

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR EN
ESMERALDAS**



ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

TESIS DE GRADO

**CARACTERIZACIÓN DE METODOLOGÍAS DE MÉTRICAS DE
SOFTWARE PARA LA OPTIMIZACIÓN DEL DESARROLLO WEB**

LINEA DE INVESTIGACIÓN:

GOBIERNO DE TECNOLOGÍAS DE LA INFORMACIÓN

**Previo a la obtención del título de Ingeniero de Sistemas y
Computación**

AUTOR:

Mejía Cruel Nathali Yomira

ASESOR:

Mgt. Jaime Sayago Heredia

Esmeraldas-Marzo , 2018

Tesis de grado aprobada luego de haber
dado cumplimiento a los requisitos
exigidos, previo a la obtención del título
de INGENIERO DE SISTEMAS Y
COMPUTACIÓN.

TRIBUNAL DE GRADUACIÓN

Título: “CARACTERIZACIÓN DE METODOLOGÍAS DE MÉTRICAS DE
SOFTWARE PARA LA OPTIMIZACIÓN DEL DESARROLLO WEB”

Autor: NATHALI YOMIRA MEJIA CRUEL

Mgt. Jaime Sayago Heredia f.-.....
Asesor

Mgt. Wilson Chango Sailema f.-.....
Lector N° 1

Mgt. Marc Grob f.-.....
Lector N° 2

Mgt. Xavier Quiñónez Kú f.-.....
Director de Escuela

Ing. Maritza Demera Mejía f.-.....
Secretaria General PUCE-E

Esmeraldas, Marzo 2018

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, **NATHALI YOMIRA MEJIA CRUEL** con cédula de identidad No. **080431907-7** declaro mediante la presente que los resultados en el proyecto de investigación que presento como tesis de grado, previo a la obtención del título de “Ingeniero en Sistemas y Computación” son totalmente personales, únicos, y legítimos.

Al mismo tiempo, declaro que todo el contenido incluyendo resultados, conclusiones y por otro lado los efectos académicos y legales que se desglosan de esta investigación son y serán de exclusiva responsabilidad académica y legal.

NATHALI YOMIRA MEJIA CRUEL

CI 080431907-7

DEDICATORIA

Este trabajo se lo dedico a mis padres que me han brindado su comprensión y apoyo en este largo y arduo camino, porque siempre han sido mi más grande motivación para seguir adelante y cumplir todas las metas que me he trazado.

Nathali Yomira Mejía Cruel

AGRADECIMIENTOS

Agradezco principalmente a mis padres por ser la ayuda moral, espiritual, porque a pesar de las dificultades, siempre han sido mi ejemplo para mí, a mis hermanos por apoyarme y ayudarme a conseguir cualquier meta planteada. A toda mis familiares y amigos que de cualquier manera colaboraron para que esta culminación de carrera universitaria haya llegado a su mejor fin. Y sobre todo agradezco a Dios.

Nathali Yomira Mejía Cruel

ÍNDICE DE CONTENIDO

Páginas Preliminares

TRIBUNAL DE GRADUACIÓN	ii
DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD	iii
DEDICATORIA	iv
AGRADECIMIENTOS	v
RESUMEN	ix
ABSTRACT.....	x
1. INTRODUCCIÓN	1
1.1. PRESENTACIÓN DEL TEMA DE INVESTIGACIÓN	1
1.2. PLANTEAMIENTO DEL PROBLEMA	1
1.3. JUSTIFICACIÓN.....	3
1.4. OBJETIVOS	3
1.4.1. Objetivo General.	3
1.4.2. Objetivos Específicos.	3
CAPÍTULO I: MARCO TEÓRICO	3
1.1. ANTECEDENTES.....	4
1.2. BASES TEÓRICAS CIENTÍFICAS	5
1.2.1. Ingeniería De Software	5
1.2.2. Calidad Del Software.....	7
1.2.3. Métricas.....	8
1.2.4. Modelos Y Estándares De Calidad De Software.....	8
1.2.5. Modelos De Calidad De Software A Nivel Proceso.	9
1.2.6. Estándares De Calidad De Software.	18
1.3. BASES LEGALES.....	25
CAPÍTULO II: METODOLOGÍA	26
2.1. DESCRIPCIÓN Y CARACTERIZACIÓN DEL LUGAR	26
2.2. TIPO DE INVESTIGACIÓN	26
2.3. MÉTODOS Y TÉCNICAS	26
2.4. POBLACIÓN Y MUESTRA (TÉCNICAS DE MUESTREO)	27
2.5. DESCRIPCIÓN Y VALIDACIÓN DEL INSTRUMENTO	27
2.6. DESCRIPCIÓN DE LAS TÉCNICAS DE PROCESAMIENTO Y ANÁLISIS DE DATOS	27
2.6.1. Análisis De Las Entrevistas Para La Evaluación Del Proceso	27
2.6.2. Proceso de Evaluación.....	29
2.7. NORMAS ÉTICAS.....	31
CAPÍTULO III: RESULTADOS	32

3.1. ANÁLISIS COMPARATIVO DE MODELOS Y ESTÁNDARES DE CALIDAD DE SOFTWARE	32
3.2. COMPARATIVA DE MODELOS Y ESTANDARES DE CALIDAD DE SOFTWARE CON PONDERACIONES	36
3.3. EVALUACIÓN DEL PROCESO DE MANTENIMIENTO DE SOFTWARE CON LA NORMA ISO/IEC/IEEE 12207	38
CAPÍTULO IV: DISCUSIÓN.....	39
CAPITULO V: PROPUESTA	41
CAPÍTULO VI: CONCLUSIONES.....	43
CAPÍTULO VII: RECOMENDACIONES	44
REFERENCIAS BIBLIOGRÁFICAS	45
ANEXOS	49
ANEXO A: ESTRUCTURA DE ISO 9001:2008	49
ANEXO B: ENTREVISTA.....	52
ANEXO C: FICHA DE OBSERVACIÓN	53
ANEXO D: EVIDENCIAS DE LAS ENTREVISTAS	54

Índice de Tablas

Tabla 1. Modelos y Estándares de Calidad de Software a nivel Proceso. Fuente: Propia	9
Tabla 2. Niveles de Capacidad y Madurez. Fuente: (Chrissis et al. ,2010).....	11
Tabla 3. Medidas de PSP	15
Tabla 4. Análisis de las entrevistas	27
Tabla 5. Propósito y Outcomes del proceso de mantenimiento de software.	31
Tabla 6. Análisis Comparativo de Metodologías de Calidad de Software.	33
Tabla 7. Análisis de modelos de calidad con ponderaciones.....	36
Tabla 8. Resultado de la evaluación del proceso de mantenimiento de software.....	38
Tabla 9. Comparación de la evaluación del proceso de mantenimiento en distintas investigaciones.....	39

Índice de Figuras

Figura 1. Categorías de las Áreas de Proceso de CMMI-DEV v1.3. Fuente: Elaboración propia	10
Figura 2. Estructura ISO 9001:2008	19
Figura 3. Grupos De Procesos Del Ciclo De Vida ISO 12207.	21
Figura 4. Niveles de Capacidad SPICE. (Ruvalcaba, 2004).....	22
Figura 5. Ciclo de Vida de ITIL. Fuente: (Ricardo,2015).	23
Figura 6. Descripción del Ciclo de vida de ITIL. Fuente: Tecnofor (2010).	23
Figura 7. Principios de COBIT V5. (Armendáriz, 2017).	24
Figura 8. Niveles de capacidades definidos (según ISO/ IEC 15504)	25
Figura 9. Niveles de capacidad y atributos de procesos. Fuente: (ISO/IEC 15504, 2003)	29
Figura 10. Criterios de evaluación. Fuente: (ISO/IEC 15504, 2003)	30
Figura 11. Evaluación de los atributos de procesos. Fuente: (ISO/IEC 15504, 2003) ...	30

RESUMEN

La carencia del hábito de usar metodologías que permitan la medición de la calidad al momento del desarrollo de software promueve que la gestión de calidad y el aseguramiento de la misma sea un tema de gran importancia en las empresas, dado que es un método fiable evaluar y mejorar el proceso de desarrollo de software para mejorar la calidad del mismo; esta investigación tiene la finalidad de realizar un estudio de las características de las distintas metodologías de métricas que determinan el nivel de calidad de un proceso determinado del software, para posterior a esto efectuar un análisis comparativo de las mismas y realizar la adecuada elección para el caso PUCE Esmeraldas; algunos de los parámetros a tomarse en cuenta para el análisis comparativo son: tamaño de la organización, objetivo, niveles de madurez, plazos de los resultados, recursos de la organización para la implementación, ámbito de aplicación. Con la metodología elegida se aplicaron las métricas al proceso de mantenimiento de software de la Pontificia Universidad Católica del Ecuador Esmeraldas para luego analizar los resultados y extraer generalizaciones significativas que permitan la mejora de este proceso.

Para la elaboración del análisis comparativo se realizó una investigación bibliográfica previa, que permitió determinar de cada metodología los parámetros que establecidos en el cuadro comparativo.

Se utilizó la técnica de la entrevista, misma que fue aplicada al personal involucrado en el proceso de mantenimiento de software; adicional a esta se usó la técnica de la observación, en la cual la información obtenida por las fichas de observación analizada para la comprobación de los datos obtenidos en las entrevistas. Ambas técnicas fueron trianguladas con la documentación formal obtenida en el departamento de TIC de PUCE Esmeraldas, para realizar la medición del proceso de mantenimiento de software aplicando la norma ISO/IEC 12207. Como resultado de la evaluación se generó una propuesta acorde a la ISO aplicada con actividades a realizar en el proceso.

Palabras clave: calidad, métricas, modelos, estándares.

ABSTRACT

The lack of the habit of using methodologies that allow the measurement of quality at the time of software development promotes that quality management and assurance of it is an issue of great importance in companies, given that it is a reliable method to improve the software development process to improve the quality of it; this research has the purpose of carrying out a study of the characteristics of the different methodologies of parameters that determine the level of quality of a certain process of the software, in order to carry out a comparative analysis of the methodologies and make the appropriate choice of the methodology. for the PUCE Esmeraldas case; some parameters of analysis for the comparative analysis are: size of the organization, objective, maturity levels, application deadlines, resources of the organization for the implementation, scope of application. With the chosen Methodology, the techniques were applied to the software maintenance process of the Pontifical Catholic University of Esmeraldas to analyze the results and extract generalizations that allow the improvement of this process.

For the elaboration of the comparative analysis a previous bibliographical investigation was carried out, which allowed to determine what it was about.

The interview technique was used, which was applied to the personnel involved in the software maintenance process; Additionally, the technique of observation is used, in which the information obtained by the observation sheets analyzed for the verification of the data obtained in the interviews. Both techniques were triangulated with the formal documentation obtained in the ICT department of PUCE Esmeraldas, to perform the measurement of the software maintenance process applying the ISO / IEC 12207 standard. As a result of the evaluation, a proposal according to the ISO was generated. with activities to be carried out in the process.

Keywords: quality, metrics, models, standards.

1. INTRODUCCIÓN

1.1. PRESENTACIÓN DEL TEMA DE INVESTIGACIÓN

La falta de hábito al usar metodologías que permitan la medición de la calidad al momento del desarrollo de software promueve que la gestión de calidad y el aseguramiento de la misma sea un tema de gran importancia en las empresas, debido a que las empresas de desarrollo deberían ofrecer software que cumpla con los niveles de calidad establecidos.

Para entregar a los clientes software de calidad es indispensable realizar el control de algunos aspectos como documentación, requerimientos, pero especialmente sobre los procesos del software.

En esta investigación se presentan temas como la calidad del software, las métricas de software, y además los modelos y estándares de calidad de software enfocados a nivel de proceso.

La presente investigación tiene la finalidad de realizar un estudio de las características de las distintas metodologías de métricas que determinan el nivel de calidad de un proceso determinado del software, para posterior a esto efectuar un análisis comparativo de las metodologías y realizar la elección de la metodología que mejor se adapte a la organización.

Con la metodología elegida se han aplicado las métricas al proceso de mantenimiento de software de la Pontificia Universidad Católica de Esmeraldas, posterior a esto se analizaron los resultados y se extrajeron generalizaciones significativas que van a permitir la mejora de este proceso.

1.2. PLANTEAMIENTO DEL PROBLEMA

Según SOMMERVILLE (2005) “Los ingenieros de software adoptan un enfoque sistemático y organizado con el objetivo de elaborar sistemas o aplicaciones óptimas y

de alta calidad.” El proceso de análisis y diseño debe llevar una orientación de aseguramiento de la calidad para cumplir el objetivo principal.

Sin embargo, el numeroso crecimiento de la producción de aplicaciones ha generado que las empresas que elaboran los sistemas de software se despreocupen en cierta parte de la calidad del software. Debido a que tienen la idea errónea de que al ocupar tiempo en la aplicación de estándares o modelos de calidad de software hacen que la compañía sea menos competitiva. Este crecimiento de producción ha provocado el incremento del uso de metodologías que se encargan de la medición de software, gestión de proceso y mejora de calidad, para ello un componente importante es el uso de las métricas de software.

Piedad Metaute y Serna (2016) afirma que "las métricas son un buen medio para monitorizar, controlar y probar el desarrollo del software y los planes de mantenimiento".

El uso de metodologías desde el inicio del desarrollo de un proyecto podría prevenir una cantidad considerable de errores en el software y así como también generaría el aumento de su calidad; para ello es necesario un análisis previo para realizar una elección, la cual va a depender de los requerimientos del software.

La evolución del software es necesaria para la optimización del mismo, se realiza a través de metodologías que mediante la elaboración de ajustes permiten generar una nueva versión del software.

La evaluación de un software está basada en modelos y métricas de calidad que nos permiten conocer la valoración de la calidad del software y puntualizar áreas problemáticas de manera que puedan generarse soluciones para su mejora.

Para realizar la medición de una aplicación web o de un proceso de software es fundamental analizar y seleccionar de manera acertada una metodología o modelo de calidad software con los parámetros y enfoque del proceso o software a medirse.

1.3. JUSTIFICACIÓN

A través de esta investigación se aporta con la aplicación de una metodología específica de medición de software, gestión de proceso y mejora de calidad, puesto que elaborar un producto de calidad involucra que el producto desarrollado debe tener semejanza con su especificación o requerimiento, es decir, que el producto cumple con todos los requerimientos que hizo el cliente.

En ese sentido, la investigación permitió la elección adecuada de la metodología que se aplicó mediante una caracterización de metodologías de métricas de software para la optimización del contexto web universitario que en este caso es el proceso de mantenimiento de software de la PUCE Esmeraldas.

Desde la perspectiva metodológica, se ha implantado un conjunto de criterios que sirven como guía de orientación al momento de elegir una metodología de métricas de software.

A nivel práctico, la investigación aborda los problemas específicos que se presentan en el proceso en cual se han aplicado las métricas de software siendo esto de gran importancia para el aumento del nivel de calidad del mismo.

1.4. OBJETIVOS

1.4.1. Objetivo General.

Caracterizar metodologías de métricas de software mediante un análisis comparativo para la optimización del desarrollo web.

1.4.2. Objetivos Específicos.

- Organizar los distintos modelos y estándares de calidad de software a nivel proceso.
- Describir las características que poseen los distintos modelos y estándares de calidad de software.
- Efectuar un análisis comparativo entre paradigmas actuales que se emplean para la medición de software, gestión de proceso y mejora de calidad.
- Realizar la evaluación de un proceso con la metodología que se adapta a la organización.

CAPÍTULO I: MARCO TEÓRICO

1.1. ANTECEDENTES

En esta sección se presentan varios trabajos de investigación relacionados con las métricas, gestión de proceso y mejora de calidad del software, mismos que darán un aporte significativo a esta investigación.

Osorio y Castro (2011) en su artículo “Gestión de calidad en desarrollo de software” afirman que “El proceso de medición de software consiste en dar un valor numérico a un atributo del software o del proceso de software.” Para que se dé la identificación de las entidades o atributos a medirse todo proceso de medición de software debe satisfacer las necesidades de información.

Uno de los trabajos relevantes es la investigación sobre “Aseguramiento de Calidad de Software: Estudio Comparativo de Estrategias de Medición y Evaluación” en la cual se realiza una evaluación de las estrategias integradas de medición y evaluación.

Según Papa (2012):

Una estrategia integrada de medición y evaluación debe estar basada en tres principios o fundamentos, a saber: una especificación explícita y formal de un proceso de medición y evaluación; un modelo conceptual que defina todos los términos, propiedades y relaciones del dominio, que a su vez se agrupen en módulos o componentes; y un soporte metodológico y tecnológico, en el que métodos y herramientas se puedan instanciar a partir del proceso y componentes del marco conceptual. (p.137)

Ruiz y Cordero (2013) en su investigación “Estimación en proyectos de software integrando los métodos de Boehm y Humphrey” afirman que para la medición de determinados tipos de software existen los denominados métodos de estimación, los cuales dependen del dominio o tipo de desarrollo del software. Además, aseveran que “Los métodos de producción de software se han basado en los modelos de estimación publicados desde los años 60 hasta la actualidad”.

Perurena y Moraguez (2013) desarrollaron una investigación sobre "Usabilidad de los sitios Web, los métodos y las técnicas para la evaluación" la cual hace referencia a las

distintas definiciones acerca de la usabilidad, su agregación en el proceso de ingeniería y el vínculo con la ingeniería de software, sus propiedades y métodos de evaluación. En esta investigación los autores concluyen que es importante evaluar la usabilidad del software en todo su desarrollo y de manera más precisa cuando concluye cada ciclo iterativo, lo que le permitirá al desarrollador conocer el nivel de mejora que debe aplicar para cumplir sus objetivos de usabilidad.

Constanzo, Casas y Marcos (2014) definen a las Métricas de Software como la mejor manera de proporcionar evidencia que permita comprender la complejidad del software; estas métricas se deben aplicar durante el proceso de elaboración del software y cuando el producto esté terminado. Todas estas evidencias son utilizadas para la mejora continua del software. Las métricas se clasifican en: directas, no necesitan de una entidad o atributo; e indirectas, necesitan de una o más métricas directas. Dependiendo el contexto también pueden estar clasificadas como: métricas de proceso, métricas de proyecto y métricas de producto.

Pinto et al. (2013) en su investigación “QUCO2: Una herramienta para Medir la Calidad de las Aplicaciones Web.” afirma que “la calidad es un tema subjetivo que depende del nivel de satisfacción que tenga el usuario con respecto al producto manipulado”, todo software se encuentra sujeto a la evaluación de su calidad de la misma manera que cualquier otro producto, con lo cual los usuarios pueden establecer el grado en el cual esté satisface sus necesidades.

1.2. BASES TEÓRICAS CIENTÍFICAS

1.2.1. Ingeniería De Software

“La ingeniería de software se encuentra formada por un proceso, un conjunto de métodos (prácticas) y un arreglo de herramientas que permiten a los profesionales elaborar software de alta calidad.” (Pressman, 2010, p. 1). Por tanto, se puede decir, que la ingeniería de software busca mejorar la calidad de sus productos para conseguir que se ajusten a los requerimientos de los usuarios aumentando la competitividad entre los mismos productos. Estos productos son llamados software, que son un conjunto de instrucciones que al ser ejecutados cumplen una función específica.

Pressman (2010) afirma que las aplicaciones web (webapps) son un “conjunto vinculado de archivos de hipertexto que muestran información con uso de texto y gráficas limitadas”. Las webapps poseen varios atributos que las diferencian de los demás softwares como son:

Uso intensivo de redes. Toda webapp se encuentra alojada en una red, ya sea con acceso mundial (internet) o limitado(intranet).

Concurrencia. Varios usuarios pueden acceder a la webapp al mismo tiempo.

Carga impredecible. El número de usuarios de la webapp cambia de un día a otro.

Rendimiento. El tiempo de respuesta por parte del sistema debe ser mínimo para que el usuario decida irse a otra parte.

Disponibilidad. Se espera que se posea una disponibilidad lo más cercano al 100% puesto que en muchas webapps populares los usuarios demandan acceso las 24 horas de los 365 días del año.

Orientadas a los datos. Las webapps tienen como uso común acceder a las bases de datos y a la que en ellas se encuentra.

Contenido sensible. La calidad de una webapp se encuentra definida en gran proporción por la calidad y naturaleza estética del contenido de la misma.

Evolución continua. Las aplicaciones web evolucionan en forma continua especialmente su contenido.

Inmediatez. Es usual que las webapps lleguen al mercado en plazos de algunos días o semanas.

Seguridad. Debe implementarse medidas de seguridad dentro de las webapps que permita brindar modos seguros de transmisión de los datos, debido a que no se puede limitar la población de usuarios finales.

Estética. Para el éxito de una webapp es importante una buena estética, ya que la apariencia es el atractivo de toda webapp.

1.2.2. Calidad Del Software.

El grado de una combinación específica de atributos que se desea tener, es denominado calidad del software. Enriquez y Casas (2014) afirman que “Definir calidad de software para un sistema es semejante a definir una lista de atributos de calidad del software requeridos para ese sistema” (p. 26).

Osorio y Castro (2011) afirman que: “La calidad es definida como el grado de relación que tiene el producto para satisfacer las necesidades del usuario. Un software que cumple con todos los requisitos..., y que sus procesos se ejecuten correctamente garantiza una buena Calidad.” (p. 66).

Según Poveda (2010) “Como respuesta a las diferentes necesidades que posee la humanidad en cuanto a servicio, tecnología, comunicación y comodidades; la calidad es la expresión de la creación y del conocimiento en todos los ámbitos en los que se desarrolla el hombre.”

SQA (Software Quality Assurance) o gestión de calidad del software es una actividad para la protección del software que se realiza en todo el proceso del mismo, es decir, abarca procedimientos, métodos, herramientas y técnicas para la revisión, además de estrategias y técnicas de pruebas, mecanismos de medida e información.

Serrano (2006) asegura que la gestión de calidad de un sistema de software abarca varias áreas como: a) **Planificación del desarrollo y del mantenimiento**, en esta área se ve involucrada la gestión de los recursos(humanos o materiales) que sirven para producir el producto de software, también se incluye el entrenamiento de los potenciales usuarios del sistema a desarrollar; b) **Control de la calidad técnica del producto**, involucra las métricas y procedimientos de validación que sirven para asegurar un nivel de calidad establecido; entre estos podemos considerar: revisiones y auditorías de las actividades, documentos y productos intermedios generados, actividades de prueba, gestión de errores y medidas correctoras en general; c) **Control de versiones y configuraciones**, en este se debe considerar la gestión de la documentación y del producto de software, es decir: protección, almacenamiento, control de acceso, uso, distribución y mantenimiento durante la evolución del producto; d) **Gestión de riesgos**, esta área abarca el control de los proveedores externos tanto como el control de la

calidad de los componentes que estos entregan, el abastecimiento de componentes solicitados y el de las soluciones técnicas existentes.

1.2.3. Métricas.

Las métricas son medidas realizadas sobre un proceso específico o sobre algún atributo del sistema, que permite tomar decisiones para la mejora del proceso o atributo, mediante la comparación con valores de referencias o indicadores. García, López, Pérez y Victore (2013) definen las métricas como “medidas cuantitativas del grado en que un sistema, componente o proceso posee un atributo dado”.

Es necesario establecer un modelo de medición o modelo de calidad para lograr proporcionar valores cuantitativos a las métricas. Luque (2015) clasifica a las métricas de la siguiente manera: a) *Métricas de Software*, tiene como objetivo la valoración cuantitativa de la calidad del software. Estas mediciones sirven para confirmar el correcto funcionamiento del software, realizan la evaluación de la calidad y de los modelos de diseño, los casos de prueba y el código fuente; b) *Métricas de Proyecto*, son las más relevantes a nivel de equipo de proyecto, de vital importancia sobre todo a la hora de la planificación y adecuando la estrategia que se tomará para llevar a cabo el proyecto; c) *Métricas de Proceso*, se obtienen concretando los atributos específicos de un proceso, desarrollaremos un conjunto de métricas significativas sobre la información que disponemos de estos atributos, finalmente se obtienen los indicadores que para futuras situaciones ayudan a obtener una estrategia que mejoraría el proceso.

1.2.4. Modelos Y Estándares De Calidad De Software.

Los modelos y estándares de calidad de software sirven tanto para diseñar como para evaluar al software, estos modelos se encargan de describir las características de los productos, como pueden ser medidas y cómo se puede interpretar la medición de estos. Scalone (2006) asegura que la implementación de estos modelos y estándares posee algunas ventajas como: la mejora continua de la calidad en los productos y procesos de software, la certificación de la competitividad internacional que es necesaria para competir con los demás mercados, así como también el enfoque de cumplir con los requerimientos del cliente de parte del personal de la empresa. Los Modelos/Estándares de Calidad de Software que se plantean en esta investigación se encuentran enfocados a nivel proceso como se visualiza en la Tabla 1.

Tabla 1. Modelos y Estándares de Calidad de Software a nivel Proceso. Fuente: Propia

Modelos de Calidad del Software	Estándares de Calidad del Software
CMMi	ISO 90003
Personal SW Process (PSP)	ISO 12207
Team SW Process (TSP)	ISO 15504 (SPICE)
	ITIL
	Cobit 5.0

1.2.5. Modelos De Calidad De Software A Nivel Proceso.

(CMMI-DEV) VERSIÓN 1.3. CAPABILITY MATURITY MODEL INTEGRATION

CMMI son conjuntos de buenas prácticas establecidas por miembros del Gobierno, Industria y del Instituto de Ingeniería de Software (SEI) para ayudar a las organizaciones en la mejora de sus procesos. CMMI para desarrollo (CMMI-DEV) ofrece una colección completa de guías para el desarrollo de productos y servicios que satisfagan con las necesidades de los clientes, estas prácticas cubren el ciclo de vida del producto es decir la concepción, la entrega y hasta el mantenimiento. CMMI DEV 1.3, consta de 22 áreas de procesos, de las cuales: 16 son áreas de proceso base, 1 es un área de proceso compartida y 5 son áreas de proceso específicas de desarrollo (Chrissis et al., 2010).

Las 22 áreas de procesos se encuentran agrupadas en cuatro categorías: Gestión de Proyectos, Gestión de Procesos, Ingeniería y Soporte, como se muestran en la Figura 1.

Gestión de Proyectos	Gestión de Procesos	Procesos de Ingeniería	Soporte
<ul style="list-style-type: none"> •Gestión Integrada de Proyectos (IPM) •Control y Monitoreo de Proyectos (PMC) •Planificación de Proyectos (PP) •Gestión de Requerimientos (REQM) •Gestión de Riesgos (RSKM) •Gestión Cuantitativa de Proyectos (QPM) •Gestión del Acuerdo de Proveedores (SAM) 	<ul style="list-style-type: none"> •Desempeño del Proceso Organizacional (OPP) •Definición de Proceso Organizacional (OPD) •Enfoque en el Proceso Organizacional (OPF) •Gestión del Desempeño Organizacional (OPM) •Entrenamiento Organizacional (OT) 	<ul style="list-style-type: none"> •Desarrollo de requisitos (RD) •Solución técnica (TS) •Integración de Productos (PI) •Verificación (VER) •Validación (VAL) 	<ul style="list-style-type: none"> •Análisis y Resolución de causas (CAR) •Gestión de la Configuración (CM) •Análisis y Resolución de las Decisiones (DAR) •Medición y Análisis (MA) •Aseguramiento de la Calidad de Proceso y Producto (PPQA)

Figura 1. Categorías de las Áreas de Proceso de CMMI-DEV v1.3. Fuente: Elaboración propia

Cada área de proceso posee varios componentes como (Ver Fig. 2): declaración del propósito, notas introductorias, áreas de proceso relacionadas, metas específicas, prácticas específicas, metas genéricas, prácticas genéricas, sub-prácticas, elaboración de productos de trabajo, ejemplos, elaboraciones de práctica genérica.

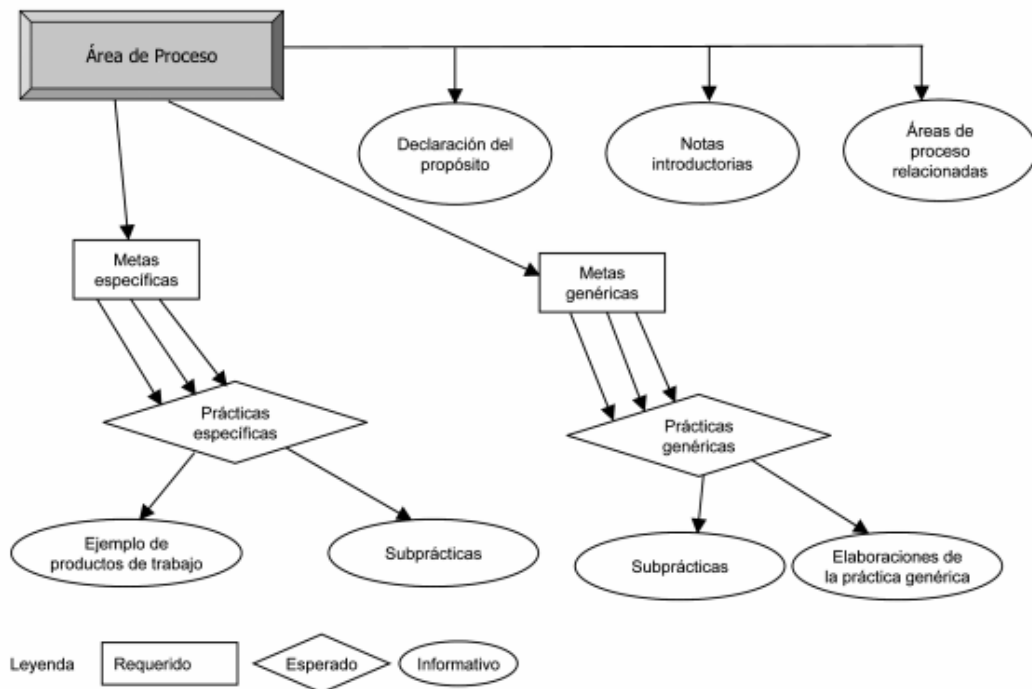


Figura 2. Componentes del modelo CMMI. Fuente: (Chrissis et al. ,2010).

En CMMI-DEV los niveles corresponden al resultado de calificación en las evaluaciones, como se visualiza en la Tabla 2 estos niveles dependen de la representación que se elija ya sea: **representación continua**, utiliza el nivel de capacidad y determina el estado de los procesos con respecto a un área; **representación por etapas**, utiliza el nivel de madurez y determina el estado global de la organización.

Tabla 2. Niveles de Capacidad y Madurez. Fuente: (Chrissis et al. ,2010).

NIVEL	REPRESENTACIÓN CONTINUA	REPRESENTACIÓN POR ETAPAS
	NIVELES DE CAPACIDAD	NIVELES DE MADUREZ
NIVEL 0	Incompleto	
NIVEL 1	Realizado	Inicial
NIVEL 2	Gestionado	Gestionado
NIVEL 3	Definido	Definido
NIVEL 4		Gestionado Cuantitativamente
NIVEL 5		En optimización

PERSONAL SOFTWARE PROCESS (PSP).

Delgadillo, Abud-Figueroa, Peláez-Camarena, Alor-Hernández y García (2016) definen al PSP como un marco de trabajo que permite mejorar tanto los procesos como el rendimiento de los programadores, funciona a nivel personal y se establece en los procesos de construcción del software.

Este modelo consta de varios principios: a) los procesos son definidos y medidos por los desarrolladores, para mejora continua del desempeño; b) los desarrolladores deben responsabilizarse de la calidad de sus trabajos; c) encontrar y reparar defectos en fases tempranas del desarrollo para reducir costos; d) Es más eficiente prevenir los defectos que buscarlos y repararlos; e) El camino correcto es siempre el más rápido y barato de trabajar.

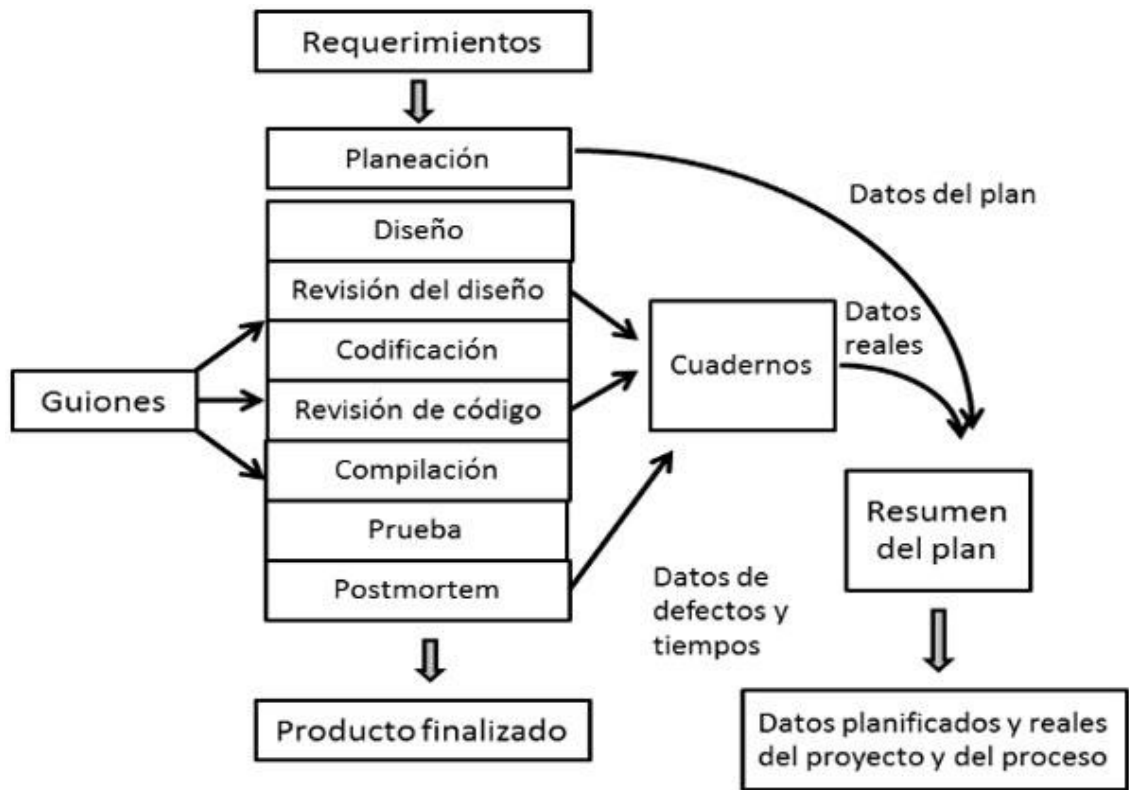


Figura 3. Flujo de procesos del Modelo PSP. Fuente: (Puello et al.,2014)

El proceso de PSP (Figura 3) empieza con los requerimientos, luego con el script de planeación, junto con el resumen de planeación en el que se registran los datos de la planificación. Además, se hace el registro del tiempo y los datos de los defectos, los cuales serán sumariados en la etapa de Postmortem, en donde también se va a medir el tamaño del programa; todos estos datos se colocan en el resumen del plan (Figura 4), para luego entregar el producto terminado con su respectivo resumen de planificación.

Resumen del Plan del Proyecto				
Estudiante			Fecha	
Programa			Programa #	
Instructor			Lenguaje	
Tiempo en la Fase (min.)	Plan	Actual	A la fecha	A la fecha %
Planeación				
Diseño				
Codificación				
Compilación				
Pruebas				
Postmortem				
Total				
Defectos Encontrados		Actual	A la fecha	A la fecha %
Planeación				
Diseño				
Codificación				
Compilación				
Pruebas				
Total Desarrollo				
Defectos Removidos		Actual	A la fecha	A la fecha %
Planeación				
Diseño				
Codificación				
Compilación				
Pruebas				
Total Desarrollo				
Después de Desarrollo				

Figura 5. Resumen de la Planificación de PSP

PSP contiene siete niveles cada uno de ellos divididos en procesos descritos por medio de “scripts” que detallan los pasos a seguir y los criterios de entrada y salida de cada paso como lo indica la Figura 5.

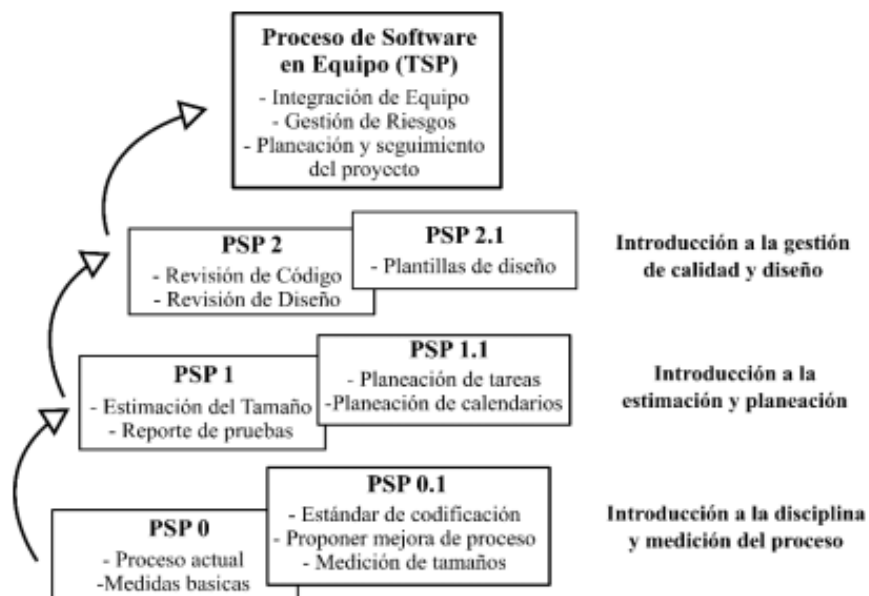


Figura 5. Niveles de Madurez de PSP Fuente: (Delgadillo et al.,2016)

PSP0 y PSP0.1 – Línea Base del Proceso Personal

La línea base del proceso personal establece una base inicial que incluye mediciones de tiempo, de datos de defectos y un formato de informes. PSP0 es el proceso existente que usan los Ingenieros para escribir software, es decir el proceso actual. PSP0 pasa a PSP0.1 al implementar código estándar, medición del tamaño y mejoramiento al proceso. Este paso inicial posee una estructura que contiene:

- 1- Planeamiento – Planear el trabajo y documentar el plan
- 2- Diseño – Diseñar el programa
- 3- Codificación – Implementar el diseño
- 4- Compilación – Compilar y eliminar los defectos encontrados en el programa
- 5- Prueba – Probar y eliminar todos los defectos encontrados en el programa
- 6- Post mortem – Registrar tiempo actual, datos de los defectos y del tamaño del plan

PSP1 y PSP1.1 – Gestión de proyectos personales

En PSP1 y PSP1.1 se realizan estimaciones de tamaño y esfuerzo utilizando el método PROBE (Proxy Based Estimating - Estimación basada en Proxy), en el cual los Ingenieros hacen una estimación inicial empleando el tamaño relativo de un proxy, para la conversión del tamaño relativo del proxy a LOC (líneas de código) utilizan los datos históricos. Los proxies son los objetos como clases, funciones o procedimientos.

Al emplear PROBE, se deben identificar los objetos a desarrollarse, es decir determinar el tipo o categoría general del componente (computacional, input / output, control lógico, etc) y el tamaño relativo (very small, small, medium, large y very large); posterior a esto mediante una tabla de rango de tamaño apoyada en los datos de tamaño histórico del proxy convertir el tamaño relativo a líneas de código (LOC). PSP1 agrega planeamiento a los pasos de PSP0 y PSP1.1 introduce el seguimiento del estado y la planificación programada.

PSP2 y PSP2.1 – Gestión de Calidad Personal

En PSP2 y PSP2.1 se tiene como objetivo implementar métodos de administración de calidad que permitan encontrar y eliminar defectos antes de realizar la primera

compilación. Estos métodos son: revisión de códigos, técnicas de verificación de diseño (revisión, notación, plantillas), medidas para administrar la calidad del proceso y del producto.

En PSP la medida “yield” representa el porcentaje de defectos encontrados y eliminados antes de realizar la primera compilación, el fin de PSP2 es guiar a los ingenieros hacia la obtención de un “yield” de 100%. En PSP2 se realiza el planeamiento de calidad que incluye la estimación del número de defectos a encontrarse en cada etapa, agrega a PSP1 diseño personal y revisiones de código.

PSP3 – Proceso Personal Cíclico

PSP3 está enfocado a escalar eficazmente sin perjudicar la calidad o productividad, introduce diseño y revisión de alto nivel. Con PSP3 los ingenieros realizan planeamiento cíclico, es decir particionan un proyecto en una serie de ciclos de PSP2.1, los cuales son integrados y probados teniendo costos menores debido a la alta calidad de los programas producidos con PSP2.1. El PSP proporciona un grupo de medidas (básicas y de calidad) para evaluar desde distintas perspectivas la calidad de los programas. Ver en Tabla 3.

Tabla 3. Medidas de PSP

Medidas Básicas de PSP	Tiempo de desarrollo	Se evalúa el número de minutos de cada etapa de PSP
	Tamaño	Son el número de líneas de código que sirven para estimar el tiempo de desarrollo
	Defectos	Cambios que deben realizarse en el diseño o codificación para la correcta compilación del programa.
Medidas de Calidad de PSP	Densidad de defectos	Se refiere a los defectos nuevos o modificados por línea de código (KLOC) encontrados en un programa.
	Porcentaje de revisión	Los Ingenieros reúnen los datos de sus revisiones y determinan qué rápido se pueden revisar los programas para encontrar todos o la mayoría de los defectos.

	Porcentaje de Tiempo de Desarrollo TSP usa 3 porcentajes en la evaluación de procesos:	1) Tiempo de diseño respecto del tiempo de codificación. 2) Tiempo de revisión del diseño respecto del tiempo de diseño. 3) Tiempo de revisión del código respecto del tiempo de codificación.
	Porcentaje de Defectos	Compara los defectos encontrados en una etapa respecto de otra. Son defectos encontrados en la revisión del código dividida en defectos encontrados en la compilación y defectos encontrados en la revisión del diseño.
	Defectos por Hora	Medida que se puede usar para guiar el planeamiento personal.

TEAM SW PROCESS (TSP)

Según Chavarría, Espejo, Oré, Bayona, y Pastor (2016) TSP es un proceso de desarrollo que se realiza a través de ciclos de desarrollo, mismos que contienen: un lanzamiento (planificación) y un postmortem (proceso de cierre). TSP se basa en siete principios (ver Figura 6) que permiten que los proyectos sean entregados con el tiempo, presupuesto y calidad establecidos. Aplicar la metodología TSP mejora la estimación del esfuerzo y tamaño del software, así como también reduce los defectos en las distintas fases del ciclo de vida del software.



Figura 6. Principios de TSP

TSP tiene como objetivo ayudar a los equipos de ingeniería a desarrollar productos de calidad cumpliendo los costos y tiempo establecido, conjuntamente TSP va a permitir optimizar la actuación del equipo a lo largo de todo el proyecto.

Todos los miembros de los equipos deberán tener un entrenamiento previo en PSP; para conformar los equipos en TSP se lleva a cabo la tarea denominada “Lanzamiento de Equipo” en la cual se realiza la planificación del proyecto para los próximos 3/4 meses, en la Figura 7 se representa el proceso de lanzamiento, cada lanzamiento posee un script con la descripción detallada de la actividad.

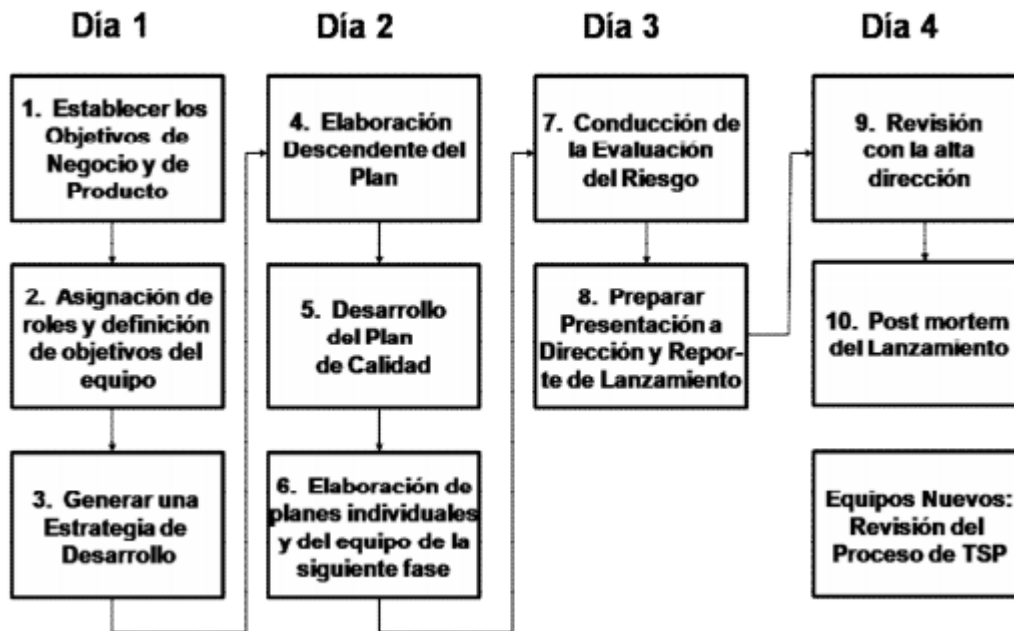


Figura 7. Proceso de Lanzamiento de TSP

Los equipos de TSP pueden realizar un plan de calidad durante el lanzamiento del equipo sólo si cuentan con 11 datos históricos de los defectos, caso contrario se debe usar la guía de planeamiento de la calidad que propone TSP. El encargado de comprobar que se cumpla lo establecido en el plan de calidad es el Gerente de Calidad.

TSP proporciona una estructura que permite aplicar PSP en equipos que desarrollan software de alta calidad. Esta metodología ha sido diseñada para equipos de 2 a 20 miembros, sin embargo, sus resultados son más eficaces en equipos de 3 a 12 miembros.

1.2.6. Estándares De Calidad De Software.

ISO 9003:2014

Según Callejas, Alarcón, y Álvarez (2017) la ISO 9003:2014 es un conjunto de estándares utilizados para el suministro, desarrollo y mantenimiento del software, cuya finalidad es proveer una guía de aplicación de la norma 9001 permitiendo a las organizaciones demostrar que se encuentran aptas para el desarrollar de software de calidad. Esta norma identifica los problemas a tratarse y es independiente de la tecnología, los modelos del ciclo de vida, los procesos de desarrollo, las actividades y la estructura organizativa utilizada por una organización.

La ISO 90003:2014 puede ser aplicada en todo software que: 1) Forma parte de un contrato comercial con otra organización, 2) Está incrustado en un producto de hardware, 3) Es usado para apoyar los procesos de una organización y 4) Está relacionado a servicios de software. EL software debe involucrar al menos una de las características, no necesariamente todas.

Las actividades específicas por considerarse al momento de implementar el sistema de gestión de calidad en un software se encuentran establecidas en los cinco capítulos que propone la norma: 1) Sistema de Gestión de la Calidad, 2) Responsabilidad de la Dirección, 3) Gestión de los Recursos, 4) Realización del Producto y 5) Medida, Análisis y Mejora.

La ISO 9001: 2008 especifica los requisitos para un sistema de gestión de calidad en organizaciones que tienen como objetivo la mejora de la satisfacción del cliente. Los requisitos de la ISO 9001:2008 pueden ser aplicables a todas las organizaciones, independientemente del tipo, tamaño y producto proporcionado.

La estructura de norma ISO 9001:2008 se visualiza en la Figura 2, la estructura completa se encuentra en el ANEXO A.



Figura 2. Estructura ISO 9001:2008

ISO / IEC 12207: 2017

La ISO 12207: 2017 establece un marco de referencia para los procesos del ciclo de vida del software que contiene procesos, actividades y tareas que deberían aplicarse durante la adquisición, suministro, desarrollo y mantenimiento del software. Este estándar facilita un proceso que puede utilizarse para definir, controlar y mejorar los procesos del ciclo de vida del software, además posee terminología bien definida que puede ser referenciada por la industria del software. No es obligatorio incluir todos los procesos de la norma, se puede seleccionar un conjunto de procesos adecuados al proyecto u organización.

Las actividades, procesos y tareas de esta norma pueden aplicarse para la adquisición de sistemas que contengan software, también se puede aplicar en conjunto con la ISO / IEC / IEEE 15288: 2015, Sistemas de ingeniería de sistemas y procesos del ciclo de vida del sistema.

El objetivo de esta norma internacional es brindar un conjunto de procesos definidos que facilitan la comunicación entre los personajes (proveedores, adquirientes, operadores, etc) que intervienen en el ciclo de vida del software.

La ISO 12207 agrupa las actividades que se pueden realizar durante el ciclo de vida del software como se muestra en la Figura 3 en siete grupos de procesos: a) Procesos de Acuerdo b) Procesos de Habilidad de Proyecto organizacionales c) Procesos de Proyectos d) Procesos Técnicos e) Procesos de Implementación de Software f) Procesos de Soporte de Software g) Procesos de Reutilización de Software. Cada grupo de procesos presenta el propósito y los resultados que se esperan obtener, así como también las actividades que se deben realizar para obtener esos resultados.

La norma ISO 12207:2017 establece un Modelo Referencial de Procesos (MRP) para las organizaciones que desean establecer el nivel de capacidad de sus procesos mediante la evaluación de los mismos. El propósito y los resultados(outcomes-salidas) de cada proceso se encuentran detallados en el MRP, estas salidas son utilizados para demostrar si el propósito del proceso se cumple, permitiendo a los auditores determinar la capacidad de implementación de los procesos en la organización y provee material para planificar la mejora del proceso. Este MRP ha sido concebido para evaluar procesos usando la ISO/IEC 15504-2.



Figura 3. Grupos De Procesos Del Ciclo De Vida ISO 12207.

ISO / IEC 15504 (SPICE).

SPICE (Software Process Improvement and Capability Determination) es un estándar internacional que como sus siglas lo indican se encarga de la “Mejora y Determinación de la Capacidad de Procesos de Software”, Según Espinel, Acosta y García (2017) este estándar es aplicable a cualquier organización desee conocer y mejorar la capacidad de sus procesos.

Este modelo fue diseñado para identificar los riesgos, controlar los costos y el cumplimiento de las actividades de acuerdo con el calendario, además se encarga de mejorar la eficiencia y calidad de los procesos de software. El modelo ISO/ IEC 15504, tiene como objetivo principal mejorar la calidad del software dando soporte a la planificación a través de la generación de métricas de procesos y de producto, según Solarte, Muñoz y Arias (2009) este objetivo se cumple usando una guía que abarque la medición de un proceso y sirva para la evaluación de proyectos, es decir, que al aplicar esta medición propone: el uso y manejo de métricas de calidad y la administración de datos.

Los procesos serán evaluados en cada nivel de acuerdo con sus atributos, los cuales son comunes en todos los procesos, en la Figura 4 se observan los respectivos atributos de cada nivel de capacidad de este estándar.



Figura 4. Niveles de Capacidad SPICE. (Ruvalcaba, 2004).

ITIL V3.

Information Technology Infrastructure Library (Biblioteca de Infraestructura de Tecnologías de Información), es un marco de trabajo que contiene mejores prácticas y su objetivo principal según Sanchez, Fernandez y Moratilla (2013), es la gestión de la entrega de todos los servicios de Tecnología de Información que se encuentran integrados en el negocio. Consta de una estructura iterativa, multidimensional y ciclo de vida de la estructura. El ciclo de vida de la gestión de servicios se describe en cinco libros principales los cuales son: Estrategia de servicio (ST), Diseño de servicio(SD), transición de servicio (ST), operación de servicio (SO) y mejora continua del servicio (SCI) ver Figura 5.



Figura 5. Ciclo de Vida de ITIL. Fuente: (Ricardo,2015).

ITIL puede aplicarse a todo tipo de organización (privada o pública, pequeña o grande, etc), brinda el detalle de los procesos relevantes que se deben cumplir en cualquier organización de TI, así como también procedimientos, prácticas y métodos para implementar la gestión de servicios de TI. Estos procesos se encuentran integrados en el ciclo de vida de ITIL ver Figura 6.

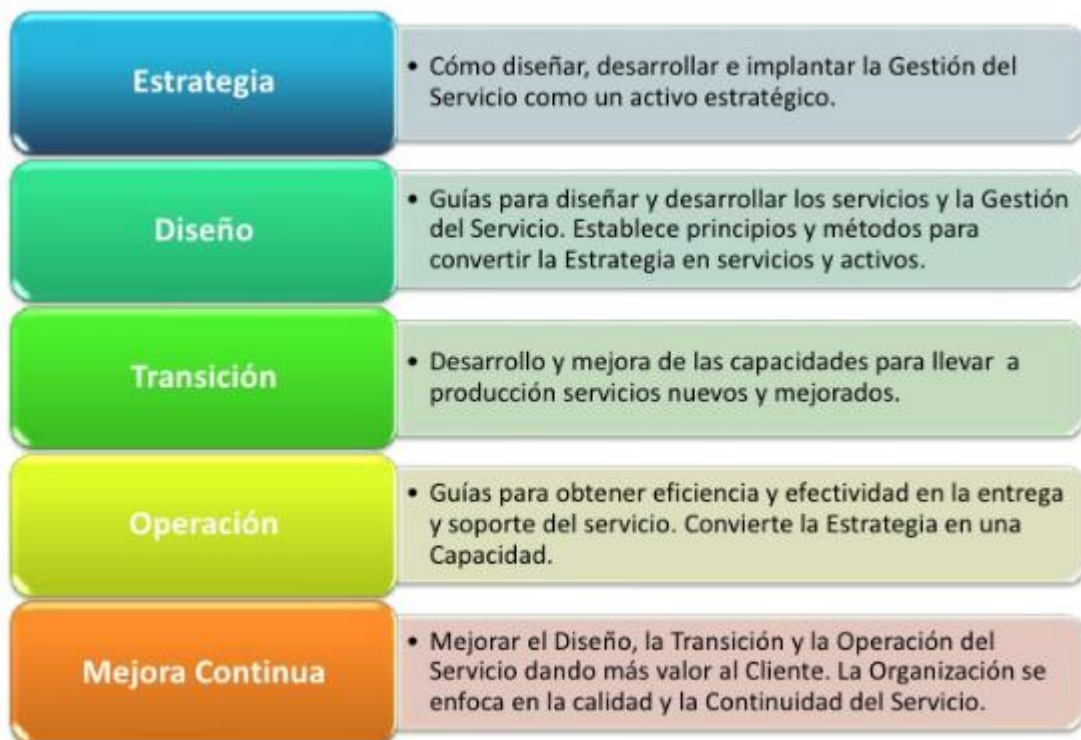


Figura 6. Descripción del Ciclo de vida de ITIL. Fuente: Tecnofor (2010).

COBIT 5.

Según Gordillo (2016) el acrónimo COBIT que en español significa (Objetivos de Control para Información y Tecnologías Relacionadas) facilita a las empresas el alcance de sus objetivos para el gobierno y la gestión de las TI corporativas, es decir, permite a las empresas mantener el equilibrio entre el uso de los recursos y la generación de beneficios.

Arisméndiz (2017) asegura que COBIT V5 se basa en la integración de varios normas y marcos de referencia como lo son: COBIT 4.1, Val IT y Risk IT, Information Technology Infrastructure Library (ITIL) y normas ISO. COBIT 5 es genérico y útil para empresas de todos los tamaños, tanto comerciales, como del sector público; se basa en 5 principios claves para el gobierno y la gestión de las TI empresariales como se muestra en la Figura 7:

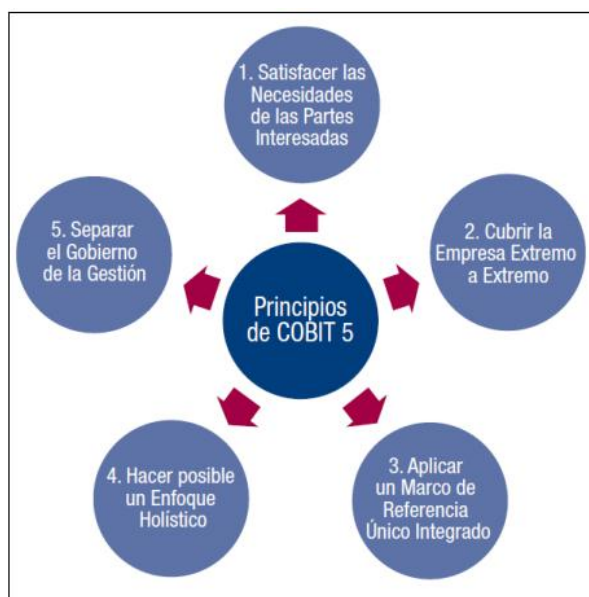


Figura 7. Principios de COBIT V5. (Armendáriz, 2017).

Al momento de realizar el diagnóstico e implementación de los niveles de capacidad de los procesos Cobit 5 toma como marco de referencia al modelo de capacidad de referencia de la ISO 15504:2, debido a que esta norma dispone de un sistema de medición único aplicable a todos los procesos como se visualiza en la Figura 8. La Guía

de autoevaluación de COBIT: Uso de COBIT 5¹ y el Modelo de Evaluación de Proceso (PAM) de COBIT: Uso de COBIT 5² de ISACA explican en detalladamente cómo realizar el proceso de evaluación.

Figura 1. Niveles de capacidades de procesos	
Nivel de proceso	Capacidad
0 (Incompleto)	El proceso no se ha implementado o no logra su propósito. En este nivel, hay evidencia escasa o nula de un logro sistemático del propósito del proceso.
1 (Realizado)	El proceso implementado logra su propósito.
2 (Gestionado)	El proceso realizado ahora se implementa de manera gestionada (planificada, supervisada y ajustada) y sus productos de trabajo se establecen, controlan y mantienen de forma apropiada.
3 (Establecido)	El proceso gestionado ahora se implementa mediante un proceso definido que es capaz de lograr los resultados del proceso.
4 (Predecible)	El proceso establecido ahora opera dentro de los límites definidos para lograr los resultados del proceso.
5 (Optimización)	El proceso predecible se mejora continuamente para cumplir con las metas del negocio, tanto actuales como proyectadas.

Figura 8. Niveles de capacidades definidos (según ISO/ IEC 15504)

1.3. BASES LEGALES

Desde el 2009 hasta la actualidad dentro de la República del Ecuador existe un documento legal titulado “NORMAS DE CONTROL INTERNO PARA LAS ENTIDADES, ORGANISMOS DEL SECTOR PÚBLICO Y PERSONAS JURÍDICAS DE DERECHO PRIVADO QUE DISPONGAN DE RECURSOS PÚBLICOS”. En dicho documento se presentan distintas normas para la administración de Tecnologías de la Información, tales como: Organización Informática, Políticas y Procedimientos, Administración de Proyectos, Desarrollo y Adquisición de Proyectos, Seguridad de la Tecnología de la Información, Monitoreo y Evaluación de procesos y servicios, Sitio Web, Servicios de Internet e Intranet, etc. Estas se encuentran distribuidas a través de códigos que van desde el número 410-01 hasta el número 410-17.

¹ ISACA, *COBIT Self-assessment Guide: Using COBIT 5*, USA, 2013, www.isaca.org/COBIT/Pages/Self-Assessment-Guide.aspx.

² ISACA, *COBIT Process Assessment Model (PAM): Using COBIT 5*, USA, 2013, www.isaca.org/COBIT/Pages/COBIT-5-PAM.aspx

CAPÍTULO II: METODOLOGÍA

2.1. DESCRIPCIÓN Y CARACTERIZACIÓN DEL LUGAR

Esta investigación se realizó en el departamento de Tecnologías de Información y Comunicación de la Pontificia Universidad Católica del Ecuador - Esmeraldas, que a través de las áreas de: soporte técnico, desarrollo de software y redes; se responsabiliza de dar soporte y brindar los medios tecnológicos adecuados para el desarrollo de las actividades de administrativos, docentes, estudiantes y área de investigación de la universidad.

2.2. TIPO DE INVESTIGACIÓN

La investigación elaborada es una investigación descriptiva cualitativa puesto que tiene como objeto describir las características fundamentales de los distintos modelos y estándares de Calidad de software y a su vez adoptar el más adecuado. Según Sampieri (2008) “La investigación descriptiva, especifica propiedades, características y rasgos importantes de cualquier fenómeno que se analice”; a través de este tipo de investigación se obtuvo los datos involucrados en la medición de un proceso y luego se analizaron meticulosamente los resultados con el fin de extraer generalizaciones significativas que permitan la mejora de este proceso.

2.3. MÉTODOS Y TÉCNICAS

Para esta investigación se implementó el método deductivo debido a que partimos de los tipos de modelos de métricas de software para seleccionar e implementar el modelo más significativo y adecuado. Adicionalmente se aplica el método analítico puesto que se separaron las metodologías de medición de software según su clasificación para luego estudiarlas en forma individual.

Se utilizó la técnica de la entrevista (ANEXO B), la cual fue aplicada al jefe del departamento de TICs y a los encargados del proceso de mantenimiento de software para la recolección de los datos mismos que sirvieron para realizar la medición.

Adicional se empleó la técnica de la observación (ANEXO C) la cual se realizó para establecer de manera detallada los pasos que intervienen en el proceso de mantenimiento de software.

2.4. POBLACIÓN Y MUESTRA (TÉCNICAS DE MUESTREO)

Para realizar la medición del proceso de mantenimiento de software, se han tomado datos de los desarrolladores (3 personas) y del jefe del Departamento de TICS; sin embargo, el total de la población lo conforma un grupo de 6 personas.

2.5. DESCRIPCIÓN Y VALIDACIÓN DEL INSTRUMENTO

En la entrevista se utilizaron preguntas abiertas las cuales sirvieron para la elección del proceso al cual se le va a aplicar las métricas de software.

La observación realizada es científica - directa; científica, debido a que la misma se realizó para establecer los detalles del proceso de mantenimiento del software; y directa, porque se desarrolló en el departamento involucrado donde se realiza el proceso.

2.6. DESCRIPCIÓN DE LAS TÉCNICAS DE PROCESAMIENTO Y ANÁLISIS DE DATOS

La información obtenida por las fichas de observación fue tabulada y analizada para corroborar la información obtenida en las entrevistas, es decir que el proceso se cumple como lo indicaron los entrevistados.

2.6.1. Análisis De Las Entrevistas Para La Evaluación Del Proceso

La información obtenida por las entrevistas como se muestra en la Tabla 4 fue rigurosamente organizada y analizada por el investigador para extraer conclusiones de la experiencia investigativa y asimismo proponer las mejoras del proceso.

Tabla 4. Análisis de las entrevistas

#	PREGUNTAS	RESPUESTAS
1	¿Cómo considera usted que se lleva a cabo el proceso de mantenimiento de software en el departamento?	Este proceso empieza con una petición o requerimiento del usuario, este a veces es formal (documentado) y otras veces es informal (teléfono, correo, etc); luego se realiza el análisis de lo que implica el cambio y posterior a esto se trabaja en el cambio.

2	<p>Cuando se solicita un cambio en un sistema. ¿se identifica el impacto de estos cambios en la organización, o con los sistemas con los que se relacione?</p>	<p>Se realiza un análisis de lo que implica el cambio, que sistemas salen afectados, si implica un cambio sobre la base de datos o solo cambio de un módulo.</p>
3	<p>¿Qué tipo de documentación existe en el proceso de mantenimiento?</p>	<p>Se realiza un registro general de los cambios, la poca documentación (física y digital) existente es sobre cambios grandes y para esto utilizan ficha sobre el requerimiento (ver ANEXO D).</p>
4	<p>¿Cómo demuestran el correcto funcionamiento de las partes modificadas del sistema?</p>	<p>Se cuenta con un ambiente de prueba, pruebas individuales de funcionalidad y pruebas con el usuario que realiza la petición.</p>
5	<p>¿Cómo realizan la migración de las actualizaciones del software al entorno del cliente?</p>	<p>Para aplicaciones Access, se cuenta con un software instalado en el cliente que detecta los cambios que difieren de la fecha actual y los actualiza en el cliente una vez que haya cerrado el sistema.</p> <p>Para los aplicativos webs se realiza la copia al ambiente de producción.</p> <p>Para las aplicaciones de visual estudio, se utiliza la herramienta “ClickOnce” que detecta e instala las actualizaciones cuando el usuario intenta abrir la aplicación.</p>
6	<p>¿Qué información es comunicada a los usuarios afectados?</p>	<p>Se realiza una capacitación personalizada a cada uno de los tipos de usuarios, se comunican a todos solo los cambios significativos.</p>

2.6.2. Proceso de Evaluación

Para la evaluación del proceso se utiliza el modelo referencial de proceso (MRP) que se encuentra contenido en el documento de la ISO 12207. De acuerdo con la Norma ISO/IEC 15504-2 el MRP establece los métodos para la evaluación de procesos, tomando de esta norma los atributos de procesos.

El modelo referencial de proceso está compuesto por: a) el propósito y, b) los resultados (outcomes) de cada uno de los procesos que forman parte del ciclo de vida del software. Para evaluar cualquiera de estos procesos se debe tomar en cuenta los atributos del proceso con relación a los resultados del proceso (outcomes).

En la Figura 9 se detallan los atributos que apoyan al desarrollo de cada nivel de capacidad.

Nivel de capacidad	Atributo de proceso (PA)
Nivel 1: Proceso Realizado	PA 1.1 Realización del proceso
Nivel 2: Proceso Gestionado	PA 2.1 Gestión de la realización PA 2.2 Gestión del producto de trabajo
Nivel 3: Proceso Establecido	PA 3.1 Definición del proceso PA 3.2 Despliegue del proceso
Nivel 4: Proceso Predecible	PA 4.1 Medición del proceso PA 4.2 Control del proceso
Nivel 5: Proceso en optimización	PA 5.1 Innovación del proceso PA 5.2 Optimización continua

Figura 9. Niveles de capacidad y atributos de procesos. Fuente: (ISO/IEC 15504, 2003)

Los atributos de proceso son comunes para todos los procesos, no son específicamente para un nivel de capacidad debido a que deben mantenerse progresivamente para establecer el nivel de capacidad del proceso. La evaluación de los atributos de proceso se realiza de acuerdo con los criterios que se indican en la figura 10.

Calificación	Rango de Cumplimiento
CI Completamente Implementado	Entre 86% y 100% se cumple el atributo en el proceso evaluado y existen las evidencias de un logro significativo.
AI Ampliamente Implementado	Entre 51% y 85% se cumple el atributo en el proceso evaluado y existen las evidencias de un logro significativo. Sin embargo pueden existir inconsistencias en algunas áreas de trabajo.
PI Parcialmente Implementado	Entre el 16% y 50% se cumple el atributo en el proceso evaluado y existe la evidencia de algún logro; pero algunos aspectos del proceso no han sido implementados por completo.
NT No Implementado	Entre 0% y 15% se cumple el atributo definido por el proceso evaluado y existe poca o ninguna evidencia.

Figura 10. Criterios de evaluación. Fuente: (ISO/IEC 15504, 2003)

Para determinar el nivel de capacidad del proceso la evaluación de los atributos de procesos debe estar referenciada de acuerdo con la Figura 11.

NIVEL DE CAPACIDAD	ATRIBUTO DE PROCESO	CALIFICACIÓN
Nivel 1	PA 1.1 Realización del proceso	AI ó CI
Nivel 2	PA 1.1 Realización del proceso	CI
	PA 2.1 Gestión de la realización	AI ó CI
	PA 2.2 Gestión de los productos de trabajo	AI ó CI
Nivel 3	PA 1.1 Realización del proceso	CI
	PA 2.1 Gestión de la realización	CI
	PA 2.2 Gestión de los productos de trabajo	CI
	PA 3.1 Definición del proceso	AI ó CI
	PA 3.2 Despliegue del proceso	AI ó CI

Figura 11. Evaluación de los atributos de procesos. Fuente: (ISO/IEC 15504, 2003)

El proceso evaluado con la Norma ISO 12207 en el departamento de TICs de la PUCE Esmeraldas es el proceso de mantenimiento de software, esta evaluación se realizó de acuerdo al modelo referencial de proceso (MRP) de la ISO 12207.

Tabla 5. Propósito y Outcomes del proceso de mantenimiento de software.

Propósito	El propósito es modificar un producto de software después de su entrega, para corregir las fallas, mejorar el rendimiento u otros atributos o adaptarlo a los cambios del entorno
Outcomes	<ol style="list-style-type: none"> 1. Se desarrolla una estrategia de mantenimiento para manejar la modificación, migración y retiro de productos de acuerdo con la estrategia de entrega. 2. Se identifica el impacto de los cambios en el sistema existente, en la organización, operaciones o interfaces; 3. Se actualiza la documentación del software del sistema afectado según sea necesario; 4. Se desarrollan los productos modificados con <u>pruebas</u> asociadas que demuestren que los requisitos no son los acordados; 5. Se migran las actualizaciones del producto al entorno del cliente; y 6. La modificación del software del sistema se comunica a todas las partes afectadas.

2.7. NORMAS ÉTICAS

Los datos o parámetros que se proporcionaron sobre el proceso de mantenimiento de software de la PUCE Esmeraldas al investigador, para la implementación de un modelo de métricas de procesos no serán divulgados bajo ningún concepto. Las citas expuestas en esta investigación se han realizado tomando en cuenta la normativa APA y son de exclusiva responsabilidad del investigador.

CAPÍTULO III: RESULTADOS

3.1. ANÁLISIS COMPARATIVO DE MODELOS Y ESTÁNDARES DE CALIDAD DE SOFTWARE

El análisis comparativo de modelos y estándares de calidad de software puede servir de apoyo a las empresas desarrolladoras de software al momento de seleccionar el Modelo o Estándar de calidad adecuado, conforme a las necesidades de la empresa; por lo cual se han tomado los criterios más relevantes de tres comparativas realizadas en otras investigaciones.

En la tabla 4 se pueden observar los criterios de comparación de los modelos y estándares de calidad de software estudiados en esta investigación.

- **Tamaño de la Organización:** Con este parámetro se determinará si el modelo o estándar es aplicable a organizaciones pequeñas, medianas o grandes.
- **Objetivo:** Para establecer el propósito del modelo o estándar.

De la comparativa de Arruzazabala, Dapozo, y Thomas (2012) se han seleccionado:

- **Niveles de madurez:** En este punto de comparación se muestran los posibles valores de nivel de madurez que pueden tomar los resultados obtenidos en la evaluación del proceso.

De la investigación de Espejo Chavarría (2016) se han seleccionado:

- **Plazos de los resultados:** Se indica si los resultados son evidentes a corto o mediano plazo.
- **Recursos de la organización para la implementación:** En este criterio se indica si se incurre en gastos adicionales.
- **Ámbito de aplicación:** se usa para indicar en donde se puede aplicar el modelo o estándar.

Tabla 6. Análisis Comparativo de Metodologías de Calidad de Software.

	Ámbito de aplicación	Objetivo	Tamaño de la Organización	Niveles de madurez / capacidad	Plazos de los resultados	Recursos de la organización para la implementación
CMMi	-Aplicable a áreas de la organización (Continuo). -Aplicable a todos los procesos de la organización (Etapas).	Determinar la capacidad y madurez de procesos, para la mejora de procesos	Empresas pequeñas (CMMi Continuo). Empresas grandes (CMMi por etapas).	CMMi Continuo 0. Incompleto 1. Realizado 2. Gestionado 3. Definido CMMi por etapas 1. Inicial 2. Gestionado 3. Definido 4. Gestionado Cuantitativamente 5. En optimización	Del nivel 1 al 2: 23 meses 2 al 3: 22 meses 4 al 4: 28 meses 4 al 5: 17 meses	Bajo al inicio (CMMi Continuo). Alto al inicio (CMMi por etapas).
PSP	A nivel personal -Aplicable a los procesos de construcción del software	Mejorar tanto los procesos como el rendimiento de los programadores.	Indiferente de tamaño debido a que es a nivel personal	PSP0 y PSP0.1 Línea base del proceso personal PSP1 Gestión de proyectos personales PSP2 Gestión de calidad personal PSP3 proceso personal cíclico	A corto plazo	Capacitador del modelo
TSP	A nivel de proyecto	Ayudar a los equipos de desarrollo a cumplir los	Cualquier tamaño de la	No presenta niveles	A corto plazo	-Capacitador en PSP (en caso de

		costos y tiempo establecido.	organización			que los equipos no estén capacitados)
ISO 9000-3	- Desarrollo de sistemas de información - Procesos del ciclo de vida - Calidad de software	Proveer una guía de aplicación de la norma 9001 permitiendo a las organizaciones demostrar que se encuentran aptas para el desarrollar de software de calidad.	Grandes y medianas	No presenta niveles	A corto plazo	
ISO 12207	Aplicable a nivel proceso o a nivel proyecto	Proporcionar una estructura común para los procesos del ciclo de vida del software con terminología bien definida que puede ser referencia por la industria de software.	Cualquier tamaño de organización	1. Realizado 2. Gestionado 3. Establecido 4. Predecible o Controlado 5. Optimizado	Corto plazo Relativo de acuerdo con los procesos que evalúa	Pocos recursos
SPICE ISO 15504	Aplicable a toda la organización o a un área de la misma.	Valoración de procesos y guía para la mejora.	Cualquier tamaño de la organización	0. Inmadura 1. Básica 2. Gestionada 3. Establecida 4. Predecible 5. Optimizada	Relativo de acuerdo con los procesos que evalúa.	Alto
ITIL v3	Aplicable a nivel organización	Facilitar la entrega de servicios de Tecnologías de la Información	Cualquier tamaño de organización	No presenta niveles	Corto plazo	Alto
Cobit 5.0	Aplicable a nivel organización	Facilitar a las empresas el alcance de sus objetivos	Cualquier tamaño de	0. Incompleto 1. Ejecutado		Alto si la empresa es pequeña

		para el gobierno y la gestión de las TI corporativas	organización	2. Gestionado 3. Establecido 4. Predecible 5. Optimizado		
--	--	--	--------------	---	--	--

En base a la información establecida en el análisis comparativo de la tabla 4 se puede determinar que los modelos y estándares de calidad de software deben ser elegidos de acuerdo a lo que necesite medir de la empresa, su tamaño y el objetivo del modelo o estándar, de esta manera: **CMMi**, puede aplicarse a un área o a la organización completa sin embargo los resultados de implementación se pueden visualizar a largo plazo; **PSP**, se aplica a nivel personal y mejora el rendimiento de los programadores pero solo se enfoca en las fases del desarrollo del software(diseño y pruebas) siendo esto una limitación en el aumento de la calidad de todo el ciclo de vida del software; **TSP**, es aplicable a equipos de desarrollo de software mejorando el desempeño tanto del equipo como el de los individuos, este requiere que los miembros del equipo se encuentren entrenados en PSP ; **ISO 90003**, aplicable a nivel de producto, tiene como objetivo brindar una guía de aplicación de la norma ISO 9001 y no cuenta con niveles de madurez ni capacidad debido a que este modelo se implementa para mejorar los procesos mas no para evaluar; **ISO 12207**, es un estándar específico para los procesos del ciclo de vida del software y puede emplearse para definir, controlar o mejorar estos procesos, puede aplicarse a nivel proyecto o a nivel proceso; **ISO 15504**, es un estándar que tiene como propósito evaluar los procesos para la mejora de la calidad de los mismos, es aplicable a nivel proceso o a nivel organización; **ITIL v3**, es un marco de referencia destinado a la gestión de la entrega de servicios de tecnología de la información integrados en el negocio; **Cobit 5.0**, conjunto de buenas práctica que facilita a las empresas el alcance de los objetivos para el gobierno y la gestión de las TI corporativas y se aplica a cualquier tipo de organización.

3.2. COMPARATIVA DE MODELOS Y ESTANDARES DE CALIDAD DE SOFTWARE CON PONDERACIONES

Para realizar una demostración de medición se ha establecido en la Tabla 6 un cuadro comparativo de los modelos de calidad de software con ponderaciones para seleccionar la metodología que más acorde se encuentre al caso PUCE Esmeraldas, para ello se establecieron 3 etiquetas que representan un valor numérico: A = 2, M = 1, B = 0.

Tabla 7. Análisis de modelos de calidad con ponderaciones

Criterios	Detalle	CMMi	PSP	TSP	ISO 90003	ISO 12207	ISO 15504	ITIL v3	Cobit 5.0
Ámbito de aplicación	¿Se aplica a nivel proceso?	A	M	B	B	A	A	B	B
Objetivo	¿Es para mejorar procesos del ciclo de vida del software?	B	M	B	A	A	M	B	B
Tamaño de la Organización	¿Es para medianas empresas?	A	A	A	A	A	A	A	A
Niveles de madurez / capacidad	¿Posee niveles de madurez / capacidad?	A	B	B	B	M	A	B	A

Plazos de los resultados	¿Los resultados son a corto plazo?	B	A	A	M	M	M	A	M
Recursos de la organización para la implementación	¿Usa pocos recursos?	M	M	M	M	A	B	B	B
TOTAL		7	7	5	6	10	8	4	5

De acuerdo con el “TOTAL” de la Tabla 6 el modelo elegido para el caso PUCE Esmeraldas es la ISO/IEC 12207, cumpliendo con casi todos los parámetros o criterios con respecto a su detalle.

3.3. EVALUACIÓN DEL PROCESO DE MANTENIMIENTO DE SOFTWARE CON LA NORMA ISO/IEC/IEEE 12207

La evaluación del proceso de mantenimiento de software se realizó en base a los resultados de las entrevistas a través del procedimiento descrito en el apartado 2.6. Una evidencia de la toma de información para la evaluación se detalla en el ANEXO D.

Tabla 8. Resultado de la evaluación del proceso de mantenimiento de software.

ATRIBUTO	Nivel - Calificación					Evaluación del proceso
	1	2	3	4	5	
PA 1.1 Realización del Proceso	AI					REALIZADO
PA 2.1 Gestión de la Realización		PI				
PA 2.2 Gestión de los Productos de Trabajo		PI				
PA 3.1 Definición de Proceso			PI			
PA 3.2 Implantación del Proceso			NT			
PA 4.1 Medición del Proceso				NT		
PA 4.2 Control del Proceso				NT		
PA 5.1 Innovación del Proceso					NT	
PA 5.2 Optimización del Proceso					NT	

Conforme a la Tabla 6 el proceso de Mantenimiento del Software de acuerdo con la Norma ISO 12207 aplicado en la PUCESE tiene un nivel de capacidad REALIZADO, es decir el proceso es implementado y parcialmente gestionado, pero no es controlado, ni medido ni mejorado.

CAPÍTULO IV: DISCUSIÓN

Entre los resultados obtenidos en esta investigación, se ha presentado un cuadro comparativo con su respectivo análisis mismo que puede ser utilizado como guía al momento de elegir de una metodología de métrica de software, además, se presenta un cuadro de evaluación en el cual de acuerdo con el atributo de proceso se selecciona el nivel de capacidad, determinando que en la PUCE Esmeraldas el proceso de mantenimiento de software posee un nivel de capacidad “REALIZADO”.

Si bien la investigación de Aguinaga Villamar (2013) titulada “Evaluación de la norma ISO 12207 en el proceso de desarrollo de software de la Empresa Logiciel Cia. Ltda.” (investigación A) realiza las evaluaciones a varios procesos de la Empresa Logiciel Cia. Ltda., en relación con el proceso de mantenimiento de software los resultados arrojados coinciden con la presente (investigación B) en los atributos de procesos: PA 1.1 Realización del Proceso, PA 2.1 Gestión de la Realización, PA 2.2 Gestión de los Productos de Trabajo y PA 3.1 Definición de Proceso con los criterios de evaluación indicados en la tabla 9. Sin embargo, discrepan en el atributo de proceso PA 3.2 Implantación del Proceso.

Tabla 9. Comparación de la evaluación del proceso de mantenimiento en distintas investigaciones.

ATRIBUTO	Investigación A Nivel - Calificación					Investigación B Nivel - Calificación				
	1	2	3	4	5	1	2	3	4	5
PA 1.1 Realización del Proceso	AI					AI				
PA 2.1 Gestión de la Realización		PI					PI			
PA 2.2 Gestión de los Productos de Trabajo		PI					PI			
PA 3.1 Definición de Proceso			PI					PI		
PA 3.2 Implantación del Proceso			PI					NT		
PA 4.1 Medición del Proceso				NT					NT	
PA 4.2 Control del Proceso				NT					NT	
PA 5.1 Innovación del Proceso					NT					NT
PA 5.2 Optimización del Proceso					NT					NT

Cabe indicar que no se pueden aplicar todas las metodologías a una misma organización debido a que estas poseen enfoques distintos y para emplear una va a depender de los recursos y cualidades de la organización.

Elaborar una investigación encaminada a la gestión de procesos y mejora de calidad promueve a las empresas a utilizar y conocer sobre los beneficios de utilizar metodologías que mejoren la calidad del software desde su creación.

Scalone (2006) en su investigación realiza un estudio comparativo de los modelos y estándares de calidad del software referenciándose en uno de estos estándares, sin embargo, en esta investigación la comparativa se realiza de acuerdo con los parámetros comunes entre las metodologías, estableciendo en sí una guía para la elección de metodologías de calidad de software a nivel proceso. Pese a lo que se ha mencionado, ambas investigaciones concuerdan con la importancia de la evaluación de la calidad en el software tanto en sus procesos como en el software.

CAPITULO V: PROPUESTA

De acuerdo con análisis realizado luego de aplicar la ISO 12207 en el proceso de mantenimiento de software, se propone integrar las actividades que la norma establece para este proceso para que se pueda dar la mejora de este. Según la ISO 12207 este proceso consta de las siguientes actividades:

a) Implementación del proceso

- Preparar, documentar y ejecutar planes y procedimientos para llevar a cabo este proceso.
- Establecer procedimientos para recibir, registrar y hacer seguimiento de los problemas y peticiones de modificación de los usuarios.

b) Análisis de Problemas y Modificaciones

- Analizar el informe del problema o petición de modificación con relación a su impacto en la organización, el sistema existente y los sistemas con que interactúa.
- El responsable de mantenimiento deberá reproducir o comprobar el problema.
- Preparar alternativas para implementar la modificación.

c) Implementación de las Modificaciones

- Análisis para determinar que documentación, unidades de software y versiones requieren ser modificadas.
- Ejecutar el proceso de desarrollo para implementar las modificaciones.

d) Revisión/Aceptación del Mantenimiento

- Revisión con la organización que autoriza las modificaciones para determinar si el sistema modificado preserva la integridad del sistema anterior.

e) Migración

- Preparar, documentar y ejecutar un plan de migración que deberá ser comunicado a los usuarios. Las actividades de ese plan deberán incluir a los usuarios.
- Se deberá llevar a cabo una revisión post-operación para evaluar el impacto al cambio del nuevo entorno.

f) Retirada de Software

- Deberá prepararse y documentarse un plan de retirada para el cese del soporte activo por parte de operarios y organización de mantenimiento.

Nota: el producto software se retirará por petición del propietario.

CAPÍTULO VI: CONCLUSIONES

Las metodologías de métricas de software son los distintos modelos y estándares de calidad que sirven para mejorar la calidad del software desde su diseño y creación hasta la implementación del mismo, estas poseen dos enfoques: a nivel de proceso y a nivel de producto; sin embargo, en esta investigación se han organizado a nivel de proceso en modelos y estándares las metodologías elegidas así como también se ha realizado una descripción detallada las características de cada una de las metodologías.

Para realizar un análisis comparativo sobre metodologías de métricas de software es necesario abastecerse de la información suficiente que permita determinar los puntos o criterios en común de estas metodologías, y así poder analizar las discrepancias entre los mismos. De acuerdo con análisis comparativo de modelos y estándares de calidad de software establecido en esta investigación se puede concluir que la elección de la metodología depende de la organización y de los recursos de esta.

La norma ISO 12207 puede ser aplicada a todos los procesos del ciclo de vida del software o a un grupo de procesos, el modelo referencial de procesos (MRP) que se encuentra establecido en esta norma para la evaluación de procesos se referencia con la norma ISO/IEC 15504-2, de la cual se utiliza los niveles de capacidad con los atributos de proceso de cada nivel. Para evaluar cualquier proceso se debe tomar en cuenta los atributos del proceso con relación a los resultados del proceso (outcomes).

Aplicar un modelo o estándar de calidad de software a nivel proceso tiene como finalidad la mejora continua del software, lo que permite el beneficio tanto de las empresas desarrolladoras como el beneficio del cliente.

CAPÍTULO VII: RECOMENDACIONES

Para organizar los distintos modelos y estándares de calidad de software se recomienda una amplia revisión bibliográfica que permita elegir para la investigación los más representativos.

Además, se recomienda investigar las características principales de todas las metodologías.

En la elaboración de un análisis comparativo íntegro es recomendable establecer los criterios comunes revisando la información de las fuentes oficiales de las metodologías.

Aunque todas las metodologías se encuentran encaminadas a mejorar la calidad del software no es recomendable aplicar cualquier metodología a una organización, puesto que cada de ellas posee un propósito distinto. Es importante que la metodología se encuentre afín a la organización.

Es recomendable en el caso PUCE Esmeraldas aplicar las actividades que establece la ISO /IEC 12207 en cuanto al proceso de mantenimiento de software ya que carece de varios niveles para llegar a su optimización, es decir implementar la propuesta (Capítulo V), con sus respectivas actividades.

REFERENCIAS BIBLIOGRÁFICAS

- Aguinaga Villamar, G. F. (2013). Evaluación de la norma ISO 12207 en el proceso de desarrollo de software de la empresa Logiciel Cia. Ltda (Bachelor's thesis, SANGOLQUI/ESPE/2013).
- Armendáriz, D. N. L. (2017). Modelo de gestión de los servicios de tecnología de información basado en COBIT, ITIL e ISO/IEC 27000. *Revista Tecnológica-ESPOL*, 30(1).
- Arruzazabala, M., Dapozo, G., & Thomas, P. (2012). Análisis comparativo de modelos de evaluación de procesos de Ingeniería de Requerimientos, II Jornadas de Investigación en Ingeniería del NEA y Países Limítrofes.
- Callejas-Cuervo, Mauro, Alarcón-Aldana, Andrea Catherine, & Álvarez-Carreño, Ana María. (2017). Modelos de calidad del software, un estado del arte. *Entramado*, 13(1), 236-250. <https://dx.doi.org/10.18041/entramado.2017v13n1.25125>
- Chavarría, Alex Espejo, Oré, Sussy Bayona, & Pastor, Carlos. (2016). Aseguramiento de la Calidad en el Proceso de Desarrollo de Software utilizando CMMI, TSP y PSP. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, (20), 62-77. <https://dx.doi.org/10.17013/risti.20.62-77>
- Chrissis, M., Konrad, M., & Shrum, S. (2010). CMMI para Desarrollo: Guía para la integración de procesos y la mejora de productos, Versión 1.3. Retrieved from [http://www.sei.cmu.edu/library/assets/whitepapers/Spanish Technical Report CMMI V 1 3.pdf](http://www.sei.cmu.edu/library/assets/whitepapers/Spanish%20Technical%20Report%20CMMI%20V%201%203.pdf)
- Constanzo, M. A., Casas, S. I., & Marcos, C. A. (2014). Comparación de modelos de calidad, factores y métricas. *Informe Científico Técnico UNPA*, 6(1), 1-36.
- Delgadillo, M. L. U., Abud-Figueroa, M. A., Peláez-Camarena, S. G., Alor-Hernández, G., y García, A. I. S. (2016). Propuesta de un modelo de integración de PSP y Scrum para mejorar la calidad del proceso de desarrollo en una MiPyME. *Research in Computing Science*, 120, 147-157.
- Enriquez, J. G., y Casas, S. I. (2014). Usabilidad en aplicaciones móviles. *Informes Científicos-Técnicos UNPA*, 5(2), 25-47.

- Figueredo Ibáñez, L. C., y Roa Rodríguez, M. I. (2013). Los sellos de calidad:¿ garantía de procesos exitosos en las organizaciones?.
- García, J. A. L., López, S. T., Pérez, P. P., & Victore, R. D. (2013). CONTROL DE LA EJECUCIÓN DE PROYECTOS BASADO EN INDICADORES Y LÓGICA BORROSA PROJECTS EXECUTION CONTROL BASED ON INDICATORS AND FUZZY LOGIC.
- García Márquez, M. (2015). Ensayo del modelo de calidad aplicado a software de minería de datos.
- Gordillo Pasquel, M. P. (2016). Propuesta de modelo de referencia para la optimización de procesos de gestión de tecnologías de la información para PyMES que proveen servicios integrados de telecomunicaciones, aplicado a la empresa INTEGRAL DATA SA (Master's thesis, Quito, 2016.).
- ISO/IEC 9001:2008 Quality management systems – Requirements. <https://www.iso.org/standard/46486.html>
- ISO/IEC 90003:2014 Software engineering -- Guidelines for the application of ISO 9001:2008 to computer software. <https://www.iso.org/standard/66240.html>
- ISO / IEC 12207: 2017 Systems and software engineering -- Software life cycle processes. <https://www.iso.org/standard/43447.html>
- ISO / IEC 15504-2: 2003 Information technology -- Process assessment -- Part 2: Performing an assessment. <https://www.iso.org/standard/37458.html>
- Luque Claveras, J. (2015). Métricas de productividad del software para la gestión de proyectos (Bachelor's thesis, Universitat Oberta de Catalunya).
- Merino, J., Caballero, I., Rivas, B., Serrano, M., y Piattini, M. (2016). A data quality in use model for big data. *Future Generation Computer Systems*, 63, 123-130.
- Osorio Beltrán, N., y Castro León, G. (2011). Gestión de calidad en desarrollo de software. *Revista de Investigación de Sistemas e Informática*, 65-69.
- Papa, M. (2012). Aseguramiento de Calidad de Software: Estudio Comparativo de Estrategias de Medición y Evaluación (Tesis de maestría). Universidad Nacional de La Plata, Argentina.

- Pérez, L. S. V., Hernández, A. M. S., & Tornés, A. F. G. (2016). EVALUACIÓN DE HERRAMIENTAS PARA EL DESARROLLO DE SISTEMAS DE INFORMACIÓN EN LA FORMACIÓN INGENIERIL. ANFEI Digital, (5).
- Perurena Cancio, L., y Moráguez Bergues, M. (2013). Usabilidad de los sitios Web, los métodos y las técnicas para la evaluación. Revista Cubana de Información en Ciencias de la Salud.
- Piedad Metaute, P., y Serna, A. (2016). Diagnostic on the appropriation of metrics in software medium enterprises of Medellin city Diagnóstico sobre la apropiación de las métricas en medianas empresas de software de la ciudad de Medellín. REVISTA ANTIOQUEÑA DE LAS CIENCIAS COMPUTACIONALES Y LA INGENIERÍA DE SOFTWARE, 6.
- Pinto, N., Tortosa, N., Cuenca Pletsch, L. R., Acuña, C., y Estayno, M. G. (2013, June). QUCO2: una herramienta para medir la calidad de las aplicaciones web. In XV Workshop de Investigadores en Ciencias de la Computación.
- Poveda, P. (2010). Implementar un sistema de gestión de calidad según ISO 9001.
- Pressman, R. S. (2010). Ingeniería del Software Un enfoque práctico, Séptima edición ed.
- Puello, Plinio, Oviedo, Stalin D, & Franco, David A. (2014). METODOLOGÍA PARA EL DIAGNÓSTICO DE PRÁCTICAS DEL MODELO PROCESO PERSONAL DE SOFTWARE. Información tecnológica, 25(2), 57-66. <https://dx.doi.org/10.4067/S0718-07642014000200007>
- Ricardo, F. (2015). Ciclo de vida de ITILG. [Imagen] Recuperado de: <http://www.bitcompany.biz/que-es-til-cursos/#.Wgfm0lTWzIV> [Consultado 12 Nov. 2017].
- Rodríguez, M., y Piattini, M. (2015). Experiencias en la industria del software- Certificación del Producto con ISO/IEC 25000. In XVIII Congreso Iberoamericano en Ingeniería de Software (pp. 1-14).
- Ruiz Constanten, Y., y Cordero Morales, D. (2013). Estimación en proyectos de software integrando los métodos de Boehm y Humphrey. Revista Cubana de Ciencias Informáticas, 23-36.

- Ruvalcaba Mara (2004). Características ISO 25010. [Imagen] Recuperado de: <https://sg.com.mx/revista/1/procesos-software#.WWOcaYiGPIU>. [Consultado 10 Jul. 2017].
- Hernández Sampieri Roberto, 2010, Metodología de la Investigación, 5° ed., Ed. El comercio, Perú, p.601.
- Sánchez Peña, J., & Fernández Vicente, E., & Moratilla Ocaña, A. (2013). ITIL, COBIT and EFQM: Can They Work Together?. *International Journal of Combinatorial Optimization Problems and Informatics*, 4 (1), 54-64. Recuperado de <http://www.redalyc.org/html/2652/265225625006/>.
- Serrano, G. L. (1996). Ingeniería de sistemas de software. Isdefe.
- SOMMERVILLE, Ian. (2005), Ingeniería del software, (7ma ed). Pearson.
- SOLARTE, G; MUÑOZ, L; ARIAS, B; (2009). MODELOS DE CALIDAD PARA PROCESOS DE SOFTWARE. *Scientia Et Technica*, XV() 375-379. Recuperado de <http://www.redalyc.org/articulo.oa?id=84916714069>.
- Tecnofor (2010). Ciclo de vida de los servicios. [Imagen] Recuperado de: <https://es.slideshare.net/tecnofor/ciclo-de-vida-de-los-servicios> [Consultado 12 Nov. 2017].

ANEXOS

ANEXO A: ESTRUCTURA DE ISO 9001:2008

1. **Ámbito**
2. **Normas para la consulta**
3. **Términos y definiciones**
4. **Sistema de gestión de la calidad**
 - 4.1. **Requisitos generales**
 - 4.2. **Documentación de requisitos**
 - 4.2.1. **Generalidades**
 - 4.2.2. **Manual de Calidad**
 - 4.2.3. **Control de los documentos**
 - 4.2.4. **Control de los registros**
 - 4.2.4.1. **Las evidencias de conformidad con los requisitos**
 - 4.2.4.2. **Las evidencias de operaciones efectivas**
 - 4.2.4.3. **Retención y disposición.**
5. **Responsabilidad de la dirección**
 - 5.1. **Compromiso de la dirección**
 - 5.2. **Enfoque al cliente**
 - 5.3. **Política de calidad**
 - 5.4. **Planificación**
 - 5.4.1. **Objetivos de la calidad**
 - 5.4.2. **Planificación del sistema de gestión de la calidad**
 - 5.5. **Responsabilidad, autoridad y comunicación**
 - 5.5.1. **Responsabilidad y autoridad**
 - 5.5.2. **Representante de la dirección**
 - 5.5.3. **Comunicación interna**
 - 5.6. **Revisión por la dirección**
 - 5.6.1. **Generalidades**
 - 5.6.2. **Información para la revisión**
 - 5.6.3. **Resultados de la revisión**
6. **Gestión de los recursos**
 - 6.1. **Provisión de recursos**
 - 6.2. **Recursos humanos**

- 6.2.1. Generalidades
- 6.2.2. Competencia, toma de conciencia y formación
- 6.3. Infraestructura
- 6.4. Ambiente de trabajo
- 7. Realización del producto
 - 7.1. Planificación de la realización del producto
 - 7.1.1. Ciclo de vida del Software
 - 7.1.2. Planificación de la calidad
 - 7.2. Procesos relacionados con el cliente
 - 7.2.1. Determinación de los requisitos relacionados con el producto
 - 7.2.1.1. Procesos relacionados con el cliente
 - 7.2.1.2. Requisitos adicionales resueltos por la organización
 - 7.2.2. Revisión de los requisitos relacionados con el producto
 - 7.2.2.1. Intereses de la organización
 - 7.2.2.2. Riesgos
 - 7.2.2.3. Representantes del cliente
 - 7.2.3. Comunicación con el cliente
 - 7.2.3.1. Generalidades
 - 7.2.3.2. Comunicación con el cliente durante el desarrollo
 - 7.2.3.3. Comunicación durante funcionamiento y mantenimiento
 - 7.3. Diseño y desarrollo
 - 7.3.1. Planificación del diseño y desarrollo
 - 7.3.1.1. Planificación del diseño y desarrollo
 - 7.3.1.2. Análisis, verificación y validación
 - 7.3.1.3. Responsabilidades y autoridades
 - 7.3.1.4. Interfaces
 - 7.3.2. Elementos de entrada para el diseño y desarrollo
 - 7.3.3. Resultados del diseño y desarrollo
 - 7.3.4. Revisión del diseño y desarrollo
 - 7.3.5. Verificación del diseño y desarrollo
 - 7.3.6. Validación del diseño y desarrollo
 - 7.3.6.1. Validación
 - 7.3.6.2. Pruebas
 - 7.3.7. Control de los cambios del diseño y del desarrollo

- 7.4. Compras
 - 7.4.1. Proceso de compras
 - 7.4.1.1. Proceso de compras
 - 7.4.1.2. Control de los productos comprados
 - 7.4.2. Información de las compras
 - 7.4.3. Verificación de los productos comprados
 - 7.5. Producción y prestación de servicios
 - 7.5.1. Control de la producción y de la prestación de servicios
 - 7.5.1.1. Producción y prestación de servicios en el soporte lógico
 - 7.5.1.2. Construcción y versiones
 - 7.5.1.3. Reproducción
 - 7.5.1.4. Entrega
 - 7.5.1.5. Instalación
 - 7.5.1.6. Funcionamientos
 - 7.5.1.7. El mantenimiento
 - 7.5.2. Validación de los procesos de la producción y de la prestación del servicio
 - 7.5.3. Identificación y trazabilidad
 - 7.5.4. Propiedad del cliente
 - 7.5.5. Preservación del producto
 - 7.6. Control de los dispositivos de seguimiento y de medición
8. Medición, análisis y mejora
 - 8.1. Generalidades
 - 8.2. Seguimiento y medición
 - 8.2.1. Satisfacción del cliente
 - 8.2.2. Auditoria interna
 - 8.2.3. Seguimiento y medición de los procesos
 - 8.2.4. Seguimiento y medición del producto
 - 8.3. Control del producto no conforme
 - 8.4. Análisis de los datos
 - 8.5. Mejora
 - 8.5.1. Mejora continua
 - 8.5.2. Acción correctiva
 - 8.5.3. Acción preventiva

9. Bibliografía

ANEXO B: ENTREVISTA

Fecha:

Nombre del Entrevistado:

OBJETIVO: Conocer toda la información sobre el proceso de mantenimiento de software

1. ¿Cómo considera usted que se lleva a cabo el proceso de mantenimiento de software en el departamento?

2. Cuando se solicita un cambio en un sistema ¿Se identifica el impacto de estos cambios en la organización, o con los sistemas con los que se relacione?

3. ¿Qué tipo de documentación existe en el proceso de mantenimiento?

4. ¿Cómo demuestran el correcto funcionamiento de las partes modificadas del sistema?


5. ¿Cómo realizan la migración de las actualizaciones del software al entorno del cliente?

6. ¿Qué información es comunicada a los usuarios afectados?

ANEXO C: FICHA DE OBSERVACIÓN

FICHA DE OBSERVACIÓN	
FICHA N°	Fecha y Duración:
ELABORADO POR:	
LUGAR:	
LO OBSERVADO	DETALLE

ANEXO D: EVIDENCIAS DE LAS ENTREVISTAS



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

Sede Esmeraldas *

Requerimiento de Cambio ReC – Aplicación de Admisiones

Versión	Fecha	Quién	Comentario sobre cambio adjunto
1	2016/10/26	María E. Vélez	Levantamiento del requerimiento

Requerimiento Proyecto TIC			
Nombre del proyecto	Actualización de sistema de admisiones		
Tipo de proyecto	Requerimiento Funcional	Estado	abierto
Nombre del solicitante	María E. Vélez	Departamento	Financiero
Sistemas involucrados	Sistemas de matrícula y aranceles	Prioridad	alta
Plazo	Octubre 2016	Código	FIN-001

Descripción y argumentación sobre el requerimiento

El régimen académico establece que los estudiantes egresados deben matricularse después de un semestre, el sistema de aranceles actualmente no detecta dicha especificación y puede darse el caso de que alguien se matricule por desconocimiento. La situación antes expresada puede originar una multa de 10 salarios mínimos a la sede.

Se solicita que la programación en el sistema de aranceles universitarios detecte cuando un estudiante egresado se matricule antes de la fecha correspondiente al periodo académico.

Alcance e hitos

- Generación de informes de estudiantes egresados matriculados
- Cumplir con lo establecido en el Régimen Académico para las universidades y evitar las multas a la sede.

Descripción de los procesos afectados con BPMN

Requerimientos	
Funcionales	No funcionales
<ul style="list-style-type: none"> Cumplimiento de normativa Listado de estudiantes egresados por año académico 	<ul style="list-style-type: none"> Todos los egresados de las diferentes unidades académicas

Riesgo / Impacto al no implementar

Matricular por error a estudiante egresado, que se denuncie y el CES multe a la sede con 10 salarios mínimos.

Crterios de aceptación - GO / noGO
Condiciones mínimas para poder poner en producción el cambio requerido

- Informe de estudiantes egresados matriculados hasta la fecha

Responsable de mantenimiento una vez implementado el cambio

- Dirección Financiera - Recaudación

Comentarios adicionales