



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

**VICERRECTORADO DE INVESTIGACIÓN Y  
VINCULACIÓN CON LA COLECTIVIDAD**

**MAESTRÍA EN INGENIERÍA DEL SOFTWARE**

**III PROMOCIÓN**

**TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE  
MAGISTER EN INGENIERÍA DEL SOFTWARE**

**TEMA:**

**MODELO DE CALIDAD PARA LA GESTIÓN DE PROCESOS  
DE DESARROLLO DE SOFTWARE APLICADO AL SISTEMA  
INFORMÁTICO EN USO DE LA UNIVERSIDAD TÉCNICA DE  
MACHALA.**

**AUTORA:                   ING. MIRIAM FAREZ ARIAS**

**DIRECTOR:                ING. LUCAS GARCÉS MGS.**

**LATACUNGA**

**2015**

**UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE**

**MAESTRÍA EN INGENIERÍA DE SOFTWARE**

**CERTIFICADO**

**ING. LUCAS ROGELIO GARCÉS GUAYTA MGS.**

Que en el trabajo titulado “**MODELO DE CALIDAD PARA LA GESTIÓN DE PROCESOS DE DESARROLLO DE SOFTWARE APLICADO AL SISTEMA INFORMÁTICO EN USO EN LA UNIVERSIDAD TÉCNICA DE MACHALA.**”, realizado por la Ing. Miriam Rocio Farez Arias, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la Universidad de las Fuerzas Armadas ESPE Extensión Latacunga.

Debido a que constituye un trabajo de excelente contenido científico que coadyuvará a la publicación de conocimientos y al desarrollo profesional. Se recomienda su publicación.

El mencionado trabajo constara de un empastado y un Disco Compacto el cual contiene los archivos en formato PDF. Autorizan a la Ing. Miriam Rocio Farez Arias que lo entreguen al **Ing. José Gioberty Bucheli Andrade** Jefe de Investigación y Vinculación con la Colectividad de la ESPE Extensión Latacunga.

Latacunga, 30 de julio del 2015

---

Ing. Lucas Garcés MGS.

**DIRECTOR DE TESIS**

**UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE**

**MAESTRÍA EN INGENIERÍA DE SOFTWARE**

**DECLARACIÓN DE RESPONSABILIDAD**

**Yo, MIRIAM ROCIO FAREZ ARIAS**

**DECLARO QUE:**

El proyecto de grado denominado “**MODELO DE CALIDAD PARA LA GESTIÓN DE PROCESOS DE DESARROLLO DE SOFTWARE APLICADO AL SISTEMA INFORMÁTICO EN USO EN LA UNIVERSIDAD TÉCNICA DE MACHALA.**”, ha sido desarrollado en base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan el pie de las páginas correspondiente, cuyas fuentes se incorporan en la bibliografía. Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Latacunga, 30 de Julio del 2015

---

**Ing. Miriam Rocio Farez Arias**

**C.C.: 0703813345**

**UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE**

**MAESTRIA EN INGENIERIA DE SOFTWARE**

**AUTORIZACIÓN**

Yo, **MIRIAM ROCIO FAREZ ARIAS**

Autorizo a la Universidad de las Fuerzas Armadas ESPE Extensión Latacunga, la publicación en la biblioteca virtual de la Institución, del trabajo denominado:

**“MODELO DE CALIDAD PARA LA GESTIÓN DE PROCESOS DE DESARROLLO DE SOFTWARE APLICADO AL SISTEMA INFORMÁTICO EN USO EN LA UNIVERSIDAD TÉCNICA DE MACHALA”**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

---

**Ing. Miriam Rocio Farez Arias**

**C.C.: 0703813345**

## AGRADECIMIENTO

Agradezco infinitamente a mi familia por el apoyo incondicional que me han brindado durante este recorrido quienes han sido el pilar fundamental para seguir adelante cumpliendo cada uno de mis objetivos.

A la Universidad de las Fuerzas Armadas Espe por brindarme la oportunidad en ser parte de esta prestigiosa Institución.

Al director de tesis Ing. Lucas Garcés quién con sus conocimientos me guio durante el proceso de desarrollo y quién con su gestión hizo posible concluir satisfactoriamente este objetivo.

A todos Gracias

Miriam

## **DEDICATORIA**

Con infinito amor dedico este trabajo a mis padres, Eladio e Inés, a mis hermanas Nancy, Lila, Jessenia, Julissa y a toda mi familia.

A mi esposo Miguel Ángel y a mi hijo quienes han sido fuente de motivación durante este proceso.

Con todo mi cariño esta tesis se las dedico a ustedes

## ÍNDICE GENERAL

<b>PORTADA.....</b>	<b>i</b>
<b>CERTIFICADO .....</b>	<b>ii</b>
<b>DECLARACIÓN DE RESPONSABILIDAD .....</b>	<b>iii</b>
<b>AUTORIZACIÓN.....</b>	<b>iv</b>
<b>AGRADECIMIENTO .....</b>	<b>v</b>
<b>DEDICATORIA .....</b>	<b>vi</b>
<b>ÍNDICE GENERAL .....</b>	<b>vii</b>
<b>ÍNDICE DE TABLAS.....</b>	<b>xi</b>
<b>ÍNDICE DE FÍGURAS.....</b>	<b>xii</b>
<b>RESUMEN.....</b>	<b>xiii</b>
<b>ABSTRACT .....</b>	<b>xiv</b>
<b>CAPÍTULO I.....</b>	<b>1</b>
<b>1. DEFINICIÓN DEL PROBLEMA .....</b>	<b>1</b>
1.1 Introducción .....	1
1.2 Justificación e Importancia .....	2
1.3 Planteamiento del problema.....	2
1.4 Formulación del problema .....	3
1.5 Objetivo general .....	3
1.6 Objetivos específicos .....	3
1.7 Formulación de hipótesis .....	3
1.7.1 Hipótesis de Investigación .....	3
1.7.2 Identificación y clasificación de variables .....	3
1.8 Conclusiones .....	4

<b>CAPÍTULO II .....</b>	<b>5</b>
<b>2. MARCO TEÓRICO .....</b>	<b>5</b>
2.1 Introducción .....	5
2.2 Antecedentes Históricos.....	6
2.2.1 Evolución de estándares de calidad en el proceso de desarrollo de software .....	6
2.3 Antecedentes Contextuales y Referenciales.....	12
2.3.1 Caracterización Gnoseológica del proceso de desarrollo de software.....	12
2.3.2 Caracterización gnoseológica de los estándares de calidad para el proceso de desarrollo de software.....	22
2.4 Antecedentes Contextuales .....	38
2.5 Conclusiones .....	41
 <b>CAPÍTULO III.....</b>	 <b>42</b>
<b>3. PROPUESTA DEL MODELO DE CALIDAD PARA LA GESTIÓN DE PROCESOS DE DESARROLLO DE SISTEMA DE LA UNIDAD DE SISTEMAS DE LA UNIVERSIDAD TÉCNICA DE MACHALA.....</b>	<b>42</b>
3.1 Introducción .....	42
3.2 Proceso General del desarrollo de Software .....	42
3.3 Selección de los modelos y estándares de calidad de sistema para el proceso de desarrollo de software. ....	44
3.4 Análisis Comparativos de los modelos o estándares de calidad de sistema .....	51
3.5 Procesos de desarrollo de software en la Unidad de Sistemas de la Universidad Técnica de Machala. ....	53
3.5.1 Metodología del proceso de desarrollo de la Unidad de Sistemas de la Universidad Técnica de Machala. ....	53
3.6 Características de la propuesta del modelo de calidad para el proceso de desarrollo de software .....	56
3.6.1 Objetivo del modelo de calidad.....	56
3.6.2 Propuesta del modelo de calidad.....	56
3.6.3 Estructura del modelo de calidad .....	61



3.6.2.3 Actividades.....	62
3.7 Conclusiones .....	69
<b>CAPÍTULO IV .....</b>	<b>70</b>
<b>4. IMPLEMENTACIÓN DEL MODELO .....</b>	<b>70</b>
4.1 Introducción .....	70
4.2 Aplicación del modelo de calidad propuesto en la fase de requisitos.....	70
4.3 Procesos principales del ciclo de vida.....	71
4.3.1 Desarrollo de requisitos.....	71
4.4 Conclusiones .....	74
<b>CAPÍTULO V.....</b>	<b>75</b>
<b>5. RESULTADOS DE LA APLICACIÓN.....</b>	<b>75</b>
5.1 Introducción .....	75
5.2 Procesamiento de los resultados.....	75
5.2.1 Análisis y resultados del instrumento aplicado a los desarrolladores .....	75
5.2.2 Análisis y resultados de las encuestas aplicadas a los usuarios del sistema informático de la Unidad de Sistemas.....	76
5.3 Prueba de hipótesis con Chi cuadrado .....	77
5.3.1 Planteamiento de la hipótesis .....	77
5.3.2 Estadística de contraste de hipótesis .....	77
5.3.3 Cálculo del valor Chi-cuadrado .....	78
5.3.4 Comparación entre el valor esperado y el valor critico.....	101
5.4 Conclusión de los resultados .....	101
<b>CAPÍTULO VI.....</b>	<b>103</b>
<b>6. CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>103</b>
6.1 Conclusiones .....	103
6.2 Recomendaciones.....	104

<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>105</b>
<b>ANEXOS .....</b>	<b>108</b>
<b>ANEXO 1:</b> Cuestionario para la Unidad de Sistemas	
<b>ANEXO 2:</b> Especificación de Requisitos	
<b>ANEXO 3:</b> Documento Estándar para Arquitectura de Software IEEE-1471-2000	
<b>ANEXO 4:</b> Cuestionario para desarrolladores y jefe de la Unidad de Sistemas	
<b>ANEXO 5:</b> Cuestionario para desarrolladores	

## ÍNDICE DE TABLAS

<b>Tabla 3.1.</b> Puntaje de los criterios a evaluar para seleccionar el modelo o estándar.....	53
<b>Tabla 3.2</b> Niveles de madurez de la Norma ISO 15504.....	57
<b>Tabla 4.1</b> Requerimientos del Software .....	71
<b>Tabla 4.2.</b> Plan de pruebas .....	73
<b>Tabla 4.3.</b> Pruebas de calificación del sistema.....	74
<b>Tabla 5.1.</b> Resultados de aprobación de los desarrolladores y Jefe de sistemas del modelo de calidad.....	76
<b>Tabla 5.2.</b> Variable Independiente Modelo de calidad.....	78

## ÍNDICE DE FIGURAS

<b>Figura 2.1.</b> Proceso del Desarrollo de software .....	17
<b>Figura 2.2.</b> Modelo Secuencial Lineal .....	19
<b>Figura 2.3.</b> Modelo en Espiral.....	20
<b>Figura 2.4.</b> Procesos evaluados por SCE (Software CapabilityEvaluation) .....	29
<b>Figura 2.5.</b> Niveles de procesos PSP .....	32
<b>Figura 2.6.</b> Estructura del modelo TSP .....	33
<b>Figura 2.7.</b> Organigrama funcional de la Unidad de Sistemas de la Universidad Técnica de Machala, 2005 .....	38
<b>Figura 3.1.</b> Ciclo de vida de desarrollo de software.....	44
<b>Figura 3.2.</b> Niveles de capacidad y procesos del CMMI.....	45
<b>Figura 3.3.</b> Estructura del estándar ISO 15504.....	50
<b>Figura 3.4.</b> Fase de especificación.....	55
<b>Figura 3.5.</b> Fase de diseño.....	55
<b>Figura 3.6.</b> Procesos de los niveles de madurez de la norma ISO 15504.....	58
<b>Figura 3.7.</b> Procesos de la norma ISO 12207:2008.....	59
<b>Figura 3.8.</b> Procesos de la norma ISO 12207 a implementar en el departamento de sistemas de la Universidad Técnica de Machala.....	60
<b>Figura 3.9.</b> Propuesta del modelo de calidad para el proceso de desarrollo de sistema UTMACH.....	61
<b>Figura 4.1.</b> Área considerada para la implementación del modelo.....	70
<b>Figura 5.1.</b> Criterio de aceptación del modelo de calidad en La Unidad de Sistemas de la UTMACH .....	76

## **RESUMEN**

La calidad del sistema es un factor muy importante a lo largo del ciclo de vida del desarrollo de sistema y en base a esta importancia se pretende con la presente investigación dar a conocer algunos modelos o estándares enfocados al proceso de desarrollo de sistema, pretendiendo dar una solución a la insatisfacción de los usuarios que provoca el no desarrollar sistema perfilado a la calidad, tomando como referencia la Unidad de Sistemas de la Universidad Técnica de Machala y bajo el proceso investigativo que se realizó se detectó que en la Unidad no existe un orden de desarrollo, las diversas planificaciones que se efectúan muchas de las veces quedan fuera del alcance del equipo de trabajo. En la presente investigación se inicia con una investigación bibliográfica en lo que respecta al proceso de desarrollo de sistema donde se determina que desde años atrás no madura el proceso de desarrollo y esto ocasiona que los integrantes de un equipo de sistema se dediquen solo a la codificación y corregir en el proceso si existe alguna falla, debido a esta circunstancia y estas preliminares detectadas se pretende en base a un análisis comparativo de los diversos modelos o estándares aplicados a la calidad de sistema crear un entorno de trabajo que permita a la Unidad de Sistemas mejorar el proceso de desarrollo de sistema, formando en sí el modelo de calidad de sistema que se acoplará a la Unidad de Sistemas.

### **PALABRAS CLAVE:**

- **SOFTWARE - ESTÁNDARES DE CALIDAD**
- **DESARROLLO DE SOFTWARE**
- **SOFTWARE - NORMAS ISO**
- **SOFTWARE - NORMAS IEEE**
- **UNIVERSIDAD TÉCNICA DE MACHALA**

## ABSTRACT

System quality is a very important throughout the life cycle of system development factor and based on this important and with this research to present some models or standards focused on the system development process. Pretending to give a solution about the user dissatisfaction that causes no profiling to develop quality system. And taking to with reference to the systems unit of the Technical University of Machala and under the investigative process that took place it was detected that there is no order of development in the Unit, the various schedules that are performed many times outside the scope of work team. In the present study it starts with a literature research with regard to the process of system development where it is determined that from years ago not mature development process and this causes the members of a system team engaged only encoding and correct the process if there is a fault. Due to this fact and these preliminary detected based on a comparative analysis of the various models or standards applied to the quality of system to create a working environment that allows Systems Unit improve the system development process. Forming itself the system quality model that will engage the Systems Unit.

### WORDS KEY:

- **SOFTWARE - QUALITY STANDARDS**
- **SOFTWARE DEVELOPMENT**
- **SOFTWARE - ISO STANDARDS**
- **SOFTWARE - IEEE**
- **TECHNICAL UNIVERSITY MACHALA**

## **CAPÍTULO I**

### **1. DEFINICIÓN DEL PROBLEMA**

#### **1.1 Introducción**

En este capítulo se describe los problemas encontrados al corroborar que no se utiliza normativas para el desarrollo de sistema y que se continúa desarrollando sin considerar ciertos estándares que permiten mejorar el desempeño tanto del personal como del producto que se desarrolla.

En la ciudad de Machala Provincia de El Oro se encuentra la Universidad Técnica de Machala la cual presta los servicios de Educación Superior conformada por un alrededor de 12000.00 estudiantes por lo que por la gran demanda de datos de cada uno se crea la Unidad de Sistemas para la Universidad, el que ha venido trabajando por un lapso de 10 años en desarrollo del Sistema Integrado que consta de módulos contables, académico, financiero.

La forma en cómo comenzó el desarrollo ha venido provocando un sinnúmero de problemas con las autoridades, las prácticas que se utilizan durante el proceso de construcción presentan diferentes inconformidades tanto en la estimación de tiempo, herramientas, y talento humano.

Factores como la falta de coordinación entre los desarrolladores y la jefa del departamento la forma inusual en el levantamiento de requisitos, la escasez de documentación formal.

La baja calidad con la que se desarrolla y entrega el producto sistema, son razones por las que la universidad ha tenido pérdidas económicas y la insatisfacción de los clientes (estudiantes).

## **1.2 Justificación e Importancia**

Esta investigación es necesaria para la institución ya que al contar con un modelo de calidad que garantice los procesos de desarrollo de sistema minimizará la presencia de fallas o errores que puedan dificultar la mayor parte de las operaciones de negocio en la que estén involucrados.

Debido a los diversos estándares de calidad para el desarrollo de sistema y por su gran extensión no muy aplicables a la realidad de nuestro país, se proyecta plantear un modelo de calidad que se ajuste a las necesidades de garantizar la calidad de los procesos de desarrollo de sistema de un área específica.

La importancia del tema a investigar está relacionada con un problema actual que es contar con productos de sistema de calidad, aplicable de tal forma que sus resultados aporten con estándares de calidad en el área de la Ingeniería de sistema y que la sociedad tan dependiente de las tecnologías en sus tareas cotidianas cuenten con productos con aseguramiento de su calidad y mejora continua.

Sin embargo para la Universidad Técnica de Machala sería muy importante que se desarrolle este tema ya que cuenta con una Unidad de Sistemas que se encarga de desarrollar el sistema integrado con todos los procesos que se manejan dentro de la institución, y al contar con un modelo de calidad permitiría liberar sistema enfocado a asegurar la calidad y usabilidad.

## **1.3 Planteamiento del problema**

Uno de los aspectos más descuidados en el proceso de desarrollo de sistema es la calidad, muchos de los desarrolladores se rigen a cumplir con los requisitos del usuario en la mayoría de los casos y no es suficiente, para ello hace falta definir un modelo de calidad, para todos los proyectos, es decir, estandarizado y que incluya las mejores prácticas de desarrollo bajo un modelo que asegure la mejora continua.



Una alternativa para alcanzar competitividad en la industria del software requiere desarrollar y aplicar un modelo basado en metodologías o procedimientos estándares para el planeamiento, especificación y ejecución de los procesos del software.

#### **1.4 Formulación del problema**

¿Existe una óptima gestión de procesos de desarrollo de software que utilicen normas de calidad en la Unidad de Sistemas de la Universidad Técnica de Machala?

#### **1.5 Objetivo general**

Elaborar un modelo de calidad para la gestión del proceso de desarrollo de software en la Unidad de Sistemas de la Universidad Técnica de Machala por medio de un análisis de los diferentes estándares de calidad para proponer el modelo de calidad que permita optimizar la eficiencia y eficacia en el sistema desarrollado.

#### **1.6 Objetivos específicos**

- Establecer el marco de trabajo
- Determinar la metodología de desarrollo
- Desarrollar la propuesta del modelo de calidad de sistema
- Aplicar el modelo de calidad de sistema sugerido

#### **1.7 Formulación de hipótesis**

##### **1.7.1 Hipótesis de Investigación**

Si se implementa un modelo de calidad para el sistema desarrollado en la Unidad de Sistemas de la Universidad Técnica de Machala permitirá optimizar la gestión del proceso de desarrollo de software que se aplica en esa dependencia.

##### **1.7.2 Identificación y clasificación de variables**

###### **Variable Independiente**

Modelo de calidad para el sistema desarrollado en la Unidad de Sistemas de la Universidad Técnica de Machala.

**Variable dependiente**

Gestión del proceso de desarrollo de software

**1.8 Conclusiones**

Se determina el proceso de investigación que se considerará en el presente estudio definir el problema el objetivo a conseguir y la hipótesis planteada durante la investigación.

## CAPÍTULO II

### 2. MARCO TEÓRICO

#### 2.1 Introducción

El proceso de desarrollo de software es un término tan complicado de interpretar ya que para tener un éxito total del mismo se requiere no solamente de programar o simplemente de agregar código fuente si no que vas más allá de este proceso.

Para lograr plasmar las fase que amerita el desarrollo de software dentro del ciclo de vida tales como análisis de requisitos, diseño, implementación, pruebas y mantenimiento requiere seguir un verdadero proceso de ingeniería de software donde según los estudios realizados Ingeniería de software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de software [5 ]

Y realmente se considera que la ingeniería de software trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable, que sea fiable y trabaje en máquinas reales [1]

Personas como Pete McBreen (autor de "Sistema Craftmanship") cree que el término IS implica niveles de rigor y prueba de procesos que no son apropiados para todo tipo de desarrollo de software [6].

A partir de este panorama y en el presente escenario se encuentra la Unidad de Sistemas de la Universidad Técnica de Machala donde se desarrolla software a la medida que a partir del 2008 surgió como una necesidad por la cantidad de usuarios que tiene la universidad y por cumplir con una parte de uno de los criterios de Evaluación por parte del Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior (CEAACES), tomando de estas expectativas el mejorar en el servicio tecnológico que ofrece la universidad.

## 2.2 Antecedentes Históricos

### 2.2.1 Evolución de estándares de calidad en el proceso de desarrollo de software

Según Piattini, García & Caballero [7] indica que; tradicionalmente la ingeniería del software se ha centrado en metodologías y lenguajes de programación, modelos de desarrollo y herramientas. Sin embargo y teniendo en cuenta la creciente complejidad del software, se hacía necesario incluir determinadas áreas que hoy en día son críticas para la ingeniería del software, como las infraestructuras de gestión y organización, por lo que surge la denominada ingeniería del software fundamentada en el proceso.

Tal como se destaca en múltiples estudios, actualmente la calidad de cualquier producto no puede ser asegurada simplemente inspeccionando el producto por sí mismo o desarrollando controles de calidad estadísticos. Esta afirmación se basa en que existe una correlación directa entre la calidad del proceso y calidad del producto obtenido y en consecuencia una organización no puede garantizar la entrega de productos de calidad centrandose sus programas de calidad únicamente en el producto. Durante las últimas tres décadas, el estudio de los procesos de producción de software han llevado al desarrollo de varios ciclos de vida en la ingeniería del software, por ejemplo, los modelos cascadas, evolutivos y en espiral. Estos modelos del ciclo de vida ayudan a los ingenieros y a los gestores a comprender mejor el proceso software, y determinar el orden de las actividades necesarias en la vida de un producto software.

Como indica Juran (1995) [8], ya en aquel siglo XI antes de Cristo en China se consolidó un software para controlar el desarrollo de productos artesanales con dos dependencias encargadas de la calidad de los productos:

- Uno de formular y ejecutar estándares
- Supervisión y prueba

Inclusive se anunció decretos para prohibir la venta de productos de calidad inferior y se prestó también mucha atención a la medición de longitud, capacidad y peso.

La gestión y la definición de “calidad moderna” surgen en 1924 en los Bell Telephone Laboratories, para considerar las reclamaciones de los clientes que instalaban teléfonos. Se crea el Departamento de Aseguramiento de Calidad del que formó parte W.A Shewhart, y a quién se le considera el padre del moderno aseguramiento de la calidad. Por otra parte, los japoneses después de la Segunda Guerra Mundial se enfrentaron, entre otros muchos problemas, al desafío de cambiar su reputación de productos de mala calidad. Juran señala 3 contribuciones primordiales importantes que llevarían a modificar notoriamente tal situación:

- Cursos de formación organizados por la Civil Communications Section de las fuerzas de ocupación norteamericana
- Conferencias Deming sobre control estadístico de la calidad
- Conferencias de Juran sobre gestión de la calidad.

En estos años empezaron a utilizar la familia de las Normas ISO 9000 en Europa y se desarrollan toda una abundancia de premios sobre la calidad. En los años noventa se desarrolla más aun los temas sobre calidad y surgen nuevos enfoques como los es Seis- Sigma. Para culminar el autor menciona que el siglo XX fue llamado el Siglo de la Productividad, el siglo XXI es conocido como el Siglo de la Calidad.

El uso de modelos y estándares de calidad ayudan a mejorar el proceso de desarrollo de software y esto se ha venido observando a través de la historia donde se conoce que la calidad del software no puede garantizarse centrandolo los programas de calidad únicamente en el producto software ya que este se relaciona directamente en la forma de desarrollo durante el proceso.

Frente a esta situación los modelos y normas de evaluación y mejora de procesos han ido adquiriendo un papel muy importante para la identificación, medición, integración y optimización de las buenas prácticas existentes para el desarrollo del software.

Y en los últimos años existe una gran proliferación de propuestas para la evaluación y mejora de los procesos.

Destacando Serrad y Lake (1998) [10] la gran cantidad de marcos de trabajo que puedan convertir el campo de la calidad en “una ciénaga en la que se empantanen los esfuerzos de mejora de procesos si una organización no es cuidadosa”.

A continuación se muestra como han ido apareciendo las normas y estándares de calidad enrolados con la evaluación al proceso de desarrollo de software teniendo lo siguiente:

El modelo CMMI [15] que tuvo cabida desde la década de los 80 creada por el Instituto de Ingeniería de Software (SEI, Software Engineering Institute) de la Universidad de Carnegie Mellon se ha centrado básicamente en proporcionar las bases necesarias y complementarias para mejorar el desarrollo de software tomando en cuenta a las tareas del desarrollo de software como una serie de procesos que se puede delimitar, medir y controlar sus procesos de desarrollo, este modelo ha sido creado con la función de:

- Evaluar la madurez de los procesos de desarrollo de software dentro de una organización dedicada.
- Proponer un plan de mejora de los procesos de desarrollo de software de acuerdo a una serie de niveles.

Este modelo al establecer el nivel de madurez de los procesos de desarrollo se establece en cinco niveles de capacidad, el Inicial, repetible, definido, gestionado y optimizado.

El modelo SCE (Software Capability Evaluation) según (Byrnes y Philips, 1996) [18] es el método desarrollado para evaluar los procesos software de una organización con el objetivo de determinar su capacidad.

Las áreas de aplicación del modelo SCE son:

- La selección del suministrador
- La monitorización del proceso
- La evaluación interna

El objetivo del modelo SCE es la evaluación del proceso software y en general se centra en un conjunto de procesos que se agrupan en tres categorías: procesos organizacionales, gestión de proyectos y los procesos de ingeniería.

El modelo IDEAL (McFeeley, 1996; Gremba y Myers, 1997) [19] es un marco de mejora de procesos de Ingeniería de Software al que se define un marco de ciclo de vida para la mejora de procesos.

Este modelo fue concebido originalmente como un ciclo de vida para la mejora de procesos software basado en el modelo CMM, y está compuesto por cinco fases, cada una está formada por una serie de actividades.

- Iniciación
- Diagnostico
- Establecimiento
- Actuación
- Aprendizaje

El modelo PSP (Personal Software Process) (Humphrey, 1997) [19] apoya a las empresas que van a implementar un plan de mejoras de procesos basados en un modelo como CMM, ayudando a crear personal capacitado y disciplinado en su trabajo, está basado en CMM y permite implementar las prácticas del proceso de ingeniería del software actuando en las áreas de.

- Planeación y seguimiento del proyecto
- Revisiones e inspecciones
- Proceso de Ingeniería del producto
- Enfoque y medición cuantitativa del proceso de software
- Prevención de defectos
- Evaluación de la calidad

El modelo TSP (Team Software Process) El proceso de Software de equipo (Humphrey, 2000) [21] ayuda a conformar equipos para el desarrollo de software de calidad proporcionando un marco de trabajo, que se construye sobre la base de PSP,

con bases muy definidas en las que los productos de software se generan en varios ciclos.

El origen de TSP se debe a las limitaciones que PSP tenía en el ámbito industrial (McAndrews, 2001) y Humphrey desarrollo el TSP como una respuesta operacional al implementar PSP que era el que englobaría solo las fases del desarrollo de software desde el diseño hasta las pruebas, mientras que TSP proporciona un proceso operacional definido para guiar a los ingenieros y gestores sobre los pasos necesarios en la construcción de equipos.

El modelo CBA-IPI(CMM – BASED APPRAISAL FOR INTERNAL PROCESS IMPROVEMENT) (Dunaway y Masters, 2001) [18] define que es un método que facilita a una organización conocer la capacidad de sus procesos de desarrollo de software mediante la identificación de las fortalezas y debilidades en base al modelo CMM, para dar prioridad a los planes de mejora de software y facilitar a que la organización se centre en los objetivos del negocio y las fases que este método de evaluación de proceso software son: planificar y realizar la preparación para la evaluación, realizar la evaluación e informar los resultados de la evaluación.

La norma ISO/IEC 15504 [26] creada por la Organización Internacional de Normalización (ISO) creada en el año de 1998 proporciona un marco de trabajo para la evaluación de procesos software y establece los requisitos para evaluar la repetibilidad y consistencia de la valoración obtenida.

El estándar está compuesto por cinco partes que son las siguientes:

- Conceptos y vocabularios
- Realización de la evaluación
- Guía para la realización de la evaluación
- Guía sobre el uso para la mejora del proceso y la determinación de la capacidad del proceso
- Modelo de evaluación de procesos.

La norma ISO/IEC 12207 creada por la Organización Internacional de Normalización (ISO) [24] creada en el 1 de agosto de 1995 fue el primer estándar



internacional que proporciona tareas enfocadas al proceso del ciclo de vida del software que se agrupan en procesos principales, proceso de soporte, y procesos generales de la organización, así como un proceso que permite adaptar el ciclo de vida a cada caso concreto.

El propósito de la ISO/IEC 12207: 2008 es proveer un conjunto de procesos definidos para facilitar la comunicación entre adquirientes, proveedores y demás involucrados en el ciclo de vida del software.

Esta norma internacional está dirigida a adquirientes de productos y servicios software a proveedores, desarrolladores, operadores, personal de mantenimiento, administradores, gerentes de aseguramiento de calidad y usuarios del producto software.

La norma ISO/IEC 90003 creada por la Organización Internacional de Normalización (ISO) creada en el año 2004 proporciona la guía necesaria en las organizaciones, este estándar surge debido a que la gestión de calidad que propone ISO 9001 siendo un buen marco de partida es muy general y no permite abordar proyectos de diseño e implantación de sistemas de gestión de la calidad más especializados.

Según esta norma ISO 90003 la organización debe establecer, documentar implementar y mantener un sistema de gestión de calidad software para mejorar continuamente su eficacia de acuerdo a los siguientes requisitos generales.

- Identificar los procesos necesarios
- Determinar la secuencia e interacción de estos procesos
- Determinar los criterios y métodos necesarios
- Asegurarse de la disponibilidad de recursos e información
- Realizar el seguimiento, la medición y el análisis de estos procesos
- Implementar las acciones necesarias

## **2.3 Antecedentes Contextuales y Referenciales**

### **2.3.1 Caracterización Gnoseológica del proceso de desarrollo de software**

Según la definición, alega que el software es un vocablo que procede del idioma inglés, pero que gracias a su uso, ha sido adoptada por la Real Academia Española. Conforme a la Real Academia Española alega que “el software es un cúmulo de programas, instrucciones y reglas informáticas que permiten emprender distintas labores en una computadora (Española, 2001).

Existen varias definiciones de software según los autores tal como se cita a continuación y es por ello que según (Freedman, 1984) “el programa es sencillamente el conjunto de instrucciones que contiene la computadora, ya sean instrucciones para poner en funcionamiento el propio sistema informático (software de sistema) o instrucciones concretas dirigidas a programas particulares del usuario (software específico)” [13].

En lo que respecta a los tipos de software, uno de los más significativos es el *software* o *sistema de base*, que permite al usuario tener el control sobre el hardware (componentes físicos) y dar soporte a otros programas informáticos. La industria del desarrollo de software se ha convertido en un protagonista importante dentro de la economía global, debido a que mueve millones de dólares al año. (Definición de, 2008) [2].

#### **2.3.1.1 Definición de proceso de Software**

Sin embargo, es importante establecer la diferencia entre proceso software y ciclo de vida del software, debido a que en muchas ocasiones son tratados como conceptos similares.

Y por esta razón que según la ISO 12207: define que un ciclo de vida software es “Un marco de referencia que contiene los procesos, las actividades y las tareas

involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del software desde la definición de los requisitos hasta la finalización del software”. [25]

De esta forma se puede interpretar que el ciclo de vida software define los principios y las direcciones que tendrá un proyecto de software y de acuerdo a las cuales se deben llevar a cabo cada una de las etapas que contiene el ciclo de vida, considerando que antes de iniciar una nueva fase se deben haber finalizado los artefactos que contiene en la fase anterior.

El proceso software es un tema mucho más amplio, que toma como base el ciclo de vida del software y por lo tanto abarca los elementos necesarios que están relacionados con las actividades involucradas en la vida de un producto software.

Bajo este contexto cabe indicar que un ciclo de vida de software tiene un marco de referencia para los procesos de desarrollo software mientras que un proceso de desarrollo de software se define como un conjunto de actividades correlacionadas que transforman entradas en salidas.

Un proceso se define quien está haciendo que, cuando, y como alcanzar un objetivos planteados dentro de un proyecto.

Para sustento de lo mencionado anteriormente se cita la siguiente definición en lo que respecta al proceso software:

- “Conjunto de actividades, métodos, prácticas y transformaciones que la gente usa para desarrollar y mantener software y los productos de trabajo asociados (planes de proyecto, diseño de documentos, códigos, pruebas y manuales de usuario)” [27]

La necesidad de la participación humana de forma creativa y la ausencia de acciones que se repiten constantemente hacen que ni el desarrollo ni el mantenimiento de software sea un proceso considerado como fabricación, pero existen algunos procesos similares entre ambos tipos de proceso que son útiles para comprender los procesos software con una perspectiva más amplia.

Por lo tanto, el proceso software es un cambio de estudio amplio y complejo en la ingeniería del software y en la diversidad de elementos relacionados.

A pesar de su importancia de los avances en la investigación en estos temas, muy pocas propuestas han sido aplicadas de forma práctica en el desarrollo de software. Sin embargo y a pesar de que el tema de los procesos software no se han establecido aun como un método que se enseñe y practique universalmente por el desarrollo del software es de esperar que en el futuro las tecnologías de soporte a los procesos software maduren y sean adoptadas por las organizaciones.

### **2.3.1.2 Gestión de los Procesos Software.**

Los requisitos de calidad más significativos de los procesos de software son:

- Que produzcan los resultados esperados
- Que estén basados en una correcta definición y,
- Que sean mejorados en función de los objetivos de negocio, muy cambiantes ante la gran competitividad de las empresas de hoy.

Estos son los objetivos de la Gestión del Proceso software. Para aplicar esta gestión de forma efectiva es necesario asumir cuatro responsabilidades claves:

- Definir
- Medir
- Controlar

De acuerdo a estas responsabilidades, para llevar a cabo de una forma eficiente la mejora del proceso es necesario tener en cuenta los siguientes aspectos que se citan a continuación:

- **Definición del Proceso.-** La definición del proceso es la primera responsabilidad clave que hay que asumir para poder realizar una gestión efectiva de los mismos. Para ello, es necesario modelar los procesos, es decir, representar los elementos de interés que intervienen. El modelado de los procesos sistema, por lo tanto, constituye un paso fundamental para la comprensión y mejora continua de los procesos de una organización [7].
- **Ejecución y Control del Proceso.-** Los proyectos de software de una empresa se llevan a cabo de acuerdo a los modelos de procesos definidos. En este sentido es importante poder controlar en todo momento la ejecución de estos proyectos para garantizar que se obtienen los resultados esperados. Para ello en las dos últimas décadas los denominados “Entornos de

Ingeniería del Sistema orientados a Procesos”, [7] que son los sistemas sistema que ayudan en el modelado de los procesos software utilizando un determinado lenguaje y en su posterior automatización por medio de sus rectificación [7].

- **Medición y Mejora.**- Existe una importante correlación entra la medición y la mejora de los procesos software. Antes de poder mejorar un proceso es necesario llevar a cabo una evaluación, cuyo objetivo es destacar los aspectos que se pueden mejorar. Para ello, es conveniente disponer de un marco de trabajo efectivo que facilite la identificación de las entidades relevantes candidatas a ser medidas. Con los resultados de la medición de los procesos es posible disponer de una información objetiva que permita planificar, identificar, y llevar a cabo de una manera eficiente las acciones de mejora necesarias. (Piattini Velthuis, García Rubio, & Muñoz-Reja, 2010)

### **2.3.1.3 Caracterización de las actividades del desarrollo de software**

Basilo Víctor en su obra *The Future Engineering of Software: A Management Perspective*, manifiesta que definir las características del producto software y del proceso que lo elabora, para que su desarrollo sea conducido y concluido con éxito. Pese a esta diferencia de opiniones referidas a las características del sistema, distintos autores, coinciden que la Maleabilidad - capacidad de modificación indefinida, constituye su característica distintiva.

Esta característica, la cual se enlaza con la aparente facilidad para la adaptación al cambio, produce abundantes inconvenientes. Se debe, a que durante el proceso de cambio, no se recuerda que al software hay que pensarlo y gestionarlo “como un software”. Tomando en cuenta este aspecto, se puede asegurar que el cambio es posible e inevitable. Pero también se sugiere que al distinguir al sistema como software, la gestión del cambio presenta retos complejos.

Mapear los cambios producidos en los distintos elementos que existen en el ambiente de desarrollo, simboliza uno de los principales problemas en el campo de la

Ingeniería del software la gestión de los cambios producidos debe contemplar las siguientes actividades.

Se pueden establecer distintas características de los procesos de desarrollo de software, cuya relación depende de cómo el autor defina a dicho proceso. Un punto sustancial para analizar es la visibilidad del proceso de desarrollo; no por ser la característica principal sino por estar relacionada con el cambio, la visibilidad de un proceso de desarrollo de software se manifiesta en la documentación como resultado de haber puesto en ejecución a dicho proceso.

Prosiguiendo con el análisis de las características de este proceso, se fijará que un buen proceso de software debe ser predecible, es decir, se debe conocer la valoración de costos y el cronograma, el producto resultante debe ser enérgico y ofrecer la funcionalidad requerida, esta postura se vincula con el punto analizado anteriormente “la visibilidad “. (Vanzetti, 2006).

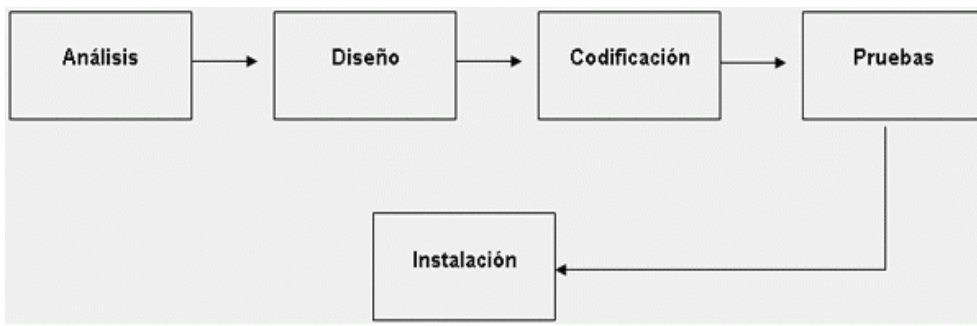
Según Méndez, Gonzalo (2008); indica que dentro de la caracterización de actividades de desarrollo de software se considera las siguientes:

- **Entendible**
- **Visibilidad:** Grado en que las actividades del proceso proporcionan resultados
- **Soportable:** por herramientas CASE
- **Aceptabilidad:** Grado en que los desarrolladores aceptan y usan el proceso
- **Fiabilidad:** Capacidad de evitar o detectar errores antes de que sean defectos
- **Robustez:** Continuidad del proceso a pesar de los problemas
- **Mantenible:** Capacidad de evolución para adaptarse
- **Rapidez:** Velocidad en que el proceso puede proporcionar un sistema a partir de una especificación

### 2.3.1.4 Modelos de desarrollo de software

Cuando se va a desarrollar un sistema intervienen muchas personas, como lo es el cliente quien tiene el problema en su empresa y aspira a que sea solucionado, para esto existe el analista de sistema quien es el encargado de hacer llegar todos los requerimientos y necesidades que tiene el cliente a los programadores quienes son las personas gestoras de la codificación y diseño del sistema para después probarlo e implementar.

Es así como intervienen varias personas debido a que una sola no podría disponer de todo lo necesario, lo más seguro es que le haga falta algún requerimiento o parte del nuevo sistema y entre más estén involucradas mejor es para cubrir con todos los requerimientos del sistema.



**Figura 2.1. Proceso del Desarrollo de software**

El proceso de desarrollo de software se presenta gráficamente en la parte superior, a continuación se desarrollara una breve explicación del mismo.

El primer paso del proceso es el análisis, es ahí, donde el analista se pone en contacto con la empresa para observar cómo está conformada, a que se dedica, conocer todas las actividades que realiza de manera general para posteriormente ver cuáles son las necesidades que la empresa tiene en aquel momento y poder realizar un análisis de la misma.

Es importante conocer cuáles son las exigencias que la empresa tiene por qué en la mayoría de los casos los sistemas se desarrollan pero no pensando en el cliente y es ahí donde el software no satisface las necesidades que existen en la empresa, según

los requerimientos se empieza a realizar el diagrama relacional todo debe de llevar una secuencia lógica de las actividades, esto se realiza de manera manual para ver cómo será su diseño lógico y de pantallas es en este paso donde se plasma todo y queda perfectamente definido como va a formar la funcionalidad del sistema.

El segundo paso es el de diseño aquí interviene todo el diseño del software, es decir, las pantallas, base de datos, todo esto debe de cumplir con ciertos estándares los mismos que se toman en cuenta para poder desarrollar el diseño con calidad y poder ofrecer un diseño amigable en cuestión de colores, tamaños de botones, entre otros.

El tercer paso es la codificación es aquí donde se desarrolla todo el código del software por parte del programador, esto se hace dependiendo de cada programador y él tiene sus bases o formas para realizarlo pero en si deben todos llegar al mismo objetivo que es ofrecerle funcionalidad al software siempre y cuando ligado a las especificaciones del cliente.

El cuarto paso son las pruebas, es donde al software se coloca a prueba como su palabra lo indica para así poder saber cuáles son los posibles errores que está generando el software y con ello mejorarlo para eliminar todos los errores que se suelen presentar, porque un programa con mínimo de errores mayor calidad puede llegar a tener.

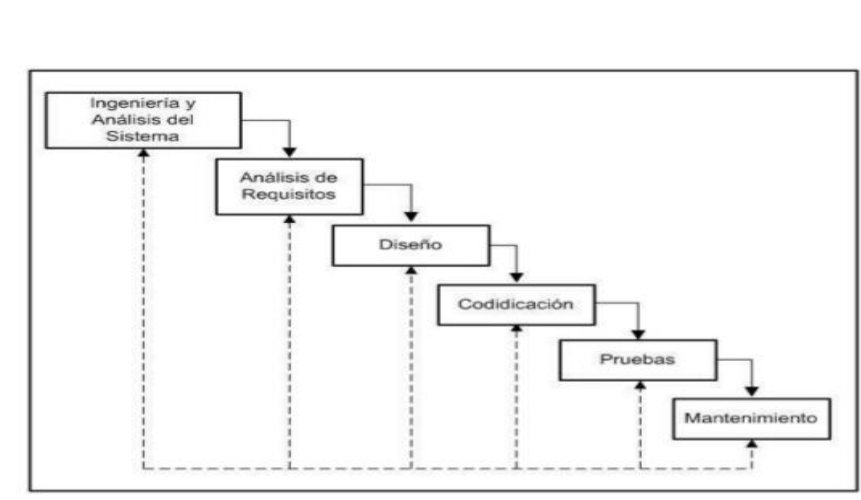
El quinto y último paso es la instalación una vez realizado las pruebas correspondientes al software y haberlo corregido íntegramente se procede a la instalación del mismo ya en la empresa para su uso correspondiente, todo con la finalidad de que los procesos se realicen de una manera más eficiente suprimiendo costos, tiempo y esfuerzo dentro de la organización [5].

#### **a. Modelo Secuencial Lineal**

Es también conocido como "Modelo Clásico", "Modelo Tradicional" o "Modelo en Cascada". Es el más básico de todos los modelos, y ayuda como bloque de



construcción para los demás modelos de ciclo de vida. La visión del modelo cascada del desarrollo de sistema es muy simple; dice que el desarrollo de sistema puede ser a través de una secuencia simple de fases. Cada fase tiene un conjunto de metas bien definidas, y las actividades dentro de ella contribuyen a la satisfacción de metas de esa fase o quizás a una subsecuencia de metas de la fase. Las flechas muestran el flujo de Información entre las fases. La flecha de avance muestra el flujo normal. Las flechas hacia atrás representan la retroalimentación.



**Figura 2.2. Modelo Secuencial Lineal**

### **b. Modelo de Construcción de Prototipos**

El patrón de requerimientos es la creación de una implementación parcial de un software, para el propósito explícito de aprender sobre los requerimientos del software. Un prototipo es construido de una manera rápida tal como sea posible. Esto es dado a los usuarios, posibilitando que ellos experimenten con el prototipo. Estos Individuos luego proveen la retroalimentación sobre lo que a ellos les gustó o no, acerca del prototipo proporcionado, quienes capturan en la documentación actual de la especificación de requerimientos la información entregada por los usuarios para el desarrollo del sistema real.

### **c. Modelo en Espiral**

Este es un modelo de proceso de software evolutivo, el cual enlaza la

Naturaleza iterativa de la construcción de prototipos, pero conservado aquellas propiedades del modelo en cascada.



**Figura 2.3. Modelo en Espiral**

El modelo en espiral fue desarrollado por Boehm, quien lo describe así;

El modelo de desarrollo en espiral es un generador de modelo de proceso guiado por el riesgo que se emplea para conducir sistemas intensivos de ingeniería de sistema concurrente y a la vez con muchos usuarios.

#### **d. Modelo en Cascada**

Existen ocasiones en que los requisitos de un problema se entienden de una manera razonable: cuando el trabajo fluye desde la comunicación a través del despliegue de una manera casi lineal. Esta situación se encuentra a veces cuando es necesario hacer adaptaciones o mejoras bien definidas a un sistema existente. Esto puede ocurrir también en un número limitado de proyectos de nuevos desarrollos, pero sólo cuando los requerimientos están bien definidos y son estables en forma razonable.

El modelo en cascada, algunas veces llamado el ciclo de vida clásico, sugiere un enfoque sistemático, secuencial hacia el desarrollo del sistema, que se inicia con la especificación de requerimientos del cliente y que continúa con la planeación, el modelado, la construcción y el despliegue para culminar en el soporte del sistema terminado. El modelo en cascada es el paradigma más antiguo para la ingeniería del sistema. (Pt, 2011)

Considerando que todos estos modelos de desarrollo de software dan un muy buen aporte a la presente investigación ya que cada uno de estos modelos tiene un ciclo de vida dentro del desarrollo de software.

### **2.3.1.5 Gestión de proyectos de software**

Según lo que indica Bedini Alejandro, Un proyecto es una asociación de esfuerzos, limitado en el tiempo, con un objetivo definido, que requiere del acuerdo de un conjunto de especialidades y recursos. También puede definirse como una organización temporal con el fin de lograr un propósito específico. Cuando los objetivos de un proyecto son logrados se entiende que el proyecto está completo.

La gran diversidad de elementos que intervienen en un proyecto, hacen que éste sea único; pese a ello, es posible aplicar técnicas y métodos comunes para asistir su gestión. Los proyectos informáticos acatan esta definición, pero además se caracterizan por el impacto directo e indirecto que provocan en toda la organización, la casi inevitable existencia de relaciones con otros proyectos informáticos, el estar altamente tendente a sufrir de obsolescencia, tecnológica y la intensa participación de recurso humano de distintas áreas durante su desarrollo.

La gestión de proyectos es un proceso continuo, este proceso requiere de una estrategia global, apoyada por herramientas de trabajo que incrementen la productividad. El propósito de planificar y controlar es proveer una propuesta uniforme para el desarrollo y la administración de los proyectos. Los planes deben apoyar los niveles estratégicos, tácticos y operacionales de las organizaciones con el fin de llegar a las metas corporativas de largo, mediano y corto plazo. Mediante el ciclo de vida de un proyecto, se conforman dos categorías de actividades a realizar y que se encuentran estrechamente relacionadas: las actividades de gestión y de desarrollo del sistema.

Las actividades de gestión son aquellas relacionadas con la administración de las organizaciones, personas, sistemas y procedimientos comprometidos en el proceso de

planificación y construcción del sistema. La planificación del proyecto, junto con las actividades de control, es iterada para cada fase del proyecto y proveen de la estrategia de administración con la cual las actividades de desarrollo del sistema son estimadas, programadas y ejecutadas.

Las actividades de desarrollo del sistema se reúnen en el desarrollo mismo. Las metodologías de desarrollo están peculiarmente organizadas en diversas fases, agrupadas en áreas funcionales de estudio, diseño y construcción, fundamentadas en una estructura de partición del trabajo. La administración y planificación de proyectos requiere de la integración de dos modelos comprendido de trabajo, usualmente no reconocidos: el modelo de administración y el modelo de desarrollo. (Bedini González).

### **2.3.2 Caracterización gnoseológica de los estándares de calidad para el proceso de desarrollo de software**

En la actualidad la calidad no puede centrarse únicamente en el producto software, porque este se encuentra relacionado de manera muy directa con la forma en cómo se desarrolla es decir en el proceso que siguen las organizaciones desarrolladoras de software.

El resultado ha sido un sinnúmero de aristas de un mismo objeto, partiendo cada una de ellas de un marco de referencia analítico diferente y cada una con su propia terminología, es por todo esto que [19] ha planteado que pueden identificarse cinco aproximaciones principales de calidad:

- La trascendente de la filosofía
- La basada en el producto
- La basada en el usuario
- La basada en la fabricación
- La basada en el valor.

Se puede afirmar que casi todas las definiciones existentes de calidad se sitúan en algunas de las aproximaciones antes enumeradas [14].

Según Piattini, García & Caballero, “La calidad se ha convertido hoy en día en uno de los principales objetivos estratégicos para las organizaciones debido a que, cada vez más, su supervivencia depende de la calidad de los productos y servicios que ponen a disposición de los usuarios y clientes y de la satisfacción de éstos” [7].

Según el Diccionario de la Real Academia Española de la Lengua [2], la calidad es (en sus cuatro primeras acepciones):

- Propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor.
- Buena calidad, superioridad o excelencia.
- Carácter, genio, índole.
- Condición o requisito que se pone en un contrato.

Aunque coloquialmente podría parecer más adecuada la segunda definición a la hora de evaluar la calidad de un producto o un servicio, las organizaciones están interesadas en la primera y tercera acepción de calidad. En efecto, se intenta determinar las propiedades inherentes a una cosa que nos permita conseguir que sea mejor que las otras, pero será relativo, ya que dependerá del punto de vista utilizado. Por otra parte, las organizaciones deberán asegurar los requisitos que se fijan en los contratos.

Históricamente, los diferentes gurús de esta área han dado diversas definiciones de calidad

- **W.A Shewhart:** “Existen dos aspectos de la calidad. El primero tiene que ver con la consideración de la calidad de una cosa con una realidad objetiva independiente de la existencia del hombre. La otra tiene que ver con lo que pensamos, sentimos o creemos como resultado de la realidad objetiva. En otras palabras hay un lado subjetivo de la calidad”
- **Philip B. Crosby:** “La primera suposición errónea es que calidad significa bondad, lujo, brillo o peso. La palabra calidad se utiliza para significar el valor relativo de las cosas en frases como Buena calidad, Mala calidad y la expresión calidad de vida. Calidad de vida es un cliché porque cada oyente asume que la persona habla entiende exactamente lo que para él significa la

frase. Esta es precisamente la razón por la que debemos definir calidad como conformidad con los requisitos si queremos gestionarla”

- **Genichi Taguchi:** “La calidad es la pérdida que un producto causa a la sociedad después de ser entregada... además de las pérdidas causadas por su función intrínseca”
- **Armand V. Feigenbaum:** “La calidad de producto o servicio puede ser definido como las características totales compuestas de producto y servicio de marketing, ingeniería, fabricación y mantenimiento por medio de las cuales el producto y servicio en uso **cumplirá las expectativas del cliente**”.
- **Kaoru Ishikawa:** “Debemos enfatizar la orientación al cliente... Cómo uno interpreta al término calidad es importante. Interpretado restringidamente, calidad significa calidad de producto, calidad de servicio, calidad de información, calidad de proceso, calidad de división calidad del personal – incluyendo trabajadores, ingenieros, directivos y ejecutivos-, calidad del sistema, calidad de la empresa, calidad de objetivos, etc.”.
- **W, Edwards Deming:** “La dificultad de definir calidad es traducir las necesidades futuras del usuario en características medibles, de manera que un producto pueda ser diseñado y producido para dar **satisfacción al usuario** al precio que paga ¿Qué es calidad? La calidad sólo se puede definir en términos del agente”.
- **Joseph M. Juran:** “La palabra calidad tiene múltiples significados. Los dos significados que denominan el uso de la palabra son:
  - a) La calidad consiste en las características del producto que satisfacen las necesidades de los clientes y les proporcionan por tanto satisfacción con el producto.
  - b) Calidad consiste en ausencia de deficiencias...Es conveniente estandarizar en una corta definición la palabra calidad como adecuación al uso”

Por otra parte, en las principales normas internacionales, la calidad se define como “el grado en el que un conjunto de características inherentes cumple con los requisitos” [4]. Diferente definición de calidad es la proporcionada por **ISO 8402:**

“Conjunto de propiedades o características de un producto o servicio que le confieren aptitud para satisfacer unas necesidades expresadas o implícitas” [8].

Así se puede ver que la calidad no se trata de un concepto absoluto: el consumidor juzga con todo relativismo en un producto. En general, es posible considerarla como un concepto multidimensional, sujeta restricciones. Inclusive, se puede considerar que no es ni totalmente subjetiva ni totalmente objetiva, así, la calidad no es absoluta es multidimensional.

Y es mediante estos conceptos que se ha investigado sobre la calidad en el siguiente apartado se explicará de manera detallada los diferentes estándares y modelos que han sido creados por diferentes organizaciones para evaluar la calidad durante el proceso de desarrollo de software.

No sin antes hacer énfasis que a nivel mundial existen varias normas y modelos de calidad para evaluar el producto y el proceso software, recalando que para la presente investigación se ha tomado como centro de estudio únicamente los estándares y modelos de calidad que existen para evaluar el proceso de desarrollo de software.

### **2.3.2.1 Modelo de calidad de Software CMMI**

#### **a. Descripción del modelo**

Integración de Modelos de Madurez de Capacidades o Capability Maturity Model Integration (CMMI) [15] es un modelo que ha sido creado para la mejora y evaluación de los procesos de desarrollo de productos de software. Fue desarrollado por el Instituto de Ingeniería del Software de la Universidad Carnegie Mellon (SEI), y publicado en su primera versión en enero de 2002.

CMMI para Desarrollo (CMMI-DEV) provee una circunstancia para evitar estos huecos que suelen quedar cuando no se aplica buenas prácticas de desarrollo. CMMI para Desarrollo consta de buenas prácticas que tratan el proceso aplicado al desarrollo de productos software. Empieza las prácticas que cubren el ciclo de vida del producto desde la fase inicial hasta la liberación del producto.

## **b. Estructura del modelo**

Las prácticas del modelo CMMI-DEV se centran en las actividades de la organización desarrolladora.

Según la investigación realizada se puede determinar que el modelo CMMI establece la medida de la organización a través de las áreas claves de proceso y dependiendo de la aplicación de buenas prácticas en cada área el modelo ubica a la organización en el nivel en el que se encuentre, mencionando que los niveles que ubican a cada organización a la aplicación de buenas prácticas son las siguientes:

El modelo CMMI tiene 5 NIVELES:

**Inicial o Nivel 1 CMM - CMMI.** Este es el nivel en donde están todas las organizaciones que no tienen procesos definidos. Los presupuestos se disparan, la planificación falla no se puede entregar el proyecto en fechas pactadas, se debe emplear mayor esfuerzo humano para terminar un proyecto.

No hay control sobre el avance del proyecto, el desarrollo es completamente desorganizado, no se logra determinar con exactitud qué está pasando en el proceso de desarrollo del software.

**Repetible o Nivel 2 CMM - CMMI.** En este nivel se puede encontrar la diferencia con el Nivel 1 que los proyectos son gestionados y controlados durante el desarrollo del mismo. El desarrollo está mejor organizado y se puede determinar el estado del proyecto de software en el momento que lo requiera.

Los procesos que hay que implantar para alcanzar este nivel son:

- Gestión de requisitos
- Planificación de proyectos
- Seguimiento y control de proyectos
- Gestión de proveedores
- Aseguramiento de la calidad
- Gestión de la configuración



**Definido o Nivel 3 CMM - CMMI.** En este nivel en cambio la forma de desarrollar proyectos (gestión e ingeniería) está determinada, y documentada existen métricas para la obtención de objetivos precisos.

Los procesos que hay que implantar para alcanzar este nivel son:

- Desarrollo de requisitos
- Solución Técnica
- Integración del producto
- Verificación
- Validación
- Desarrollo y mejora de los procesos de la organización
- Definición de los procesos de la organización
- Planificación de la formación
- Gestión de riesgos
- Análisis y resolución de toma de decisiones

**Cuantitativamente Gestionado o Nivel 4 CMM - CMMI.** Los proyectos de software usan objetivos medibles para alcanzar las necesidades de los clientes y la organización desarrolladora, usan métricas para gestionar la organización.

Los procesos que hay que implantar para alcanzar este nivel son:

- Gestión cuantitativa de proyectos
- Mejora de los procesos de la organización

**Optimizado o Nivel 5 CMM - CMMI.** Los procesos de los proyectos de software y de la organización desarrolladora están encaminados a la mejora de las actividades y los procesos. Mejoras incrementales e transformadoras de los procesos que mediante métricas son identificadas, y orientadas a la práctica.

Los procesos que hay que implantar para alcanzar este nivel son:

- Innovación organizacional

- Análisis y resolución de las causas

Según la investigación realizada se ha determinado que la implantación de un modelo de estas características es un proceso largo y costoso que puede costar varios años de esfuerzo a la organización desarrolladora.

#### **d. Ventajas del modelo**

Se puede resumir que:

- Incluir buenas prácticas de desarrollo por áreas del proceso permiten la efectividad en cada una de ellas.
- Es una guía para la organización a través de los niveles de madurez.
- Existe una transición de aprendizaje en la organización desarrolladora.

#### **e. Desventajas del modelo**

Las desventajas encontradas en el modelo son:

- El modelo es muy detallado y prescriptivo.
- Requiere bastante inversión
- Complejidad al entender la forma de aplicar
- Su atención va en alcanzar el siguiente nivel antes que en los objetivos que se quiere alcanzar dentro de la organización
- Es muy rígido y el enfoque va a nivel organizativo antes que en el mismo proceso de desarrollo

### **2.3.2.2 Modelo SCE (Software Capability Evaluation)**

#### **a. Descripción del modelo**

Este es un método para evaluar los procesos de desarrollo de software dentro de una organización para determinar la capacidad del proceso de desarrollo y se evalúa al efectuar un proceso determinado.

#### **b. Estructura del modelo**

El proceso de evaluación definido en SCE está conformado por las siguientes actividades:

- Planificar y preparar la evaluación
- Llevar a cabo la evaluación
- Informar sobre los resultados de la evaluación



Figura 2.4. Procesos evaluados por SCE (Software Capability Evaluation)

Fuente: [18]

**c. Ventajas**

- Define el alcance de evaluación
- Selecciona los proyectos que va a evaluar
- Analiza los datos

**d. Desventajas**

- Se rige solo a la parte organizacional
- En la evaluación de los procesos no contempla los de ingeniería de software.

**2.2.3.3 Modelo CBA-IPI (CMM – Based Appraisal for Internal Process Improvement)**

**a. Descripción del modelo**

Este método de evaluación fue creado por (Dunaway y Masters, 2001) permite a las organizaciones conocer la capacidad de los procesos software identificando

fortalezas y debilidades tomando como base el modelo CMM dando prioridad a los planes de mejora de software.

#### **b. Estructura del modelo**

El método consiste en la evaluación de la capacidad del proceso software de una organización mediante un grupo de profesionales entrenados debidamente para valorar las distintas áreas clave del proceso de CMM que se encuentra en la evaluación.

#### **c. Ventajas del modelo**

Es un método para asignar valor a la capacidad para la mejora de procesos.

#### **d. Desventajas del modelo**

Para poder funcionar necesita que se haya implementado anteriormente el modelo CMMI

### **2.3.2.4 Modelo IDEAL**

#### **a. Descripción del modelo**

Este modelo es el que define un marco de ciclo de vida para la mejora de procesos de desarrollo de software crea un enfoque usable y entendible estableciendo los pasos que se deben tomar como guía para desarrollar un programa de mejora proporcionando un enfoque ingenieril.

#### **b. Estructura del modelo**

El modelo IDEAL contiene 5 fases conformada por diferentes actividades:

**Iniciar:** en la fase uno el propósito es formar los fundamentos básicos para garantizar la iniciativa de mejoramiento de los procesos, se aclaran con la gerencia cuales son los objetivos de la empresa que serán beneficiados por el esfuerzo.

Las actividades de esta fase determinan el éxito o el fracaso del programa.

**Diagnosticar:** evalúa través de un método formal las fortalezas y debilidades del proceso seguido por los proyectos.

**Establecer:** en esta fase el modelo realiza la planificación específica que se desea alcanzar se desarrolla un plan de proyecto donde se logra definir las estrategias y las prioridades.

**Actuar:** realiza el mejoramiento de los procesos llevando a cabo el plan de acción para mejorar los procesos, se estrena a los respectivos niveles de personal, se miden los avances logrados, se realiza proyectos pilotos, se implantan los procesos mejorados en los proyectos nuevos o existentes, se hacen mini evaluaciones para constatar la evolución del plan.

**Difundir:** aprender de la experiencia del ciclo recién realizado y aumentar la habilidad de la organización para mejorar los procesos en forma continua.

### **2.3.2.5 Modelo PSP (Personal Software Process)**

#### **a. Descripción del modelo**

El modelo Personal Software Process (PSP) [19] brinda al personal de desarrollo de software de una organización establecer un marco de disciplina personal.

El proceso PSP consiste en un grupo de métodos, actividades que muestran como planificar, medir y administrar el trabajo de cada integrante desarrollador de una organización.

#### **b. Estructura del modelo**

En la siguiente figura se muestra los procesos que constituye PSP agrupados por niveles y para alcanzar un nivel deben cumplir con los requisitos que tiene cada nivel.



**Figura 2.5. Niveles de procesos PSP**

**FUENTE: [19]**

**c. Ventajas del modelo**

- Se puede evaluar el proceso de los desarrolladores de software y determinar las buenas prácticas de desarrollo

**d. Desventajas del modelo**

- Es un modelo que debe tener como base el modelo CMM para implementar sus procesos de evaluación.
- La evaluación está enfocada solamente al personal de desarrollo de software de la organización.
- Si no existe el modelo CMM implementado en la organización difícilmente se podría acceder a la evaluación

### 2.3.2.6 Modelo TSP (Team Software Process)

#### a. Descripción del modelo

El TSP es un modelo que permite guiar a los desarrolladores en el uso de métodos de trabajo que comparten un objetivo común los cuales deben tener roles y estos deben darles un sentido de pertenencia y liderazgo

#### b. Estructura del modelo

La primera condición para implementar TSP es que debían haber sido entrenados anteriormente bajo el modelo PSP ya que en relación con este entrenamiento cada integrante del equipo de desarrollo sabe cómo planear detalladamente un proyecto, utilizar datos del proceso, medir la calidad del producto, utilizar los problemas operacionales.

Y su estructura se basa tal como se muestra en la siguiente figura:



**Figura 2.6. Estructura del modelo TSP**

**FUENTE: [20]**

**a. Ventajas del modelo**

- Fácil implementación cuando el equipo de trabajo tiene la predisposición.

**b. Desventajas del modelo**

- Requiere que se haya implementado el modelo CMM y PSP para el éxito total de la evaluación de procesos con este modelo.
- Es muy costoso al implementar los modelos mencionados anteriormente.

**2.3.2.7 Estándar ISO 90003**

**a. Descripción del modelo**

Según la Organización de Estandarización Internacional (ISO) [35] 9000 se ha especializado en todo lo referente a la solución del software en la ISO 9000-3, esta disciplina tiene características adecuadas como para diferenciarse del proceso de producción ha definido un grupo de estándares que son generalmente aplicables a los procesos de producción de desarrollo de software.

Según la ISO 9000 [35] proporciona un conjunto de estándares para la gestión de la calidad en cualquier actividad relacionada con el proceso de producción.

Las empresas desarrolladoras están a favor de crear sistemas de calidad que permita tomar todas las fases de sus procesos de producción.

Un sistema de calidad define los requisitos para el desarrollo de los procesos de una organización, algunas de las actividades tomadas por el sistema son:

- Auditoria de los proyectos para asegurar que los controles de calidad son respetados.
- Comprobar que ha mejorado la calidad del sistema.

Proporcionar al grupo de desarrollo una serie de guías como pueden ser nuevas notaciones, procedimientos y estándares. También se generaran documentos destinados a la dirección del grupo de desarrollo.



La ISO 9000 se ha especializado en todo lo referente a la solución del software en la ISO 9000-3, puesto que esta disciplina tiene características propias diferentes como para distinguirse del proceso de producción en general

### **2.3.2.8 Estándar ISO 12207:2008**

Según la Organización Internacional de Normalización (ISO – IEC) [23] Esta norma fue creada específicamente para el proceso de desarrollo de software y consta de los siguientes procesos:

- Procesos Principales
- Procesos de Apoyo
- Procesos de Gestión

#### **Procesos principales**

En este proceso cumple con la necesidad de preparar una solicitud y seleccionar un proveedor, determinar procedimientos y recursos para gestionar el proyecto. Contiene actividades de análisis, diseño para el producto sistema. Puede contener actividades a nivel de sistema. Cubre la operación del producto sistema y apoyo a los usuarios. Las actividades y tareas hacen referencia al sistema.

Modificar el producto sistema preservando su integridad. Incluye la migración y retirada del producto.

#### Actividades

- Adquisición.
- Suministro.
- Desarrollo.
- Explotación.
- Mantenimiento

#### **Procesos de apoyo**

Este proceso tiene como el objetivo obtener información, identificar, definir y versionar, mediante líneas bases, los elementos del sistema, así como también asegurar la completitud y correctitud de los elementos que pertenecen a la

configuración, de controlar su manejo, persistencia y entrega de los mismos. Proveer de mecanismos para objetiva e independientemente asegurar que los productos y/o servicios cumplan con los estándares y requerimientos establecidos, y que el desarrollo de otros procesos se apege lo más posible a lo planificado originalmente. Proveer las evaluaciones referentes a la verificación de un producto o servicio de una actividad dada, determinar si un sistema ya construido cumple con las especificaciones y requerimientos para los cuales fue realizado, proveer un marco que favorezca la integración entre inspector e inspeccionado es proveer un marco adecuado para establecer auditorías formales y contractuales sobre un determinado producto o servicio provisto.

El propósito de este proceso es proveer mecanismos para la creación de procesos capaces de resolver problemas y tomar acciones correctivas para remover nuevos problemas detectados.

Actividades:

- Documentación
- Gestión de la configuración.
- Aseguramiento de calidad.
- Verificación. Validación.
- Revisión conjunta.
- Auditoría.
- Resolución de problemas.

### **Procesos de gestión**

Este proceso permite gestionarse para acoplarse con otros procesos.

Actividades:

- Inicio y Definición de Alcance
- Planificación
- Ejecución y Control
- Revisión y Evaluación

- Terminación

**a. Ventajas del modelo**

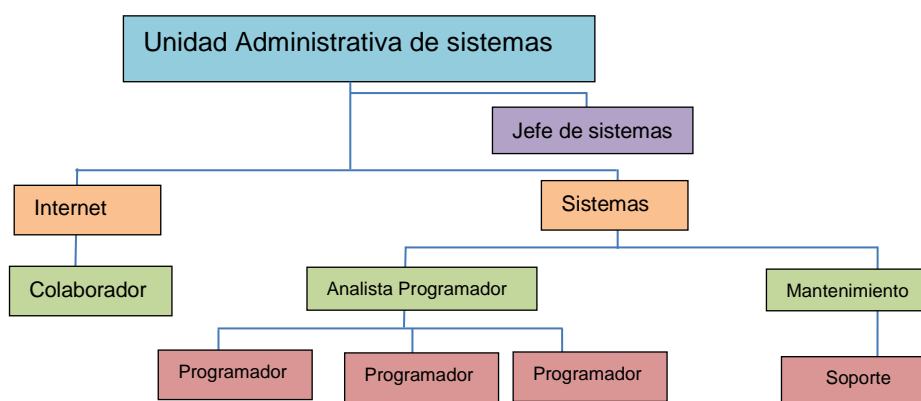
- Proporcionar procesos bien definidos que se encuentran dentro de todo el proceso de ciclo de vida del software.
- Se puede implementar desde un acuerdo informal hasta un contrato jurídicamente establecido.
- El modelo es flexible permite seleccionar los procesos necesarios para adecuarlos a la organización o proyecto ya que no es todas las organizaciones o proyectos es necesario incluir todos los procesos de la norma.

**b. Desventajas del modelo**

- No posee documentación detallada en cuestión de nombres, formatos, contenidos específicos,

## 2.4 Antecedentes Contextuales

El Departamento de Informática conocido en la actualidad como Unidad Administrativa de Sistemas de Información de la Universidad Técnica de Machala, actualmente como Jefe de Sistemas la Ing. Betty Pachucho quedando para el 13 de Julio de 1992 como Jefe encargada de la Unidad Administrativa de Sistemas, en la actualidad cumple con diferentes actividades, además contando con tres personas más que cumplen el rol de programadores, a continuación el organigrama de la unidad de sistemas.



**Figura 2.7. Organigrama funcional de la Unidad de Sistemas de la Universidad Técnica de Machala, 2014**

Este departamento fue creado con la finalidad de implementar un sistema integrado con todas las funciones operativas que realizan cada una de las Unidades Académicas, principalmente para atender las necesidades de los estudiantes que son los usuarios activos para la Universidad siendo en la actualidad más de 11000 estudiantes a los que se debe atender.

Siendo el sistema informático desarrollado a la medida de las necesidades de la Universidad Técnica de Machala está en la capacidad de atender al 100% de usuarios tantos estudiantes, personal docente, personal administrativo.

Para la debida justificación del problema encontrado en esta investigación se realizó una encuesta al personal que labora en el departamento de desarrollo de software de la Universidad Técnica de Machala, dando como resultado lo siguiente:

**1. ¿Existe documento de planificación de proyectos de sistema?**

**Análisis:** el 100% de las personas entrevistadas supieron decir que efectivamente no existe documentación alguna que respalde la planificación de proyecto de un nuevo módulo en el sistema SIUTMACH.

**2. ¿Se han presentado retrasos en la entrega de un nuevo módulo al proyecto?**

**Análisis:** el 95% de las personas entrevistadas manifestaron que siempre existen retrasos en la entrega de un módulo del proyecto ya que las autoridades siempre les aparecen algo más importante.

**3. ¿Existe la especificación de requisitos mediante un documento o algún estándar?**

**Análisis:** el 100% de las personas entrevistadas manifestaron que no existe ningún documento que respalde la especificación de requisitos que lo hacen de manera informal no se basan en ningún estándar.

**4. ¿Un estimado de cuantos incidentes atienden en el día?**

**Análisis:** Se obtuvo como resultado el 100% de las personas entrevistadas manifestaron que atienden un promedio de 3 a 4 incidentes diarios.

**5. ¿Existe algún estándar para el asunto de pruebas del sistema?**

**Análisis:** el 100% de las personas entrevistadas manifestaron que las pruebas las realizan de la misma forma que elaboran casi todo en lo que respecta a documentación de manera informal pero no existe ni documentación ni estándar alguno alegan que no existe el tiempo suficiente.

**6. ¿Ha existido inconformidad por parte de los clientes (estudiantes o personal administrativo con respecto al sistema SIUTMACH)?**

**Análisis:** Se obtuvo que el 71% afirman que si existe inconformidad por parte de los usuarios del sistema mientras que el 14.5% ratifican que no existe inconformidad y el otro 14.5 % mencionan que a veces existen la inconformidad por parte de los usuarios, teniendo como un total de respuestas mayoritarias para detectar que existe un problema.

**7. ¿Considera conveniente que se aplique un modelo de calidad para el proceso de desarrollo de sistema?**

**Análisis:** el 100% de las personas entrevistadas manifestaron que sería muy conveniente aplicar un modelo que garantice el uso de buenas prácticas para el proceso de desarrollo de sistema.

El instrumento de la encuesta que se aplicó a las personas que laboran en el departamento de la unidad de sistemas de la Universidad Técnica de Machala se puede apreciar en el **anexo #1**.

Según esta encuesta aplicada se ha podido determinar que dentro de la Unidad de Sistemas existen dificultades durante el proceso de desarrollo de sistema tales como falta de documentación en los procesos de planificación del proyecto, en la especificación de requisitos, usuarios insatisfechos, el mismo hecho de no formalizar el asunto de pruebas sobre el sistema desarrollado presentan incidentes en el mismo, incumplimiento en el tiempo de entrega de algún módulo propuesto por la Institución.

Al aplicar la encuesta sobre la necesidad de contar con un modelo de calidad que permita el correcto proceso de desarrollo de sistema el 100% del personal manifestaron que si porque permitiría optimizar el tiempo y organizar el proceso de desarrollo de software y de esta manera mejorar la calidad en cuanto al producto generado por el departamento de la Unidad de Sistemas de la Universidad Técnica de Machala.

## **2.5 Conclusiones**

De este capítulo se establece que está centrado en contener la parte teórica que servirán como base para la investigación realizada de los cuales se divide en los antecedentes históricos que es donde se recopila información basada en el tema de estudio que es el proceso de desarrollo de software.

## CAPÍTULO III

### **3. PROPUESTA DEL MODELO DE CALIDAD PARA LA GESTIÓN DE PROCESOS DE DESARROLLO DE SISTEMA DE LA UNIDAD DE SISTEMAS DE LA UNIVERSIDAD TÉCNICA DE MACHALA**

#### **3.1 Introducción**

En este capítulo se encuentra la propuesta del modelo de calidad basado en un análisis comparativo de los modelos y estándares de calidad aplicadas al proceso de desarrollo de software, para lo cual se ha tomado diversos criterios que corresponde a cada uno, que permitirán evaluar cuál de ellos se acopla más a la realidad de la Unidad de Sistemas de la Universidad Técnica de Machala.

Una vez seleccionada la norma a ser aplicada, ésta definirá los procesos necesarios para plantear una posible solución al problema encontrado en la presente investigación.

#### **3.2 Proceso General del desarrollo de Software**

Según Santander (2008) “un proceso de desarrollo de software es la descripción de una secuencia de actividades que se deben acoplar a un equipo de trabajo para generar un conjunto coherente de productos software” [29].

Santander (2008) afirma que “el objetivo de un proceso de desarrollo de software es la formalización de las actividades relacionadas con el desarrollo del software” [29]. La mayoría de los sistemas que se desarrollan, tardan en la entrega, cuesta mucho más de lo estimado.

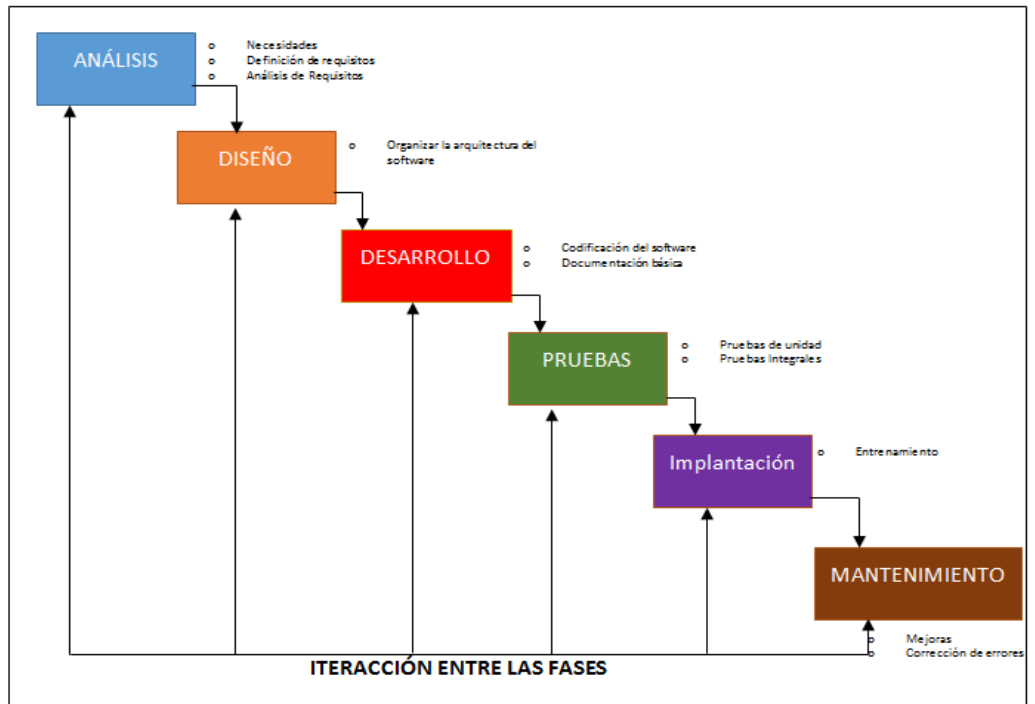
Debido a este concepto la razón básica por la que se requiere disponer de un proceso de desarrollo es mejorar la calidad del producto software, proporcionar a la organización desarrolladora de software plantillas de desarrollo del proyecto en el que quede definido lo que cada integrante del equipo de trabajo que interviene debe realizar y los productos que debe generar a lo largo de él, mejorar la calidad del producto que se genera en función de disminuir el número de defectos que se



producen y que deben ser corregidos, disminuir la rigidez de los defectos que al final pueden persistir en el producto final, mejorar la reusabilidad, de forma que el trabajo realizado pueda ser reutilizado en próximos proyectos.

Mediante este análisis se considera que a más de la organización y actividades planteadas dentro del proceso de desarrollo de software se debe tomar el proceso del ciclo de vida del software y para esta investigación se ha tomado como ejemplo el ciclo de vida tradicional, ya que si revisamos la teoría investigada con respecto al ciclo de vida del software es el modelo tradicional el que permite que se haga un seguimiento correcto en el desarrollo de software desde el inicio hasta la implementación y que permite tener como base el modelo conocido como el modelo cascada que es el que abarca las cuatro etapas del proceso de desarrollo de software, análisis, diseño, implantación y mantenimiento y se ha seleccionado este modelo debido a que en la Unidad de Sistemas de la Universidad Técnica de Machala, la forma del desarrollo es por iteraciones, o sea que dentro del mismo sistema van implementando diferentes módulos de desarrollo al sistema integrado.

Según los resultados que se muestran en el **Apartado II**, la Unidad de desarrollo de software de la Universidad Técnica de Machala no aplican una sola metodología ya que si lo hicieran de esa forma se demorarían demasiado tiempo por lo que manifiestan que el ciclo de vida de software lo hacen del modelo tradicional como lo indica el siguiente Figura.



**Figura 3.1. Ciclo de vida de desarrollo de software**

### 3.3 Selección de los modelos y estándares de calidad de sistema para el proceso de desarrollo de software.

En la investigación se ha seleccionado los modelos y estándares que se acoplan al proceso de desarrollo de software, si bien es cierto a nivel mundial se encuentran diferentes modelos de evaluación de calidad enfocados al producto y al proceso; para el presente estudio se centra en el proceso de desarrollo de software; consecuentemente, serán los que permitan hacer el análisis comparativo de sus características y acoplar al modelo planteado para la Unidad de Sistemas de la Universidad Técnica de Machala.

#### a. CMMI (Capability Maturity Model)

**Propósito.-** proporcionar una única guía unificada para la mejora de múltiples disciplinas tales como la ingeniería de sistemas, ingeniería del sistema.

## Actividades

Este modelo tiene 4 áreas del conocimiento o disciplinas que incluyen.

- Ingeniería de Sistema (SW)
- Ingeniería de Sistemas (SE)
- Desarrollo Integrado de Productos y Procesos (IPPD)
- Acuerdos con Proveedores (SS).

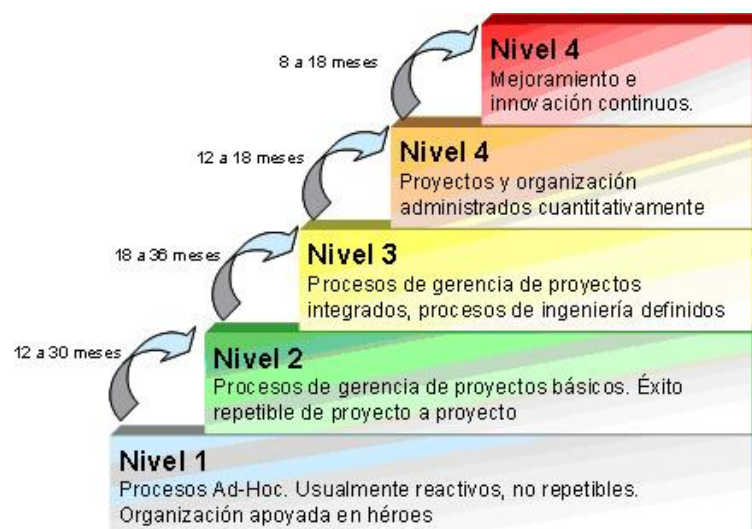
**Ingeniería de Sistemas:** Abarca el desarrollo total del sistema que puede o no incluir el desarrollo de sistema.

**Ingeniería de Sistema:** Cubre el desarrollo de sistema y su mantenimiento.

**Desarrollo integrado de Productos y Procesos:** Contempla un enfoque sistemático para la colaboración de los involucrados relevantes a través de la vida del producto.

**Acuerdo con Proveedores:** En proyectos complejos se requiere de la incorporación de proveedores para ejecutar funciones o añadir modificaciones a productos.

Cabe recalcar que este modelo ha sido elaborado por etapas en el Figura que se encuentra a continuación se detalla los procesos que apoya cada nivel y el tiempo que toma en implementar cada nivel.



**Figura 3.2. Niveles de capacidad y procesos del CMMI**

**FUENTE: [15]**

De la misma forma dependiendo de cómo se va ubicando cada organización en el nivel que amerite la situación de cada aplicación que se realice hay procesos que se deben cumplir dentro de cada nivel para lograr la calificación adecuada.

A continuación se detalla los procesos que debe cumplir cada nivel del modelo CMMI.

## **Nivel 2 (Administrado)**

### **Administración de Procesos**

- Organization Process Focus (OPF)
- Organization Process Definition (OPD)
- Organizational Training (OT)
- Organizational Process Performance (OPP)
- Organizational Innovation and Deployment (OID)

### **Administración de Proyectos**

- Project Planning (PP)
- Project Monitoring and Control (PMC)
- Supplier Agreement Management (SAM)
- Integrated Project Management for IPPD (IPPD)
- Risk Management (RSKM)
- Integrated Teaming (IT)
- Integrated Supplier Management (IPM)
- Quantitative Project Management (QPM)

### **Ingeniería**

- Requirements Management (REQM)
- Requirements Development (RD)
- Technical Solution (TS)
- Product Integration (PI)
- Verification (VER)
- Validation (VAL)

## **Soporte**

- Measurement and Analysis (M&A)
- Process and Product Quality Assurance (PPQA)
- Configuration Management (CM)
- Organizational Environment for Integration (OEI)
- Decision Analysis and Resolution (DAR)
- Causal Analysis and Resolution (CAR)

## **Modelo PSP (Personal Sistema Process)**

El PSP es una versión pequeña de CMM donde se preocupa solo por un conjunto de las áreas del proceso.

La estructura de PSP es como sigue:

### **PSP 0 Medición Personal**

Se relaciona con la estimación del tiempo para desarrollar un producto sistema y la identificación, clasificación y manejo de los Defectos producidos durante el proceso de desarrollo. Los datos recopilados mediante a la aplicación de los scripts sirven de base para la realización de estimaciones más reales en procesos futuros.

#### **PSP 0.1**

Es un nivel que extiende y complementa a PSP0, a través del establecimiento de estándares de codificación orientados a mejorar las estimaciones de tamaño, así como la consideración de propuesta de mejora al nivel por parte de los ingenieros.

### **PSP 1 Planeación Personal**

Le agrega pasos de planeamiento a PSP0. El primer paso agrega estimaciones de tamaño y recursos y un reporte de prueba.

#### **PSP 1.1**

En este proceso se realiza el seguimiento del proyecto:

- Entender la relación entre el tamaño de los programas que escriben y el tiempo que les toma desarrollarlos.

- Aprender a realizar compromisos que puedan cumplir.
- Preparar un plan ordenado para realizar su trabajo
- Establecer una base para realizar un seguimiento de su trabajo.

## **PSP 2 Calidad Personal**

La calidad, está relacionada con la cantidad de defectos que el producto de sistema contiene. En este nivel se introducen las inspecciones en las fases de diseño y codificación, como un mecanismo para aumentar la calidad de los productos.

### **PSP 2.1**

Es el proceso del diseño.

El objetivo no es decirles a los desarrolladores como diseñar sino orientar el criterio para la finalización del diseño, es decir cuando han terminado que es lo que deben haber obtenido.

## **PSP 3 Cíclico Personal**

Subdivide el proceso personal de desarrollo de grandes programas en elementos en la escala de PSP2. Estos programas son diseñados para ser desarrollados en pasos incrementales. Para cada iteración se utiliza un PSP2 completo, incluyendo diseño, codificación, compilación y pruebas.

- De esta manera los desarrolladores pueden concentrarse en la verificación de la calidad del último incremento sin preocuparse por defectos en ciclos anteriores.
- Si un incremento anterior tiene muchos defectos, la prueba será más compleja y los beneficios de escalar PSP se pierden. Esta es una razón para enfatizar revisiones de diseño y código en los pasos anteriores de PSP.

**b. Estándar ISO 12207:2008**

A nivel del proceso de sistema este estándar 12207 es usado para los procesos primarios, de soporte y organizacionales y en cada uno de ellos existe subproceso que se deben hacer para cumplir con el estándar tal como se cita a continuación.

En los **procesos principales del ciclo de vida del sistema** se utiliza para:

- Adquirir
- Suministrar
- Desarrollar
- Mantener
- Operar

En los **procesos de apoyo** se utiliza para:

- Documentación
- Administración de la configuración del sistema,
- Aseguramiento de la calidad
- Verificación
- Validación
- Revisiones conjuntas
- Auditorias
- Resolución de problemas.

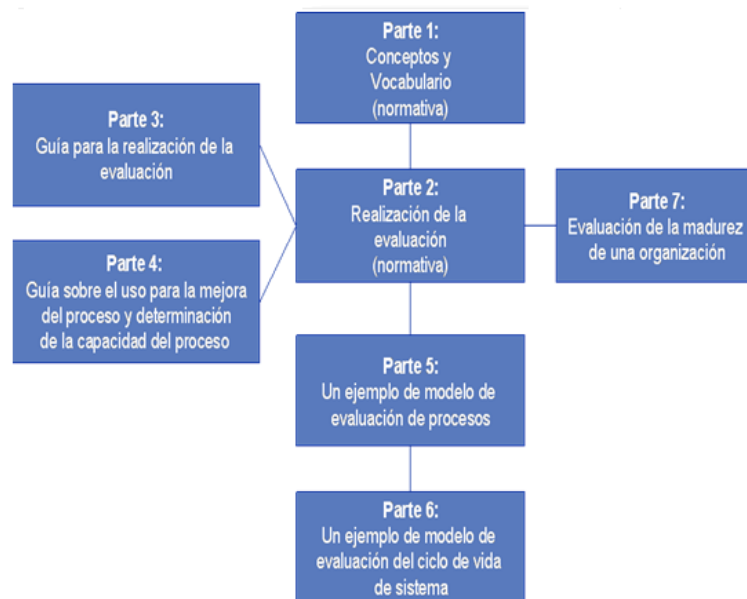
En los **procesos organizacionales se utiliza para:**

- Gestión
- Infraestructura,
- Mejoras
- Recursos Humanos.

### c. Norma ISO 15504

#### Características

- Establece un marco para métodos de evaluación, no es un método o modelo en sí.
- Comprende la evaluación de procesos, mejora de procesos, determinación de capacidad.
- Está alineado con el estándar ISO/IEC 12207 que define los procesos del ciclo de vida del desarrollo, mantenimiento y operación de los sistemas de sistema.
- Equivalencia y compatibilidad con CMMI. ISO forma parte del panel elaborador del modelo *CMMI* y *SEI* mantiene la compatibilidad y equivalencia de ésta última con 15504.



**Figura 3.3. Estructura del estándar ISO 15504**

**FUENTE: [26]**



### 3.4 Análisis Comparativos de los modelos o estándares de calidad de sistema

Para elaborar un análisis comparativo de los modelos y estándares de calidad enfocados al proceso de desarrollo de software se ha tomado varios conceptos relacionados con la calidad de software en la fase del proceso de desarrollo de software.

En lo que respecta al proceso de desarrollo de software según (Cockburn, 2010) dentro del proceso de ingeniería se basa en 4 actividades fundamentales [27]:

- Especificación del software
- Desarrollo del software
- Validación del software
- Evolución del software

Y son las actividades que se considerarán y los que más se aproximen al modelo de calidad que se aplicará dentro de la Unidad de Sistemas de la Universidad Técnica de Machala.

Además de considerar estos parámetros se toma como referencia un estudio realizado por la Universidad Tecnológica Nacional de Buenos Aires en una tesis de Maestría en Ingeniería de calidad elaborado por la Lic. Fernanda Scalone [31] donde consta de lo siguiente:

#### **Criterios a evaluar de los modelos de calidad para el proceso del software.**

**Criterio 1: Disponibilidad:** el grado en que es posible acceder a la información existente [31].

1. La información no se encuentra disponible al público en general
2. Hay disponibilidad de algunos documentos pero es limitado el acceso.
3. Se encuentra información suficiente disponible para ser usada.

**Criterio 2: Claridad:** posee mecanismos explicativos sobre el uso tiene factores tales como: estructura, idioma, y la presentación del modelo.

1. El modelo no posee mecanismos de ayuda sobre el modelo de aplicarlo.
2. El modelo es claro pero no posee mecanismos de aplicarlo.
3. El modelo posee mecanismos acerca de su modo de empleo.

**Criterio 3: Adaptabilidad:** posee la capacidad de adaptarse a diferentes situaciones que se presente el proyecto.

1. El modelo no es adaptable se presenta de forma rígida para su uso.
2. El modelo es adaptable pero exige ciertas reglas a seguir
3. El modelo puede ser adaptado.

**Criterio 4: Completitud:** Posee descripción de atributos, métricas y mecanismos de ayuda para llegar a su medición.

1. El modelo se encuentra incompleto
2. El modelo describe medianamente sus componentes
3. El modelo describe todas sus partes

**Criterio 5: Área de aplicación:** se puede aplicar el modelo a las diferentes áreas de calidad del sistema.

1. Modelo de proceso, metodología o estándar.
2. Puede ser modelo de proceso y producto al mismo tiempo
3. Modelo para producto sistema

**Criterio 6: Tipos de proyectos:** a que tipos de proyectos se puede aplicar.

1. En su mayoría se puede aplicar a proyectos robustos de gran tamaño y tiempo
2. Puede ser aplicado a varios proyectos que varían en tiempo y tamaño
3. Es aplicable a toda clase de proyectos ya que se puede aplicar mediante interacciones.

A más de considerar el puntaje aplicado para cada criterio desde el valor 1 hasta 3 existen modelos que **no aplica** el criterio y se considerara el estado **N/A** que significa que no tiene puntaje alguno.

**Tabla 3.1.****Puntaje de los criterios a evaluar para seleccionar el modelo o estándar**

ESTANDAR / MODELO	CRITERIOS DE VALORACIÓN						Total
	Criterio 1	Criterio 2	Criterio 3	Criterio 4	Criterio 5	Criterio 6	
<b>CMMI</b>	2	2	1	2	1	1	9
<b>PSP</b>	2	2	2	2	1	2	11
<b>ISO 15504</b>	2	2	2	2	3	1	12
<b>ISO 12207</b>	2	2	2	2	2	3	13

En la presente investigación se ha tomado como referencia los resultados de la tabla 3.1 donde se pudo determinar que el estándar ISO 12207 tiene la más alta calificación aunque se haya tomado a todos los modelos o estándar que son parte del proceso de desarrollo de sistema se eligió aplicar la ISO 12207 complementando con la norma ISO 15504 por ser la norma que evalúa la madurez del proceso y de esta forma adaptar lo que puede corresponder a la Unidad de Sistemas de la Universidad Técnica de Machala.

### **3.5 Procesos de desarrollo de software en la Unidad de Sistemas de la Universidad Técnica de Machala.**

Los elementos del proceso de desarrollo de software que se consideraran para la presente investigación son el ciclo de vida de software que es aquel que describe desde la fase inicial hasta la fase final del desarrollo de software y de esta manera garantizar que exista un correcto cumplimiento de requisitos para el cliente y para ello se utilizará el modelo de ciclo de vida tradicional, y la metodología tradicional que es la metodología que permite ir realizando las actividades que amerita cada una de las etapas que tiene la metodología y es por lo que a continuación se detalla el entorno de trabajo de la Unidad de Sistemas de la Universidad Técnica de Machala.

#### **3.5.1 Metodología del proceso de desarrollo de la Unidad de Sistemas de la Universidad Técnica de Machala.**

La metodología que se utilizará para la investigación desarrollada será la metodología tradicional debido a que según el estudio realizado por Carvajal, J.

(2008). Metodologías ágiles: herramientas y modelo de desarrollo para aplicaciones java como metodología empresarial. Tesis Final de Máster, manifiesta que una metodología es una guía de procesos que garantiza el producto y cumple con las características de especificación de requisitos a lo largo del proyecto, basados en los procesos, proyectos documentados, gestión predictiva es decir una planificación de proyectos y estas metodologías tradicionales van de la mano con el ciclo de vida tradicional que se detalla a continuación.

Fases que contiene el ciclo de vida de desarrollo de software:

- a. Fase de especificación
- b. Fase de Análisis y diseño
- c. Fase de Construcción
- d. Fase de Pruebas

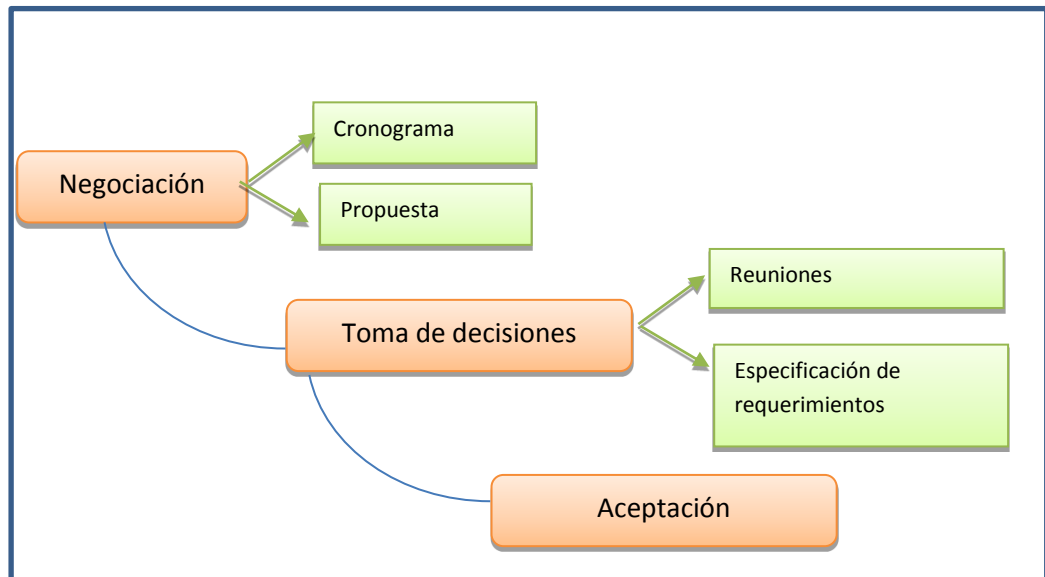
**a. Fase de Especificación**

En esta fase se identifican los requerimientos del sistema es el momento donde se reúnen con el cliente que para este caso sería el personal administrativo de la Institución con el jefe de sistemas.

También permite organizar al equipo de trabajo donde se estima el tiempo de demora o sea el cronograma y los recursos a utilizar.

Cabe recalcar que la especificación de los requisitos está basada en el estándar IEEE 830, del que se obtiene el primer documento a presentar Ver Anexo # 2.

Se presenta gráficamente esta fase:

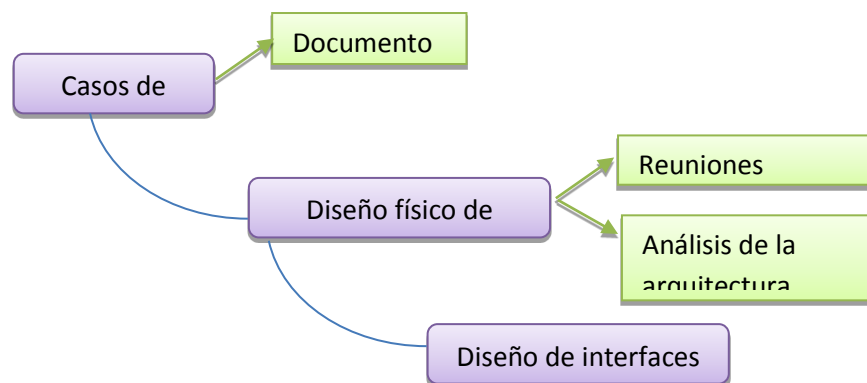


**Figura 3.4. Fase de especificación**

El equipo de desarrollo entrega como artefactos el cronograma y el documento de la especificación de requerimientos de Sistema (ERS).

#### b. Fase de diseño

En esta fase el equipo de trabajo realiza un conjunto de actividades en las que se analiza los requerimientos para producir un conjunto de artefactos que se requieren tales como casos de uso, diseño del modelo físico de datos, especificación de datos, análisis de la arquitectura.



**Figura 3.5. Fase de diseño**

### **c. Fase de Codificación**

El equipo de trabajo para esta fase se centra en la generación del código es decir en la construcción del sistema de acuerdo a las especificaciones planteadas al inicio del proyecto.

### **d. Fase de Pruebas**

Validan el sistema para comprobar si es lo que el cliente necesitaba y está de acuerdo a los requisitos planteados inicialmente en el proyecto.

## **3.6 Características de la propuesta del modelo de calidad para el proceso de desarrollo de software**

### **3.6.1 Objetivo del modelo de calidad**

Optimizar el proceso de desarrollo de software en la unidad de sistemas de la Universidad Técnica de Machala mediante la selección de procesos de un estándar calificado y que se acople a la necesidad del departamento con la finalidad de aplicar nuevas prácticas de desarrollo.

### **3.6.2 Propuesta del modelo de calidad**

El modelo a proponer consta de la unión de procesos del modelo de calidad la norma ISO 12207:2008 donde se creará un marco de trabajo que permitirá a la Unidad de Sistemas de la Universidad mencionando que en este apartado se integra el estudio de los estándares que se adecuan de mejor manera a la investigación realizada y que dio como resultado la Norma ISO 12207 para evaluar el proceso de desarrollo de software, y la Norma ISO 15504 que permitirá medir la madurez del proceso de desarrollo de software, la metodología tradicional y el modelo de ciclo de vida tradicional que es donde se define el desarrollo de software desde la fase inicial hasta la implementación del software desarrollado, que al unir todas estas actividades que tiene cada elemento mencionado garantizará la necesidad de desarrollar software de calidad en la Unidad de Sistemas de la Universidad Técnica de Machala.

Con base a lo anteriormente mencionado se tomará la norma ISO 15504 siendo esta norma enfocada a la evaluación de procesos para pequeños grupos de desarrollo de software mediante niveles de madurez.

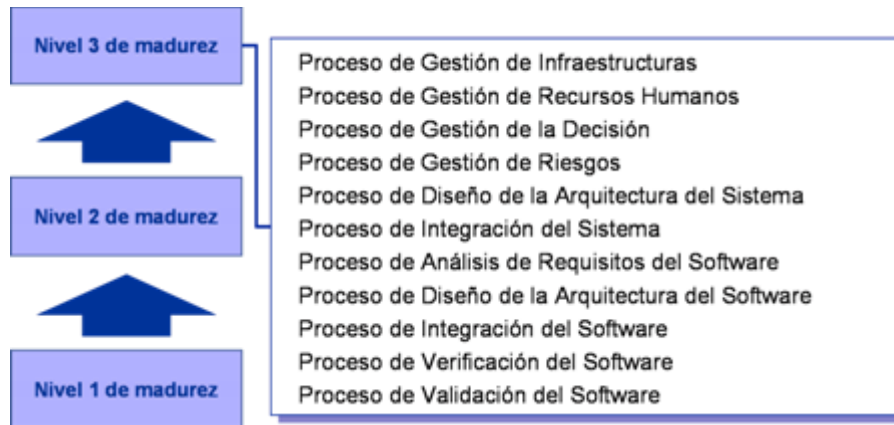
Observando en la siguiente tabla los niveles de madurez de la Norma

**Tabla 3.2.**

**Niveles de madurez de la Norma ISO 15504**

<b>Nivel</b>	<b>Estado</b>
Nivel 0 - Organización inmadura	La organización no tiene un implementación efectiva de los procesos
Nivel 1 - Organización básica	La organización implementa y alcanza los objetivos de los procesos
Nivel 2 - Organización gestionada	La organización gestiona los procesos y los productos de trabajo se establecen, controlan y mantienen
Nivel 3 - Organización establecida	La organización utiliza procesos adaptados basados en estándares
Nivel 4 - Organización predecible	La organización gestiona cuantitativamente los procesos
Nivel 5 - Organización optimizando	La organización mejora continuamente los procesos para cumplir los objetivos de negocio

De la misma forma encontramos los procesos que contiene cada nivel de **madurez**.



**Figura 3.6. Procesos de los niveles de madurez de la norma ISO 15504**

**FUENTE:** [26]

Para la creación del modelo se considera los siguientes Áreas del proceso

Según esta norma las actividades que se pueden llevar a cabo durante el ciclo de vida del SW se pueden agrupar en:

- 5 procesos principales
- 8 procesos de soporte
- 3 procesos de organización o generales

En el **Figura 3.7.** Se muestra los subprocesos con los que cuenta esta norma y de los que se seleccionará para el modelo propuesto

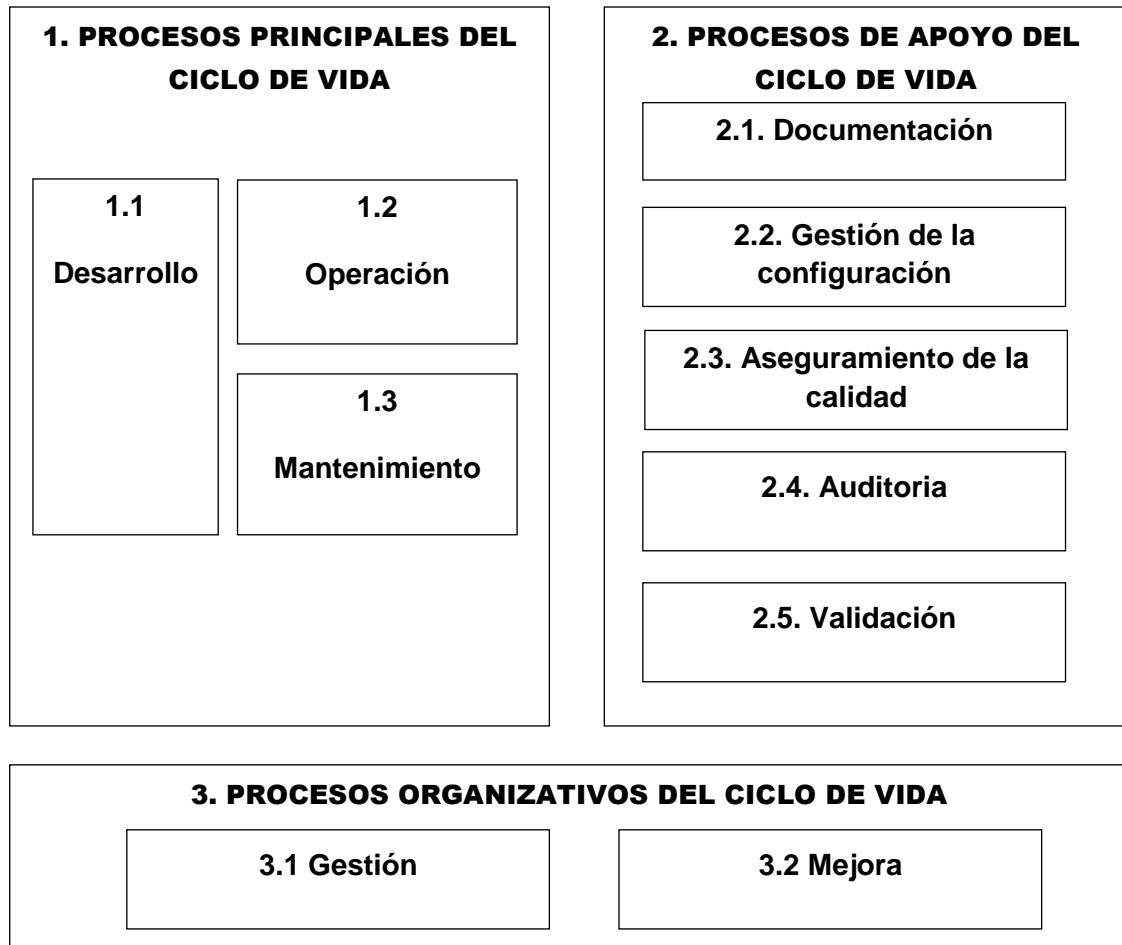




**Figura 3.7. Procesos de la norma ISO 12207:2008**

**FUENTE: [25]**

Cuando se ha seleccionado la Norma a aplicar y el modelo que permitirá evaluar la calidad de desarrollo de software a continuación se demuestra gráficamente que proceso de la ISO 12207: 2008 se considerara para la propuesta del modelo de calidad a aplicar en la Unidad de Sistemas de la Universidad Técnica de Machala.



**Figura 3.8. Procesos de la norma ISO 12207 a implementar en el departamento de sistemas de la Universidad Técnica de Machala**

### 3.6.3 Estructura del modelo de calidad

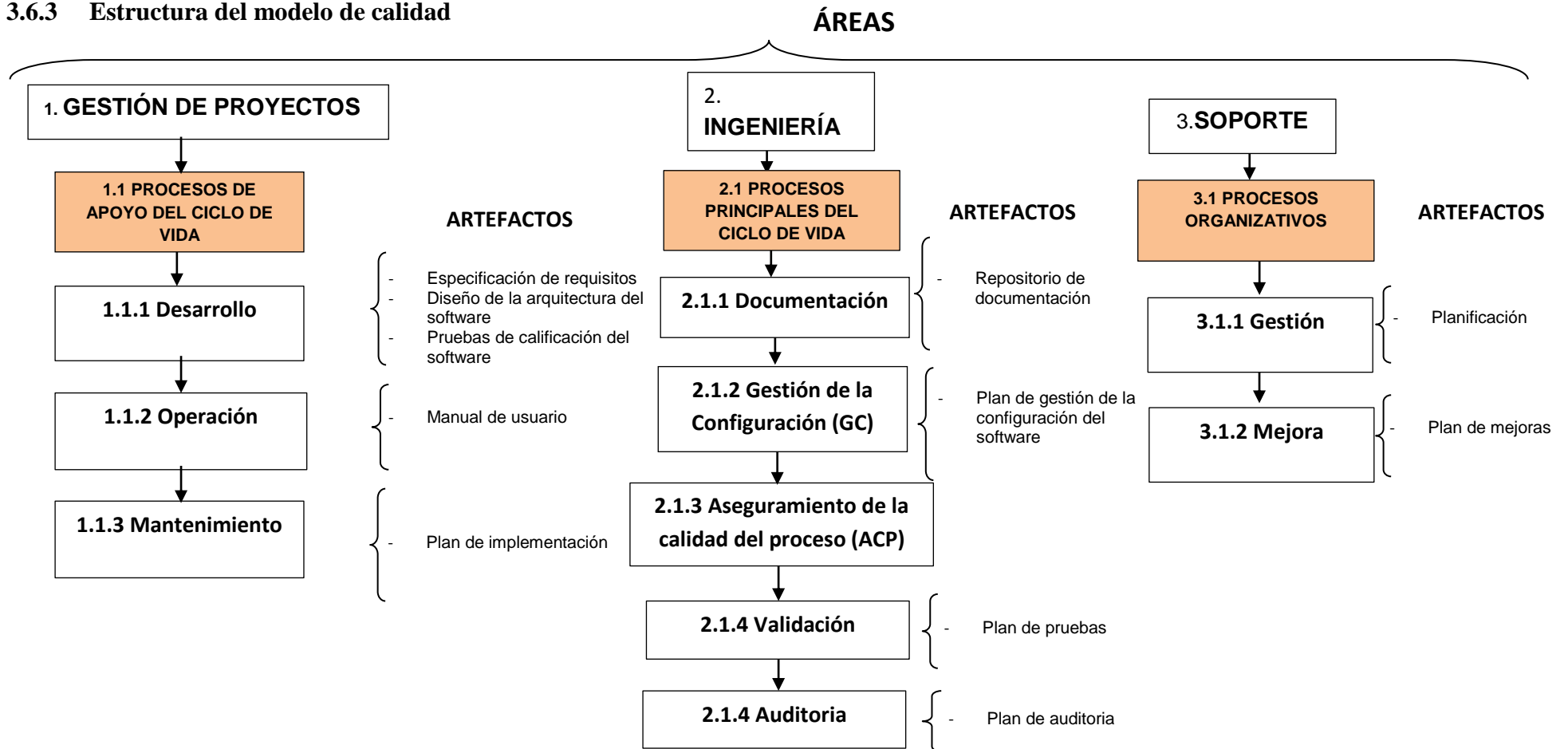


Figura 3.9. Propuesta del modelo de calidad para el proceso de desarrollo de sistema en la UTMACH

### 3.6.2.3 Actividades

#### 1. Gestión de proyectos

##### 1.1 Procesos de apoyo del ciclo de vida

###### 1.1.1 Desarrollo

- Análisis de requerimientos del software
- Diseño de la arquitectura del software
- Codificación y pruebas del software
- Integración del software
- Pruebas de calificación del software
- Apoyo a la aceptación del software

###### Artefactos de este proceso

- Especificación de requisitos (ERS IEEE 830)
- Diseño de la arquitectura del sistema
- Plan de pruebas de calificación

###### 1.1.2 Operación

- Implementación del proceso
- Pruebas de operación
- Operación del sistema
- Soporte al usuario

###### Artefactos de este proceso

- Manual de usuario

###### 1.1.3 Mantenimiento

- Implementación del proceso
- Análisis de problemas y modificaciones
- Implementación de las modificaciones
- Revisión / aceptación del mantenimiento
- Migración

###### Artefactos de este proceso

- Manual de usuario

## **2. Ingeniería**

### **2.1 Procesos principales del ciclo de vida**

#### **2.1.1 Documentación**

- Implementación del proceso

##### **Artefactos de este proceso**

- Repositorios de Documentación

#### **2.1.2 Gestión de configuración**

- Implementación del proceso
- Identificación de la configuración
- Control de la configuración
- Determinación del estado de la configuración
- Evaluación de la configuración
- Gestión de liberaciones y entregas

##### **Artefactos de este proceso**

- Plan de gestión de la configuración del software

#### **2.1.3 Aseguramiento de calidad**

- Implementación del proceso
- Aseguramiento del proceso
- Aseguramiento del sistema de calidad

#### **2.1.4 Validación**

- Implementación del proceso
- Validación

##### **Artefactos de este proceso**

- Plan de pruebas

#### **2.1.5 Auditoria**

- Implementación del proceso
- Auditoria

### **Artefactos de este proceso**

- Plan de auditoria

## **3. Soporte**

### **3.1 Procesos organizativos**

#### **3.1.1 Gestión**

- Inicio y definición de alcance
- Planificación
- Ejecución y control
- Revisión y evaluación
- Terminación

#### **3.1.2 Mejora**

- Establecimiento del proceso
- Evaluación del proceso
- Mejora del proceso

Una vez que se ha definido la estructura del modelo de calidad se procede a detallar cada uno de estos procesos y los factores a considerar al momento de la aplicación del modelo.

## **1. Gestión de proyectos**

### **1.1 Procesos de apoyo del ciclo de vida**

#### **1.1.1 Desarrollo**

En este proceso se define las actividades que tiene el ciclo de vida del software.

- **Análisis de los requerimientos software (ERS)**

Por cada aplicación a desarrollar consta de las siguientes tareas:

- El jefe de la Unidad de Sistemas deberá recopilar los requisitos del sistema de acuerdo a la necesidad de la Institución.
- También deberá documentar los requisitos en base a especificaciones funcionales

Se determina los requerimientos, diseño, codificación, integración, pruebas y aceptación relacionados con el software.

El desarrollador lleva a cabo las actividades de este proceso de acuerdo a lo planteado por la universidad. [Anexo 2]

Sus actividades son:

- Diseño de la arquitectura del sistema
- Diseño de la arquitectura del sistema.
- Diseño detallado del sistema.
- Codificación y pruebas del sistema.
- Integración del sistema.
- Pruebas de calificación del sistema.
- Integración del sistema.
- Pruebas de calificación del sistema.
- Instalación del sistema.
- Apoyo a la aceptación del sistema.

### **1.1.2 Operación**

El proceso cubre la capacitación a los usuarios ya que la operación del producto software está integrada a la operación del software, las actividades y tareas de este proceso hacen referencia al software.

Sus actividades son:

- Implementación del proceso.
- Pruebas de operación.
- Operación del software.
- Soporte al usuario.

### **1.1.3 Mantenimiento**

Este proceso se inicia cuando el sistema sufre modificaciones en el código fuente y la documentación asociada, debido a un problema o a la necesidad de mejora o adaptación. El objetivo es modificar el producto software existente preservando su integridad.

Sus actividades son:

- Implementación del proceso.
- Análisis de problemas y modificaciones.
- Implementación de las modificaciones.
- Revisión/aceptación del mantenimiento.

## **2. INGENIERÍA**

### **2.1 Procesos principales del ciclo de vida**

#### **2.1.1 Documentación**

En esta actividad cubre con el detalle de lo que se revisaría en este proceso permite al jefe de la Unidad de Sistemas proporcionar mecanismos para establecer y mantener un entorno de colaboración entre equipos facilita un método común para gestionar el proyecto cuantitativamente y anticipándose a los problemas que se pueden presentar durante el desarrollo.

Además toma en consideración los recursos y el tiempo que tardarán en hacer el entregable.



- **Cronograma**

Dentro del cronograma de cada proyecto se debe considerar la fecha de inicio y fin en que se efectuará el proyecto, los recursos a utilizar, y debe existir un documento que permita ir controlando los tiempos de desarrollo.

- **Seguimiento y control del proyecto**

El seguimiento y control del proyecto tiene como principal función el control de todas las actividades de desarrollo del sistema. Es una de las labores que más se destaca en todo el desarrollo, ya que un apropiado control hace posible evitar desviaciones en cuanto al cronograma planificado, o al menos lograr detectarlas antes de ocasionar un desorden en el desarrollo.

Para ejercer un correcto seguimiento y control del proyecto es necesario que el Jefe de la Unidad de Sistemas dedique el tiempo necesario para vigilar el estado de cada una de las tareas que se están elaborando, prestando interés total a aquellas que están sufriendo algún retraso.

Las Actividades de Seguimiento y Control del proyecto se aplican desde la asignación de las tareas al equipo de proyecto. Las tareas propias del Seguimiento y Control del proyecto se realizan a medida que se ejecutan las distintas tareas de los procesos de Especificación, Diseño, Construcción, y Pruebas del Sistema.

### **2.1.2 Gestión de la Configuración**

Consiste en aplicar procedimientos Técnicos y administrativos a lo largo del ciclo de vida del sistema para: identificar, definir y establecer la línea base de los elementos del sistema; controlar modificaciones y las liberaciones de interacciones; registrar e informar los errores que se den en alguna parte del sistema.

Sus actividades son:

- Implementación del proceso.
- Identificación de la configuración.
- Control de la configuración.
- Determinación del estado de la configuración.
- Evaluación de la configuración.
- Gestión de las interacciones entregables.

### **2.1.3 Aseguramiento de la calidad**

Asegurar mediante alguna métrica que los productos y procesos software del ciclo de vida del proyecto son conformes los requerimientos especificados.

Sus actividades son:

- Implementación del proceso.
- Aseguramiento del producto.
- Aseguramiento del proceso.
- Aseguramiento del sistema de calidad.

### **2.1.4 Validación**

Es un proceso de pruebas donde se determine si los requerimientos y el software, cumple con lo estipulado en la especificación de los mismos.

La validación se puede llevar a cabo en etapas tempranas. Este proceso se puede llevar a cabo como parte del apoyo a la aceptación del producto.

Sus actividades son:

- Implementación del proceso.
- Validación.

## **3. SOPORTE**

### **3.1 Procesos Organizativos**

#### **3.1.1 Gestión**

Contiene las actividades que pueden ser aplicadas por cualquier parte que tenga que gestionar sus respectivos procesos. El jefe de sistemas será responsable de la gestión del producto, gestión del proyecto y gestión de las tareas de los procesos aplicables, tales como el de adquisición, suministro, desarrollo, operación, mantenimiento o soporte.

Sus actividades son:

- Inicio y definición del alcance.
- Planificación.
- Ejecución y control.
- Revisión y evaluación.

- Finalización.

### **1.1.2 Mejora**

Consta de un proceso para establecer, evaluar, medir, controlar y mejorar un proceso del ciclo de vida del software.

Sus actividades son:

- Establecimiento del proceso.
- Evaluación del proceso.
- Mejora del proceso.

### **3.7 Conclusiones**

En este capítulo se presentó la propuesta del modelo de calidad para el proceso de desarrollo de software que se implementará en la Unidad de sistemas de la Universidad Técnica de Machala y el mismo ha sido resultado de una investigación bibliográfica realizada en el capítulo anterior.

## CAPÍTULO IV

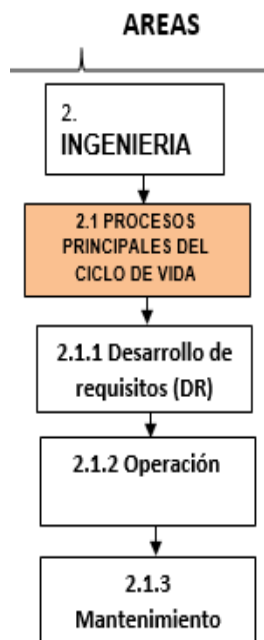
### 4. IMPLEMENTACIÓN DEL MODELO

#### 4.1 Introducción

En este capítulo se considera una fase del modelo de calidad a aplicar en la Unidad de Sistemas de la Universidad Técnica de Machala y para ello se tomó la parte de Especificación de requisitos donde se detalla la actividad que debería contener cada proceso.

#### 4.2 Aplicación del modelo de calidad propuesto en la fase de requisitos

Recordando la estructura que se planteó para la elaboración del modelo de calidad en el proceso de desarrollo software de la Unidad de Sistemas en la Universidad Técnica de Machala estando dividido en tres áreas tales como: Gestión de proyectos, Ingeniería y Soporte, se consideró el área de Ingeniería que está subdividida en lo siguiente:



**Figura 4.1. Área considerada para la implementación del modelo**

### 4.3 Procesos principales del ciclo de vida

Por tratarse de un modelo que se partirá aplicando a medida que la Unidad de Sistemas pase por el proceso de Adaptación se tomó aleatoriamente la especificación de requisitos para aplicar las actividades que enfocan en este proceso, como se detalla a continuación.

#### 4.3.1 Desarrollo de requisitos

- **Implementación del proceso.**

En la implementación del proceso el Jefe de la Unidad de Sistemas se entrevista con el cliente y recolecta requisitos de acuerdo a la necesidad de cada cliente.

- **Análisis de los requerimientos del software.**

Para esta actividad la implementación del proceso se considera el estándar para la especificación de requisitos **IEEE 830** y se detalla cada actividad que contiene el mismo ver **Anexo #2**

**Tabla 4.1**

#### Requerimientos del Software

Nombre	Descripción
Especificación de Requerimientos	<p>Se compone de una introducción y una descripción de requerimientos.</p> <p><b>1. Introducción:</b></p> <p>Descripción general del sistema y su uso en el ámbito de negocio del cliente.</p> <p><b>2. Descripción de requerimientos:</b></p> <p><b>a) Funcionales:</b> Necesidades establecidas que debe satisfacer el software cuando es usado en condiciones específicas. Las funcionalidades deben ser adecuadas, exactas y seguras.</p> <p><b>b) Interfaz con usuario:</b> Definición de aquellas características de la interfaz de usuario que permiten que el sistema sea fácil de aprender, entender y usar; que genere satisfacción al usuario, donde pueda desempeñar</p>

	<p>su tarea eficientemente. Incluyendo la descripción del prototipo de la interfaz.</p> <p><b>c) Interfaces externas:</b> Definición de las interfaces con otro software o con hardware.</p> <p><b>d) Confiabilidad:</b> Especificación del nivel de desempeño del sistema con respecto a la madurez, tolerancia a fallas y recuperación.</p> <p><b>e) Eficiencia:</b> Especificación del nivel de desempeño del software con respecto al tiempo y a la utilización de recursos.</p> <p><b>f) Mantenimiento:</b> Descripción de los elementos que facilitarán la comprensión y la realización de las modificaciones futuras del software.</p> <p><b>g) Portabilidad:</b> Descripción de las características del sistema que permitan su transferencia de un ambiente a otro.</p> <p><b>h) Seguridad</b> Mecanismos de protección del sistema contra manipulación y utilización indebida (confidencialidad, integridad, virus, etc.)</p> <p><b>i) Restricciones de diseño y construcción:</b> Necesidades impuestas por el cliente.</p>
--	--

## Resultado

Al implementar la fase del desarrollo de requisitos se obtuvo que:

- Determinan un estándar para la recolección de requisitos
- Elaboran un documento que describe el ámbito del software
- Describe los tipos de requisitos de software en el documento

- **Diseño de la arquitectura del software**

En esta actividad se elabora la arquitectura del software basado en el estándar **IEEE – 1471-2000** donde se elabora una descripción comprensiva arquitectónica del software **Ver Anexo #3** , usando un número de vistas diferentes para representar los distintos aspectos que se requieren para capturar y transportar las decisiones significativas que han sido hechas sobre el software, tales como:

- Se identifica los stakeholders sus roles y sus responsabilidades.
- Vistas de la arquitectura: casos de usos, diagrama de clases diseño arquitectónico.
- Descripción de los módulos del sistema

- **Diseño detallado del sistema.**

En esta actividad se desarrolla la base de datos con la que se va a implementar en el sistema informático.

- **Codificación y pruebas del sistema.**

En esta actividad los programadores son los encargados de plasmar el código fuente con el que de acuerdo a los requisitos planteados se irá formando el nuevo módulo que se agregue al sistema informático.

**Tabla 4.2.**

**Plan de pruebas**

**Resultados**

Nombre	Descripción
<b>Plan de pruebas del sistema</b>	Identificación de pruebas requeridas para el cumplimiento de los requerimientos especificados.
<b>Reporte de Pruebas del Sistema</b>	Registro de participantes, fecha, lugar, duración y de defectos encontrados.

- **Pruebas de calificación del sistema.**

En esta actividad se crea los reportes de las pruebas de validación y verificación del sistema.

**Tabla 4.3.****Pruebas de calificación del sistema**

<b>Nombre</b>	<b>Descripción</b>
<b>Reporte(s) de Verificación</b>	Registro de participantes, fecha, lugar, duración y defectos encontrados.
<b>Reporte(s) de Validación</b>	Registro de participantes, fecha, lugar, duración y defectos encontrados.

**4.4 Conclusiones**

Al aplicar una área del modelo de calidad de software a la Unidad de Sistemas de la Universidad Técnica de Machala se logró determinar que aplican criterios de calidad en la especificación de requisitos de software pero aún falta definirlos concretamente y a estos se puede madurar más el proceso implementando en su totalidad el modelo de calidad propuesto para el proceso de desarrollo de software.



## CAPÍTULO V

### 5. RESULTADOS DE LA APLICACIÓN

#### 5.1 Introducción

En este capítulo se presenta los resultados obtenidos al aplicar el modelo de calidad diseñado para la Unidad de Sistemas de la Universidad Técnica de Machala en el área de Sistemas en la especificación de requisitos y en medir la satisfacción de los clientes con el sistema SIUTMACH, para este proceso se elaboró una encuesta para la siguiente población.

- Desarrolladores de la Unidad de sistemas, UTM.
- Estudiantes
- Personal Administrativo
- Personal Docente

#### 5.2 Procesamiento de los resultados.

La presente investigación se direcciona a comprobar de qué forma se relaciona el modelo de calidad con el proceso de desarrollo de software.

Se establece niveles de cualificación para cada pregunta elaborada en el instrumento.

Valoración entre 1 y 4 para cada pregunta

1	2	3	4
Nunca	Casi nunca	A Veces	Siempre

Para la interpretación se ha dividido los resultados en 4 grupos de porcentaje y dependiendo del puntaje en cada interrogativa se ubicará la calificación del modelo propuesto.

##### 5.2.1 Análisis y resultados del instrumento aplicado a los desarrolladores

La interpretación hace referencia al instrumento aplicado a los desarrolladores y al jefe de la Unidad de Sistemas de la Universidad Técnica de Machala.

Los resultados obtenidos son:

**Tabla 5.1.**

**Resultados de aprobación de los desarrolladores y jefe de sistemas del modelo de calidad**

	VALORACIÓN	Frecuencia	Porcentaje	Porcentaje válido
<b>Válidos</b>	Siempre	4	57,1	57,1
	Casi Siempre	3	42,9	42,9
	Casi Nunca	0	0,00	
	Nunca	0	0,00	
	Total	7	100,0	100,0



**Figura 5.1. Criterio de aceptación del modelo de calidad en La Unidad de Sistemas de la UTMACH**

Según el **Figura 5.1.** Se puede observar que el grado de aceptación por parte del personal del desarrollo de la Unidad de sistemas consiguiendo un 97% de aceptación considerando como aceptables a la valoración de siempre y casi siempre.

### **5.2.2 Análisis y resultados de las encuestas aplicadas a los usuarios del sistema informático de la Unidad de Sistemas**

En este apartado se presenta el análisis y la interpretación de la información de la encuesta aplicada a los desarrolladores y jefe de sistemas se muestra en el anexo 4.

### 5.3 Prueba de hipótesis con Chi cuadrado

#### 5.3.1 Planteamiento de la hipótesis

Al aplicar la prueba de hipótesis medirá el grado de relación entre las dos variables de la investigación y se logra determinar la pertinencia y dependencia para poder ejecutar.

**Hipótesis investigativa:** Si se implementa un modelo de calidad para el sistema desarrollado en la Unidad de Sistemas de la Universidad Técnica de Machala permitirá optimizar la gestión del proceso de desarrollo de software que se aplica en esa dependencia.

#### **Variable Independiente**

Modelo de calidad para el sistema desarrollado en la Unidad de Sistemas de la Universidad Técnica de Machala.

#### **Variable dependiente**

Optimización en la gestión del proceso de desarrollo de software

**Hipótesis Nula (Ho):** No se relaciona el modelo de calidad con la optimización del proceso de desarrollo de software.

**Hipótesis Alternativa (Ha):** Se relaciona directamente el modelo de calidad en la optimización del proceso de desarrollo de software.

#### **II: Regla teórica para toma de decisión**

Si el Valor  $p \geq 0.05$  se Acepta la Hipótesis Nula (Ho). Si el Valor  $p < 0.05$  se Acepta la Hipótesis Alternativa (Ha).

#### 5.3.2 Estadística de contraste de hipótesis

La contrastación de hipótesis se utilizó la correlación de Pearson entre la variable independiente: modelo de calidad y la variable dependiente: proceso de desarrollo de software tomando como base a la encuesta aplicada a los desarrolladores de la Unidad de Sistemas Ver **Anexo # 6**.

Tabla 5.2.

**Variable Independiente Modelo de calidad**

VALORACIÓN	Desarrollador	Jefe del proyecto	Total
Siempre	4	3	7
Casi Siempre	3	4	7
Casi Nunca	0	0	0
Nunca	0	0	0

VALORACIÓN	Frecuencia	Porcentaje	Porcentaje válido
<b>Válidos</b> Siempre	4	71,4	71,4
Casi Siempre	3	28,6	28,6
Casi Nunca	0	0,00	0,00
Nunca	0	0,00	0,00
Total	7	100,0	100,0

**5.3.3 Cálculo del valor Chi-cuadrado**

Después de haber obtenido las frecuencias observada y esperada se procede a calcular el Chi – cuadrado que es método de Pearson utilizando la herramienta SPSS que permitirá definir si existe o no relación entre las dos variables utilizadas para esta investigación.

**¿Considera que el modelo de calidad propuesto es fácil de entender? \***

**¿Considera usted que aplicando el modelo de calidad propuesto se organice de mejor manera el proceso de desarrollo de software?**

**Pruebas de chi-cuadrado**

	Valor	g	Sig. asintótica (bilateral)	Sig. exacta (bilateral)	Sig. exacta (unilateral)
Chi-cuadrado de Pearson	,058 <sup>a</sup>	1	,809		
Corrección por continuidad	,000	1	1,000		
Razón de verosimilitudes	,058	1	,810		
Estadístico exacto de				1,000	,714

CONTINÚA 

Fisher					
Asociación lineal por lineal	,050	1	,823		
N de casos válidos	7				
<p>a. 4 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es ,86.</p> <p>b. Calculado sólo para una tabla de 2x2.</p>					

**¿Considera que el modelo de calidad propuesto es fácil de entender? \* ¿Cree usted que se alcanzaría a definir claramente las actividades a desarrollar para cada proyecto planteado aplicando el modelo de calidad?**

### Pruebas de chi-cuadrado

	Valor	gl	Sig. asintótica (bilateral)	Sig. exacta (bilateral)	Sig. exacta (unilateral)
Chi-cuadrado de Pearson	2,100 <sup>a</sup>	1	,147		
Corrección por continuidad <sup>b</sup>	,365	1	,546		
Razón de verosimilitudes	2,831	1	,092		
Estadístico exacto de Fisher				,429	,286
Asociación lineal por lineal	1,800	1	,180		
N de casos válidos	7				

a. 4 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es ,86.

b. Calculado sólo para una tabla de 2x2.

**¿Considera que el modelo de calidad propuesto es fácil de entender? \***  
**¿Considera al modelo propuesto para el proceso de desarrollo de software que está absorbiendo todas las funciones prescindibles para la actividad que realiza en la Unidad de Sistemas de la UTMACH?**

#### Pruebas de chi-cuadrado

	Valor	gl	Sig. asintótica (bilateral)	Sig. exacta (bilateral)	Sig. exacta (unilateral)
Chi-cuadrado de Pearson	,058 <sup>a</sup>	1	,809		
Corrección por continuidad <sup>b</sup>	,000	1	1,000		
Razón de verosimilitudes	,058	1	,810		
Estadístico exacto de Fisher				1,000	,714
Asociación lineal por lineal	,050	1	,823		
N de casos válidos	7				

a. 4 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es ,86.

b. Calculado sólo para una tabla de 2x2.

**¿Considera que el modelo de calidad propuesto es fácil de entender? \* ¿Cree Usted que las actividades que se realizan actualmente en la Unidad de Sistemas mejoraran al aplicar el modelo de calidad propuesto?**

**Pruebas de chi-cuadrado**

	Valor	gl	Sig. asintótica (bilateral)
Chi-cuadrado de Pearson	3,733 <sup>a</sup>	2	,155
Razón de verosimilitudes	4,557	2	,102
Asociación lineal por lineal	2,500	1	,114
N de casos válidos	7		

a. 6 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es ,29.

**Respecto al modelo de calidad considera que se puede adaptar a otros proyectos ¿Considera al modelo de calidad entendible y preciso con las actividades a realizar? \* ¿Cree Usted que el modelo propuesto tiene factores que comprenden la calidad de software?**

### Pruebas de chi-cuadrado

	Valor	gl	Sig. asintótica (bilateral)	Sig. exacta (bilateral)	Sig. exacta (unilateral)
Chi-cuadrado de Pearson	1,556 <sup>a</sup>	1	,212		
Corrección por continuidad <sup>b</sup>	,024	1	,876		
Razón de verosimilitudes	1,923	1	,166		
Estadístico exacto de Fisher				,429	,429
Asociación lineal por lineal	1,333	1	,248		
N de casos válidos	7				

a. 4 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es ,43.

b. Calculado sólo para una tabla de 2x2.



**Respecto al modelo de calidad considera que se puede adaptar a otros proyectos ¿Considera al modelo de calidad entendible y preciso con las actividades a realizar? \* ¿Considera usted que aplicando el modelo de calidad propuesto se organice de mejor manera el proceso de desarrollo de software?**

#### Pruebas de chi-cuadrado

	Valor	gl	Sig. asintótica (bilateral)	Sig. exacta (bilateral)	Sig. exacta (unilateral)
Chi-cuadrado de Pearson	1,215 <sup>a</sup>	1	,270		
Corrección por continuidad <sup>b</sup>	,109	1	,741		
Razón de verosimilitudes	1,243	1	,265		
Estadístico exacto de Fisher				,486	,371
Asociación lineal por lineal	1,042	1	,307		
N de casos válidos	7				

a. 4 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es 1,29.

b. Calculado sólo para una tabla de 2x2.

**Respecto al modelo de calidad considera que se puede adaptar a otros proyectos ¿Considera al modelo de calidad entendible y preciso con las actividades a realizar? \* ¿Cree usted que se alcanzaría a definir claramente las actividades a desarrollar para cada proyecto planteado aplicando el modelo de calidad?**

<b>Pruebas de chi-cuadrado</b>					
	Valor	gl	Sig. asintótica (bilateral)	Sig. exacta (bilateral)	Sig. exacta (unilateral)
Chi-cuadrado de Pearson	,194 <sup>a</sup>	1	,659		
Corrección por continuidad <sup>b</sup>	,000	1	1,000		
Razón de verosimilitudes	,196	1	,658		
Estadístico exacto de Fisher				1,000	,629
Asociación lineal por lineal	,167	1	,683		
N de casos válidos	7				
<p>a. 4 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es 1,29.</p> <p>b. Calculado sólo para una tabla de 2x2.</p>					

**Respecto al modelo de calidad considera que se puede adaptar a otros proyectos ¿Considera al modelo de calidad entendible y preciso con las actividades a realizar? \* ¿Considera al modelo propuesto para el proceso de desarrollo de software que está absorbiendo todas las funciones prescindibles para la actividad que realiza en la Unidad de Sistemas de la UTMACH?**

<b>Pruebas de chi-cuadrado</b>					
	Valor	gl	Sig. asintótica (bilateral)	Sig. exacta (bilateral)	Sig. exacta (unilateral)
Chi-cuadrado de Pearson	,194 <sup>a</sup>	1	,659		
Corrección por continuidad <sup>b</sup>	,000	1	1,000		
Razón de verosimilitudes	,196	1	,658		
Estadístico exacto de Fisher				1,000	,629
Asociación lineal por lineal	,167	1	,683		
N de casos válidos	7				
<p>a. 4 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es 1,29.</p> <p>b. Calculado sólo para una tabla de 2x2.</p>					

**Respecto al modelo de calidad considera que se puede adaptar a otros proyectos  
¿Considera al modelo de calidad entendible y preciso con las actividades a  
realizar? \* ¿Cree Usted que las actividades que se realizan actualmente en la  
Unidad de Sistemas mejoraran al aplicar el modelo de calidad propuesto?**

<b>Pruebas de chi-cuadrado</b>				
		Valor	gl	Sig. asintótica (bilateral)
Chi-cuadrado de Pearson	de	1,556 <sup>a</sup>	2	,459
Razón de verosimilitudes	de	1,923	2	,382
Asociación lineal por lineal		,750	1	,386
N de casos válidos		7		
a. 6 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es ,43.				

**¿Considera al modelo de calidad entendible y precisa con las actividades a realizar? \* ¿Cree Usted que el modelo propuesto tiene factores que comprenden la calidad de software?**

<b>Pruebas de chi-cuadrado</b>					
	Valor	gl	Sig. asintótica (bilateral)	Sig. exacta (bilateral)	Sig. exacta (unilateral)
Chi-cuadrado de Pearson	,875 <sup>a</sup>	1	,350		
Corrección por continuidad <sup>b</sup>	,000	1	1,000		
Razón de verosimilitudes	1,243	1	,265		
Estadístico exacto de Fisher				1,000	,571
Asociación lineal por lineal	,750	1	,386		
N de casos válidos	7				
<p>a. 4 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es ,43.</p> <p>b. Calculado sólo para una tabla de 2x2.</p>					

**¿Considera al modelo de calidad entendible y preciso con las actividades a realizar? \* ¿Considera usted que aplicando el modelo de calidad propuesto se organice de mejor manera el proceso de desarrollo de software?**

<b>Pruebas de chi-cuadrado</b>					
	Valor	gl	Sig. asintótica (bilateral)	Sig. exacta (bilateral)	Sig. exacta (unilateral)
Chi-cuadrado de Pearson	,194 <sup>a</sup>	1	,659		
Corrección por continuidad <sup>b</sup>	,000	1	1,000		
Razón de verosimilitudes	,196	1	,658		
Estadístico exacto de Fisher				1,000	,629
Asociación lineal por lineal	,167	1	,683		
N de casos válidos	7				
<p>a. 4 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es 1,29.</p> <p>b. Calculado sólo para una tabla de 2x2.</p>					

**¿Considera al modelo de calidad entendible y preciso con las actividades a realizar? \* ¿Cree usted que se alcanzaría a definir claramente las actividades a desarrollar para cada proyecto planteado aplicando el modelo de calidad?**

### Pruebas de chi-cuadrado

	Valor	gl	Sig. asintótica (bilateral)	Sig. exacta (bilateral)	Sig. exacta (unilateral)
Chi-cuadrado de Pearson	,194 <sup>a</sup>	1	,659		
Corrección por continuidad <sup>b</sup>	,000	1	1,000		
Razón de verosimilitudes	,196	1	,658		
Estadístico exacto de Fisher				1,000	,629
Asociación lineal por lineal	,167	1	,683		
N de casos válidos	7				

a. 4 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es 1,29.

b. Calculado sólo para una tabla de 2x2.

**¿Considera al modelo de calidad entendible y precisa con las actividades a realizar? \* ¿Considera al modelo propuesto para el proceso de desarrollo de software que está absorbiendo todas las funciones prescindibles para la actividad que realiza en la Unidad de Sistemas de la UTMACH?**

<b>Pruebas de chi-cuadrado</b>					
	Valor	gl	Sig. asintótica (bilateral)	Sig. exacta (bilateral)	Sig. exacta (unilateral)
Chi-cuadrado de Pearson	1,215 <sup>a</sup>	1	,270		
Corrección por continuidad <sup>b</sup>	,109	1	,741		
Razón de verosimilitudes	1,243	1	,265		
Estadístico exacto de Fisher				,486	,371
Asociación lineal por lineal	1,042	1	,307		
N de casos válidos	7				
<p>a. 4 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es 1,29.</p> <p>b. Calculado sólo para una tabla de 2x2.</p>					



**¿Considera al modelo de calidad entendible y precisa con las actividades a realizar? \* ¿Cree Usted que las actividades que se realizan actualmente en la Unidad de Sistemas mejoraran al aplicar el modelo de calidad propuesto?**

**Pruebas de chi-cuadrado**

	Valor	gl	Sig. asintótica (bilateral)
Chi-cuadrado de Pearson	1,556 <sup>a</sup>	2	,459
Razón de verosimilitudes	1,923	2	,382
Asociación lineal por lineal	,750	1	,386
N de casos válidos	7		

a. 6 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es ,43.

**¿Considera al modelo propuesto para el proceso de desarrollo de software que está absorbiendo todas las funciones prescindibles para la actividad que realiza en la Unidad de Sistemas de la UTMACH? \* ¿Cree Usted que el modelo propuesto tiene factores que comprenden la calidad de software?**

<b>Pruebas de chi-cuadrado</b>					
	Valor	gl	Sig. asintótica (bilateral)	Sig. exacta (bilateral)	Sig. exacta (unilateral)
Chi-cuadrado de Pearson	7,000 <sup>a</sup>	1	,008		
Corrección por continuidad <sup>b</sup>	1,215	1	,270		
Razón de verosimilitudes	5,742	1	,017		
Estadístico exacto de Fisher				,143	,143
Asociación lineal por lineal	6,000	1	,014		
N de casos válidos	7				
<p>a. 3 casillas (75,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es ,14.</p> <p>b. Calculado sólo para una tabla de 2x2.</p>					

**¿Considera al modelo propuesto para el proceso de desarrollo de software que está absorbiendo todas las funciones prescindibles para la actividad que realiza en la Unidad de Sistemas de la UTMACH? \* ¿Considera usted que aplicando el modelo de calidad propuesto se organice de mejor manera el proceso de desarrollo de software?**

### Pruebas de chi-cuadrado

	Valor	gl	Sig. asintótica (bilateral)	Sig. exacta (bilateral)	Sig. exacta (unilateral)
Chi-cuadrado de Pearson	1,556 <sup>a</sup>	1	,212		
Corrección por continuidad <sup>b</sup>	,024	1	,876		
Razón de verosimilitudes	1,923	1	,166		
Estadístico exacto de Fisher				,429	,429
Asociación lineal por lineal	1,333	1	,248		
N de casos válidos	7				

a. 4 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es ,43.

b. Calculado sólo para una tabla de 2x2.

**¿Considera al modelo propuesto para el proceso de desarrollo de software que está absorbiendo todas las funciones prescindibles para la actividad que realiza en la Unidad de Sistemas de la UTMACH? \* ¿Cree usted que se alcanzaría a definir claramente las actividades a desarrollar para cada proyecto planteado aplicando el modelo de calidad?**

#### Pruebas de chi-cuadrado

	Valor	gl	Sig. asintótica (bilateral)	Sig. exacta (bilateral)	Sig. exacta (unilateral)
Chi-cuadrado de Pearson	,875 <sup>a</sup>	1	,350		
Corrección por continuidad <sup>b</sup>	,000	1	1,000		
Razón de verosimilitudes	1,243	1	,265		
Estadístico exacto de Fisher				1,000	,571
Asociación lineal por lineal	,750	1	,386		
N de casos válidos	7				

a. 4 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es ,43.

b. Calculado sólo para una tabla de 2x2.

**¿Considera al modelo propuesto para el proceso de desarrollo de software que está absorbiendo todas las funciones prescindibles para la actividad que realiza en la Unidad de Sistemas de la UTMACH? \* ¿Considera al modelo propuesto para el proceso de desarrollo de software que está absorbiendo todas las funciones prescindibles para la actividad que realiza en la Unidad de Sistemas de la UTMACH?**

#### Pruebas de chi-cuadrado

	Valor	gl	Sig. asintótica (bilateral)	Sig. exacta (bilateral)	Sig. exacta (unilateral 1)
Chi-cuadrado de Pearson	,875 <sup>a</sup>	1	,350		
Corrección por continuidad <sup>b</sup>	,000	1	1,000		
Razón de verosimilitudes	1,243	1	,265		
Estadístico exacto de Fisher				1,000	,571
Asociación lineal por lineal	,750	1	,386		
N de casos válidos	7				

a. 4 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es ,43.

b. Calculado sólo para una tabla de 2x2.

**¿Considera al modelo propuesto para el proceso de desarrollo de software que está absorbiendo todas las funciones prescindibles para la actividad que realiza en la Unidad de Sistemas de la UTMACH? \* ¿Cree Usted que las actividades que se realizan actualmente en la Unidad de Sistemas mejoraran al aplicar el modelo de calidad propuesto?**

#### Pruebas de chi-cuadrado

	Valor	Gl	Sig. asintótica (bilateral)
Chi-cuadrado de Pearson	1,556 <sup>a</sup>	2	,459
Razón de verosimilitudes	1,923	2	,382
Asociación lineal por lineal	1,042	1	,307
N de casos válidos	7		

a. 6 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es ,14.

**Estaría de acuerdo en implementar el modelo de calidad a la Unidad de Sistemas \* ¿Cree Usted que el modelo propuesto tiene factores que comprenden la calidad de software?**

**Pruebas de chi-cuadrado**

	Valor	gl	Sig. asintótica (bilateral)	Sig. exacta (bilateral)	Sig. exacta (unilateral)
Chi-cuadrado de Pearson	,875 <sup>a</sup>	1	,350		
Corrección por continuidad <sup>b</sup>	,000	1	1,000		
Razón de verosimilitudes	1,243	1	,265		
Estadístico exacto de Fisher				1,000	,571
Asociación lineal por lineal	,750	1	,386		
N de casos válidos	7				

a. 4 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es ,43.

b. Calculado sólo para una tabla de 2x2.

**Estaría de acuerdo en implementar el modelo de calidad a la Unidad de Sistemas \* ¿Considera usted que aplicando el modelo de calidad propuesto se organice de mejor manera el proceso de desarrollo de software?**

**Pruebas de chi-cuadrado**

	Valor	gl	Sig. asintótica (bilateral)	Sig. exacta (bilateral)	Sig. exacta (unilateral)
Chi-cuadrado de Pearson	,194 <sup>a</sup>	1	,659		
Corrección por continuidad <sup>b</sup>	,000	1	1,000		
Razón de verosimilitudes	,196	1	,658		
Estadístico exacto de Fisher				1,000	,629
Asociación lineal por lineal	,167	1	,683		
N de casos válidos	7				

a. 4 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es 1,29.

b. Calculado sólo para una tabla de 2x2.



**Estaría de acuerdo en implementar el modelo de calidad a la Unidad de Sistemas \* ¿Cree usted que se alcanzaría a definir claramente las actividades a desarrollar para cada proyecto planteado aplicando el modelo de calidad?**

**Pruebas de chi-cuadrado**

	Valor	gl	Sig. asintótica (bilateral)	Sig. exacta (bilateral)	Sig. exacta (unilateral)
Chi-cuadrado de Pearson	1,215 <sup>a</sup>	1	,270		
Corrección por continuidad <sup>b</sup>	,109	1	,741		
Razón de verosimilitudes	1,243	1	,265		
Estadístico exacto de Fisher				,486	,371
Asociación lineal por lineal	1,042	1	,307		
N de casos válidos	7				

a. 4 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es 1,29.

b. Calculado sólo para una tabla de 2x2.

**Estaría de acuerdo en implementar el modelo de calidad a la Unidad de Sistemas \* ¿Considera al modelo propuesto para el proceso de desarrollo de software que está absorbiendo todas las funciones prescindibles para la actividad que realiza en la Unidad de Sistemas de la UTMACH?**

### Pruebas de chi-cuadrado

	Valor	gl	Sig. asintótica (bilateral)	Sig. exacta (bilateral)	Sig. exacta (unilateral)
Chi-cuadrado de Pearson	,194 <sup>a</sup>	1	,659		
Corrección por continuidad <sup>b</sup>	,000	1	1,000		
Razón de verosimilitudes	,196	1	,658		
Estadístico exacto de Fisher				1,000	,629
Asociación lineal por lineal	,167	1	,683		
N de casos válidos	7				

a. 4 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es 1,29.

b. Calculado sólo para una tabla de 2x2.

**Estaría de acuerdo en implementar el modelo de calidad a la Unidad de Sistemas \* ¿Cree Usted que las actividades que se realizan actualmente en la Unidad de Sistemas mejoraran al aplicar el modelo de calidad propuesto?**

#### **Pruebas de chi-cuadrado**

	Valor	Gl	Sig. asintótica (bilateral)
Chi-cuadrado de Pearson	4,278 <sup>a</sup>	2	,118
Razón de verosimilitudes	5,742	2	,057
Asociación lineal por lineal	3,521	1	,061
N de casos válidos	7		

a. 6 casillas (100,0%) tienen una frecuencia esperada inferior a 5. La frecuencia mínima esperada es ,43.

#### **5.3.4 Comparación entre el valor esperado y el valor crítico**

En este análisis de la correlación se encontró que la relación entre ambas variables es significativa ( $p < 0.05$ ,  $p < 0.01$ ) y, por lo tanto, existe correlación alguna. Estos resultados indican entonces que, en el caso de la muestra estudiada, si existe una relación entre modelo de calidad con el proceso de desarrollo de software, es decir, el modelo de calidad se relaciona directamente con el proceso de desarrollo de software. La relación entre las dos variables es significativa y, por lo tanto, existe correlación entre ellas.

#### **5.4 Conclusión de los resultados**

Una vez obtenido los resultados por medio de la prueba de Chi cuadrado se concluye que para que exista el resultado esperado en el proceso de desarrollo de software de la Unidad de Sistemas depende directamente de la aplicación correcta de un modelo de calidad que permita cumplir las tareas planteadas.

Según la encuesta aplicada dan como resultado que el modelo de calidad propuesto es idóneo para aplicar en la Unidad de Sistemas y de esta manera lograr corroborar la hipótesis planteada al inicio del estudio donde indica que al implementar un modelo de calidad para el sistema desarrollado en la Unidad de Sistemas de la Universidad Técnica de Machala permitirá optimizar la gestión del proceso de desarrollo de software que se aplica en esa dependencia.

La unidad de sistemas se está capacitando para tomar en cuenta los criterios de calidad y de esta forma determinar que la aplicación precisa del modelo permitirá enfocarse a las tareas correctas y dar un enfoque más organizado al proceso de desarrollo de software.

## CAPÍTULO VI

### 6. CONCLUSIONES Y RECOMENDACIONES

#### 6.1 Conclusiones

- La Unidad de Sistemas no se perfila con ninguna metodología ni métrica de desarrollo de sistema han tratado de acoplarse tomando lo que consideran más importante de la metodología tradicional y tomando el ciclo de vida tradicional para el desarrollo de sistema y creen que esta circunstancia provoca que muchas veces no se pueda cumplir con la planificación estipulada ocasionando desconformidad en el sistema informático.
- El avance evolutivo de las normas y estándares de calidad para el proceso de desarrollo de software ha contribuido de alguna manera para mejorar el proceso de desarrollo.
- Las buenas prácticas aplicadas en el proceso de desarrollo de software han permitido acoplar en las diferentes fases involucradas en el desarrollo.
- La propuesta del modelo de calidad para el proceso de desarrollo de software en la Unidad de Sistemas de la Universidad Técnica de Machala se basó en el estudio de los diferentes modelos y estándares de calidad específicamente la Norma Internacional IEC / 12207 que permitieron acoplarse al ciclo de vida del sistema considerando que las actividades propuestas en el modelo se adaptan muy fácilmente a la situación actual de la Unidad de Sistemas.
- El modelo de calidad propuesto según los resultados obtenidos de la aplicación a una parte del proceso de desarrollo de software es aplicable ya que está basado en normativas de calidad y se mejorará el proceso de desarrollo garantizando la calidad.

## 6.2 Recomendaciones

Se recomienda a la Unidad de Sistemas de la Universidad Técnica de Machala lo siguiente:

- Documentar la gestión de requisitos de sistema por más simple que se considere el requisito.
- Asignar un calendario de trabajo por cada interacción que se desarrolle en el departamento.
- Asignar roles a cada integrante del equipo de trabajo
- El jefe de sistemas deberá llevar un control específico de todas las reuniones de trabajo con las autoridades de la Universidad en cuanto a requerimientos.
- Se recomienda la aplicación del modelo propuesto ya que se adapta al ciclo de vida del sistema y de esta forma complementar los procesos que se van aplicar en la Unidad de Sistemas.
- Con los resultados obtenidos se puede observar que el modelo de calidad propuesto se adapta de manera adecuada y se propone aplicar el modelo en toda la estructura planteada para obtener buenos resultados.
- Con la finalidad de controlar de una mejor manera la aplicación del modelo se sugiere llevar un registro de cada actividad que contiene el modelo de calidad propuesto.

## REFERENCIAS BIBLIOGRÁFICAS

- [1]. Bedini González, A. (s.f.). *Gestión de Proyectos de Sistema*. Universidad de Buenos Aires.
- [2]. Definiciones de software. (2008). *Definiciones*. Recuperado el 26 de Junio de 2014, de <http://definicion.de/hardware/>
- [3]. Escobar, Y., Fuentes, R., Martínez, Y., & Yanes, J. (2006). *Desarrollo del Sistema*. Bolívar.
- [4]. García, L. (2012). *Calidad de Sistema*. Instituto Tecnológico Superior Lerdo. AllRightsReserved.
- [5]. James. T, P. (2001). *La Gestión de la Calidad Total* (Primera ed., Vol. I). (A. Otero, Ed.) Madrid, España: Prentice Hall Iberia.
- [6]. Pérez, J., & Wilson, M. (2005). *Modelo de Calidad de Sistema*. Teg. Análisis y Desarrollo de Sistemas.
- [7]. Piattini Velthuis, M., García Rubio, F., & Muñoz-Reja, I. (2010). *Calidad de Sistemas Informáticos* (Primera ed.). México, Distrito Federal, Madrid, España: RA-MA Editorial.
- [8]. Pressman, R. (2006). *Un Enfoque Práctico*. Ingeniería de Software. McGraw-Hill, México, Distrito federal.
- [9]. Pt, K. (29 de Mayo de 2011). *Modelos de Desarrollo de Software*. Recuperado el 29 de Junio de 2014, de <http://es.slideshare.net/kellypt1/modelos-de-desarrollo-de-sistema>
- [10]. Quispe, R. (2008). *Que es la Calidad de Software*. Ingeniería en Software. Ra-Ma.
- [11]. San Gabino, N., Hernández, C., & Darcy, N. (2007). *Maestría en Informática Aplicada*. (U. d. Cienfuegos", Ed.) Recuperado el 24 de Junio de 2014, de Monografías.com S.A: <http://www.monografias.com/trabajos59/calidad-sistema/calidad-sistema2.shtml>

- [12]. Sicilia, M. A. (24 de Noviembre de 2008). *Openstax CNX*. (V. De la Morena, Ed.) Recuperado el 25 de Junio de 2014, de <http://cnx.org/content/m17405/latest/>
- [13]. Vanzetti, J. J. (2006). *Un modelo del proceso de desarrollo de sistema*. Tesis de Ingeniería, Universidad Nacional de la Plata Argentina, Informática, La Plata.
- [14]. VALENCIA, María Eugenia, *La Calidad del software y la Metodología Orientada a Objetos*, pág. 21, 2003. Valencia.
- [15]. Capability Maturity Model Integration (CMMI), Version 1.1” Continuous Representation (CMU/SEI-2002-TR-011). Pittsburgh, PA: Software Engineering
- [16]. Cuatrecasas, Luis, “Gestión Integral de la Calidad”; Gestión 2000, Barcelona, 2001, 2d ed., 356 p., ISBN 84-8088-609-9
- [17]. Felhmann, Thomas M, “Six Sigma for Software”, Zurich, Switzerland, 2003.
- [18]. Florac, W; Park, R; Carleton, A; Practical Software Measurement (PSM): Measuring for Process Management Improvement” (CMU/SEI-97-HB-003). Pittsburgh, PA:
- [19]. Humphrey, Watts, ”The Personal Software Process (PSP)” (CMU/SEI-2000-TR-022).
- [20]. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000.
- [21]. Humphrey, Watts., ”The Team Software Process (TSP)” (CMU/SEI-2000-TR-023).
- [22]. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000.
- [23]. ISO/IEC 12207:1995, “Information Technology – Software Life Cycle Processes”, 1995.
- [24]. ISO/IEC 12207:1995 /AMD 1:2002, “Information Technology – Software Life Cycle Processes”, 2002.



- [25]. ISO/IEC 12207:1995 /AMD 2:2004, “Information Technology – Software Life Cycle Processes”, 2004.
- [26]. ISO/IEC TR 15504-1:1998, “Information technology - Software process assessment --Part 1: Concepts and guide introductory guide”, 1998
- [27]. Cockburn, A. (2010). *Desarrollo de Software Ágil*. EEUU: Addison-Wesley Professional.
- [28]. Española, R. A. (2001). *Definiciones de Software*.
- [29]. Fernanda, S. (25 de Junio de 2006). Estudio Comparativo de los modelos y estandares de calidad de software. Buenos Aires, Argentina.
- [30]. Freedman, A. (1984). *Catalogo*.
- [31]. Universidad Tecnológica Buenos Aires, Maestría en calidad de Sistema; (Recuperado el 29 de Junio de 2014); url:<http://posgrado.frba.utn.edu.ar/investigacion/tesis/MIC-2006-Scalone.pdf> 293 – 296: 2006
- [32]. Instituto Politécnico Nacional ; Calidad del Software; (Recuperado el 03 de Julio de 2014) <http://itzamna.bnct.ipn.mx/dspace/bitstream/123456789/5874/1/C2.305.pdf> 90-99
- [33]. Diseño metodológico ; (Recuperado el 03 de Julio de 2014); [http://www.univo.edu.sv:8081/tesis/019939/019939\\_Cap3.pdf](http://www.univo.edu.sv:8081/tesis/019939/019939_Cap3.pdf)
- [34]. Pablo R. Fillotrani; Calidad en el desarrollo de software; (Recuperado el 03 de Julio de 2014) <http://www.cs.uns.edu.ar/~prf/teaching/SQ07/clase6.pdf>
- [35]. ISO/IEC 90003:2004, “Software e Ingeniería de Sistemas – Guía para la aplicación de la Norma ISO 9001:2000 para el software”, 2004

# ANEXOS

**UNIVERSIDAD DE LA FUERZAS ARMADAS – ESPE**  
**MAESTRÍA EN INGENIERÍA DEL SOFTWARE**

**CERTIFICACIÓN**

Se certifica que el presente trabajo fue realizado por la Señora Miriam Rocio Farez Arias bajo mi supervisión.

---

**Ing. Lucas Garcés G. MSc.**  
**DIRECTOR DE TESIS**

---

**Ing. Javier Montaluisa MSc.**  
**OPONENTE**

---

**Ing. Lucas Garcés G. MSc.**  
**COORDINADOR DE LA MAESTRÍA EN INGENIERÍA DEL  
SOFTWARE**

---

**Dr. Rodrigo Vaca.**  
**SECRETARIO ACADÉMICO.**