

ESCUELA DE INGENIERÍA EN SISTEMAS

Tema:

**MODELO DE EVALUACIÓN DE CALIDAD EN EL DESARROLLO DE
SOFTWARE PARA PERSONAS CON DISCAPACIDAD VISUAL**

**Proyecto de investigación previo a la obtención del
título de Ingeniera de Sistemas y Computación**

Línea de Investigación:

SISTEMAS DE INFORMACIÓN Y/O NUEVAS TECNOLOGÍAS DE LA
INFORMACIÓN Y COMUNICACIÓN Y SUS APLICACIONES

Autor: SÁNCHEZ JAYA VERÓNICA LUCÍA

Director: MG. DARÍO JAVIER ROBAYO JÁCOME

Ambato – Ecuador

Agosto 2020

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
SEDE AMBATO**

HOJA DEPROBACIÓN

Tema:

**MODELO DE EVALUACIÓN DE CALIDAD EN EL DESARROLLO DE
SOFTWARE PARA PERSONAS CON DISCAPACIDAD VISUAL**

Línea de Investigación:

**SISTEMAS DE INFORMACIÓN Y/O NUEVAS TECNOLOGÍAS DE LA
INFORMACIÓN Y COMUNICACIÓN Y SUS APLICACIONES**

Autor: VERÓNICA LUCÍA SÁNCHEZ JAYA

Darío Javier Robayo Jácome, Ing. Mg.

f. 

CALIFICADOR

Galo Mauricio López Sevilla, Ing. Mg.

f. 

CALIFICADOR

José Marcelo Balseca Manzano, Ing. Mg.

f. 

CALIFICADOR

Santiago Alejandro Acurio Maldonado, Ing. Mg.

f. 

DIRECTOR ESCUELA DE SISTEMAS

Hugo Rogelio Altamirano Villarroel

f. 

SECRETARIO GENARAL PUCESA

Ambato – Ecuador

Agosto 2020

DECLARACIÓN DE AUTENCIDAD Y RESPONSABILIDAD

Yo: **VERÓNICA LUCÍA SÁNCHEZ JAYA**, con **CC. 1803798212**, autora del trabajo de graduación intitulado: “MODELO DE EVALUACIÓN DE CALIDAD EN EL DESARROLLO DE SOFTWARE PARA PERSONAS CON DISCAPACIDAD VISUAL”, previa a la obtención del título profesional de **INGENIERA EN SISTEMAS Y COMPUTACIÓN**, en la escuela de INGENIERÍA EN SISTEMAS.

1. Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de la Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.
2. Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio *web* de la Biblioteca de la PUCE Ambato, el referido trabajo de graduación, respetando las políticas de propiedad intelectual de Universidad.

Ambato, agosto 2020



VERÓNICA LUCÍA SÁNCHEZ JAYA

CC.1803798212

AGRADECIMIENTO

A Dios por el regalo de la vida.

A la mujer que siempre estuvo apoyándome en la vida, mi madre: Margarita Jaya, y a mis hermanos: Fernando, David y Talía, por el apoyo incondicional.

A un gran amigo y excelente ser humano, que día a día con sus palabras, su apoyo y su presencia ha guiado mi vida, Padre Genaro Cordovilla.

A los Ingenieros: Darío Robayo, Marcelo Balseca y Galo López, quienes con paciencia me han brindado sus conocimientos, para la elaboración de este proyecto de investigación.

Y sin dejar atrás a cada una de las personas que en el trayecto de mi vida universitaria fueron un apoyo y aliciente para seguir adelante.

Verónica Lucía Sánchez

DEDICATORIA

A Margarita, Talía y Celeste

RESUMEN

El presente trabajo tiene como objetivo proponer un modelo de evaluación de calidad para el desarrollo del *software* orientado a personas con discapacidad visual. Para el estudio del mismo, se realizó una investigación exhaustiva en documentos de Google académico, Red CEDIA, Repositorio Digital UPCT, SciELO y otros.

Cada una de las investigaciones realizadas estuvo basada en el análisis profundo de su contenido parte desde lo que es un diseño de calidad de software, el desarrollo de software y software orientado a personas con discapacidad visual. Lo que lanzó como resultado que la mayoría de los modelos existentes para evaluar la calidad de *software* son genéricos, es decir, no son específicamente orientados a este tipo de personas.

En la elaboración del trabajo propuesto, se contrastó diversos modelos, normas y estándares de calidad en el desarrollo de software, así como también criterios desde la perspectiva de los desarrolladores de *software*, que se obtuvieron por medio de encuestas.

Se adaptó la Metodología en Cascada de manera empírica, de forma que, se pudieron adaptar tres fases para la elaboración del modelo, estas son: análisis, diseño y validación; en la primera, se analizaron las diferentes propuestas de los modelos de calidad, normas y estándares internacionales. En la segunda etapa, se puntualizaron los elementos factibles que contendrá el modelo. Por último, la validación del modelo fue realizada por desarrolladores de software y expertos en aplicaciones para personas con discapacidad visual. Además, se pudo aplicar el modelo para evaluar a nivel proceso software y aplicaciones orientados a personas no videntes, y a nivel producto fue evaluado por usuarios de las aplicaciones y software mencionados.

Palabras claves: modelo de calidad, desarrollo de software, discapacidad visual, métricas.

ABSTRACT

The aim of this study is to propose a quality assessment model for the development of software for people with a visual disability. A thorough study was conducted using documents in Academic Google, the CEDIA network, the UPCT digital repository, SciELO, among others. Each one of the investigations were based on a deep analysis of their content stemming from the concept of a software quality design, software development and software for people with visual disabilities. From the gathered information, it is noted that the models to evaluate software quality are generic, meaning that they are not specifically oriented toward people with this disability. In the development of the proposal, several models, regulations and quality standards in software development were compared, as well as the criteria of software developers which was gathered by survey. Waterfall methodology was applied empirically in order to adapt three phases for the development of the model-analysis, design and validation. In the first phase, several quality models, regulations and international standards proposals were analyzed. In the second phase, feasible elements that the model must include were pointed out. Finally, the validation of the model was carried out by software developers and applications for people with visual disabilities experts.

Key words: quality model, software development, visual disability, metrics.

ÍNDICE

DECLARACIÓN DE AUTENCIDAD Y RESPONSABILIDAD.....	iii
AGRADECIMIENTO	iv
DEDICATORIA	v
RESUMEN.....	vi
ABSTRACT	vii
ÍNDICE	viii
INTRODUCCIÓN.....	1
CAPÍTULO I. ESTADO DEL ARTE Y LA PRÁCTICA	3
1.1 Calidad en el desarrollo de Software.....	3
1.2 Modelo de Calidad	4
1.2.1 Objetivos de un Modelo de Calidad.....	4
1.3 Criterios de calidad	5
1.4 Modelos de calidad en el desarrollo de software	6
1.4.1 Modelos de Calidad a nivel de Proceso	6
1.4.1.1 CMMI (Capability Maturity Model Integration).....	7
1.4.1.2 ISO/IEC TR 15504 (Software Process Improvement and Capability Determination)	8
1.4.1.3 ISO /IEC 29110 (Software Process Improvement and Capability Determination)	9
1.4.2 Modelos de Calidad nivel Producto	10
1.4.2.1 McCall.....	10
1.4.2.2 BOEHM.....	12
1.4.2.3 ISO/IEC 9126: 2001 (International for Standardization).....	13
1.4.2.4 ISO/IEC 25000 (System and Software Quality Requirements and Evaluation).....	17
1.5 Software para personas con discapacidad visual	20
1.5.1 Criterios de calidad	21
1.5.1.1 Accesibilidad.....	21
1.5.1.2 Usabilidad	22
CAPÍTULO II. DISEÑO METODOLÓGICO.....	22
2.1 Tipo de investigación	22
2.1.1 Búsqueda bibliográfica.....	22

2.1.2	Análisis-síntesis	22
2.1.3	Encuesta.....	23
2.1.3.1	RESULTADO DE LA APLICACIÓN DE LA ENCUESTA A DESARROLLADORES DE LA CIUDAD DE AMBATO.....	24
2.2	Metodología de Desarrollo	29
2.2.1	Adaptación de la Metodología	30
	CAPÍTULO III. RESULTADOS	31
	Fuente: elaboración propia	53
	CONCLUSIONES	55
	RECOMENDACIONES	56
	ANEXOS.....	61
	Anexo 1. Encuesta para desarrolladores.....	61
	Anexo 2. Encuesta de Satisfacción	63
	Anexo 4. Validación Matriz.....	69

ÍNDICE DE TABLAS

Tabla 1: Características de los modelos de calidad.....	20
Tabla 2. Factores y Métricas	25
Tabla 3. Factores de Calidad	31
Tabla 4. Selección de factores de calidad	32
Tabla 5. Análisis de los factores de calidad de las encuestas aplicadas	33
Tabla 6. Modelo de evaluación de calidad en el desarrollo de software para personas con discapacidad visual.....	38
Tabla 7. Cuadro Lógico de ladov ISI (índice satisfacción individual)	42
Tabla 8. Cuadro Lógico de ladov aplicado a desarrolladores	43
Tabla 9. Cuadro Lógico de ladov aplicado a desarrolladores	44

ÍNDICE DE FIGURAS

Figura 1. Criterios de Calidad.....	5
Figura 2. Tipos de modelos de calidad.....	6
Figura 3. Áreas de proceso por niveles.....	7
Figura 4. División de la ISO/IEC 29110.....	9
Figura 5. Modelo de Calidad Interna y Externa.....	14
Figura 6. Organización ISO 25000.....	17
Figura 7. Modelo de Calidad del Producto Software.....	19
Figura 8. Diseño de modelo de calidad.....	33
Figura 9. Índice de satisfacción individual y grupal.....	45
Figura 10. Escala de valores del ISG.....	45

ÍNDICE DE GRÁFICOS

Gráfico 1. Metodología de desarrollo	24
Gráfico 2. Modelo de aseguramiento de calidad	25
Gráfico 3. Responsable de evaluación	26
Gráfico 4 .Software personas discapacidad visual	27
Gráfico 5. Control de calidad.....	28

INTRODUCCIÓN

Los modelos de calidad han sido creados debido a la necesidad que tienen las empresas de desarrollo de *software* de una herramienta que les permita mejorar la calidad de sus productos. Scalone (2006), cuando hace referencia a los modelos y/o estándares de calidad del *software*, sostiene que sirven de apoyo cuando, se pone en práctica el concepto general de calidad, pues proporcionan una definición más operacional. Además, menciona que, dentro de estos modelos, uno de los primeros es el de McCall, del cual, se han originado otros modelos, como el modelo de Boehm. Por su parte, Nuviala, Tamayo Fajardo, Fernández Martínez, & Pérez Turpin (2011), al hacer mención a la calidad de *software*, sostienen que ésta alude a las diversas cualidades que determinan su beneficio: es el nivel en que el *software* lleva a efecto las condiciones determinadas. Mientras Estévez (2014), manifiesta que, desde que hicieron su aparición las diversas tecnologías del *software*, se ha tratado de hacer una evaluación de la calidad de los propios productos, se tiene como meta el mejoramiento de cada una de sus características con el fin de obtener productos *software* más cualificados que complazcan las exigencias de los clientes.

En los Modelos de Calidad, esta es definida de manera jerárquica, una definición derivada de una serie de subconceptos, los mismos que son evaluados por medio de una escala de indicadores o métricas. Puesto que un modelo de calidad de *software* representa las necesidades y requerimiento propios de los usuarios finales, cada modelo estará asociado a un conjunto de características, las cuales definen las áreas claves a satisfacer para lograr, asegurar y controlar la calidad tanto en el producto como en el proceso.

En los últimos años, debido al avance tecnológico, se ha incrementado el desarrollo de productos de *software*, los mismos en muchos casos, han sido generados en función de modelos de calidad genéricos, no orientados específicamente a ciertos tipos particulares de *software*. Ante esta realidad, se tiene presente el artículo 16 de la Constitución de la República del Ecuador que el avance tecnológico de ninguna forma excluirá a persona, de las ventajas que el mismo ofrece, se propone un Modelo de evaluación de calidad en el Desarrollo de *Software* para personas con

discapacidad visual, con el fin de brindar los mejores servicios en el avance de la tecnología a este grupo específico.

El contar con un modelo orientado a ese tipo de *software*, permitirá alcanzar productos que incorporen características de accesibilidad y usabilidad para las personas con discapacidad visual, para así ayudarles con una herramienta útil para que puedan desenvolverse mejor en su vida cotidiana.

Mediante la investigación del tema, se observa que actualmente no existe un Modelo de Evaluación de Calidad en el Desarrollo de *Software* para personas con discapacidad visual, a causa de varios factores, uno de ellos es la poca importancia, que se da al desarrollo de *software* orientado a este grupo de personas, debido a que no reciben todos los beneficios adecuados que la ofrece la tecnología. Además, como se mencionó anteriormente, los modelos existentes de evaluación del *Software* son genéricos, por tanto, esto hace que no posean los lineamientos necesarios y específicos para su desarrollo y evaluación.

A partir del problema planteado, se traza como objetivo general: Proponer un modelo de evaluación de calidad en el desarrollo de software orientado a personas con discapacidad visual, basado en los siguientes objetivos específicos: sustentar teóricamente los aspectos relacionados a la evaluación de calidad en el desarrollo del software dirigido a personas con discapacidad visual, contrastar modelos de evaluación de calidad utilizados en el desarrollo de software dirigido a personas con discapacidad visual para la identificación de sus fortalezas y debilidades, diseñar los componentes del modelo de software dirigido a personas con discapacidad visual, validar el Modelo por expertos.

La investigación, se realiza utilizando la técnica de investigación bibliográfica, la cual está basada en el análisis de documentos de Google Académico, Red CEDIA, Repositorio Digital UPCT, SciELO con el fin de recopilar la información necesaria para el presente proyecto en donde, se ubicaron varios documentos referentes al tema, los mismos que fueron analizados de forma particular. Asimismo, se empleó el método analítico sintético que como lo menciona Delgado (2010), es indispensable en el estudio de las cuestiones históricas analizar los sucesos descomponiéndolos en todas sus partes para conocer sus posibles raíces. Este método servirá para estudiar minuciosamente cada uno de los requerimientos necesarios para aplicarlos en el siguiente proyecto de investigación.

CAPÍTULO I. ESTADO DEL ARTE Y LA PRÁCTICA

1.1 Calidad en el desarrollo de *Software*

En el transcurso de su evolución, el *software* ha tenido avances por demás significativos, ha llegado a convertirse en una gran industria, es de ayuda a quienes lo desarrollan a alcanzar programas de un mantenimiento fácil y escalable. Sommerville Sommerville (2005), menciona que es muy importante tener cuidado y mantenimiento especial en el *software*, desde el instante de su elaboración hasta el momento en que los usuarios adquieren finalmente el producto. Así mismo, señala que la Ingeniería de *Software* no abarca solamente procesos técnicos sino también, diferentes aspectos tales como la gestión de proyectos de *software* y la creación de teorías, métodos y herramientas de apoyo, al momento de su elaboración. De manera similar, en el artículo Modelos y Métricas para Evaluar calidad de *software*, de Estayno, Dapozo , Cuenca Pletsch, Liliana Raquel, & Greiner (2009), se manifiesta que los Modelos y Métricas para Evaluar la calidad de *software*, “son el conjunto de factores de calidad, y de relaciones entre ellos, que proporciona una base para la especificación de requisitos de calidad y para evaluación de la calidad de los componentes de *software*” (p. 382-388). Por consiguiente, el definir un modelo de calidad durante el desarrollo de *software* permitirá que aumente el nivel de satisfacción de los clientes y/o usuarios.

Por su parte, Calero, Moraga, & Piattini, Calidad del producto y proceso software (2010), indican que desde el punto de vista del *software* se consigue distinguir proceso, producto y recursos, donde la calidad de los dos primeros son los que cobran mayor importancia. Para Peñalva (2014), el modelo de desarrollo de *software* incluye mejorar la vida del usuario en la intercomunicación con el *software* alcanzando un grado de satisfacción en utilidad, productividad y confianza. A lo que, la ISO/IEC (25000:2015), señala que es el grado en que un sistema de *software* satisface las necesidades de los usuarios. Asimismo, Callejas, Alarcón, & Álvarez (2017), señalan que un modelo de calidad de *software*, se refiere al nivel de ejecución de las principales características con las que obligatoriamente cumple un sistema computacional durante su ciclo de vida. De las definiciones mencionadas, se determina que un modelo de calidad en el desarrollo de *software*, se refiere al nivel

de cumplimiento de las principales características durante su ciclo de vida, para brindar el mejor servicio a los usuarios. Mientras que, Callejas, Alarcón, & Álvarez (2017), mencionan que un modelo de calidad de *software*, se refiere al nivel de ejecución de las principales características con las que necesariamente cumple un sistema computacional durante su ciclo de vida. A criterio del autor un modelo de calidad de *software* describe las características que conforman la calidad del mismo, y tienen varios fines que será posible comprender a través de las definiciones dadas por varios autores que aportan significativamente para el desarrollo del presente proyecto.

1.2 Modelo de Calidad

Gálaz (2014), acota que el modelo de calidad sirve, también, para que las organizaciones puedan realizar sus autoevaluaciones, a través de una revisión integral de sus estrategias y prácticas de gestión. Mientras que, Pérez & Luque (2018), menciona que un modelo de calidad es un conjunto de características y las relaciones entre ellas, que proporciona un marco para la especificación de requisitos de la calidad y su evaluación. Además, la Asociación Española para la Calidad QAEC, (2018), indica que los modelos contienen directrices para mejorar el cumplimiento de la calidad de un sistema y no requisitos, así mismo, indica que las organizaciones desarrollarán modelos de calidad orientados a la mejora del proyecto.

1.2.1 Objetivos de un Modelo de Calidad.

ISO/IEC 25000 menciona que uno de los objetivos de un modelo de calidad es:

“Guiar la evaluación de calidad de producto *software* estableciendo criterios para la especificación de requisitos de calidad de *software*, sus métricas y su evaluación”.

Según Gómez J. (2015), la ISO/IES (9001:2015),

Los objetivos de calidad son metas, retos, que se definen a partir de la planificación estratégica de la empresa y de su política de calidad.

Se escogerán aquellos objetivos de calidad que van más en el avance de las políticas de calidad.

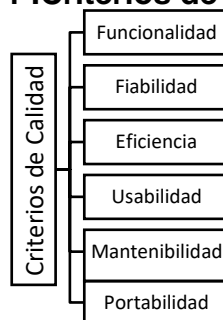
Los objetivos de calidad serán establecidos por la alta dirección de la organización. Tienen que ser coherentes con la política de calidad y perseguir la mejora continua. Según la norma ISO los objetivos de la calidad son:

- Ser similares con la política de la calidad.
- Ser medibles.
- Considerar los requisitos aplicables.
- Ser apropiados para la aprobación de los productos y servicios y para el acrecentamiento del gusto del cliente.
- Ser ente de seguimiento.
- Ser comunicados.
- Actualizarse, según sea necesario.

1.3 Criterios de calidad

Los criterios de calidad son aspectos o condiciones que un producto cumple para alcanzar un nivel adecuado de eficiencia, funcionalidad, y grado de satisfacción de los usuarios, así, en la ISO 9000:2015, se consideran los siguientes criterios:

Figura 1 .Criterios de Calidad



Fuente: ISO 9000:2015

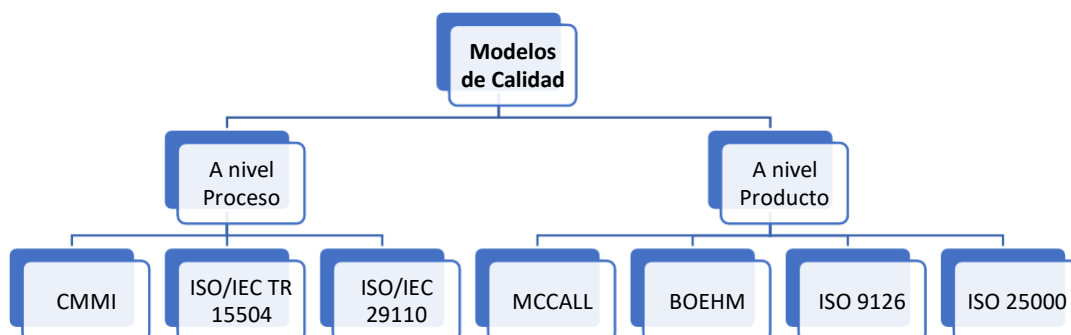
- **Funcionalidad:** que el *software* cumpla con los requerimientos.
- **Fiabilidad:** el *software* no tenga fallos.
- **Eficiencia:** optimización de recursos del sistema.
- **Usabilidad:** facilidad en la que el *software* será entendido y usado.

- **Mantenibilidad:** facilidad con la que el *software* soportará cambios y/o actualizaciones.
- **Portabilidad:** que el *software* sea adaptable a otras plataformas de hardware y *software*.

1.4 Modelos de calidad en el desarrollo de *software*

Estayno, Daposo, Cuenca, & Greiner (2009) y Callejas, Alarcón, & Álvarez (2017), mencionan que la calidad del *software* es una compleja mezcla de componentes y a partir de las especificaciones presentan los tipos de modelos de calidad en el desarrollo de *software*:

Figura 2. Tipos de modelos de calidad



Fuente: Estayno, Daposo, Cuenca, & Greiner (2009); Callejas, Alarcón, & Álvarez (2017)

Autor: elaboración propia

1.4.1 Modelos de Calidad a nivel de Proceso

Callejas, Alarcón, & Álvarez (2017), menciona que la calidad del *software* será programada desde el inicio del proyecto, y posteriormente en cada etapa del proceso de desarrollo, para garantizar un óptimo nivel del cumplimiento de los

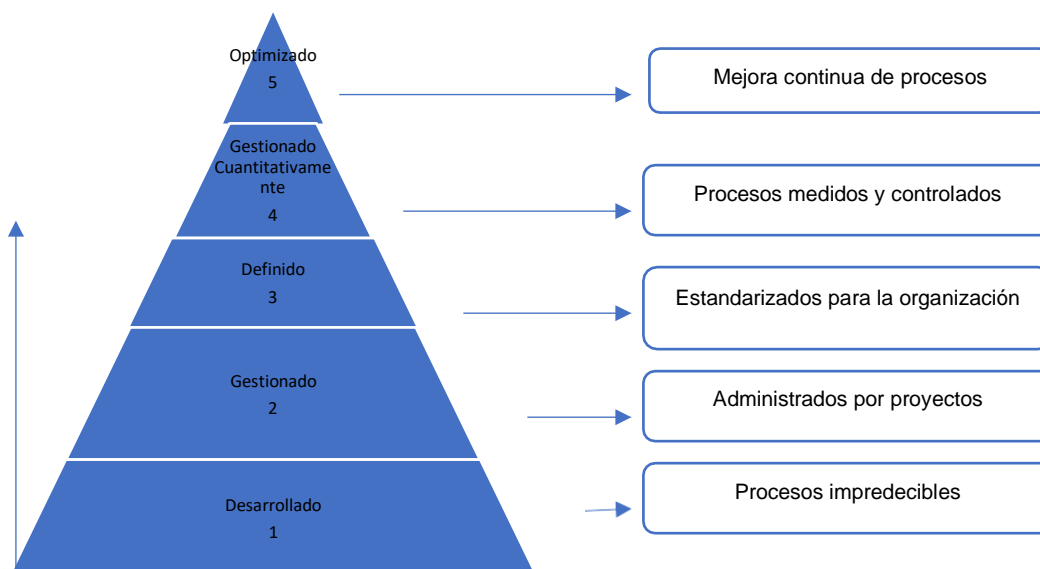
factores de calidad en la evolución del mismo. Los modelos de calidad a nivel de proceso son:

1.4.1.1 CMMI (Capability Maturity Model Integration)

Según Tuya, Ramos, & Dolado (2007) y Yepes, César, & Gómez (2015) coinciden que CMMI es un modelo que integra la arquitectura, enfoque, contenido y aplicación para obtener productos de calidad; además, mencionan que este modelo contiene un enfoque disciplinado para mejorar los procesos de una organización y para que, se puedan implantar procesos de calidad. Enfatizando lo antes mencionado, el modelo CMMI ayudará a determinar de mejor manera los procesos esenciales para optimizar el proceso del *software* y obtener productos de calidad.

Madrid (2008), indica que CMMI tiene cinco niveles de madurez los que, se observan a continuación (Ver Figura 3):

Figura 3. Áreas de proceso por niveles



Fuente: elaboración propia

1. Inicial: en esta etapa inicial según Puello (2009), en este nivel las organizaciones no cuentan con un ambiente estable para el desarrollo y mantenimiento de

software. El éxito de los proyectos, se fundamenta la mayor parte de las ocasiones en el esfuerzo personal.

2. Administrado: “En este nivel las organizaciones disponen de prácticas institucionalizadas de gestión de proyectos, existen métricas básicas y un razonable seguimiento de la calidad. La relación con subcontratistas y clientes está gestionada sistemáticamente”. (Esterkin & Pons, 2017, p.449-463).

3. Definido: En este nivel, las organizaciones cuentan con correctos funcionamientos, técnicas de ingeniería más precisa y un nivel más avanzado de métricas en los procesos.

4. Administrado Cuantitativamente: “Las organizaciones disponen de un conjunto de métricas significativas de calidad y productividad, que se usan de modo sistemático para la toma de decisiones y la gestión de riesgos”. (Chacon , Rodas, & Vinueza, 2015).

5. Optimizado: “La organización completa está orientada a la mejora continua de los procesos. Se hace uso intensivo de las métricas y, se gestiona el proceso de innovación”. (Chacon , Rodas, & Vinueza, 2015).

1.4.1.2 ISO/IEC TR 15504 (Software Process Improvement and Capability Determination)

Loon (2004), menciona que es un modelo que, se centra en la realización y cumplimiento de los procesos de desarrollo, mantenimiento de sistemas de información y productos de *software*; igualmente, menciona que este modelo contiene los requisitos necesarios para garantizar evaluaciones de procesos consistentes; y, que muchas de las organizaciones lo utilizan para evaluar la capacidad de sus proveedores.

De la misma forma, la Asociación Española para la Calidad – QAEC, señala que:

está diseñada para facilitar una aproximación común para realizar valoraciones de procesos, haciéndose posibles comparaciones entre los resultados de las mismas. Estos resultados, se basarán en diferentes modelos de valoración (siempre que sean compatibles con el estándar) y métodos de valoración.

Destacando lo mencionado por los autores, es necesario tener en cuenta este modelo porque permite realizar valoraciones con diferentes modelos y garantiza que las evaluaciones de los procesos sean estables.

1.4.1.3 ISO /IEC 29110 (*Software Process Improvement and Capability Determination*)

Yepes, César, & Gómez (2015), indican que la ISO/IEC 29110 es un estándar que está encaminado a las pequeñas y medianas organizaciones, con el fin de alcanzar niveles internacionales de competitividad. Además, busca mantener sus productos con atributos de calidad tanto en el proceso como en el producto. Igualmente, indica que es una norma que, se desarrolló con el fin de mejorar la calidad del producto y/o servicio de *software* mejorando el desempeño de la organización.

Madruñero (2018), indica que:

Es una norma ISO que lleva como denominación Ingeniería de *Software* - Perfiles de Ciclo de Vida en Pequeñas Entidades, esta norma es basada en las siguientes normas: mexicana NMX-I-059-NYCE-2011 (MoProSoft), en la ISO/IEC 12207, la ISO/IEC 15289, la ISO/IEC 15504 entre otras. La propuesta mexicana fue ofrecer sus procesos de la categoría operación como un perfil básico, la categoría de gerencia como perfil intermedio y la categoría alta Dirección como perfil Avanzado (p. 47).

Figura 4. División de la ISO/IEC 29110

ISO/IEC 29110	Título	Público objetivo
Parte 1	Descripción general	Microempresas, clientes, asesores, productores de normas, proveedores de herramientas y de metodología.
Parte 2	Marco y taxonomía	Productores de normas, proveedores de herramientas y de metodología. No está diseñado para microempresas.
Parte 3	Guía de evaluación	Asesores, clientes y microempresas
Parte 4	Especificaciones del perfil	Productores de normas, proveedores de herramientas y de metodología. No está diseñado para microempresas.
Parte 5	Gestión y guía de ingeniería	Microempresas y clientes

Fuente: Madruñero (2018)

Parte 1. Descripción general: dirigido a MPO, evaluadores, productores de estándares, vendedores de herramientas y vendedores de metodologías.

Parte 2. Marco de trabajo y taxonomía: dirigido a productores de estándares, vendedores de herramientas y vendedores de metodologías. No está dirigido para la MPO.

Parte 3. Guía de evaluación: está dirigido a evaluadores y MPO.

Parte 4. Especificaciones de perfil: dirigido a productores de estándares, vendedores de herramientas y vendedores de metodologías. No está dirigido para la MPO.

Parte 5. Guía de gestión e ingeniería: está dirigido a MPO.

El autor menciona algunas de las características con las que cuenta este estándar:

- Especifica el desarrollo y mantenimiento de *software*.
- Se encuentra dirigido para las pequeñas organizaciones que cuenta entre 1 a 25 personas laborando.
- Se encuentra orientado a proyecto internos como externos.
- Se certifica de conformidad con vigencia de 3 años y vigilancia anual.

1.4.2 Modelos de Calidad nivel Producto

Calero, Moraga, & Piattini, Calidad del producto y proceso software (2010), indican que la finalidad del modelo de calidad a nivel de producto es especificar y evaluar el cumplimiento de criterios del producto, para lo cual, se aplican medidas internas y/o medidas externas.

1.4.2.1 McCall

Medina (2014), indica que este modelo aparece en la Fuerza Aérea Estadounidense, que lo utilizó en los productos de *software* del Departamento de Defensa en 1977. McCall, se basa en once factores de calidad los mismos que están organizados en tres ejes, donde el usuario examinará la calidad del producto. De la misma manera, Jorge, Carlos , Jorge , & María (2008), señala que este modelo desglosa el concepto de calidad en tres usos importantes para un producto de *software*.

Según ambos autores los ejes o usos son:

Operación del Producto: se refiere a las características de operación y los factores de calidad que integran este apartado son:

- Facilidad de Uso, por parte de los usuarios del sistema
- Integridad, para proteger al programa de accesos que no han sido autorizados
- Eficiencia, en la ejecución del programa y en la utilización de recursos por parte del mismo.
- Corrección o exactitud
- Fiabilidad, que el sistema no falle.

Revisión del producto: es la habilidad para ser cambiado e incluye los siguientes factores:

- Facilidad de prueba, asegurar que el programa esté libre de errores y conoce las especificaciones del usuario.
- Facilidad de Mantenimiento, esfuerzo requerido para encontrar y solucionar errores que, se presenten en la operación del sistema.
- Flexibilidad, facilidad de realizar cambios.

Transición del Producto: describe la adaptación al nuevo ambiente e incluye los siguientes factores de calidad:

- Reusabilidad, se vuelve a usar el *software*.
- Portabilidad, capacidad de transferir un programa de un ambiente a otro.
- Interoperabilidad, se logra unir a un sistema con otro.

Según Arenas (2015), la lista de factores que determinan la calidad en el desarrollo de *software* en el modelo de McCall es:

- **Corrección:** mide el grado en que el *software* cumple lo especificado por el usuario.
- **Fiabilidad:** mide el grado de precisión del *software* y si este cumple con las funciones requeridas.
- **Eficiencia:** mide la cantidad de recurso de la computadora, así como el código que son necesarios para que, se realicen las funciones del *software*.
- **Integridad:** es el grado de seguridad que, se le da al *software*.
- **Facilidad de Uso:** esfuerzo necesario para aprender un programa y, se pueda interpretar la información.

- **Facilidad de Mantenimiento:** facilidad de localizar y corregir programas.
- **Facilidad de Prueba:** esfuerzo requerido para transportar el *software* a otro entorno.
- **Flexibilidad:** esfuerzo requerido para modificar un sistema operativo.
- **Portabilidad:** esfuerzo requerido para transferir un *software* de un hardware a un entorno de sistema a otro.
- **Reusabilidad:** grado en que, se vuelve a utilizar el *software* en otro.
- **Facilidad de Interoperación:** esfuerzo requerido para asociar un programa a otro.

1.4.2.2 BOEHM

Este modelo fue desarrollado por Barry Boehm en el año de 1978; el factor principal en el que, se centra el modelo es que el *software* realice lo que el usuario desee que realice, el *software* tiene que utilizar correcta y eficazmente los recursos del computador:

- Tiene que ser de fácil de uso y aprendizaje para los usuarios.
- Estar bien diseñado, codificado; ser probado y mantenido fácilmente.
- En su estructura, el modelo Boehm, presenta tres niveles para las características: de alto nivel, de nivel intermedio y características primitivas. Cada una de estas características contribuye a tener un nivel general de calidad.

Scalone (2006), presentan tres niveles las características de la calidad del *software*:

Características de alto nivel:

- Utilidad per-se: qué tan eficiente, usable y confiable es el producto en sí mismo.
- Mantenibilidad: qué fácil es entender, modificar y retestearlo.
- Utilidad general: en qué grado seguirá utilizándose si, se cambia el entorno.

Características de nivel intermedio:

- portabilidad (utilidad general),
- confiabilidad (utilidad per-se),

- eficiencia (utilidad per-se¹),
- usabilidad (utilidad per-se),
- testeabilidad (mantenibilidad),
- facilidad de entendimiento (mantenibilidad),
- modificabilidad o flexibilidad (mantenibilidad).

Características Primitivas:

- **de portabilidad:**
 - independencia de dispositivos,
 - auto-contención.
- **de confiabilidad:**
 - auto-contención,
 - exactitud,
 - completitud,
 - consistencia,
 - robustez/integridad.

1.4.2.3 ISO/IEC 9126: 2001 (International for Standardization)

Según Carreón (2008):

El ISO/IEC 9126-1 está definido por características generales de *software*, que son refinadas en subcaracterísticas; que a su vez, se descomponen en atributos, pertenecientes a una jerarquía multinivel; pueden aparecer jerarquías de atributos y subcaracterísticas. Al final de la jerarquía aparecen los atributos de medición, quienes dan los valores para ser usados en la métrica (p.36).

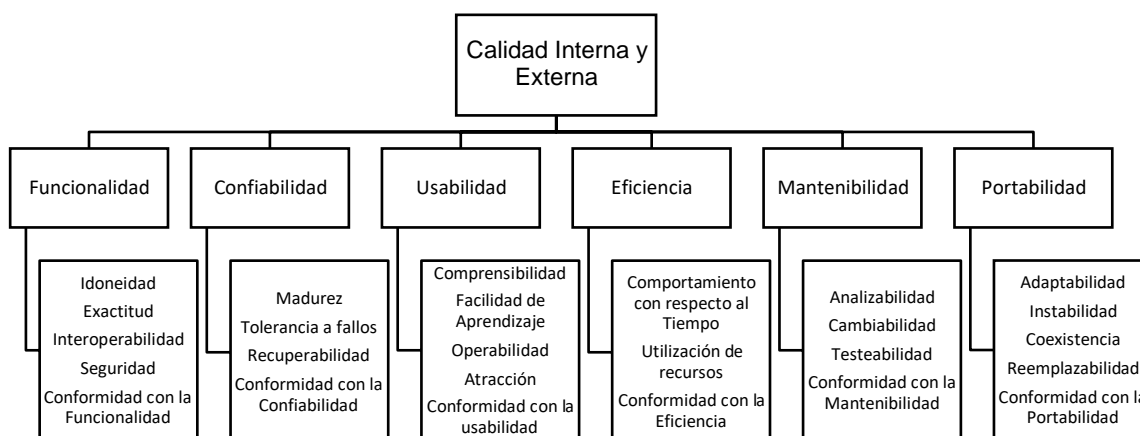
Domínguez (2016), menciona que este modelo de calidad contiene 6 características que evalúan la calidad interna y externa. La calidad externa evalúa el comportamiento del producto, toma en cuenta las especificaciones del usuario, mientras que la calidad interna evalúa el total de atributos que un *software* abarca para satisfacer teniéndose en cuenta condiciones especificadas. Por tanto, los criterios y especificaciones que presenta la ISO para realizar la evaluación de un producto,

¹ Por sí mismo, de por sí.

indica que es necesario tomar en cuenta que medidas aplicar en las métricas para el desarrollo del presente proyecto.

En el siguiente gráfico, se observan los criterios que la norma ISO: 9126 toma en cuenta para medir la calidad del producto.

Figura 5. Modelo de Calidad Interna y Externa



Fuente: (ISO/IEC 9126-1, 2001)

- **Funcionalidad:** certificar que el producto marche de acuerdo a las especificaciones del usuario.
 - **Exactitud:** Capacidad del producto *software* para proveer resultados adecuados con un preciso nivel de puntualidad.
 - **Interoperabilidad:** Capacidad del producto para comunicarse con otros sistemas.
 - **Seguridad:** Capacidad del producto *software* para proteger información.
 - **Conformidad con la Funcionalidad:** “Capacidad del producto *software* para adherirse a normas, convenciones o regulaciones en leyes y prescripciones similares relacionadas con la funcionalidad”.
- **Confiabilidad:** El *Institute of Electrical and Electronic Engineers* IEEE (2007), define a la confiabilidad como “la habilidad que tiene un sistema o componente de realizar sus funciones requeridas bajo condiciones específicas en período de tiempo determinados” (pág. 8). Musa (2002), indica que es “la probabilidad o la capacidad de que un sistema de funciones

trabaje sin falla en un periodo de tiempo y bajo condiciones o un medio ambiente específico”

- **Madurez:** Habilidad del producto *software* para prescindir fallos en el *software*.
 - **Tolerancia a fallos:** Adaptación del *software* para conservar un grado especificado de prestaciones en caso de fallos *software* o de vulnerar sus interfaces.
 - **Capacidad de recuperación:** Idoneidad del producto *software* para restituir un cierto grado de beneficios y de readquirir los datos concisamente afectados en caso de fallo.
 - **Cumplimiento de la fiabilidad:** Potencial del producto *software* para asociarse a normas, convenios o medidas relacionadas con la fiabilidad.
- **Usabilidad:** Según la IEEE “es la facilidad con la que un usuario aprende a operar, realizar inputs e interpretar outputs de un sistema”. La norma ISO (9241:2008), menciona que “es la actividad, eficiencia y satisfacción con la que usuarios específicos alcanzarán objetivos específicos en un medio particular”.
- **Capacidad para ser entendido:** Cabida del producto *software* que admite al usuario poder entender si el *software* es correcto y si éste consigue ser aplicado a otros trabajos.
 - **Capacidad para ser aprendido:** Habilidad del producto *software* que permite al usuario aprender acerca de su aplicación.
 - **Capacidad para ser administrado:** Habilidad del producto *software* que proporciona que el usuario pueda administrarlo y controlarlo.
 - **Capacidad de ser atractivo:** Cabida del producto *software* para ser atrayente al usuario.
 - **Cumplimiento de la usabilidad:** Capacidad del producto *software* para adherirse a normas, convenciones, guías de estilo o regulaciones relacionadas con la usabilidad.
- **Eficiencia:** Capacidad del producto *software* para proporcionar prestaciones apropiadas, relativas a la cantidad de recursos usados, bajo condiciones determinadas

- **Comportamiento temporal:** Facultad del producto *software* para proveer tiempos de respuesta y de proceso de respuesta al ejecutar sus funciones dentro de diferentes contextos.
- **Utilización de recursos:** Habilidad del producto *software* para utilizar las cantidades y tipos de recursos apropiados cuando el *software* es ejecutado en determinados escenarios.
- **Cumplimiento de la eficiencia:** Aptitud del producto *software* para adherirse a criterios o convenciones relacionadas con la efectividad.
- **Mantenibilidad:** Se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al *software*, ya sea por la corrección de errores o por el incremento de funcionalidad.
 - **Capacidad de ser analizado:** Capacidad del producto *software* que admite implementaciones específicas.
 - **Estabilidad:** Aprueba valorar los conflictos de imprevistos a causa de las modificaciones.
 - **Capacidad para ser probado:** es el empeño necesario para evaluar el funcionamiento del *software* después de ser modificado.
- **Portabilidad:** capacidad del *software* para ser transportado de un ambiente a otro.
 - **Adaptabilidad:** Habilidad que tienes el producto *software* para adaptarse a otros entornos.
 - **Instalabilidad:** Facilidad del producto *software* para ser instituido en un cierto entorno.
 - **Coexistencia:** Competitividad del producto *software* para entenderse con otro *software* independiente.
 - **Capacidad para reemplazar:** Facilidad del producto *software* para ser usado en lugar de otro.
 - **Cumplimiento de la portabilidad:** Operatividad del producto *software* para incorporarse a reglas o convenios relacionados con la portabilidad.

El estándar ISO 9126 presenta detalladamente una guía para el mejoramiento tanto en el desarrollo como en el producto final del *software*, por ello, es importante que las empresas u organizaciones que, se dedican al desarrollo de *software* tengan un modelo para evaluar la calidad de sus productos.

1.4.2.4 ISO/IEC 25000 (System and Software Quality Requirements and Evaluation)

Laporte (2017), indica que el objetivo de esta norma es crear un marco de trabajo para la calidad de productos de *software*, y este estándar es el progreso de la ISO 9126. Está dividida en 5 estándares.

Figura 6. Organización ISO 25000



Fuente: ISO/IEC 25000:2015

Abuchar (2015), explica los diferentes apartados de esta norma:

- **ISO/IEC 2500n División de Gestión de Calidad**
 - **ISO/IEC 25000 Guía de SQuaRE:** abarca el modelo de la arquitectura de SQuaRE, la terminología, una síntesis de las partes, los usuarios predichos y las partes asociadas y los modelos de referencia.
 - **ISO/IEC 20001 Planificación y Administración:** establece los requerimientos y orientaciones para gestionar la evaluación.

- **ISO/IEC 2501n – División de Modelo de Calidad**

Presenta las características específicas internas, externas y en uso del producto.

- **ISO/IEC 25010 – Modelos de Calidad del Software y Sistemas:** detalla el modelo de calidad para el producto *software* y para la calidad en uso.
- **ISO/IEC 25012 – Modelo de Calidad de los Datos:** precisa un modelo general para la calidad de los datos, adaptable a los datos que estén recopilados y formen parte de un Sistema de Información.
- **ISO/IEC 2502n División de Medición de Calidad**

Abarcan un modelo de referencia de la medición de la calidad del producto, definiciones de medidas de calidad a nivel interno, externo y en uso, y pautas prácticas para su aplicación.
- **ISO/IEC 25020 Guía y Modelo de Referencia de Mediciones:** muestran una explicación y un modelo de referencia común a los elementos de medición de la calidad.
- **ISO/IEC 25021 Elementos de Métricas de Calidad:** delimita un conjunto de métricas base y derivadas que puedan ser usadas a lo largo de todo el ciclo de vida del *software*.
- **ISO/IEC 25022 – Métricas de Calidad en Uso:** puntualiza las métricas para realizar la medición de la calidad en uso del producto.
- **ISO/IEC 25023 Métricas de Calidad del Producto y Sistema de Software:** establece las métricas para realizar la exactitud de la calidad de productos y sistemas *software*.
- **ISO/IEC 25024 Métricas de Calidad de los Datos:** implanta las métricas para realizar la medición de la calidad de datos que son parte de un Sistema de Información y están almacenados de manera estructurada.
- **ISO/IEC 2503n División de Requerimientos de Calidad estas normas ayudan a** definir requerimientos de calidad que serán utilizados en el proceso de elicitación de requerimientos de calidad del producto de *software* a desarrollar o como entrada del proceso de evaluación.
 - **ISO/IEC 25030 Requerimientos de Calidad:** provee un conjunto de recomendaciones para realizar la determinación de los requerimientos de calidad del producto de *software*.
- **ISO/IEC 2504n División de Evaluación de Calidad**

Contiene normas que facilitan requisitos, recomendaciones y pautas para realizar el proceso de evaluación del producto *software*.

- **ISO/IEC 25040 Guía y Modelo de Referencia de la Evaluación:** plantea un modelo de referencia general para la evaluación, que considera las entradas al proceso de evaluación, las restricciones y los recursos que son ineludibles para alcanzar salidas oportunas.
- **ISO/IEC 25041 Guía de Evaluación para los Desarrolladores, Compradores y Evaluadores Independientes:** desde el punto de vista de los desarrolladores detalla los requisitos y recomendaciones para la implementación práctica de la evaluación del producto *software*.
- **ISO/IEC 25042 Módulos de Evaluación:** define el Módulo de Evaluación y la documentación, estructura y contenido que, se utilizan a la hora de definir los módulos.
- **ISO/IEC 25045 – Módulo de Evaluación para la Recuperabilidad:** delimita un módulo para la evaluación de la subcaracterística Recuperabilidad.

Figura 7. Modelo de Calidad del Producto Software



Fuente: ISO/IEC 25010

A continuación, se muestra un cuadro de resumen de características de los modelos de calidad.

Tabla 1. Características de los modelos de calidad

Modelo	Descripción	Características
A nivel de Proceso		
CMMI	Contiene directrices para obtener productos de calidad.	Verifica el cumplimiento de estándares de calidad. Se representa en dos maneras: escalonada y continua.
ISO/IEC TR 15504	Contiene los requisitos necesarios para garantizar evaluaciones de procesos consistentes.	Organizado en cinco niveles de capacidad.
ISO/IEC 29110	Es una norma que está dirigida a pequeñas organizaciones. Mejorar la calidad del producto y del servicio de <i>software</i> .	Dirigido a pequeñas y medianas organizaciones. Se divide en 5 partes que contienen directrices de aplicación para alcanzar los niveles de madurez.
A nivel de Producto		
MCCALL	El usuario consigue examinar la calidad del producto a través de los once factores de calidad.	Organizado en 3 ejes que contienen 11 factores de calidad. Los 3 ejes son: Operación del producto. Revisión del producto y Transición del producto.
BOEHM	El <i>software</i> realiza lo que el usuario desea que, se realice.	Es un modelo incremental dividido en regiones de tareas. Comprende características de alto nivel, características nivel intermedio y características primitivas.
ISO 9126	Establece una guía para la evaluación de la calidad del <i>software</i> .	Compuesto por seis características que evalúan la calidad del <i>software</i> interna y externamente.
ISO 25000	Crea un escenario de trabajo para evaluar la calidad del producto <i>software</i> .	Guía al desarrollo de <i>software</i> permite la organización, el incremento y combinación de las series que comprenden la especificación de requisitos de calidad del <i>software</i> .

Fuente: elaboración propia

1.5 Software para personas con discapacidad visual

En Ecuador, como señalan Martínez , Siva & Naranjo (2016), en los últimos años, se ha tratado de incluir a las personas con discapacidad visual en el ámbito de la

tecnología; esto ha provocado que, tanto en las instituciones educativas, como a nivel comercial, se generen proyectos de aplicaciones o tecnologías que permitan ayudar a las personas no videntes.

La ayuda de las nuevas tecnologías, como destaca Ortiz (2009), son una herramienta para integrar a las personas invidentes en un mundo moderno; además, manifiesta que de cierta forma esta inclusión permitirá desaparecer las fronteras entre personas videntes y no videntes, en razón de que el avance de la tecnología ha hecho que ésta, se vuelva un medio de sobrevivencia y competitividad.

1.5.1 Criterios de calidad

Según Rojas (2013), los criterios a tomarse en cuenta para el desarrollo de *software* orientado a personas con discapacidad visual son:

1.5.1.1 Accesibilidad

La asociación SIDAR (Comunidad Iberoamericana de Desarrolladores y Diseñadores Web) define a la accesibilidad como “el grado en que toda persona puede acceder a utilizar un objeto, visitar un lugar o acceder a un servicio independiente de sus capacidades técnicas o físicas”. Este indica los siguientes elementos a tener en cuenta:

- **Percepción:** es la información y componentes de la interfaz de usuarios que son presentados y la manera en que los usuarios la perciben.
- **Operatividad:** Tanto los componentes de la interfaz y la navegación, se ejecutarán correctamente.
- **Entendible:** la información que proporciona el sistema y la interfaz estarán descifrables.
- **Robustez:** La *Web Content Accessibility Guidelines* (2008) define que el contenido tiene que ser adecuadamente robusto de tal forma que pueda ser descifrado apropiadamente

De igual manera, EN (normas europeas) 301 549, indica que la accesibilidad es un requisito a cumplirse de los sistemas, aplicaciones y servicios para que sean fáciles, alcanzables, aptas y disponibles para todos los usuarios indistintamente de sus capacidades. Además, Aguado; Estrada (2017) acotan, que la accesibilidad es una capa en la que el fabricante o desarrollador consigue implementar con el fin de facilitar el acceso a personas con discapacidad a las funcionalidades del sistema. Por tanto, el software, aplicaciones o servicios

cumplirán los requisitos específicos a las necesidades del usuario final incluye a personas con problemas sensoriales, cognitivas u otras.

1.5.1.2 Usabilidad

La norma ISO/IEC 9126 define que: "La usabilidad se refiere a la cabida de un *software* de ser entendido, asimilado, usado y ser atrayente para el usuario, en condiciones específicas de uso".

Además, Fernández (2018), indican que la usabilidad no, se valora al estudiar un producto de manera aislada. Esta definición hace hincapié a concretar o eliminar una arquitectura y, finalmente decidir que es accesible. Por tanto, tener en cuenta estos criterios de calidad para el presente proyecto, permitieron definir elementos que ayuden a mejorar la calidad del producto final.

CAPÍTULO II. DISEÑO METODOLÓGICO

2.1 Tipo de investigación

2.1.1 Búsqueda bibliográfica

Se aplicó la búsqueda bibliográfica que, según Benavent, G. González , & Gonzáles de Dios (2011), permite verificar y obtener información. Por lo que, el realizar una investigación exhaustiva de diferentes fuentes en Internet permitió obtener elementos bibliográficos acerca del tema propuesto.

Por tanto, se realizó una búsqueda adecuada en documentos de Google Académico, Red CEDIA, Repositorio Digital UPCT, SciELO la misma que permitió analizarlos con el fin de recopilar datos necesarios para el presente proyecto, en donde, se ubicaron varios documentos referentes al tema, los mismos que, se examinaron de forma particular.

2.1.2 Análisis-síntesis

Se empleó el método analítico sintético que, como lo menciona Delgado García G (2010), es indispensable en el estudio de las cuestiones históricas analizar los

sucesos descomponiéndolos en todas sus partes para conocer sus posibles raíces.

2.1.3 Encuesta

La encuesta, como lo mencionan Anguita, Labrador, & Campos (2003), es una técnica que permite obtener información de manera rápida y eficaz. Además, Alvira (2011), acota que es un instrumento útil que sirve para recolectar información sobre un tema determinado.

Por tanto, para recolectar datos necesarios, se aplicó una encuesta a empresas desarrolladoras de *Software* de la Ciudad de Ambato, como desarrolladores aportan con ideas para la realización del modelo, a desarrolladores de *software* de la PUCE Ambato que desde su perspectiva indican que aspectos técnicos, se incluirán en el proyecto de tal forma que, se pueda recolectar información necesaria para el presente proyecto propuesto.

Población: 8 empresas desarrolladoras de *software* de la ciudad de Ambato, y 3 desarrolladores del departamento de TI PUCE Ambato.

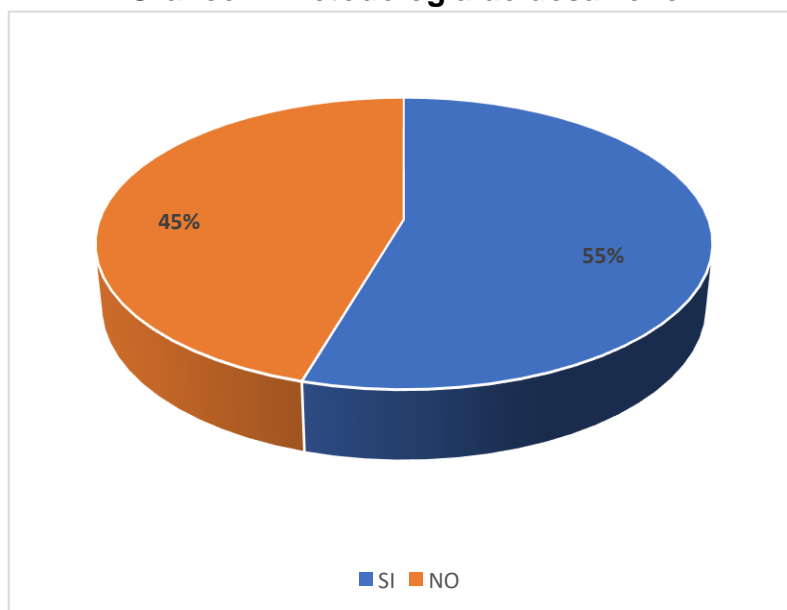
2.1.3.1 RESULTADO DE LA APLICACIÓN DE LA ENCUESTA A DESARROLLADORES DE LA CIUDAD DE AMBATO

Pregunta 1. ¿Aplica usted metodología de desarrollo?

SI () ¿Cuál?

NO ()

Gráfico 1. Metodología de desarrollo



Fuente: elaboración propia

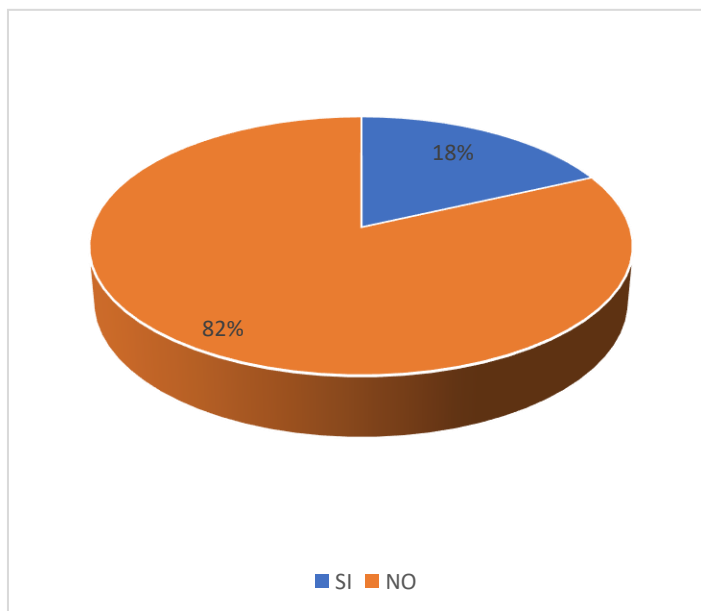
De acuerdo a la información obtenida, la mayoría de desarrolladores, afirman que usan una metodología de desarrollo, entre las más relevantes, se mencionan: *Rapid Application Development* (RAD), Espiral, XP y Scrum.

Por tanto, utilizar una metodología para el desarrollo de *software* es importante dado que permite parametrizar los procesos que fomenten la calidad del producto.

Pregunta 2: ¿Aplica algún modelo de aseguramiento de calidad en el desarrollo de *software*?

SI () ¿Qué modelo utiliza?

NO ()

Gráfico 2. Modelo de aseguramiento de calidad**Fuente:** elaboración propia

Mediante los datos recolectados, se denota que 2 de los 11 desarrolladores utilizan un modelo de aseguramiento de calidad, los cuales son: *Test-driven development* (TDD) y Supervisión XP (pruebas unitarias integrales).

La información obtenida muestra el desconocimiento de una herramienta para asegurar la calidad del *software*, lo cual permite la creación de un modelo de evaluación de calidad en el desarrollo que contribuya a evaluar la calidad del mismo.

Pregunta 3. ¿Qué factores o métricas toma en cuenta el modelo que utiliza?

Tabla 2. Factores y Métricas

Factor/ Métrica	Muy Importante	Importante	Moderadamente Importante	De poca importancia	Sin importancia
Usabilidad	8	2	-	-	-
Fiabilidad	8	1	1	-	-
Flexibilidad	9	1	-	-	-

Accesibilidad	7	3	-	-	-
Mantenibilidad	9	1	-	-	-

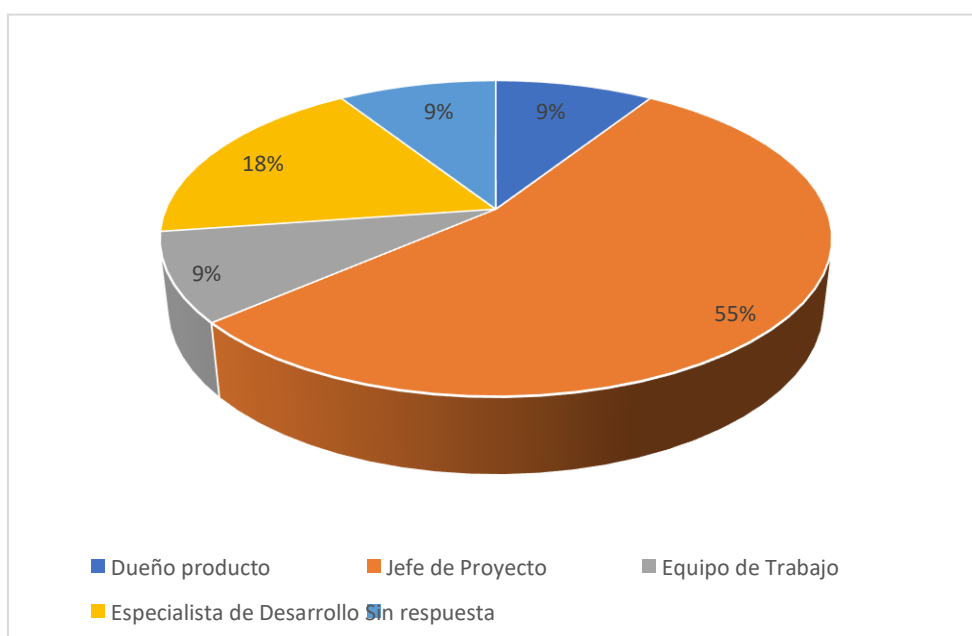
Otros:

Una cantidad elevada de desarrolladores, afirman que los factores y métricas expuestos en la columna 1, son considerados muy importantes, Además, de los factores descritos mencionan que, se consideran a tomar en cuenta los siguientes: adaptabilidad, seguridad, disponibilidad e integridad de datos.

Por esta razón, es de importancia establecer factores y métricas en el modelo de calidad a proponer, las mismas permitirán evaluar de forma acertada y acorde a la normativa propuesta como: la funcionalidad, el comportamiento y rendimiento del *software*.

Pregunta 4. Dentro de su empresa ¿Quién es la persona responsable de evaluar la calidad del *software*?

Gráfico 3. Responsable de evaluación



Fuente: elaboración propia

La persona encargada de evaluar la calidad del *software*, en la mayoría de los casos, es el jefe de proyecto. Se asume que no todos los departamentos de desarrollo de *software* realizan dicha evaluación.

Pregunta 5. ¿Con qué frecuencia evalúa la calidad del *software*?

R1. En las fechas programadas de entrega (2).

R2. Con cada actualización en el código del proyecto (commit).

R3. Cada 3 meses.

R4. Cada 30 días.

R5. En cada módulo.

R6. Al finalizar cada módulo.

R7. Diariamente.

R8. No responde.

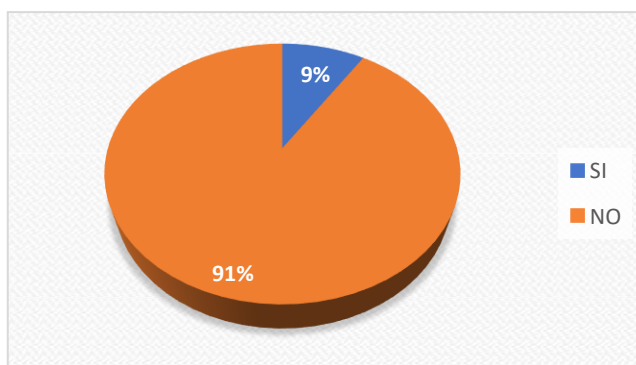
R9. No responde.

R10. No responde.

Se evidencia que, en los resultados obtenidos, se encuentra múltiple información, por lo cual, se asume que los desarrolladores evalúan su producto sin una normativa fija.

Pregunta 6. ¿Ha desarrollado usted *software* para personas con discapacidad visual?

Gráfico 4 .Software personas discapacidad visual



Fuente: elaboración propia

Conforme a la información obtenida, casi la totalidad de los encuestados no han desarrollado *software* para personas con discapacidad visual. Razón evidente para el estudio y propuesta de un modelo de evaluación de calidad para este tipo de aplicativos.

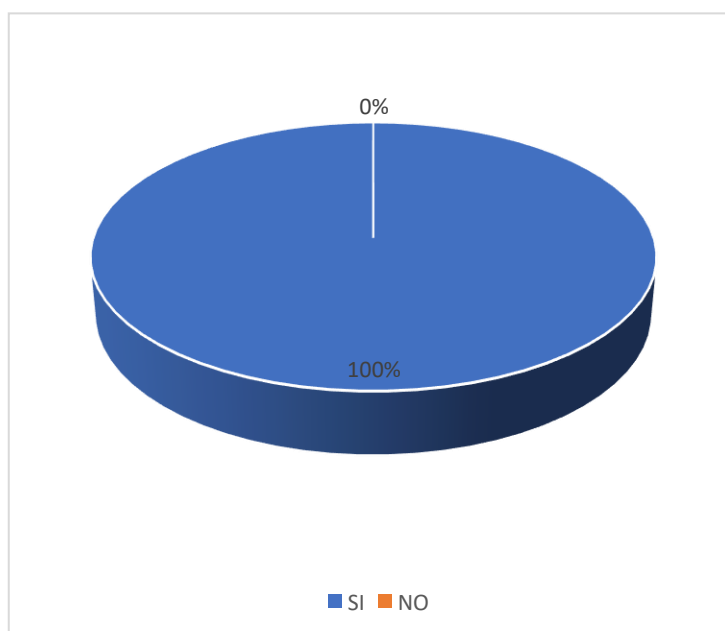
Pregunta 7. ¿Qué aspectos considera al evaluar la calidad de *software* para personas con discapacidad visual?

En los resultados obtenidos, se evidencia que cada persona encuestada proporciona sus propias consideraciones como: que el *software* sea de fácil manejo, sencillo y concreto; que cuente con otras ayudas sensoriales como audibles y táctiles; que pueda acceder a cada contenido, tanto texto como elementos multimedia, la forma de interacción con la interfaz sea intuitiva.

Los aspectos proporcionados son un valioso aporte para la creación de un modelo específicamente orientado a personas con discapacidad visual pues incorporan características propias y necesarias para este grupo de personas.

Pregunta 8. ¿Considera que el desarrollo de *software* para personas con discapacidad visual exige un mayor control de calidad?

Gráfico 5. Control de calidad



Fuente: elaboración propia

Todos los desarrolladores encuestados coinciden en que el desarrollo de *software* para personas con discapacidad visual exige un mayor control de calidad, por tanto, el proponer un modelo para evaluar la calidad en el desarrollo del *software* para personas no videntes ayudará a que, se desarrollen *software* de calidad, que brinde el mejor servicio a estos usuarios, el mismo que integrará normas y estándares propios, con el fin de establecer una guía para evaluar la accesibilidad y usabilidad de los mismos.

2.2 Metodología de Desarrollo

Para Gómez (2017), esta metodología, se caracteriza porque, se ejecuta de manera secuencial en razón de que a la finalización de una fase empieza la otra, toma en cuenta como datos de entrada, los resultados de la fase anterior.

La metodología comprende las siguientes fases:

Fase I Análisis- Definición de requerimientos:

En esta primer etapa Cervantes & Gómez (2012), acotan que en esta etapa, se determinan los servicios que proporciona el sistema conforme a las necesidades de los clientes y/o usuarios mientras que Goicochea (2016), menciona que el cliente y/o usuarios expone el entorno global y da a conocer las especificaciones para el sistema.

Fase II Diseño del sistema y del software:

En el proceso de diseño según Cervantes & Gómez (2012), se logra diferenciar cuales son los requerimientos del hardware y *software*. Además, se establecen los subsistemas y las dependencias entre ellos.

Fase III Implementación y validación de unidades:

Gonzáles (2019), menciona que, se codifica y se verifica la funcionalidad de los diferentes subsistemas individualmente.

Fase IV Integración y validación del sistema:

Cervantes & Gómez (2012), mencionan que al cumplirse lo requerido de la fase anterior, se integran los subsistemas para formar un sistema completo y comprobar que se cumpla con lo requerido por clientes y/o usuarios.

2.2.1 Adaptación de la Metodología

Se utilizó la Metodología en Cascada de manera empírica para el desarrollo del modelo. De manera que, se puedan usar sus fases para adaptarlas al Desarrollo de la Propuesta. De las 4 fases que, se mencionan anteriormente se utilizaron y, se adaptaron las siguientes:

Fase I: Análisis

Se tomaron en cuenta las diversas propuestas desde los factores de calidad propuestos por McCall, Boehm, CMMI, los estándares internacionales ISO/IEC TR 15504, ISO/IEC 29110, ISO 9126 e ISO 25000.

Fase II Diseño (modelo de calidad)

Con base en los resultados de la etapa anterior, se definieron los posibles aspectos técnicos y factores de calidad que abarcará el modelo.

Fase III Validación

La evaluación del producto, se realizó aplicando la técnica de V.A ladov, instrumento que mide indirectamente el índice de satisfacción de los usuarios. La técnica consiste en la aplicación de 5 preguntas: 3 abiertas y 2 cerradas que permitirán obtener información para la elaboración del cuadro lógico de ladov y la interpretación correspondiente

CAPÍTULO III. RESULTADOS

Fase I: Análisis

Para medir la evaluación de calidad en el desarrollo del *software* orientado a personas invidentes, se considera las diversas propuestas desde los factores de calidad a nivel proceso y a nivel producto propuestos por McCall, Boehm; los estándares internacionales ISO/IEC 9000, ISO 9126 e ISO 25000. De igual manera, se realizó una comparación acerca de los factores para la evaluación de *software* para personas no videntes. Además, se consideraron también los factores mencionados por los desarrolladores de *software* en las encuestas aplicadas los cuales, se analizaron por separado.

Para el análisis, se consideraron los factores de calidad de los modelos mencionados y se evaluaron si los modelos y estándares de calidad estudiados los incluyen. Se realizó una evaluación preliminar de cada uno de los factores de calidad y se eligen aquellos que cumplan con la mayor cantidad de los requerimientos. La valoración que, se estipula a cada uno de los factores es de 1 y se realiza la sumatoria total. Los resultados se muestran en la Tabla 1.

Análisis de los factores de calidad de los diferentes modelos estudiados.

Tabla 3. Factores de Calidad

	ISO 9000:2015	McCALL	Boehm	ISO/IEC 9126: 2001	ISO/IEC 25000:2015	Sumatoria
Funcionalidad	1	0	0	1	1	3
Fiabilidad	1	0	0	0	1	2
Eficiencia	1	0	1	1	1	4
Usabilidad	1	0	1	1	0	3
Mantenibilidad	1	1	1	1	1	5
Portabilidad	1	1	1	1	1	5
Facilidad uso	0	1	1	0	1	3
Integridad	0	1	0	0	0	1
Corrección	0	1	0	0	0	1
Facilidad de Prueba	0	1	0	0	0	1
Flexibilidad	0	1	1	0	0	2
Reusabilidad	0	1	0	0	0	1

Utilidad	0	0	1	0	0	1
Confiabilidad	0	0	1	1	0	2
Testeabilidad	0	0	1	0	0	1
Seguridad	0	0	1	0	0	1
Compatibilidad	0	0	1	0	0	1
Eficiencia en el desempeño	0	0	0	0	1	1
Adecuación funcional	0	0	0	0	1	1

Fuente: elaboración propia

Los factores que, se seleccionaron para integrar el modelo propuesto son aquellos cuya calificación es igual o superior a 3, como se observa en la siguiente tabla.

Tabla 4. Selección de factores de calidad

Factores	Puntaje
Mantenibilidad	5
Portabilidad	5
Eficiencia	4
Funcionalidad	3
Usabilidad	3
Facilidad de uso	3

Fuente: elaboración propia

Además, se consideraron los factores de calidad de las encuestas aplicadas que los desarrolladores de *software* consideran importantes a partir de las encuestas aplicadas. Los factores que, se tomaron en cuenta son los de calificaciones superiores a 6 en razón de que la mayoría de los encuestados coinciden en que los mismos son importantes al momento de evaluar la calidad del *software*.

Tabla 5. Análisis los factores de calidad de las encuestas aplicadas

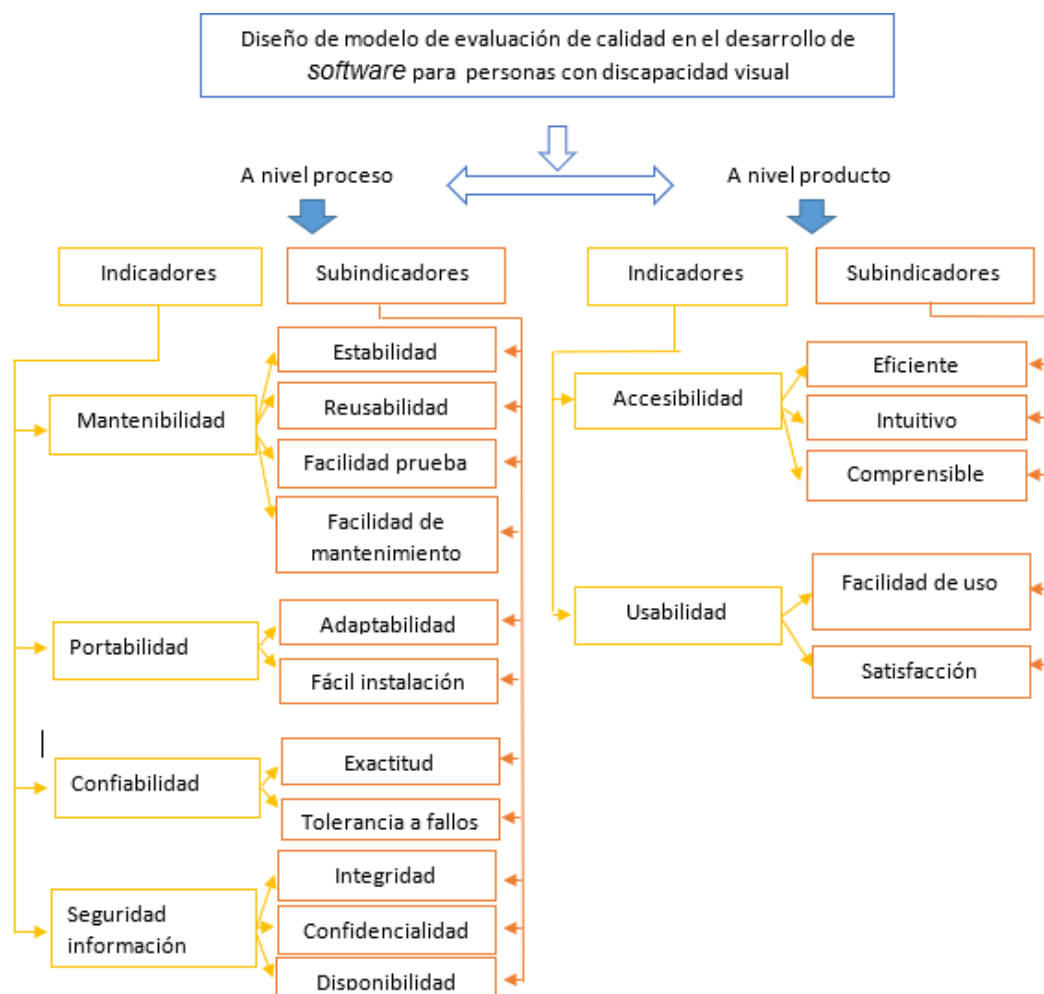
Factor/Métrica	Muy Importante
Mantenibilidad	9
Flexibilidad	9
Usabilidad	8
Fiabilidad	8
Accesibilidad	7

Fuente: elaboración propia

Fase II Diseño (modelo de calidad)

A continuación, se esquematiza el diseño propuesto.

Figura 8. Diseño de modelo de calidad



Fuente: elaboración propia

Se definieron los posibles indicadores y subindicadores de calidad que abarcará el modelo, así como las recomendaciones obtenidas a través de las encuestas.

Al puntualizar la calidad del *software*, se distinguen dos criterios: la calidad del producto de *software* y la calidad del proceso de desarrollo. Calero, Moraga, & Piattini (2010), indican que es esencial asegurar la calidad de los recursos, procesos y producto a nivel *software*, pero los dos últimos son de mayor trascendencia. A lo que, Mascheroni, Greiner, Dapozo, & Estayno, (2012), acotan que la calidad del producto de *software* define la calidad del proceso de desarrollo.

A nivel proceso, Rodríguez, Oviedo, & Piattini (2016), manifiestan que, se mide la calidad del proceso porque de este depende que el producto sea de calidad, los resultados son correctos para cada objetivo de uso si se realizan las evaluaciones durante su proceso. Los indicadores definidos:

Mantenibilidad: es la capacidad del *software* de ser entendido, de saber cómo funciona; la posibilidad de ser cambiado o modificado por cualquier programador, y ejecutarse sin ningún fallo. Los subindicadores que lo conforman son:

- **Estabilidad:** es el mínimo nivel de fallos existentes en el sistema. A lo que, Crouch (2019), acota que es la capacidad que el *software* tiene de encontrar errores y cambiarlo si es necesario. Además, los desarrolladores son los encargados de realizar evaluaciones durante todo el proceso y necesariamente realizar una prueba final.

En este subindicador, se evalúa que el nivel de fallos sea mínimo, teniendo como objetivo asegurar el funcionamiento del sistema sin la necesidad de eliminar todos sus errores.

- **Reusabilidad:** Para Juan Sebastián (2010), es uno de los factores importantes para los desarrolladores visto que continuamente siguen patrones semejantes porque agilitan el o los desarrollos de *software* mediante el uso de componentes.

- **Facilidad de prueba:** es la facilidad con la que, se logra probar un programa.
- **Facilidad de mantenimiento:** Para Ávila (2016), es una característica del *software* que facilita al desarrollador ubicar un error y este será solucionado, lo que garantiza que el *software* logra extenderse en su utilidad al mismo tiempo que, se reducen los fallos o imprevistos.

Portabilidad: Durango (2015), señala que es la capacidad que el *software* pueda ser trasladado y funcionar adecuadamente en otro entorno, sin necesitar mayores modificaciones que las previstas. Además, Gonzáles, André, & Hernández (2015), mencionan que es una de las características de calidad que más se toman en cuenta para la calidad. Se evaluarán las siguientes características:

- **Adaptabilidad:** este subindicador se evalúa la facilidad de adecuación del sistema a otros entornos y conservar el objetivo principal que es la optimización del sistema, es decir, que se pone atención a los cambios que pueda ocurrir en cada uno de los procesos.
- **Fácil de instalación:** es la capacidad del *software* de instalarse y utilizarse en otro entorno.

Confiabilidad

- **Exactitud:** característica de la aplicación en arrojar resultados apropiados, que al usuario le brinde confianza para su uso.
- **Tolerancia a fallos:** particularidad del sistema para continuar funcionando correctamente en caso de que el mismo presente fallos en la ejecución de alguno de sus componentes o módulos, y que no exista pérdida de información.

Seguridad de la información: ayuda a resguardar la información para evitar que personas que no estén debidamente autorizada accedan a esta. Para Solarte, Enríquez & Benavides (2015), la disponibilidad, confidencialidad e integridad son ejes importantes para amparar la información.

- **Integridad:** garantiza que la información no sea modificada por personas externas.
- **Confidencialidad:** garantizar que a la información no tenga acceso ninguna persona extraña.
- **Disponibilidad:** es el grado de disposición del sistema y la información en todo momento.

También, se incluye la calidad a nivel producto, la mayoría de las organizaciones desarrolladoras de *software* en la actualidad manifiestan interés por asegurar la calidad del mismo. Puesto que la calidad de este según lo mencionan Callejas, Alarcón, & Álvarez (2017), se programará desde el inicio, evaluarse en cada una de las etapas del proceso de desarrollo y valorar el cumplimiento de los factores de calidad, además, mencionan que, si, se deja de lado alguna de las verificaciones, se verá afectados tanto la calidad del proceso como del producto final. Por tanto, el tomar en cuenta estos dos criterios en el modelo propuesto permitieron establecer indicadores que contribuyan a alcanzar un producto de calidad.

Accesibilidad: es la simplicidad con la que las personas logran acceder a algún sitio, sistema u otros, particularmente las personas que tengan alguna discapacidad. Para, Mariño (2012), la accesibilidad es uno de los requisitos para la obtención de un producto de calidad. A su vez que, permite que todos los usuarios puedan acceder, entender e interactuar con los contenidos.

- **Eficiente:** se busca que el *software* cumpla apropiadamente su función, es decir, que tenga el menor retraso posible al momento que el usuario lo utilice. Este subindicador evalúa que el *software*, se desempeñe adecuadamente con los recursos que utiliza y las operaciones las realice en el menor tiempo posible.
- **Comprensible:** que la información contenida sea clara para el usuario.
- **Intuitivo:** el *software* tiene que entenderse y comprenderse de inmediato, sin necesidad de razonarlo, es decir, interactuar con el *software* de manera natural.

Usabilidad: Enríquez & Casas (2013), mencionan que es uno de los factores de calidad de gran importancia, pues mide el nivel de facilidad al momento de utilizar un producto, y el grado de satisfacción que tienen los usuarios después de utilizar el sistema, además, indican que esta comprende la forma y condiciones propias de los mismos.

- **Facilidad de uso:** indica la agilidad con la que el usuario consigue desarrollar las tareas después de conocer el funcionamiento del sistema.
- **Satisfacción:** señala el agrado de los usuarios de utilizar el sistema ya sea por su facilidad o simplicidad al momento de interactuar con el mismo.

Tabla 6. MODELO DE EVALUACIÓN DE CALIDAD EN EL DESARROLLO DE SOFTWARE PARA PERSONAS CON DISCAPACIDAD VISUAL

Nombre del software o aplicación: _____

Fecha: _____

Nombre del responsable a evaluar: _____

Mantenibilidad		Descripción	Preguntas			
Estabilidad	Que el nivel de fallos sea mínimo	¿Considera que el nivel de fallos es mínimo?	SI		NO	
Reusabilidad	Nivel en que pueda volver a utilizar el <i>software</i> o parte de él en otras aplicaciones.	¿Considera que, se reutilizará algún módulo o componentes del <i>software</i> en otras aplicaciones?	SI		NO	
Facilidad de prueba	Es la facilidad con la que, se logra probar un programa.	¿Es fácil realizar las pruebas de funcionamiento?	SI		NO	
Facilidad de mantenimiento	Facilita al desarrollador ubicar un error y este será solucionado.	¿El desarrollador ubica un error con facilidad y logra solucionarlo?	SI		NO	
Portabilidad						
Adaptabilidad	Facilidad de adecuación del sistema a otros entornos y lograr mantener la optimización del sistema.	¿El <i>software</i> logra adaptarse a otros entornos multiplataforma?	SI		NO	
Facilidad de instalación	El <i>software</i> , se instalará y utilizará en otros entornos de hardware y <i>software</i> .	¿Es el <i>software</i> fácil en su instalación?	SI		NO	
Confiabilidad						
Exactitud	Los resultados que arroja el sistema son apropiados.	¿El <i>software</i> , se ejecuta y funciona precisamente a lo prescrito?	SI		NO	
Tolerancia a fallos	Capacidad del <i>software</i> de seguir en función en caso de que falle algún componente o módulos, y que no exista pérdida de información.	¿El sistema seguirá funcionando correctamente en caso de fallo de alguno de sus componentes?	SI		NO	

A NIVEL PROCESO

Seguridad de Información	Descripción	Preguntas				
Integridad	Garantiza que la información se guarde apropiadamente y no sea modificada por personas externas.	¿Se encuentra la información que compone el sistema protegida ante modificaciones de personas externas?	SI		NO	
Confidencialidad	Garantiza que la información sea accesible únicamente a personas autorizadas.	¿Es la información del sistema accedida únicamente por personas autorizadas?	SI		NO	
Disponibilidad	Que la información esté accesible en cualquier momento.	¿Los usuarios logran acceder sin ningún inconveniente a la información?	SI		NO	

Observaciones: _____

Firma:

A NIVEL PRODUCTO

Nombre del software o aplicación: _____

Fecha: _____

Nombre del responsable a evaluar: _____

Accesibilidad		Descripción	Preguntas			
Eficiente	Que el <i>software</i> se desempeña adecuadamente con los recursos utilizados.	¿El <i>software</i> realiza las operaciones en el menor tiempo posible?	SI		NO	
Comprensible	Facilidad con la que el usuario interactúa con el programa.	¿El usuario entiende con facilidad las instrucciones para el uso del <i>software</i> ?	SI		NO	
Intuitivo	El <i>software</i> tiene que entenderse y comprenderse de inmediato.	¿El <i>software</i> es fácil de comprender?	SI		NO	
Usabilidad		Descripción	Preguntas			
Satisfacción	Grado con el que el <i>software</i> cumple con los requerimientos.	¿El <i>software</i> cumple con los requerimientos establecidos?	SI		NO	
Facilidad de uso	Agilidad con la que el usuario desarrolla las tareas después de conocer el funcionamiento del sistema.	¿El usuario consigue interactuar con facilidad con el sistema?	SI		NO	

Observaciones _____

Firma

Fase III Validación

La evaluación del producto, se realizó aplicando la técnica de V.A. Iadov, que según Kuzmina (1970); Villacís (2017), es un instrumento que mide de forma indirecta el grado de satisfacción de los usuarios. La técnica consiste en la aplicación de 5 preguntas: 3 abiertas y 2 cerradas que permiten obtener información para la elaboración del cuadro lógico de V.A Iadov y su análisis correspondiente. De igual manera, se añadieron las respectivas respuestas o sugerencias que los encuestados consideraron oportunas (ver nota final del anexo 2).

Así mismo, con los datos obtenidos, se calculó el índice de satisfacción grupal (ISG), el cual está interpretado en una escala numérica que va desde +1 (máxima satisfacción), 0,5 (más satisfecho que insatisfecho), 0 (no definido), -0,5 (más insatisfecho que satisfecho), hasta -1 (máxima insatisfacción).

El ISG, se calcula con la siguiente fórmula²:

$$\text{ISG} = \frac{A(+1)+B(+0.5)+C(0)+D(-0.5)+E(-1)}{N}$$

Para determinar los datos, se elaboran dos preguntas con opciones de respuesta: SI, NO y NO SÉ y una pregunta que pueda incluir las opciones de:

- Me gusta mucho
- Me gusta más de lo que me disgusta
- Me da lo mismo o Indiferente
- Me disgusta más de lo que me gusta
- No me gusta
- No sé qué decir

² En la fórmula del ISG: ISG es índice de satisfacción grupal; A frecuencia de 1; B frecuencia de 0.5; C frecuencia de 0; D frecuencia de -0.5; E frecuencia de -1; y, N total de encuestados

Los resultados, se consiguen al relacionar las tres preguntas de la encuesta de satisfacción, en el cuadro lógico de ladov (ver Tabla 5), el cual fue establecido por Kuzmina (1970).

Tabla 7. Cuadro Lógico de ladov ISI (índice satisfacción individual)

Pregunta	Pregunta								
	NO			NO SÉ			SI		
	Pregunta								
	Si	No sé	No	Si	No sé	No	Si	No sé	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

Fuente: Kuzmina (1970)

Para la interpretación de los resultados del ISI (Índice de satisfacción individual) e ISG (Índice de satisfacción grupal), se muestra a continuación la escala de satisfacción de la técnica.

1. Clara satisfacción
2. Más satisfecho que insatisfecho
3. No definida
4. Más insatisfecho que satisfecho
5. Clara insatisfacción
6. Contradictoria

Para alcanzar los resultados, se relacionan los datos que, se obtuvieron en las encuestas (ver Anexo 2):

- Sí la pregunta 1 responde “NO”, se marca la casilla NO que esté debajo de la de la PREGUNTA 1.
- Sí la pregunta 2 responde “SI”, se marca la casilla SI que este debajo de la opción NO de la pregunta anterior.
- Finalmente, sí la pregunta 3 responde “Me gusta mucho”, se busca en la tabla la parte en la cual coinciden las tres opciones de respuesta.

Tabla 8. Cuadro Lógico de ladov aplicado a desarrolladores

Desarrollador 1

Pregunta 3: ¿Usted aplicaría la matriz del Modelo de evaluación de calidad en el desarrollo de <i>software</i> para personas con discapacidad visual?	Pregunta 1: ¿Conoce los beneficios de utilizar un modelo de evaluación de calidad en el desarrollo de <i>software</i>?								
	NO				NO SÉ			SI	
	Pregunta 2: ¿Es importante utilizar un modelo para evaluar la calidad del <i>software</i> orientado a personas con discapacidad visual?								
	SI	NO	N	SI	NO	N	S	NO	NO
	SÉ	O		SÉ	O	I	SÉ		
Me gusta mucho	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

Clara Satisfacción (1)

Desarrollador 2

Tabla 9. Cuadro Lógico de ladov aplicado a desarrolladores

Pregunta 3: ¿Usted aplicaría la matriz del Modelo de evaluación de calidad en el desarrollo de <i>software</i> para personas con discapacidad visual?	Pregunta 1: ¿Conoce los beneficios de utilizar un modelo de evaluación de calidad en el desarrollo de <i>software</i>?								
	NO			NO SÉ			SI		
	Pregunta 2: ¿Es importante utilizar un modelo para evaluar la calidad del <i>software</i> orientado a personas con discapacidad visual?								
	SI	NO SÉ	NO	SI	NO SÉ	NO	SI	NO SÉ	NO
Me gusta mucho	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

Más satisfecho que insatisfecho (2)

De este modo, se nota que el índice de satisfacción del instrumento de los datos del desarrollador 1 es 1, el mismo que indica que existe “Clara Satisfacción”. Mientras que del desarrollador 2, es 2 que indica que es “Más satisfecho que insatisfecho”.

Los resultados al aplicar la fórmula con los valores predeterminados se muestran a continuación.

Figura 9. Índice de satisfacción individual y grupal

ÍNDICE DE SATISFACCIÓN INDIVIDUAL (ISI)						
	1	2	3	4	5	6
Nivel de satisfacción		Clara satisfacción	Más satisfecho que insatisfecho	No definido o contradictorio	Más insatisfecho que satisfecho	Clara Insatisfacción
Frecuencia	1		0.5	0	-0.5	-1
Escala	A		B	C	D	E
Resultado ISI	7		2	1	0	0

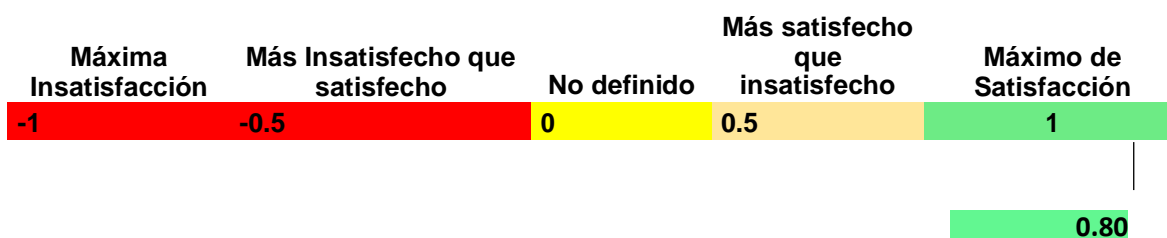
ÍNDICE DE SATISFACCIÓN GRUPAL (ISG)

$$ISG = \frac{A(+1) + B(+0.5) + C(0) + D(-0.5) + E(-1)}{N}$$

$$ISG = \frac{7(+1) + 2(+0.5) + 1(0) + 0(-0.5) + 0(-1)}{10}$$

$$ISG = 0.80$$

Figura 10. Escala de valores del ISG



Fuente: elaboración propia

Al obtener un ISG de 0.80, se aprecia un acercamiento a la máxima satisfacción de los desarrolladores en relación a la aplicación del Modelo de evaluación de calidad en el desarrollo de *software* para personas con discapacidad visual propuesto.

³ Donde A y +1: máximo de satisfacción, B y 0.5: más satisfecho que insatisfecho, C y 0: no definido, D y -0.5: más insatisfecho que satisfecho, E y -1: máxima insatisfacción.

Software utilizado para aplicar el modelo de evaluación:

A nivel proceso

- NVDA (Non Visual Desktop Access), es un lector de pantalla, que posibilita a personas con discapacidad visual acceder a un computador. Lo que este software hace es leer lo que, se presenta en la pantalla con comandos de teclado. Está disponible en más de 40 idiomas. Fue validado por el Ingeniero Eduardo Remache, del Departamento de TI de la PUCESA, quién, se basó en los documentos de codificación, procesos y desarrollo de la aplicación para realizar la validación correspondiente.
- Braille Teacher, es una aplicación basada en el alfabeto braille español, lo transforma en audio, además, consigue realizar acciones como compartir la traducción por redes sociales o correo. El Ing. Francisco Javier García Camacho, desarrollador de Braille, validó esta aplicación con la matriz diseñada.
- El sistema Academics y la plataforma educativa Moodle de la PUCESA, que es una plataforma que brinda espacios de aprendizaje adaptados para los estudiantes, profesores y administrativos. Fue evaluado por el Ingeniero José Enríquez, desarrollador de *software* y DBA del Departamento de TI.
- El recurso didáctico multimedia, desarrollado por el Ing. Leonardo Ayavaca, cuenta con características para estudiantes con discapacidad visual y auditiva, es aplicado para refuerzo en Ciencias Naturales de estudiantes de cuarto, quinto y sexto año de educación especial básica media. Lo que realiza este recurso es desplegar en la pantalla el contenido de las unidades por medio de sonido, gráficos, animaciones y texto. Depende el caso de discapacidad que se elija. Este recurso, también, fue diseñado con la matriz del modelo propuesto.

A nivel producto

De igual forma, se tuvo la colaboración de algunos usuarios para realizar la evaluación a nivel producto de cada software y aplicaciones propuestas.

Los estudiantes de Psicología Grace Sánchez, Estefanía Llerena y Orlando Sailema, que padecen de ceguera del 75% y 100% participaron en la validación de la plataforma Moodle y Academics de la PUCESA, y, como usuarios de NVDA.

La aplicación Braille Teacher, fue validada por: Alejandro Castro(Veracruz, México) y Nancy Kenwa (Palmira, Colombia), quienes son profesores que padecen el 100% de ceguera y enseñan a utilizar aplicaciones a las personas con discapacidad visual, en la Universidad de México y en una biblioteca llamada Consentidos (Colombia) la misma que cuenta con una sala que es adaptada específicamente para personas no videntes.

Finalmente, la validación del recurso didáctico multimedia, la realizó el Ing. Leonardo Ayavaca, desarrollador del recurso.

En las siguientes tablas, se presenta detalladamente la validación del instrumento por cada desarrollador y usuario respectivamente.

Indicadores	Subindicadores	Preguntas	NVDA (Non Visual Desktop Access)	Recurso didáctico multimedia	ACADEMICS PUCESA	MOODLE PUCESA	Braille Teacher
Mantenibilidad	Estabilidad	¿Considera que el nivel de fallos es mínimo?	SI	SI	NO	NO	SI
	Reusabilidad	¿Considera que, se reutilizará algún módulo o componentes del software en otras aplicaciones?	SI	SI	SI	SI	SI
	Facilidad de prueba	¿Es fácil realizar las pruebas de funcionamiento?	SI	SI	NO	NO	SI
	Facilidad de mantenimiento	¿El desarrollador logra ubicar un error con facilidad y solucionarlo?	SI	SI	SI	SI	SI
Portabilidad	Adaptabilidad	¿El software consigue adaptarse a otros entornos multiplataforma?	NO	SI	SI	SI	NO
	Facilidad de instalación	¿Es el software fácil en su instalación?	SI	SI	SI	SI	SI
Confiabilidad	Exactitud	¿El software se ejecuta y funciona precisamente a lo prescrito?	SI	SI	SI	SI	SI
	Tolerancia a fallos	¿El sistema sigue funcionando correctamente en caso de fallo de alguno de sus componentes?	NO	SI	SI	SI	SI
Seguridad de Información	Integridad	¿Se encuentra la información que compone el sistema protegida ante modificaciones de personas externas?	SI	NO	SI	SI	SI
	Confidencialidad	¿Es la información del sistema accedida únicamente por personas autorizadas?	SI	NO	SI	SI	SI
	Disponibilidad	¿Los usuarios acceden sin ningún inconveniente a la información?	SI	SI	NO	NO	SI

Tabla 1. Cuadro comparativo validación *software* no videntes a nivel proceso

Fuente: elaboración propia

Para la validación de *software* y aplicaciones a nivel proceso, se observa en la tabla anterior, que la mayoría de las respuestas son positivas, por ende, para cada uno de los desarrolladores que evaluaron cada una de las aplicaciones y *software* propuestos, comprobaron que existe calidad a nivel del proceso de los mismos.

Para medir la calidad a nivel producto de las aplicaciones y *software* sugeridos, se aplicó el instrumento estadístico KR20, Kuder-Richarson (1937) como lo señalan Ríos, Olivera, Ballena-López, & Peralta (2013); Ruíz(2019), es un indicador de la fidelidad de las respuestas, dando así un grado de confiabilidad, exactitud y seguridad en la consistencia de los resultados.

El KR20, se calcula utilizando la siguiente fórmula⁴:

$$KR20 = \frac{n}{n - 1} * \frac{Vt - \sum pq}{Vt}^5$$

El resultado final, se lo ubica dentro de los rangos que indica la siguiente tabla y se define el grado de confiabilidad.

Tabla 2. Criterios de decisión e interpretación KR20

Rango	Confiabilidad
0.8 – 1	Muy Alta o elevado
0.6 – 0.8	Alta
0.4 – 0.6	Moderada
0.2 – 0.4	Baja
0.01 – 0.20	Muy baja

Fuente: elaboración propia

⁵ En la fórmula de KR20: Kuder Richardson; n es el número de ítems; Vt es varianza total; p es la proporción de sujetos que pasan un ítems sobre el total; q es igual a 1-p.

Tabla 3. Cuadro comparativo validación Academics y Moodle PUCESA

Indicadores	Subindicadores	Preguntas	ACADEMICS PUCESA Estudiante 1	ACADEMICS PUCESA Estudiante 2	ACADEMICS PUCESA Estudiante 3	MOODLE PUCESA Estudiante 1	MOODLE PUCESA Estudiante 2	MOODLE PUCESA Estudiante 3
Accesibilidad	Eficiente	¿El software realiza las operaciones en el menor tiempo posible?	NO	SI	SI	SI	SI	SI
	Comprensible	¿El usuario entiende con facilidad las instrucciones para el uso del software?	SI	NO	NO	SI	SI	SI
	Intuitivo	¿El software es fácil de comprender?	SI	SI	SI	SI	SI	SI
Usabilidad	Satisfacción	¿El software cumple con los requerimientos establecidos?	SI	SI	NO	NO	SI	SI
	Facilidad de Uso	¿El usuario logra interactuar con facilidad con el sistema?	NO	NO	NO	NO	SI	SI

Fuente: elaboración propia

Tabla 4. KR20 Moodle

SUJETOS	PREGUNTAS					TOTAL
	1	2	3	4	5	
1	1	1	1	0	0	3
2	1	1	1	1	1	5
3	1	1	1	1	1	5
P	1	1	1	0.67	0.67	
q(1-p)	0	0	0	0.33	0.33	Vt 1.33
Pq	0	0	0	0.22	0.22	$\sum pq$ 0.44

N	5
n-1	4
n/n-1	1
Vt- $\sum pq$	0.89
Vt- $\sum pq/Vt$	0.67
KR20	0.7
Alta	(0.61-0.80)

Fuente: elaboración propia

Al evaluar la calidad de la plataforma educativa Moodle PUCESA, el resultado del instrumento de confiabilidad KR20, corroboró que tiene una calidad Alta para los usuarios.

Tabla 5. KR20 Academics

SUJETOS	PREGUNTAS					TOTAL
	1	2	3	4	5	
1	0	1	1	1	0	3
2	1	0	1	1	0	3
3	1	0	1	0	0	2
P	0.67	0.33	1	0.67	0	Vt 0.3
q(1-p)	0.33	0.67	0.00	0.33	1.00	
pq	0.22	0.22	0	0.22	0	$\sum pq$ 0.67

N	5
n-1	4
n/n-1	1
$Vt-\sum pq$	-0.33
$Vt-\sum pq/Vt$	-1.00
KR20	-1

Fuente: elaboración propia

La aplicación del instrumento de confiabilidad KR20, para evaluar la calidad del sistema Academics PUCESA, determinó que no es un software de calidad para los usuarios.

Tabla 6. Cuadro comparativo validación NVDA

Indicadores	Subindicadores	Preguntas	NVDA Usuario 1	NVDA Usuario 2
Accesibilidad	Eficiente	¿El software realiza las operaciones en el menor tiempo posible?	SI	SI
	Comprensible	¿El usuario entiende con facilidad las instrucciones para el uso del software?	NO	SI
	Intuitivo	¿El software es fácil de comprender?	NO	SI
Usabilidad	Satisfacción	¿El software cumple con los requerimientos establecidos?	NO	SI
	Facilidad de Uso	¿El usuario consigue interactuar con facilidad con el sistema?	NO	NO

Fuente: elaboración propia

Tabla 7. KR20 NVDA

SUJETOS	PREGUNTAS					TOTAL
	1	2	3	4	5	
1	1	1	1	1	0	4
2	1	0	0	0	0	1
p	1	0.5	0.5	0.5	0	
q(1-p)	0	0.5	0.5	0.5	1	Vt 4.5
pq	0	0.25	0.25	0.25	0	$\sum pq$ 0.75

N	5
n-1	4
n/n-1	1.25
Vt- $\sum pq$	3.75
Vt- $\sum pq/Vt$	0.833
R	1

Muy Alta

Fuente: elaboración propia

Al aplicar el instrumento de confiabilidad KR20, se evidenció una puntuación de 1, por lo que, se demostró que el software posee una calidad muy alta.

Tabla 8. Cuadro comparativo validación Braille Teacher

Indicadores	Subindicadores	Preguntas	COMMCISO Braille Teacher Usuario 1	COMMCISO Braille Teacher Usuario 2
Accesibilidad	Eficiente	¿El software realiza las operaciones en el menor tiempo posible?	SI	SI
	Comprensible	¿El usuario entiende con facilidad las instrucciones para el uso del software?	SI	SI
	Intuitivo	¿El software es fácil de comprender?	SI	SI
Usabilidad	Satisfacción	¿El software cumple con los requerimientos establecidos?	SI	SI
	Facilidad de Uso	¿El usuario logra interactuar con facilidad con el sistema?	SI	SI

Fuente: elaboración propia

En la validación de Braille Teacher, no fue necesaria la aplicación del instrumento de confiabilidad KR20, puesto que las respuestas son positivas en su totalidad, lo que indica que para los usuarios este es un software de calidad.

CONCLUSIONES

- La sustentación de manera teórica, de los aspectos relacionados a la evaluación de calidad, permitió concluir que la accesibilidad y usabilidad, son factores relevantes y fundamentales, en el momento que, se emiten criterios y; se definen indicadores de calidad al momento de evaluar la calidad del software de evaluación en del desarrollo del desarrollo del *software*.
- El contraste de modelos de evaluación de calidad utilizados en el desarrollo de software para identificar sus fortalezas y debilidades, permitió el análisis de diversos modelos de calidad en el desarrollo de *software*, por tanto, se concluye que existen algunos elementos en común que sirven para la elaboración del modelo propuesto.
- La formulación del modelo de evaluación de calidad en el desarrollo del software permitió seleccionar cuidadosamente los indicadores y subindicadores que ayuden a medir la calidad tanto del proceso como del *software* para obtener un producto que cumpla con sus expectativas.
- La validación del modelo propuesto, se realizó con desarrolladores de *software* con amplia experiencia en el campo, los mismos que calificaron el modelo como satisfactorio para su aplicación, este provee los parámetros adecuados que sirven para medir la calidad en el desarrollo de aplicativos para personas invidentes.

RECOMENDACIONES

- Aplicar el modelo propuesto en futuros desarrollos de aplicativos para personas con discapacidad visual, para medir la calidad durante las etapas de su desarrollo e implementación.
- Es recomendable que este modelo, se encuentre en constante revisión, dado que los procesos de desarrollo de aplicación siempre están en proceso de mejora continua.
- La aplicación de este modelo permitirá conocer los elementos vulnerables evaluados, los cuales serán considerados para realizar mejoras en su proceso y de esta forma obtener mejores resultados.
- El resultado que se obtiene al aplicar este modelo permitirá conocer la efectividad del software, por lo que es recomendable que este se lo aplique desde su desarrollo, permite así conocer los puntos vulnerables y tomar acciones correctivas a tiempo.

BIBLIOGRAFÍA

- Abuchar, A. (2015). DEL DEBATE ENTRE LO CUALITATIVO Y LO CUANTITATIVO. *TIA Tecnología, investigación y academia*, 1-5.
- Aguado, J., & Estrada, F. (2017). *Guía de Accesibilidad de Aplicaciones Móviles*.
- Alarcón, H., Hurtado, A., Pardo, C., Collazos, C., & Pino, F. (2007). *Repositorio Institucional UN*. Obtenido de <http://www.bdigital.unal.edu.co/15329/>
- Alvira, F. (2011). *La encuesta: una perspectiva general metodológica*. Madrid.
- Anguita, J., Labrador, J., & Campos, J. (2003). La encuesta como técnica de investigación. unidaddocentemfyclaspalmas.org.es.
- Areba, J. (2001). *Metodología del análisis estructurado de sistemas*. Madrid.
- Arenas, J. (2015). Obtenido de <https://es.slideshare.net/clauddiaa/factores-de-calidad-segn-mc-call>
- Arguedas, O. (2009). La búsqueda bibliográfica. *SCielo*.
- Ávila, J. (2016). *El ciclo de vida del desarrollo de aplicaciones*.
- Benavent, A., G. González, A., & Gonzáles de Dios, J. (Abril de 2011). *Evidence based medicine Sources of information in Pediatrics: Where can the*. Obtenido de file:///C:/Users/USUARIO/Downloads/FORMACIONEINFORMACION_IBsquedas.pdf
- Calero, C., Moraga, M. Á., & Piattini, M. (2010). *Calidad del producto y proceso software*. España.
- Calero, C., Moraga, M., & Piattini, M. (2010). *Calidad del producto y proceso software*.
- Callejas, M., Alarcón, A., & Álvarez, A. M. (2017). Modelos de calidad del software, un estado del arte*. *Entramado*, 236-250.
- Campo Arias, A., & Oviedo, H. (2018). Propiedades Psicométricas de una Escala: la Consistencia Interna. *SCielo*.
- Cervantes, J., & Gómez, M. (2012). Taxonomía de los modelos y metodologías de desarrollo de software más utilizados. *UDUAL*.
- Chacon, A., Rodas, J., & Vinueza, M. (Abril de 2015). *Repositorio UNEMI*. Obtenido de <http://repositorio.unemi.edu.ec/handle/123456789/3074>
- Crouch, S. (12 de abril de 2019). *Software Sustainability Institute*. Obtenido de <https://software.ac.uk/resources/guides/developing-maintainable-software>
- Castro, C. (2012). El futuro de las tecnologías digitales aplicadas al aprendizaje de personas con necesidades educativas especiales. *RED. Revista de Educación a Distancia*.

- Delgado García, G. (2010). Conceptos y metodología de la investigación histórica. *Revista cubana de salud pública*.
- Delgado García, G. (2010). *Redalyc*. Obtenido de <https://www.redalyc.org/pdf/214/21416134003.pdf>
- Enríquez, J., & Casas, S. (2013). Usabilidad en aplicaciones móviles. *Revista de informes científicos-técnicos UNPA*.
- Estayno, M., Dapozo, G., Cuenca Pletsch, Liliana Raquel, & Greiner, C. (2009). *Repositorio Institucional de la UNLP*. Obtenido de <http://sedici.unlp.edu.ar/handle/10915/19762>
- Esterkin, V., & Pons, C. (2017). Evaluación de calidad en el desarrollo de software dirigido por modelos. *Scielo*, 449-463.
- Estévez, Y. (12 de junio de 2014). Obtenido de www.unihorizonte.edu.co/revistas/index.php/TECKNE/article/view/129
- Fernández, P. (2018). *Usabilidad Web Teoría y uso*. Ra-Ma.
- Gálaz Álvarez, C. (2014). *Repositorio universidad de Chile*. Obtenido de <http://repositorio.uchile.cl/handle/2250/116371>
- Goicochea, A. (23 de 03 de 2016). <http://tesis.pucp.edu.pe/repositorio/handle/123456789/6523>. Obtenido de http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/6523/GOICOCHEA_ANTO_NY_EVALUACI%C3%93N_EXPERIMENTAL_SOFTWARE.pdf?sequence=1&isAllowed=y
- Gómez, A. (2017). *Diseño funcional y de la interactividad de productos multimedia*. ARGN0110. Málaga.
- Gómez, J. (2015). *Guía para la aplicación de UNE-EN ISO 9001:2015*.
- González, A., André, M., & Hernández, A. (2015). Análisis comparativo de modelos y estándares para evaluar la calidad del producto de software. *RCI*.
- González, A. (2019). Obtenido de <http://profesores.elo.utfsm.cl/~agv/elo329/1s10/lectures/SoftwareEngineeringParte2.pdf>
- Jorge, E., Carlos, C., Jorge, V., & María, M. (2008). *researchgate.net*. Obtenido de https://www.researchgate.net/profile/Maria_Macias_Mendoza/publication/28792101_Analisis_De_Los_Modelos_De_Calidad_De_Software_Existentes_Y_Su_Apoyo_Al_Cumplimiento_De_Requerimientos_En_Empresas_No_Dedicadas_Al_Desarrollo_De_Software/links/540db7f00cf2f2b2
- Jump. (4 de mayo de 2014). Obtenido de <https://software.ac.uk/resources/guides/developing-maintainable-software>

Laporte, L. (2017). *Software Quality Assurance*.

Madrid, J. (2008). Obtenido de http://www.tesis.uchile.cl/tesis/uchile/2008/madrid_jm/sources/madrid_jm.pdf

Madruñero, E. (abril de 2018). *repositorio utn*. Obtenido de <http://repositorio.utn.edu.ec/bitstream/123456789/8165/1/PG%20638%20TESIS.pdf>

Mariño, S., Godoy, M., Alfonzo, P., Acevedo, J., Gómez, L., & Fernández, A. (2012). Accesibilidad en la definición de requerimientos no funcionales. Revisión de herramientas. *Redalyc*.

Martínez, C., Siva, R., & Naranjo, B. (2016). *Repositorio Digital de la Universidad de Cuenca*. Obtenido de <http://dspace.ucuenca.edu.ec/handle/123456789/26349>

Moreno, M. (Mayo de 2017). *repositorio pucesa*. Obtenido de <http://repositorio.pucesa.edu.ec/bitstream/123456789/1929/1/76434.pdf>

Nigel, B. (1999). Meeting User Needs for Quality. *Journal of systems and software*, 89-96.

Nuviala, N., Tamayo Fajardo, Fernández Martínez, A., & Pérez Turpin, J. (2011). CALIDAD DEL SERVICIO DEPORTIVO EN LA EDAD. *Redalyc*.

Pérez, C., & Luque, S. (2018). Repositorio Universitat Jaume I. *adComunica. Revista Científica de Estrategias, Tendencias e Innovación en Comunicación*. Obtenido de <http://hdl.handle.net/10234/173990>

Puello, O. (2009). Modelo de Verificación y Validación Basado en CMMI. *Investigación E Innovación En Ingenierías*.

QAEC. (2018). *Asociación Española para la Calidad*. Obtenido de <https://www.aec.es/>

Ríos, A., Olivera, W., Ballena-López, J., & Peralta, J. (2013). Validación de un instrumento para medir el nivel de conocimiento sobre depresión mayor en médicos de atención primaria en Chiclayo, Perú. *SCIELO*.

Rodríguez, M., Oviedo, J., & Piattini, M. (2016). Obtenido de <http://www.aqclab.es/images/AQCLab/Noticias/SQP/software-quality-management-evaluation-of-software-product-functional-suitability-a-case-study.pdf>

Rojas, Y., & Rojas Galindo, Y. (2013). *repositorio umsa*. Obtenido de <http://repositorio.umsa.bo/xmlui/handle/123456789/7771>

Ruíz, B. (2019). *red docente*. Obtenido de <http://200.11.208.195/blogRedDocente/alexisduran/wp-content/uploads/2015/11/CONFIABILIDAD.pdf>

Scalone, F. (2006). *Estudio comparativo de los modelos y estándares de calidad del software*. Obtenido de <http://laboratorios.fi.uba.ar/lsi/scalone-tesis-maestria-ingenieria-en-calidad.pdf>

- Sebastián, J. (30 de octubre de 2010). *ComuSOFT.com*. Obtenido de <http://www.comusoft.com/factores-de-calidad-del-software-seguridad-legibilidad-y-reusabilidad>
- Solarte, F., Enríquez, E., & Benavidez, M. (2015). Metodología de análisis y evaluación de riesgos aplicados a la seguridad informática y de información bajo la norma. *ESPOL*.
- Sommerville, I. (2005). *Ingeniería del software*.
- Villacís, L. (2017). Propuesta de una guía pedagógica teatral en el desarrollo de la habilidad lingüística. *PAG Revista Iberoamericana de Producción Académica y Gestión Educativa*. Obtenido de <http://www.pag.org.mx/index.php/PAG/article/view/689>
- Web Content Accessibility Guidelines*. (2008). Obtenido de [http://www.w3.org/TR/WCAG20/Web Content Accessibility Guidelines \(WCAG\) 2.0](http://www.w3.org/TR/WCAG20/WebContentAccessibilityGuidelines(WCAG)2.0)
- Yepes, J., César, P., & Gómez, O. (Diciembre de 2015). Obtenido de <http://rte.espol.edu.ec/index.php/tecnologica/article/view/454/319>

ANEXOS

Anexo 1. Encuesta para desarrolladores



ENCUESTA PARA DESARROLLADORES DE LA EMPRESA ALQUIMIASOFT

Indicaciones: Señor(es) desarrolladores de *software* de la ciudad de Ambato, con el fin de conocer el grado de satisfacción de la aplicación del MODELO DE EVALUACIÓN DE CALIDAD EN EL DESARROLLO DE *SOFTWARE* PARA PERSONAS CON DISCAPACIDAD VISUAL, solicito su ayuda en la siguiente encuesta. Gracias por su colaboración.

Indicaciones: Señor(es) desarrolladores de *software* de la ciudad de Ambato, con el fin de conocer la necesidad de utiliza un modelo de evaluación de calidad en el desarrollo de *software*, solicito su ayuda en la siguiente encuesta la misma que servirá para recopilar información necesaria. Gracias por su colaboración.

1. ¿Aplica usted metodología de desarrollo?

SI () ¿Cuál? _____

NO ()

2. ¿Aplica algún modelo de aseguramiento de calidad en el desarrollo de *software*?

SI () ¿Qué modelo utiliza _____

NO ()

3. ¿Qué factores o métricas toma en cuenta el modelo que utiliza?

Factor/Métrica	Muy importante	Importante	Moderadamente importante	De poca importancia	Sin importancia
Usabilidad					

Fiabilidad					
Flexibilidad					
Accesibilidad					
Mantenibilidad					
Otros					

Otros ()

¿Cuáles?

4. Dentro de su empresa ¿Quién es la persona responsable de evaluar la calidad del *software*?

5. ¿Con qué frecuencia evalúa la calidad del *software*?

6. ¿Ha desarrollado usted *software* para personas con discapacidad visual?

SI ()

NO ()

7. ¿Qué aspectos considera al evaluar la calidad de *software* para personas con discapacidad visual?

8. ¿Considera que el desarrollo de *software* para personas con discapacidad visual exige un mayor control de calidad?

Anexo 2. Encuesta de Satisfacción



ENCUESTA PARA DESARROLLADORES DE *SOFTWARE* LA EMPRESA ALQUIMIASOFT

Indicaciones: Señor(es) desarrolladores de *software* de la ciudad de Ambato, con el fin de conocer el grado de satisfacción de la aplicación del MODELO DE EVALUACIÓN DE CALIDAD EN EL DESARROLLO DE *SOFTWARE* PARA PERSONAS CON DISCAPACIDAD VISUAL, solicito su ayuda en la siguiente encuesta. Gracias por su colaboración.

PREGUNTAS	SI		NO		NO SÉ	
¿Conoce los beneficios de utilizar un modelo de evaluación de <i>software</i> ?						
¿Es importante utilizar un modelo para evaluar la calidad del <i>software</i> orientado a personas con discapacidad visual?						
¿Usted aplicaría la matriz del Modelo de evaluación de calidad en el desarrollo de <i>software</i> para personas con discapacidad visual?	Me gusta mucho	Me gusta más de lo que me disgusta	Me da lo mismo	Me disgusta más de lo que me gusta	No me gusta nada	No sé qué decir

Anexo 3. Cuadro Lógico de ladov aplicado a desarrolladores

Desarrollador 1	Pregunta 1: ¿Conoce los beneficios de utilizar un modelo de evaluación de calidad en el desarrollo de <i>software</i> ?									
Pregunta 3: ¿Usted aplicaría la matriz del Modelo de evaluación de calidad en el desarrollo de <i>software</i> para personas con discapacidad visual?	NO			NO SÉ			SI			
	Pregunta 2: ¿Es importante utilizar un modelo para evaluar la calidad del <i>software</i> orientado a personas con discapacidad visual?									
	SI	NO SÉ	NO	SI	NO SÉ	NO	SI	NO SÉ	NO	
Me gusta mucho	1	2	6	2	2	6	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6	6
Me da lo mismo	3	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5	5
No sé qué decir	2	3	6	3	3	3	6	3	4	4

Resultado: Clara Satisfacción (1).

Desarrollador 2	Pregunta 1: ¿Conoce los beneficios de utilizar un modelo de evaluación de calidad en el desarrollo de <i>software</i> ?									
Pregunta 3: ¿Usted aplicaría la matriz del Modelo de evaluación de calidad en el desarrollo de <i>software</i> para personas con discapacidad visual?	NO			NO SÉ			SI			
	Pregunta 2: ¿Es importante utilizar un modelo para evaluar la calidad del <i>software</i> orientado a personas con discapacidad visual?									
	SI	NO SÉ	NO	SI	NO SÉ	NO	SI	NO SÉ	NO	
Me gusta mucho	1	2	6	2	2	6	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6	6
Me da lo mismo	3	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5	5
No sé qué decir	2	3	6	3	3	3	6	3	4	4

Resultado: Más satisfecho que insatisfecho(2).

Desarrollador 3	Pregunta 1: ¿Conoce los beneficios de utilizar un modelo de evaluación de calidad en el desarrollo de <i>software</i> ?									
Pregunta 3: ¿Usted aplicaría la matriz del Modelo de evaluación de calidad en el desarrollo de <i>software</i> para personas con discapacidad visual?	NO			NO SÉ			SI			
	Pregunta 2: ¿Es importante utilizar un modelo para evaluar la calidad del <i>software</i> orientado a personas con discapacidad visual?									
	SI	NO SÉ	NO	SI	NO SÉ	NO	SI	NO SÉ	NO	
Me gusta mucho	1	2	6	2	2	6	6	6	6	
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6	
Me da lo mismo	3	3	3	3	3	3	3	3	3	
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4	
No me gusta nada	6	6	6	6	4	4	6	4	5	
No sé qué decir	2	3	6	3	3	3	6	3	4	

Resultado: Clara Satisfacción (1).

Desarrollador 4	Pregunta 1: ¿Conoce los beneficios de utilizar un modelo de evaluación de calidad en el desarrollo de <i>software</i> ?									
Pregunta 3: ¿Usted aplicaría la matriz del Modelo de evaluación de calidad en el desarrollo de <i>software</i> para personas con discapacidad visual?	NO			NO SÉ			SI			
	Pregunta 2: ¿Es importante utilizar un modelo para evaluar la calidad del <i>software</i> orientado a personas con discapacidad visual?									
	SI	NO SÉ	NO	SI	NO SÉ	NO	SI	NO SÉ	NO	
Me gusta mucho	1	2	6	2	2	6	6	6	6	
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6	
Me da lo mismo	3	3	3	3	3	3	3	3	3	
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4	
No me gusta nada	6	6	6	6	4	4	6	4	5	
No sé qué decir	2	3	6	3	3	3	6	3	4	

Resultado: No Definido (3).

Desarrollador 5	Pregunta 1: ¿Conoce los beneficios de utilizar un modelo de evaluación de calidad en el desarrollo de <i>software</i> ?								
Pregunta 3: ¿Usted aplicaría la matriz del Modelo de evaluación de calidad en el desarrollo de <i>software</i> para personas con discapacidad visual?	NO			NO SÉ			SI		
	Pregunta 2: ¿Es importante utilizar un modelo para evaluar la calidad del <i>software</i> orientado a personas con discapacidad visual?								
	SI	NO SÉ	NO	SI	NO SÉ	NO	SI	NO SÉ	NO
Me gusta mucho	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

Resultado: Más insatisfecho que satisfecho (4).

Desarrollador 6	Pregunta 1: ¿Conoce los beneficios de utilizar un modelo de evaluación de calidad en el desarrollo de <i>software</i> ?								
Pregunta 3: ¿Usted aplicaría la matriz del Modelo de evaluación de calidad en el desarrollo de <i>software</i> para personas con discapacidad visual?	NO			NO SÉ			SI		
	Pregunta 2: ¿Es importante utilizar un modelo para evaluar la calidad del <i>software</i> orientado a personas con discapacidad visual?								
	SI	NO SÉ	NO	SI	NO SÉ	NO	SI	NO SÉ	NO
Me gusta mucho	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

Resultado: No Definido (3).

Desarrollador 7	Pregunta 1: ¿Conoce los beneficios de utilizar un modelo de evaluación de calidad en el desarrollo de <i>software</i> ?								
Pregunta 3: ¿Usted aplicaría la matriz del Modelo de evaluación de calidad en el desarrollo de <i>software</i> para personas con discapacidad visual?	NO			NO SÉ			SI		
	Pregunta 2: ¿Es importante utilizar un modelo para evaluar la calidad del <i>software</i> orientado a personas con discapacidad visual?								
	SI	NO SÉ	NO	SI	NO SÉ	NO	SI	NO SÉ	NO
Me gusta mucho	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

Resultado: Más satisfecho que insatisfecho (2).

Desarrollador 8	Pregunta 1: ¿Conoce los beneficios de utilizar un modelo de evaluación de calidad en el desarrollo de <i>software</i> ?								
Pregunta 3: ¿Usted aplicaría la matriz del Modelo de evaluación de calidad en el desarrollo de <i>software</i> para personas con discapacidad visual?	NO			NO SÉ			SI		
	Pregunta 2: ¿Es importante utilizar un modelo para evaluar la calidad del <i>software</i> orientado a personas con discapacidad visual?								
	SI	NO SÉ	NO	SI	NO SÉ	NO	SI	NO SÉ	NO
Me gusta mucho	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

Resultado: Clara Satisfacción (1).

Desarrollador 9	Pregunta 1: ¿Conoce los beneficios de utilizar un modelo de evaluación de calidad en el desarrollo de <i>software</i> ?								
Pregunta 3: ¿Usted aplicaría la matriz del Modelo de evaluación de calidad en el desarrollo de <i>software</i> para personas con discapacidad visual?	NO			NO SÉ			SI		
	Pregunta 2: ¿Es importante utilizar un modelo para evaluar la calidad del <i>software</i> orientado a personas con discapacidad visual?								
	SI	NO SÉ	NO	SI	NO SÉ	NO	SI	NO SÉ	NO
Me gusta mucho	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

Resultado: Más satisfecho que insatisfecho (2).

Desarrollador 10	Pregunta 1: ¿Conoce los beneficios de utilizar un modelo de evaluación de calidad en el desarrollo de <i>software</i> ?								
Pregunta 3: ¿Usted aplicaría la matriz del Modelo de evaluación de calidad en el desarrollo de <i>software</i> para personas con discapacidad visual?	NO			NO SÉ			SI		
	Pregunta 2: ¿Es importante utilizar un modelo para evaluar la calidad del <i>software</i> orientado a personas con discapacidad visual?								
	SI	NO SÉ	NO	SI	NO SÉ	NO	SI	NO SÉ	NO
Me gusta mucho	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

Resultado: Clara satisfacción (1).

Anexo 4. Validación Matriz

MODELO DE EVALUACIÓN DE CALIDAD EN EL DESARROLLO DE SOFTWARE PARA PERSONAS CON DISCAPACIDAD VISUAL A NIVEL PRODUCTO

Nombre del software o aplicación: Braille Teacher

Fecha: 05/06/19

Nombre del responsable a evaluar: Francisco Javier García Camacho

Braille Teacher
Realizado por Ing. Francisco Javier García Camacho

Seguridad de Información		Descripción	Preguntas			
Integridad	Garantiza que la información se guarde apropiadamente y no sea modificada por personas externas.	¿Se encuentra la información que compone el sistema protegida ante modificaciones de personas externas?	SI	x	NO	
Confidencialidad	Garantiza que la información sea accesible únicamente a personas autorizadas.	¿Es la información del sistema accedida únicamente por personas autorizadas?	SI	x	NO	
Disponibilidad	Que la información esté accesible en cualquier momento.	¿Los usuarios acceder sin ningún inconveniente a la información?	SI	x	NO	
Mantenibilidad		Descripción	Preguntas			
Estabilidad	Que el nivel de fallos del software sea mínimo.	¿Considera que el nivel de fallos es mínimo?	SI	x	NO	
Reusabilidad	Nivel en que pueda volver a utilizar el software o parte de él en otras aplicaciones.	¿Considera que se reutilizará algún módulo o componentes del software en otras aplicaciones?	SI	x	NO	
Facilidad de prueba	Es la facilidad con la que se llega a probar un programa.	¿Es fácil realizar las pruebas de funcionamiento?	SI	x	NO	
Facilidad de mantenimiento	Facilita al desarrollador ubicar un error y este consigue ser solucionado.	¿El desarrollador ubica un error con facilidad y lo soluciona?	SI	x	NO	
Portabilidad						
Adaptabilidad	Facilidad de adecuación del sistema a otros entornos y perdura la optimización del sistema.	¿El software consigue adaptarse a otros entornos multiplataforma?	SI		NO	x
Facilidad de instalación	El software logra instalarse y utilizarse en otros entornos de hardware y software.	¿Es el software fácil en su instalación?	SI	x	NO	
Confiabilidad						
Exactitud	Los resultados que arroja el sistema son apropiados.	¿El software se ejecuta y funciona precisamente a lo prescrito?	SI	x	NO	
Tolerancia a fallos	Capacidad del software de seguir funcionando en caso de que falle algún componente o módulos, y que no exista pérdida de información.	¿El sistema seguirá funcionando correctamente en caso de fallo de alguno de sus componentes?	SI	x	NO	

A NIVEL PRODUCTO

Nombre del software o aplicación: Braille Teacher

Fecha: 05/06/19

Nombre del responsable a evaluar: Nancy Kenwa

**MODELO DE EVALUACIÓN DE CALIDAD EN EL DESARROLLO DE SOFTWARE PARA PERSONAS CON DISCAPACIDAD VISUAL
A NIVEL PROCESO**

Nombre del software o aplicación: Recurso Didáctico Multimedia

Fecha: 05/06/19

Nombre del responsable a evaluar: Bolívar Leonardo Ayavaca

Accesibilidad		Descripción	Preguntas			
Eficiente	Que el software se desempeñe adecuadamente con los recursos utilizados.	¿El software realiza las operaciones en el menor tiempo posible?	SI	x	NO	
Comprensible	Facilidad con la que el usuario interactúa con el programa.	¿El usuario entiende con facilidad las instrucciones para el uso del software?	SI	x	NO	
Intuitivo	El software tiene que entenderse y comprenderse de inmediato.	¿El software es fácil de comprender?	SI	x	NO	
Usabilidad		Descripción	Preguntas			
Satisfacción	Grado con el que el software cumple con los requerimientos.	¿El software cumple con los requerimientos establecidos?	SI	x	NO	
Facilidad de uso	Agilidad con la que el usuario alcanza a desarrollar las tareas después de conocer el funcionamiento del sistema.	¿El usuario interactúa con facilidad con el sistema?	SI	x	NO	

Seguridad de Información		Descripción	Preguntas	SI	NO	x
Integridad	Garantiza que la información se guarde apropiadamente y no sea modificada por personas externas.	¿Se encuentra la información que compone el sistema protegida ante modificaciones de personas externas?	SI	NO	x	
Confidencialidad	Garantiza que la información sea accesible únicamente a personas autorizadas.	¿Es la información del sistema accedida únicamente por personas autorizadas?	SI	NO	x	
Disponibilidad	Que la información esté accesible en cualquier momento.	¿Los usuarios acceden sin ningún inconveniente a la información?	SI	NO	x	

Mantenibilidad		Descripción	Preguntas	SI	NO	x
Estabilidad	Que el nivel de fallos del software sea mínimo.	¿Considera que el nivel de fallos es mínimo?	SI	NO	x	
Reusabilidad	Nivel en que pueda volver a utilizar el software o parte de él en otras aplicaciones.	¿Considera que se alcanza a reutilizar algún módulo o componentes del software en otras aplicaciones?	SI	NO	x	
Facilidad de prueba	Es la facilidad con la que se logra probar un programa.	¿Es fácil realizar las pruebas de funcionamiento?	SI	NO	x	
Facilidad de mantenimiento	Facilita al desarrollador ubicar un error y este logra ser solucionado.	¿El desarrollador ubica un error con facilidad y lo soluciona?	SI	NO	x	
Portabilidad						
Adaptabilidad	Facilidad de adecuación del sistema a otros entornos y lograr mantener la optimización del sistema.	¿El software logra adaptarse a otros entornos multiplataforma?	SI	NO	x	
Facilidad de instalación	El software consigue instalarse y utilizarse en otros entornos de hardware y software.	¿Es el software fácil en su instalación?	SI	NO	x	
Confiabilidad						
Exactitud	Los resultados que arroja el sistema son apropiados.	¿El software se ejecuta y funciona precisamente a lo prescrito?	SI	NO	x	
Tolerancia a fallos	Capacidad del software de seguir funcionando en caso de que falle algún componente o módulos, y que no exista pérdida de información.	¿El sistema sigue funcionando correctamente en caso de fallo de alguno de sus componentes?	SI	NO	x	

A NIVEL PRODUCTO

Nombre del software o aplicación: Recurso Didáctica Multimedia

Fecha: 05/06/19

Nombre del responsable a evaluar: Bolívar Leonardo Ayavaca

A NIVEL PRODUCTO

Nombre del software o aplicación: Braille Teacher

Fecha: 05/06/19

Nombre del responsable a evaluar: Alejandro Castro

Accesibilidad		Descripción	Preguntas			
Eficiente	Que el software se desempeñe adecuadamente con los recursos utilizados.	¿El software realiza las operaciones en el menor tiempo posible?	SI	x	NO	
Comprensible	Facilidad con la que el usuario interactúa con el programa.	¿El usuario entiende con facilidad las instrucciones para el uso del software?	SI	x	NO	
Intuitivo	El software tiene que entenderse y comprenderse de inmediato.	¿El software es fácil de comprender?	SI	x	NO	
Usabilidad		Descripción	Preguntas			
Satisfacción	Grado con el que el software cumple con los requerimientos.	¿El software cumple con los requerimientos establecidos?	SI	x	NO	
Facilidad de uso	Agilidad con la que el usuario alcanza a desarrollar las tareas después de conocer el funcionamiento del sistema.	¿El usuario consigue interactuar con facilidad con el sistema?	SI	x	NO	

Accesibilidad		Descripción	Preguntas			
Eficiente	Que el software se desempeñe adecuadamente con los recursos utilizados.	¿El software realiza las operaciones en el menor tiempo posible?	SI	X	NO	
Comprensible	Facilidad con la que el usuario interactúa con el programa.	¿El usuario entiende con facilidad las instrucciones para el uso del software?	SI	X	NO	
Intuitivo	El software tiene que entenderse y comprenderse de inmediato.	¿El software es fácil de comprender?	SI	X	NO	
Usabilidad		Descripción	Preguntas			
Satisfacción	Grado con el que el software cumple con los requerimientos.	¿El software cumple con los requerimientos establecidos?	SI	x	NO	
Facilidad de uso	Agilidad con la que el usuario alcanza a desarrollar las tareas después de conocer el funcionamiento del sistema.	¿El usuario interactúa con facilidad con el sistema?	SI	x	NO	

MODELO DE EVALUACIÓN DE CALIDAD EN EL DESARROLLO DE SOFTWARE PARA PERSONAS CON DISCAPACIDAD VISUAL

Nombre del software o aplicación: NUDA Fecha: 05-06-2019

Nombre del responsable a evaluar: Estefanía Llovera

A NIVEL PRODUCTO

Accesibilidad		Preguntas	
Eficiente	Que el <i>software</i> se desempeña adecuadamente con los recursos utilizados.	¿El <i>software</i> realiza las operaciones en el menor tiempo posible?	SI <input checked="" type="checkbox"/> NO <input type="checkbox"/>
Comprensible	Facilidad con la que el usuario puede interactuar con el programa.	¿El usuario entiende con facilidad las instrucciones para el uso del <i>software</i> ?	SI <input type="checkbox"/> NO <input checked="" type="checkbox"/>
Intuitivo	El <i>software</i> tiene que entenderse y comprenderse de inmediato.	¿El <i>software</i> es fácil de comprender?	SI <input type="checkbox"/> NO <input checked="" type="checkbox"/>
Usabilidad		Preguntas	
Satisfacción	Grado con el que el <i>software</i> cumple con los requerimientos.	¿El <i>software</i> cumple con los requerimientos establecidos?	SI <input type="checkbox"/> NO <input checked="" type="checkbox"/>
Facilidad de uso	Agilidad con la que el usuario pueden desarrollar las tareas después de conocer el funcionamiento del sistema.	¿El usuario puede interactuar con facilidad con el sistema?	SI <input type="checkbox"/> NO <input checked="" type="checkbox"/>

MODELO DE EVALUACIÓN DE CALIDAD EN EL DESARROLLO DE SOFTWARE PARA PERSONAS CON DISCAPACIDAD VISUAL

Nombre del software o aplicación: Moodle Fecha: 05/06/2019

Nombre del responsable a evaluar: Estefanía Cerezo

A NIVEL PRODUCTO

Accesibilidad		Descripción		Preguntas	
Eficiente	Que el software se desempeña adecuadamente con los recursos utilizados.	¿El software realiza las operaciones en el menor tiempo posible?	SI	X	NO
Comprensible	Facilidad con la que el usuario puede interactuar con el programa.	¿El usuario entiende con facilidad las instrucciones para el uso del software?	SI	X	NO
Intuitivo	El software tiene que entenderse y comprenderse de inmediato.	¿El software es fácil de comprender?	SI	X	NO
Usabilidad		Descripción		Preguntas	
Satisfacción	Grado con el que el software cumple con los requerimientos.	¿El software cumple con los requerimientos establecidos?	SI	X	NO
Facilidad de uso	Agilidad con la que el usuario pueden desarrollar las tareas después de conocer el funcionamiento del sistema.	¿El usuario puede interactuar con facilidad con el sistema?	SI	X	NO

MODELO DE EVALUACIÓN DE CALIDAD EN EL DESARROLLO DE SOFTWARE PARA PERSONAS CON DISCAPACIDAD VISUAL

Nombre del software o aplicación: Moodle Fecha: 04/06/2019
 Nombre del responsable a evaluar: Orlando Salena

A NIVEL PRODUCTO

Accesibilidad		Preguntas	
Eficiente	Que el software se desempeña adecuadamente con los recursos utilizados.	¿El software realiza las operaciones en el menor tiempo posible?	SI <input checked="" type="checkbox"/> NO <input type="checkbox"/>
Comprensible	Facilidad con la que el usuario puede interactuar con el programa.	¿El usuario entiende con facilidad las instrucciones para el uso del software?	SI <input checked="" type="checkbox"/> NO <input type="checkbox"/>
Intuitivo	El software tiene que entenderse y comprenderse de inmediato.	¿El software es fácil de comprender?	SI <input checked="" type="checkbox"/> NO <input type="checkbox"/>
Usabilidad		Preguntas	
Satisfacción	Grado con el que el software cumple con los requerimientos.	¿El software cumple con los requerimientos establecidos?	SI <input checked="" type="checkbox"/> NO <input type="checkbox"/>
Facilidad de uso	Agilidad con la que el usuario pueden desarrollar las tareas después de conocer el funcionamiento del sistema.	¿El usuario puede interactuar con facilidad con el sistema?	SI <input checked="" type="checkbox"/> NO <input type="checkbox"/>

Orlando Salena

MODELO DE EVALUACIÓN DE CALIDAD EN EL DESARROLLO DE SOFTWARE PARA PERSONAS CON DISCAPACIDAD VISUAL

Nombre del software o aplicación: Moodle PUCESA

Fecha: 04/06/2019

Nombre del responsable a evaluar: Grace Sánchez (Experto Psicología)

A NIVEL PRODUCTO

Accesibilidad		Descripción		Preguntas	
Eficiente	Que el software se desempeña adecuadamente con los recursos utilizados.	¿El software realiza las operaciones en el menor tiempo posible?	SI	X	NO
Comprensible	Facilidad con la que el usuario puede interactuar con el programa.	¿El usuario entiende con facilidad las instrucciones para el uso del software?	SI	X	NO
Intuitivo	El software tiene que entenderse y comprenderse de inmediato.	¿El software es fácil de comprender?	SI	X	NO
Usabilidad		Descripción		Preguntas	
Satisfacción	Grado con el que el software cumple con los requerimientos.	¿El software cumple con los requerimientos establecidos?	SI	X	NO
Facilidad de uso	Agilidad con la que el usuario pueden desarrollar las tareas después de conocer el funcionamiento del sistema.	¿El usuario puede interactuar con facilidad con el sistema?	SI	X	NO

Grace Sánchez

MODELO DE EVALUACIÓN DE CALIDAD EN EL DESARROLLO DE SOFTWARE PARA PERSONAS CON DISCAPACIDAD VISUAL

Nombre del software o aplicación: Academics Fecha: 24/06/2019

Nombre del responsable a evaluar: Delando Sailema

A NIVEL PRODUCTO

Accesibilidad		Preguntas	
Eficiente	Que el software se desempeñe adecuadamente con los recursos utilizados.	¿El software realiza las operaciones en el menor tiempo posible?	SI <input checked="" type="checkbox"/> NO <input type="checkbox"/>
Comprendible	Facilidad con la que el usuario puede interactuar con el programa.	¿El usuario entiende con facilidad las instrucciones para el uso del software?	SI <input type="checkbox"/> NO <input checked="" type="checkbox"/>
Intuitivo	El software tiene que entenderse y comprenderse de inmediato.	¿El software es fácil de comprender?	SI <input checked="" type="checkbox"/> NO <input type="checkbox"/>
Usabilidad		Preguntas	
Satisfacción	Grado con el que el software cumple con los requerimientos.	¿El software cumple con los requerimientos establecidos?	SI <input checked="" type="checkbox"/> NO <input type="checkbox"/>
Facilidad de uso	Agilidad con la que el usuario pueden desarrollar las tareas después de conocer el funcionamiento del sistema.	¿El usuario puede interactuar con facilidad con el sistema?	SI <input type="checkbox"/> NO <input checked="" type="checkbox"/>

Delando Sailema

MODELO DE EVALUACIÓN DE CALIDAD EN EL DESARROLLO DE SOFTWARE PARA PERSONAS CON DISCAPACIDAD VISUAL

Nombre del software o aplicación: Grace Sanchez

Fecha: 04/06/2019

Nombre del responsable a evaluar: Academix

A NIVEL PRODUCTO

Accesibilidad	Descripción	Preguntas	SI	NO
Eficiente	Que el software se desempeña adecuadamente con los recursos utilizados.	¿El software realiza las operaciones en el menor tiempo posible?	SI	NO
Comprensible	Facilidad con la que el usuario puede interactuar con el programa.	¿El usuario entiende con facilidad las instrucciones para el uso del software?	SI	NO
Intuitivo	El software tiene que entenderse y comprenderse de inmediato.	¿El software es fácil de comprender?	SI	NO
Usabilidad	Descripción	Preguntas	SI	NO
Satisfacción	Grado con el que el software cumple con los requerimientos.	¿El software cumple con los requerimientos establecidos?	SI	NO
Facilidad de uso	Agilidad con la que el usuario pueden desarrollar las tareas después de conocer el funcionamiento del sistema.	¿El usuario puede interactuar con facilidad con el sistema?	SI	NO

Grace Sanchez

MODELO DE EVALUACIÓN DE CALIDAD EN EL DESARROLLO DE SOFTWARE PARA PERSONAS CON DISCAPACIDAD VISUAL

Nombre del software o aplicación: Academicas

Nombre del responsable a evaluar: Estefanía Herrera

Fecha: 05/06/2019

A NIVEL PRODUCTO

Accesibilidad	Descripción	Preguntas	
Eficiente	Que el software se desempeña adecuadamente con los recursos utilizados.	¿El software realiza las operaciones en el menor tiempo posible?	SI <input checked="" type="checkbox"/> NO <input type="checkbox"/>
Comprensible	Facilidad con la que el usuario puede interactuar con el programa.	¿El usuario entiende con facilidad las instrucciones para el uso del software?	SI <input type="checkbox"/> NO <input checked="" type="checkbox"/>
Intuitivo	El software tiene que entenderse y comprenderse de inmediato.	¿El software es fácil de comprender?	SI <input checked="" type="checkbox"/> NO <input type="checkbox"/>
Usabilidad	Descripción	Preguntas	
Satisfacción	Grado con el que el software cumple con los requerimientos.	¿El software cumple con los requerimientos establecidos?	SI <input type="checkbox"/> NO <input checked="" type="checkbox"/>
Facilidad de uso	Agilidad con la que el usuario pueden desarrollar las tareas después de conocer el funcionamiento del sistema.	¿El usuario puede interactuar con facilidad con el sistema?	SI <input type="checkbox"/> NO <input checked="" type="checkbox"/>

MODELO DE EVALUACIÓN DE CALIDAD EN EL DESARROLLO DE SOFTWARE PARA PERSONAS CON DISCAPACIDAD VISUAL

Nombre del software o aplicación: Moodle

Fecha: 31/05/2019

Nombre del responsable a evaluar: Ing. José Enriquez

A NIVEL PROCESO

Mantenibilidad	Descripción	Preguntas	
Estabilidad	Que el nivel de fallos sea mínimo	SI	NO
Reusabilidad	Nivel en que pueda volver a utilizar el software o parte de él en otras aplicaciones.	SI	NO
Facilidad de prueba	Es la facilidad con la que se puede probar un programa.	SI	NO
Facilidad de mantenimiento	Facilita al desarrollador ubicar un error y este puede ser solucionado.	SI	NO
Portabilidad			
Adaptabilidad	Facilidad de adecuación del sistema a otros entornos y manteniendo la optimización del sistema.	SI	NO
Facilidad de instalación	El software puede instalarse y utilizarse en otros entornos de hardware y software.	SI	NO
Confiabilidad			
Exactitud	Los resultados que arroja el sistema son apropiados.	SI	NO
Tolerancia a fallos	Capacidad del software de seguir funcionando en caso de que falle algún componente o módulos, y que no exista pérdida de información.	SI	NO

Seguridad de Información	Descripción	Preguntas		
Integridad	Garantiza que la información se guarde apropiadamente y no sea modificada por personas externas.	¿Se encuentra la información que compone el sistema protegida ante modificaciones de personas externas?	SI	NO
Confidencialidad	Garantiza que la información sea accesible únicamente a personas autorizadas.	¿Es la información del sistema accedida únicamente por personas autorizadas?	SI	NO
Disponibilidad	Que la información esté accesible en cualquier momento.	¿Pueden los usuarios acceder sin ningún inconveniente a la información?	SI	NO

Observaciones: _____

Firma: 

MODELO DE EVALUACIÓN DE CALIDAD EN EL DESARROLLO DE SOFTWARE PARA PERSONAS CON DISCAPACIDAD VISUAL

Nombre del software o aplicación: Academics Fecha: 31 Oct 2019

Nombre del responsable a evaluar: José Enrique

A NIVEL PROCESO

Mantenibilidad	Descripción	Preguntas			
Estabilidad	Que el nivel de fallos sea mínimo	¿Considera que el nivel de fallos es mínimo?	SI	NO	X
Reusabilidad	Nivel en que pueda volver a utilizar el software o parte de él en otras aplicaciones.	¿Considera que se puede reutilizar algún módulo o componentes del software en otras aplicaciones?	SI	NO	X
Facilidad de prueba	Es la facilidad con la que se puede probar un programa.	¿Es fácil realizar las pruebas de funcionamiento?	SI	NO	X
Facilidad de mantenimiento	Facilita al desarrollador ubicar un error y este puede ser solucionado.	¿Puede el desarrollador ubicar un error con facilidad y solucionarlo?	SI	NO	X
Portabilidad					
Adaptabilidad	Facilidad de adecuación del sistema a otros entornos y manteniendo la optimización del sistema.	¿El software puede adaptarse a otros entornos multiplataforma?	SI	NO	X
Facilidad de instalación	El software puede instalarse y utilizarse en otros entornos de hardware y software.	¿Es el software fácil en su instalación?	SI	NO	X
Confiabilidad					
Exactitud	Los resultados que arroja el sistema son apropiados.	¿El software se ejecuta y funciona precisamente a lo prescrito?	SI	NO	X
Tolerancia a fallos	Capacidad del software de seguir funcionando en caso de que falle algún componente o módulos, y que no exista pérdida de información.	¿El sistema puede seguir funcionando correctamente en caso de fallo de alguno de sus componentes?	SI	NO	X

Seguridad de Información	Descripción	Preguntas		
Integridad	Garantiza que la información se guarde apropiadamente y no sea modificada por personas externas.	¿Se encuentra la información que compone el sistema protegida ante modificaciones de personas externas?	SI	NO
Confidencialidad	Garantiza que la información sea accesible únicamente a personas autorizadas.	¿Es la información del sistema accedida únicamente por personas autorizadas?	SI	NO
Disponibilidad	Que la información esté accesible en cualquier momento.	¿Pueden los usuarios acceder sin ningún inconveniente a la información?	SI	NO

Observaciones: _____

Firma: 

MODELO DE EVALUACIÓN DE CALIDAD EN EL DESARROLLO DE SOFTWARE PARA PERSONAS CON DISCAPACIDAD VISUAL

Nombre del software o aplicación: NUNDA
 Nombre del responsable a evaluar: Ing. Eduardo Pemeche

Fecha: 28/05/2019

A NIVEL PROCESO

Mantenibilidad	Descripción	Preguntas	SI	NO
Estabilidad	Que el nivel de fallos sea mínimo	¿Considera que el nivel de fallos es mínimo?	SI	NO
Reusabilidad	Nivel en que pueda volver a utilizar el software o parte de él en otras aplicaciones.	¿Considera que se puede reutilizar algún módulo o componentes del software en otras aplicaciones?	SI	NO
Facilidad de prueba	Es la facilidad con la que se puede probar un programa.	¿Es fácil realizar las pruebas de funcionamiento?	SI	NO
Facilidad de mantenimiento	Facilita al desarrollador ubicar un error y este puede ser solucionado.	¿Puede el desarrollador ubicar un error con facilidad y solucionarlo?	SI	NO
Portabilidad				
Adaptabilidad	Facilidad de adecuación del sistema a otros entornos y manteniendo la optimización del sistema. El software puede instalarse y utilizarse en otros entornos de hardware y software.	¿El software puede adaptarse a otros entornos multiplataforma?	SI	NO
Facilidad de instalación		¿Es el software fácil en su instalación?	SI	NO
Confiabilidad				
Exactitud	Los resultados que arroja el sistema son apropiados.	¿El software se ejecuta y funciona precisamente a lo prescrito?	SI	NO
Tolerancia a fallos	Capacidad del software de seguir funcionando en caso de que falle algún componente o módulos, y que no exista pérdida de información.	¿El sistema puede seguir funcionando correctamente en caso de fallo de alguno de sus componentes?	SI	NO

Seguridad de Información	Descripción	Preguntas	
Integridad	Garantiza que la información se guarde apropiadamente y no sea modificada por personas externas.	¿Se encuentra la información que compone el sistema protegida ante modificaciones de personas externas?	SI <input checked="" type="checkbox"/> NO <input type="checkbox"/>
Confidencialidad	Garantiza que la información sea accesible únicamente a personas autorizadas.	¿Es la información del sistema accedida únicamente por personas autorizadas?	SI <input checked="" type="checkbox"/> NO <input type="checkbox"/>
Disponibilidad	Que la información esté accesible en cualquier momento.	¿Pueden los usuarios acceder sin ningún inconveniente a la información?	SI <input checked="" type="checkbox"/> NO <input type="checkbox"/>

Observaciones: _____


Firma:

