura 6

Relación entre los Tipos de Métricas de Calidad



Nota. El gráfico representa las relaciones y sus dependencias de las métricas de calidad: interna, externa y en uso. Tomado de (INTECO, 2020).

La calidad externa se utiliza a fin de calcular el comportamiento del producto de software, siendo empleada en la fase de validación y verificación mientras se encuentra en ejecución (INTECO, 2020).

La calidad interna se emplea en todas las fases del proceso de desarrollo de software con el objetivo de determinar los problemas o errores a fin de tomar medidas correctivas (INTECO, 2020).

La calidad en uso se emplea a fin de comprobar si las necesidades del cliente son las adecuadas, recordando que se las debe realizar en un ambiente real (INTECO, 2020).

Métricas de Calidad del Producto

Las métricas evalúan las características que se definieron en el modelo de calidad del producto de software.

La tabla 1, describe las métricas a fin de realizar el proceso de evaluación de la calidad interna y externa.

Tabla 1

Métricas para la Calidad Interna y Externa

Características	Subcaracterísticas	Métricas
Adecuación	Completitud funcional	Completitud de la implementación funcional.
	Exactitud funcional	Exactitud
		Precisión computacional
Fiabilidad	Madurez	Disipación del fallo
		Suficiencia de las pruebas
		Tiempo medio entre fallos
	Disponibildad	Tiempo de servicio
		Tiempo medio de inactividad
	Tolerancia a fallos	Prevención de fallas
		Redundancia
		Anulación de operación incorrecta
	Recuperabilidad	Tiempo medio de recuperación
Eficiencia en el desempeño	Comportamiento temporal	Tiempo de respuesta
		Tiempo de espera
		Rendimiento
	Utilización de recursos	Líneas de código
		Utilización de CPU
		Utilización de la memoria
		Utilización de los dispositivos de E/S
	Capacidad	Número de peticiones
		Numero de accesos simultáneos

Características	Subcaracterísticas	Métricas
		Sistema de transmisión de ancho de banda
	Capacidad de reconocer su adecuación	Integridad de descripción
		Capacidad de demostración.
	Capacidad de ser entendido	Funciones evidentes
		Efectividad de la documentación del usuario o ayuda del sistema
	Operatividad	Recuperabilidad de error operacional
		Claridad de mensajes
		Consitencia operacional
		Posibilidad de personalización
	Protección contra errores del usuario	Verificación de entradas válidas
Facilidad de Uso		Prevención del uso incorrecto
	Estética de la interfaz del usuario	Personalización de la apariencia de la interfaz del usuario
	Accesibilidad técnica	Accesibilidad física
	Confidencialidad	Capacidad de control de acceso
		Encriptación de datos
	Integridad	Prevención de corrupción de datos
	No repudio	Utilización de firma digital
Seguridad	Responsabilidad	Capacidad de auditoría de acceso
	Autenticidad	Métodos de autenticación
	Co – existencia	Co – existencia disponible
	Interoperatividad	Conectividad con sistemas externos
Compatibilidad		Capacidad de intercambiar de datos
	Modularidad	Capacidad de condensación
		Acoplamiento de clases
	Reusabilidad	Ejecución de reusabilidad
	Capacidad de ser analizado	Capacidad de pistas de auditoria
		Diagnóstico de funciones suficientes
Mantenibilidad	Capacidad de ser modificado	Complejidad ciclomática
		Profundidad de herencia

Características	Subcaracterísticas	Métricas
		Grado de localización de corrección de impacto
		Complejidad de modificación
		Índice de éxito de modificación
	Capacidad de ser probado	Completitud funcional de funciones de pruebas
		Capacidad de prueba autónoma
		Capacidad de reinicio de pruebas
	Adaptabilidad	Adaptabilidad en entorno hardware
		Adaptabilidad en entorno de software
		Adaptabilidad en entorno organizacional
	Capacidad de ser instalado	Eficiencia en el tiempo de instalación
Portabilidad		Facilidad de instalación
	Capacidad de ser reemplazado	Consistencia en la función de soporte al usuario
		Inclusividad funcional
		Uso continuo de datos

Nota: Esta tabla muestra las características, subcaracteristicas y métricas usadas en la calidad interna y externa. (INTECO, 2020).

Modelo de Evaluación de Calidad usando ISO/IEC 25040

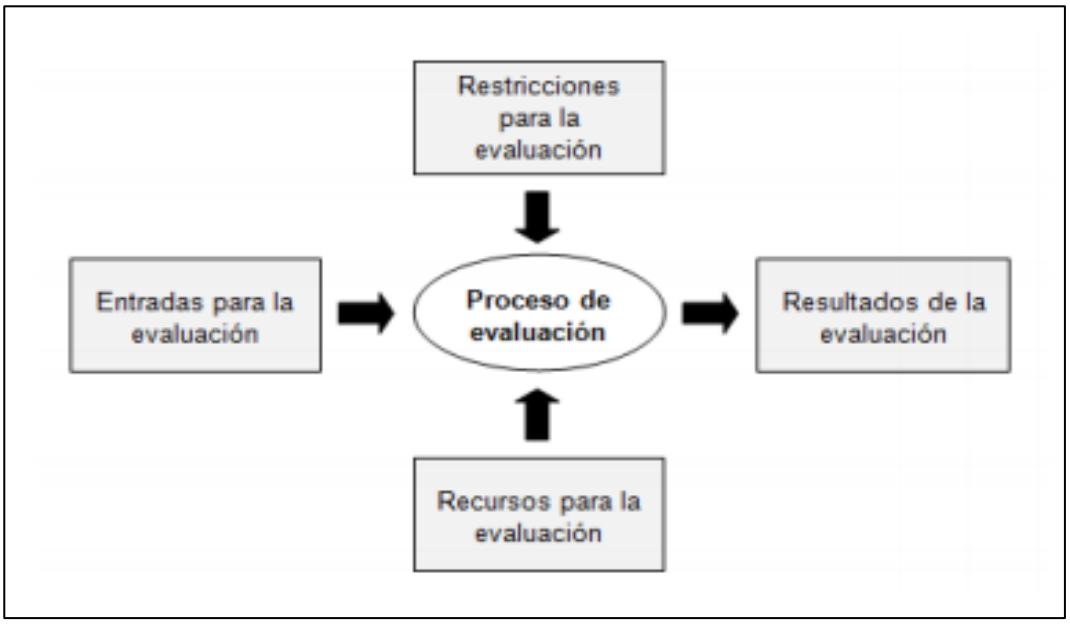
Provee una descripción referente al proceso de evaluación de la calidad e instituye las necesidades, empleándola tanto en la calidad externa, interna y en uso (INEN, INEN Servicio Ecuatoriano de Normalización NT INEN-ISO/IEC25040, 2014).

Modelo de referencia para la Evaluación de Calidad del Producto Software

El modelo que se observa en la figura 7, representa los ingresos, limitaciones, recursos y las consecuencias de emplearlo (INEN, INEN Servicio Ecuatoriano de Normalización NT INEN-ISO/IEC25040, 2014).

Figura 7

Modelo para la Evaluación de la Calidad del Producto Software



Nota. El gráfico representa el modelo de referencia a fin de ejecutar el proceso de evaluación de la calidad. Tomado de (INEN, INEN Servicio Ecuatoriano de Normalización NT INEN-ISO/IEC25040, 2014).

Luego del análisis realizado por los diferentes modelos de calidad y tomando en consideración el contexto del presente estudio, se concluye tomar como referencia la Norma ISO/IEC 25000 debido a que describe las particularidades de un modelo de calidad y aborda el proceso de evaluación a nivel de productos software,

centralizándose en empresas de desarrollo y/o aplicaciones propias permitiendo una mayor eficiencia en la definición del producto.

2.3.3. Caracterización Tecnológica

Arquitectura de Software. Según la IEEE una arquitectura de Software es “La organización fundamental de un sistema incorporado en sus componentes, sus relaciones con los demás y con el ambiente, y los principios que guían su diseño y evolución”. El planteamiento de una arquitectura bien realizada facilita la comprensión a toda persona que tenga contacto con el sistema de gran manera, por lo que se tiene que tomar en cuenta el entorno interno del sistema como el externo (Salazar Hernández, 2017).

Arquitectura de Microservicios. James Lewin y Marting Fowler, ambos expertos en el campo definen una arquitectura de microservicios de la siguiente manera: “El estilo de arquitectura de microservicios es un enfoque para desarrollar una aplicación individual como un conjunto de pequeños servicios, cada uno corriendo su propio proceso y comunicándose con mecanismos livianos”. A menudo la comunicación entre las aplicaciones se realiza por medio de un recurso HTTP (James Lewis, 2014).

Las aplicaciones de microservicios tienen algunas características importantes en común. A continuación, se presenta el listado de estas características presentadas: (Salazar Hernández, 2017).

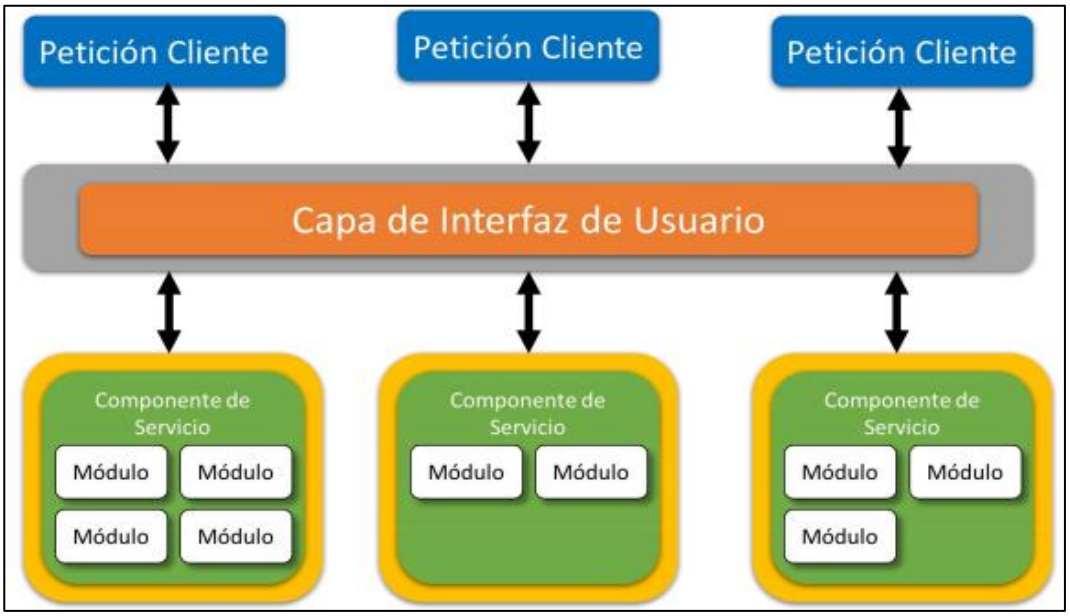
- Pequeñas en tamaño
- Mensajería activada
- Limitado por contextos
- Desarrollo de manera autónoma
- Despliegue independiente

- Descentralizado
- La construcción y el reléase es hecho por procesos automáticos

Una arquitectura de microservicios tiene la característica esencial de ser distribuida, aspecto a tomar en cuenta, ya que se agrega complejidad a su implementación.

Figura 8

Patrón Básico de Arquitectura de Microservicios



Nota. El gráfico representa la arquitectura general de microservicios, dividiendo sus elementos y haciéndolos independientes entre sí. Tomado de (Richards, 2015).

Una de las ventajas de utilizar microservicios es la capacidad de publicar una aplicación grande como un conjunto de pequeñas aplicaciones (microservicios) que se pueden desarrollar, desplegar, escalar, manejar y visualizar de forma independiente. Los microservicios permiten a las empresas gestionar las aplicaciones de código base grande usando una metodología más práctica donde las mejoras incrementales son

ejecutadas por pequeños equipos en bases de código y despliegues independientes. La agilidad, reducción de costes y la escalabilidad granular, traen algunos retos de los sistemas distribuidos y las prácticas de gestión de los equipos de desarrollo que deben ser considerados (López & Maya, 2017).

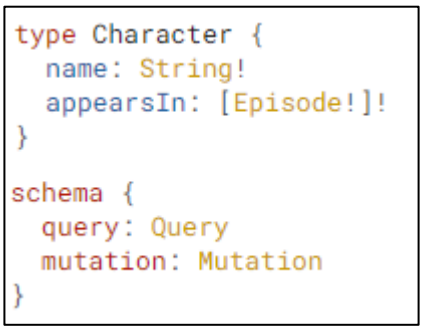
Lenguaje de consulta GraphQL. Es un lenguaje de consulta fuertemente tipado desarrollado por la compañía de medios sociales llamada Facebook en el año 2012 para describir las capacidad y requisitos de los modelos de datos para aplicaciones cliente – servidor (GraphQL, GraphQL Specification Versions, 2018).

El lenguaje fue concebido por varios objetivos generales tales como: reducir la sobrecarga de la transparencia de datos en relación con los modelos de servicio web similares a REST, en términos tanto de la cantidad de datos transferidos innecesariamente, como del número de consultas separadas requeridas para hacerlo. Reducir los errores causados por consultas no válidas por parte del cliente (SOLANO, 2019).

Schema, se define en términos de los tipos y directivas que soporta, así como los tipos de operaciones de: consulta, mutación y suscripción (GraphQL, GraphQL Specification Versions, 2018). Los componentes básicos son los tipos de objetos, que representan un tipo de objeto que puede obtener de sus servicios y los campos que tiene. La Figura 9 muestra la representación de un schema GraphQL (GraphQL, Introduction to GraphQL, 2016).

Figura 9

Definición de esquema GraphQL



```
type Character {
  name: String!
  appearsIn: [Episode!]!
}

schema {
  query: Query
  mutation: Mutation
}
```

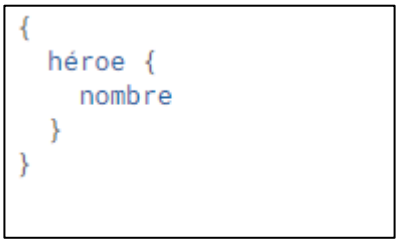
Nota. El gráfico representa la esquematización de la tecnología GraphQL, tomando en cuenta los tipos y directicas que soporta y sus operaciones. Tomado de (GraphQL, Introduction to GraphQL, 2016).

La definición de Schema GraphQL, define en la raíz el elemento de esquematización que se divide en consultas (queries) y Mutaciones (Mutation). Un tipo concreto, como el Usuario, tiene campos a rellenar, en la que un signo de admiración indica que el campo no puede ser nulo. Los corchetes alrededor de la definición indican una matriz del tipo cerrado. Los parámetros como en las consultas y mutaciones se nombran, por lo tanto, se pueden especificar en cualquier orden (SOLANO, 2019).

Queries, se trata de solicitar campos específicos en objetos como se puede observar en la figura 10, y al momento de ejecutarla se puede visualizar que la respuesta tiene exactamente la misma forma, como se muestra en la figura 11. Esto es esencial, ya que siempre se obtiene la estructura de datos que se espera y el servidor sabe lo que solicita el cliente (GraphQL, Introduction to GraphQL, 2016).

Figura 10

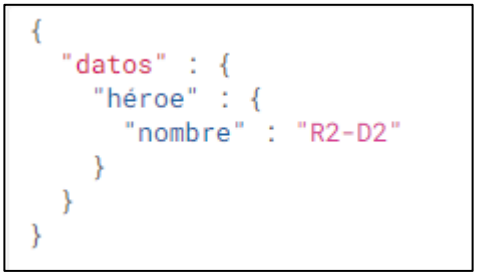
Consulta en GraphQL



Nota. El gráfico representa la esquematización de las consultas en GraphQL. Tomando de (GraphQL, Introduction to GraphQL, 2016).

Figura 11

Resultado de la Consulta en GraphQL



Nota. El gráfico representa los resultados esperados al ejecutar las consultas en GraphQL. Tomado de (GraphQL, Introduction to GraphQL, 2016).

Mutaciones, las mutaciones permiten modificar datos en el servidor GraphQL, las mutaciones devuelven valores igual que las consultas, por lo tanto, un cliente puede consultar datos en función del valor de retorno de una mutación (GraphQL, Introduction to GraphQL, 2016).

Figura 12

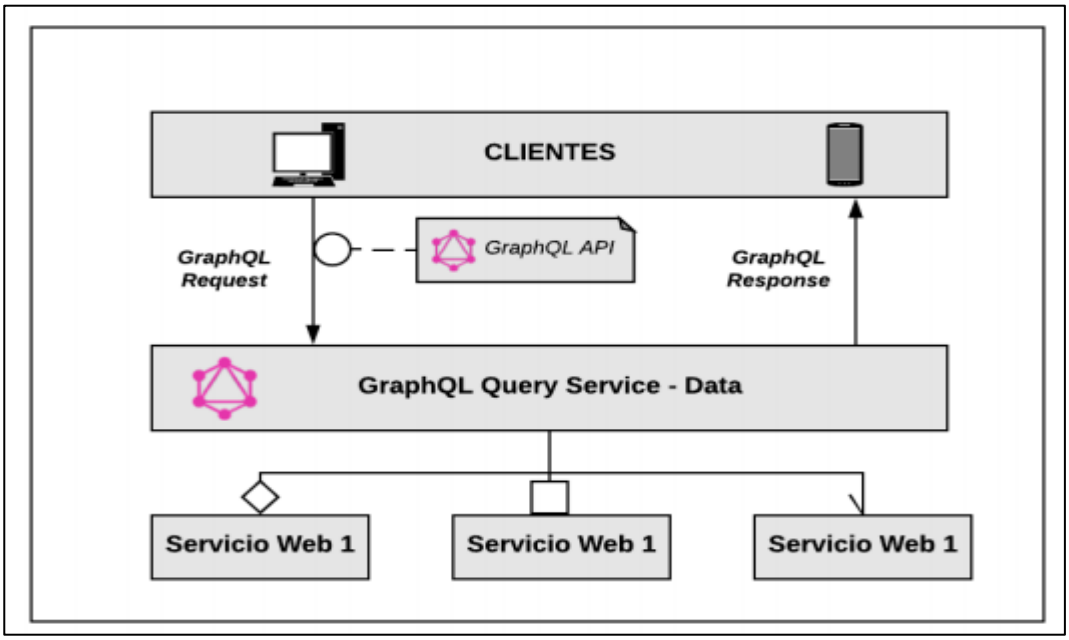
Mutación en GraphQL

Nota. El gráfico representa la esquematización de las mutaciones en GraphQL. Tomado de (GraphQL, Introduction to GraphQL, 2016).

Arquitectura GraphQL. GraphQL requiere una interfaz, que es proporcionada por un servidor, este debe procesar consultas y devolver un conjunto de datos especificado por el cliente, en la figura 13 se muestra que GraphQL tiene como posibilidad usar una puerta de enlace API en la que se puede encapsular el acceso a los sistemas externos.

Figura 13

Descripción Arquitectónica que Utiliza el Servicio de Consulta GraphQL



Nota. El gráfico representa la esquematización de la arquitectura que utiliza GraphQL. Tomando de (SOLANO, 2019).

El servicio de consultas es una instancia que se ejecuta por separado del cliente y puede considerarse como un servicio web. El servidor GraphQL permite que el cliente envíe consultas más potentes de lo que sería posible con otro estándar como SOAP o RESTful.

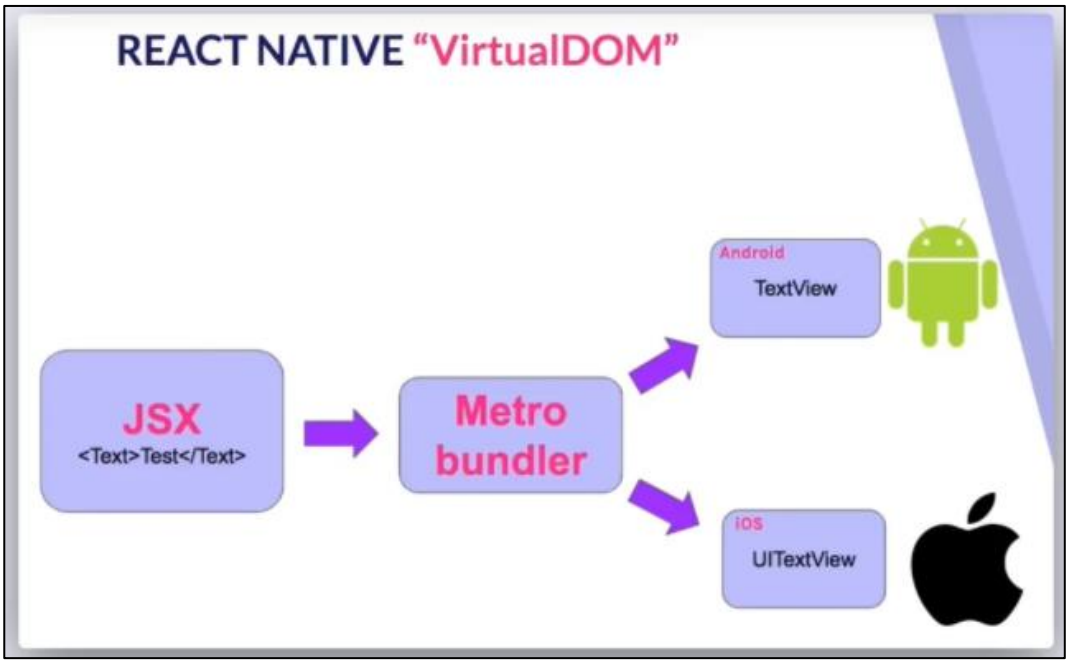
React Native. Es un marco de JavaScript para escribir aplicaciones móviles y naturales para iOS y Android. Está basado en React, la biblioteca de JavaScript de Facebook para construir interfaces de usuario, pero en vez de apuntar al navegador, apunta a las plataformas móviles.

En otras palabras: los desarrolladores web pueden ahora escribir aplicaciones móviles que se vean y se sientan verdaderamente "nativas", todo desde la comodidad

de una biblioteca de JavaScript que ya conocemos y amamos. Además, como la mayoría del código que escribes puede ser compartido entre formas de plataforma, React Native facilita el desarrollo simultáneo para Android y iOS. Las aplicaciones de React Native están escritas usando una mezcla de javaScript y marcado XML, conocido como JSX. Luego, bajo el capó, el "puente" de React Native invoca el APis nativo de renderizado en Objective-C (para iOS) o Java (para Android) (Eisenman, 2016).

Figura 14

Descripción de React Native



Nota. El gráfico representa la arquitectura utilizada al momento de desarrollar aplicaciones en React Native. Tomando de (Martín, 2020).

2.4. Antecedentes Contextuales

El presente proyecto de investigación se desarrollará en el contexto de la industria de software, específicamente en la empresa ADS Software Cía. Ltda. La empresa cuenta con 17 años de experiencia en el mercado nacional e internacional

ofreciendo soluciones informáticas, funcionales y adaptables a los continuos cambios empresariales, para satisfacer las necesidades de los clientes ofreciendo soporte técnico en sistemas operativos, bases de datos, lenguajes de programación, diseño, control y evaluación de proyectos. Cuenta con ingenieros en software, en sistemas e informática con una amplia especialización, permitiendo brindar orientación permanente a sus clientes. El producto de software que brindó reconocimiento a nivel nacional e internacional es el Sistema Administrativo Fénix, siendo una herramienta que soluciona problemas de procesamiento y obtención de resultados del área Contable, Financiera y Tributaria, permitiendo aprovechar de mejor manera el tiempo y los recursos. El sistema abrió las puertas para que la empresa ADS Software extienda sucursales en las ciudades de: Quito, Santo Domingo, Cuenca y Guayaquil.

En ADS – Software se desarrollan varias soluciones tecnologías y empresariales que son de uso interno y externo, siendo una de ellas la aplicación de registro de actividades, órdenes de trabajo, solicitud de permisos, vacaciones, viáticos fuera de la oficina, sin embargo no la utilizan debido a la baja eficiencia, extenso volumen de conexiones y transacciones que genera cada usuario al momento de utilizarla ocasionando pérdidas de tiempo, esto debido a la falta de conocimiento, capacitación y aplicabilidad de normas y estándares de evaluación de la calidad en productos finales. Por tales motivos, se pretende desarrollar un modelo de calidad externa de software que sirva como instrumento de evaluación a los productos finales de software implementándolo por medio de un caso de estudio, para lo cual se elaboraron y aplicaron encuestas orientadas al personal de la Empresa ADS Software. De las cuales se obtuvieron los siguientes resultados:

1.- ¿Con qué frecuencia realiza el registro de sus actividades realizadas dentro o fuera de la empresa?

El 50% del personal tienen una frecuencia de 1 a 3 veces a la semana, el 18,8% del personal tiene una frecuencia de 4 a 6 veces a la semana, mientras que el 31,3% del personal tiene una frecuencia de registro mayor a 6 veces a la semana.

2.- ¿Con qué frecuencia realiza el registro de sus actividades realizadas desde la aplicación móvil?

El 73,3% del personal tiene una frecuencia de 1 a 3 veces a la semana el registro de actividades desde la aplicación móvil, mientras que el 13,3% del personal tiene una frecuencia de 4 a 6 veces y más de 6 veces a la semana.

3.- ¿Ha generado órdenes de trabajo desde la aplicación móvil cuando realiza las tareas fuera de la oficina?

El 25% del personal ha generado por lo menos una orden de trabajo desde la aplicación móvil, mientras que el 75% no o ha realizado.

4.- ¿Las veces que ha utilizado la aplicación móvil de registro de actividades y órdenes de trabajo (Support Control) fueron de su total facilidad de uso?

El 46,7% del personal manifiesta que al utilizar la aplicación móvil fue de su total facilidad de uso, mientras que el 53,3% tuvo dificultades al usarla.

5.- ¿Cómo calificaría la usabilidad de la aplicación móvil de registro de actividades y órdenes de trabajo (Support Control)?

El 18,8% del personal determinaron que la usabilidad de la aplicación móvil es excelente, el 62,5% determinaron que es regular, mientras que el 18,7% determinador que es mala.

6.- ¿Le tomó mucho tiempo realizar el registro de actividades o generar una orden de trabajo desde la aplicación móvil (Support Control)?

El 75% del personal determinaron que les tomo mucho tiempo realizar el registro de actividades o generar una orden de trabajo, mientras que al 25% no le tomo mucho tiempo realizarlo.

7.- ¿Cree usted que la calidad que ofrece la aplicación móvil (Support Control) en términos de eficiencia y portabilidad (instalación de la aplicación) es buena?

El 56,3% del personal está de acuerdo que la calidad que ofrece la aplicación móvil es buena, mientras que el 43,7% no está de acuerdo.

8.- ¿Le gustaría tener una aplicación móvil que sea eficiente, portable y usable al momento de realizar el registro de actividades o generar las órdenes de trabajo?

El 100% del personal les gustaría tener una aplicación móvil eficiente, portable y usable.

9.- ¿Piensa usted que la creación de un modelo de calidad externa de software optimizará la eficiencia y portabilidad del producto final de las aplicaciones móviles?

El 100% del personal determinaron que la creación de un modelo de calidad externa de software optimizará la eficiencia y portabilidad del producto final.

10.- ¿Piensa usted que en el departamento de programación será bien recibido la implementación y ejecución de un modelo de calidad externa de software?

El 87,5% del personal piensa que el modelo de calidad externa de software será bien recibida e implementada en el departamento de programación, mientras que el 12,5% piensa todo lo contrario.

Analizando los resultados de la encuesta aplicada, el 70% del personal de la empresa coinciden que tienen problemas de eficiencia en las aplicaciones móviles, dados especialmente por falta de evaluación de la calidad en términos de eficiencia y portabilidad.

De los resultados obtenidos en la aplicación de la encuesta, el 95,83% del personal de la Empresa ADS Software piensa que un modelo de calidad externa de software mejorará la eficiencia y portabilidad del producto final en aplicaciones móviles.

A la interrogante del desarrollo de un modelo de calidad externa de software para mejorar la eficiencia y portabilidad del producto final en aplicaciones móviles, con los resultados producidos gracias a la encuesta se demuestra que existe la factibilidad.

2.5. Conclusiones del Capítulo

Este capítulo se centró en la fundamentación teórica que servirá para el desarrollo de la investigación. Se dividió en: especificar los antecedentes históricos en el cual se recompila información referente a la evolución de la calidad en el proceso de desarrollo de software. Los antecedentes conceptuales y referenciales, donde se obtiene la conceptualización de los modelos de calidad externa de software en el cual se concluye la utilización de la Norma ISO/IEC 25000 como base referencial para el desarrollo de la propuesta, así como la caracterización tecnológica de su: arquitectura, tecnología y ambiente de desarrollo. Los antecedentes contextuales que justifican la factibilidad del desarrollo e implementación del modelo de calidad externa de software a

fin de ejecutar la evaluación de calidad en productos de software de la Empresa ADS – Software.

CAPÍTULO III

3. Desarrollo del modelo de calidad externa de software

3.1. Introducción

El presente capitulo muestra el desarrollo del modelo de calidad externa de software. Este instrumento servirá como guía para ejecutar la evaluación en los productos software por medio de un conjunto estructurado de características, subcaracteristicas, atributos y métricas de calidad. El modelo de referencia a ser desarrollado contiene la estructura de actividades y áreas cubiertas por las normas, ciclo de vida, características y roles que interactúan en el proceso de evaluación de la calidad del producto de software. El desarrollo del modelo está basado en un análisis de las normas ISO/IEC 25000.

3.2. Análisis de la familia de normas ISO/IEC 25000

En esta sección, se analizará las normas ISO/IEC 25000 también denominadas SQuaRE, tomando como información sus principales componentes: referencia general del modelo de calidad, ciclo de vida, estructura y proceso de evaluación. Con la finalidad de proponer un modelo para la evaluación de la calidad externa del producto de software.

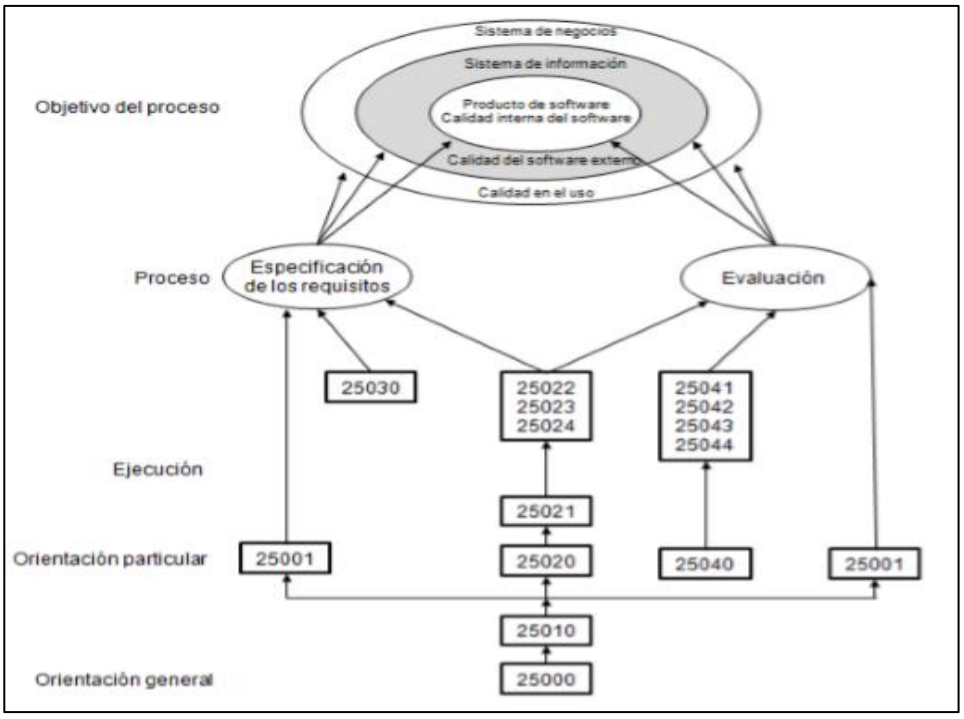
3.2.1. Modelo de referencia general de SQuaRE

El objetivo principal del modelo de referencia es establecer una secuencia del uso de las normas SQuaRE a fin de realizar las actividades de la especificación de los requisitos y evaluación de los sistemas de negocio, sistemas de información y productos de software ver Figura 15. Empezando por la actividad de orientación general donde se incluye la norma 25000 que proporciona una visión rápida sobre las distintas normas y sobre la terminología, orientación particular que contiene la norma 25010 que presenta el modelo de referencia dentro de SQuaRE, las normas 25001, 25020, y 25040 que

describen el soporte necesario para la gestión, la actividad de ejecución la cual utiliza como base lo asimilado en la norma 25020, se establece la 25021 y a su vez las normas 25022,25023 y 25024 las que describen la calidad interna, externa y en uso respectivamente. La norma 25030 detalla los requisitos de calidad, del mismo modo utilizando como base lo aprendido en la norma 25040 se instruye la norma 25041 que especifica los módulos de evaluación, como también las 25042, 25043 y 25044 que describen los procesos de evaluación para desarrolladores, compradores y evaluadores.

Figura 15

Modelo Square de Referencial General



Nota. El gráfico representa una secuencia del uso de las normas para la evaluación de calidad. Tomado de (INEN, Instituto Ecuatoriano de Normalización NTE INEN-ISO/IEC 25000, 2014).

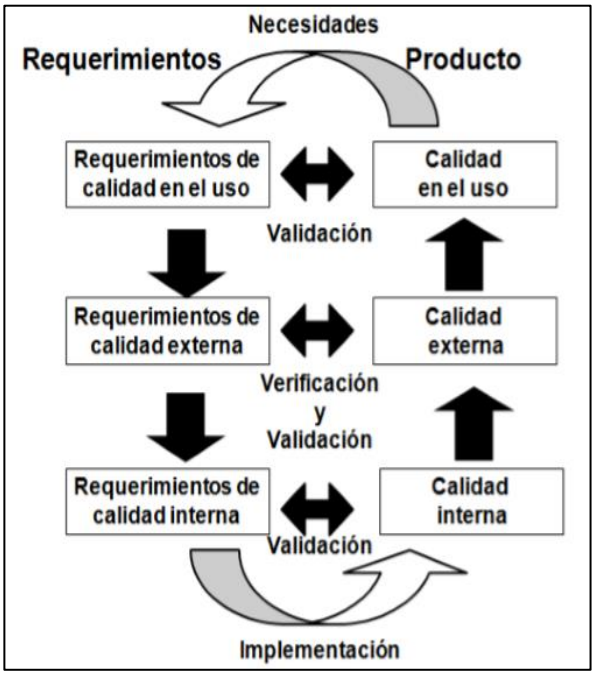
Luego del análisis realizado de las normas SQuaRE y tomando en cuenta el contexto del presente estudio, en la siguiente sección se establecerá un modelo de referencia basado en la evaluación de la calidad externa de los sistemas de información que hasta el momento nos hemos referido como producto de software.

3.2.2. Modelo de ciclo de vida de la calidad del producto de software

El modelo de ciclo de vida enfoca la calidad del producto en tres fases: desarrollo, se refiere a la calidad interna; operación, se refiere a la calidad externa y en uso, se refiere a la calidad propiamente dicha, también consta de tipos de requerimientos: calidad en uso derivados de las necesidades de cada contexto; calidad externa derivados de los requerimientos de la calidad en uso y calidad interna derivados de los requerimientos de la calidad externa, ver Figura 16.

Figura 16

Modelo de Ciclo de Vida de Calidad del Producto de Software.



Nota. El gráfico representa el ciclo de vida de la calidad a nivel de productos. Tomando de (INEN, Instituto Ecuatoriano de Normalización NTE INEN-ISO/IEC 25000, 2014).

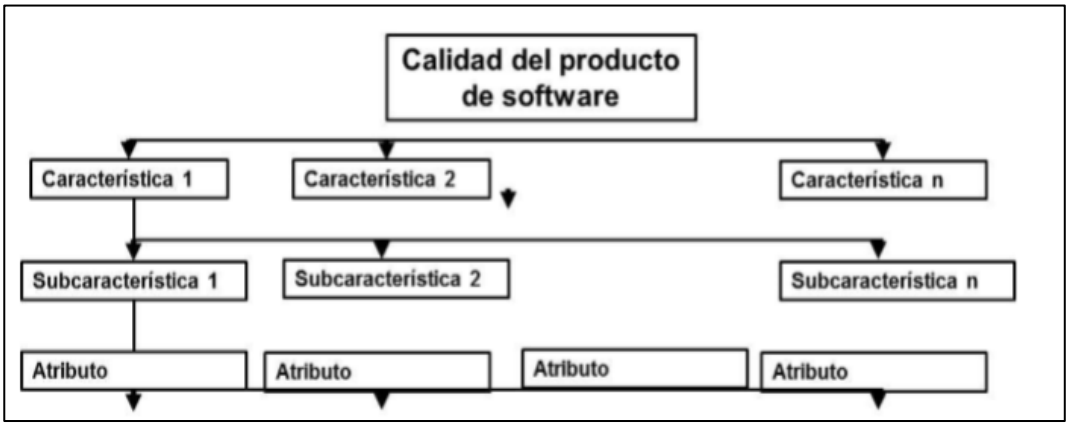
Luego del análisis del modelo de ciclo de vida de calidad del producto de software y tomando en cuenta el contexto del desarrollo del proyecto de investigación, en la siguiente sección se establecerá el ciclo de vida del modelo de calidad externa.

3.2.3. Estructura del modelo de calidad

El modelo de calidad consiste en dos partes, la calidad externa e interna y la calidad en uso, la cual se estructura por características, subcaracteristicas y atributos de calidad que pueden ser medidos de forma cuantitativa o cualitativa, ver Figura 17 (INEN, Instituto Ecuatoriano de Normalización NTE INEN-ISO/IEC 25000, 2014).

Figura 17

Estructura del Modelo de Calidad



Nota. El gráfico representa la estructura general y la distribución de la calidad a nivel del producto. Tomando de (INEN, Instituto Ecuatoriano de Normalización NTE INEN-ISO/IEC 25000, 2014).

Luego del análisis realizado a la estructura del modelo de calidad y tomando en cuenta el contexto del presente estudio, en la siguiente sección se establecerá la estructura y características del modelo de calidad externa.

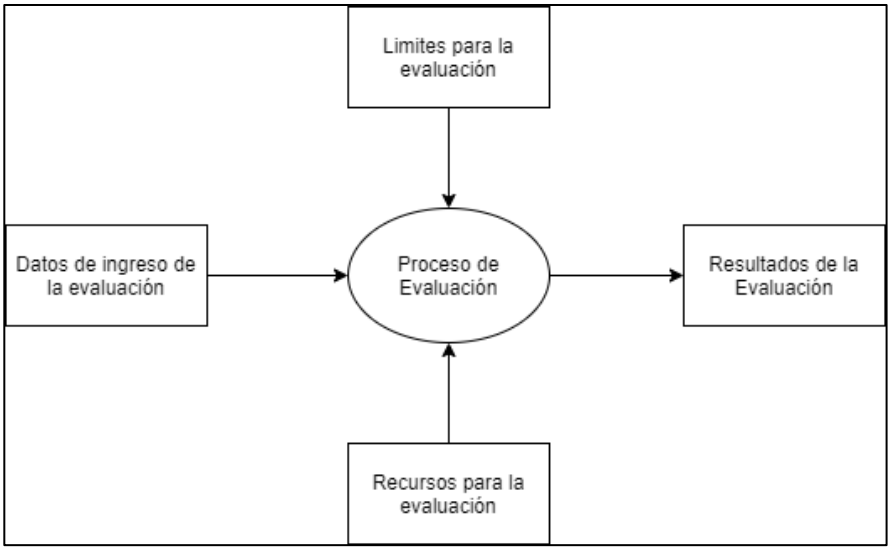
3.2.4. Proceso de evaluación

El proceso de evaluación se conforma de pasos, actividades y tareas que permiten controlar y estabilizar las mismas, estas sirven de guía en la evaluación de la calidad enfocados en tres roles principales: desarrollador, se usa durante el proceso de desarrollo del producto; comprador, se usa cuando se pretende obtener o reusar una solución tecnológica y/o empresarial y evaluador, se usa para evaluar un producto desarrollado o por desarrollarse.

El modelo de referencia general para la evaluación se refiere al proceso, actividades y tareas, que son utilizadas para guiar el proceso de evaluación de la calidad del producto de software, ver Figura 18.

Figura 18

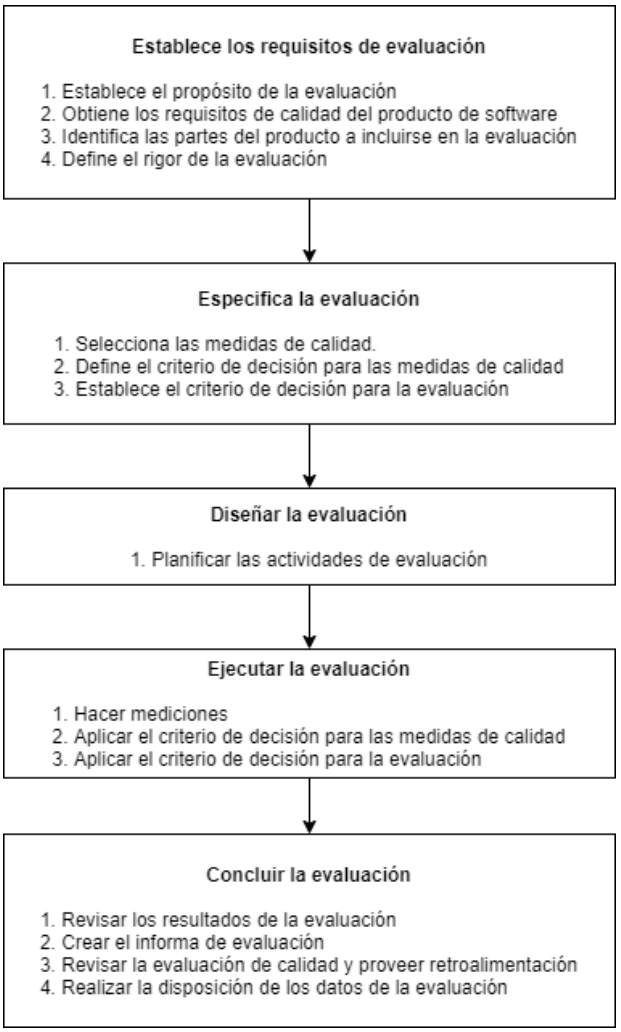
Modelo de Referencia General para la Evaluación



El proceso general para la evaluación de calidad describe de forma detallada los procesos, actividades y tareas que son usadas como guiar para ejecutar el proceso de evaluación de calidad del producto de software, ver Figura 19.

Figura 19

Proceso General para la Evaluación de Calidad



Nota. El gráfico representa el proceso de evaluación de la calidad, con sus pasos y actividades a ejecutarse. Tomando de (INEN, INEN Servicio Ecuatoriano de Normalización NT INEN-ISO/IEC25040, 2014).

Luego del análisis del proceso de evaluación y tomando en cuenta el contexto del presente estudio, en la siguiente sección se establecerá el proceso de evaluación del modelo de calidad externa en situación de evaluador, debido a que se tomará en cuenta la evaluación de calidad externa de producto de software después del desarrollo. La norma ISO/IEC 25000 denominada SQuaRE, servirá de apoyo para realizar la tarea de evaluación en posición de usuario evaluador. La tabla 2 detalla las normas utilizadas para realizar la tarea de evaluación en posición antes mencionada:

Tabla 2

Intersección de Normas para Ejecución de Tareas por Usuarios

Tarea	Usuario	Normas
Evaluación del producto de software después del desarrollo	Evaluador	ISO/IEC 25000 - 25001 25010 - 25020 25030 - 25040 25021 - 25023 25041 - 25044

Nota. Esta tabla muestras las normas utilizadas en el rol de usuario evaluador (INEN, Instituto Ecuatoriano de Normalización NTE INEN-ISO/IEC 25000, 2014).

3.3. Propuesta del modelo de calidad externa de software

Basados en el análisis de la familia de normas ISO/IEC 25000 denominadas SQuaRE y enfocadas desde el rol de evaluador de la calidad externa realizada en la sección 3.2. Se establece un modelo de calidad que servirá como instrumento a fin de ejecutar el proceso de evaluación de los productos de software.

3.3.1. Objetivo del modelo de calidad externa de software

El modelo de calidad externa de software abreviado MCES, servirá como instrumento para llevar a cabo la evaluación de los productos de software; mediante un conjunto estructurado de características y subcaracterísticas, las mismas que se encuentran divididas en atributos y métricas de calidad.

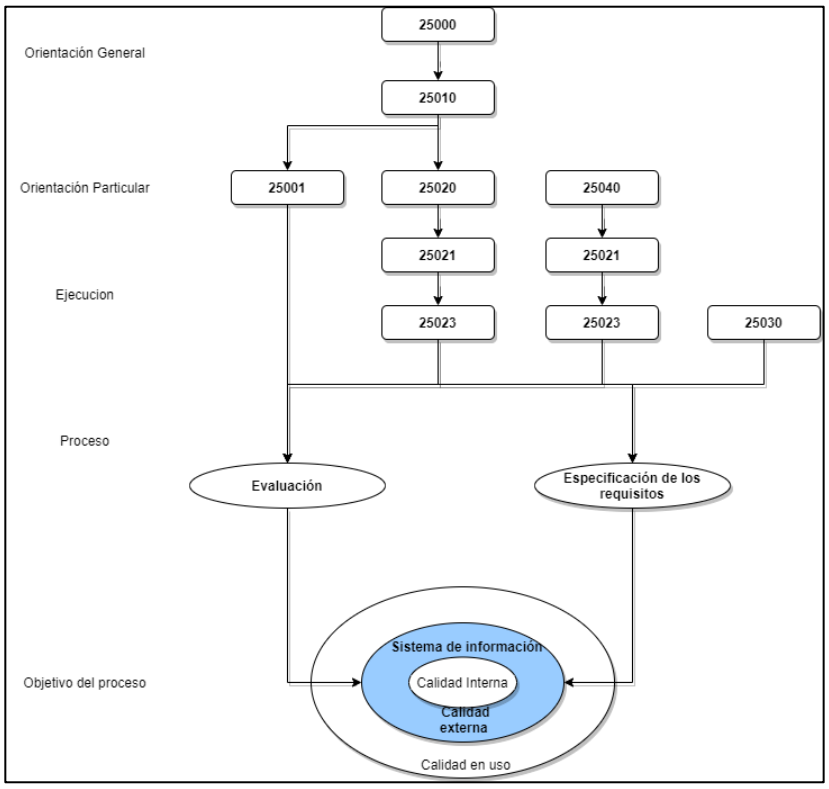
3.3.2. Modelo de referencia de calidad externa

En esta sección proponemos un modelo de referencia de calidad externa en rol de usuario evaluador, considerando las normas: ISO/IEC 25000, 25001, 25010, 25020, 25030, 25040, 25021, 25023, 25041 y 25044 que ofrece el mismo.

La Figura 20 muestra la secuencia de normas y sus actividades que deben ser usadas en el proceso de evaluación. Empezando por la actividad de orientación general la cual incluye la norma 25000; establece una vista rápida sobre las distintas normas y sobre la terminología, la norma 25010 que presenta el modelo de referencia dentro de SQuaRE, las normas 25001, 25020, 25040; describen la gestión necesaria para el soporte, orientación particular utiliza como base lo asimilado en las normas 25020 y 25040, se establece las normas 25021,25023 las que describen la calidad externa, la norma 25030 detalla los requisitos de calidad. Todo este proceso se orienta a la especificación de los requisitos y evaluación en los sistemas de información.

Figura 20

Modelo referencial de calidad externa



El modelo referencial de calidad externa, ver figura 20; tiene como objetivo facilitar el uso de las normas ISO establecidas en el rol de usuario evaluador a fin de realizar de forma rápida, ágil y concreta las actividades de especificación de los requisitos y evaluación enfocándose en la calidad externa de los productos software.

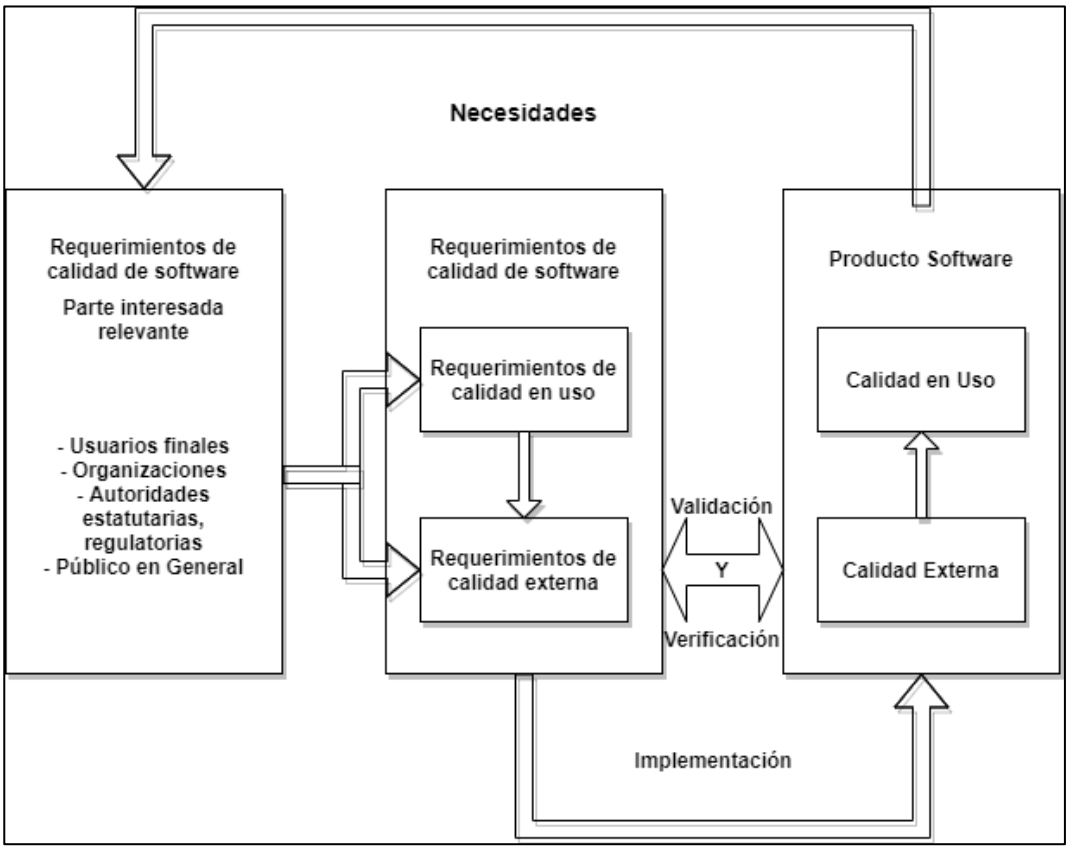
3.3.3. Ciclo de vida del modelo de calidad externa

En esta sección proponemos el ciclo de vida del modelo de calidad externa en rol de usuario evaluador. La figura 21, enfoca la calidad del producto en dos fases: producto en uso, enfocado en la calidad en uso y producto en operación, enfocado en la calidad externa, además muestra dos tipos de requerimientos de calidad de software: requerimientos de calidad en uso cuyo objetivo es la validación del producto de software y requerimientos de calidad externa que se enfoca en la verificación técnica y validación

del producto de software. Hay que considerar también que los requerimientos de calidad de software vienen dados por parte de usuarios finales, organizaciones, autoridades y público en general.

Figura 21

Ciclo de vida de calidad externa



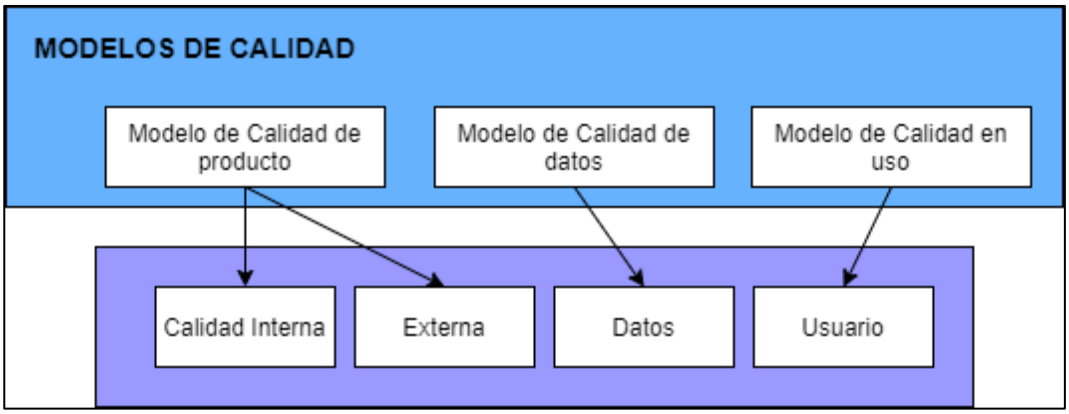
El ciclo de vida de calidad externa propuesto en la figura 21, tiene como objetivo facilitar la implementación de la estructura, procesos, actividades y tareas relacionadas con el modelo de calidad externa a fin de mejorar la calidad del producto software deseado.

3.3.4. Estructura y Características del modelo de calidad externa

En esta sección proponemos la estructura y características del modelo de calidad externa de software en rol de usuario evaluador, la Figura 22 muestra la relación entre las características, subcaracterísticas y propiedades de calidad.

Figura 22

Modelo de Calidad



El modelo de calidad de productos (calidad interna y externa) categoriza las propiedades de calidad de producto en ocho características. La estructura utilizada para el modelo de calidad externa de software se desarrolló con base en el análisis de los atributos que se muestran en la tabla 1; como se observa en la tabla 3.

Tabla 3

Estructura y características del modelo de calidad externa de software

Características	Subcaracterísticas	Métricas
Adecuación	Completitud funcional	Completitud de la implementación funcional.
	Exactitud funcional	Exactitud
		Precisión computacional
	Madurez	Disipación del fallo
		Suficiencia de las pruebas

Características	Subcaracterísticas	Métricas
Fiabilidad	Disponibildad	Tiempo medio entre fallos
		Tiempo de servicio
		Tiempo medio de inactividad
	Tolerancia a fallos	Prevención de fallas
		Redundancia
		Anulación de operación incorrecta
	Recuperabilidad	Tiempo medio de recuperación
Eficiencia en el desempeño	Comportamiento temporal	Tiempo de respuesta
		Tiempo de espera
		Rendimiento
	Utilización de recursos	Líneas de código
		Utilización de CPU
		Utilización de la memoria
		Utilización de los dispositivos de E/S
	Capacidad	Número de peticiones
		Numero de accesos simultáneos
		Sistema de transmisión de ancho de banda
Facilidad de Uso	Capacidad de reconocer su adecuación	Integridad de descripción
		Capacidad de demostración.
	Capacidad de ser entendido	Funciones evidentes
		Efectividad de la documentación del usuario o ayuda del sistema
	Operatividad	Recuperabilidad de error operacional
		Claridad de mensajes
		Consistencia operacional
		Posibilidad de personalización
	Protección contra errores del usuario	Verificación de entradas válidas
		Prevención del uso incorrecto
	Estética de la interfaz del usuario	Personalización de la apariencia de la interfaz del usuario
	Accesibilidad técnica	Accesibilidad física
	Confidencialidad	Capacidad de control de acceso
		Encriptación de datos
	Integridad	Prevención de corrupción de datos
	No repudio	Utilización de firma digital

Características	Subcaracterísticas	Métricas
Seguridad	Responsabilidad	Capacidad de auditoría de acceso
	Autenticidad	Métodos de autenticación
	Co – existencia	Co – existencia disponible
	Interoperatividad	Conectividad con sistemas externos
Compatibilidad		Capacidad de intercambiar de datos
	Modularidad	Capacidad de condensación
		Acoplamiento de clases
	Reusabilidad	Ejecución de reusabilidad
	Capacidad de ser analizado	Capacidad de pistas de auditoria
		Diagnóstico de funciones suficientes
	Capacidad de ser modificado	Complejidad ciclomática
		Profundidad de herencia
Mantenibilidad		Grado de localización de corrección de impacto
		Complejidad de modificación
		Índice de éxito de modificación
	Capacidad de ser probado	Completitud funcional de funciones de pruebas
		Capacidad de prueba autónoma
		Capacidad de reinicio de pruebas
	Adaptabilidad	Adaptabilidad en entorno hardware
		Adaptabilidad en entorno de software
		Adaptabilidad en entorno organizacional
	Capacidad de ser instalado	Eficiencia en el tiempo de instalación
		Facilidad de instalación
Portabilidad	Capacidad de ser reemplazado	Consistencia en la función de soporte al usuario
		Inclusividad funcional
		Uso continuo de datos

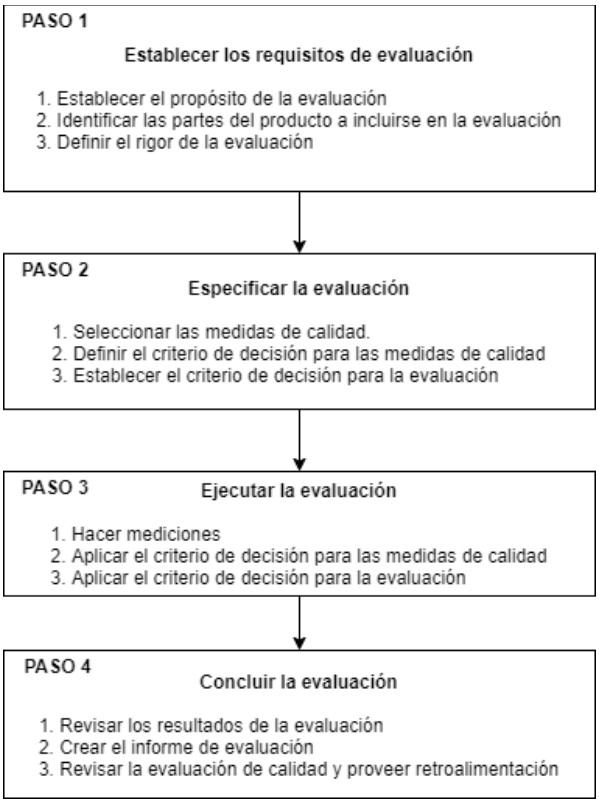
Nota. Esta tabla muestra las caractesticas, subcaracteristicas y métricas que serán utilizadas en el modelo de calidad externa.

3.3.5. Evaluación del modelo de calidad externa de software

En esta sección proponemos el proceso de evaluación del modelo de calidad externa en rol de usuario evaluador. La Figura 23 se refiere al proceso, actividades y tareas, cuyos fines e información complementaria se puede utilizar para guiar el proceso de evaluación de la calidad externa del producto de software (INEN, INEN Servicio Ecuatoriano de Normalización NT INEN-ISO/IEC25040, 2014).

Figura 23

Proceso para la Evaluación de Calidad Externa



El proceso para la evaluación de calidad externa propuesto en la figura 23, tiene como objetivo facilitar la evaluación y ejecución de la calidad en el rol de usuario evaluador a fin de verificar las características, subcaracteristicas y métricas del modelo de calidad externa de software y compararlas con las ponderaciones establecidas en

cada una de ellas. Se segmenta en cuatro pasos, cada uno describe los detalles de las actividades y tareas, cuyos fines e información sirve para guiar la evaluación de calidad externa de software, siendo estos:

Establecer los requisitos

Establecer el propósito de la evaluación, se trata del objetivo o los objetivos que se pretende alcanzar al aplicar el modelo de calidad. Por ejemplo, la aceptación del producto.

Identificar las partes del producto a incluirse en la evaluación, se refiere al tipo de producto intermedio o final a ser evaluado. Por ejemplo, Si se desea seleccionar un producto entre productos alternativos, los productos a ser evaluados son principalmente productos finales o componentes software (INEN, INEN Servicio Ecuatoriano de Normalización NT INEN-ISO/IEC25040, 2014).

Definir el rigor de la evaluación, se define con el propósito de proporcionar confianza en la calidad de los productos, de acuerdo con su uso previsto (INEN, INEN Servicio Ecuatoriano de Normalización NT INEN-ISO/IEC25040, 2014).

Especificar la evaluación

Seleccionar las medidas de calidad, se refiere a la selección de medidas de calidad que se utilizarán para cubrir todos los requerimientos de evaluación.

Definir el criterio de decisión para las medidas de calidad, se definirán de acuerdo con las medidas individuales seleccionadas, que servirán para determinar la necesidad de una acción (INEN, INEN Servicio Ecuatoriano de Normalización NT INEN-ISO/IEC25040, 2014).

Establecer el criterio de decisión para la evaluación, se debe prepara un procedimiento a futuro, con criterios separados por características, cada uno de los cuales puede ser en términos de subcaracteristicas individuales y medidas de calidad (INEN, INEN Servicio Ecuatoriano de Normalización NT INEN-ISO/IEC25040, 2014).

Ejecutar la evaluación

Hacer mediciones, se aplicarán al producto de software y componentes (INEN, INEN Servicio Ecuatoriano de Normalización NT INEN-ISO/IEC25040, 2014).

Aplicar el criterio de decisión para las medidas de calidad, se aplicarán a los valores medidos (INEN, INEN Servicio Ecuatoriano de Normalización NT INEN-ISO/IEC25040, 2014).

Aplicar el criterio de decisión para la evaluación, se refiere a una declaración del grado en que le producto de software cumple con los requerimientos de calidad (INEN, INEN Servicio Ecuatoriano de Normalización NT INEN-ISO/IEC25040, 2014).

Concluir la evaluación

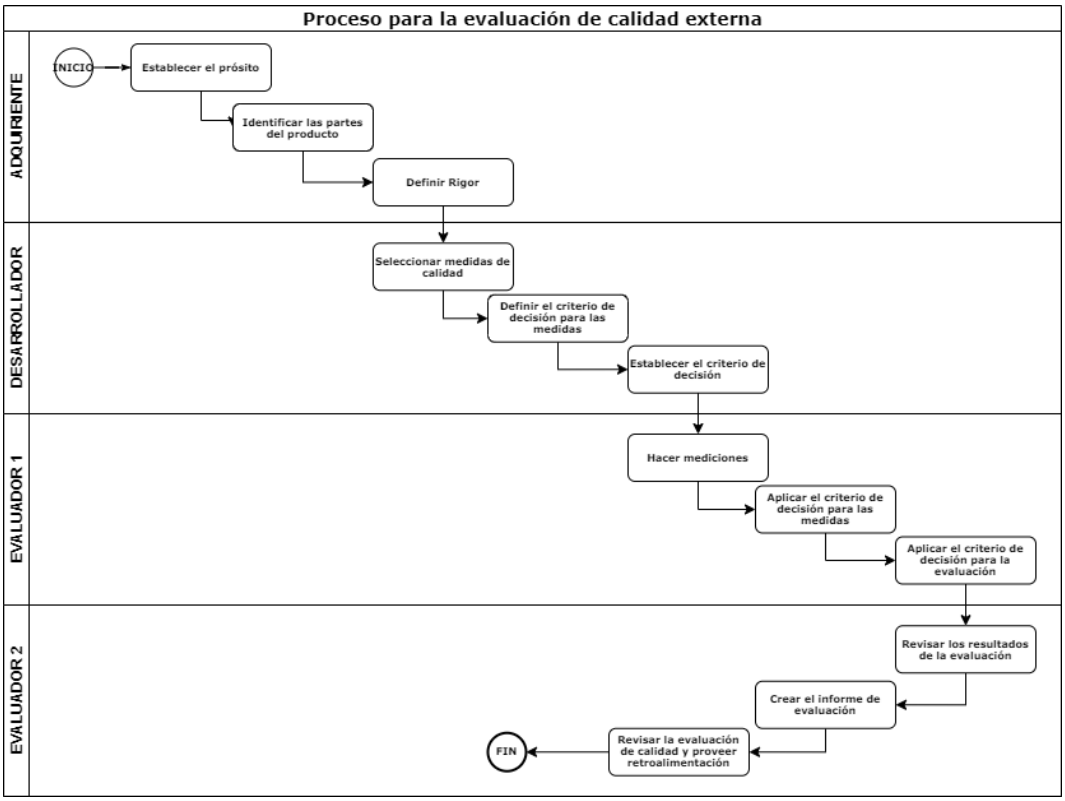
Revisar los resultados de la evaluación, se realizará una revisión conjunta de los resultados de la evaluación (INEN, INEN Servicio Ecuatoriano de Normalización NT INEN-ISO/IEC25040, 2014).

Crear el informe de evaluación, se realizará el detalle de los puntos implementados anteriormente.

Revisar la evaluación de calidad y proveer retroalimentación, se revisará los resultados de la evaluación y la validez del proceso de evaluación.

Figura 24

Diagrama de proceso para evaluación de calidad



3.3.6. Modelo de Calidad Externa de Software del Caso de Estudio

El modelo de calidad externa de software se desarrolló juntamente con el jefe del departamento de I+D (investigación – desarrollo), debido a la falta de conocimiento, capacitación e investigación de normas, estándares y criterios enfocados en la calidad externa. En la empresa ADS – Software, se ejecuta la evaluación de los productos de software en aplicaciones móviles centrándose en la característica de funcionalidad, esta no es suficiente para evaluar de forma clara y concreta la calidad del producto. Al finalizar la propuesta del modelo de calidad externa de software, se puede evidenciar el modelo desarrollado para las características de eficiencia y portabilidad, las mismas que serán aplicadas al momento de comparar dos aplicaciones móviles. La figura 4 muestra las características, subcaracteristicas, métricas, nivel de recomendación que consta de

su clasificación: muy recomendable [MR] (utilizar la medida de calidad siempre), recomendable [R] (utilizar la medida de calidad cuando sea apropiado) y dicresión del usuario [DU] (utilizar la medida de calidad como referencia), porcentaje asignado por característica; en la cual se tomó en cuanta la clasificación dada anteriormente para colocar sus ponderaciones y del mismo modo el porcentaje determinado por subcaracteristica.

Tabla 4

Modelo de Calidad Externa de Software

Características	**Subcaracteristicas**	**Metricas**	**% Carac**	**% Subcar.**
Eficiencia	Comportamiento temporal	Tiempo de Respuesta	87	50
	Utilización de recursos	Utilización CPU		12
	Capacidad	Número de peticiones		25
Portabilidad	Adaptabilidad	Adaptabilidad entorno hardware	13	10
	Capacidad de ser instalado	Facilidad de instalación		3
Total			100	100

3.4. Conclusiones del Capítulo

En el presente capítulo se desarrolló la propuesta del modelo de calidad externa de software, mismo que contiene su modelo de referencia en el cual consta las actividades y áreas cubiertas por las normas utilizadas en el rol de usuario evaluador,

ciclo de vida y características que intervienen en el proceso de evaluación de la calidad del producto de software.

CAPÍTULO IV

4. Medición del modelo de calidad externa de software

4.1. Introducción

En este capítulo se presenta la medición del modelo de calidad externa que se desarrolló en el capítulo anterior, juntamente con la implementación y comparación de una aplicación móvil desarrollada previamente denominada “Support Control” y la nueva aplicación llamada “Activities”. La medición del modelo se realizó a través de los métodos empíricos: experimento y encuesta.

El experimento permite controlar la situación y manipular el comportamiento del mismo, enfocándonos en sus pasos de: alcance, planificación, operación y análisis e interpretación, este tiene como objetivo comparar la evaluación de eficiencia de dos aplicaciones móviles desarrolladas con arquitecturas y tecnologías diferentes, basadas en las características de eficiencia y subcaracteristicas de comportamiento temporal, utilización de recursos y capacidad establecidas en el modelo de calidad externa.

La encuesta permite evaluar la característica de portabilidad y subcaracteristicas de capacidad de ser instalado y adaptabilidad al entorno hardware, esta consistió en realizar la instalación de dos aplicaciones móviles en una muestra de 17 personas durando un máximo de cinco minutos.

Al finalizar la medición de cada característica y subcaracteristica, se obtienen valores para cada una de ellas en el rango de 0 a 1, generalmente entre más se acerque al valor 1 mucho mejor es.

4.2. Desarrollar la aplicación móvil de registro de actividades

4.2.1. Análisis

En esta fase se define la metodología a ser aplicada en el desarrollo de la aplicación, la misma que se usa en la empresa ADS – Software siendo una combinación de la metodología XP y Kanban.

4.2.1.1. Levantamiento. El levantamiento de los requisitos está definido por tarjetas front of card (anverso de la tarjeta), donde se escribe las necesidades del cliente de forma no técnica y back of card (reverso de la tarjeta) donde se colocan las validaciones necesarias escritas en lenguaje técnico. Estas se utilizan cuando existe un primer acercamiento con el cliente y/o usuario que requiera la necesidad de automatizar sus procesos, siendo estas de forma presencial, vía llamada telefónica o video conferencia.

Figura 25

Tarjeta Front y Back of Card

FRONT OF CARD	BACK OF CARD
Quiero logearme con mis credenciales de usuario y contraseña proporcionadas y que me muestre únicamente las	1. Se necesita crear un token de autenticidad 2. Se necesita validar a que opciones del menú tiene acceso el usuario y mostrárselas por medio de su perfil de usuario.

4.2.1.2. Definición de Requisitos. Los requisitos están definidos por historias de usuario de acuerdo con la metodología de ADS, las cuales se han realizado a partir de las tarjetas front y back of card, adquiridas al momento de realizar el levantamiento, las mismas que son estimadas por medio del planning poker.

Tabla 5

Historia de Usuario 1

Historias de Usuario 1	
Número: 1	Usuario: Administradores, Personal de la empresa
Nombre Historia: Acceso a la Aplicación	
Prioridad: Alta	Riesgo en Desarrollo: Media
Puntos Estimados: 13	Iteración Asignada: 1
Programador Responsable: Paul Vega	
Descripción: Los tipos de usuarios de la aplicación tendrán un nombre de usuario clave única y token con la que podrán ingresar, se les generará su menú dependiendo del perfil de usuario.	
Observaciones: Solo los usuarios que estén definidos en el sistema tendrán accesos a sus funcionalidades.	
Criterios de Aceptación:	
Debe ser capaz de reestablecer su contraseña por medio de nombre de usuario.	

Tabla 6

Historia de Usuario 2

Historia de Usuario 2	
Número: 2	Usuario: Administradores, Personal de la empresa
Nombre Historia: Visualización de agenda	
Prioridad: Media	Riesgo en Desarrollo: Media

Puntos Estimados: 8	Iteración Asignada: 1
Programador Responsable: Paul Vega	
Descripción: Los usuarios podrán observar su agenda por medio de una visualización calendarizada por mes.	
Observaciones: Se podrá observar las actividades que cada uno tiene asignado.	
Criterios de Aceptación:	
Debe ser capaz de visualizar las actividades por medio de un calendario de forma organizada.	

4.2.1.3. Release Planning. A partir de las historias de usuario y su estimación realizada, se les asigna un orden para el desarrollo, como se muestra en la tabla6.

Tabla 7

Release Planning

Historias	Iteración	Prioridad	Esfuerzo	Fecha Inicio	Fecha Fin
Historia 1	1	Alta	13	28/09/2020	29/09/2020
Historia 2	1	Media	8	30/09/2020	01/10/2020
Historia 3	1	Media	13	02/10/2020	03/10/2020
Historia 4	1	Alta	13	04/10/2020	05/10/2020
Historia 5	1	Media	5	06/10/2020	07/10/2020
Historia 6	1	Alta	13	08/10/2020	09/10/2020
Historia 7	1	Media	5	10/10/2020	11/10/2020
Historia 8	1	Media	8	12/10/2020	13/10/2020

4.2.2. Diseño

El diseño del proyecto se lo realizó en una iteración de 3 semanas, donde se definió la arquitectura tecnológica, y un diagrama del proceso de la aplicación.

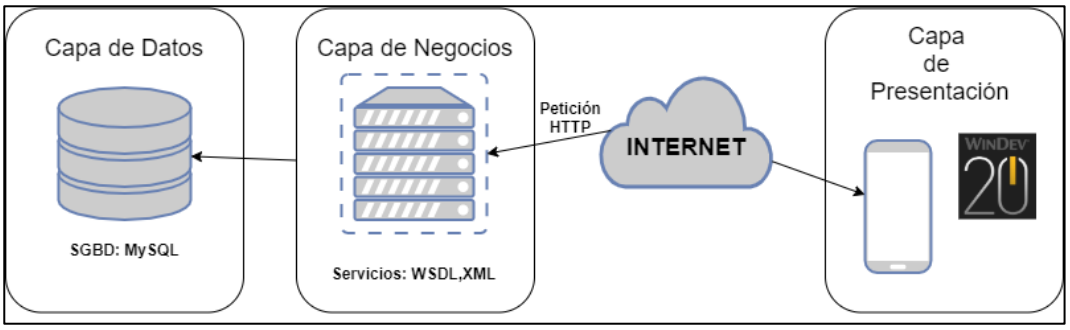
4.2.2.1. Arquitectura Tecnológica. El presente proyecto consiste en evaluar dos arquitecturas de dos aplicaciones móviles.

Arquitectura Tecnológica APP Support Control

La aplicación móvil se desarrolló utilizando una arquitectura de servicios web. En lado del servidor se encuentra las tecnologías: WSDL y la base de datos MySQL; y en el cliente las tecnologías: Windev, como se visualiza en la Figura 26.

Figura 26

Arquitectura APP Support Control

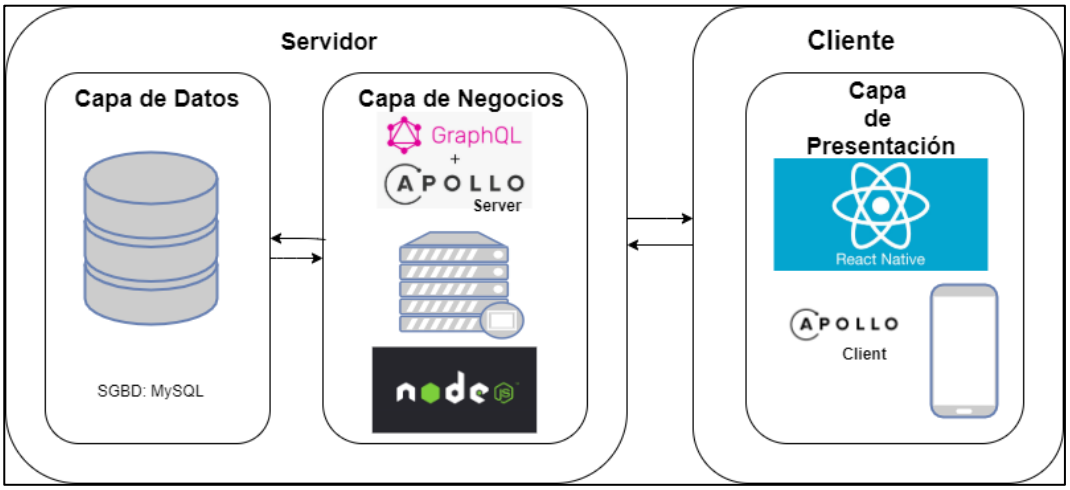


Arquitectura Tecnológica APP Activities

La aplicación móvil se va a desarrollar utilizando una arquitectura orientada a microservicios en un entorno de ejecución Node.js. En el lado del servidor se encuentra la Tecnología: Express, Graphql y la base de datos MySQL; y en el cliente las tecnologías: Apollo Client y React Native, como se muestra en la Figura 27.

Figura 27

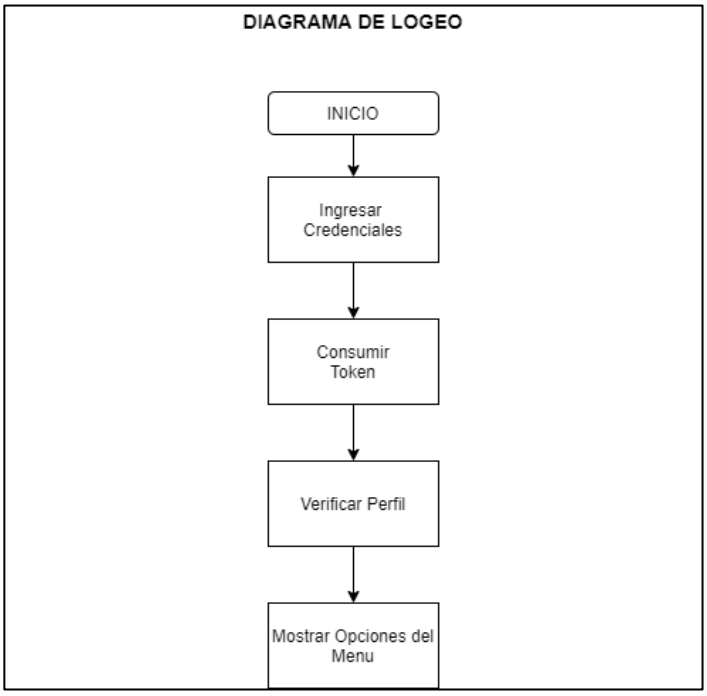
Arquitectura APP Activities



4.2.2.2. ***Diagrama de Proceso.*** En la Figura 28, se muestra el diagrama de proceso para el logeo de la aplicación móvil.

Figura 28

Diagrama de Logeo



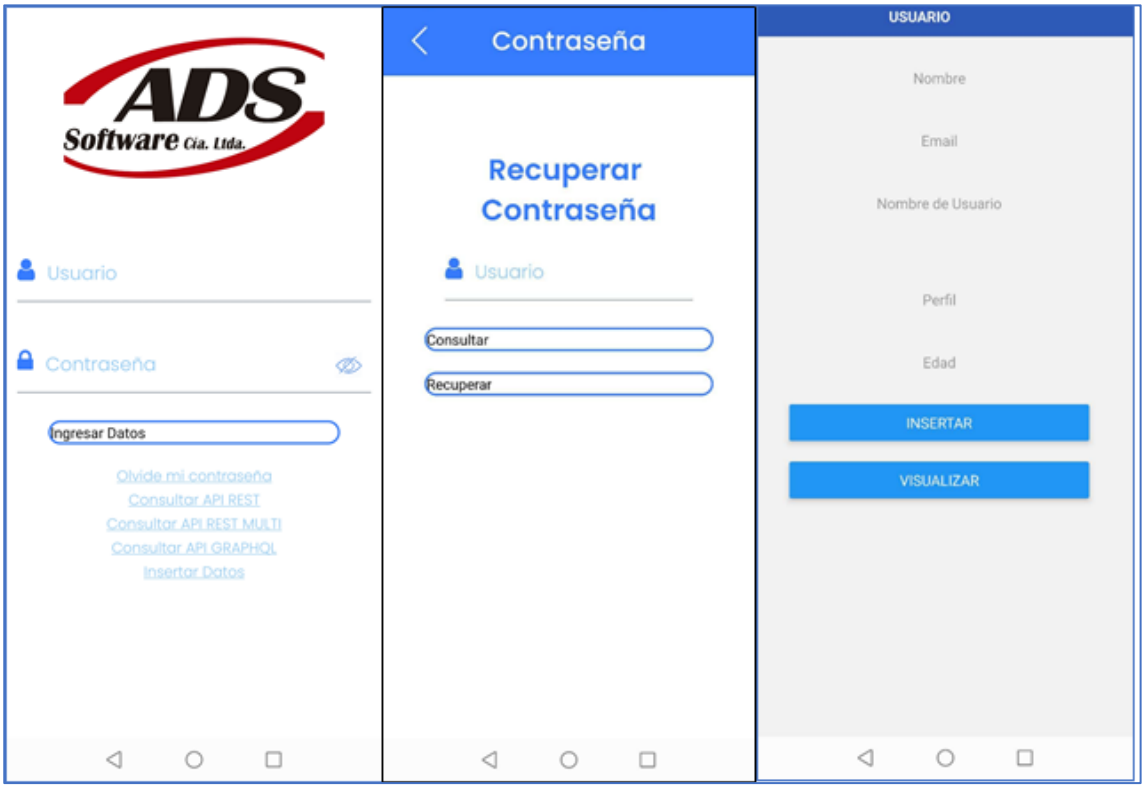
4.2.3. Desarrollo

El desarrollo se lo realizó de forma iterativa-incremental, el cual contiene los siguientes eventos: la reunión para conocer la necesidad del cliente, el desarrollo de las historias de usuario y sus estimaciones, para formar el release planning, construcción del software, pruebas funcionales y la reunión de revisión.

4.2.3.1. ***Reunión de Revisión***. En el presente se tomó en cuenta los criterios de aceptación de la historia de Usuario que se muestra en la tabla N 4, el cual consiste en poder reestablecer la contraseña del usuario cuando se haya olvidado, por medio del ingreso de su usuario y contraseña; a continuación de muestran varias pantallas.

Figura 29

Vista de Pantallas



4.3. Medición del Modelo de Calidad Externa de Software

En esta sección la implementación del modelo, se lo realizará a través de los métodos empíricos: experimento y encuesta; los cuales enfatizan el uso de estudios empíricos los mismo que sirven para acumular conocimiento.

4.3.1. Experimento

Se ponen en marcha cuando queremos controlar la situación y manipular el comportamiento de forma directa, precisa y sistemática, pueden realizarse en línea, lo que significa que la investigación se ejecuta sobre un contexto real. Son orientados a: las personas; aplicando diferentes tratamientos a los objetos y la tecnología: usando herramientas a los objetos (Wohlin, y otros, 2012).

Para evaluar la característica de eficiencia se empleará este método empírico aplicando lo siguiente:

Alcance

El presente experimento consiste en evaluar la calidad externa de software, tomando en cuenta la característica de eficiencia y subcaracteristicas de: comportamiento en el tiempo, utilización de recursos (CPU) y capacidad, por medio de una comparación entre las API's: Rest y GraphQL.

Planificación

Para la ejecución de este se toma en cuenta:

La Api GraphQL y Api Rest, en esta se desarrolló dos:

- La primera es un endpoint especifico mediante una consulta compuesta relacionada al caso de uso del experimento a la cual llamaremos API Rest.

- La segunda son diferentes endpoints para su consumo la cual se denominará API Rest Multi.

Las API’s se encuentran implementadas en un ordenador de forma local con las siguientes características:

- Sistema Operativo Windows 10 Pro
- Procesador Inter Core i7-5500U CPU 2.40
- Memoria RAM 8 GB
- Disco Duro 1 T
- Batería con duración 10 horas

Para su consumo, se lo realizará por medio de un dispositivo móvil con las siguientes características:

- Marca Huawei P30 lite
- Modelo MAR-LX3A
- Versión de Android 10
- Procesador Hisilicon Kirin 710
- RAM 4GB
- Almacenamiento 128 GB
- Resolución 2312 x 1080

Para la evaluación de comportamiento temporal, se tomará en cuenta el tiempo de respuesta desde que se realiza la petición hasta su contestación, para lo cual definimos casos de uso de consultas (select) de datos como se muestra en la tabla 8 y 9, definiendo hasta 3 niveles de relación con cantidades de: 1, 100, 1000, 10000, 30000, 50000 registros con tres repeticiones, lo cual implica la evaluación de las

diferentes API. En la misma se evaluará la utilización de recurso, mediante la toma de velocidad de procesamiento del CPU.

Tabla 8

Caso de Uso 1

Caso de Uso 1	
Escenario	Consulta de datos
Actores	Usuario del Sistema
Paso	Observación
Acceso a la intranet	
Consulta de datos 1 Nivel	Se requiere visualizar la información de consumo de tiempos de las actividades que realiza cada empleado.
Consulta de datos 2 Nivel	Se requiere visualizar la información de consumo de tiempo de las actividades que realiza cada empleado, juntamente con el detalle de la actividad realizada, el estado en el que se encuentra, el modo en lo que se realizó, la prioridad establecida para la tarea, el proyecto al cual pertenece, la persona que solicito realizarla, el tipo de actividad, la empresa en donde se realiza la actividad, y la prefactura que se le emitirá al cliente por el trabajo realizado.
Consulta de datos 3 Nivel	Se requiere visualizar la información de consumo de tiempo de las actividades que realiza cada empleado, juntamente con el detalle de la actividad realizada, el estado en el que se encuentra, el modo en lo que se realizó, la prioridad establecida para la tarea, el proyecto al

Caso de Uso 1	
	cual pertenece, la persona que solicito realizarla, el tipo de actividad, la empresa en donde se realiza la actividad, la prefactura que se le emitirá al cliente por el trabajo realizado, el distribuidor por donde se contactó para realizar la actividad y la tarifa que se maneja de acuerdo al cliente.

Tabla 9

Caso de Prueba 1

Caso de Prueba 1	
Objetivo del caso de prueba	Validar la consulta de datos por medio de la API GRAPHQL vs API REST, manejando 3 niveles de consulta.
Nombre del Caso	Consulta de Actividades
Precondiciones	Ejecutar las API's Graphql y Rest. Tener un log de tiempo de respuesta. Las consultas se realizan por cantidad de datos y concurrencia (las veces que se llama)
Paso	Resultado Esperado
Ejecutar la App Móvil	Se debe mostrar la interfaz de login de la app
Click en el Botón Consultar	Se mostrará el tiempo que se tomó en responder la consulta.

Para la evaluación de capacidad, se tomará en cuenta el número de peticiones síncronas que se realizan a cada API, para lo cual realizaremos un caso de uso como

se muestra en la tabla 10, realizando un conteo de la cantidad de registros ingresados en 5, 10, 20, 30 segundos. En la misma se evaluará la utilización de recursos mediante la toma de velocidad de procesamiento del CPU.

Tabla 10

Caso de Uso 2

Caso de Uso 2	
Escenario	Inserción de Datos
Actores	Usuario del Sistema
Paso	Observación
Acceso a la intranet	
Inserción de Datos	Se realizará la inserción de datos. Se contabilizará la cantidad de inserciones en el tiempo de 10 segundos. Los datos manejados serán: { "ConsumoTiempos":{ "detalle":"", "fechaFin":"", "fechaInicio":"", "actividad_id":, "consumidor_id":, "prefactura_id":, "proyecto_id":, "tipo_id": } }

Operación

La operación consta de:

Preparación, se realiza un estudio para comprobar el tiempo de respuesta y su rendimiento de CPU de dos tecnologías, al momento de realizar tres repeticiones con distintas cantidades de registros, también para probar la cantidad de transacciones realizadas en un determinado lapso.

Ejecución; al ejecutar la toma de datos de tiempo de respuesta que ofrece las dos tecnologías y su rendimiento de CPU, se ingresa los datos en una hoja de Excel de acuerdo con su nivel, cantidad de registros y repeticiones, para la toma de información de capacidad se ejecutará un conteo de transacciones en los tiempos indicados ingresando los datos en una hoja de Excel.

Análisis e interpretación

Para el análisis de los datos se empleará el programa SPSS que nos permite generar matrices de correlación de Pearson, los cuales nos permitirá visualizar el grado de relación lineal que contiene cada variable, también se usará un análisis discriminante el que permite observar diferencias significativas entre las tecnologías aplicadas en la experimentación, aplicando el estadístico de Lambda de Wilks.

Comportamiento Temporal – Utilización de Recursos

Los datos recogidos al terminar la experimentación, con respecto a comportamiento temporal son los que se muestran en la tabla 11, del mismo modo se presentan los datos del uso de rendimiento del CPU que se pueden visualizar en la tabla 12.

Tabla 11

Toma de Datos Comportamiento Temporal

Nivel	Nro. Registros	Nro. Repetición	Tiempo Rest (ms)	Tiempo Rest Multi (ms)	Tiempo GraphQL (ms)
1	1	1	29.16510	26.98542	0.74062
		2	20.65990	21.34583	1.09896
		3	14.67396	19.36146	1.16458
	100	1	24.76146	29.71250	1.00156
		2	19.32656	21.00365	1.12344
		3	17.89010	23.42708	1.67083
	1000	1	22.63906	24.97604	0.81094
		2	22.11615	16.44010	1.35104
		3	16.23542	20.03646	0.92448
	10000	1	34.24427	28.48438	2.99687
		2	20.99479	21.00260	1.77500
		3	18.09010	17.99792	1.72917
	30000	1	32.33698	24.62031	1.26458
		2	16.11719	20.14948	1.06563
		3	22.19688	22.69375	0.96354
	50000	1	29.53281	25.56250	1.36927
		2	21.46458	19.70052	1.15989
		3	19.20417	18.99896	1.09792
2	1	1	39.67917	25.97083	2.65938

Nivel	Nro. Registros	Nro. Repetición	Tiempo Rest (ms)	Tiempo Rest Multi (ms)	Tiempo GraphQL (ms)
		2	19.59531	15.21771	2.09844
		3	19.44896	19.39896	1.21198
		1	27.60417	25.41927	2.93021
	100	2	21.34948	23.92396	1.19687
		3	17.27240	21.90052	1.78333
		1	29.62708	48.86510	2.07917
	1000	2	18.26042	15.39635	1.10521
		3	16.74115	23.24427	1.40104
		1	26.66875	28.88854	2.59479
	10000	2	24.87708	17.44740	1.91875
		3	18.70000	18.63594	1.04375
		1	33.22865	27.20521	2.47135
	30000	2	20.68958	16.37865	1.02865
		3	19.60990	16.17083	1.35208
		1	31.13021	24.22292	2.60208
	50000	2	20.94375	16.86927	1.29688
		3	18.02344	12.79479	1.89010
		1	23.61563	27.74583	2.71354
	1	2	16.96250	21.07188	2.35677
3		3	18.22448	17.60625	1.17552
		1	27.63490	29.17396	3.51406
	100	2	15.57917	14.56042	1.16927

Nivel	Nro. Registros	Nro. Repetición	Tiempo Rest (ms)	Tiempo Rest Multi (ms)	Tiempo GraphQL (ms)
		3	18.30938	23.73490	2.07917
		1	27.36094	32.04323	2.47135
	1000	2	19.78542	14.24375	1.11979
		3	16.14115	17.11250	1.84063
		1	42.77813	45.85833	2.49062
	10000	2	16.21458	24.23646	1.30781
		3	16.00000	17.27240	1.93281
		1	26.85990	24.10156	2.59115
	30000	2	17.76771	41.30104	1.84896
		3	17.71042	22.66094	2.01250
		1	23.32917	29.72500	2.58333
	50000	2	19.88854	15.67396	1.26615
		3	17.05729	14.83646	1.17396

Nota. El gráfico representa los datos obtenidos del comportamiento temporal por cada nivel de consulta, cantidad de registros y repeticiones realizadas en función de milisegundos.

Tabla 12

Toma de Datos Rendimiento CPU

Rendimiento CPU Rest (ms) / (GHz)		**Rendimiento CPU Rest Multi (ms) / (GHz)**		**Rendimiento CPU GraphQL (ms) / (GHz)**	
429	2.331E-09	734	1.3624E-09	8	0.000000125
28	3.57143E-08	21	4.7619E-08	9	1.11111E-07
8	0.000000125	8	0.000000125	8	0.000000125
38	2.63158E-08	49	2.04082E-08	70	1.42857E-08
32	3.125E-08	24	4.16667E-08	156	6.41026E-09
11	9.09091E-08	20	0.00000005	153	6.53595E-09
65	1.53846E-08	109	9.17431E-09	395	2.53165E-09
34	2.94118E-08	83	1.20482E-08	250	0.000000004
60	1.66667E-08	81	1.23457E-08	152	6.57895E-09
265	3.77358E-09	383	2.61097E-09	425	2.35294E-09
440	2.27273E-09	404	2.47525E-09	360	2.77778E-09
430	2.32558E-09	498	2.00803E-09	359	2.78552E-09
734	1.3624E-09	864	1.15741E-09	540	1.85185E-09
1211	8.25764E-10	1183	8.45309E-10	321	3.11526E-09
724	1.38122E-09	1021	9.79432E-10	415	2.40964E-09
1139	8.77963E-10	5140	1.94553E-10	510	1.96078E-09
1263	7.91766E-10	1694	5.90319E-10	485	2.06186E-09
1520	6.57895E-10	2950	3.38983E-10	350	2.85714E-09
67	1.49254E-08	59	1.69492E-08	44	2.27273E-08

Rendimiento CPU Rest (ms) / (GHz)		Rendimiento CPU Rest Multi (ms) / (GHz)		Rendimiento CPU GraphQL (ms) / (GHz)	
52	1.92308E-08	50	0.00000002	31	3.22581E-08
43	2.32558E-08	8	0.000000125	24	4.16667E-08
68	1.47059E-08	65	1.53846E-08	70	1.42857E-08
65	1.53846E-08	60	1.66667E-08	153	6.53595E-09
41	2.43902E-08	55	1.81818E-08	145	6.89655E-09
3500	2.85714E-10	2651	3.77216E-10	380	2.63158E-09
6280	1.59236E-10	880	1.13636E-09	314	3.18471E-09
7945	1.25865E-10	1359	7.35835E-10	320	3.125E-09
6280	1.59236E-10	3584	2.79018E-10	141	7.0922E-09
4800	2.08333E-10	8405	1.18977E-10	424	2.35849E-09
1400	7.14286E-10	8632	1.15848E-10	378	2.6455E-09
125000	8E-12	143500	6.96864E-12	641	1.56006E-09
258451	3.86921E-12	111500	8.96861E-12	520	1.92308E-09
255000	3.92157E-12	111420	8.97505E-12	410	2.43902E-09
280451	3.56569E-12	258000	3.87597E-12	710	1.40845E-09
256520	3.89833E-12	249000	4.01606E-12	654	1.52905E-09
67	1.49254E-08	69	1.44928E-08	51	1.96078E-08
52	1.92308E-08	65	1.53846E-08	23	4.34783E-08
43	2.32558E-08	41	2.43902E-08	20	0.00000005
68	1.47059E-08	71	1.40845E-08	623	1.60514E-09
57	1.75439E-08	55	1.81818E-08	583	1.71527E-09

Rendimiento CPU Rest (ms) / (GHz)		Rendimiento CPU Rest Multi (ms) / (GHz)		Rendimiento CPU GraphQL (ms) / (GHz)	
55	1.81818E-08	67	1.49254E-08	538	1.85874E-09
2490	4.01606E-10	2690	3.71747E-10	438	2.28311E-09
1140	8.77193E-10	980	1.02041E-09	405	2.46914E-09
3720	2.68817E-10	3945	2.53485E-10	398	2.51256E-09
4800	2.08333E-10	4975	2.01005E-10	397	2.51889E-09
6280	1.59236E-10	1120	8.92857E-10	350	2.85714E-09
7950	1.25786E-10	8410	1.18906E-10	350	2.85714E-09
120000	8.33333E-12	115000	8.69565E-12	650	1.53846E-09
126000	7.93651E-12	128500	7.7821E-12	500	0.000000002
150000	6.66667E-12	120651	8.28837E-12	495	2.0202E-09
246000	4.06504E-12	258300	3.87147E-12	700	1.42857E-09
258000	3.87597E-12	250000	4E-12	684	1.46199E-09
240000	4.16667E-12	225000	4.44444E-12	571	1.75131E-09
67	1.49254E-08	69	1.44928E-08	51	1.96078E-08

Nota. El gráfico representa los datos de rendimiento del CPU por cada nivel de consulta, cantidad de registros y repeticiones realizadas en función de milisegundos y transformados a Gigahertz.

Para realizar el análisis de los datos de la tabla 11 y 12 se utiliza el programa estadístico SPSS permitiendo generar la matriz de correlación de Pearson por cada

nivel, que consiste en medir el grado de relación lineal entre cada variable. Los valores de correlación se ubican en -1 y +1.

Figura 30

Matriz de Correlación Nivel 1

Correlaciones

		APIS_ REST	APIS_ REST MULTI	APIS_ GRAPHQL	CPU_ REST	CPU_ REST MULTI	CPU_ GRAPHQL
APIS_ REST	Correlación de Pearson	1	,718**	,363	,110	,270	,394
	Sig. (bilateral)		,001	,139	,664	,279	,106
	N	18	18	18	18	18	18
APIS_ REST MULTI	Correlación de Pearson	,718**	1	,161	-,138	,056	,003
	Sig. (bilateral)	,001		,524	,584	,825	,992
	N	18	18	18	18	18	18
APIS_ GRAPHQL	Correlación de Pearson	,363	,161	1	-,086	-,044	,316
	Sig. (bilateral)	,139	,524		,735	,864	,202
	N	18	18	18	18	18	18
CPU_ REST	Correlación de Pearson	,110	-,138	-,086	1	,786**	,620**
	Sig. (bilateral)	,664	,584	,735		,000	,006
	N	18	18	18	18	18	18
CPU_ REST MULTI	Correlación de Pearson	,270	,056	-,044	,786**	1	,527*
	Sig. (bilateral)	,279	,825	,864	,000		,025
	N	18	18	18	18	18	18
CPU_ GRAPHQL	Correlación de Pearson	,394	,003	,316	,620**	,527*	1
	Sig. (bilateral)	,106	,992	,202	,006	,025	
	N	18	18	18	18	18	18

**. La correlación es significativa en el nivel 0,01 (bilateral).

*. La correlación es significativa en el nivel 0,05 (bilateral).

En los resultados de la figura 30, se puede visualizar que la API Rest Multi y la API Rest tienen una correlaciona lineal positiva de 0.718. El rendimiento de GraphQL y el rendimiento de Rest tienen una relación lineal positiva de 0.620.

Figura 31

Matriz de Correlación Nivel 2

Correlaciones

		APIS_ REST	APIS_ REST MULTI	APIS_ GRAPHQL	CPU_ REST	CPU_ REST MULTI	CPU_ GRAPHQL
APIS_ REST	Correlación de Pearson	1	,560*	,754**	-,026	,064	,011
	Sig. (bilateral)		,016	,000	,919	,801	,966
	N	18	18	18	18	18	18
APIS_ REST MULTI	Correlación de Pearson	,560*	1	,436	-,318	-,259	-,122
	Sig. (bilateral)	,016		,071	,199	,300	,630
	N	18	18	18	18	18	18
APIS_ GRAPHQL	Correlación de Pearson	,754**	,436	1	-,109	,042	-,111
	Sig. (bilateral)	,000	,071		,668	,870	,661
	N	18	18	18	18	18	18
CPU_ REST	Correlación de Pearson	-,026	-,318	-,109	1	,917**	,786**
	Sig. (bilateral)	,919	,199	,668		,000	,000
	N	18	18	18	18	18	18
CPU_ REST MULTI	Correlación de Pearson	,064	-,259	,042	,917**	1	,850**
	Sig. (bilateral)	,801	,300	,870	,000		,000
	N	18	18	18	18	18	18
CPU_ GRAPHQL	Correlación de Pearson	,011	-,122	-,111	,786**	,850**	1
	Sig. (bilateral)	,966	,630	,661	,000	,000	
	N	18	18	18	18	18	18

*. La correlación es significativa en el nivel 0,05 (bilateral).

**. La correlación es significativa en el nivel 0,01 (bilateral).

En los resultados de la figura 31, se puede evidenciar que el rendimiento de GraphQL y el rendimiento de Rest Multi tienen una correlación positiva de 0.850. El Api Rest y el Api Graphql tienen una correlación positiva de 0.754.

Figura 32

Matriz de Correlación Nivel 3

Correlaciones

		APIS_ REST	APIS_ REST MULTI	APIS_ GRAPHQL	CPU_ REST	CPU_ REST MULTI	CPU_ GRAPHQL
APIS_ REST	Correlación de Pearson	1	,709**	,580*	-,092	-,077	,117
	Sig. (bilateral)		,001	,012	,716	,761	,643
	N	18	18	18	18	18	18
APIS_ REST MULTI	Correlación de Pearson	,709**	1	,583*	-,065	-,042	,046
	Sig. (bilateral)	,001		,011	,799	,869	,855
	N	18	18	18	18	18	18
APIS_ GRAPHQL	Correlación de Pearson	,580*	,583*	1	-,150	-,131	,059
	Sig. (bilateral)	,012	,011		,553	,604	,816
	N	18	18	18	18	18	18
CPU_ REST	Correlación de Pearson	-,092	-,065	-,150	1	,996**	,582*
	Sig. (bilateral)	,716	,799	,553		,000	,011
	N	18	18	18	18	18	18
CPU_ REST MULTI	Correlación de Pearson	-,077	-,042	-,131	,996**	1	,587*
	Sig. (bilateral)	,761	,869	,604	,000		,010
	N	18	18	18	18	18	18
CPU_ GRAPHQL	Correlación de Pearson	,117	,046	,059	,582*	,587*	1
	Sig. (bilateral)	,643	,855	,816	,011	,010	
	N	18	18	18	18	18	18

**. La correlación es significativa en el nivel 0,01 (bilateral).

*. La correlación es significativa en el nivel 0,05 (bilateral).

En los resultados de la figura 32, el rendimiento de Rest y Rest Multi tienen una correlación lineal positiva de 0.996. La Api Graphql y Api Rest tienen una correlación lineal positiva de 0.580, mientras que con la Api Rest Multi tiene una correlación positiva de 0.583, por lo tanto, estos elementos paracen medir la misma característica.

Posterior a realizar la matriz de correlación se aplicará un análisis discriminante que nos permita visualizar si existen diferencias significativas entre las tecnologías de comportamiento temporal, para ellos utilizaremos el estadístico de Lambda de Wilks que permitirá contrastar que los grupos son iguales y en consecuencia no existe diferencia entre ellos, donde se evalúa el p-valor (normalmente 0,05) y el nivel de significancia.

Figura 33

Prueba de Medias Grupos Comportamiento Temporal - CPU

Prueba de igualdad de medias de grupos

	Lambda de Wilks	F	gl1	gl2	Sig.
APIS_ REST	,976	,622	2	51	,541
APIS_ REST MULTI	,986	,371	2	51	,692
APIS_ GRAPHQL	,808	6,049	2	51	,004
CPU_ REST	,860	4,149	2	51	,021
CPU_ REST MULTI	,872	3,739	2	51	,031
CPU_ GRAPHQL	,911	2,481	2	51	,094

Como se pueda visualizar en la figura 33, el nivel de significancia de API GraphQL es del 0.004, siendo este el más bajo, lo que se concluye que existe diferencias entre los grupos y se puede determinar que la tecnología GrapQL tiene una gran ventaja entre las otras. Del mismo modo en temas de rendimiento de CPU, podemos observar que el valor 0.21 de CPU Rest es menor al p-valor (0.05).

A fin de evaluar las subcaracteristicas de comportamiento temporal y utilización de recursos, se analiza los valores obtenidos del estadístico de lambda de wilks y su nivel de significancia donde se puede evidenciar que: la tecnología GraphQL tiene un valor de 0.808 en escala de 0 a 1 y de 0.004 respectivamente, del mismo modo el rendimiento del CPU de Rest tiene un valor de 0.860 en escala de 0 a 1 y de 0.021.

Capacidad – Utilización de Recursos

Al culminar con la experimentación los datos tomados, con respecto a la capacidad se visualizan en la tabla 13, del mismo modo se presentan los datos del rendimiento del CPU que se observan en la tabla 14.

Tabla 13

Toma de Datos Capacidad

Nro. Campos	Tiempo Segundos	Cantidad Rest	Cantidad Rest Multi	Cantidad GraphQL
8	5	5	8	12
	10	13	16	26
	20	24	16	52
	30	39	24	80

Nota. El gráfico representa la cantidad de registros insertados por el número de campos y el tiempo empleado.

Tabla 14

Toma de Datos Capacidad de Rendimiento CPU

Rendimiento Capacidad - CPU Rest (ms / GHz)		Rendimiento Capacidad – CPU Rest Multi (ms / GHz)		Rendimiento Capacidad – CPU GraphQL (ms / GHz)	
192.65	5.19076E-09	190.56	5.2477E-09	93.45	1.0701E-08
198.78	5.03069E-09	198.78	5.0307E-09	106.46	9.3932E-09
200.45	4.98878E-09	201.35	4.9665E-09	110.35	9.0621E-09
350.1	2.85633E-09	350.02	2.857E-09	130.96	7.6359E-09

Nota. El gráfico representa el rendimiento del CPU por el número de campos y el tiempo empleado.

Para realizar el análisis de los datos de la tabla 12 y13 se aplicará los mismos métodos estadísticos, que la subcaracterísticas de Comportamiento Temporal – Utilización de Recursos.

Figura 34

Matriz de Correlación Capacidad

Correlaciones

		APIS_ REST	APIS_ REST MULTI	APIS_ GRAPHQL	CPU_ REST	CPU_ REST MULTI	CPU_ GRAPHQL
APIS_ REST	Correlación de Pearson	1	,942	,999**	,869	,877	,979*
	Sig. (bilateral)		,058	,001	,131	,123	,021
	N	4	4	4	4	4	4
APIS_ REST MULTI	Correlación de Pearson	,942	1	,926	,840	,849	,985*
	Sig. (bilateral)	,058		,074	,160	,151	,015
	N	4	4	4	4	4	4
APIS_ GRAPHQL	Correlación de Pearson	,999**	,926	1	,854	,862	,967*
	Sig. (bilateral)	,001	,074		,146	,138	,033
	N	4	4	4	4	4	4
CPU_ REST	Correlación de Pearson	,869	,840	,854	1	1,000**	,905
	Sig. (bilateral)	,131	,160	,146		,000	,095
	N	4	4	4	4	4	4
CPU_ REST MULTI	Correlación de Pearson	,877	,849	,862	1,000**	1	,912
	Sig. (bilateral)	,123	,151	,138	,000		,088
	N	4	4	4	4	4	4
CPU_ GRAPHQL	Correlación de Pearson	,979*	,985*	,967*	,905	,912	1
	Sig. (bilateral)	,021	,015	,033	,095	,088	
	N	4	4	4	4	4	4

**. La correlación es significativa en el nivel 0,01 (bilateral).

*. La correlación es significativa en el nivel 0,05 (bilateral).

En los resultados de la figura 34, se puede visualizar que la API GraphQL y la API Rest tienen una correlación linear positiva de 0.999.

Figura 35

Prueba de Igualdad de Medias de Grupos Capacidad - CPU

Prueba de igualdad de medias de grupos

	Lambda de Wilks	F	gl1	gl2	Sig.
APIS_ REST	,976	,622	2	51	,541
APIS_ REST MULTI	,986	,371	2	51	,692
APIS_ GRAPHQL	,808	6,049	2	51	,004
CPU_ REST	,860	4,149	2	51	,021
CPU_ REST MULTI	,872	3,739	2	51	,031
CPU_ GRAPHQL	,911	2,481	2	51	,094

Como se puede visualizar en la figura 35, el nivel de significancia de la API GraphQL es de 0.004, cuando se toma en consideración el tema de capacidad, lo que se concluye que la API GraphQL tiene mayor capacidad de transacción. Del mismo modo en temas de rendimiento de CPU, podemos observar que el valor 0.21 de CPU Rest es menor al p-valor (0.05).

A fin de evaluar las subcaracteristicas de capacidad y utilización de recursos, se compara los valores obtenidos del estadístico de lambda de wilks y su nivel de significancia donde se puede justificar que: la tecnología GraphQL tiene un valor de 0.808 en escala de 0 a 1 y de 0.004 respectivamente, del mismo modo el rendimiento del CPU de Rest tiene un valor de 0.860 en escala de 0 a 1 y de 0.021.

4.3.2. Encuesta

Se utilizan cuando una técnica o herramienta ya ha sido usada o antes de comenzar a hacerlo, son estudios retrospectivos de las relaciones y los resultados de una situación (Wohlin, y otros, 2012). Para evaluar la característica de portabilidad se

aplicará este método empírico con el objetivo de comparar los atributos de capacidad de ser instalado y adaptabilidad de las aplicaciones móviles “Support Control y Activities”.

Característica de Portabilidad

Para evaluar la portabilidad, se ejecutará una encuesta con una muestra de 17 personas, este durará máximo cinco minutos, previo a ella se efectuará la instalación de dos aplicaciones móviles en las cuales se tomará en cuenta las subcaracteristicas, atributos y métricas de:

Capacidad de ser instalado – Facilidad de instalación

Para verificar la métrica, se realiza lo detallado anteriormente, y se lo completará con la siguiente encuesta, para la recolección de datos.

ENCUESTA

La encuesta tiene como finalidad conocer la satisfacción del usuario al instalar dos aplicaciones móviles según las características de sus equipos.

1.- De acuerdo con su criterio, cuál de las aplicaciones fue más fácil de instalar.

a. Support Control
b. Activities
c. Ambas
d. Ninguna

2.- Califique el grado de dificultad de la instalación de las aplicaciones móviles.

Support Control	Activities

Muy Difícil	Muy Difícil
Difícil	Difícil
Neutral	Neutral
Fácil	Fácil
Muy Fácil	Muy Fácil

3.- Según su criterio cuál aplicación necesita una guía de instalación.

a. Suppor Control
b. Activities
c. Ambas
d. Ninguna

4.- ¿En su dispositivo móvil pudo instalar la aplicación Support Control?

a. Si
b. No

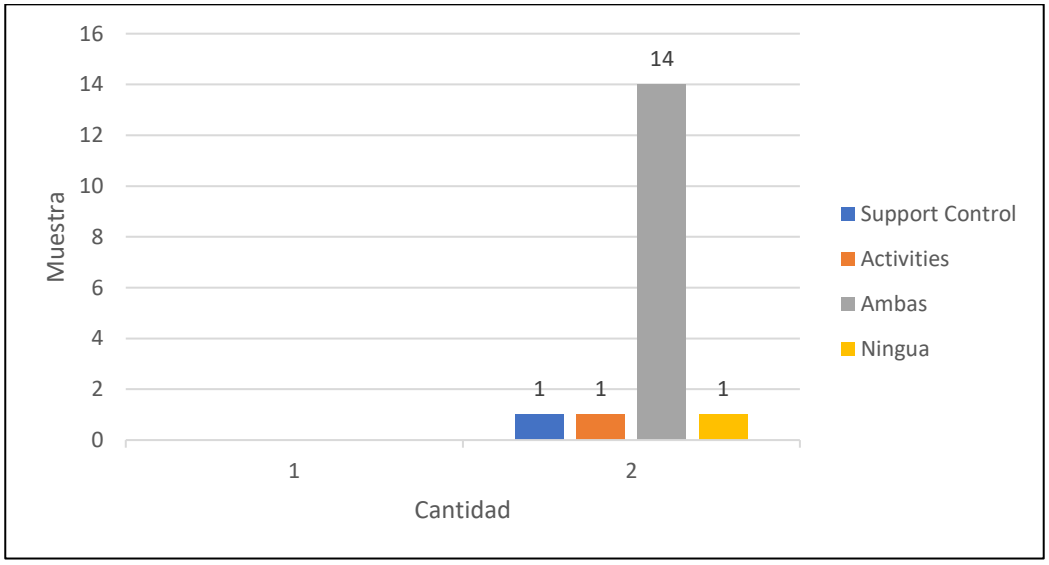
5.- ¿En su dispositivo móvil pudo instalar la aplicación Activities?

a. Si
b. No

Análisis de capacidad de ser instalado – facilidad de instalación

Figura 36

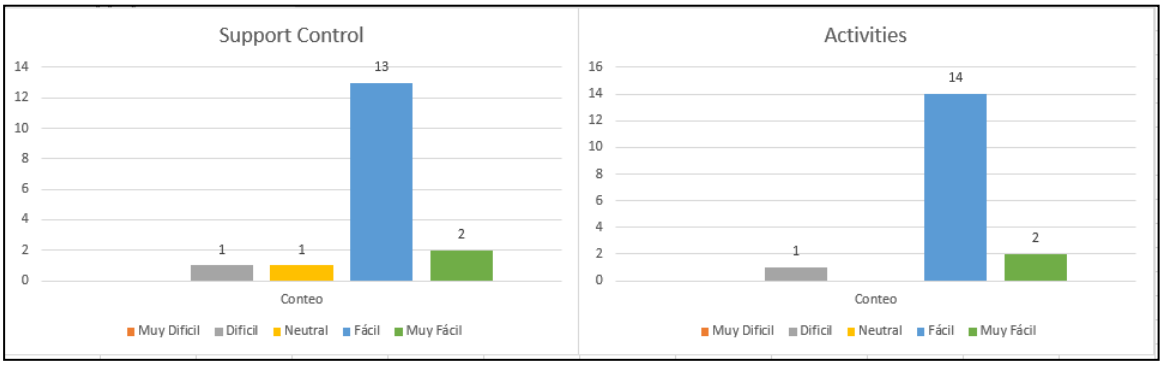
Pregunta 1 Capacidad de ser Instalado



Como se observa en la figura 36, tenemos que 14 personas están de acuerdo que ambas aplicaciones son fáciles de instalar, 1 que la instalación de la aplicación Support Control fue fácil, 1 que considera que lo fue la aplicación Activities y 1 que ninguna fue fácil de instalar.

Figura 37

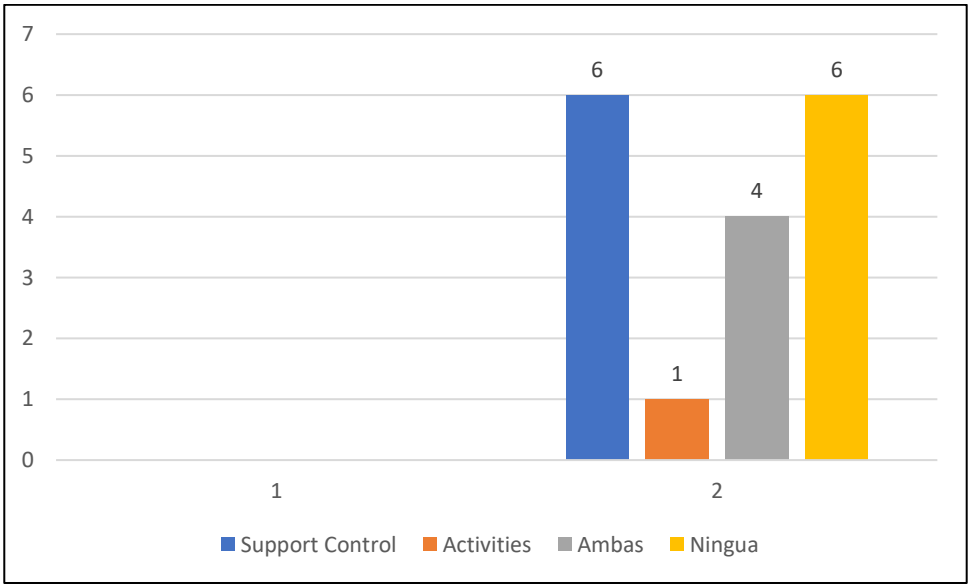
Pregunta 2 Capacidad de ser Instalado



En la figura 37 podemos observar la dificultad de instalación de cada aplicación, en la aplicación Support Control se visualiza que 13 personas consideran fácil la instalación, 2 la considera muy fácil, 1 neutral y 1 difícil. En la aplicación Activities 14 personas la consideran fácil, 2 muy fácil y 1 difícil.

Figura 38

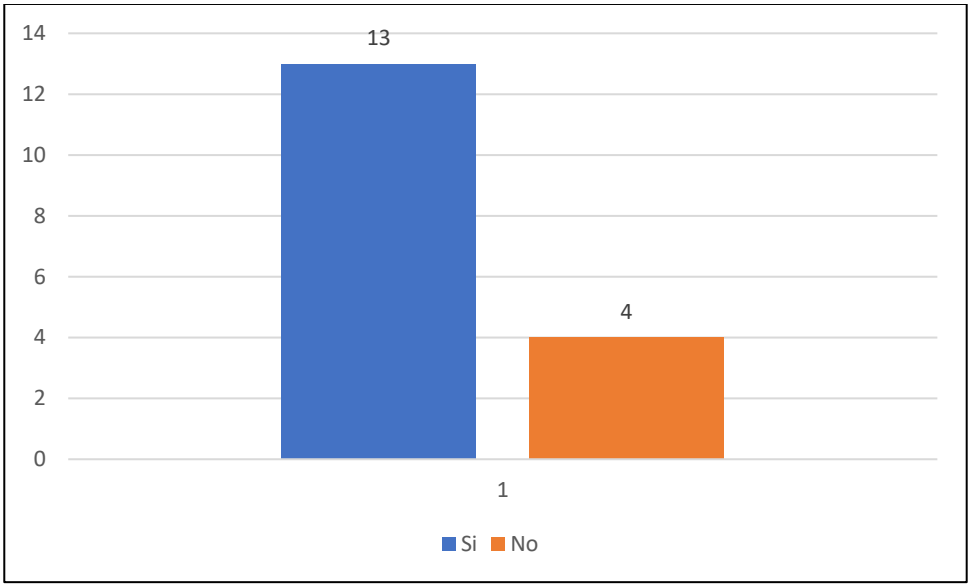
Pregunta 3 Capacidad de ser Instalado



En la figura 38, se observa cuál de las aplicaciones necesita de una guía para su instalación, por lo que visualizamos que, 6 de las personas consideran que se necesita una guía en la aplicación Support Control, 6 que ninguna la necesita, 4 que ambas y 1 que la aplicación activities necesita una guía.

Figura 39

Pregunta 4 Capacidad de ser Instalado



En la figura 39, se puede observar que en 13 dispositivos móviles se pudo instalar la aplicación Support Control y en 4 no se pudieron. A fin de evaluar la subcaracteristica de facilidad de instalación se emplea la fórmula:

$$X = \frac{A}{B}$$

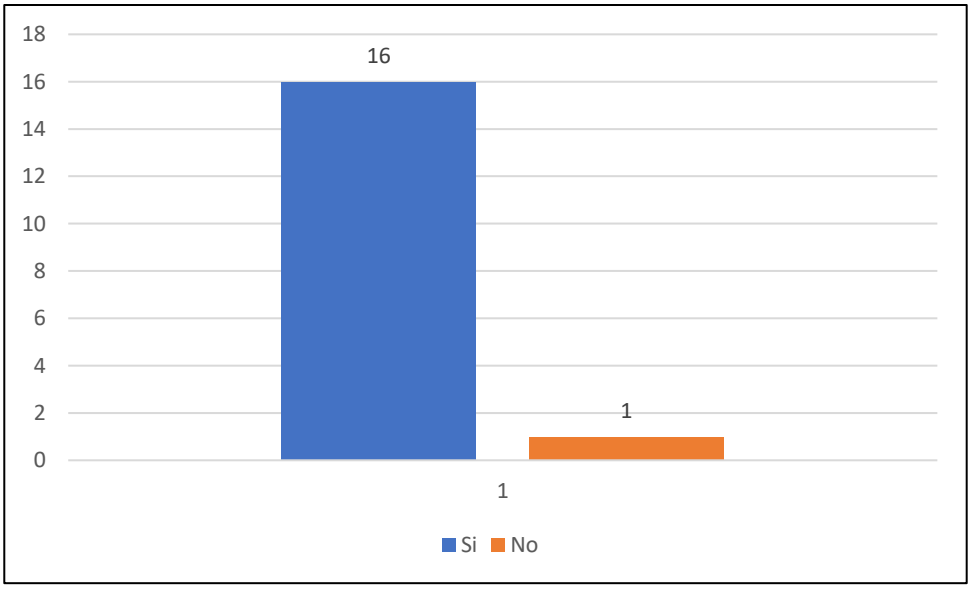
$$X = \frac{13}{17} = 0.77$$

Donde; A es el número de casos en los que el usuario tuvo éxito en la instalación, haciendo referencia a los 13 dispositivos en los cuales se instaló y B es el número de casos en los que el usuario personaliza el procedimiento de instalación

tomando en consideración a los 17 dispositivos que se les pidió instalar la aplicación Support Control, dando un valor de 0.77 en su facilidad de instalación.

Figura 40

Pregunta 5 Capacidad de ser Instalado



En la figura 40, se puede observar que en 16 dispositivos móviles se pudo instalar la aplicación Activities y en 1 no se pudo. Tomando en consideración que se pretende evaluar la subcaracteristica de facilidad de instalación, se emplea la formula.

$$X = \frac{A}{B}$$

$$X = \frac{16}{17} = 0.94$$

Donde; A es el número de casos en los que el usuario tuvo éxito en la instalación, haciendo referencia a los 16 dispositivos en los cuales se instaló sin ninguna novedad y B es el número de casos en los que el usuario personaliza el procedimiento de instalación tomando en consideración a los 17 dispositivos que se les

pidió instalar la aplicación Activities, dando un valor de 0.94 en su facilidad de instalación.

Al terminar el análisis de los datos, podemos concluir que en la subcaracteristica estudiada, la aplicación desarrollada (Activities) es mejor a la aplicación existente (Support Control), mediante la evaluación realizada por medio del modelo de calidad externa de software.

Adaptabilidad – Adaptabilidad al entorno hardware

Para comprobar la métrica, se ejecuta lo detallado con anterioridad, terminando la recolección de datos por medio de una encuesta la cual es:

ENCUESTA

La presente encuesta tiene como finalidad conocer las características de su dispositivo móvil donde fueron instaladas las aplicaciones.

1.- ¿Qué marca de teléfono tiene?

a. Samsung.
b. Huawei.
c. Xiaomi.
d. Honor.
e. Sony.
f. LG.
g. Nokia.
h. Oppo.
i. Iphone
j. Otro

2.- ¿Qué sistema operativo y versión tiene su teléfono móvil?

Sistema Operativos			
a) Android	b) IOS	c) Ni idea	d) Otro: _____

Versiones	
a. 5 o más	a. 6.1.6
b. 6 o más	b. 7.1.2
c. 7 o más	c. 9.3.6
d. 8 o más	d. 10.3.4
e. 9	e. 12.4.8
f. 10	f. 13.6
g. 11	g. 14.0.1
h. Ni idea	h. Ni idea

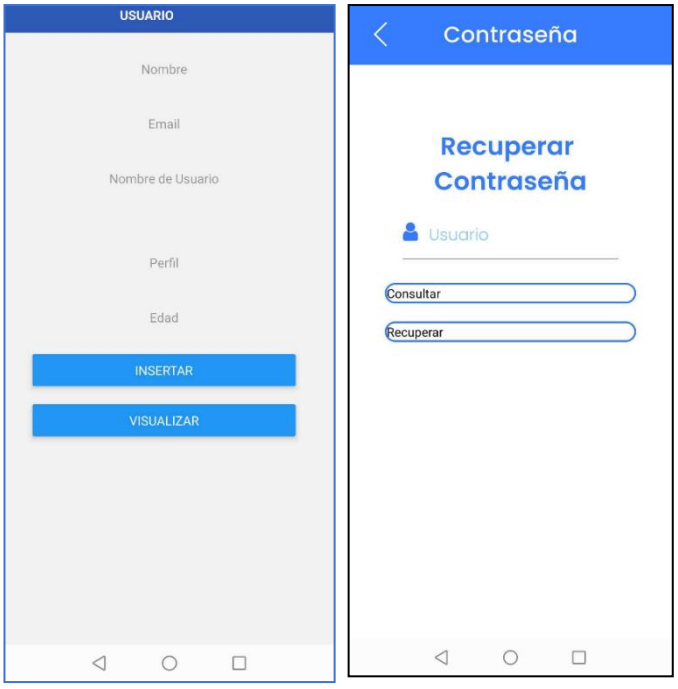
3.- ¿Al utilizar la App móvil, se visualiza correctamente las pantallas?

Support Control

- Si
- No

Activities

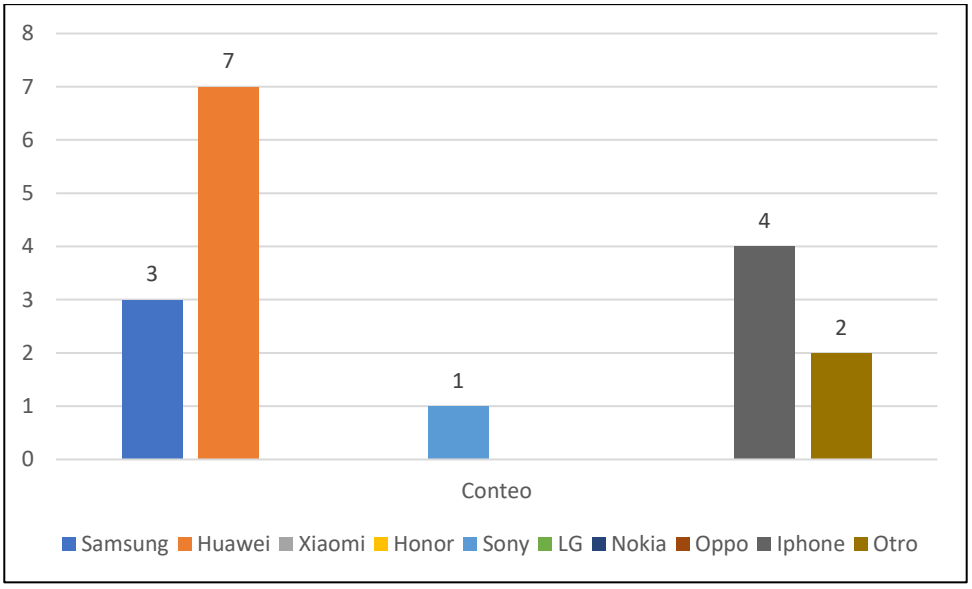
- Si
- No



Adaptabilidad – Adaptabilidad al entorno hardware

Figura 41

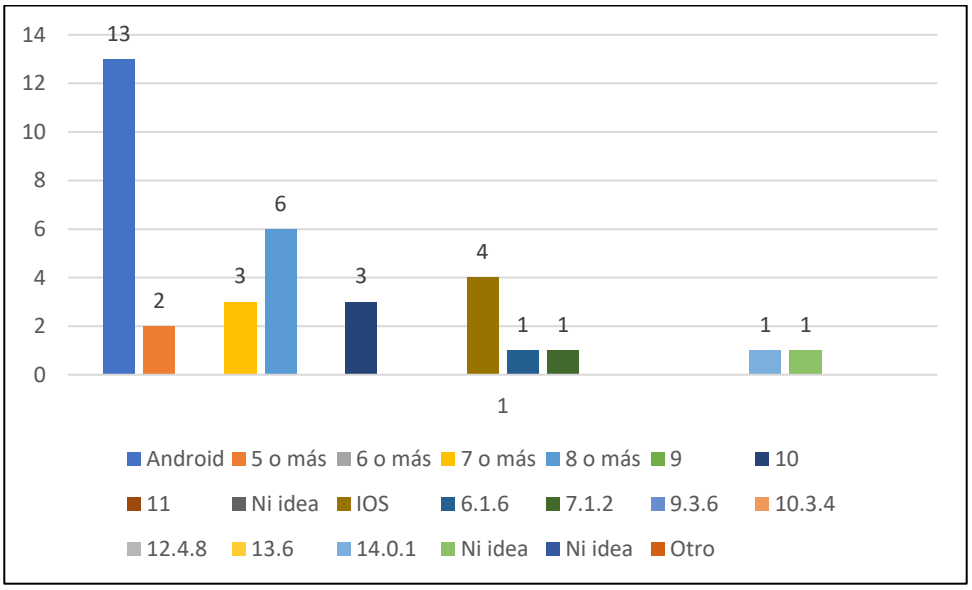
Pregunta 1 Adaptabilidad



Al visualizar la figura 41, podemos determinar que en las 17 personas que nos ayudaron en la experimentación, 7 personas cuentan con un teléfono móvil marca Huawei, 4 usan iphone, 3 samsung, 2 otros y 1 utiliza sony.

Figura 42

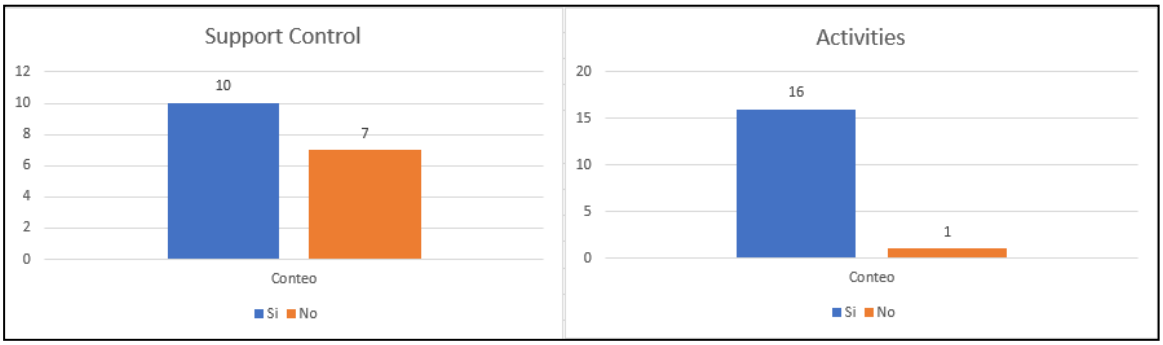
Pregunta 2 Adaptabilidad



En la figura 42, podemos observar que 13 de los dispositivos utilizados en el experimento tiene un sistema operativo Android, de los cuales 6 de ellos funcionan con la versión 8 o superior, 3 con la versión 7 y 10, y 2 con la versión 5. También 4 utilizan el sistema operativo IOS, que funcionan desde la versión 6.1.6 en adelante.

Figura 43

Pregunta 3 Adaptabilidad



En la figura 43, podemos determinar que en la aplicación Support Control 7 dispositivos no pudieron visualizar correctamente los formularios que se consideraron y 10 los pudieron observar. También pusimos en consideración los formularios en la aplicación Activities en donde 16 de los dispositivos pudieron visualizar los mismo, y solamente 1 no lo logro. A fin de evaluar la característica de adaptabilidad al entorno de hardware se emplea la formula.

$$X = 1 - \frac{A}{B}$$

Donde; A es el número de funciones que no se completaron y B es el número de funciones que han sido probadas.

Analizando la aplicación Support Control y mediante la fórmula mencionada, se observa que A son los 7 dispositivos en los cuales no se observó correctamente los formularios y B son los 17 dispositivos en los cuales se solicitó visualizarlos, dando como resultado un valor de 0.59.

$$X = 1 - \frac{7}{17} = 0.59$$

Del mismo modo se analizó la aplicación Activities; en la cual A es el único dispositivo donde no se observó correctamente los formularios y B son los 17 dispositivos, generando un valor de 0.94

$$X = 1 - \frac{1}{17} = 0.94$$

Al terminar el análisis de la subcaracteristica de Adaptabilidad, podemos determinar por medio de la evaluación del modelo de calidad externa de software que la aplicación Activities es mejor en la adaptabilidad del entorno de hardware, tomando en consideración que funciona tanto en sistemas operativos Android como IOS**.**

4.4. Conclusiones del capítulo

En el presente capitulo se realizó la medición del modelo de calidad externa de software, empleando los métodos empíricos de: experimento y encuesta. El experimento permitió comparar la eficiencia de dos aplicaciones móviles “Support Control y Activities”, que se desarrollaron con arquitecturas y tecnologías distintas, donde por medio del programa SPSS se generó matrices de correlación por niveles y un análisis discriminante, en el cual se aplicó el estadístico de lambda de wilks. Al concluir la medición y al aplicar el experimento; podemos determinar que, en la métrica de tiempo de respuesta, la tecnología GraphQL tiene una gran ventaja entre la tecnología Rest debido a la velocidad de respuesta obtenido en cada repetición y con la cantidad de 50000 registros consultados en 1.17396 milisegundos. Del mismo modo, en la métrica de capacidad se puede evidenciar su ventaja, al realizar la inserción de 80 registros en 30 segundos. De igual modo se determinó por medio de la encuesta que la aplicación “Activities” es mucho mejor al momento de su instalación y su adaptabilidad.

CAPÍTULO V

5. Evaluación del modelo de calidad externa de software

5.1. Introducción

En este capítulo se presenta la evaluación del modelo de calidad externa de software. La evaluación del modelo se realizó a través de los datos obtenidos en la aplicación Activities conseguidos en el apartado anterior y empleando el proceso de evaluación de la calidad propuesto en la sección 3.3.5, mismo que sirve como guía para ejecutarla, segmentándola en cuatro procesos: establecer los requisitos, especificarlos, ejecutarlos y concluirlos, mismo que se ejecutará en el rol de evaluador.

5.2. Organización de la recogida y procesamiento de los datos

En esta sección se evaluará el modelo de calidad externa de software, en la cual se hará referencia al proceso de evaluación de calidad externa propuesto, se segmenta en cuatro procesos.

5.2.1 Establecer los requisitos

Instituye el propósito de la evaluación

Tiene como objetivo determinar la eficiencia y portabilidad en la aplicación móvil desarrollada y medida mediante el modelo de calidad externa de software.

Define el rigor de la evaluación

Se considera las características, subcaracteristicas, su nivel de recomendación, en esta existen tres niveles: muy recomendable (MR), recomendable (R) y discreción de usuario (DU) y sus ponderaciones correspondientes; como se muestra en la tabla 15.

Tabla 15

Rigor de evaluación

Características	Subcaracteristicas	Metricas	Nivel	% Carac	% Subcar.
Eficiencia	Comportamiento temporal	Tiempo de Respuesta	MR	20	4
	Utilización de recursos	Utilización CPU	MR		1
	Capacidad	Número de peticiones	R		2
Portabilidad	Adaptabilidad	Adaptabilidad entorno hardware	MR	3	1
	Capacidad de ser instalado	Facilidad de instalación	MR		0.25

5.2.2 Especificar la evaluación

Selecciona las medidas de calidad

Se toma en cuenta las características, subcaracteristicas, métricas y sus porcentajes de las características establecidas a fin de evaluar el modelo de calidad externa de software, como se muestra en la tabla 16.

Tabla 16

Selección de medidas de calidad

Características	Subcaracteristicas	Metricas	% Carac	% Subcar.
Eficiencia	Comportamiento temporal	Tiempo de Respuesta	20	4
	Utilización de recursos	Utilización CPU		1
	Capacidad	Número de peticiones		2
Portabilidad	Adaptabilidad	Adaptabilidad entorno hardware	3	1
	Capacidad de ser instalado	Facilidad de instalación		0.25

Define el criterio de decisión para las medidas de calidad

La definición de criterio de decisión se lo planteara tomando como referencia la escala de Likert, siendo esta una clasificación de: excelente en un rango entre 1 – 0.90, bueno entre 0.89 – 0.75, aceptable entre 0.74 – 0.65, regular entre 0.64 – 0.45 y deficiente menor a 0.44, como se puede observar en la tabla 17.

Tabla 17

Criterios de Decisión

Interpretación de Resultados	Rango de Indicadores
EXCELENTE	$0.85 \leq X \leq 1$
BUENO	$0.75 \leq X \leq 0.84$
ACEPTABLE	$0.65 \leq X \leq 0.74$
REGULAR	$0.45 \leq X \leq 0.64$
DEFICIENTE	$X \leq 0.44$

Establece el criterio de decisión para la evaluación

El criterio de decisión para la evaluación se tomará como referencia la tabla 17, con la diferencia de realizar un promedio entre las medidas de calidad seleccionadas para su evaluación.

5.2.3 Ejecutar la evaluación

Hacer mediciones

Al aplicar los métodos empíricos de: experimento y encuesta utilizados en la sección 4.3, se obtiene los datos correspondientes a cada una de las subcaracteristicas; como se muestra en la tabla 18.

Tabla 18

Resultados de la Medición

Características	Subcaracteristicas	Metricas	Valor X
Eficiencia	Comportamiento temporal	Tiempo de Respuesta	0.81
	Utilización de recursos	Utilización CPU	0.86
	Capacidad	Número de peticiones	0.81
Portabilidad	Adaptabilidad	Adaptabilidad entorno hardware	0.94
	Capacidad de ser instalado	Facilidad de instalación	0.94

Tomando en consideración la definición del rigor de evaluación, se analiza los porcentajes correspondientes de acuerdo con los datos obtenidos en la tabla 18, principalmente se modifica las ponderaciones correspondientes a fin de trabajarlas en rango de 0 a 100%.

Tabla 19

Resultados de medición en porcentajes

Caract	Subcar	Metricas	% Carac	% Subcar.	% Md Carac	% Md Subcc
Eficiencia	Comportamiento temporal	Tiempo de Respuesta	87	50	70	40
	Utilización de recursos	Utilización CPU		12		10
	Capacidad	Número de peticiones		25		20
Portabilidad	Adaptabilidad	Adaptabilidad entorno hardware	13	10	12	9
	Capacidad de ser instalado	Facilidad de instalación		3		3

Como se puede observar en la tabla 19, tenemos las columnas de características, subcaracteristicas, métricas, porcentaje de característica, subcaracteristica esperado de acuerdo con el modelo de calidad externa de software y el porcentaje de característica, subcaracteristica medido al momento de ejecutar el experimento y la encuesta.

Aplicar el criterio de decisión para las medidas de calidad

Al analizar los datos obtenidos en la tabla 18 y tomando como referencia los criterios de decisión para las medidas de calidad se puede determinar que:

- Tiempo de respuesta, con su valor de 0.81 se encuentra en el rango de excelente.
- Utilización de recursos, con su valor de 0.86 se encuentra en el rango de excelente.
- Capacidad, con su valor de 0.81 se encuentra en el rango de excelente.
- Adaptabilidad, con su valor de 0.94 se encuentra en el rango de excelente.
- Capacidad de instalación, con su valor de 0.94 se encuentra en el rango de excelente.

Aplicar el criterio de decisión para la evaluación

Del mismo modo que se aplicó en la parte anterior, y tomando como crónica los criterios de decisión para la evaluación, se genera un promedio entre los datos obtenidos de las subcaracteristicas que se visualizan en la tabla 18, dando como resultado 0.87 encontrándose en el rango de excelente.

$$\frac{0.81 + 0.86 + 0.81 + 0.94 + 0.94}{5} = 0.87$$

5.2.4 Concluir la evaluación

Revisar los resultados de la evaluación

Al realizar el análisis de los datos obtenidos al momento de ejecutar la evaluación, se determinó que:

- La evaluación realizada al modelo de calidad externa de software es de 0.87 lo que equivale a un 82% sobre 100 considerando el modelo en un rango de excelente.

5.3. Conclusiones del capítulo

En el presente capitulo se realizó la evaluación del modelo de calidad externa de software, empleando el proceso de evaluación donde se determinó las subcaracteristicas a ser evaluadas de acuerdo con sus ponderaciones establecidas en el modelo y sus criterios de decisión para las medidas y la evaluación en las cuales se usó la escala de Likert, con sus diferentes rangos de aceptación. Al concluir la evaluación y al aplicar los criterios de decisión establecidos podemos determinar que las subcaracteristicas de comportamiento temporal, utilización de recursos, capacidad, adaptabilidad y capacidad de instalación se encuentran en un criterio de decisión excelente, y al realizar el promedio por características de eficiencia y portabilidad los mismos se encuentran en un criterio de decisión excelente. También se determinó que el grado de eficiencia es de un 70 % lo cual se considera excelente y la característica de portabilidad es de 12%.

CAPÍTULO VI

6. Conclusiones y recomendaciones

6.1. Conclusiones

La construcción del marco teórico estableció la base conceptual apropiada para fundamentar el desarrollo y evaluación del modelo de calidad externa de software presentado en el proyecto de investigación.

Mediante el estudio de la serie de normas ISO/IEC 25000 denominadas SQuaRE, se pudo abstraer el modelo de referencia, ciclo de vida, estructura y proceso de evaluación para entender, estructurar y desarrollar un modelo de calidad externa el cual sirvió como guía con el fin de ejecutar el proceso de evaluación de la calidad en el desarrollo de aplicativos móviles en la empresa ADS – SOFTWARE.

La implementación del modelo de calidad externa de software MCES, se enfocó en medir y evaluar las características de eficiencia y portabilidad de dos aplicaciones móviles desarrolladas con arquitecturas y tecnologías diferentes, mediante los métodos empíricos de: experimento y encuesta, donde se determinó que la tecnología GraphQL tiene una gran ventaja de la otra en términos de eficiencia, y también se comprobó que la aplicación denominada “Activities” tiene una gran facilidad de instalación y adaptabilidad al entorno hardware en los diferentes sistemas operativos.

Gracias a la implementación del modelo de calidad externa de software MCES podemos determinar a qué giro de negocio y/o personas van a ir dirigidas nuestras soluciones tecnológicas, con los alcances y limitaciones propuestos, como también se puede determinar los sistemas operativos y versiones en los cuales las aplicaciones móviles se pueden desarrollar.

En la validación del modelo de calidad externa de software MCES, se ejecutó el proceso de evaluación de calidad, donde se determinó que el grado de eficiencia es de un 70% siendo considerada excelente y la característica de portabilidad es de 12%, evidenciando la mejora de las características de eficiencia y portabilidad. Debido a que el modelo de calidad externa permite inspeccionar la calidad a nivel de producto y dar valores establecidos a cada característica, subcaracteristica y métrica, a fin detectar posibles fallos y desviaciones.

6.2. Recomendaciones

Capacitar al departamento de I+D (investigación y desarrollo) en normas y estándares enfocados en la calidad con el objetivo de contar con el recurso y conocimiento suficiente para la implementación del modelo propuesto en el proyecto de investigación a fin de ejecutar el proceso de evaluación de la calidad en los productos de software.

Aplicar en la empresa ADS – Software el modelo de calidad externa en las soluciones tecnológicas de escritorio o móviles desarrolladas o por desarrollarse, con el objetivo de detectar errores, defectos, fallas e incidentes de tal manera que implique un ahorro de costos, tiempo y recursos, asegurando la eficiencia y portabilidad en su implementación sin poner en compromiso con el resto de los sistemas y diferenciándose de los competidores en los tiempos de entrega y satisfacción de los clientes.

Implementar el modelo de calidad externa MCES en las empresas que requieran una solución tecnológica para la automatización de sus procesos, con el objetivo de conocer y comparar el producto que adquieren tanto en características de eficiencia y portabilidad, así también reducir costos finales de la compra del software.

Como trabajo futuro se recomienda desarrollar un software, el cual automatice los procesos del modelo de calidad externa, con el fin de poder ejecutar el proceso de evaluación de calidad externa de forma fácil, rápida y automatizada.

Considerar en el desarrollo de aplicaciones móviles o escritorio la implementación de la arquitectura de microservicios, estas funcionan como un conjunto de pequeños productos ejecutándose de forma independiente y autónoma facilitando la modificación y mantenimiento sin afectar su infraestructura.

Capacitar al departamento de I+D (investigación y desarrollo) en la tecnología GraphQL con el objetivo de contar con el conocimiento suficiente para el desarrollo de API’s rápidas, flexibles y sencillas, permitiendo al cliente manejar grandes cantidades de información sin generar una sobrecarga y obteniendo lo que solicita.

Bibliografía

25000, I. (30 de Mayo de 2020). *ISO 25000 CALIDAD DE SOFTWARE Y DATOS*. Obtenido de ISO 25000 CALIDAD DE SOFTWARE Y DATOS: https://iso25000.com/index.php/normas-iso-25000/7-iso-iec-2501n

25000, I. (30 de Mayo de 2021). *ISO 25000 CALIDAD DE SOFTWARE Y DATOS*. Obtenido de ISO 25000 CALIDAD DE SOFTWARE Y DATOS: https://iso25000.com/index.php/normas-iso-25000?start=0

25000, I. (5 de JUNIO de 2021). *ISO 25000 CALIDAD DE SOFTWARE Y DATOS*. Obtenido de ISO 25000 CALIDAD DE SOFTWARE Y DATOS: https://iso25000.com/index.php/normas-iso-25000/iso-25010

25000, I. (5 de JUNIO de 2021). *ISO 25000 CALIDAD DE SOFTWARE Y DATOS*. Obtenido de ISO 25000 CALIDAD DE SOFTWARE Y DATOS: https://iso25000.com/index.php/normas-iso-25000/iso-25010

Alena González Reyes, A. H. (2016). *Initial basic model of external quality for products of software.* La Habana: S/E.

Callejas Cuervo, M., Alarcón Aldana, A. C., & Álvarez Carreño, A. M. (2017). Modelos de calidad del software, un estado del arte. *Entramado*, 236-250.

César Augusto López Zapata, O. I. (31 de Mayo de 2020). *Modelos Wiki*. Obtenido de Modelos Wiki: https://modelos.fandom.com/es/wiki/El_modelo_SQAE_-_1995

Chisaguano, E. A. (S/D de Octubre de 2014). Evaluación de calidad de productos software en empresas de desarrollo de software aplicando la norma iso/iec

25000. *Evaluación de calidad de productos software en empresas de desarrollo de software aplicando la norma iso/iec 25000*. Quito, Pichincha, Ecuador.

Diez, E. (2013). Aseguramiento de la Calidad en la Construcción de Sistemas Basados en el Conocimiento: Un Enfoque Práctico. *Revista Latinoamericana de Ingenieria de Software*, 167-206.

Drake, J. M. (06 de Junio de 2020). *Programación orientada a objetos: Lenguages, metodologías y herramientas.* Obtenido de Programación orientada a objetos: Lenguages, metodologías y herramientas: https://www.ctr.unican.es/asignaturas/MC_OO/Doc/OO_08_I2_Proceso.pdf

Eisenman, B. (2016). *Learning React Native.* s/c: O'Reilly Media.

Enrique Morales, J. Y. (31 de Mayo de 2020). *Estándares del Software*. Obtenido de Estándares del Software: https://estandarsw.wordpress.com/category/iso/iso-9126/

Estayno, M. D. (S/D de S/M de 2012). Métodos y herramientas orientados a la calidad del software. *Métodos y herramientas orientados a la calidad del software*. S/C, Buenos Aires, Argentina: S/E.

Ferreyra Moreno, J. (5 de Abril de 2016). *Garantia de Calidad del Software (SQA/GCS)*. Obtenido de Garantia de Calidad del Software (SQA/GCS): https://silo.tips/download/81-conceptos-de-calidad

GARCIA, C. A. (30 de Mayo de 2019). *GUIA TECNICA PARA LA EVALUACIÓN DE SOFTWARE.* Obtenido de Sisfo.exe: https://jrvargas.files.wordpress.com/2009/03/guia_tecnica_para_evaluacion_de_software.pdf

Global, A. (06 de Junio de 2020). *DESARROLLO DE SOFTWARE*. Recuperado el Octubre de 2020, de DESARROLLO DE SOFTWARE: https://universidades.cr/carreras/desarrollo-de-software

GraphQL. (S/D de S/M de 2016). *Introduction to GraphQL*. Obtenido de Introduction to GraphQL: https://graphql.org/learn/schema/

GraphQL. (1 de JUNIO de 2018). *GraphQL Specification Versions*. Obtenido de GraphQL Specification Versions: http://spec.graphql.org/June2018/

INEN. (2014). *INEN Servicio Ecuatoriano de Normalización NT INEN-ISO/IEC25040.* QUITO: INEN.

INEN. (2014). *Instituto Ecuatoriano de Normalización NTE INEN-ISO/IEC 25000.* QUITO, PICHINCHA, ECUADOR: INEN.

INEN. (2014). *Instituto Ecuatoriano de Normalización NTE INEN-ISO/IEC 25010.* QUITO: INEN.

INEN. (2014). *Instituto Ecuatoriano de Normalización NTE INEN-ISO/IEC 25030.* QUITO: INEN.

INTECO. (20 de 03 de 2020). INTE/ISO/IEC 25023:2020. *Ingeniería de sistemas de Software - Requisits y evaluación de calidad de sistemas y del software (SQuaRE) - Medición de calidad de sistemas y productos Software*. Cuba: INTECO.

ISO 25000 calidad de software y datos. (31 de Mayo de 2020). Obtenido de ISO 25000: https://iso25000.com/index.php/normas-iso-25000/iso-25010

James Lewis, M. F. (25 de Marzo de 2014). *Microservices*. Obtenido de Microservices: https://martinfowler.com/articles/microservices.html#CharacteristicsOfAMicroserviceArchitecture

Jorge BenzaquenDe las Casas, M. P.-C. (2016). El ISO 9001 y TQM en las empresas. *Globalización, Competitividad y Gobernabilidad*, 153-176.

Ladino, P. (31 de Mayo de 2020). *Evaluación de la Calidad de la Tecnología Educativa*. Obtenido de Evaluación de la Calidad de la Tecnología Educativa: http://evaluaciondetecnologiaeducativa.blogspot.com/p/modelo-medida.html

Ladino, P. (31 de Mayo de 2020). *Evaluación de la Calidad de la Tecnología Educativa*. Obtenido de Evaluación de la Calidad de la Tecnología Educativa: http://evaluaciondetecnologiaeducativa.blogspot.com/p/modelo.html

Llaneza, M., Dapozo, G., Greiner, C., & Estayno, M. (2013). Análisis comparativo de modelos de calidad orientado al desarrollo de software. *XV WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACION*, 601-605.

López, D., & Maya, E. (2017). Arquitectura de Software basada en Microservicios para Desarrollo de Aplicaciones Web. *Séptima Conferencia de Directores de Tecnología de Información, TICAL 2017 Gestión de las TICs para la Investigación y la Colaboración, San José, del XX al XX de julio de 2017.* Quito: TICAL.

Martín, Á. J. (18 de Agosto de 2020). *React Native: ¿Qué es y para que sirve este framework de programación?* Obtenido de React Native: ¿Qué es y para que sirve este framework de programación?: https://openwebinars.net/blog/react-native-que-es-para-que-sirve/

Morales, I., Ojeda, R., Borges, l., & Rodriguez, J. (06 de Junio de 2020). *Proceso del Desarrollo de Software*. Obtenido de Proceso del Desarrollo de Software: https://sites.google.com/site/iunesricomputacion/proseso-del-desarrollo-de-software

RED, M. E. (31 de Mayo de 2020). *Modulo Evaluación RED*. Obtenido de Modulo Evaluación RED: https://sites.google.com/site/moduloevaluacionred/modelo-de-calidad-boehm

Richards, M. (2015). *Software Architecture Patterns.* S/C: O'Reilly Media, Inc.

S/A. (06 de Junio de 2020). *CONTROL DE CALIDAD DEL SOFTWARE*. Obtenido de CONTROL DE CALIDAD DEL SOFTWARE: http://www.angelfire.com/my/jimena/ingsoft/calidad.htm

S/A. (06 de Junio de 2020). *GARANTIA DE CALIDAD DEL SFOTWARE*. Obtenido de GARANTIA DE CALIDAD DEL SFOTWARE: https://ingenieriasoft.webcindario.com/control-de-calidad-del-software/garantia-de-calidad-del-software.html

S/A. (06 de Junio de 2020). *INGENIERIA DE SOFTWARE*. Obtenido de INGENIERIA DE SOFTWARE: http://ingenieraupoliana.blogspot.com/2010/10/garantia-de-calidad-del-software.html

S/A. (30 de Mayo de 2020). *ISO 25000 CALIDAD DE SOFTWARE Y DATOS*. Obtenido de ISO 25000 CALIDAD DE SOFTWARE Y DATOS: https://iso25000.com/index.php/normas-iso-25000

S/A. (31 de Mayo de 2020). *Modelos de evaluación de los R.E.D. Grupo 8 UDES W*. Obtenido de Modelos de evaluación de los R.E.D. Grupo 8 UDES W:

https://modelos-de-evaluacion-de-los-red-grupo-8-udes.fandom.com/es/wiki/Modelo_de_Calidad_Boehm

S/A. (31 de Mayo de 2020). *Modelos de evaluación de los R.E.D. Grupo 8 UDES W*. Obtenido de Modelos de evaluación de los R.E.D. Grupo 8 UDES W: https://modelos-de-evaluacion-de-los-red-grupo-8-udes.fandom.com/es/wiki/Modelo_de_Calidad_Gilb

Salazar Hernández, W. E. (S/D de Febrero de 2017). IMPLEMENTACIÓN DE ARQUITECTURA DE MICROSERVICIOS UTILIZANDO VIRTUALIZACIÓN POR SISTEMA OPERATIVO. *IMPLEMENTACIÓN DE ARQUITECTURA DE MICROSERVICIOS UTILIZANDO VIRTUALIZACIÓN POR SISTEMA OPERATIVO*. Guatemal, S/P, Guatemala: S/E.

SOLANO, C. A. (s/d de Abril de 2019). ANÁLISIS COMPARATIVO ENTRE LOS ESTÁNDARES ORIENTADO A SERVICIOS WEB SOAP, REST Y GRAPHQL. *ANÁLISIS COMPARATIVO ENTRE LOS ESTÁNDARES ORIENTADO A SERVICIOS WEB SOAP, REST Y GRAPHQL.* Esmeraldas, Esmeraldas, Ecuador: s/e.

Wohlin, C., Host, M., Regnell, B., Runeson, P., Ohlsson, M., & Wesslen, A. (2012). Experimentation in Software Engineering. *Springer*, 5-231.