



# **ESPE**

**UNIVERSIDAD DE LAS FUERZAS ARMADAS**  
**INNOVACIÓN PARA LA EXCELENCIA**

**VICERRECTORADO DE LA INVESTIGACIÓN, INNOVACIÓN Y  
TRANSFERENCIA DE TECNOLOGÍA**

**CENTRO DE POSGRADOS**

**MAESTRIA EN GERENCIA DE SISTEMAS**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO  
DE MAGISTER EN GERENCIA DE SISTEMAS**

**TEMA: METODOLOGÍA PARA LA EVALUACIÓN DE LA CALIDAD DEL  
SOFTWARE, Y CASO DE ESTUDIO SISTEMA DE GESTIÓN  
ACADÉMICA DE LA ESPAM MFL**

**AUTORES:**

**RÍOS SALGADO SANTIAGO RAMIRO**

**VILLAO LOOR MARÍA BELÉN**

**DIRECTORA: GUERRERO IDROVO, ROSA GRACIELA**

**SANGOLQUÍ**

**2018**



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y  
TRANSFERENCIA DE TECNOLOGÍA

CENTRO DE POSGRADOS

## CERTIFICACIÓN

Certifico que el trabajo de titulación ***"METODOLOGÍA PARA LA EVALUACIÓN DE LA CALIDAD DEL SOFTWARE, Y CASO DE ESTUDIO SISTEMA DE GESTIÓN ACADÉMICA DE LA ESPAM MFL"*** fue realizado por los Ingenieros: **SANTIAGO RAMIRO RÍOS SALGADO** y **MARÍA BELÉN VILLO LOOR**, el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 6 de Noviembre del 2018

Una firma manuscrita en tinta azul que parece decir "Graciela Guerrero Idrovo".

Ing. Graciela Guerrero Idrovo, MSc.

CC: 1720513322



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y  
TRANSFERENCIA DE TECNOLOGÍA

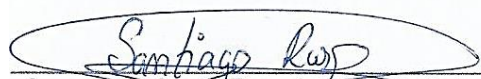
CENTRO DE POSGRADOS

## AUTORÍA RESPONSABILIDAD

Nosotros, **SANTIAGO RAMIRO RÍOS SALGADO**, con cédula de ciudadanía No. 1717824401 y **MARÍA BELÉN VILLAO LOOR** con cédula de ciudadanía No. 1314189091, declaramos que el contenido, ideas y criterios del trabajo de titulación **"METODOLOGÍA PARA LA EVALUACIÓN DE LA CALIDAD DEL SOFTWARE, Y CASO DE ESTUDIO SISTEMA DE GESTIÓN ACADÉMICA DE LA ESPAM MFL"** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz

Sangolquí, 6 de Noviembre del 2018

  
Ing. Santiago Ramiro Ríos Salgado  
CC. 1717824401

  
Ing. María Belén Villao Loor  
CC. 1314189091



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y  
TRANSFERENCIA DE TECNOLOGÍA

CENTRO DE POSGRADOS

## AUTORIZACIÓN

Nosotros, **SANTIAGO RAMIRO RÍOS SALGADO**, con cédula de ciudadanía No. 1717824401 y **MARÍA BELÉN VILLAGO LOOR**, con cédula de ciudadanía No. 1314189091, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación "**METODOLOGÍA PARA LA EVALUACIÓN DE LA CALIDAD DEL SOFTWARE, Y CASO DE ESTUDIO SISTEMA DE GESTIÓN ACADÉMICA DE LA ESPAM MFL**" en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Sangolquí, 6 de Noviembre del 2018

  
Ing. Santiago Ramiro Ríos Salgado

CC. 1717824401

  
Ing. María Belén Villao Loor

CC. 1314189091

## **DEDICATORIA**

A mi familia que, con su ejemplo e incondicional apoyo a través del tiempo, se han convertido en el pilar fundamental en mi vida. Todo este trabajo ha sido posible gracias a ellos.

**Santiago Ramiro Ríos Salgado**

## DEDICATORIA

Marina y Larissa por ustedes y para ustedes, fueron mi motivación más grande para concluir con éxito esta etapa, el cariño que ustedes me dan son el motivo de mi felicidad, de mi esfuerzo y de mis ganas de ser mejor cada día. Por ser el lado dulce y la alegría de mi vida les dedico este trabajo.

**María Belén Villao Loor**

## **AGRADECIMIENTO**

### **A Dios**

Por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor.

### **A mis padres y a mi hermano**

Por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada por su amor.

### **A Sandrita y a Sol Emilia**

Por darme todo su apoyo y amor, así como su paciencia para concluir con una meta más, así como todas las que nos faltan juntos.

**Santiago Ramiro Ríos Salgado**

## **AGRADECIMIENTO**

### **A Dios**

Guía de mis pasos y luz de mi sendero, proveedor de salud y trabajo, fuerza motora para seguir adelante cada día

### **A mis Padres y hermano**

Tomada de sus manos inicié mi aprendizaje en la vida, hoy casi todo lo que soy se los debo a sus ejemplos de esfuerzos y tenacidad.

### **A mi esposo**

Gracias por confiar en mí, cuando creía que ya no me quedaban fuerzas para continuar, su apoyo fue de vital importancia para concluir con éxito esta etapa de mi vida.

**María Belen Villao Loor**



## ÍNDICE

CERTIFICACION .....	i
AUTORÍA DE RESPONSABILIDAD .....	ii
AUTORIZACIÓN .....	iii
DEDICATORIA .....	iv
DEDICATORIA .....	v
AGRADECIMIENTO.....	vi
AGRADECIMIENTO.....	vii
ÍNDICE.....	viii
ÍNDICE DE TABLAS .....	xii
ÍNDICE DE FIGURAS.....	xiv
RESUMEN .....	xvi
ABSTRACT .....	xvii
CAPÍTULO I.....	1
INTRODUCCIÓN.....	1
1.1 Antecedentes.....	1
1.2 Planteamiento Del Problema De Investigación .....	3

1.2.1	Análisis de los involucrados.....	5
1.2.2	Árbol de problemas.....	6
1.2.3	Árbol de objetivos.....	6
1.2.4	Análisis de alternativas.....	7
1.2.5	Estructura analítica del proyecto.....	8
1.3	Formulación del Problema.....	8
1.4	Justificación e Importancia.....	9
1.5	Objetivos.....	10
1.5.1	Objetivo General.....	10
1.5.2	Objetivos Específicos.....	10
1.6	Alcance.....	10
CAPÍTULO II.....		11
MARCO TEÓRICO.....		11
2.1	Bases teóricas de la Calidad de Software.....	11
2.1.1	Software.....	11
2.1.2	Calidad.....	12
2.1.3	Evaluación de la Calidad.....	20
2.2	Modelo de Evaluación de Calidad del Software.....	29

2.2.1	Modelos Existentes (cuadro comparativo).....	30
2.3	Descripción de estándares escogidos para trabajo de investigación.....	35
2.3.1	Norma ISO/IEC 25000.....	35
2.3.2	Metodología Bootstrap.....	37
CAPÍTULO III .....		52
DEFINICIÓN DE LA METODOLOGÍA .....		52
3.1	Metodología de Desarrollo Propuesta.....	52
3.1.1	Determinación de los componentes de la metodología .....	52
3.1.2	Planteamiento de la metodología.....	111
CAPÍTULO IV.....		114
VALIDACIÓN DE LA METODOLOGÍA A TRAVÉS DE UN CASO DE ESTUDIO .....		114
4.1	Desarrollo del caso de estudio .....	114
4.1.1	Objetivo.....	114
4.1.2	Sistema de Gestión Académica (gestión notas y matrículas).....	114
4.1.3	Proceso de Evaluación .....	115
4.1.4	Discusión.....	142
CAPÍTULO V.....		143
CONCLUSIONES Y RECOMENDACIONES.....		143

5.1.	Conclusiones.....	143
5.2.	Recomendaciones.....	144
	BIBLIOGRAFÍA .....	145

## ÍNDICE DE TABLAS

<b>Tabla 1</b>	<i>Análisis de los involucrados</i> .....	5
<b>Tabla 2</b>	<i>Análisis de usuarios</i> .....	6
<b>Tabla 3</b>	<i>Árbol de problemas</i> .....	8
<b>Tabla 4</b>	<i>Cuadro Comparativo de Modelos de Evaluación del Software</i> .....	30
<b>Tabla 5</b>	<i>Fases de la calidad del software</i> .....	53
<b>Tabla 6</b>	<i>Métricas</i> .....	63
<b>Tabla 7</b>	<i>Métricas de Calidad en Uso</i> .....	89
<b>Tabla 8</b>	<i>Variable para fórmulas</i> .....	99
<b>Tabla 9</b>	<i>Fórmulas de cálculo de calidad</i> .....	99
<b>Tabla 10</b>	<i>Escalas de Calidad</i> .....	100
<b>Tabla 11</b>	<i>Etapa 1</i> .....	100
<b>Tabla 12</b>	<i>Etapa 2</i> .....	103
<b>Tabla 13</b>	<i>Etapa 3</i> .....	105
<b>Tabla 14</b>	<i>Etapa 4</i> .....	107
<b>Tabla 15</b>	<i>Etapa 5</i> .....	109
<b>Tabla 16</b>	<i>Perfiles</i> .....	112
<b>Tabla 17</b>	<i>Entrevista para obtener información</i> .....	115
<b>Tabla 18</b>	<i>Tabla nivel de importancia de características y sub-características</i> ..	116
<b>Tabla 19</b>	<i>Fórmula de muestreo</i> .....	117
<b>Tabla 20</b>	<i>Muestreo</i> .....	117
<b>Tabla 21</b>	<i>Docentes calificación de experiencias de elaboración de rúbricas</i> ...	119
<b>Tabla 22</b>	<i>Calificación experiencia en proceso de ingreso de calificaciones</i> .....	120

<b>Tabla 23</b>	<i>Calificación experiencia en proceso consulta de reportes materia....</i>	121
<b>Tabla 24</b>	<i>Calificación de experiencia en el proceso de matriculación .....</i>	122
<b>Tabla 25</b>	<i>Calificación de experiencia en el proceso de consulta de notas .....</i>	123
<b>Tabla 26</b>	<i>Califique su experiencia en procesos de consulta de malla.....</i>	124
<b>Tabla 27</b>	<i>Consulta trayectoria académica .....</i>	125
<b>Tabla 28</b>	<i>Califique la rapidez del sistema.....</i>	125
<b>Tabla 29</b>	<i>Califique la facilidad de uso de la aplicación .....</i>	126
<b>Tabla 30</b>	<i>Conocimiento de Manuales de usuarios .....</i>	127
<b>Tabla 31</b>	<i>Calificación de herramientas de ayuda .....</i>	128
<b>Tabla 32</b>	<i>Conocimiento acerca del manejo de la aplicación .....</i>	129
<b>Tabla 33</b>	<i>Mensajes claros .....</i>	130
<b>Tabla 34</b>	<i>Disponibilidad de la aplicación .....</i>	130
<b>Tabla 35</b>	<i>Cuenta con usuario y contraseña.....</i>	131
<b>Tabla 36</b>	<i>Conoce si el sistema cuenta con funciones para discapacitados.....</i>	132
<b>Tabla 37</b>	<i>Personalización interfaz gráfica .....</i>	132
<b>Tabla 38</b>	<i>Uso en tablets, smartphones o similares .....</i>	133
<b>Tabla 39</b>	<i>Califique la experiencia en estos dispositivos .....</i>	134
<b>Tabla 40</b>	<i>Calificación en General.....</i>	135
<b>Tabla 41</b>	<i>Análisis general de resultados .....</i>	136
<b>Tabla 42</b>	<i>Cálculo de datos Calidad Externa .....</i>	137
<b>Tabla 43</b>	<i>Calificación de Calidad Externa .....</i>	140
<b>Tabla 44</b>	<i>Calificación de Calidad Externa .....</i>	141
<b>Tabla 45</b>	<i>Calificación Final.....</i>	141

## ÍNDICE DE FIGURAS

<b>Figura 1</b> Árbol de problemas .....	6
<b>Figura 2</b> Árbol de objetivos .....	6
<b>Figura 3</b> Metodología propuesta .....	52
<b>Figura 4</b> Calidad externa .....	54
<b>Figura 5</b> Calidad en uso.....	59
<b>Figura 6</b> Etapa 1 .....	102
<b>Figura 7</b> Etapa 2 .....	104
<b>Figura 8</b> Etapa 3 .....	106
<b>Figura 9</b> Etapa 4 .....	108
<b>Figura 10</b> Etapa 5 .....	110
<b>Figura 11</b> Etapas de metodología.....	112
<b>Figura 12</b> Experiencia elaboración de rúbricas.....	119
<b>Figura 13</b> Experiencia ingreso de calificaciones.....	120
<b>Figura 14</b> Calificación reporte de materias .....	121
<b>Figura 15</b> Experiencia proceso de matriculación .....	122
<b>Figura 16</b> Experiencia Consulta de notas .....	123
<b>Figura 17</b> Experiencia consulta de mallas .....	124
<b>Figura 18</b> Experiencia consulta trayectoria académica.....	125
<b>Figura 19</b> Califique rapidez del sistema.....	126
<b>Figura 20</b> Calificación Calidad de uso .....	127
<b>Figura 21</b> Conocimiento Manuales de usuarios.....	127
<b>Figura 22</b> Calidad de herramientas de ayuda.....	128

<b>Figura 23</b> Conocimiento de manejo de aplicación .....	129
<b>Figura 24</b> Claridad en los mensajes .....	130
<b>Figura 25</b> Disponibilidad de la aplicación.....	131
<b>Figura 26</b> Cuenta con usuario y contraseña .....	131
<b>Figura 27</b> Funciones de personas con discapacidad.....	132
<b>Figura 28</b> Personalización interfaz gráfica.....	133
<b>Figura 29</b> Uso en tablets, smartphones o similares .....	133
<b>Figura 30</b> Calificación en otros dispositivos.....	134
<b>Figura 31</b> Calificación General .....	135



## RESUMEN

A partir de los avances tecnológicos y la expansión de los sistemas informáticos, que han llegado a abarcar todos los sectores productivos: financieros, gubernamentales, comerciales, industriales, etc. El software se ha constituido en la base para la competitividad y desarrollo de las diferentes industrias y economías a nivel mundial. Los errores dentro del software aquejan a todos los sectores, actualmente se desarrolla software de calidad a un costo razonable. Los profesionales experimentados y las empresas bien organizadas son prudentes y saben que deben aplicar distintas técnicas de control y prevención, además de un buen proceso de desarrollo. El desarrollo de este trabajo definió una metodología para la evaluación de la calidad del software, basado en la Norma ISO/IEC 25040 y la metodología Bootstrap, en donde se definen características, sub-características, métricas, procesos, formatos, formas de ponderar, entre otros elementos que servirán como herramienta para aplicar una evaluación que salvaguarda la calidad en todas sus etapas, adicional a esto una vez definidos todo los elementos, se procedió a validar la efectividad de la metodología en el Sistema de Gestión Académica de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López que luego de haber aplicado el proceso respectivo alcanzó un puntaje de 78,78 ubicándose en el Nivel 3 Satisfactorio de la escala establecida en la metodología.

### **PALABRAS CLAVE:**

- **CALIDAD DE SOFTWARE**
- **EVALUACIÓN DE CALIDAD**
- **METODOLOGÍA DE CALIDAD**

## **ABSTRACT**

From the technological advances and the expansion of the computer systems that have reached and covered all the productive sectors: financial, governmental, commercial, industrial, etc. The software has become the mainstay for the competitiveness and development of different industries and economies worldwide. Software failures affect all sectors and all countries. Currently, it is feasible to develop reliable software at a reasonable cost. Professionals and well-organized companies know that control and prevention techniques must be applied, as well as a good development process. The development of this work defines a methodology for the evaluation of software quality, which is based on the ISO / IEC 25040 and the Bootstrap methodology, where the characteristics, sub-characteristics, metrics, processes, formats, measuring forms, among others, are defined. Elements that serve as a tool for an evaluation that safeguards the quality in all its stages, in addition, once all the elements were defined, the effectiveness of the methodology was validated in the Sistema de Gestión Académica of the Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, which after having applied the respective process it reached a score of 78.78, locating it in the Level 3 "Satisfactory" of the scale established in the methodology.

### **KEYWORDS:**

- **SOFTWARE QUALITY**
- **QUALITY EVALUATION**
- **QUALITY METHODOLOGY**

# CAPÍTULO I

## INTRODUCCIÓN

### **1.1 Antecedentes**

La creciente demanda de productos de software y la constante preocupación por la calidad en la industria del software tiene como objetivo principal el desarrollo sistemático de productos y servicios de mejor calidad y el cumplimiento de las necesidades y expectativas de los clientes.

El software está presente en todos los sectores de la actividad humana. La relación de la tecnología con las empresas es cada vez más cercana y dependiente, debido a que el software permite la automatización y optimización de procesos, transformándose de esta manera en uno de los principales objetivos estratégicos de las instituciones.

El problema fundamental con respecto al software es el incumplimiento de los requisitos solicitados por el cliente final, originado por la falta de participación del usuario, requerimientos parciales, cambios imprevistos o de última hora, entre otros.

Incluso, se tienen proyectos de software que no alcanzan los mínimos de calidad esperada, ya que no se realizan validaciones específicas por la falta de parámetros de medición de la calidad del software.

La problemática se origina por los siguientes factores:

- Diversidad de criterios para valoración de escala para la medición de un producto de software.
- Diferentes criterios de medición.
- No se establecen normas de medición estandarizadas.
- El usuario final que requiere utilizar el software no se interesa por sus componentes internos.

En un intento de manejar la evaluación de la calidad del software, se ha planteado la ISO/IEC 25000, conocida como SQuaRE (System and Software Quality Requirements and Evaluation), que es un conjunto de normas, cuyo objetivo es la creación de un marco de trabajo común para evaluar la calidad del producto software.

La ISO/IEC 25000 es el resultado de la evolución de otras normas anteriores, principalmente de las normas ISO/IEC 9126 (que describe las particularidades de un modelo de calidad del producto software) e ISO/IEC 14598, que abordaba el proceso de evaluación de productos software. Este conjunto de normas ISO/IEC 25000 se encuentra compuesta por cinco divisiones.

Por otro lado, y como complemento se tiene la metodología Bootstrap que busca extender y adaptar el modelo de madurez del Instituto de Ingeniería de Software (SEI), para hacerlo aplicable a un amplio espectro de casos prácticos. El modelo Bootstrap utiliza los niveles de madurez del SEI, pero además incorpora:

1. Guías de calidad para ayudar a la mejora de procesos de las compañías.
2. Una distinción importante entre organización, metodología y tecnología.

3. Un modelo de proceso base (basado en el modelo utilizado por la Agencia Especial Europea).

La metodología Bootstrap desarrolla un modelo de capacidad integrado (CMM). Este sustituye al software y a los sistemas de ingeniería basada en CMM e integra a otros modelos de ingeniería. Tiene como objetivos los siguientes:

- Proporcionar apoyo a la evaluación de capacidad de proceso, incluyendo las mejores prácticas de otros procesos.
- Apoyar la evaluación como el estándar de referencia.
- Asegurar fiabilidad de la evaluación.
- Identificar, en la organización, procesos fuertes y débiles.
- Apoyar la planificación de mejora con resultados convenientes y confiables.
- Apoyar el logro de los objetivos de la organización planeando acciones de mejora.
- Ayudar a aumentar la eficiencia de los procesos poniendo en práctica la exigencia de estándares en la organización.

## ***1.2 Planteamiento Del Problema De Investigación***

Hoy en día, existe una total interdependencia entre las empresas y el software (Estayno, 2009). La automatización de las actividades, la generación y disponibilidad de información para la toma de decisiones, logradas a través del software, son claves para la consecución de los objetivos y supervivencia de las organizaciones. En virtud de esto, para que una empresa se mantenga competitiva necesita contar con software de calidad,

convirtiéndose esto en un punto diferenciador fundamental, que genera mayor satisfacción del cliente, disminuye costos y optimiza recursos, por lo que se han ido estableciendo normas, marcos de referencia o metodologías enfocadas hacia todo el proceso de desarrollo del software con el propósito de garantizar la calidad del producto final, sin embargo recientemente se han enfocado los esfuerzos en evaluar el producto final de una manera objetiva y que permita eficazmente verificar la calidad del producto final.

No obstante, evaluar la calidad del software aplicando normas o metodologías no es una tarea fácil, debido a que el software es un producto complejo que involucra diversas variables y se ve afectado por muchas situaciones. Existen empresas que en su afán de minimizar recursos (costo, tiempo, entre otros) omiten dicha revisión y esto repercute una vez que se ha implementado el sistema informático, ya que los costos de operación de programas que no se probaron o que no cumplen en su totalidad con los procesos del negocio inflan el presupuesto destinado al proyecto, agregándole a esto la percepción negativa por parte del cliente.

La necesidad de disponer de una metodología que proporcione una guía para llevar a cabo la evaluación del producto software se vuelve indispensable en este sentido, ya que existen varios instrumentos, tales como normas ISO (nombradas anteriormente), o metodologías más generales que abarcan el control de calidad desde el punto de vista de desarrollo de Software, sin embargo, para la evaluación de software no se cuentan con herramientas concisas. En consideración a esto, el proyecto

contempla la elaboración de una metodología de evaluación de software que combine de una manera integral y lógica la norma ISO/IEC 25040 y la metodología Bootstrap.

Con el objetivo de definir claramente el problema, se ha usado la Matriz de Marco Lógico (MML), proceso que está descrito a continuación:

### 1.2.1 Análisis de los involucrados

Se ha realizado un análisis de los grupos de actores que son tocados

**Tabla 1**

*Análisis de los involucrados*

GRUPOS	INTERESES	PROBLEMAS PERCIBIDOS	MANDATOS RECURSOS
<b>Usuarios del Software</b>	<ul style="list-style-type: none"> <li>Realizar su trabajo por medio de herramientas automatizadas</li> <li>No repetir su trabajo por fallas de software.</li> <li>Realizar sus tareas de manera rápida.</li> </ul>	<ul style="list-style-type: none"> <li>Un software de mala calidad puede generar trabajo doble.</li> <li>Un software de mala calidad provoca que las tareas tomen más tiempo.</li> <li>Un software de mala calidad genera descontento y desconfianza en la empresa.</li> </ul>	<ul style="list-style-type: none"> <li>Tiempo</li> <li>Esfuerzo</li> </ul>
<b>Empresas que compran software</b>	<ul style="list-style-type: none"> <li>Mejorar su productividad.</li> <li>Automatizar procesos</li> <li>Aumentar ingresos</li> </ul>	<ul style="list-style-type: none"> <li>Dificultad para identificar problemas con la calidad del software.</li> <li>Falta de procesos de verificación de la calidad de Software.</li> </ul>	<ul style="list-style-type: none"> <li>Dinero</li> <li>Políticas</li> </ul>
<b>Proveedores de Software</b>	<ul style="list-style-type: none"> <li>Generar ingresos a partir de la venta de sistemas.</li> <li>Mejorar su posición de mercado frente a competidores.</li> <li>Mejorar la competitividad de sus clientes.</li> <li>Identificar los errores del software antes que este sea entregado al cliente.</li> </ul>	<ul style="list-style-type: none"> <li>Dificultad para identificar problemas con la calidad del software.</li> <li>Software no cumple con expectativas de usuarios finales.</li> </ul>	<ul style="list-style-type: none"> <li>Reputación</li> <li>Dinero</li> </ul>

Tabla 2

## Análisis de usuarios

GRUPOS	ESTRATEGIA
Usuarios del Software	<ul style="list-style-type: none"> <li>Definición de usuarios expertos.</li> <li>Generación de documentación técnica con apoyo del departamento de tecnologías de la información.</li> </ul>
Empresas que compran software	<ul style="list-style-type: none"> <li>Apoyarse en un proceso formal que permita evaluar correctamente un software.</li> </ul>
Proveedores de Software	<ul style="list-style-type: none"> <li>Mejorar los procesos de control de calidad en la producción del software.</li> </ul>

## 1.2.2 Árbol de problemas



Figura 1 Árbol de problemas

## 1.2.3 Árbol de objetivos

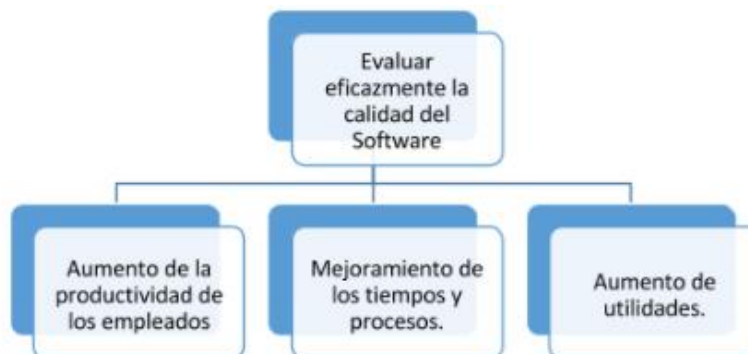


Figura 2 Árbol de objetivos



#### **1.2.4 Análisis de alternativas**

Se ha realizado un análisis de las actuales herramientas que están a disposición identificando las siguientes:

- Norma ISO 25040: Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE) -- Evaluation process.
- La metodología Bootstrap (Kuvaja et al., 1994) que es el resultado de un proyecto europeo basado en los modelos CMM (Paulk, Curtis, Chrissis, & Weber, 1993) e ISO 9000 (Guler, Guillen, & Macpherson, 2002).

Sin embargo, si bien la norma ISO 25040 contiene requerimientos y recomendaciones para la evaluación de la calidad del software, y clarifica los conceptos generales, ésta no ofrece un paso a paso, o un grupo de mecanismos y procedimientos que permita evaluar la calidad de un producto Software. Por otro lado, la metodología Bootstrap se enfoca en la evaluación para establecer el diagnóstico de un proceso para desarrollo de software (el cual incluye a la planeación, los métodos y la capacidad de ingeniería, las herramientas y la tecnología), y la creación de un plan de acción que defina los pasos, los detalles de la implantación y los marcos temporales para que la organización aumente su capacidad de entrega de productos y servicios de calidad, su orientación es la evaluación calidad.

### 1.2.5 Estructura analítica del proyecto

**Tabla 3**

*Árbol de problemas*

<p><b>Fin</b></p> <ul style="list-style-type: none"> <li>• Diseñar una metodología de evaluación de la calidad del software basada en la norma ISO/IEC 25040 y metodología Bootstrap para determinar la calidad del Sistema de Gestión Académica de la ESPAM MFL, a través de una investigación cualitativa basada en teoría fundamentada.</li> </ul>
<p><b>Propósito</b></p> <ul style="list-style-type: none"> <li>• Aumento de la productividad de los empleados</li> <li>• Mejoramiento de los tiempos y procesos.</li> <li>• Aumento de utilidades.</li> </ul>
<p><b>Componentes</b></p> <ul style="list-style-type: none"> <li>• Modelo de Calidad             <ul style="list-style-type: none"> <li>• Modelo de calidad externa e interna</li> <li>• Calidad de uso</li> </ul> </li> <li>• <b>Modelo de Calidad Externa e Interna.</b> <ul style="list-style-type: none"> <li>• Funcionalidad</li> <li>• Confiabilidad</li> <li>• Usabilidad</li> <li>• Eficiencia</li> <li>• Capacidad de mantenimiento</li> <li>• Portabilidad</li> <li>• Calidad en uso</li> </ul> </li> </ul>
<p><b>Actividades</b></p> <ul style="list-style-type: none"> <li>• <b>Fase 1: Análisis de modelos y estándares para la evaluación de la calidad de Software.</b> <ul style="list-style-type: none"> <li>• Revisión del modelo ISO 25040</li> <li>• Revisión de la metodología Bootstrap</li> <li>• Documentación de literatura base</li> </ul> </li> <li>• <b>Fase 2: Definición de la metodología.</b> <ul style="list-style-type: none"> <li>• Planeamiento de la metodología</li> <li>• Determinación de los componentes de la metodología.</li> <li>• Desarrollo de la metodología.</li> <li>• Documentación de la metodología</li> </ul> </li> <li>• <b>Fase 3: Validación de la metodología por medio de un caso de estudio.</b> <ul style="list-style-type: none"> <li>• Desarrollo del caso de estudio "Sistema de Gestión Académica de la ESPAM MFL".</li> </ul> </li> </ul>

### 1.3 Formulación del Problema

- ¿Qué aspectos debe considerar una metodología de la evaluación de la calidad del software basada en la norma ISO/IEC 25040 y metodología Bootstrap?

- ¿La aplicación de la metodología planteada servirá para evaluar la calidad del Sistema de Gestión Académica de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López?
- ¿Qué nivel de Calidad de Software alcanza el Sistema de Gestión Académica de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, desde la perspectiva de la metodología propuesta?

#### **1.4 Justificación e Importancia**

Se debe considerar la calidad como un factor importante del producto software que debe ser evaluado, para ello es necesario contar con una metodología que ayude a realizar esta revisión y control de calidad. El desarrollo de la metodología basada en normas universalmente aceptadas, como son la ISO/IEC 25040 y la metodología Bootstrap será abordado como parte de este trabajo de investigación, además permitirá diagnosticar de forma ordenada aquello que las empresas desean resolver y mejorar.

Se presentará un caso de estudio con la finalidad de medir la efectividad de la metodología, también servirá como guía de trabajo en el momento que alguna persona o institución decida aplicar la evaluación, permitiendo un desarrollo en el conocimiento teórico y práctico en este tipo de evaluaciones. Además, servirá como aporte a la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, que tendrá un informe de evaluación de la calidad del Sistema de Gestión Académica considerado el software principal en esta institución, en el caso de presentarse alguna debilidad en torno a la calidad, los encargados del Unidad de Producción de Software podrán tomar las medidas necesarias para corregirlas.

## **1.5 Objetivos**

### **1.5.1 Objetivo General**

Diseñar una metodología de evaluación de la calidad del software basada en la norma ISO/IEC 25040 y metodología Bootstrap para determinar la calidad del Sistema de Gestión Académica de la ESPAM MFL, a través de una investigación cualitativa basada en teoría fundamentada.

### **1.5.2 Objetivos Específicos**

- Definir una metodología de evaluación de la calidad del software basada en la norma ISO/IEC 25040 y metodología Bootstrap.
- Aplicar la metodología propuesta en el Sistema de Gestión Académica de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López
- Presentar un informe de resultados de la aplicación de la metodología en el Sistema de Gestión Académica de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López MFL.

## **1.6 Alcance**

El resultado de esta investigación será la propuesta de una metodología para la evaluación de la calidad del producto del software, la misma que tendrá como base la norma de calidad ISO/IEC 25040 y el método Bootstrap y luego de esto se medirá su aplicabilidad en el Sistema de Gestión Académica de la ESPAM MFL al final se realizará un informe de la ejecución de la metodología.

## CAPÍTULO II

### MARCO TEÓRICO

#### **2.1 Bases teóricas de la Calidad de Software**

##### **2.1.1 Software**

(Pressman, 2010) define al software como “un conjunto de instrucciones (programas de computadoras) que cuando se ejecutan proporcionan la función y el rendimiento deseado. El software se puede desarrollar para un cliente en particular o para un mercado en general.

Otra definición acerca del software la da (Aparicio, 2012), en donde presenta algunas características para realizar una definición más acertada. El autor manifiesta que el software se ha convertido en el elemento clave de la evolución de los sistemas y productos informáticos, y por tal razón no se puede tomar como sólo el conjunto de programas, instrucciones y estructuras de datos. A continuación, se presentan las características del Software:

- Se desarrolla, no se fabrica: se utiliza un modelo de proceso de desarrollo que comprende análisis, diseño, desarrollo, implementación y evaluación para obtener un producto de calidad.
- No se “estropea”, pero se deteriora: El software durante su vida sufre cambios por lo que es probable que surjan fallos y defectos que si no se corrigen permiten que el software se vaya deteriorando.

- Se construye a medida: a medida que el software evoluciona se crean estándares de diseño. El software debe diseñarse e implementarse para que pueda ser reutilizable.

### **2.1.2 Calidad**

La calidad se debe mirar desde distintos puntos de vistas que se van a detallar, por ejemplo, un concepto general de la calidad lo define el diccionario de la Real Academia Española y manifiesta que es “la propiedad o conjunto de propiedades inherentes a una cosa que permite apreciarla como igual, mejor o peor que las restantes de su misma especie”. Esta definición muestra las dos características esenciales del término. Por una parte, la subjetividad de su valoración, y por otra su relatividad. No es una cualidad absoluta que se posee o no se posee, sino un atributo relativo: se tiene más o menos calidad. (Diplomado a Distancia "Informática Médica")

Centrándose en el producto, el término calidad se entiende como un concepto relativo que se genera a partir de la relación del binomio producto/cliente. Reúne un conjunto de cualidades relacionadas entre sí que todos los bienes y servicios poseen en mayor o menor medida. (Diplomado a Distancia "Informática Médica")

La calidad constituye un modo de ser del bien o servicio, en consecuencia, es subjetivo y distinto según el punto de vista de quien la ofrece y de quien la consume. En la literatura especializada hay varias definiciones, todas ellas se han formado en función tanto de las características del bien o servicio como de la satisfacción de las necesidades y exigencias del consumidor.

La calidad es (Diplomado a Distancia "Informática Médica"):

- El grado de satisfacción que ofrecen las características del producto con relación a las exigencias del consumidor.
- El conjunto de especificaciones y características de un producto o servicio referidas a su capacidad de satisfacer las necesidades que se conocen o presuponen (ISO 9004-2).
- El conjunto de propiedades y características de un producto o servicio que le confieren la aptitud para satisfacer unas necesidades manifiestas o implícitas (UNE 66.901).
- El conjunto de todas las propiedades y características de un producto que son apropiadas para cumplir las exigencias del mercado al que va destinado.
- El cumplimiento de las especificaciones, los requisitos del diseño del producto o servicio.
- El grado de adecuación de un determinado producto o servicio a las expectativas del usuario o a ciertos parámetros tecnológicos o científicos expresados mediante normas concretas.

### **2.1.2.1 Calidad del Software**

La conferencia organizada por el Grupo GIDIS (1999), define a la Calidad del Software como la concordancia de los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo, explícitamente documentados, y con las características implícitas que se espera de todo software

desarrollado profesionalmente. En dicha conferencia se concluye lo siguiente con respecto a la Calidad del Software (Norberto & Gloria, 2011):

- Los requisitos del software son la base de las medidas de calidad. La falta de concordancia con los requisitos es una falta de calidad
- Los estándares o metodologías definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software. Si no se sigue ninguna metodología siempre habrá falta de calidad.
- Existen algunos requisitos implícitos o expectativas que a menudo no se mencionan, o se mencionan de forma incompleta (por ejemplo, el deseo de un buen mantenimiento), que también pueden implicar una falta de calidad.

Además, (Ayala, 2010) en su tesis titulada “Calidad del Software”, manifiesta que algunas características de calidad con las que debe contar cualquier producto de software para poder cumplir los requerimientos de usuario, así como tiempos de entrega y confiabilidad, son:

- Corrección: Que se refiere a si el producto de software hace lo que se quiere.
- Fiabilidad: Si el producto trabaja de forma fiable todo el tiempo.
- Eficiencia: Si el software se ejecutará en el hardware lo mejor que pueda.
- Seguridad (integridad): Se hace la pregunta de si el software es seguro.
- Facilidad de uso: Que tanto está diseñado para ser usado.
- Facilidad de Mantenimiento: Si después de entregado, se puede corregir.
- Flexibilidad: En algún momento se puede llegar a cambiar.



- Facilidad de prueba: Que tan fácil es probarlo.
- Portabilidad: Se refiere a si se puede usar en diferentes máquinas o dispositivos.
- Reusabilidad: Si se podrá utilizar alguna parte del software para otros proyectos.

Actualmente existe un gran interés por la calidad de los productos o servicios. En el mercado actual, que es tan competitivo, no basta con producir y distribuir los productos o servicios, vender es lo importante y esto se genera con la aceptación por parte del cliente, se dice que la calidad no tiene un concepto solo se reconoce. Sin embargo, la calidad en el software es un concepto complejo que no es directamente comparable con la calidad de un producto. El software se ha convertido en la actualidad en uno de los principales objetivos estratégicos de las organizaciones debido a que, cada día, los procesos más importantes de las organizaciones y su supervivencia dependen del funcionamiento del software (Cendejas, 2017).

Según (Pressman, 2010), es la concordancia del software producido con los requerimientos explícitamente establecidos, y con los estándares de desarrollo prefijados, y con los requerimientos implícitos no establecidos formalmente que desea el usuario. Otra definición que contempla (Vega, Rivera, & García, 2008) en su libro, y que es propuesta por la organización internacional de estándares (ISO/IEC DEC 9126): “La totalidad de características de un producto de software que tienen como habilidad satisfacer necesidades explícitas o implícitas”.

La calidad del software se puede observar en una característica o atributo. Como un atributo, la calidad se refiere a características medurables, es decir cosas que se pueden comparar para conocer estándares, como longitud, color, propiedades eléctricas y maleabilidad. Sin embargo, el software que es una entidad intelectual, tiene la complejidad de caracterizar los objetos físicos. No obstante, existen mediciones que nos permiten evaluar las características de un programa. Dichas propiedades incluyen complejidad psicósomática, número de puntos de función, líneas de código, entre otras.

Los principales atributos de la calidad del software se describen a continuación:

- ***CALIDAD INTERNA***

Cuando se desarrolla un software, se desarrolla con una calidad medida según sus requisitos, que pueden determinar su alto grado de calidad, midiendo una serie de parámetros. Pero estos parámetros no reflejan ni tienen en cuenta la calidad del proceso de creación del software en sí, ni la calidad del software escrito, solo la calidad externa, es decir, la función que realiza el software (Rubio, 2010).

La calidad interna, no obstante, mide y tiene presente la manera en la que se ha desarrollado el código, de tal forma que pueda mantenerse (corregirse, ampliarse y adaptarse) de forma rápida y sencilla, gracias a un diseño e implementación limpia, simple y clara

Para un desarrollador, la calidad interna debe ser igual o más importante que la calidad externa, puesto que, si un software hace lo que debe, pero una de cada 1000 veces no funciona, puede deberse a un fallo interno difícil de detectar que, pueda incluso

verse agravado por la mala calidad con la que se ha desarrollado (diseñado o implementado) el sistema en sí.

Con respecto al tiempo, un software desarrollado con una calidad baja o nula, puede ser desarrollado en un tiempo muy pequeño, ya que no se tiene en consideración muchos aspectos necesarios y útiles, como, por ejemplo, una buena orientación a objetos, modularización del código, reutilización, algoritmos excesivamente complejos u ofuscados, etc.

Cuando, por un fallo (mantenimiento correctivo) hay que volver al código para corregir un defecto, y se detecta un error de diseño, modificar un código mal hecho es más complicado que corregir un defecto en un código que está mejor desarrollado en base a patrones y reglas básicas de análisis, diseño o codificación (Rubio, 2010).

En otras palabras, un programa con calidad interna baja es un programa que generará deuda técnica.

Características y sub-características de calidad interna (Valdez, 2013):

- Adecuación funcional: funcionalidad adecuada, funcionalidad correcta, funcionalidad completa.
- Confiabilidad: madurez, disponibilidad, tolerancia a fallos, recuperabilidad.
- Eficiencia de rendimiento: tiempo de respuesta, utilización de recursos, capacidad.

- Operabilidad: reconocimiento de funcionalidad adecuada, facilidad de uso, facilidad de aprendizaje, protección contra errores de usuario, accesibilidad, estética de la interfaz de usuario.
  - Seguridad: confidencialidad, integridad, no rechazo, responsabilidad, autenticidad.
  - Compatibilidad: interoperabilidad, capacidad de coexistencia.
  - Mantenibilidad: modularidad, reusabilidad, capacidad de ser analizado, capacidad de ser modificado, capacidad de ser verificado/probado.
  - Transmisibilidad/Portabilidad: instalabilidad, adaptabilidad, reemplazabilidad.
- **CALIDAD EXTERNA**

Según (Valdez, 2013) la calidad externa es la capacidad de un producto software para desarrollar el comportamiento de un sistema, de forma que satisfaga las necesidades declaradas, e implícitas de un sistema utilizado, bajo ciertas condiciones especificadas.

Características y sub-características de calidad externa:

- Satisfacción de uso: utilidad, confianza, placer, comodidad.
- Seguridad de uso: mitigación de riesgos económicos, mitigación de riesgos para el usuario, mitigación de riesgos ambientales.
- Flexibilidad de uso: cobertura del contexto, flexibilidad.

- Efectividad de uso.
- Eficiencia de uso.
- **CALIDAD EN USO**

La calidad en uso ha sido motivo de investigación recientemente, y es uno de los aspectos incorporados al estándar ISO/IEC 9126 2001. Este modelo de calidad es definido como “la opinión que tiene un usuario al utilizar una aplicación de software”, la cual se deriva de los resultados obtenidos al evaluar dicha aplicación (Moreno, Imbert, Nakasone, & Pow-Sang, 2008).

La definición de la calidad en uso de acuerdo al ISO/IEC-9126-1 es “la capacidad de un producto de software de facilitar a usuarios específicos, alcanzar metas específicas con efectividad, productividad seguridad y satisfacción en un contexto de uso específico”.

La definición de estas características se comenta a continuación:

- **Efectividad:** “Es la capacidad del software para permitir a los usuarios obtener metas específicas con precisión, y completamente especificadas en el contexto del software”.
- **Productividad:** “Es la característica del software para ser utilizado a su máxima capacidad, en relación a la efectividad alcanzada en un contexto específico de uso”.
- **Seguridad:** “Es la capacidad del producto de software para alcanzar niveles aceptables de riesgo de dañar a las personas, el negocio o el software”.

- **Satisfacción:** “Es la capacidad del software para satisfacer a los usuarios en un contexto de uso” (Moreno, Imbert, Nakasone, & Pow-Sang, 2008).

- ***CALIDAD TOTAL***

Es el conjunto de principios de estrategia global, que intenta movilizar a toda la empresa con el fin de obtener una mejor satisfacción del cliente, al menor coste. La calidad total del software es un trabajo de todos, y se puede conseguir mediante un adecuado análisis, diseño, codificación y prueba, también mediante la aplicación de revisiones técnicas formales, de una estrategia de prueba multinivel, un mejor control de la documentación del software y sus modificaciones, y la aplicación de estándares de desarrollo de software.

### ***2.1.3 Evaluación de la Calidad***

En términos generales, se entiende por evaluación el proceso mediante el cual se intenta obtener un juicio de valor o una apreciación de la bondad de un objeto, de una actividad, de un proceso o de sus resultados. Este proceso pone en relieve las cualidades, ventajas y debilidades de aquello que se evalúa. La evaluación permite obtener información fiable para la toma de decisiones en las tareas relacionadas con el desarrollo, mantenimiento y gestión de servicios de información. La evaluación es una herramienta de toma de decisiones (Castillo, 2013).

La evaluación es el proceso de identificar y reunir datos acerca de servicios o actividades específicas, estableciendo criterios para valorar su éxito y determinando el grado hasta donde el servicio o actividad cumple sus fines y objetivos establecidos. El

proceso de evaluación requiere como punto de partida plantearse cuales son los fines reales que se persiguen. Para evaluar es necesario disponer de un referente con el que comparar. La evaluación se suele basar en la toma de datos sobre los resultados obtenidos, que permitan llegar a conclusiones que redunden en la mejora de la organización (Castillo, 2013).

### **2.1.3.1 Evaluación de la Calidad del Software**

Según (Juran, 1992), la calidad para poder ser entendida de una mejor manera, y posteriormente ser medida con eficacia, debe ser expresada por medio de otros términos, que tengan más sentido para el usuario. En el caso del software. Estos factores son el medio por el cual se traduce el término “calidad” al lenguaje de las personas que manejan la tecnología.

Los factores de calidad que afectan a la calidad del software se dividen en dos grandes grupos:

- Los que miden directamente (defectos descubiertos en las pruebas).
- Los que se miden indirectamente (facilidad de uso o de mantenimiento).

En cada caso debe presentarse una medición, se debe comparar el software con algún conjunto de datos, y obtener así algún indicio sobre la calidad. (McCall, Richards, & Walters, 1977) propusieron una clasificación de los factores que afectan directamente a la calidad del software.

- Características operativas.
- Capacidad para experimentar cambios.

- Capacidad para adaptarse a nuevos entornos.

A continuación, se describen los factores que propone (McCall, Richards, & Walters, 1977) :

- **Corrección:** El grado en que el programa cumple con su especificación y satisfacer los objetivos que propuso el cliente.
- **Confiabilidad:** El grado en que se esperaría que un programa desempeña su función con la precisión requerida.
- **Eficiencia:** La cantidad de código y de recursos de cómputo necesarios para que un programa realice su función.
- **Integridad:** El grado de control sobre el acceso al software o los datos por parte de las personas no autorizadas.
- **Facilidad de uso:** El esfuerzo necesario para aprender, operar y preparar los datos de entrada de un programa interpretan la salida.
- **Facilidad de mantenimiento:** El esfuerzo necesario para localizar y corregir un error en un programa.
- **Flexibilidad:** El esfuerzo que demanda probar un programa con el fin de asegurar que realiza su función.
- **Portabilidad:** El esfuerzo necesario para transferir el programa de un entorno de hardware o software a otro.
- **Facilidad de reutilización:** El grado en que un programa o partes de él pueden reutilizarse en otras aplicaciones (en relación con el empaquetamiento y el alcance de las funciones que realiza el programa).



- **Interoperabilidad:** El esfuerzo necesario para acoplar un sistema con otro.

### **2.1.3.2 Medidas, métricas e indicadores**

La medición asigna números o símbolos a atributos de entidades reales (Fernández, 2017), esto requiere un modelo de medición que abarque un conjunto existente de reglas. En el contexto de la ingeniería del software una medida proporciona una indicación cuantitativa de la extensión, la cantidad, la dimensión, la capacidad o el tamaño de algún atributo de un producto o proceso. La medición ocurre como resultado de la recopilación de uno o más puntos de datos. Una métrica de software relaciona de alguna manera las medidas individuales, de igual manera un ingeniero de software recopila medidas, y desarrolla métricas para obtener los indicadores (Pallares, 2012).

Un indicador es una métrica, o una combinación de métricas, que proporcionan conocimientos acerca del proceso del desarrollo de software, un proyecto de software o el propio producto. Un indicador proporciona conocimientos que permiten a los ingenieros de software ajustar el proceso, el proyecto, o el producto para que las cosas mejoren. Existe la necesidad de medir y controlar la complejidad en el desarrollo del software, debe de tenerse la posibilidad de desarrollar medidas de diferentes atributos internos del programa. Estas medidas y las métricas derivadas de ellas se utilizan como indicadores independientes de la calidad de los modelos de análisis y diseño.

Antes de generar e introducir una serie de métricas del producto se necesita:

- Ayudar a evaluar los modelos de análisis y diseño.

- Ofrecer una indicación de la complejidad de los diseños procedimentales y el código fuente.
- Facilitar el diseño de pruebas más efectivas.

Es importante comprender los principios básicos de la medición, (Roche, 1994) sugiere un proceso de medición en el que se caracterizan cinco actividades primordiales las cuales son:

1. **Formulación:** La derivación de medidas y métricas apropiadas para la representación del software que se considera.
2. **Recolección:** El mecanismo con que se acumulan los datos necesarios para derivar las métricas formuladas.
3. **Análisis:** El cálculo de las métricas y la aplicación de herramientas matemáticas.
4. **Interpretación:** La evaluación de las métricas en un esfuerzo por conocer mejor la calidad de la representación.
5. **Retroalimentación:** Recomendaciones derivadas de la interpretación de las métricas del producto transmitidas al equipo del software.

Las métricas del software sólo serán útiles si son caracterizadas de manera efectiva, y se validan para probar su valor. Según (Lethbridge, 2003), los siguientes principios son representativos de muchos otros que podrían proponerse para caracterizar y validar las métricas. Una métrica debe tener propiedades matemáticas deseables, es decir que el valor de la métrica debe estar en un rango significativo, por ejemplo, de cero a uno, donde cero realmente significa ausencia, uno indica el valor

máximo y 0.5 representa el punto medio. Además, una métrica pretende estar en una escala racional, no debe contar con componentes que sólo se miden en una escala ordinal. Cuando una métrica representa una característica de software que aumenta cuando se presentan rasgos positivos o que disminuya al encontrar rasgos indeseables, el valor de la métrica debe aumentar o disminuir en el mismo sentido.

Cada métrica debe validarse empíricamente en una amplia variedad de contextos antes de publicarse, o aplicarse la toma de decisiones. Una métrica debe medir el factor de interés, independientemente de otros factores, debe crecer para aplicarse a sistemas grandes, funcionar en diversos lenguajes de programación, y dominios de sistemas. Aunque la formulación, caracterización, y validación son críticas, la recopilación y el análisis son las actividades que dirigen el proceso de medición. (Roche, 1994) sugiere las siguientes directrices para estas actividades:

1. Siempre que sea posible deben automatizarse la recopilación de datos y su análisis.
2. Deben aplicarse técnicas estadísticas válidas para establecer relaciones entre los atributos internos del producto y las características externas de la calidad.
3. Para cada métrica deben establecerse directrices y recomendaciones para la interpretación.

Se han propuesto una gran cantidad de métricas para el desarrollo y evaluación del software, pero no todas proporcionan un soporte práctico para el ingeniero de software. Algunas exigen mediciones demasiado complejas otras son demasiado

especializadas que pocos profesionales podrían comprenderlas, y otras violan las nociones básicas de lo que es el software de alta calidad. (Ejioqu, 1991), define un conjunto de atributos que toda métrica efectiva del software debe abarcar. La métrica derivada y las medidas que llevan a ella deben ser:

- **Simples incalculables:** Debe ser relativamente fácil aprender a derivar la métrica, y su cálculo no debe exigir cantidades anormales de tiempo o esfuerzo.
- **Empírica e intuitivamente persuasivas:** La métrica debe satisfacer las nociones intuitivas del ingeniero acerca del atributo del producto que se está construyendo.
- **Consistentes y objetivas:** La métrica siempre debe arrojar resultados que no permitan ambigüedad alguna.
- **Consistentes en el uso de unidades y dimensiones:** El cálculo matemático de la métrica debe emplear medidas que no lleven a combinaciones extrañas de unidades.
- **Independientes del lenguaje de programación:** Las métricas deben basarse en el modelo de análisis o diseño o en la estructura del propio programa.
- **Mecanismos efectivos para la retroalimentación de alta calidad:** Es decir, la métrica debe llevar a un producto final de la más alta calidad.

Aunque casi todas las métricas de software satisfacen esos atributos, algunas métricas de uso común no cumplen con una o dos de ellas. Aunque se ha propuesto una

amplia variedad de taxonomía en métricas, el siguiente esquema atiende a las cuatro más importantes en el desarrollo del software.

- **Métricas para el modelo de análisis.** Estas métricas atienden varios aspectos de la etapa de análisis en donde se incluyen:
  - **Funcionalidad entregada:** proporciona una medida indirecta de la funcionalidad que se empaqueta con el software.
  - **Tamaño del sistema:** mide el tamaño general del sistema, definido desde el punto de vista de la información disponible como parte del modelo de análisis.
  - **Calidad de la especificación:** Proporciona un indicador específico o el grado en que se ha completado la especificación de los requisitos.
- **Métricas para el modelo de diseño.** Estas métricas cuantifican los atributos del diseño, de manera tal que le permiten al ingeniero de software evaluar la calidad del diseño, la métrica incluye:
  - Métricas arquitectónicas. Proporcionan un indicio de la calidad del diseño arquitectónico.
  - **Métricas al nivel de componente:** mide la complejidad de los componentes del software y otras características que impactan la calidad.
  - **Métricas de diseño de la interfaz:** se concentran principalmente en la facilidad de uso.

- **Métricas especializadas en diseño orientado a objetos:** Miden características de clases, además de las correspondientes a comunicación y colaboración.
- **Métricas para el código fuente.** Estas métricas miden el código fuente y se usan para evaluar su complejidad, además de la facilidad con que se mantiene y prueba entre otras características como:
  - **Métricas de complejidad:** miden la complejidad lógica del código fuente.
  - **Métricas de longitud:** proporcionan un indicio del tamaño del software.
- **Métricas para pruebas.** Estas métricas ayudan a diseñar casos de prueba efectivos, y evaluar la eficacia de las pruebas, en donde se incluyen:
  - **Métricas de cobertura de instrucciones y ramas:** lleva al diseño de casos de prueba que proporcionan cobertura del programa.
  - **Métricas relacionadas con los defectos:** se concentran en encontrar defectos, y no en las propias pruebas.
  - **Efectividad de la prueba:** proporciona un indicio en tiempo real de la efectividad, y de las pruebas aplicadas.
  - **Métricas en el proceso.** métricas relacionadas con el proceso de las pruebas.

En muchos casos las métricas de un modelo pueden aplicarse en actividades posteriores de la ingeniería del software. Por ejemplo, las métricas de diseño se utilizan para estimar el esfuerzo requerido para generar código fuente.

## ***2.2 Modelo de Evaluación de Calidad del Software***

Un modelo de calidad es una herramienta muy útil para la ingeniería de requisitos de calidad, así como para la evaluación temprana y el control de la calidad. Se define como un conjunto de características y relaciones entre ellas con las cuales proveer las bases para especificar los requisitos de calidad y evaluar la misma. Distintos modelos de calidad han sido propuestos. En el siguiente apartado se va a realizar un análisis de los diferentes modelos existentes mediante una tabla comparativa.

### 2.2.1 Modelos Existentes (cuadro comparativo)

**Tabla 4**

*Cuadro Comparativo de Modelos de Evaluación del Software*


MODELO	VENTAJAS	DESVENTAJAS	CARACTERÍSTICAS
<b>CMMI</b>	El modelo CMMI cuenta con gran eficacia, ha permitido mejoras de gran impacto en procesos de desarrollo de productos software, tales como reducción del coste de desarrollo, localización y resolución de defectos; mejora en la fiabilidad de la planificación, en términos de dedicación y de calendario.	El gran problema de CMMI es su falta de adecuación al enfoque a servicio que está experimentando el sector de las TI en todas sus líneas de actividad, así como el alto esfuerzo de implantación que exige.	Es un modelo que constituye una forma de medir el grado de madurez de las organizaciones, con el objetivo de establecer una guía que les permita mejorar sus procesos y su habilidad para organizar, desarrollar, adquirir y mantener productos y servicios informáticos.
<b>McCall</b>	Se focaliza en el producto final, identificando atributos claves desde el punto de vista del usuario. Focaliza en medidas precisas de alto nivel.	Las características son en general propiedades abstractas medible mediante métricas. No siempre existe una relación perfectamente lineal entre los valores de las métricas y las características que deben estimar	El modelo de McCall organiza los factores en tres ejes o puntos de vista desde los cuales el usuario puede contemplar la calidad de un producto, basándose en once factores de calidad organizados en torno a los tres ejes y a su vez cada factor se desglosa en otros criterios.
<b>Desarrollo en espiral</b>	Reduce riesgos del proyecto Incorpora objetivos de calidad Integra el desarrollo con el mantenimiento.	Genera mucho tiempo en el desarrollo del sistema Modelo costoso Requiere experiencia en la identificación de riesgos	Las actividades de este modelo se conforman en una espiral, en la que cada bucle o iteración representa un conjunto de actividades. En cada iteración se toma en cuenta los objetivos, alternativas el desarrollo y verificación del software.

Continúa 



MODELO	VENTAJAS	DESVENTAJAS	CARACTERÍSTICAS
<b>MOSCA</b>	Se enfoca tanto al producto como al proceso, garantiza la relación sinérgica entre las características de la empresa y las necesidades del usuario, constituye una herramienta efectiva de análisis y estimación de la Calidad Global Sistémica.	Proceso complicado sino se cuenta con una guía adecuada de aplicación del modelo.	MOSCA es una herramienta que soporta la Administración de la Calidad del Software en sus tres actividades: Aseguramiento de la Calidad, Planeación de la Calidad y Control de la Calidad, al establecer un marco de referencia que permite ubicar en un "nivel establecido" la calidad sistémica de sus productos.
<b>SPICE</b>	Cuenta con las ventajas de ofrecer una base para poder ser evaluada, brinda evaluaciones, con resultados repetibles, reduciendo las diferencias en los resultados, y realizando las evaluaciones en el área de problema.	Se necesita de un gran esfuerzo para implantar las evaluaciones, y resulta más caro, sin percibir alguna estrategia de mejorar el proceso.	Establece un marco y los requisitos para cualquier proceso de evaluación de procesos y proporciona requisitos para los modelos de evaluación de los procesos.
<b>Modelo Deming</b>	Estabilización y mejora de la calidad, mejora de la productividad/reducción de costos, incremento de las ventas, establecimiento de diferentes sistemas de dirección.	Genera mucho tiempo en el desarrollo	La evaluación en este modelo tiene como principal objetivo comprobar que, mediante la implantación del control de calidad en toda la organización, se obtengan buenos resultados. La producción es vista como un sistema, por lo que la mejora de la calidad abarca a toda la línea de producción, desde la recepción de los materiales hasta el consumidor (que se sitúa como la pieza más importante).

Continúa 

MODELO	VENTAJAS	DESVENTAJAS	CARACTERÍSTICAS
<b>MALCOLM BALDRIGE</b>	Aumento de la productividad, mejora en las relaciones interpersonales, aumento de la participación en el mercado.	Toma mucho tiempo y esfuerzo desarrollarlo.	Es una herramienta para la evaluación, mejora y planificación de la gestión de calidad total de una empresa.
<b>Modelo de Excelencia EFQM</b>	<ul style="list-style-type: none"> <li>-Es una forma de establecer puntos fuertes y áreas de mejora de la organización</li> <li>-Ofrece una evaluación sistemática, basada en hecho y no en percepciones subjetivas.</li> <li>-Permite la comparación con los resultados de otras organizaciones de referencia.</li> <li>- Puede emplearse en cualquier tipo de empresa, organización u departamento/unidad.</li> <li>- Favorece la gestión por procesos y permite hacer un diagnóstico sobre el estado y evolución</li> </ul>	Media Facilidad de actualización, Cierta grado de dificultad de comprensión y cierto nivel de complejidad.	Se fundamenta en los principios de la Gestión de la Calidad Total y basa su desarrollo en la autoevaluación de las organizaciones como método de mejora continua
<b>Modelo Shingo Prize</b>	Asegurar la calidad en la fuente, Adoptar el pensamiento científico, centrarse en el proceso, pensar sistémicamente.	Proceso complejo de aplicación del modelo.	El modelo está diseñado para identificar la evolución de una compañía que atraviesa por una transformación y para apoyar a los directivos a detectar en dónde se encuentran sus compañías en su jornada lean y evaluar el nivel de profundidad y entendimiento de esta filosofía dentro de su empresa
<b>Modelo Iberoamerica no de</b>	Asociación gratuita por un año a FUNDIBEQ, publicación de los resultados en medios de prensa internacionales, divulgación de las	Poca capacidad de actualización, baja capacidad de adaptación al sector público,	Su objetivo es la evaluación de la gestión de las organizaciones, identificando sus puntos Continúa 

MODELO	VENTAJAS	DESVENTAJAS	CARACTERÍSTICAS
<b>Excelencia en la Gestión</b>	mejores prácticas de la organización ganadora en toda Iberoamérica.		fuerzas y áreas de mejoras que sirvan para establecer planes de progreso y también sirva como información para el desarrollo y la planificación estratégica.
<b>Modelo de dirección por calidad</b>	Permite elevar la efectividad y eficiencia de las acciones, compartir conocimientos y experiencias, creación de valor para los productos o servicios ofrecidos.	Se necesita de un gran esfuerzo para implantar las evaluaciones	<ul style="list-style-type: none"> <li>- Fomentar la competitividad de las organizaciones mexicanas</li> <li>- Promover una cultura basada en la mejora continua y la creación de valor a clientes y usuarios finales, personal, accionistas, comunidad y entorno,</li> <li>- Promover la comunicación y el intercambio en las organizaciones,</li> <li>- Crear un lenguaje común y generar sinergia en los grupos de trabajo</li> </ul>
<b>ISO 25040</b>	<ul style="list-style-type: none"> <li>- El modelo representa la calidad esperada del producto de software.</li> <li>- Planteo del desdoblamiento de las necesidades o expectativas en calidad en uso, calidad externa y calidad interna.</li> <li>- Permite una mayor eficacia en la definición del software.</li> <li>- Plantea la evaluación de productos intermedios.</li> </ul>	<ul style="list-style-type: none"> <li>- Los esfuerzos y costos para preparar la documentación</li> <li>- El tiempo requerido para la evaluación</li> <li>- El papeleo necesario</li> </ul>	<p>Define el proceso para llevar a cabo la evaluación del producto software. Dicho proceso de evaluación consta de un total de cinco actividades:</p> <ul style="list-style-type: none"> <li>- Establecer los requisitos de evaluación</li> <li>- Especificar la evaluación</li> <li>- Diseñar la evaluación</li> <li>- Ejecutar la evaluación</li> <li>- Concluir la evaluación</li> </ul>
<b>BOOTSTRAP</b>	<ul style="list-style-type: none"> <li>- Proporcionar soporte para la evaluación de la capacidad de los procesos utilizando un conjunto de prácticas de Ingeniería del Software.</li> <li>- Incluir estándares de Ingeniería del Software reconocidos internacionalmente como fuentes</li> </ul>	<ul style="list-style-type: none"> <li>- Se implementa principalmente en Europa</li> <li>- Incompleto en comparación con otros modelos</li> </ul>	<p>La metodología Bootstrap engloba tanto la evaluación para establecer el diagnóstico de un proceso para desarrollo de software (el cual incluye a la planeación, los métodos y la capacidad de ingeniería, las herramientas y la tecnología), así como la creación de un plan de acción que defina los pasos, los</p>

Continua 

MODELO	VENTAJAS	DESVENTAJAS	CARACTERÍSTICAS
	<p>para la identificación de las prácticas a considerar.</p> <ul style="list-style-type: none"><li>- Dar soporte a la evaluación, indicando cómo el estándar de referencia ha sido aplicado en la organización evaluada.</li><li>- Asegurar la fiabilidad y capacidad de repetición de la evaluación.</li><li>- No lucrativo</li></ul>		<p>detalles de la implantación y los marcos temporales para que la organización aumente su capacidad de entrega de productos y servicios de calidad.</p>

### **2.3 Descripción de estándares escogidos para trabajo de investigación**

Para el proyecto se han escogido los estándares que se describen en el siguiente apartado:

#### **2.3.1 Norma ISO/IEC 25000**

La norma brinda una guía para la utilización de la nueva serie de estándares internacionales, llamados Requisitos y Evaluación de Calidad de Productos de Software (SQuaRE). Se constituye en una serie de normas basadas en la ISO 9126 y en la ISO 14598 (Evaluación del Software), y su objetivo principal es guiar el desarrollo de los productos de software con la especificación y evaluación de requisitos de calidad (Marcos, Arroyo, Garzas, & Piattini, 2008).

Define criterios para la especificación de requisitos de calidad de productos de software, sus métricas y su correspondiente evaluación. Incluye un modelo de calidad dividido en dos partes para unificar las definiciones de calidad de los clientes, con los atributos en el proceso de desarrollo.

SQuaRE está formada por las divisiones siguientes (Norma ISO/IEC 25000, 2017):

- **ISO/IEC 2500n.** División de gestión de calidad. Los estándares que forman esta división definen todos los modelos comunes, términos, y referencias a los que se alude en las demás divisiones de SQuaRE.

- **ISO/IEC 2501n.** División del modelo de calidad. El estándar que conforma esta división presenta un modelo de calidad detallado, incluyendo características para la calidad interna, externa y en uso.
- **ISO/IEC 2502n.** División de mediciones de calidad. Los estándares pertenecientes a esta división incluyen un modelo de referencia de calidad del producto de software, definiciones matemáticas de las métricas de calidad y una guía práctica para su aplicación. Presenta aplicaciones de métricas para la calidad de software interna, externa y en uso.
- **ISO/IEC 2503n.** División de requisitos de calidad. Los estándares que forman parte de esta división ayudan a especificar los requisitos de calidad. Estos requisitos pueden ser usados en el proceso de especificación de requisitos de calidad para un producto de software que va a ser desarrollado o como entrada para un proceso de evaluación. El proceso de definición de requisitos se guía por el establecido en la norma ISO/IEC 15288.
- **ISO/IEC 2504n.** División de evaluación de la calidad. Estos estándares proporcionan requisitos, recomendaciones y guías para la evaluación de un producto de software, tanto si la llevan a cabo evaluadores, como clientes o desarrolladores.
- **ISO/IEC 25050–25099.** Estándares de extensión SQuaRE. Incluyen requisitos para la calidad de productos de software “Off-The-Self” y para el formato común de la industria (CIF) para informes de usabilidad (Marcos, Arroyo, Garzas, & Piattini, 2008).

### **2.3.2 Metodología Bootstrap**

Bootstrap es una metodología (Kuvaja et al., 1994) que es el resultado de un proyecto europeo basado en los modelos CMM (Paulk, Curtis, Chrissis, & Weber, 1993) e ISO 9000 (Guler, Guillen, & Macpherson, 2002), el proyecto es desarrollado y mantenido por la organización European Strategic Programme for Research in Information Technology (ESPRIT), proporciona una alternativa para las organizaciones que están interesadas en mejorar su proceso de desarrollo de software, y alcanzar la certificación ISO, ya que combina y realiza las formas establecidas por CMM y la certificación ISO 9000 (Komi-Sirviö, 2004).

La metodología Bootstrap engloba tanto la evaluación para establecer el diagnóstico de un proceso para desarrollo de software (el cual incluye a la planeación, los métodos y la capacidad de ingeniería, las herramientas, y la tecnología), así como la creación de un plan de acción que defina los pasos, los detalles de la implantación y los marcos temporales para que la organización aumente su capacidad de entrega de productos y servicios de calidad. De acuerdo con (Escalone, 2006), los objetivos de la metodología Bootstrap son:

- Proporcionar soporte para la evaluación de la capacidad de los procesos utilizando un conjunto de prácticas de Ingeniería del Software.
- Incluir estándares de Ingeniería del Software reconocidos internacionalmente como fuentes para la identificación de las prácticas a considerar.

- Dar soporte a la evaluación, indicando cómo el estándar de referencia ha sido aplicado en la organización evaluada.
- Asegurar la fiabilidad y capacidad de repetición de la evaluación.
- Identificar las fortalezas y debilidades de los procesos de la organización evaluada.
- Dar soporte a la creación y aplicación de un plan de mejora que genere resultados aceptables y fiables, de forma que las acciones del plan de mejora permitan alcanzar los objetivos de la organización.
- Ayudar a incrementar la eficacia de los procesos poniendo en práctica los requisitos del estándar en la organización.

Bootstrap es una metodología de evaluación y mejora de procesos de software, que se basa en evaluar el nivel de capacidad, y de productividad de una Unidad de Desarrollo Software (Software Producing Unit, SPU), de acuerdo con (Tuya, Ramos, & Dolado, 2007) su modelo se enfoca a:

- Evaluar una SPU y sus proyectos, proporcionando perfiles analíticos para cada uno de ellos, de forma que se establezca la madurez de su proceso, identificando sus puntos fuertes y débiles.
- Deducir las áreas de mejora a partir de los perfiles analíticos, realizando un plan de alto nivel de las acciones recomendadas para conseguir la misma.
- Transformar el plan en una serie de mini-proyectos para implementar las mejoras recomendadas anteriormente.



Este proyecto fue creado por la Comisión Europea como parte del programa ESPRIT (ESPRIT 5441 BOOTSTRAP: A European Assessment Method to Improve Software Development). La administración y el mantenimiento del programa Bootstrap corresponden al Grupo Europeo de Interés Económico del Instituto Bootstrap (Bootstrap Institute European Economic Interest Group, BI EEIG) de Milán, Italia. El interés principal del programa Bootstrap es evaluar y mejorar la capacidad de las Unidades Productoras de Software (SPU, Software Producing Units).

Mediante esta metodología se tratará la mejora de procesos de software. ISO/IEC TR 15504 define un proceso como un grupo de actividades interrelacionadas, donde una entrada se transforma en una salida. Se podría decir que la mejora de procesos es en parte mejor que la reingeniería. Esta metodología mediante prácticas, herramientas y estándares de calidad internacional; mide, evalúa y propone mejoras al proceso de desarrollo de software que siguen las Unidades de Producción de Software (UPS) de las empresas.

Bootstrap surge como parte del programa estratégico europeo para investigación en tecnología de información. Este proyecto al igual que otros, tiene como principio el reducir costos y mejorar la calidad previendo problemas. Su objetivo es desarrollar un método para la evaluación de procesos de desarrollo de software. Inicialmente, se basó en el modelo de madurez de CMM añadiendo conceptos de calidad de ISO 9000. A esto incluyó conceptos para poder evaluar desarrollos de SW de otras industrias distintas a la militar y cambiar su cobertura de evaluación para tomar desde pequeñas UPS hasta grandes corporaciones. Para lograr esto, ha puesto especial énfasis en los conceptos

de ISO 9000; generando guías para mejoras en procesos de desarrollo de SW; analizado evaluaciones y mejoras de los procesos de desarrollo; y manteniendo una base de datos de soporte.

El programa Bootstrap combina las normas ISO 9000, las normas europeas para la Ingeniería de Software y el Modelo de Madurez de la Capacidad CMM para sentar una base con la cual evaluar y dar asesoría. La metodología Bootstrap engloba tanto la evaluación para establecer el diagnóstico de un proceso para desarrollo de software, el cual incluye la organización, los métodos y la capacidad de ingeniería, las herramientas y la tecnología, como la creación de un plan de acción que defina los pasos, los detalles de la implantación y los marcos temporales para que la organización aumente su capacidad de entrega de productos y servicios de calidad. El resultado de la evaluación es un perfil basado en el instrumento de evaluación de Bootstrap que añade una segunda dimensión a los niveles del CMM: el atributo de la calidad del proceso. Se pretende que mediante el programa Bootstrap se identifiquen los atributos de un proyecto de una organización que desarrolle software y que se asignen todas las preguntas del cuestionario a los atributos de la calidad del proceso, así como a los niveles de madurez.

El Instituto Bootstrap es una organización no lucrativa dedicada a la mejora continua del modelo de calidad de software llamado Bootstrap, también tiene como propósito ayudar a la industria europea del software para mejorar su competitividad.

Los principios del Instituto Bootstrap sostienen que: la metodología sea accesible a todos y crezca de forma que permita mejoras, la evolución de la metodología sea

democrática (por cada miembro un voto), provea un servicio a la industria europea y opere como una empresa no lucrativa.

El Instituto tiene como objetivos: la mejora continua de la metodología para la evaluación de la calidad de los procesos de desarrollo de SW, tomando en cuenta los estándares relevantes de ISO 9000 y otras iniciativas internacionales en el área; esto incluye, la forma de distribuirlo y el material de entrenamiento; la promoción para ampliar su cobertura en la industria europea y así consolidarse como estándar; licenciarla a terceros; el manejo apropiado de la base de datos de resultados de las evaluaciones llevadas a cabo por asesores certificados; la certificación de los asesores; y la certificación de organizaciones evaluadas.

Las principales actividades del Instituto son:

1. Evaluar a empresas.
2. Capacitación en la metodología y mejoras de la misma.
3. Certificación de asesores.
4. Recolección y administración de los datos de las evaluaciones.
5. Definir mecanismos para mantener la confidencialidad de los datos.
6. Representación en otros trabajos de estandarización.
7. Cooperación con el European Software Institute.
8. Coordinación de evaluaciones multinacionales.
9. Foro para obtener licencias y asesores independientes.

Bootstrap es un método para analizar, rediseñar y mejorar los procesos de negocio del desarrollo de software. Este se compone de: un modelo, un proceso de evaluación, una base de datos de soporte, un proceso de mejora y los instrumentos de evaluación.

Los objetivos de la metodología Bootstrap son:

- Proporcionar soporte para la evaluación de la capacidad de los procesos utilizando un conjunto de prácticas de Ingeniería del Software (SW).
- Incluir estándares de Ingeniería del Software reconocidos internacionalmente como fuentes para la identificación de las prácticas a considerar.
- Dar soporte a la evaluación, indicando como el estándar de referencia ha sido aplicado en la organización evaluada.
- Asegurar la fiabilidad y capacidad de repetición de la evaluación.
- Identificar las fortalezas y debilidades de los procesos de la organización evaluada.
- Dar soporte a la creación y aplicación de un plan de mejora que genere unos resultados aceptables y fiables, de forma que las acciones del plan de mejora permitan alcanzar los objetivos de la organización.

Ayudar a incrementar la eficacia de los procesos poniendo en práctica los requisitos del estándar en la organización.

El enfoque de esta metodología es evaluar el proceso, no el producto. Para eso se definen un conjunto de características para los procesos, provee un análisis cuantitativo, produce vistas analíticas, hace evidente fortalezas y debilidades, identifica áreas de mejora, provee recomendaciones y sugiere un plan de implementación.

El modelo Bootstrap define el paradigma Organización-Metodología-Tecnología que se usa en Bootstrap para los niveles de evaluación y agrupación de resultados. Dicho modelo ha sido estructurado en correspondencia con la arquitectura de procesos definida en la ISO 15504 V2.0.

La categoría de Organización (Organization) tiene 3 procesos, los cuales tienen una correspondencia con la ISO 15504 v.-98, que presenta algunos cambios respecto a la v.2.0 de la misma norma.

- ORG.1 Business Engineering (Ingeniería de negocio), se corresponde con Organizational Alignment. Este proceso sirve para asegurar que todo en la organización tiene una visión común respecto a los objetivos de negocio de la misma.
- ORG.2 Human Resource Management (Gestión de los recursos humanos), se corresponde con el proceso del mismo nombre, y que debe permitir conseguir las habilidades individuales y definición de roles necesarios en la organización.
- ORG.3 Infrastructure Management (Gestión de la infraestructura), se corresponde con Infrastructure, que se usa para establecer y mantener una

infraestructura estable y fiable que de soporte a los demás procesos. Esto puede incluir hardware, software, métodos, herramientas, técnicas, etc.

La categoría de Metodología (Methodology) se divide en procesos dependientes del ciclo de vida, independientes del ciclo de vida y relacionados con los procesos.

- Life Cycle Dependent (Dependientes del ciclo de vida): está formada por procesos que directamente especifican, implementan o mantienen el producto de software, su relación con el sistema y su documentación.
  - Life Cycle Independent (Independientes del ciclo de vida), se subdivide en: Management (Gestión o administración): procesos utilizados en la gestión del proyecto o algún proceso en el ciclo de vida del software.
  - Support (Soporte): formada por procesos que dan soporte a cualquiera del resto de procesos (incluidos los de soporte), en distintos puntos del ciclo de vida del software.
  - Customer-Supplier (Cliente-Proveedor): está formada por procesos que afectan directamente al cliente, soportan el desarrollo y la transición del software al cliente; y permiten la correcta operación y uso del producto y/o servicio del software.
- Process-Related (Relacionados con los procesos): estos procesos también tienen correspondencia directa con los de la ISO 15504 v.-98:
  - PRO.1 Process Definition (Definición de procesos), se corresponde con Improvement/ Process Establishment, que implica establecer y mantener

un conjunto de procesos en la organización junto con la mejora continua de los mismos.

- PRO.2 Process Improvement (Mejora de procesos), se corresponde tanto con Improvement Process como con Assessment Process, en los que se determina las fortalezas y debilidades de los procesos para poder así utilizarlo en la mejora de los mismos.
- PRO.3 Process Assessment (Evaluación del proceso)
- PRO.4 Measurement (Medición)

La categoría Tecnología cuenta con cuatro procesos, que no tienen una correspondencia directa con los de la ISO 15504. Aunque no existe una correspondencia podemos decir que los procesos TEC.2 y TEC.3 pueden ser considerados como parte del proceso de Infraestructura. Por su parte los procesos de “Innovación Tecnológica” e “Integración de herramientas” tienen un alcance distinto, el primero está referido a la forma en la que entran las nuevas tecnologías en la organización, mientras que la segunda busca incrementar el grado de integración de las herramientas en la organización.

Cada proceso tiene un conjunto de prácticas base asociadas, que describen las actividades esenciales de un proceso específico, la realización de las prácticas base indica el grado de alcance de la finalidad del proceso. Cada atributo de proceso tiene un conjunto de prácticas de gestión asociadas, que son las que implementan o institucionalizan un proceso de una manera general. La realización de las prácticas de gestión indica la consecución del atributo en esa instancia del proceso.

Respecto del Proceso de Evaluación, se puede decir que el modelo Bootstrap se basa en evaluar las unidades de producción de software (UPS) de la organización, a través de sus proyectos para hacer un cambio a toda la organización. El proceso de evaluación es parte de la mejora. Los resultados de la evaluación dan la entrada principal para el plan de acción de mejora y una realimentación para las actividades de mejora implementadas. Durante una evaluación Bootstrap los procesos organizacionales son evaluados para definir cada proceso.

La evaluación de la capacidad de los procesos está basada en el modelo de procesos Bootstrap. Dentro de este proceso, hay 4 etapas principales:

1. Preparación,
2. Ejecución de la evaluación,
3. Determinación del nivel de madurez y capacidades,
4. Presentación de los resultados de la evaluación.

En la etapa de Preparación se planean los pasos de la evaluación y se recoge la información sobre el contexto. El enfoque es respecto de las necesidades específicas y los objetivos de la organización, lo cual determina la definición y el alcance de la evaluación. Esto incluye encontrar a quién entrevistar, qué unidades organizacionales involucrar y la documentación a ser usada. La metodología Bootstrap y el método de evaluación se presentan para gestionar y proveer de personal técnico para crear una conciencia y conseguir un compromiso con la evaluación.

En esta etapa se realizan las siguientes tareas:



1. Un entrenamiento inicial para tener claros los objetivo,
2. Se seleccionan los proyectos a ser evaluados para obtener la mejor cobertura de la UPS,
3. Se define el personal de evaluación para minimizar la subjetividad,
4. Se define el personal a ser evaluado para obtener la mejor cobertura de los roles involucrados en los proyectos seleccionados,
5. Se hace el acuerdo de confidencialidad.

En la etapa de Ejecución la información sobre los procesos de la organización es recogida a través de entrevistas y evaluaciones de documentos disponibles. Esto se hace a nivel organizacional y de proyecto. A los entrevistados siempre se les pide que apoyen sus respuestas con evidencias. Las tareas de esta etapa son:

1. Una breve reunión de apertura, para obtener un enfoque del personal a ser entrevistado,
2. Completar los cuestionarios con características generales de la UPS,
3. Completar los cuestionarios del proyecto elegido, incluyendo la evaluación de cómo es aplicado el proceso de producción,
4. Revisión preliminar de la evaluación y
5. Reunión final, con el fin de presentar los resultados de la evaluación y obtener el consenso para poder pasar a la fase de mejoras.

En la etapa de Determinar el nivel de madurez y capacidades, es donde se califica cada pregunta con uno de los 5 valores posibles: nulo, débil, regular, extenso o no aplica. Para cada atributo clave se obtiene un nivel de madurez, aplicando un algoritmo

numérico, dando como resultado uno de estos niveles: 1-inicial, 2-repetible, 3-definido, 4administrado o 5-optimizado. Estos niveles de madurez están subdivididos en cuatro. Los procesos de organización y metodología se califican de 1 a 5, mientras que el de tecnología se califica sólo con dos niveles A o B.

En la etapa de Presentación de los resultados de la evaluación (4), la organización recibe 2 reportes, uno con los resultados de la evaluación de la UPS y otro con los resultados del proyecto evaluado. El correspondiente a la UPS contiene información como: un resumen ejecutivo, los objetivos de la UPS, los puntos débiles y fuertes, un plan de acción recomendado, etc. El reporte del proyecto contiene: comentarios del proyecto actual detallando lo referente a la organización, metodología y tecnología, los niveles de madurez para el proyecto, el plan de acción recomendado, etc.

Para expresar los resultados de la evaluación se suelen utilizar diferentes tipos de gráficos, de forma que muestren los valores de capacidad asignados, y otros que permitan comparar los resultados de la organización respecto a los obtenidos por otras organizaciones anteriormente evaluadas.

Una de las características principales de Bootstrap es la base de datos con que cuenta para hacer análisis. En esta base de datos se recogen automáticamente los datos obtenidos en todas las evaluaciones Bootstrap. El papel más importante de las evaluaciones Bootstrap es: (1) recoger datos de resultados de evaluaciones realizadas en Europa para proporcionar una imagen de la industria del software europea y (2) colocar cada organización evaluada dentro de un sector de la industria del software.

Respecto del Proceso de Mejora, se puede decir que otra parte importante de la metodología de Bootstrap, es el plan de mejora que sugiere. El proceso para obtener el plan de mejora consiste en, primero evaluar las necesidades de la organización tomando en cuenta las mejoras deseadas e indicadores sobre calidad del producto y servicio, tiempo de desarrollo, costos y riesgos del producto y del proyecto. Después, hacer una revisión y análisis de resultados de la evaluación, teniendo en cuenta las fortalezas y debilidades detectadas. Luego, definir las capacidades a mejorar, considerando un período entre 18 y 24 meses. Después, definir las prioridades de acuerdo a un análisis de impactos. Finalmente, en base a las actividades definidas, modificar la organización y responsabilidades para iniciar el cambio, estableciendo un marco de tiempos para su desarrollo y evaluación.

El objetivo del plan de acción es incrementar la cobertura de las soluciones organizacionales, metodológicas y de herramientas necesarias en los procesos de la UPS. El plan de acción debe considerar los riesgos de las mejoras y el impacto sobre la UPS.

El proceso de evaluación es apoyado por:

1. Cuestionarios.
2. Herramientas para el registro y presentación de resultados.
3. Guías para los asesores.

Los cuestionarios son usados para dirigir las entrevistas, donde los asesores los completan en base a discusión y análisis de material documentado. La mayoría de las

preguntas están basadas en términos como: la existencia de un procedimiento formal, la existencia de metodología, la existencia de estándares, la disponibilidad de tecnología, recomendaciones de uso de la tecnología, desempeño de tareas en base a un procedimiento, responsabilidades en la toma de decisiones, desempeño de un análisis sistemático de resultados, etc.

Las evaluaciones son una herramienta importante en cualquier organización, para establecer un marco de referencia para la mejora continua. Bootstrap ofrece esto proporcionando retroalimentación para establecer una calidad del software competitiva. Este es quizá el modelo más completo que está en operación junto con el CMM, pero su cobertura se ha enfocado sólo a Europa. Esto quizá cambie en el mediano plazo, ya que los mercados globales hacen necesario estándares internacionales y no sólo regionales. Además, permite compararse con ISO 9000, que es el estándar de calidad más difundido. Esto permite que quien se evalúa con Bootstrap, no tenga que hacer un esfuerzo adicional para mejorar y obtener una certificación en ISO 9000.

Bootstrap es un modelo mientras que ISO es un estándar, así que el modelo ha tomado en consideración los mejores puntos del estándar. Bootstrap podría ser principalmente un método de evaluación y solo sugiere mejoras. Esto da una pequeña ayuda una vez que se produce la mejora. La metodología está creada para auto-evaluación y evaluación de tercera parte, pero es también posible usarla para evaluación de segunda parte.

Aunque muchos usuarios de Bootstrap dicen tener problemas con los cuestionarios, la crítica general del modelo es en su mayoría positiva. La metodología

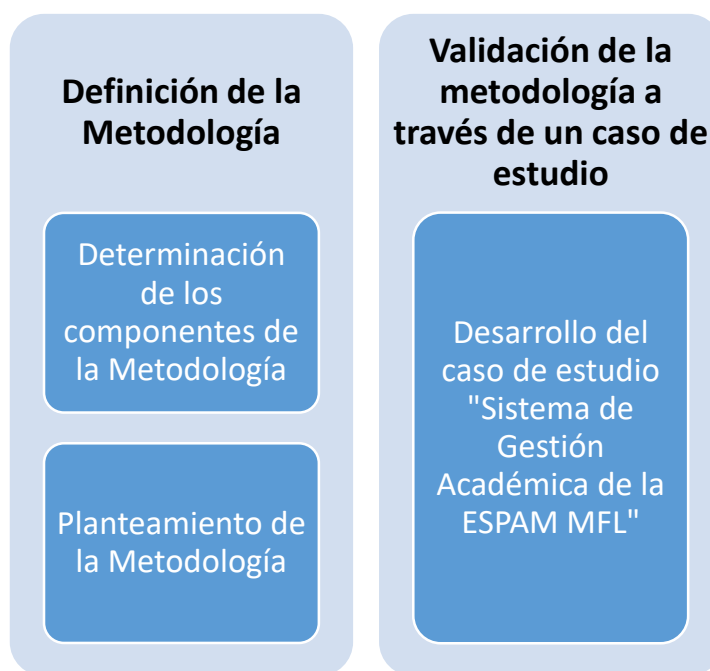
tiene una gran ventaja, compara los resultados de la evaluación con los resultados de sus competidores. Parece que da un gran resultado cuando prioriza qué necesidades se deben mejorar primero.

## CAPÍTULO III

### DEFINICIÓN DE LA METODOLOGÍA

#### 3.1 Metodología de Desarrollo Propuesta

El desarrollo de la tesis se llevó a cabo en dos fases que se detallan a continuación:



**Figura 3** Metodología propuesta

##### 3.1.1 Determinación de los componentes de la metodología

###### 3.1.1.1 Ciclo de vida de la calidad del Producto Software

En la presente metodología se maneja la calidad del producto software en tres principales fases:

**Tabla 5***Fases de la calidad del software*

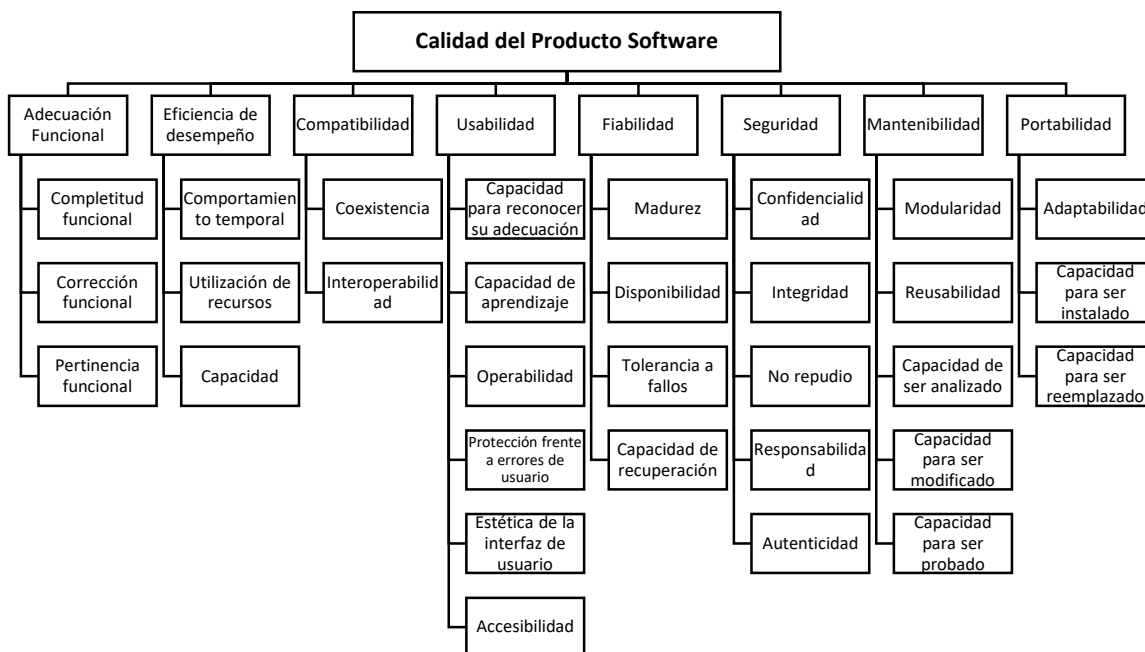
<b>Calidad</b>	<b>Descripción</b>	<b>Observación</b>
<b>Interna</b>	Cuando el producto software se encuentra en desarrollo.	Se utilizan para verificar el producto a lo largo de las distintas etapas del desarrollo y pueden utilizarse también para definir estrategias y criterios de evaluación y verificación.
<b>Externa</b>	Cuando el producto software se encuentra en funcionamiento.	Se utilizan para la verificación y validación técnica del producto. Estos requisitos ayudan a determinar los requisitos de calidad interna, pero, además, pueden servir para predecir si se alcanzará la calidad en uso deseada.
<b>En Uso</b>	Cuando el producto software se encuentra en uso.	Especifican el nivel de calidad requerido desde el punto de vista del usuario. Estos requisitos son los que determinan la validación del software por parte del usuario. Como indica el modelo de ciclo de vida, la especificación de requisitos de calidad en uso ayuda a determinar los requisitos de calidad externa.

Para cada una de estas fases existen requisitos que responden a una necesidad del producto que deberán ser implementados y validados.

**3.1.1.2 Características, sub-características y métricas**

La presente metodología, basada en la Norma ISO/IEC 25000 clasifica a la calidad del producto, en características que se dividen en sub-características y atributos de calidad, el cual consiste de dos partes:

- El modelo para la calidad interna y externa de un producto software.
- El modelo para la calidad en uso de un producto software.



**Figura 4** Calidad externa

- **MODELO DE CALIDAD DEL PRODUCTO SOFTWARE (CALIDAD INTERNA Y EXTERNA)**

La metodología define 8 características para la calidad interna y externa de un producto software: Adecuación Funcional, Fiabilidad, Eficiencia en el Desempeño, Facilidad de Uso, Seguridad, Compatibilidad, Mantenibilidad y Portabilidad, las cuales a su vez son subdivididas en sub-características. Estas sub-características pueden ser medidas con métricas internas o externas.

**Adecuación funcional:** Representa la capacidad del producto o sistema software para proporcionar las funciones necesarias para satisfacer al usuario.

Esta característica se divide en las siguientes sub-características:



- **Complejidad funcional:** Capacidad del sistema software para proporcionar un conjunto de funcionalidades apropiadas para cubrir todas las tareas y objetivos determinados por el usuario.
- **Corrección funcional:** Capacidad del sistema software para proporcionar los resultados correctos con el grado necesario de precisión.
- **Pertinencia funcional:** Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

**Eficiencia en el desempeño:** Capacidad de un producto o sistema software de proporcionar un rendimiento apropiado, respecto a la cantidad recursos utilizados bajo determinadas condiciones. Esta característica se divide en las siguientes sub-características:

- **Comportamiento Temporal:** Capacidad de un sistema software para proporcionar los tiempos de respuesta y procesamiento apropiados.
- **Utilización de Recursos:** Capacidad en que un sistema software utiliza las cantidades y tipos de recursos adecuados.
- **Capacidad:** Capacidad de un sistema software de cumplir con los requisitos determinados.

**Compatibilidad:** Capacidad de dos o más sistemas software, para llevar a cabo sus funciones intercambiando información mientras comparten el mismo entorno. Esta característica se divide en las siguientes sub-características:

- **Co-Existencia:** Capacidad de un sistema software para coexistir en un entorno en el cual comparten recursos comunes con otro software independiente.
- **Interoperabilidad:** Capacidad de dos o más sistemas software para intercambiar información y utilizar dicha información.

**Usabilidad:** Capacidad del producto o sistema software para que sea entendido, aprendido, agrado y usado por el usuario. Esta característica se divide en las siguientes sub-características:

- **Capacidad de reconocer su adecuación:** Capacidad del sistema software que permite al usuario entender si el software es adecuado para sus necesidades.
- **Capacidad para ser entendido:** Capacidad del sistema, que permite al usuario entender si el software es adecuado para alcanzar sus objetivos determinados.
- **Operatividad:** Capacidad de un sistema software que permite al usuario operarlo y controlarlo con facilidad.
- **Protección contra errores del usuario:** Capacidad en que el sistema brinda la protección necesaria contra errores que realizan los usuarios.
- **Estética de la Interfaz del usuario:** Capacidad en que la interfaz de usuario llega a satisfacer y agradar al usuario.
- **Accesibilidad técnica:** Capacidad del sistema software para que se permita ser utilizado por usuarios con determinadas discapacidades.

**Fiabilidad:** Capacidad del producto o sistema software para realizar las funciones específicas cuando se utiliza bajo ciertas condiciones y periodos de tiempo determinadas. Esta característica se divide en las siguientes sub-características:

- **Madurez:** Capacidad del sistema software para satisfacer las necesidades de fiabilidad durante el funcionamiento normal.
- **Disponibilidad:** Capacidad de un sistema software de estar operativo y accesible para su uso cuando se necesite.
- **Tolerancia a Fallos:** Capacidad de un sistema software para operar cuando se presenten fallos.
- **Recuperabilidad:** Capacidad de un sistema software para reestablecer el estado del sistema y recuperar datos que se hayan afectado, en caso de interrupción o fallo.

**Seguridad:** Capacidad de proteger la información y los datos, de manera que personas o sistemas no autorizados puedan tener acceso para consultas o actualizaciones. Esta característica se divide en las siguientes sub-características:

- **Confidencialidad:** Capacidad de proteger la información y el acceso a datos no autorizados, ya sea de manera accidental, o intencional.
- **Integridad:** Capacidad de un producto, sistema, o componente software para evitar accesos no autorizados a datos o programas de computación.
- **No – repudio:** Capacidad para demostrar que los eventos han ocurrido, de manera que dichos eventos no puedan ser refutados posteriormente.

- **Responsabilidad:** Capacidad de dar seguimiento a las acciones que fueron realizadas por una entidad.
- **Autenticidad:** Capacidad de demostrar la identidad de un sujeto o un recurso.

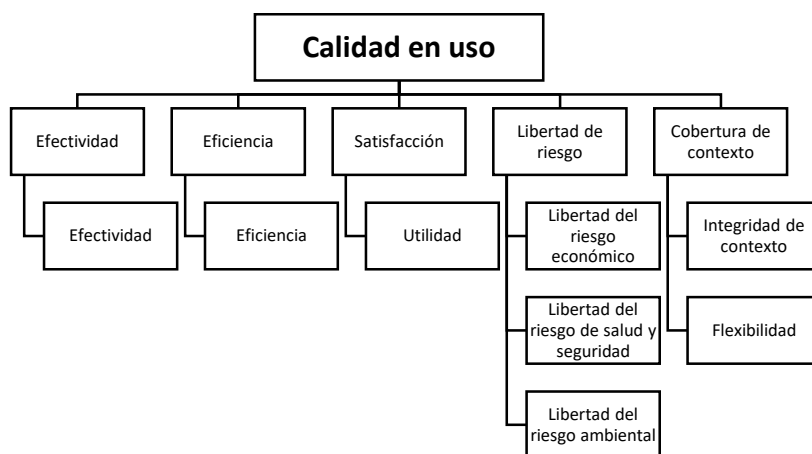
**Mantenibilidad:** Capacidad del sistema software para ser modificado o actualizado debido a necesidades evolutivas y correctivas. Esta característica se divide en las siguientes sub-características:

- **Modularidad:** Capacidad de un sistema software que cuando sea modificado no afecte a otras funcionalidades del sistema.
- **Reusabilidad:** Capacidad de un activo (Información, Software, Hardware, Usuarios) para ser utilizado en más de un sistema o en la construcción de otros activos.
- **Capacidad de ser Analizado:** Facilidad con la que se puede llevar a cabo un análisis del impacto de una determinada modificación en el sistema.
- **Capacidad de ser Modificado:** Capacidad del sistema para permitir que sea modificado sin causar daños o reducir la calidad del producto existente.
- **Capacidad de ser Probado:** Facilidad de realizar pruebas a un sistema o componente software, para determinar si se han cumplido con los requerimientos establecidos.

**Portabilidad:** Capacidad de un sistema, o componente software, de ser trasladado de un entorno a otro sin que esto afecte la funcionalidad de cada sistema. Esta característica se subdivide en las siguientes sub-características:

- **Adaptabilidad:** Capacidad de un sistema software de ser adaptado a distintos entornos.
  - **Capacidad de ser Instalado:** Capacidad de un sistema para que pueda ser fácilmente instalado y/o desinstalado.
  - **Capacidad de ser Reemplazado:** Capacidad del sistema software para ser utilizado en lugar de otro sistema en el mismo entorno y cumpliendo con el mismo objetivo.
- **MODELO PARA LA CALIDAD EN USO**

El modelo de calidad en uso define 5 características: Efectividad, Eficiencia, Satisfacción, Libertad de Riesgo y Cobertura de Contexto, las cuales a su vez son subdivididos en sub-características descritas en la Figura 6, estas sub-características pueden ser medidas con métricas de calidad en uso.



**Figura 5** Calidad en uso

El resultado de la calidad en uso depende necesariamente del logro de la calidad externa, que a su vez depende necesariamente del logro de la calidad interna.

**Efectividad:** Capacidad del sistema software para alcanzar los objetivos o necesidades del usuario, al momento de utilizar el sistema.

**Eficiencia:** Capacidad del sistema software para alcanzar los objetivos del usuario, utilizando los recursos mínimos.

**Satisfacción:** Capacidad del sistema software para satisfacer las diferentes necesidades mínimas de los usuarios al utilizarlo. Esta característica se divide en las siguientes sub-características:

- **Utilidad:** Grado en que un usuario es satisfecho cuando logra alcanzar sus objetivos planteados.

**Libertad de riesgo:** Capacidad que tiene un producto o sistema software en reducir el riesgo potencial relacionado con la situación económica, vida humana, salud o medio ambiente.

Esto incluye la salud y seguridad, tanto del usuario y aquellos afectados por el uso, así como las consecuencias materiales o económicas no deseadas.

En este caso, el riesgo es la probabilidad de ocurrencia y las posibles consecuencias negativas cuando se presenta una amenaza determinada.

Esta característica se subdivide en las siguientes sub-características las que permiten establecer el grado en el cual los objetivos podrían estar en riesgo.

- Libertad del riesgo económico.
- Libertad del riesgo de salud y seguridad.
- Libertad del riesgo ambiental.

**Cobertura de contexto:** Capacidad de un producto o sistema software para ser utilizado con efectividad, eficiencia, libertad de riesgo y satisfacción en ámbitos de uso que fueron definidos. Esta característica se subdivide en las siguientes sub-características:

- **Integridad de Contexto:** Capacidad de un sistema software para ser utilizado en los ámbitos de uso definidos.
- **Flexibilidad:** Capacidad de un sistema software para ser utilizado fuera de los ámbitos de uso que fueron definidos inicialmente.

Los usuarios deben seleccionar las características y sub-características de calidad a ser evaluadas, identificar las métricas más apropiadas y relevantes e interpretar los resultados de la medición de una manera objetiva. Las métricas de calidad interna pueden ser aplicadas durante las etapas de desarrollo del producto o sistema software (definición de requerimientos, especificación de diseño o código fuente), para que de esta manera los usuarios puedan identificar los problemas de calidad e iniciar la acción correctiva lo más antes posible en el ciclo de vida de desarrollo. Las métricas de calidad externa pueden ser usadas para medir el comportamiento del sistema software, pudiendo solo ser usadas durante las etapas de pruebas y en alguna etapa de operación. La medición se debe llevar a cabo cuando el sistema software está en ejecución. Las métricas de calidad en uso miden si un sistema software satisface las necesidades

específicas de los usuarios, para ello se debe llevar a cabo la medición en un ambiente real donde se esté ejecutando el sistema de métricas.



### 3.1.1.3 Métricas de Calidad del Producto Software

Las métricas para la calidad interna y externa evalúan las características definidas anteriormente (Calidad Interna y Externa). Las métricas para la calidad interna y externa se definen a continuación:

**Tabla 6**

#### Métricas

Características	Sub-características	Métricas	Método de aplicación	Fórmula	Valor deseado
<b>Adecuación funcional</b>	Complejidad funcional	Complejidad de la implementación funcional.	Contar el número de las funciones indicadas en la especificación de requerimientos y el número de funciones que faltan o están incorrectas.	$X = A / B$ A = Número de funciones que están incorrectas o que no fueron implementadas B = Número de las funciones establecidas en la especificación de requisitos. Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0 es el mejor
	Corrección funcional	Exactitud	Contar el número de elementos de datos implementados con el estándar específico de exactitud y el número total de elementos de datos implementados	$X = A/B$ A = Número de elementos de datos implementados con el estándar específico de exactitud B = Número total de elementos de datos implementados Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0 es el mejor

Continúa 

	Precisión computacional	Contar el número de Cálculos inexactos encontrados y tomar el tiempo de operación	$X = A/T$ A = Numero de cálculos Inexactos encontrados T = Tiempo de operación Dónde: $T > 0$	$X = A/T$ El más cercano a $0/T$ es el mejor. Donde el peor caso es $\geq 10/T$	
	Pertinencia funcional	Cumplimiento de requerimientos	Porcentaje de requerimientos cumplidos	$X=A/B$ A=Total de requerimientos se entregaron B= Total Donde: $B>0$	$0<X\leq 1$ El más cercano a 1 es el mejor
<b>Eficiencia de desempeño</b>	Comportamiento temporal	Tiempo de respuesta	Tomar el tiempo desde que se envía la petición hasta obtener la respuesta	$X = B - A$ A= Tiempo de envío de petición B = Tiempo en recibir la primera respuesta	$0\leq X\leq 1$ El más cercano a 0 es el mejor. Donde el peor caso es $\geq 15t$
		Tiempo de espera	Tomar el tiempo cuando se inicia un trabajo y el tiempo en completar el trabajo	$X = B - A$ A= Tiempo cuando se inicia un trabajo B = Tiempo en completar el trabajo	$0\leq X\leq 1$ El más cercano a 0 es el mejor. Donde el peor caso es $\geq 15t$
		Rendimiento	Contar el número de tareas completadas en un intervalo de tiempo	$X = A/T$ A= Número de tareas completadas T = Intervalo de tiempo Dónde: $T > 0$	$X = A/T$ El más lejano a $0/t$ es el mejor. Donde el mejor caso es $\geq 10/t$

Continúa 


Utilización de recursos	Líneas de código.	Contar el número de líneas de código (sin tomar en cuenta espacios ni comentarios) que existen en una determinada función	$X = A$ $A =$ Número de líneas de código	$1 \leq X \leq 50$ El más cercano a 1 es el mejor. Donde el peor caso es $\geq 50$ líneas de código
	Utilización de CPU.	Tomar el tiempo de operación y la cantidad de tiempo de CPU que se usa para realizar una tarea	$X = B - A$ $A =$ La cantidad de tiempo de CPU que realmente es usado para realizar una tarea $B =$ Tiempo de operación Dónde: $B > 0$	$0 \leq X \leq 1$ Cuanto más se acerque a 0 es lo mejor. Donde el peor caso es $\geq 15t$ .
	Utilización de la memoria.	Medir la cantidad total de espacios de memoria y la cantidad de espacios de memoria que realmente es usado para realizar una tarea	$X = B - A$ $A =$ Cantidad de espacios de memoria que realmente es usado para realizar una tarea $B =$ Cantidad total de espacios de memoria Dónde: $B > 0$	$0 \leq X \leq 15$ El más cercano a 0 es el mejor
	Utilización de los dispositivos de E/S.	Tomar el tiempo de operación y el tiempo que los dispositivos de E/S pasan ocupados para realizar la tarea	$X = B - A$ $A =$ Tiempo que los dispositivos de E/S pasan ocupados para realizar la tarea $B =$ Tiempo de operación Dónde: $B > 0$	$0 \leq X \leq 15$ El más cercano a 0 es el mejor

Continua 

Capacidad	Número de peticiones online.	Contar el número máximo de peticiones online procesadas y tomar el tiempo de operación	$X = A/T$ A= Número máximo de peticiones online procesada T = Tiempo de operación Dónde: $T > 0$	$X = A/T$ El más lejano a $0/t$ es el mejor. Donde el mejor caso es $\geq 10/t$ .
	Número de accesos simultáneos.	Contar el número máximo de accesos simultáneos y tomar el tiempo de operación	$X = A/T$ A= Número máximo de accesos simultáneos T = Tiempo de operación Dónde: $T > 0$	$X = A/T$ El más lejano a $0/t$ es el mejor. Donde el mejor caso es $\geq 10/t$
	Sistema de transmisión de ancho de banda.	Contar la cantidad máxima de transmisión de datos y tomar el tiempo de operación	$X = A/T$ A= Cantidad máxima de transmisión de datos B = Tiempo de operación Dónde: $T > 0$	$X = A/T$ El más lejano a $0/t$ es el mejor. Donde el mejor caso es $\geq 10/t$
<b>Compatibilidad</b>	Co – Existencia	Co – existencia disponible.	Número de entidades con las que el producto puede coexistir y el número de entidades en el entorno de operación que requieren de coexistencia  $X = A/B$  A = Número de entidades con las que el producto puede coexistir B = Número de entidades en el entorno de operación que requieren de coexistencia  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor

Continúa 


Interoperabilidad	Conectividad con sistemas externos.	Número de interfaces implementadas con otros sistemas y el número total de interfaces externas	$X = A/B$  A= Número de interfaces implementadas con otros sistemas B = Número total de interfaces	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	
	Capacidad de intercambiar de datos.	Número de datos que se han intercambiado sin problemas con otro sistema y el número total de datos que se intercambiaran	$X = A/B$  A= Número de datos que se han intercambiado sin problemas con otro sistema B = Número total de datos que se intercambiaran  Dónde: B > 0	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	
<b>Usabilidad</b>	Capacidad de reconocer su adecuación	Integridad de descripción	Número de funciones descritas como entendibles en la descripción del producto.	$X = A/B$  A = Número de funciones (o tipos de funciones) descritas como entendibles en la descripción del producto B = Número total de funciones (o tipos de funciones)  Dónde: B > 0	$0 \leq X \leq 1$ El más cercano a 1 es el mejor

Continúa 

	Capacidad de demostración.	Contar el número de funciones descritas correctamente y contar el número total de funciones.	$X = A/B$ A = Número de funciones implementadas con capacidad de demostración B = Número total de funciones que requieren capacidad de demostración  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor
Capacidad de aprendizaje	Funciones evidentes.	Contar las funciones evidentes al usuario y comparar con el número total de funciones	$X = A/B$ A = número de funciones (o tipos de funciones) evidentes al usuario B = total de funciones (o tipos de funciones)	$0 \leq X \leq 1$ Entre más cercano a 1, mejor.
	Efectividad de la documentación del usuario o ayuda del sistema.	Contar el número de funciones descritas correctamente y contar el número total de funciones implementadas	$X = A / B$ A= Número de funciones descritas correctamente B = Número total de funciones implementadas  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor

Continúa 

Operatividad	Recuperabilidad del error operacional.	Número de funciones implementadas con tolerancia de error de usuarios y el número total de funciones requeridas con capacidad de tolerancia.	$X=A/B$ A= Número de funciones implementadas con tolerancia de error de usuarios B = Número total de funciones requeridas con capacidad de tolerancia.  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor
	Claridad de mensajes.	Número de mensajes implementados con explicaciones claras y el número total de mensajes implementados	$X = A / B$ A= Número de mensajes implementados con explicaciones claras B = Número total de mensajes implementados  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor

Continua 

---

Consistencia  
operacional.

Número de operaciones que se comportan de manera incoherente y el número total de operaciones que se comportan de forma normal

$$X = A / B$$

A= Número de operaciones que se comportan de manera incoherente B = Número total de operaciones que se comportan de forma normal

Dónde:  
B > 0

$0 \leq X \leq 1$   
El más cercano a 0 es el mejor

---

Posibilidad de  
personalización.


Número de funciones implementadas que pueden ser personalizadas durante la operación y el número de funciones que requieran la capacidad de personalización

$$X = A / B$$

A = Número de funciones implementadas que pueden ser personalizadas durante la operación B = Número de funciones que requieran la capacidad de personalización

Dónde:  
B > 0


$0 \leq X \leq 1$   
El más cercano a 1 es el mejor

Continua 


---



Protección contra errores del usuario	Verificación de entradas válidas.	Número de ítems de entrada que son validados y el número de ítems que necesitan ser validados	$X = A/B$ <p>A= Número de ítems de entrada que son validados B = Número de ítems que necesitan ser validados</p> <p>Dónde: B &gt; 0</p>	$0 \leq X \leq 1$ <p>El más cercano a 1, es el mejor</p>
Prevenición del uso incorrecto.	Número de funciones implementadas para evitar fallos de funcionamiento provocados por un uso incorrecto y el número total de operaciones iniciales incorrectas	$X = A/B$ <p>A = Número operaciones iniciales incorrectas B = Número de funciones implementadas para evitar fallos de funcionamiento provocados por un uso incorrecto</p> <p>Dónde: B &gt; 0</p>	$0 \leq X \leq 1$ <p>El más cercano a 1, el mejor</p>	


Continua 

Estética de la Interfaz del usuario	Personalización de la apariencia de la interfaz del usuario.	Número de tipos de elementos de interfaz que pueden ser personalizados y contar el número total de tipos de elementos de interfaz	$X = A/B$ <p>A= Número de elementos de interfaz que pueden ser personalizados B =Número total de elementos de interfaz</p> <p>Dónde: B &gt; 0</p>	$0 \leq X \leq 1$ <p>El más cercano a 1, es el mejor</p>
Accesibilidad técnica	Accesibilidad física	Contar el número de funciones a las que pueden acceder personas con discapacidad y contar el número total de funciones implementadas	$X = A/B$ <p>A= Número de funciones a las que pueden acceder personas con discapacidad B = Número total de elementos de interfaz</p> <p>Dónde: B &gt; 0</p>	$0 \leq X \leq 1$ <p>El más cercano a 1, es el mejor</p>


Continua 


<b>Fiabilidad</b>	Madurez	Disipación del fallo.	Contar el número de fallas corregidas en la base de diseño/codificación/pruebas y el número de fallas detectadas en las pruebas	$X = A/B$  A = Número de fallas corregidas en la fase de diseño/codificación/pruebas  B = Número de fallas detectadas en las pruebas  Dónde: $B > 0$	$0 \leq X \leq 1$  Cuanto más se acerque a 1 es lo mejor.
		Suficiencia de las pruebas.	Contar el número de casos de pruebas realizados en un escenario de operación durante la prueba y el número de casos de prueba a ser realizados para cubrir los requerimientos	$X = A/B$  A = Número de casos de pruebas realizadas en un escenario de operación durante la prueba  B = Número de casos de prueba a ser realizados para cubrir los requerimientos  Dónde: $B > 0$	$0 \leq X \leq 1$  Cuanto más se acerque a 1 es lo mejor.

Continua 


	Tiempo medio entre fallos	Tomar el tiempo de operación y contar el número total de fallas detectadas actualmente	$X = A/T$ A = Número total de fallas detectadas actualmente T = Tiempo de operación Donde $T > 0$	$X = A/T$ El más cercano a $0/T$ es el mejor
Disponibilidad	Tiempo de servicio	Tomar el tiempo de servicio del sistema que se proporciona actualmente y tomar el tiempo de servicio del sistema regulado en el cronograma operacional.	$X = A/B$ A = Tiempo de servicio del sistema que se proporciona actualmente B = Tiempo de servicio del sistema regulado en el cronograma operacional Dónde: $B > 0$	$0 \leq X \leq 1$ Cuanto más se acerque a 1 es lo mejor
	Tiempo medio de inactividad	Tomar el tiempo total de inactividad y contar el número de fallos observados	$X = A/T$ A = Número de fallos observados T = Tiempo total de inactividad Dónde: $T > 0$	$X = A/T$ El más cercano a $0/T$ es el mejor Continua 

Tolerancia a fallos	Prevención de fallos	Contar el número de ocurrencia de fallas serias y críticas evitadas contra los casos de pruebas de fallas iniciales y el número de casos de pruebas de fallas iniciales ejecutados durante las pruebas	$X = A/B$  A = Número de ocurrencia de fallas evitadas contra los casos de pruebas de fallas iniciales B = Número de casos de pruebas de fallas iniciales ejecutados durante las pruebas Dónde: $B > 0$	$0 \leq X \leq 1$  Cuanto más se acerque a 1 es lo mejor
	Redundancia (componentes)	Contar el número total de tipos de componentes y el número de tipos de componentes instalados de forma redundante	$X = A / B$  A= Número componentes/sistemas instalados de forma redundante B = Número total de componentes/sistemas instalados  Dónde: $B > 0$	$0 \leq X \leq 1$  Cuanto más se acerque a 1 es lo mejor


Continua 

	Anulación de operación incorrecta.	Contar el número de funciones implementadas que evitan fallas críticas y serias causadas por operaciones incorrectas y contar el número operaciones incorrectas presentadas	$X = A/B$  A = Número de operaciones incorrectas presentadas  B = Número total de funciones implementadas para anular operaciones incorrectas	$0 \leq X \leq 1$  Cuanto más se acerque a 1 es lo mejor
Capacidad de recuperación	Tiempo medio de recuperación.	Tomar el tiempo que le tomó al sistema en recuperarse y contar el número de casos en los cuales se ha observado que el sistema entró en recuperación	$X = A / T$  A = Número de casos en los cuales se ha observado que el sistema entró en recuperación  T = Tiempo que le tomó al sistema en recuperarse  Dónde: $T > 0$	$X = A/T$  El más cercano a $0/T$ es el mejor.  Donde el peor caso es $\geq 10/T$
				Continua 

<b>Seguridad</b>	Confidencialidad	Capacidad de control de acceso.	Número de diferentes tipos de operaciones ilegales detectados y el número de tipos de operaciones ilegales en la especificación	$X = A / B$  A = Número de diferentes tipos de operaciones ilegales detectados B = Número de tipos de operaciones ilegales en la especificación  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor
		Encriptación de datos.	Número de elementos de datos encriptados/desencriptados correctamente y el número de elementos de datos que requiere el encriptación/desencriptación	$X = A / B$  A = Número de elementos de datos encriptados/desencriptados correctamente B = Número de elementos de datos que requiere el encriptación/desencriptación	$0 \leq X \leq 1$ El más cercano a 1, es el mejor


Continua 

Integridad	Prevención de corrupción de datos.	Número de casos de corrupción de datos ocurridos en la actualidad y el número de accesos donde se espera que ocurran daños de datos	$X = A / B$  A = Número de casos de corrupción de datos ocurridos en la actualidad B = Número de accesos donde se espera que ocurran daños de datos  Dónde: B > 0	$0 \leq X \leq 1$ El más cercano a 0, es el mejor
No repudio	Utilización de firma digital.	Número de eventos procesados usando firma digital y el número de eventos que requieran la propiedad de no - repudio	$X = A / B$  A = Número de eventos procesados usando firma digital B = Número de eventos que requieran la propiedad de no - repudio	$0 \leq X \leq 1$ El más cercano a 1, es el mejor


Continúa 




	Responsabilidad	Capacidad de auditoría de acceso.	Número de accesos al sistema y los datos registrados en el log del sistema y el número de accesos ocurridos en la realidad	$X = A / B$  A = Número de accesos ocurridos en la realidad B = Número de accesos al sistema y los datos registrados en el log del sistema  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor
	Autenticidad	Métodos de autenticación.	Contar el número de métodos de autenticación previstos	$X = A$ A = Número de métodos de autenticación previstos	$X \geq 0$ Donde X es mayor a 0, siendo X el mejor igual o mayor a 2
<b>Mantenibilidad</b>	Modularidad	Capacidad de condensación.	Número de componentes que no son afectados por cambios de otros componentes y el número total de componentes específicos	$X = A / B$  A = Número de componentes que no son afectados por cambios de otros componentes B = Número total de componentes específicos  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0, es el mejor

Continúa 


	Acoplamiento de clases.	Número de relaciones que tiene una función con respecto a otras clases	$X = A$ $A =$ Número de relaciones que tiene una función con respecto a otras clases	$1 \leq X \leq 4$ El más cercano a 1, es el mejor
Reusabilidad	Ejecución de reusabilidad.	Número de elementos reutilizados y el número total de elementos de la biblioteca reutilizable	$X = A / B$ $A =$ Número de elementos reutilizados $B =$ Número total de elementos de la biblioteca reutilizable  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor
Capacidad de ser analizado	Capacidad de pistas de auditoría.	Número de datos realmente grabadas durante la operación y el número de datos previstos a grabarse para controlar el estado del sistema durante la operación	$X = A / B$ $A =$ Número de datos realmente grabadas durante la operación $B =$ Número de datos previstos a grabarse para controlar el estado del sistema durante la operación  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor

Continúa 


	Diagnóstico de funciones suficientes.	Número de funciones de diagnóstico implementadas y contar el número de funciones de diagnóstico requeridas en la especificación de requerimientos	$X = A/B$  $A =$ Número de funciones de diagnóstico implementadas $B =$ Número de funciones de diagnóstico requeridas en la especificación de requerimientos  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor
Capacidad para ser modificado	Complejidad ciclomática.	Contar las instrucciones condicionales, bucles, salidas de métodos y cláusulas AND y OR dentro de los condicionales.	$X = A+1$ $A =$ Número de instrucciones condicionales que tiene una función	$1 \leq X < 15$ El más cercano a 1, es el mejor
	Profundidad de herencia.	Contar las jerarquías empleadas en una determinada función o método	$X = A$ $A =$ Número de jerarquías empleadas para una determinada función.	$0 \leq X \leq 4$ El más cercano a 0 es el mejor

Continúa 


Grado de localización de corrección de impacto.	Número de fallas aparecidas después que se ha resuelto un fallo y contar el número de fallas resultas	$X = A/B$  A = Número de fallas aparecidas después que se ha resuelto un fallo B = Número de fallas resueltas  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0, es el mejor
Complejidad de modificación.	Tomar el tiempo de trabajo que le toma al desarrollador modificar y contar el número de modificaciones	$X = A/T$  A = Número de modificaciones B = Tiempo de trabajo que le toma al desarrollador modificar  Dónde: $T > 0$	$X = A/T$  El más lejano a 0/t es el mejor

Continua 

	Índice de éxito de modificación.	Contar el número de problemas dentro de un determinado período antes de mantenimiento y contar el número de problemas en el mismo periodo después del mantenimiento	$X = A/B$  A = Número de problemas dentro de un determinado período antes de mantenimiento B = Número de problemas en el mismo período después del mantenimiento  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0, es el mejor
Capacidad para ser probado	Complejidad funcional de funciones de pruebas.	Contar el número de funciones de prueba implementadas y contar el número de funciones de prueba requeridas	$X = A/B$  A = Número de funciones de prueba implementadas B = Número de funciones de prueba requeridas  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor

Continua 


Capacidad de prueba autónoma.	Contar el número de pruebas que están dependiendo de otros sistemas y contar el número total de pruebas dependientes con otros sistemas	$X = A/B$  A = Número de pruebas que están dependiendo de otros sistemas B = Número total de pruebas dependientes con otros sistemas  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0, es el mejor
Capacidad de reinicio de pruebas.	Contar el número de casos en los cuales el mantenedor puede pausar y restaurar las pruebas y contar el número de casos de pausa en la ejecución de pruebas	$X = A/B$  A = Número de casos en los cuales el mantenedor puede pausar y restaurar las pruebas B = Número de casos de pausa en la ejecución de pruebas  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor

Continua 

<b>Portabilidad</b>	Adaptabilidad	Adaptabilidad en entorno hardware.	Número funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con el entorno hardware y contar el número total de funciones las cuales han sido probadas	$X = A/B$	$0 \leq X \leq 1$ El más cercano a 0, es el mejor
				A = Número funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con el entorno hardware B = Número total de funciones que han sido probadas	
				Dónde: B > 0	
		Adaptabilidad en entorno de software.	Contar el número funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con el sistema y contar el número total de funciones las cuales han sido probadas	$X = A/B$	$0 \leq X \leq 1$ El más cercano a 0, es el mejor
				A = Número de funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con el sistema B = Número total de funciones que han sido probadas	
				Dónde: B > 0	


Continúa 

	Adaptabilidad en entorno organizacional.	Número funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con usuarios del entorno empresarial y contar el número total de funciones las cuales han sido probadas	$X = A/B$  A = Número de funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con usuarios del entorno empresarial B = Número total de funciones que han sido probadas  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0, es el mejor
Capacidad para ser instalado	Eficiencia en el tiempo de instalación.	Contar el tiempo total transcurrido al instalar el sistema y contar el número de reintentos al instalar el sistema	$X = A/T$  A = Número de reintentos al instalar el sistema T = Tiempo total transcurrido al instalar el sistema	$X = A/T$ El más lejano a 0/t es el mejor

Continua 



	Facilidad de instalación.	Contar el número casos en que los usuarios tuvieron éxito al instalar el sistema cambiando proceso de instalación para su conveniencia y contar el número total de casos en que los usuarios han intentado cambiar el proceso de instalación para su conveniencia	$X = A/B$  A = Número casos en que los usuarios tuvieron éxito al instalar el sistema cambiando proceso de instalación para su conveniencia B = Número total de casos en que los usuarios han intentado cambiar el proceso de instalación para su conveniencia	$0 \leq X \leq 1$ El más cercano a 1, es el mejor
Capacidad para ser reemplazado	Consistencia en la función de soporte al usuario.	Contar el número de nuevas funciones que son consideradas como no consistentes por el usuario y contar el número de nuevas funciones	$X = A/B$  A = Número de nuevas funciones que son consideradas como no consistentes por el usuario B = Número de nuevas funciones  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0, es el mejor

Continua 

---


Inclusividad funcional.	Contar el número de funciones que producen resultados similares con anterioridad y que no se han exigido cambios y contar el número de funciones probadas que son similares a las funciones proporcionadas por otro software para ser reemplazado	$X = A/B$  A = Número de funciones que producen resultados similares con anterioridad y que no se han exigido cambios B = Número de funciones probadas que son similares a las funciones proporcionadas por otro software para ser reemplazado	$0 \leq X \leq 1$ El más cercano a 1, es el mejor
Uso continuo de datos.	Contar el número de datos que son continuamente utilizables por el software a ser reemplazado y contar el número de datos que son continuamente reutilizables por el software a ser reemplazado	$X = A/B$  A = número de datos que son continuamente solo utilizables por el software a ser reemplazado B = Número de datos que son reutilizables por el software a ser reemplazado  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor

---


Las métricas para la calidad en uso, permiten evaluar las características. Las métricas para la calidad en uso se describen en la siguiente tabla:


**Tabla 7**


*Métricas de Calidad en Uso*


Características	Sub-características	Métricas	Método de aplicación	Fórmula	Valor deseado
<b>Efectividad</b>	Efectividad	Compleitud de la tarea.	Contar el número de tareas completadas y el número total de tareas intentadas	$X = A/B$ A= Número de tareas completadas B = Número total de tareas intentadas  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, el mejor
		Efectividad de la tarea.	Tomar el valor proporcional de cada componente faltante o incorrecto en la salida de la tarea	$X = A/B$ A=Cantidad de objetivos completados por la tarea. B=Cantidad de objetivos planeados que realice la tarea	$0 \leq X \leq 1$ El más cercano a 1, el mejor  Continua 


		Frecuencia de error.	Contar el número de errores cometidos por los usuarios y contar el número de tareas	$X = A/B$ A = Número de errores cometidos por los usuarios B = Número de tareas  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0 es el mejor
<b>Eficiencia</b>	Eficiencia	Tiempo de la tarea.	Tomar el tiempo planeado y el tiempo actual	$X = A/B$ A = Tiempo actual B = Tiempo planeado  Dónde: $A > 0$	$0 \leq X \leq 1$ Si $A \leq B$ el más cercano a 0 es lo mejor. Si $A > B$ será considerado como el peor caso
		Tiempo relativo de la tarea.	Tomar el tiempo que completa una tarea un usuario normal y el tiempo que completa una tarea un usuario experto	$X = A/B$ A = Tiempo que completa una tarea un usuario experto B = Tiempo que completa una tarea un usuario normal  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, el mejor

Continúa 

				$X = A/T$ A = Número de tareas efectivas T = Tiempo de la tarea Dónde: $T > 0$	$X = A/T$ El más lejano a 0/t es el mejor
				$X = A/B$ A = Número de tareas eficientes realizadas por un usuario ordinario B = Número de tareas eficientes planeadas Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es lo mejor
				$X = A/B$ A = Tiempo de la tarea B = Tiempo de productividad. Dónde: $B > 0$	$0 \leq X \leq 1$ Si $A \leq B$ el más cercano a 0 es lo mejor. Si $A > B$ será considerado como el peor caso Continúa 


		Numero relativo de las acciones del usuario.	Contar el número de acciones realizadas por los usuarios y contar el número de acciones necesarias actualmente	$X = A/B$ A = Número de acciones realizadas por los usuarios B = Número de acciones necesarias actualmente  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es lo mejor
<b>Satisfacción</b>	Utilidad	Nivel de satisfacción.	Realizar un cuestionario sobre el nivel de satisfacción sobre el sistema.	$X = A/B$ A= Numero de preguntas con respuesta satisfactorias B = Número total de preguntas realizadas en el cuestionario.  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es lo mejor  Continua 


		Uso discrecional de las funciones.	Observación de uso	$X = A/B$ A = Número de funciones específicas del software que se utilizan B = Número total de funciones que están destinados a ser usados  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, mejor
		Porcentaje de quejas de los clientes.	Contar el número de clientes que se quejan y contar el número total de clientes	$X = A/B$ A = Número de clientes que se quejan B = Número total de clientes  Dónde: $A > 0$	$0 \leq X \leq 1$ El más cercano a 0, mejor
<b>Libertad de riesgo</b>	Libertad del riesgo económico	Retorno de la Inversión (ROI).	Consultar los beneficios obtenidos y el capital invertido	$X = A / B$ A= Beneficios obtenidos B =Beneficios esperados.  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es lo mejor  Continúa 


				$X = A/B$ A = Tiempo real para lograr el ROI B = Tiempo aceptable para lograr el ROI Dónde: $B > 0$	$0 \leq X \leq 1$ Si $A \leq B$ el más cercano a 0 es lo mejor. Si $A > B$ será considerado como el peor caso
				$X = B/A$ A = Monto de inversión de TI o las ventas planeadas de la empresa para la comparación B = Monto real de la inversión de TI o de las ventas de la empresa Dónde: $B > 0$	$0 \leq X \leq 1$ Si $B \leq A$ el más cercano a 1 es lo mejor. Si $B > A$ será considerado como el mejor caso
				$X = A/B$ A = Resultado del BSC B = BSC planeado Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, mejor.  Continúa 



				$X = A/B$ A = Tiempo de entrega planeado o retrasos en las entregas B = Tiempo de entrega actual o retrasos en las entregas Dónde: $B > 0$	$0 \leq X \leq 1$ Si $A \leq B$ el más cercano a 0 es lo mejor. Si $A > B$ será considerado como el peor caso
				$X = A/B$ A = Ingresos reales de un cliente B = Ingresos planeados de un cliente Dónde: $B > 0$	$0 \leq X \leq 1$ Si $A \leq B$ el más cercano a 1 es lo mejor. Si $A > B$ será considerado como el mejor caso
				$X = A/B$ A = Número de errores con consecuencias económicas B = Número total de situaciones de uso Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0 es el mejor

Continua 

		Corrupción del software.	Contar el número de ocurrencias de corrupción del software y contar número total de situaciones de uso.	$X = A/B$ A= Número de ocurrencias de corrupción del software B = Número total de situaciones de uso  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0 es el mejor
Libertad del riesgo de salud y seguridad		Frecuencia de problemas en la salud y seguridad del usuario.	Contar el número de usuarios que notificaron problemas de salud y contar el número total de usuarios	$X = A/B$ A = Número de usuarios que notificaron problemas de salud B= Número total de usuarios  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0 es el mejor
		Impacto en la salud y seguridad del usuario.	Contar el número de personas afectadas, tomar el tiempo y el grado de importancia	$X = A/T$ A = Número de personas afectadas T = Tiempo	$0 \leq X \leq 5$ El más cercano a 0 es el mejor  Continua 

		Seguridad de las personas afectadas por el uso del sistema.	Contar el número de personas puestas en peligro y contar el número total de personas potencialmente afectadas por el sistema	$X = A/B$ A = Número de personas puestas en peligro B = Número total de personas potencialmente afectadas por el sistema  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0 es el mejor
	Libertad del riesgo ambiental	Impacto Ambiental.	Estimar el impacto ambiental y el impacto ambiental aceptable	$X = A/B$ A = Impacto ambiental aceptable B = Impacto ambiental real  Dónde: $B > 0$	$0 \leq X \leq 1$ Si $A \leq B$ el más cercano a 0 es lo mejor. Si $A > B$ será considerado como el peor caso
<b>Cobertura de Contexto</b>	Complejidad de Contexto	Complejidad de contexto.	Contar el número de contextos con la facilidad de uso inaceptable y el número total de distintos contextos de uso	$X = A/B$ A= Número de distintos contextos de uso inaceptables B = Número total de distintos contextos de uso  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0 es el mejor  Continúa 

	Flexibilidad	Función flexible del diseño.	Contar el número de características diseñadas con completa flexibilidad y contar el número total de características de diseño	$X = A/B$ A= Número de características diseñadas con completa flexibilidad B = Número total de características de diseño  Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor
--	--------------	------------------------------	---	--	--

### 3.1.1.4 Ponderaciones

Las ponderaciones dependerán de tres factores, en primer lugar del nivel de importancia de cada una de las características, en segundo lugar del tipo de software y en tercer lugar del criterio del evaluador, la sumatoria de la ponderación para cada característica debe ser menor o igual a 100 guiados por la fórmula que se define en la siguientes tablas:

**Tabla 8**

*Variable para fórmulas*

Variable	Descripción
W	Sub-característica
X	Característica
Y	Ponderación de Característica
Z	Total

**Tabla 9**

*Fórmulas de cálculo de calidad*

Nº	Fórmula
1	$X = \left( \sum W \right) * Y$
2	$Z = \left( \sum X \right)$

El resultado del cálculo de las métricas deberá ser convertido sobre 100 con la finalidad de encontrar el resultado final.

### 3.1.1.5 Escala de calidad

Una vez que se hayan realizado todos los cálculos y se haya obtenido el resultado se podrá determinar la calidad de acuerdo a la siguiente escala:

**Tabla 10**

#### *Escalas de Calidad*

Escala de medición	Nivel	Descripción
80,00 – 100,00	Nivel 4	Muy Satisfactorio
60,00 – 79,99	Nivel 3	Satisfactorio
40,00 – 59,99	Nivel 2	Aceptable
20,00 – 39,99	Nivel 1	Poco Aceptable
0 – 19,00	Nivel 0	Inaceptable

### 3.1.1.6 Etapas

En la presente metodología se utilizarán las fases propuestas por la norma ISO/IEC 25040, y utilizará la metodología Bootstrap en el sentido de considerar entradas y salida para cada una de las etapas.

- **ETAPA 1: ESTABLECER REQUISITOS DE EVALUACIÓN**

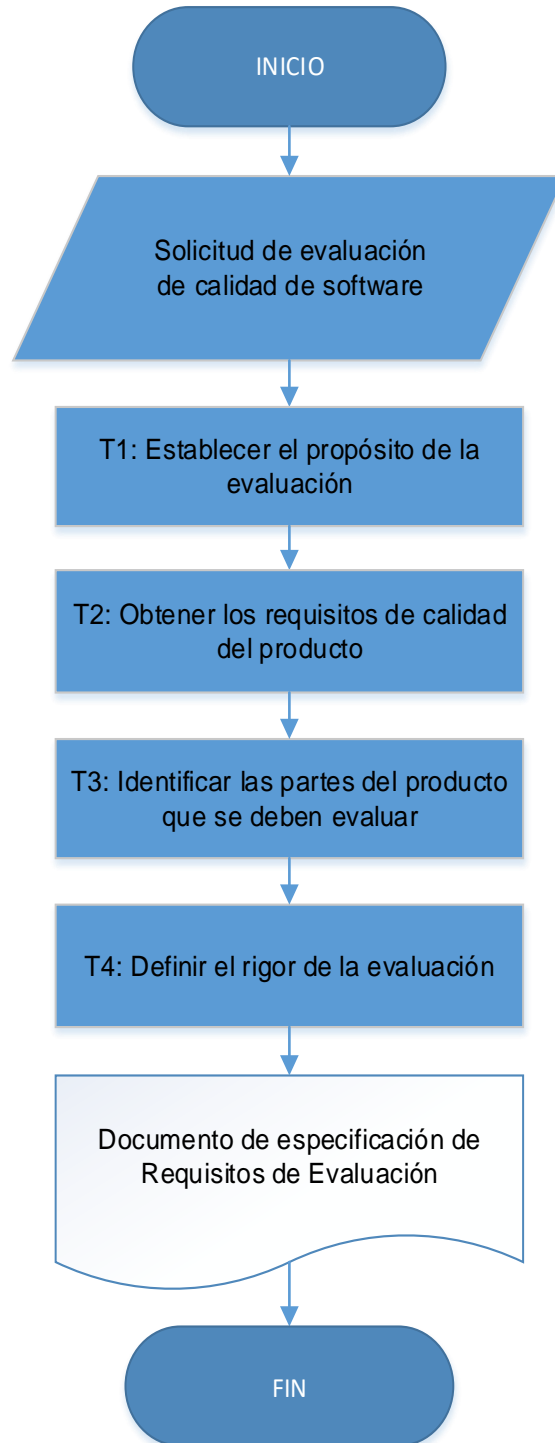
**Tabla 11**

#### *Etapas 1*

<b>Descripción</b>	<b>Definir los requisitos que se van a considerar en la evaluación. En esta fase se debe definir qué componentes se evaluarán, y las características y sub-características de calidad, además de definir las métricas que se van a considerar en la evaluación.</b>
<b>Entrada</b>	Solicitud de evaluación de calidad de software, documento en donde debe

Continua 

	constar los motivos que llevan a realizar la evaluación. (Formato de Solicitud de Evaluación ver Anexo 1)
<b>Salida</b>	Documento de especificación de Requisitos de Evaluación
<b>Tareas</b>	
<b>Tarea 1</b>	<p>Establecer el propósito de la evaluación:</p> <p>En esta tarea se documenta el propósito por el que la organización quiere evaluar la calidad de su producto software.</p>
<b>Tarea 2</b>	<p>Obtener los requisitos de calidad del producto:</p> <p>En esta tarea se identifican las partes interesadas en el producto software (desarrolladores, posibles adquirientes, usuarios, niveles de usuarios, proveedores, etc.) y se especifican los requisitos de calidad (características y sub-características) del producto utilizando un determinado modelo de calidad.</p>
<b>Tarea 3</b>	<p>Identificar las partes del producto que se deben evaluar:</p> <p>Se deben identificar y documentar las partes del producto software incluidas en la evaluación. El tipo de producto a evaluar (especificación de requisitos, diagramas de diseño, documentación de las pruebas, etc.) depende de la fase en el ciclo de vida en que se realiza la evaluación y del propósito de ésta.</p>
<b>Tarea 4</b>	<p>Definir el rigor de la evaluación:</p> <p>Se debe definir el rigor de la evaluación en función del propósito y el uso previsto del producto software, basándose, por ejemplo, en aspectos como el riesgo para la seguridad, el riesgo económico o el riesgo ambiental. En función del rigor se podrá establecer qué técnicas se aplican y qué resultados se esperan de la evaluación.</p>
<b>Recolección de información</b>	La información necesaria para el cumplimiento de la Etapa 1 se recabará en una reunión que se desarrollará entre el ente evaluado y el evaluador donde tendrá como resultado el documento de especificación de Requisitos de Evaluación (ver Anexo 2).



**Figura 6** Etapa 1

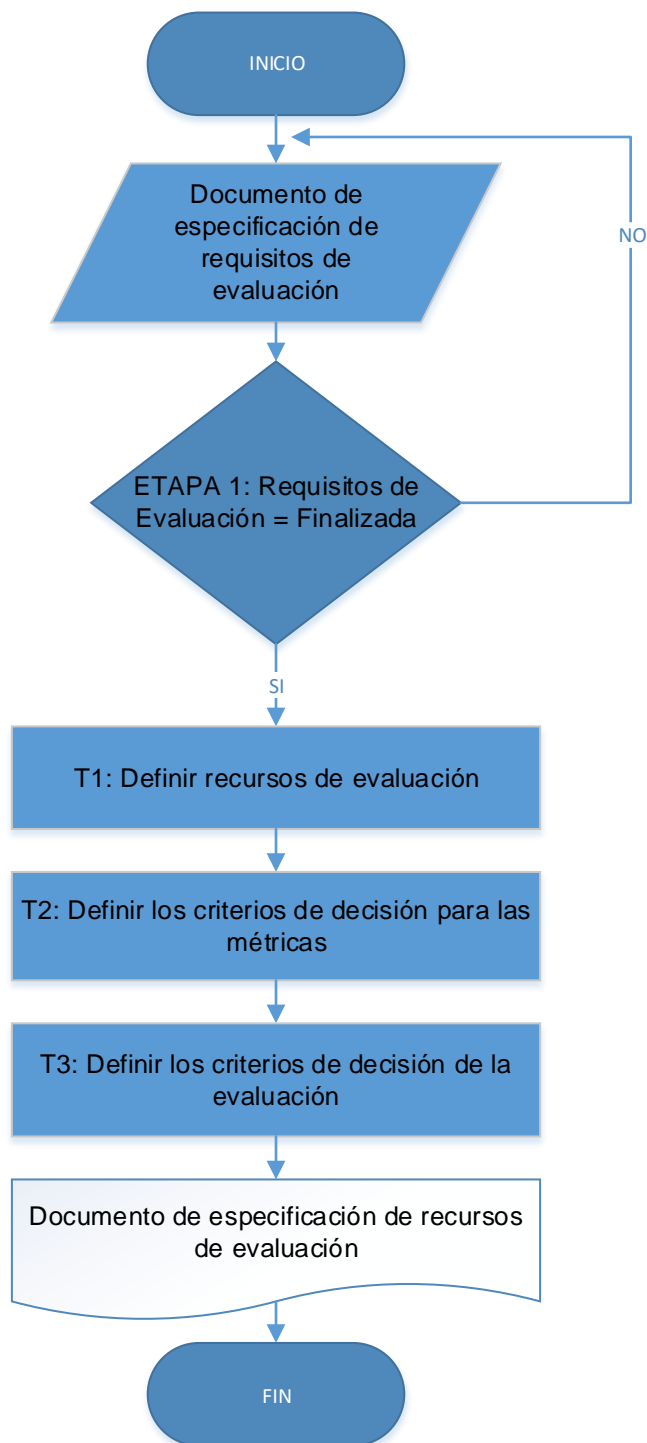


- **ETAPA 2: ESPECIFICACIÓN DE LA EVALUACIÓN**

**Tabla 12***Etapa 2*

<b>Descripción</b>	<b>Especificar las medidas de calidad, los métodos de evaluación a utilizar y los criterios de decisión para los requisitos definidos en la fase 1.</b>
<b>Precondición</b>	Que se haya finalizado la Etapa 1 con el documento de especificación de Requisitos de Evaluación
<b>Entrada</b>	Documento de especificación de requisitos de evaluación (salida de la etapa 1)
<b>Salida</b>	Documento de especificación de recursos de evaluación
<b>Tareas</b>	
<b>Tarea 1</b>	Definir recursos de evaluación: En esta tarea el evaluador selecciona las métricas de calidad, técnicas y herramientas (módulos de evaluación) que cubran todos los requisitos de la evaluación. Dichas métricas deben permitir que, en función de su valor, se puedan realizar comparaciones fiables con criterios que permitan tomar decisiones.
<b>Tarea 2</b>	Definir los criterios de decisión para las métricas: Se deben definir los criterios de decisión para las métricas seleccionadas. Dichos criterios son umbrales numéricos que se pueden relacionar con los requisitos de calidad y posteriormente con los criterios de evaluación para decidir la calidad del producto.
<b>Tarea 3</b>	Definir los criterios de decisión de la evaluación: Se deben definir criterios para las diferentes características evaluadas a partir de las sub-características y métricas de calidad. Estos resultados a mayor nivel de abstracción permiten realizar la valoración de la calidad del producto software de forma general.
<b>Recolección de información</b>	En esta etapa el ente evaluador definirá las características, sub-características y métricas, niveles de usuarios participantes, de acuerdo a los requisitos de evaluación de la calidad y la solicitud, en donde constan las características principales del producto de software a evaluar, tendrá como resultado el documento de especificación de recursos de evaluación (ver Anexo 3).

**Fuente:** (Norma ISO/IEC 25000, 2017)



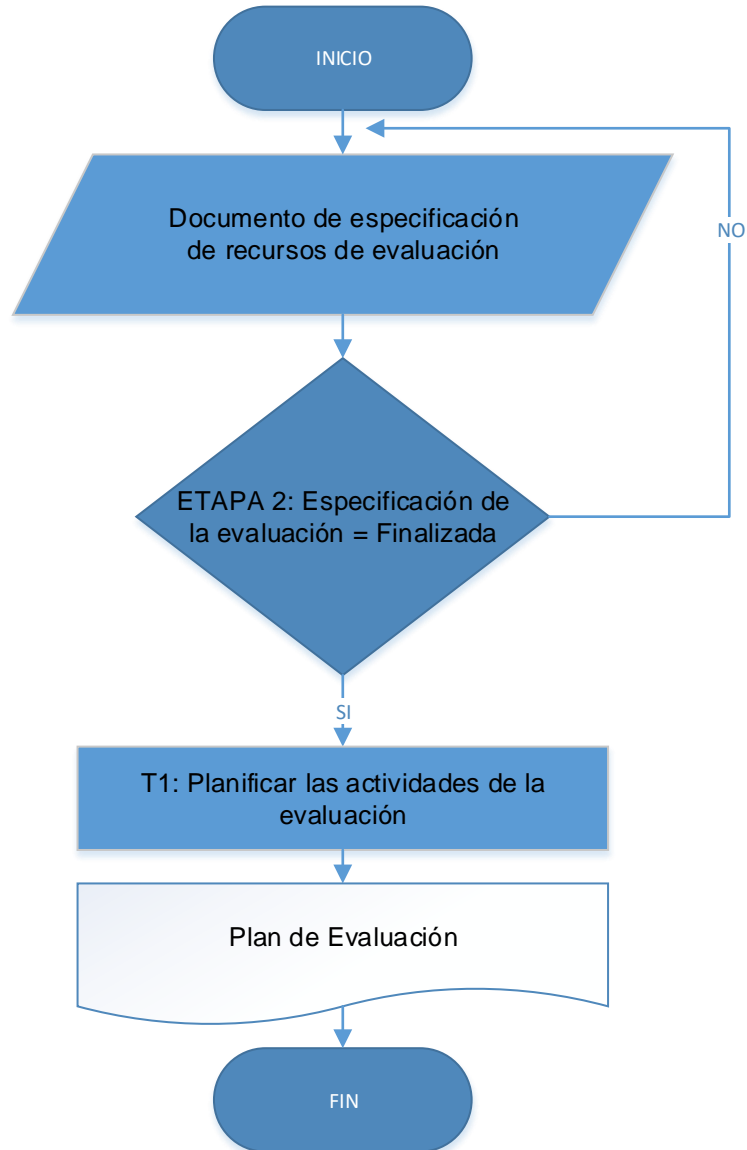
**Figura 7** Etapa 2

- **ETAPA 3: DISEÑO DE LA EVALUACIÓN**

**Tabla 13***Etapa 3*

<b>Descripción</b>	<b>Planificar las actividades de la evaluación de calidad.</b>
<b>Precondición</b>	Que se haya finalizado la Etapa 2 con el documento de especificación de recursos de evaluación
<b>Entrada</b>	Documento de especificación de requisitos de evaluación y documento de especificación de recursos de evaluación (salidas de la etapa 1 y 2)
<b>Salida</b>	Plan de Evaluación
<b>Tareas</b>	
<b>Tarea 1</b>	Planificar las actividades de la evaluación: Se deben planificar las actividades de la evaluación teniendo en cuenta la disponibilidad de los recursos, tanto humanos como materiales, que puedan ser necesarios. En la planificación se debe tener en cuenta el presupuesto, los métodos de evaluación y estándares adaptados, las herramientas de evaluación, etc. El plan de evaluación se revisará y actualizará proporcionando información adicional según sea necesario durante el proceso de evaluación.
<b>Recolección de información</b>	En esta etapa el evaluador definirá los tiempos y recursos necesarios para el desarrollo de la Evaluación de la calidad del software, proporcionará como resultado el Plan de Evaluación (ver Anexo 4).

**Fuente:** (Norma ISO/IEC 25000, 2017)



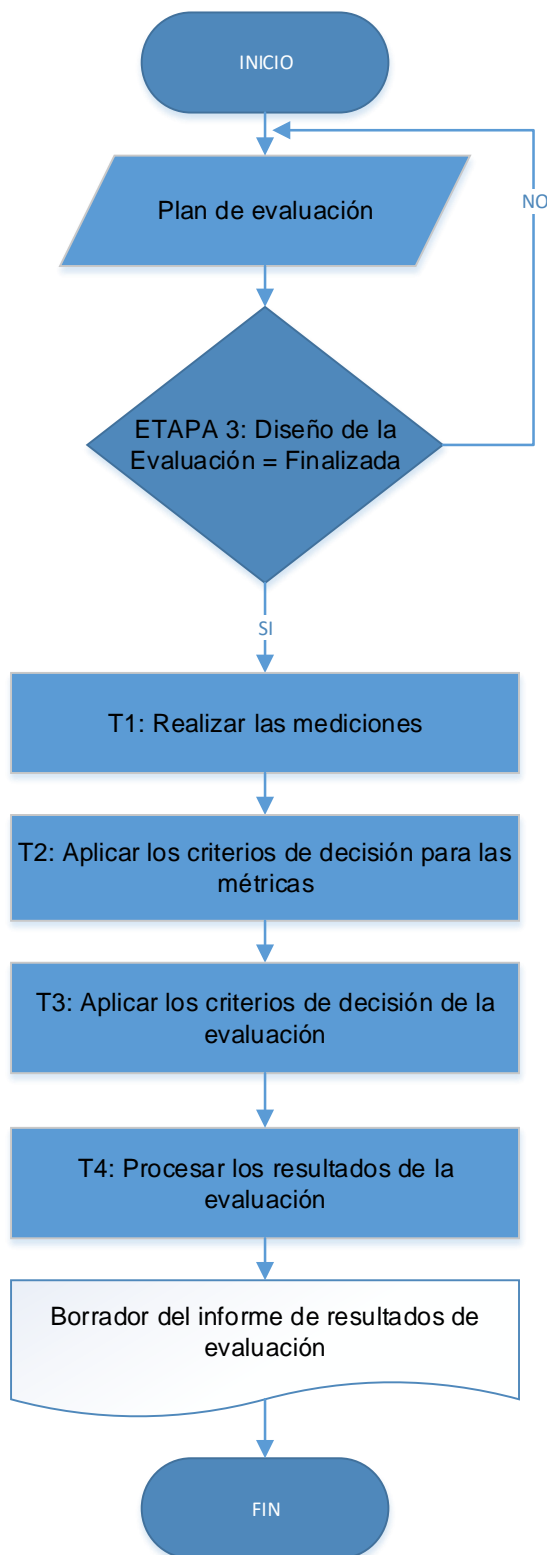
**Figura 8** Etapa 3

- **ETAPA 4: EJECUCIÓN DE LA EVALUACIÓN**

**Tabla 14***Etapa 4*

<b>Descripción</b>	<b>En esta actividad se ejecutan las actividades planificadas en la etapa 3.</b>
<b>Precondición</b>	Que se haya finalizado la Etapa 3 con el documento plan de evaluación
<b>Entrada</b>	Plan de evaluación (salida de la etapa 3)
<b>Salida</b>	Borrador del informe de resultados de evaluación
<b>Tareas</b>	
<b>Tarea 1</b>	Realizar las mediciones: Se deben realizar las mediciones sobre el producto software y sus componentes para obtener los valores de las métricas seleccionadas e indicadas en el plan de evaluación. Todos los resultados obtenidos deberán ser debidamente registrados.
<b>Tarea 2</b>	Aplicar los criterios de decisión para las métricas: Se aplican los criterios de decisión para las métricas seleccionadas sobre los valores obtenidos en la medición del producto.
<b>Tarea 3</b>	Aplicar los criterios de decisión de la evaluación: En esta tarea se deben aplicar los criterios de decisión a nivel de características y sub-características de calidad, produciendo como resultado la valoración del grado en que el producto software cumple los requisitos de calidad establecidos.
<b>Tarea 4</b>	Procesar los resultados de la evaluación: En esta tarea se tiene que realizar el procesamiento y análisis de toda la información recabada en el proceso de evaluación y se tiene que elaborar un borrador del informe de evaluación, el mismo que será sujeto a revisión para su posterior aprobación.
<b>Recolección de información</b>	En esta etapa el evaluador ejecutará

**Fuente:** (Norma ISO/IEC 25000, 2017)



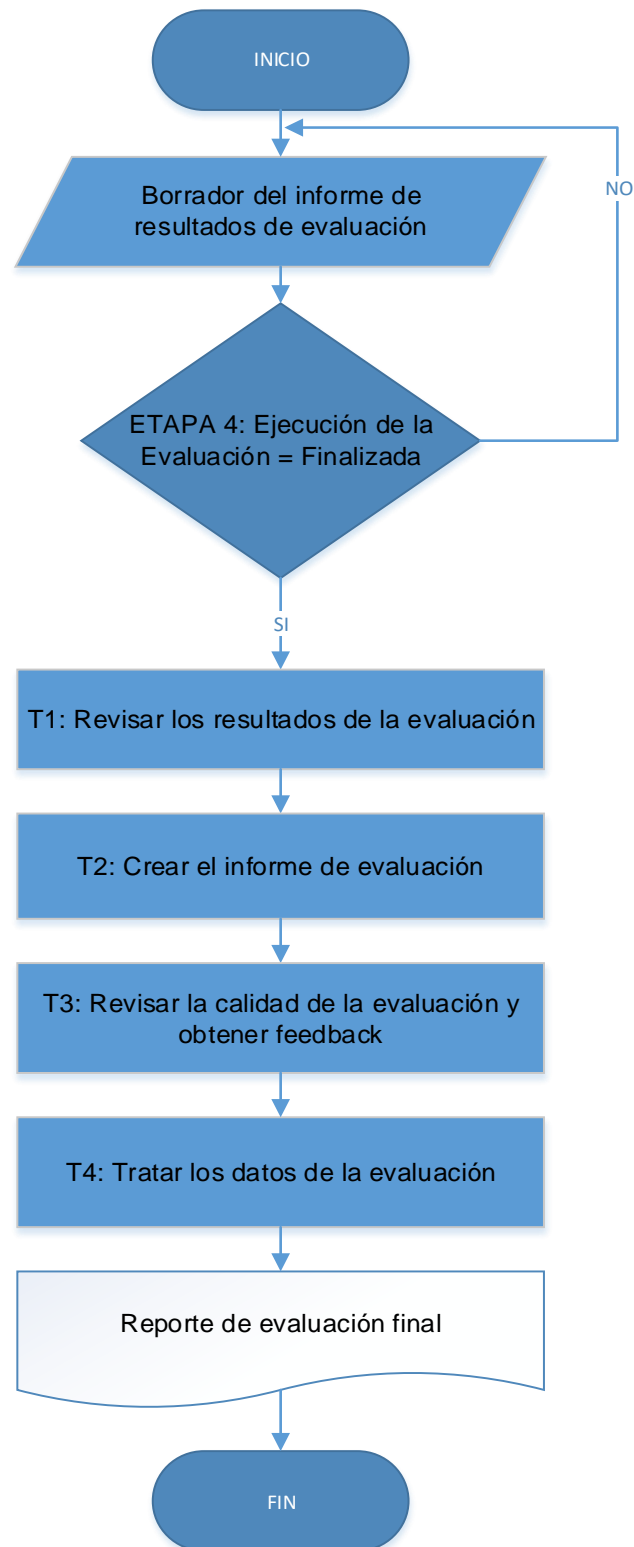
**Figura 9** Etapa 4

- **ETAPA 5: CONCLUSIÓN Y RESULTADOS DE LA EVALUACIÓN**

**Tabla 15***Etapa 5*

<b>Descripción</b>	<b>Revisar y presentar el informe final de la evaluación de calidad.</b>
<b>Precondición</b>	Que se haya finalizado la Etapa 4 con el documento borrador del informe de resultados de evaluación.
<b>Entrada</b>	Borrador del informe de resultados de evaluación (salida de la etapa 5)
<b>Salida</b>	Reporte de evaluación final (anexo 5)
<b>Tareas</b>	
<b>Tarea 1</b>	Revisar los resultados de la evaluación: Mediante esta tarea, el evaluador y el cliente de la evaluación (en caso de existir) realizan una revisión conjunta de los resultados obtenidos, con el objetivo de realizar una mejor interpretación de la evaluación y una mejor detección de errores.
<b>Tarea 2</b>	Crear el informe de evaluación: Una vez revisados los resultados, se elabora el informe de evaluación, con los requisitos de la evaluación, los resultados, las limitaciones y restricciones, el personal evaluador, etc.
<b>Tarea 3</b>	Revisar la calidad de la evaluación y obtener feedback: El evaluador revisará los resultados de la evaluación y la validez del proceso de evaluación, de los indicadores y de las métricas aplicadas. El feedback de la revisión debe servir para mejorar el proceso de evaluación de la organización y las técnicas de evaluación utilizadas.
<b>Tarea 4</b>	Tratar los datos de la evaluación: Una vez finalizada la evaluación, el evaluador debe realizar el adecuado tratamiento con los datos y los objetos de la evaluación según lo acordado con el cliente (en caso de ser una tercera parte), devolviéndolos, archivándolos o eliminándolos según corresponda.

**Fuente:** (Norma ISO/IEC 25000, 2017)



**Figura 10** Etapa 5



### 3.1.2 Planteamiento de la metodología

La metodología propuesta se basa en la necesidad de las organizaciones de disponer de una herramienta genérica para la evaluación de la calidad del software, con la finalidad de que se conozca a ciencia cierta que el sistema informático cumpla con los requisitos funcionales, no funcionales, y las exigencias mínimas de calidad, además que permita identificar las características comunes y sub-características, atributos y métricas propios del caso de estudio que se está evaluando. Para el desarrollo de la presente metodología se han considerado lineamientos establecidos en la Norma ISO/IEC 25040, así como en el método Bootstrap, y el objetivo es que basados en datos cuantificables se pueda establecer el nivel de calidad que tiene el sistema y también localizar y proporcionar recomendaciones para subsanar el mayor número de deficiencias lo antes posible.

La metodología evaluará la calidad de un producto de software que se encuentra en producción con un tiempo considerable, en donde los usuarios que van a ser entrevistados tengan conocimiento pleno del sistema, la metodología toma las 5 etapas de la norma ISO/IEC 25040 y se presentan a continuación:



**Figura 11** Etapas de metodología

En la ejecución de la evaluación se van a identificar dos partes involucradas, por un lado, el evaluador y por otro el evaluado, en el siguiente cuadro se describen los requisitos que deben cumplir cada una de las partes para que la evaluación se lleve de manera exitosa.

**Tabla 16**

*Perfiles*

PERFIL	DESCRIPCIÓN	REQUISITOS A CUMPLIR	RESPONSABILIDADES
<b>Evaluador</b>	Persona, personas o entidad preparada profesionalmente para aplicar y cumplir con el plan de evaluación de manera objetiva, debe conocer a cabalidad la metodología de evaluación utilizada.	<ul style="list-style-type: none"> <li>• Objetividad</li> <li>• Persona independiente</li> <li>• Responsabilidad</li> </ul>	<ul style="list-style-type: none"> <li>• Ejecutar el plan de evaluación</li> <li>• Entregar informe final de evaluación a la entidad evaluada</li> </ul>
<b>Evaluados</b>	Persona o personas que manejan el producto de software y serán encargados de	<ul style="list-style-type: none"> <li>• Colaborativa</li> <li>• Sincera</li> </ul>	<ul style="list-style-type: none"> <li>• Colaborar en la entrega de información veraz y oportuna</li> </ul>

Continua

---

proporcionar información  
veraz al evaluador.

- Tomar acciones correctivas basadas en resultados.
- 

Se validará la efectividad de la metodología en un sistema con las siguientes características:

- Aplicación web,
- Uso masivo (aproximadamente 3000 usuarios),
- Diferentes niveles de usuarios (administrador, tipo usuario 1, tipo usuario 2, etc.),
- Elaborado a la medida de las necesidades institucionales,
- Multiplataforma.

## **CAPÍTULO IV**

### **VALIDACIÓN DE LA METODOLOGÍA A TRAVÉS DE UN CASO DE ESTUDIO**

En el desarrollo de este capítulo se aplicará la metodología propuesta con la finalidad de obtener el nivel de calidad de Sistema de Gestión Académica (gestión notas y matrículas) de la ESPAM MFL.

#### **4.1 *Desarrollo del caso de estudio***

##### **4.1.1 *Objetivo***

Conocer el nivel de calidad del Sistema de Gestión Académica (Gestión notas y matrículas) de la ESPAM MFL a través de la metodología propuesta.

##### **4.1.2 *Sistema de Gestión Académica (gestión notas y matrículas)***

La Unidad de Producción de Software es un departamento que depende de la Carrera de Computación de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, esta unidad ha implementado varios sistemas para sustentar la gestión de los diferentes procesos institucionales, uno de estos sistemas y quizás uno de los más importante es el Sistema de Gestión Académica (gestión notas y matrículas), el cual será sujeto a evaluación en el presente caso de estudio, cuyo objetivo es medir la calidad de acuerdo a las características y sub-características propuestas en esta metodología

### 4.1.3 Proceso de Evaluación

La evaluación a la calidad al Sistema de Gestión Académica (gestión notas y matrículas), se concibió como una evaluación interna solicitada por el Coordinador de la Unidad de Producción de Software, con la finalidad de conocer la calidad externa y de uso de dicha aplicación, antes de iniciar con la evaluación, en una reunión sostenida con el ente evaluado, se procedió a llenar el documento “Formato de Solicitud de evaluación”, que se adjunta en el Anexo 1, con el cual, una vez que fue aprobado por el Coordinador, se dio inicio al proceso.

#### 4.1.3.1 Etapa 1 Establecer requisitos de evaluación

Para iniciar el proceso de evaluación, se realizó una entrevista al Coordinador de la Unidad de Desarrollo de Software, en donde se realizaron las preguntas que se detallan en la tabla 17, con la finalidad de obtener información referente al sistema a evaluar y para definir los requisitos de evaluación.

**Tabla 17**

*Entrevista para obtener información*

<b>Pregunta 1</b>	¿Cuál es el nombre y la versión del sistema a evaluar?
<b>Pregunta 2</b>	Realice una descripción del sistema a evaluar
<b>Pregunta 3</b>	¿Cuáles son las principales características del sistema a evaluar?
<b>Pregunta 4</b>	Mencione los usuarios del sistema
<b>Pregunta 5</b>	Realice una descripción del detalle técnico del sistema a evaluar
<b>Pregunta 6</b>	¿Cuáles son los requisitos funcionales del sistema a evaluar?

Una vez obtenida información preliminar, se rellenó el “Formato de Especificación de Requisitos de Evaluación” (anexo 2), en donde según la percepción del ente evaluado se definieron cuáles son las características y sub-características con mayor nivel de importancia según el área de actividad del sistema, que en el presente caso de estudio es un sistema a la mitad y se definió el nivel de importancia acuerdo a la escala que se detalla a continuación:

**Tabla 18**

*Tabla de nivel de importancia de características y sub-características*

<b>Nivel de Importancia</b>	<b>Simbología</b>	<b>Porcentaje referencial del nivel de importancia</b>	<b>Significado</b>
Alto	A	70% - 100%	El grado de importancia de la característica y sub-característica es alto por ende se realizará las mediciones
Medio	M	25% - 69%	La característica y sub-característica no es tan relevante, pero puede o no ser medida dependiendo del criterio del evaluador
Bajo	B	1% - 24%	La característica y sub-característica no tiene relevancia y no será medida.
No aplica	NA	0%	Este valor se dará a la característica y sub-característica que no se pueden medir dependiendo de diferentes factores

Una vez aprobado el documento de especificación requisitos de calidad por el evaluador y por el evaluado y se continuó con la siguiente fase.

#### **4.1.3.2 Etapa 2 Especificación de la Evaluación**

La etapa dos es una fase en la que evaluador con toda la información recabada procedió a definir todas aquellas características, sub-características, métricas, informantes de calidad que serán utilizados en la evaluación, para ello fue necesario que defina el “Formato especificación de recursos de evaluación” (anexo 3). Además, se diseñaron las encuestas pertinentes para cada tipo de usuario (anexo 6 y 7).

Una vez que se conocieron los elementos a ser evaluados, se seleccionó la población y se detalla a continuación:

- Docentes
- Estudiantes
- Departamento UPS

A partir de aquí se realizó un muestro con la población de docentes y estudiantes a través de la fórmula que se presente en la siguiente imagen, para poder utilizar dicha fórmula fue necesario conocer el tamaño de la población.

**Tabla 19**

*Fórmula de muestreo*


Fórmula de muestreo
$n = \frac{N \times Z_a^2 \times p \times q}{d^2 \times (N - 1) + Z_a^2 \times p \times q}$

Los resultados de la aplicación de la fórmula fueron los siguientes:

**Tabla 20**

*Muestreo*

ESTUDIANTES	
Tamaño de la población (N)	2263
Error Muestral (d)	0,07
Proporción de Éxito (P)	50%
Proporción de Fracaso (Q)	50%
Valor para Confianza (Z) (1)	1,65
Tamaño de muestra fórmula	131
	124
Tamaño de muestra óptimo	

Continua 

<b>DOCENTES</b>	
<b>Tamaño de la población (N)</b>	170
<b>Error muestral</b>	0,07
<b>Proporción de Éxito (P)</b>	50%
<b>Proporción de Fracaso (Q)</b>	50%
<b>Valor para Confianza (Z) (1)</b>	1,65
<b>Tamaño de muestra fórmula</b>	76
<b>Tamaño de muestra óptimo</b>	53

#### **4.1.3.3 Etapa 3 Diseño de la Evaluación**

Cuando se definieron los recursos de evaluación y se conoció de antemano el producto a evaluar se procedió a planificar, estableciendo tiempos, recursos, riesgos, entre otros elementos importantes para la ejecución de la evaluación de la calidad del producto de software, para ellos se completó el formato denominado “Plan de Evaluación” (anexo 4). Fue necesario realizar ajustes a la planificación a medida que avanzaba la evaluación, con la finalidad de proporcionar información adicional.

#### **4.1.3.4 Etapa 4 Ejecución de la Evaluación**

En la etapa número 4 se aplicaron las encuestas a la muestra de los usuarios escogidos para evaluar, se realizaron las pruebas pertinentes por parte de los evaluadores, siguiendo cada uno de los elementos planificados, se realizaron las mediciones sobre el sistema, todos los resultados fueron debidamente registrados. Además, se calcularon los valores de las métricas seleccionadas y se determinaron las ponderaciones. Esta etapa tuvo como salida el borrador del Informe de evaluación.



#### 4.1.3.4.1 Análisis del cuestionario de satisfacción a docentes y estudiantes

El cuestionario fue aplicado a 140 estudiantes y 48 docentes de las diferentes carreras de la ESPA M MFL de los cuales se obtuvieron los siguientes resultados:

- **PREGUNTA 1 DOCENTES CALIFIQUE SU EXPERIENCIA EN EL PROCESO DE ELABORACIÓN DE RÚBRICAS**

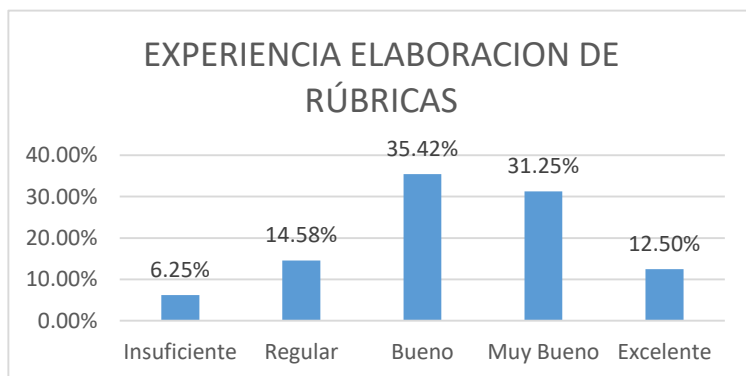
Fueron evaluados 48 docentes los cuales calificaron de la siguiente manera:

**Tabla 21**

*Docentes calificación de experiencias de elaboración de rúbricas*

Calificación	F. Absoluta	F. Relativa
Insuficiente	3	6,25%
Regular	7	14,58%
Bueno	17	35,42%
Muy Bueno	15	31,25%
Excelente	6	12,50%
<b>TOTAL</b>	<b>48</b>	<b>100%</b>

En esta pregunta se obtuvo una calificación 3,29 sobre 5, es decir el 65,83% de la calificación.



**Figura 12** Experiencia elaboración de rúbricas

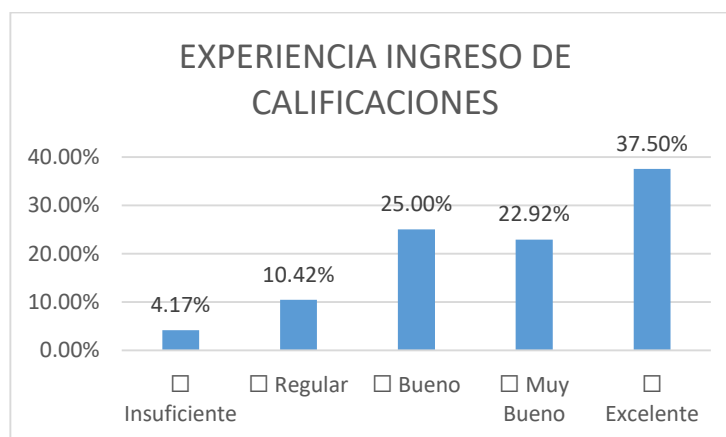
- **PREGUNTA 2 DOCENTES CALIFIQUE SU EXPERIENCIA EN EL PROCESO DE INGRESO DE CALIFICACIONES**

**Tabla 22**

*Calificación de experiencia en el proceso de ingreso de calificaciones*

Calificación	F. Absoluta	F. Relativa
Insuficiente	2	4,17%
Regular	5	10,42%
Bueno	12	25,00%
Muy Bueno	11	22,92%
Excelente	18	37,50%
<b>TOTAL</b>	<b>48</b>	<b>100%</b>

En esta pregunta se obtuvo una calificación 3,79 sobre 5, es decir el 75,83% de la calificación.



**Figura 13** Experiencia ingreso de calificaciones

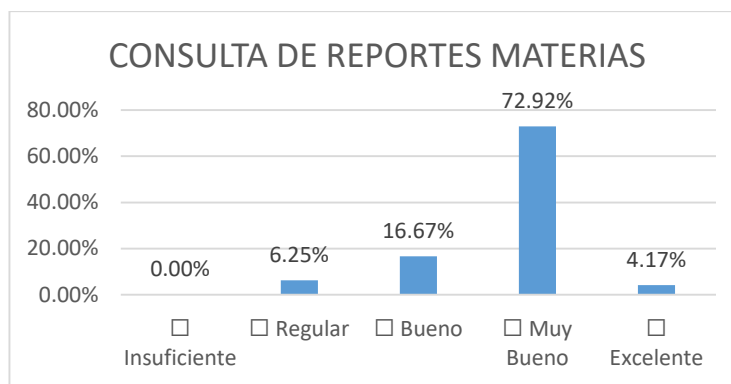
- **PREGUNTA 3 DOCENTES CALIFIQUE SU EXPERIENCIA EN PROCESO DE CONSULTA DE REPORTES DE MATERIAS**

**Tabla 23**

*Calificación de experiencia en proceso de consulta de reportes de materia*

Calificación	F. Absoluta	F. Relativa
Insuficiente	0	0,00%
Regular	3	6,25%
Bueno	8	16,67%
Muy Bueno	35	72,92%
Excelente	2	4,17%
<b>TOTAL</b>	<b>48</b>	<b>100,00%</b>

En esta pregunta se obtuvo una calificación 3,75 sobre 5, es decir el 75,00% de la calificación.



**Figura 14** Calificación reporte de materias

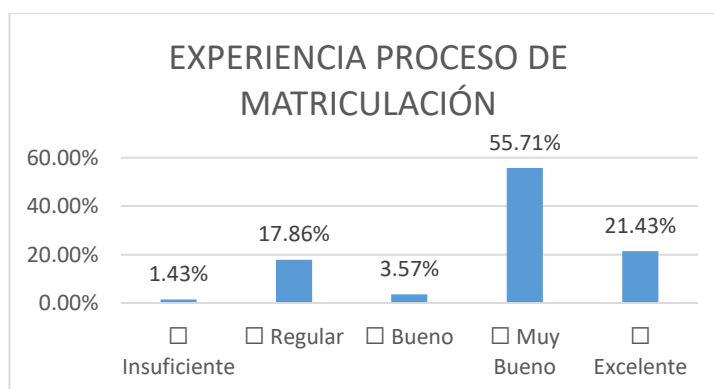
- **PREGUNTA 1 ESTUDIANTES CALIFIQUE SU EXPERIENCIA EN EL PROCESO DE MATRICULACIÓN**

**Tabla 24**

*Calificación de experiencia en el proceso de matriculación*

Calificación	F. Absoluta	F. Relativa
Insuficiente	2	1,43%
Regular	25	17,86%
Bueno	5	3,57%
Muy Bueno	78	55,71%
Excelente	30	21,43%
<b>TOTAL</b>	<b>140</b>	<b>100,00%</b>

En esta pregunta se obtuvo una calificación 3,78 sobre 5, es decir el 75,57% de la calificación.



**Figura 15** Experiencia proceso de matriculación

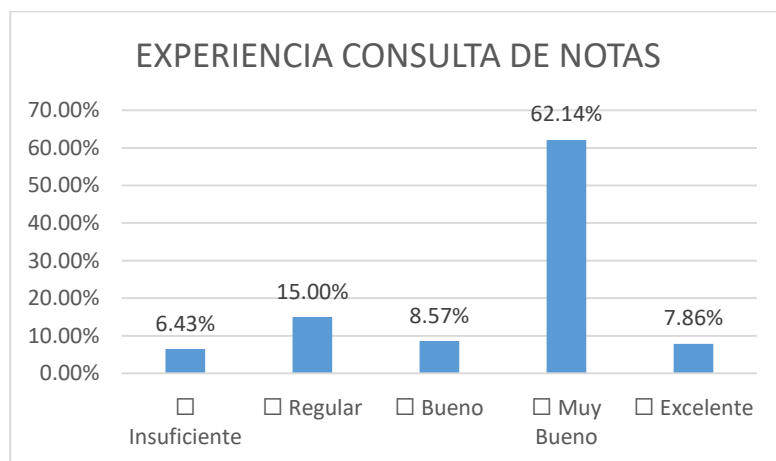
- **PREGUNTA 2 ESTUDIANTES CALIFIQUE SU EXPERIENCIA EN EL PROCESO DE CONSULTA DE NOTAS**

**Tabla 25**

*Calificación de experiencia en el proceso de consulta de notas*

Calificación	F. Absoluta	F. Relativa
Insuficiente	9	6,43%
Regular	21	15,00%
Bueno	12	8,57%
Muy Bueno	87	62,14%
Excelente	11	7,86%
<b>TOTAL</b>	<b>140</b>	<b>100,00%</b>

En esta pregunta se obtuvo una calificación 3,50 sobre 5, es decir el 70,00% de la calificación.



**Figura 16** Experiencia Consulta de notas

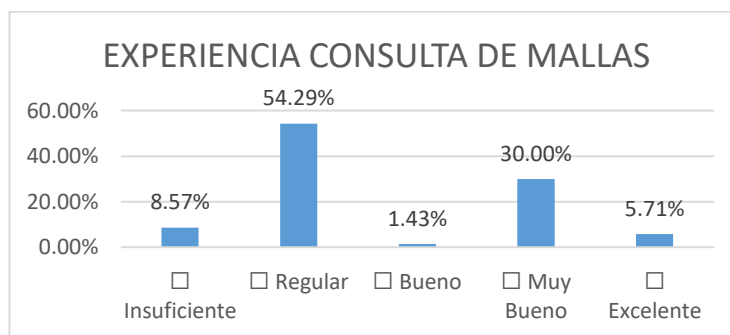
- **PREGUNTA 3 ESTUDIANTES CALIFIQUE SU EXPERIENCIA EN PROCESO DE CONSULTA DE MALLA**

**Tabla 26**

*Califique su experiencia en procesos de consulta de malla*

Calificación	F. Absoluta	F. Relativa
Insuficiente	12	8,57%
Regular	76	54,29%
Bueno	2	1,43%
Muy Bueno	42	30,00%
Excelente	8	5,71%
<b>TOTAL</b>	<b>140</b>	<b>100,00%</b>

En esta pregunta se obtuvo una calificación 2,70 sobre 5, es decir el 54,00% de la calificación.



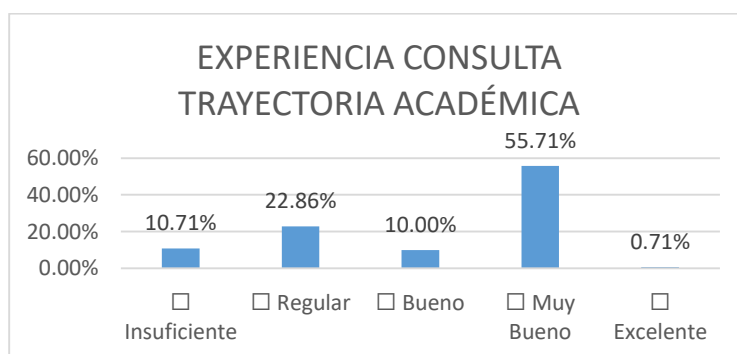
**Figura 17** Experiencia consulta de mallas

- **PREGUNTA 4 ESTUDIANTES CALIFIQUE SU EXPERIENCIA EN EL PROCESO DE CONSULTA DE TRAYECTORIA ACADÉMICA**

**Tabla 27***Consulta trayectoria académica*

Calificación	F. Absoluta	F. Relativa
Insuficiente	15	10,71%
Regular	32	22,86%
Bueno	14	10,00%
Muy Bueno	78	55,71%
Excelente	1	0,71%
<b>TOTAL</b>	<b>140</b>	<b>100,00%</b>

En esta pregunta se obtuvo una calificación 3,13 sobre 5, es decir el 62,57% de la calificación.

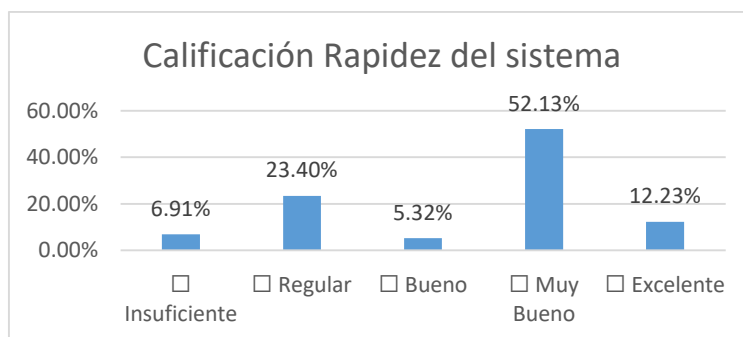
**Figura 18** Experiencia consulta trayectoria académica

- **PREGUNTA: CALIFIQUE LA RAPIDEZ DEL SISTEMA**

**Tabla 28***Califique la rapidez del sistema*

Calificación	F. Absoluta	F. Relativa
Insuficiente	13	6,91%
Regular	44	23,40%
Bueno	10	5,32%
Muy Bueno	98	52,13%
Excelente	23	12,23%
<b>TOTAL</b>	<b>188</b>	<b>100,00%</b>

En esta pregunta se obtuvo una calificación 3,39 sobre 5, es decir el 67,87% de la calificación.



**Figura 19** Califique rapidez del sistema

- **PREGUNTA: CALIFIQUE LA FACILIDAD DE USO DE LA APLICACIÓN**

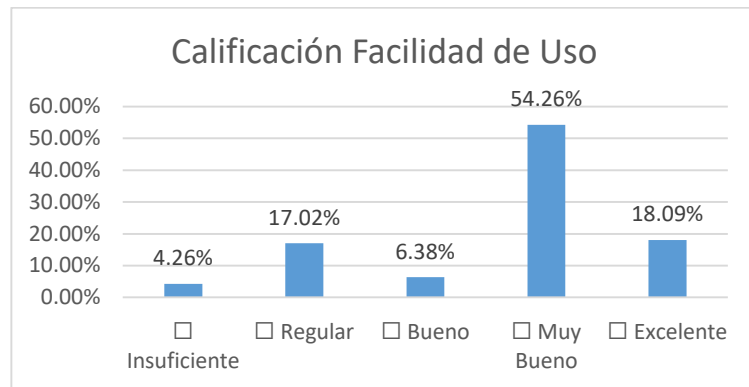
**Tabla 29**

*Califique la facilidad de uso de la aplicación*

Calificación	F. Absoluta	F. Relativa
Insuficiente	8	4,26%
Regular	32	17,02%
Bueno	12	6,38%
Muy Bueno	102	54,26%
Excelente	34	18,09%
<b>TOTAL</b>	<b>188</b>	<b>100,00%</b>

En esta pregunta se obtuvo una calificación 3,65 sobre 5, es decir el 62,98% de la calificación.





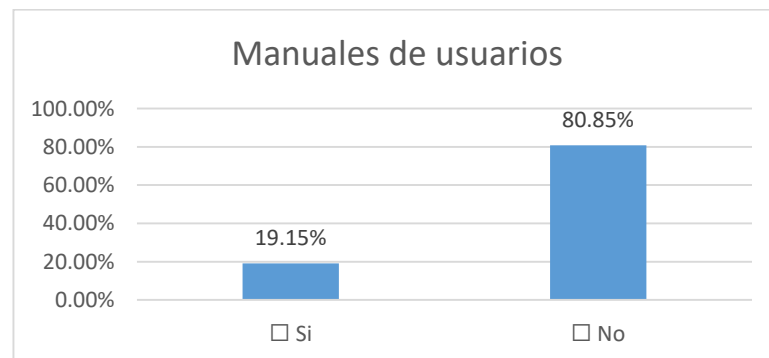
**Figura 20** Calificación Calidad de uso

- **PREGUNTA: ¿CONOCE USTED SI EXISTEN MANUALES DE USUARIO, TUTORIALES, VIDEOS, ENTRE OTRAS HERRAMIENTAS QUE LE AYUDEN A CONOCER EL FUNCIONAMIENTO DEL SISTEMA?**

**Tabla 30**

*Conocimiento de Manuales de usuarios*

Calificación	F. Absoluta	F. Relativa
<b>Si</b>	36	19,15%
<b>No</b>	152	80,85%
<b>TOTAL</b>	188	100,00%



**Figura 21** Conocimiento Manuales de usuarios

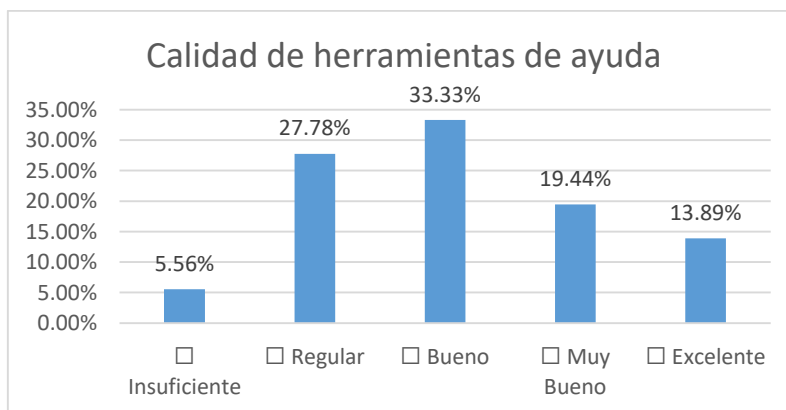
- **PREGUNTA: SI LA RESPUESTA ANTERIOR ES SI, CALIFIQUE LA CALIDAD DE ESTAS HERRAMIENTAS**

**Tabla 31**

*Calificación de herramientas de ayuda*

Calificación	F. Absoluta	F. Relativa
Insuficiente	2	5,56%
Regular	10	27,78%
Bueno	12	33,33%
Muy Bueno	7	19,44%
Excelente	5	13,89%
<b>TOTAL</b>	<b>36</b>	<b>100,00%</b>

En esta pregunta se obtuvo una calificación 3,08 sobre 5, es decir el 61,67% de la calificación.



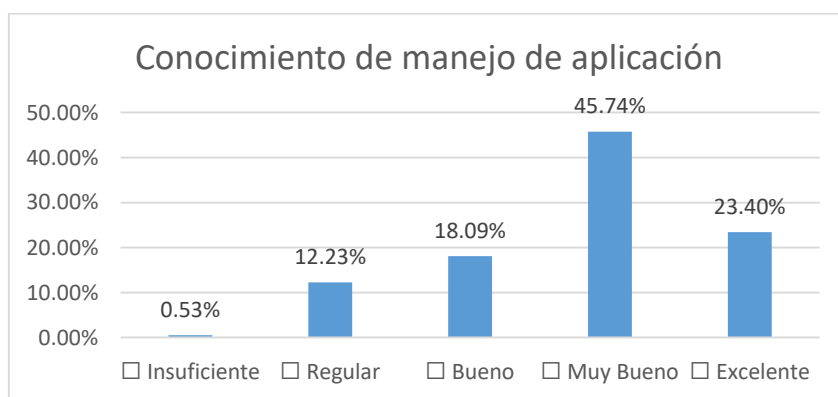
**Figura 22** Calidad de herramientas de ayuda

- **PREGUNTA: CALIFIQUE SU CONOCIMIENTO ACERCA DEL MANEJO DE LA APLICACIÓN**

**Tabla 32***Conocimiento acerca del manejo de la aplicación*

Calificación	F. Absoluta	F. Relativa
Insuficiente	2	5,56%
Regular	10	27,78%
Bueno	12	33,33%
Muy Bueno	7	19,44%
Excelente	5	13,89%
<b>TOTAL</b>	<b>36</b>	<b>100,00%</b>

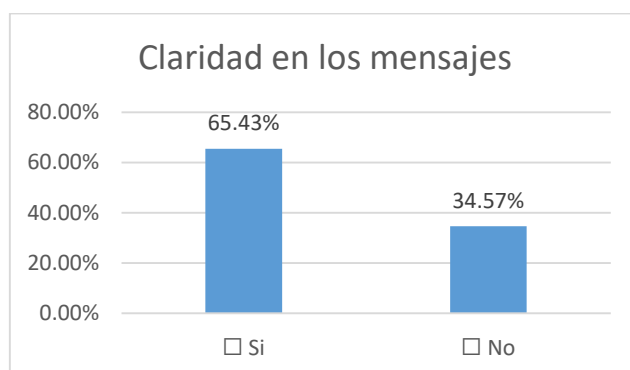
En esta pregunta se obtuvo una calificación 3,79 sobre 5, es decir el 75,85% de la calificación.

**Figura 23** Conocimiento de manejo de aplicación

- **PREGUNTA: ¿EL SISTEMA LE PROPORCIONA MENSAJES CLAROS Y PERTINENTES A LA FUNCIÓN QUE ESTÁ REALIZANDO?**

**Tabla 33***Mensajes claros*

Calificación	F. Absoluta	F. Relativa
<b>Si</b>	65	34,57%
<b>No</b>	123	65,43%
<b>TOTAL</b>	188	100,00%

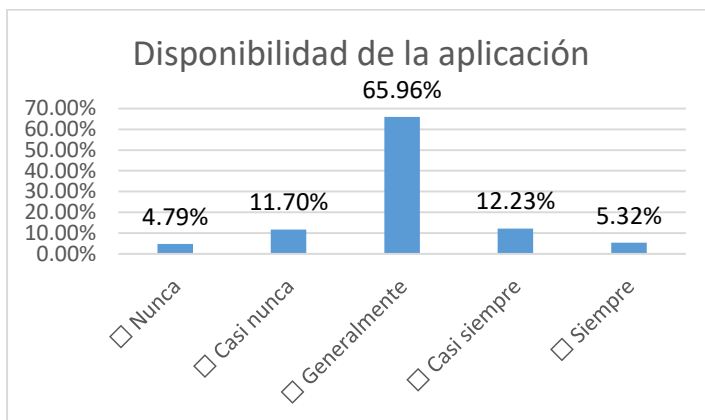
**Figura 24** Claridad en los mensajes

- **PREGUNTA: ¿LA APLICACIÓN ESTÁ DISPONIBLE CADA VEZ QUE LA NECESITA?**

**Tabla 34***Disponibilidad de la aplicación*

Calificación	F. Absoluta	F. Relativa
<b>Nunca</b>	9	4,79%
<b>Casi Nunca</b>	22	11,70%
<b>Generalmente</b>	124	65,96%
<b>Casi siempre</b>	23	12,23%
<b>Siempre</b>	10	5,32%
<b>TOTAL</b>	188	100,00%

En esta pregunta se obtuvo una calificación 3,02 sobre 5, es decir el 60,32% de la calificación.



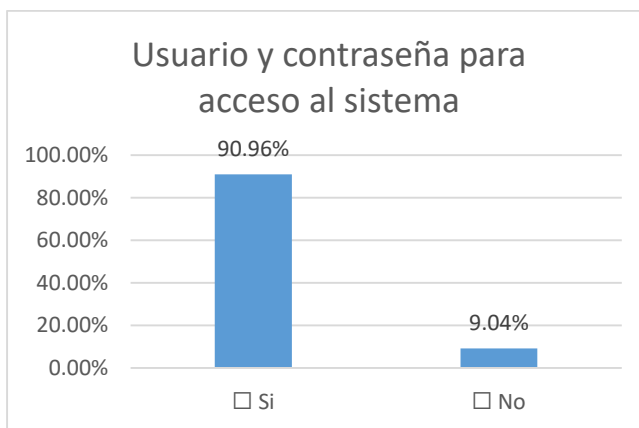
**Figura 25** Disponibilidad de la aplicación

- **PREGUNTA: ¿CUENTA CON USUARIO Y CONTRASEÑA PARA EL ACCESO AL SISTEMA?**

**Tabla 35**

*Cuenta con usuario y contraseña*

Calificación	F. Absoluta	F. Relativa
Si	171	90,96%
No	17	9,04%
<b>TOTAL</b>	<b>188</b>	<b>100,00%</b>



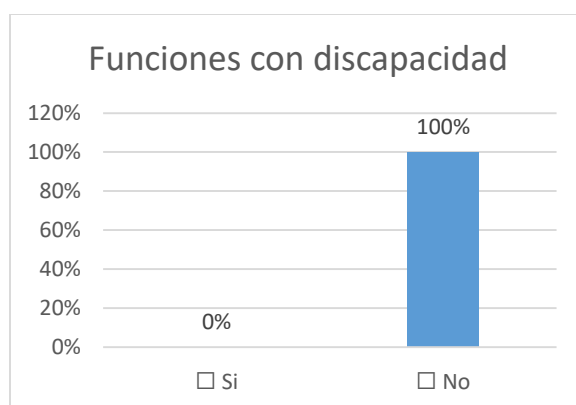
**Figura 26** Cuenta con usuario y contraseña

- **PREGUNTA: ¿CONOCE USTED SI EL SISTEMA CUENTA CON FUNCIONES PARA PERSONAS CON DISCAPACIDAD?**

**Tabla 36**

*Conoce usted si el sistema cuenta con funciones para discapacitados*

Calificación	F. Absoluta	F. Relativa
Si	0	0
No	188	100,00%
<b>TOTAL</b>	188	100,00%



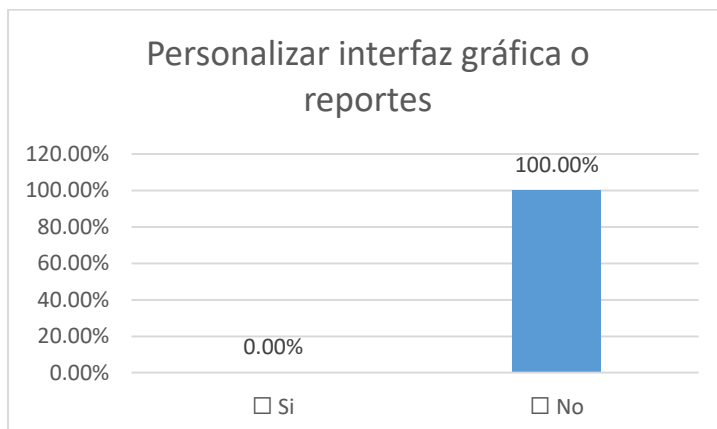
**Figura 27** Funciones de personas con discapacidad

- **PREGUNTA: ¿PUEDE PERSONALIZAR LA INTERFAZ GRÁFICA, REPORTE O CONSULTAS DE ACUERDO A SUS NECESIDADES O PREFERENCIAS?**

**Tabla 37**

*Personalización interfaz gráfica*

Calificación	F. Absoluta	F. Relativa
Si	0	0,00%
No	188	100,00%
<b>TOTAL</b>	188	100,00%



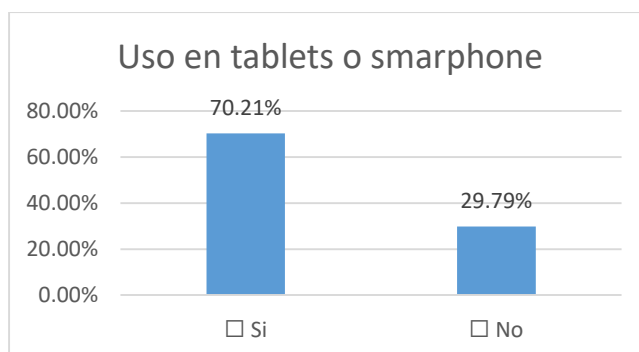
**Figura 28** Personalización interfaz gráfica

- **PREGUNTA: ¿PUEDE USAR EL SISTEMA EN TABLETS, SMARTPHONE O SIMILARES?**

**Tabla 38**

*Uso en tablets, smartphones o similares*

Calificación	F. Absoluta	F. Relativa
Si	132	70,21%
No	56	29,79%
<b>TOTAL</b>	<b>188</b>	<b>100,00%</b>



**Figura 29** Uso en tablets, smartphones o similares

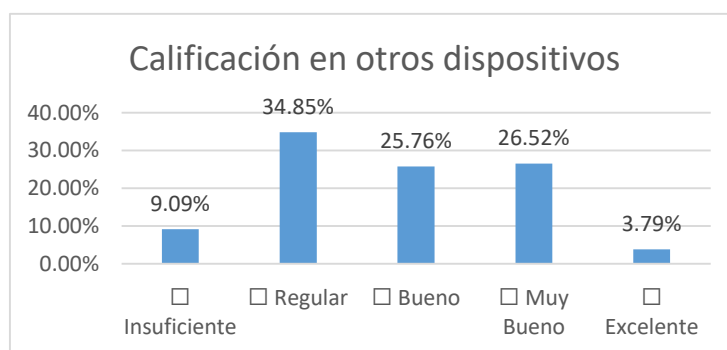
- **PREGUNTA: SI LA RESPUESTA ANTERIOR ES SI, CALIFIQUE LA EXPERIENCIA EN ESTOS DISPOSITIVOS**

**Tabla 39**

*Si la respuesta anterior es Si, califique la experiencia en estos dispositivos*

Calificación	F. Absoluta	F. Relativa
Insuficiente	12	9,09%
Regular	46	34,85%
Bueno	34	25,76%
Muy Bueno	35	26,52%
Excelente	5	3,79%
<b>TOTAL</b>	<b>132</b>	<b>100,00%</b>

En esta pregunta se obtuvo una calificación 2,81 sobre 5, es decir el 56,21% de la calificación.



**Figura 30** Calificación en otros dispositivos

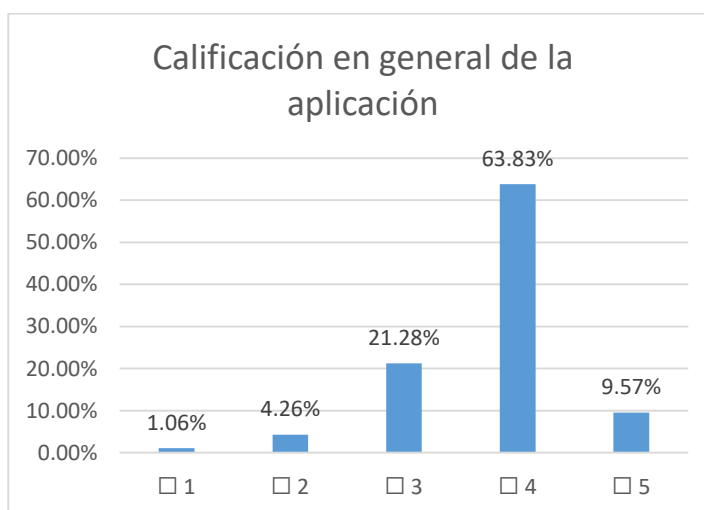
- **PREGUNTA: ¿QUÉ CALIFICACIÓN EN GENERAL LE DARÍA AL SISTEMA DE GESTIÓN NOTAS Y MATRÍCULAS?**



**Tabla 40***Calificación en General*

Calificación	F. Absoluta	F. Relativa
Insuficiente	2	1,06%
Regular	8	4,26%
Bueno	40	21,28%
Muy Bueno	120	63,83%
Excelente	18	9,57%
<b>TOTAL</b>	<b>188</b>	<b>100,00%</b>

En esta pregunta se obtuvo una calificación 3,77 sobre 5, es decir el 75,32% de la calificación.

**Figura 31** Calificación General

- **ANÁLISIS GENERAL DE RESULTADOS DEL CUESTIONARIO DE SATISFACCIÓN**

**Tabla 41***Análisis general de resultados*

<b>TIPO DE PREGUNTA</b>	<b>PREGUNTA</b>	<b>CALIFICACIÓN</b>
<b>DOCENTE</b>	Califique su experiencia en el proceso de elaboración de rúbricas	65,83
	Califique su experiencia en el proceso de ingreso de calificaciones	75,83
	Califique su experiencia en proceso de consulta de reportes de materias	75,00
<b>ESTUDIANTE</b>	Califique su experiencia en el proceso de matriculación	75,57
	Califique su experiencia en el proceso de consulta de notas	70,00
	Califique su experiencia en proceso de consulta de malla	54,00
	Califique su experiencia en el proceso de consulta de trayectoria académica	62,57
<b>PREGUNTAS EN GENERAL</b>	Califique la rapidez del sistema	67,87
	Califique la facilidad de uso de la aplicación	62,98
	Califique la calidad de estas herramientas	61,67
	Califique su conocimiento acerca del manejo de la aplicación	75,85
	¿El sistema le proporciona mensajes claros y pertinentes a la función que está realizando?	72,34
	¿La aplicación está disponible cada vez que la necesita?	60,32
	¿Cuenta con usuario y contraseña para el acceso al sistema?	92,77
	¿Conoce usted si el sistema cuenta con funciones para personas con discapacidad?	0,00
	¿Puede personalizar la interfaz gráfica, reportes o consultas de acuerdo a sus necesidades o preferencias?	0,00
	Califique la experiencia en estos dispositivos (tablets, smartphones o similares)	56,21
¿Qué calificación en general le daría al Sistema de Gestión Notas y Matrículas?	75,32	

Mediante la aplicación de este cuestionario a estudiantes y docentes, al ser usuarios masivos se determinó que el nivel de satisfacción es de 67,38.

#### 4.1.3.4.2 Cálculos de datos obtenidos


En este apartado se realizará el cálculo de las métricas en base a la información obtenida en cada uno de las sub-características y se detallan en la siguiente tabla:

- **CALIDAD EXTERNA**


**Tabla 42**

*Cálculo de datos Calidad Externa*

Subcaracterísticas	Métricas	Datos de medición	Información obtenida	Resultado
<b>Adecuación Funcional</b>				
<b>Compleitud Funcional</b>	Compleitud de la implementación funcional.	A=Número de funciones implementadas correctamente (procesos)	35 funciones	83,33
		B=Número de funciones solicitadas en los requerimientos	42 funciones	
<b>Corrección Funcional</b>	Precisión computacional	Número de cálculos inexactos encontrados en 15 minutos	0 Cálculos inexactos	100,00
<b>Pertinencia Funcional</b>	Cumplimiento de requerimientos	Total de requerimientos entregados	19 requerimientos	76,00
		Total de requerimientos solicitados	25 requerimientos	
<b>Eficencia de desempeño</b>				
<b>Comportamiento temporal</b>	Tiempo de respuesta	Tiempo de respuesta promedio desde que se envía una petición, hasta obtener una respuesta (probar con 10 peticiones)	- Tiempo de envío de petición 13h58:35 - Tiempo de recepción de petición 13h58:40 5 segundos	100,00
	Tiempo de espera	Tiempo cuando se inicia un trabajo y el tiempo en completar el trabajo (probar con 10 procesos)	-Tiempo cuando se inicia 14h02 - Tiempo cuando termina 14h09	90,00

Continua 

	Rendimiento	Número de tareas completadas en 60 minutos	8 tareas completadas	80,00
<b>Compatibilidad</b>				
<b>Co-existencia</b>	Co – existencia disponible	Número de sistemas con el cual esté enlazado el sistema	5 sistemas	71,43
		Número total de sistemas con el que necesita enlazarse	7 sistemas	
<b>Interoperabilidad</b>	Conectividad con sistemas externos.	Número de funciones enlazadas con sistemas externos	25 funciones	71,43
		Número de total de funciones	35 funciones	
	Capacidad de intercambiar de datos.	Número de datos intercambiados correctamente	3545 datos	100,00
	Número total de datos	3545 datos		
<b>Usabilidad</b>				
	Funciones evidentes.	Total de funciones intuitivas	25	71,43
<b>Capacidad de aprendizaje</b>	Efectividad de la documentación del usuario o ayuda del sistema.	Manuales de usuarios, videos o similares	Existe documentación, pero no está disponible en la página web o en la aplicación	55,83
<b>Operatividad</b>	Claridad de mensajes	Mensajes implementados con información clara y pertinente	-85 mensajes implementados claros - 90 mensajes	83,39
	Posibilidad de personalización.	Funciones con posibilidad de personalización	0 funciones	0,00
	Accesibilidad Técnica	Acceso para personas con discapacidad	0 funciones	0,00
<b>Fiabilidad</b>				
<b>Disponibilidad</b>	Tiempo medio de inactividad	Tiempo promedio de inactividad después de un fallo	120 segundos	75,00

Continua 

<b>Seguridad</b>				
<b>Confidencialidad</b>	Encriptación de datos.	Número de datos encriptados	1 tipo de dato encriptado	100,00
		Total de datos que necesitas encriptación	1 tipo	
Autenticidad	Métodos de autenticación	Total de métodos de autenticación	1 usuario y contraseña	50,00

- **CALIDAD DE USO**

Subcaracterísticas	Métricas	Datos de medición	Información obtenida	Resultado	
<b>Efectividad</b>					
<b>Efectividad</b>	Complejidad de la tarea.	Número de tareas completadas	15 tareas	100,00	
		Número totas de tareas intentadas	15 tareas		
<b>Efectividad</b>	Efectividad de la tarea.	Objetivos completados por la tarea	3 objetivos completados por la tarea de matriculación	100,00	
		Objetivos que debe cumplir	3 objetivos		
<b>Satisfacción</b>					
<b>Utilidad</b>	Nivel de Satisfacción	de	Análisis de encuesta de Usuarios	de	67,38
			satisfacción		


#### 4.1.3.4.3 Ponderaciones

Los evaluadores basados en el tipo de software y en la importancia dada por el ente evaluado a cada una de las características y sub-características determinó las ponderaciones que se muestran a continuación:

- **CALIDAD EXTERNA**

**Tabla 43***Calificación de Calidad Externa*

Sub-características	Métricas	Ponderación	Calificación	Calificación Final
<b>Adecuación Funcional</b>		<b>20%</b>		<b>17,97</b>
<b>Compleitud Funcional</b>	Compleitud de la implementación funcional.	25%	83,33	20,83
<b>Corrección Funcional</b>	Precisión computacional	50%	100,00	50,00
<b>Pertinencia Funcional</b>	Cumplimiento de requerimientos	25%	76,00	19,00
<b>Eficiencia de desempeño</b>		<b>30%</b>		<b>24,30</b>
<b>Comportamiento temporal</b>	Tiempo de respuesta	30%	100,00	30,00
	Tiempo de espera	30%	90,00	27,00
	Rendimiento	30%	80,00	24,00
<b>Compatibilidad</b>		<b>10%</b>		<b>8,14</b>
<b>Co-existencia</b>	Co – existencia disponible	30%	71,43	21,43
<b>Interoperabilidad</b>	Conectividad con sistemas externos.	35%	71,43	25,00
	Capacidad de intercambiar de datos.	35%	100,00	35,00
<b>Usabilidad</b>		<b>15%</b>		<b>8,02</b>
<b>Capacidad de aprendizaje</b>	Funciones evidentes.	30%	71,43	21,43
	Efectividad de la documentación del usuario o ayuda del sistema.	20%	55,83	11,17
<b>Operatividad</b>	Claridad de mensajes	25%	83,39	20,85
	Posibilidad de personalización.	15%	0,00	0,00
	Accesibilidad Técnica	10%	0,00	0,00
<b>Fiabilidad</b>		<b>15%</b>		<b>11,25</b>
<b>Disponibilidad</b>	Tiempo medio de inactividad	100%	75,00	75,00

Continua 

<b>Seguridad</b>		<b>10%</b>		<b>7,50</b>
<b>Confidencialidad</b>	Encriptación de datos.	50%	100,00	50,00
<b>Autenticidad</b>	Métodos de autenticación	50%	50,00	25,00
			<b>Total</b>	<b>77,18</b>

- **CALIDAD DE USO**

**Tabla 44**

*Calificación de Calidad Externa*

Subcaracterísticas	Métricas	Ponderación	Calificación	Calificación final
<b>Efectividad</b>		<b>40%</b>		40,00
<b>Efectividad</b>	Compleitud de la tarea.	50%	100,00	50,00
	Efectividad de la tarea.	50%	100,00	50,00
<b>Satisfacción</b>		<b>60%</b>		40,43
<b>Utilidad</b>	Nivel de Satisfacción	100%	67,38	67,38
			<b>Total</b>	<b>80,43</b>

#### **4.1.3.4.4 Calificación final**

Para calcular la calificación final se realizó una ponderación entre el total de la Calidad Externa y la Calidad en Uso, los evaluadores consideraron las siguientes ponderaciones:

**Tabla 45**

*Calificación Final*

Tipo de Calidad	Ponderación	Calificación	Calificación Final
Externa	60%	77,18	46,31
En uso	40%	80,43	32,17
		<b>Total</b>	<b>78,48</b>

#### **4.1.3.5 Etapa 5: Conclusión y resultados de la evaluación**

Luego de realizar cada uno de los cálculos y seguir el proceso establecido en la metodología planteada, el Sistema de Gestión Académica (Notas y Matrículas) alcanza el Nivel 3 Satisfactorio (de cuatro niveles), el análisis detallado se podrá revisar en el Informe de Evaluación del Sistema, que se considera la salida de esta etapa.

#### **4.1.4 Discusión**

La metodología es una herramienta para evaluar la calidad del sistema, en base a las características más importantes dependiendo del tipo de software y el sector en el que se utiliza. Una de las subcaracterísticas que llamó más la atención en el momento de las pruebas fue su capacidad para intercambiar datos al obtener el 100% de datos intercambiados de manera correcta y esto se debe (en lo que se pudo observar) a que cuentan con un robusto diseño a nivel de base de datos, es necesario mencionar que este criterio es subjetivo, debido a que las pruebas se realizaron a nivel externo y de uso, y se tendría que realizar un análisis exhaustivo de su estructura interna.

Además, el nivel de satisfacción de los usuarios hacia el sistema (67,38) es más bajo con respecto a la calificación total (78,48), una de las causas es porque algunas personas desconocen el manejo de la aplicación debido a que las herramientas para guía como los manuales de usuario no están disponibles en el sistema, otra de las causas es porque muchos usuarios consideran que el software no está disponible todo el tiempo.

En el anexo 8 se adjuntan los documentos llenados con la información de la ESPAM MFL.



## CAPÍTULO V

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1. Conclusiones

- Se diseñó una metodología de evaluación de la calidad del software basada en la norma ISO/IEC 25040 y metodología Bootstrap para determinar la calidad del Sistema de Gestión Académica de la ESPAM MFL, a través de una investigación cualitativa basada en teoría fundamentada.
- Al aplicar la metodología propuesta en el Sistema de Gestión Académica de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López se logró obtener el estado del arte sobre las bases teóricas de la calidad del software, modelos de evaluación de calidad de software, la norma ISO/IEC 25040 y la metodología Bootstrap.
- La definición de una metodología comprende: el ciclo de vida de la calidad del producto software, características, sub-características y métricas de calidad del software, escalas de calidad, las etapas de la metodología y la validación de la misma a través de uno o varios casos de estudio, en este caso la del “Sistema de Gestión Académica” de la ESPAM MFL
- Las métricas propuestas pueden cambiar dependiendo de la naturaleza del software, del criterio del evaluador y la aplicación de otras metodologías.

## **5.2. Recomendaciones**

- Aplicar la metodología de evaluación de la calidad de software en otros sistemas que tengan características similares al caso de estudio.
- Los encargados del desarrollo del Sistema de Gestión Académica (notas y matrículas) acojan las recomendaciones planteadas en este caso de estudio.
- Es importante que en toda institución considere la implementación de una metodología de evaluación de software, como parte de su Plan Estratégico Institucional, y Plan Operativo.

## BIBLIOGRAFÍA

- Aguayo, J. (2014). *Métricas de Calidad y el desarrollo de software competitivo en la empresa J-M Software Developer de la Ciudad de Ambato*. Ambato, Ecuador.
- Aparicio, A. (2012). *Ingeniería de Software*. Datateca, Universidad Nacional Abierta y a Distancia.
- Ayala, R. E. (2010). *Calidad de Software*. México DF: Instituto Politécnico Nacional.
- C. Calero, M. P. (2010). *La calidad del producto y proceso Software*. Editorial Ramma.
- Carvalho, J. P. (2007). *Calidad de Componentes Software*.
- Castillo, L. (2013). *Evaluación, calidad y gestión de calidad total en documentación*.
- Cendejas, J. (12 de 10 de 2017). *Implementación del modelo integral colaborativo (mdsic) como fuente de innovación para el desarrollo ágil de software en las empresas de la zona centro - occidente en México*. . Obtenido de Enciclopedia y Biblioteca Virtual de las Ciencias Sociales, Económicas y Jurídicas: <http://www.eumed.net/tesis-doctorales/2014/jlcv/calidad-software.htm>
- Corrochano, J. (2013). *La Calidad Del Producto Software*. Departamento de Informática - Universidad Carlos III de Madrid.
- Davila, L., & Mejia, P. (s.f.). *Evaluación de la calidad de Software en sistemas de información en Internet*. Obtenido de <http://delta.cs.cinvestav.mx/~pmaalvarez/davila-mejia.pdf>

De La Villa, M. (2004). *Modelos de Evaluación y Mejora de Procesos: Análisis Comparativo*.

*Diplomado a Distancia "Informática Médica"*. (s.f.). Obtenido de Facultad de Medicina UNAM:

<http://www.facmed.unam.mx/emc/computo/infomedic/presentac/modulos/ftp/documentos/calidad.pdf>

Duque, A. (15 de 11 de 2017). *Framework BASADO EN EL ESTANDAR DEL SOFTWARE ISO/IEC 25000:2005 (SQUARE) PARA LA EVALUACIÓN DE LA CALIDAD DE LAS ONTOLOGÍAS*. Obtenido de Digitum: Repositorio Institucional de la Universidad de Murcia:

<https://digitum.um.es/xmlui/bitstream/10201/47818/1/TesisAstridSinPublicaciones.pdf>

Ejiogu, L. (1991). *Software engineering with formal metrics*.

Enrique, C. (14 de 11 de 2017). *Elementos que permiten evaluar la calidad de software*.

Obtenido de Portafolio de Algorítmica y Programación:

<https://sites.google.com/site/portafoliocarlosmacallums/unidad-ii/elementosquepermitenevaluarlcalidaddesoftware>

Escalone, F. (2006). *Estudio comparativo de los modelos y estándares de calidad del Software*. Buenos Aires: Universidad Tecnológica Nacional.

Estayno, M. D. (2009). *Modelos y métricas para evaluar la calidad del software*. XI Workshop de Investigadores en Ciencias de la Computación.

- Fernández, L. (20 de 10 de 2017). *Teoría de la medición*. Obtenido de José Javier Dolado Cosín: <http://www.sc.ehu.es/jiwdocoj/remis/docs/teoriamedicion.html>
- Gonzalez, A., Ampuero, M., & Hernández, A. (2015). *Análisis comparativo de modelos y estándares para evaluar la calidad del producto de Software*. Instituto Superior Politécnico.
- Guler, I., Guillen, M., & Macpherson, J. M. (2002). Global competition, institutions and the diffusion of organizational practices: The international spread of ISO 9000 quality certificates. *Administrative Science Quarterly*.
- IEC, I. (2005). *ISO/IEC 25000 software engineering software product quality requirements and evaluation (SQuaRE) guide to SQuaRE*. Systems Engineering.
- Juran, J. (1992). *Quality by Design: The New Steps for Planning Quality into Goods and Services*. Free Press.
- Komi-Sirviö, S. (2004). *Development and Evaluation of Software Process Improvement Methods*. University of Oulu.
- Kuvaja et al. (1994). *Software Process Assessment and Improvement: The BOOTSTRAP Approach*. Oxford: Blackwell Publishers.
- Lethbridge, T. (2003). *Comunicación privada sobre métricas de software*.
- Marcos, J., Arroyo, A., Garzas, J., & Piattini, M. (2008). *La norma ISO/IEC 25000 y el proyecto KEMIS para su automatización con software libre*. *Revista Española de Innovación, Calidad e Ingeniería del Software*.

McCall, J., Richards, P., & Walters, G. (1977). *Factors in Software Quality*. Rome: Rome Air Development Center.

Moreno, A. M., Imbert, R., Nakasone, A., & Pow-Sang, J. A. (2008). *An Approach to Determine Software Requirement Construction Sequences Based on Use Cases*. IEEE.

Norberto, O., & Gloria, C. (2011). Gestión de calidad en desarrollo de software. *Revista de Investigación de Sistemas e Informática*.

Norma ISO/IEC 25000. (10 de 11 de 2017). Obtenido de ISO/IEC 25000: <http://iso25000.com>

Pallares, A. (13 de Jun de 2012). *Cómo interpretar las estadísticas de Google*. Obtenido de Smart Up: <https://smartupmarketing.com/como-interpretar-las-estadisticas-de-google-analytics/>

Paulk, M., Curtis, B., Chrissis, M. B., & Weber, C. (1993). *Capability Maturity Model*.

Pressman, R. (2010). *Ingeniería de Software - Un enfoque práctico*. México DF: McGraw-Hill.

Roche, J. (1994). *Software Metrics and measurement principles, in software engineering*. ACM.

Rubio, M. (01 de 07 de 2010). *Altenwald*. Obtenido de <http://altenwald.org/2010/07/01/calidad-interna/>

Tuya, J., Ramos, I., & Dolado, J. (2007). *Técnicas Cuantitativas para la Gestión en la Ingeniería de Software*. Coruña: Netbiblo .

Valdez, F. (2013). *Midiendo la Calidad del software*. *Software Guru*.

Vega, C., Rivera, L., & García, A. (2008). *Mejores Prácticas para el Establecimiento y Aseguramiento de la Calidad de Software*. Medellín: Universidad Cristobal Colon.