

**UNIVERSIDAD CENTROCCIDENTAL**

**“LISANDRO ALVARADO”**

**PROPUESTA DE UN MODELO SISTÉMICO DE CALIDAD PARA EL  
PROCESO DE PRUEBAS DE SOFTWARE (MOSCP)**

**ISABEL CRISTINA CORONEL MENDEZ**

**Barquisimeto, 2012**

**UNIVERSIDAD CENTROCCIDENTAL “LISANDRO ALVARADO”**

**DECANATO DE CIENCIAS Y TECNOLOGÍA**

**MAESTRIA EN CIENCIAS DE LA COMPUTACIÓN**

**MENCIÓN INGENIERÍA DEL SOFTWARE**

**PROPUESTA DE UN MODELO SISTÉMICO DE CALIDAD PARA EL  
PROCESO DE PRUEBAS DE SOFTWARE (MOSCP)**

Trabajo presentado para optar al grado de

Magister Scientiarum

Por: Ing. ISABEL CRISTINA CORONEL MENDEZ

**Barquisimeto, 2012**

## **DEDICATORIA**

A Jehová Dios, por brindarme su bendición en cada instante de mi vida.

A mi Mami Dilcia Méndez, por ser mi Ángel terrenal y apoyarme incondicionalmente.

A mi Esposo Yanys León, por ser mi cómplice incondicional y comprender mis ausencias.

A mi Padre Ramón Felipe Méndez (+), porque me enseñaste a luchar por lo que se quiere.

A toda mi Familia, porque siempre confían en mí.

## AGRADECIMIENTO

A Jehová Dios, por regalarme una vida maravillosa y permitirme culminar esta meta.

A mi Mami Dilcia Méndez, por ser un gran ejemplo para mí, eres el mejor regalo que Jehová Dios me ha dado. Gracias por compartir toda tu sabiduría conmigo.

A mi Esposo Yanys León, por su amor sin límites, confianza, amistad, y por no haber permitido que me rindiera en el camino. Gracias por enseñarme que no hay límites cuando se quiere lograr un sueño.

A mi Papá Ramón Méndez (Paito), por ser para mí un gran ejemplo de responsabilidad, dedicación, constancia y sobre todo por enseñarme lo que significa ser un padre ejemplar, siempre estarás en mi corazón.

A mi hermano, Robert Coronel, por brindarme su apoyo incondicional.

A mi familia, por el cariño y confianza brindada.

A mi tutora, la Prof .Ana Mercedes Díaz, por su confianza, dedicación, amistad, apoyo y sus valiosos aportes académicos.

A la Prof. María Angélica Pérez de Ovalles, por su profesionalismo e invaluable aporte académico.

A mis compañeros de Maestría, Jennifer Rivero y Arnaldo Suárez, gracias por la solidaridad, compañerismo y amistad brindada.

A mi Amiga Lismar Arroyo, por su amistad incondicional.

A todos que Jehová Dios los Bendiga. Los Quiero Mucho.

## ÍNDICE GENERAL

	<b>PAG.</b>
DEDICATORIA	iii
AGRADECIMIENTO	iv
ÍNDICE DE FIGURAS	x
ÍNDICE DE TABLAS	xv
RESUMEN	xviii
INTRODUCCIÓN	1
<b>CAPÍTULO</b>	
<b>I EL PROBLEMA</b>	<b>4</b>
1.1 Planteamiento del Problema	4
1.2 Objetivos	10
1.2.1 Objetivo General	10
1.2.2 Objetivos Específicos	10
1.3 Justificación e Importancia	10
1.4 Alcance y Limitaciones	12
<b>II MARCO TEÓRICO</b>	<b>14</b>
2.1 Antecedentes de la Investigación	14
2.1.1 Integración de Modelos de Capacidad de Madurez (CMMI, por sus siglas en inglés).	14
2.1.2 Modelo de Madurez de Prueba (TMM, por sus siglas en inglés)	18
2.1.3 Modelo de Mejora del Proceso de Pruebas (TPI, por sus siglas en inglés)	20
2.1.4 Modelo de Madurez Integrado de Prueba(TMMI, por sus siglas en inglés)	21

2.1.5 Área de Proceso de Pruebas Integrado con CMMI (TESPAI, por sus siglas en inglés)	26
2.1.6 ISO/IEC 29119 Software Testing	27
2.1.7 Modelo Sistémico de Calidad del Software (MOSCA)	33
2.1.7.1 Características del proceso que influyen en la Funcionalidad del producto de Software	39
2.1.7.2 Características del proceso que influyen en la Fiabilidad del producto de Software	39
2.1.7.3 Características del proceso que influyen en la Eficiencia del producto de Software	40
2.1.7.4 Características del proceso que influyen en la Usabilidad del producto de Software	40
2.1.7.5 Características del proceso que influyen en la Mantenibilidad del producto de Software	40
2.1.7.6 Características del proceso que influyen en la Portabilidad del producto de Software.	41
2.1.7.2 Algoritmo de Aplicación del modelo MOSCA	42
2.2 Bases Teóricas	47
2.2.1 Proceso de software	47
2.2.2 Modelo de Procesos	49
2.2.3 Pruebas de Software	50
2.2.3.1 Objetivos de las pruebas de Software	53
2.2.3.2 Características de las pruebas	54

2.2.3.3	Artefactos de las Pruebas	56
2.2.3.4	El Proceso de Pruebas	59
2.2.4	Calidad del Software	62
2.3.	Bases Legales	65
III	MARCO METODOLÓGICO	67
3.1	Tipo de Investigación	67
3.2	Fases del Estudio	69
3.2.1	Fase Diagnóstica	69
3.2.2	Población y Muestra	69
3.2.3	Técnicas de Recolección de Datos	70
3.2.4	Estudio de Factibilidad	71
3.2.4.1	Factibilidad Técnica	71
3.2.4.2	Factibilidad Económica	71
3.2.5	Procedimiento de la Investigación	71
3.2.5.1	Investigación Documental	72
3.2.5.2	Análisis de los Antecedentes	73
3.2.5.3	Formulación de los objetivos y el alcance de la investigación	73
3.2.5.4	Formulación de la Metodología de Investigación	73
3.2.5.5	Propuesta del Modelo MOSCP enmarcado en MOSCA	74
3.2.5.5.1	Análisis de los niveles que conformarán el Modelo	74

3.2.5.5.2 Definición del Algoritmo de Procesamiento de la Información	74
3.2.5.6 Análisis del Contexto	75
3.2.5.7 Aplicación del Método DESMET	75
3.2.5.8 Evaluación a través del Método arrojado por DESMET	77
3.2.5.9 Análisis de los Resultados	78
3.2.5.10 Definición del alcance de la siguiente iteración	78
3.2.5.11 Conclusiones y Recomendaciones	78
IV PROPUESTA DEL ESTUDIO	79
4.1 Modelo Conceptual	79
4.2 Análisis del Contexto	115
4.3 Aplicación del Método DESMET	118
4.4 Evaluación a través del método arrojado por DESMET	121
V CONCLUSIONES Y RECOMENDACIONES	161
GLOSARIO DE TÉRMINOS	164
REFERENCIAS BIBLIOGRÁFICAS	167
ANEXOS	174
Anexo I. Resultados de la Evaluación de los criterios generales del Modelo.	175
Anexo II. Resultados de la Evaluación de las Métricas del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP) y del modelo en el Proceso de Pruebas de Software del desarrollo del Sistema de Inscripción para el Control de Actividades Académicas (SICAA) del Instituto Universitario de Tecnología del Estado Yaracuy	180

(IUTY).

Anexo III. Resultados de la Evaluación del Modelo en el Proceso de 212  
Pruebas de Software del desarrollo del Sistema de Inscripción para el  
Control de Actividades Académicas (SICAA)

## ÍNDICE DE FIGURAS

	<b>Página</b>
Figura 01: Estructura de TMM	18
Figura 02: Estructura de TPI	20
Figura 03: Estructura de TMMI	22
Figura 04: Niveles y áreas Claves de TMMI	23
Figura 05: Objetivos y Prácticas Específicas de TestPAI	26
Figura 06: Estructura ISO/IEC 29119(Traducida)	28
Figura 07: Parte 2-Procesos (Versión Revisada y Traducida)	29
Figura 08: Estructura de MOSCA	38
Figura 09: Algoritmo de aplicación de MOSCA.	43
Figura 10: Modelo Conceptual asociado a los Casos de Prueba	57
Figura 11: Principales factores de calidad del producto software	63
Figura 12: Matriz de Calidad Global Sistémica	64
Figura 13: Framework Metodológico de la Investigación	72
Figura 14: Modelo Conceptual Pruebas de Software de la presente investigación.	83
Figura 15: Diagrama del modelo Sistémico de calidad MOSCA- Consideraciones de MOSCA	87
Figura 16: Propuesta del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).	114
Figura 17: Actividades en el proceso de planificación del Análisis de características.	122

Figura 18: Actividades en el proceso de un estudio de caso.	124
Figura 19: Resultados Categoría Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).	129
Figura 20: Resultados Característica Determinación de Requerimientos(CUS1) de la Categoría Cliente Proveedor (CUS) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP)	130
Figura 21: Resultados de las Características de la Categoría Ingeniería (ENG) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP)	130
Figura 22: Resultados de las Características de la Categoría Soporte (SUP) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).	131
Figura 23: Resultados de las Características de la Categoría Gestión (MAN) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP)	131
Figura 24: Resultados de las Características de la Categoría Organizacional (ORG) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).	132
Figura 25: Resultados de las Características de la Categoría Entrenamiento (TRA) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).	132
Figura 26: Resultados de las Sub-Características de la Característica Cliente-Proveedor(CUS) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).	133
Figura 27: Resultados de las Sub-Características de la Característica Planificación y Control (ENG1) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP)	134
Figura 28: Resultados de las Sub-Características de la Característica Análisis y Diseño (ENG2) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).	134
Figura 29: Resultados de las Sub-Características de la Característica Desarrollo (ENG3) del Modelo Sistémico de Calidad para el Proceso de	135

Pruebas de Software (MOSCP).

Figura 30: Resultados de las Sub-Características de la Característica 135  
Criterios de Terminación (ENG4) del Modelo Sistémico de Calidad para el  
Proceso de Pruebas de Software (MOSCP)

Figura 31: Resultados de las Sub-Características de la Característica 136  
Reportes (ENG5) del Modelo Sistémico de Calidad para el Proceso de  
Pruebas de Software (MOSCP).

Figura 32: Resultados de las Sub-Características de la Característica 136  
Actividades de Cierre (ENG6) del Modelo Sistémico de Calidad para el  
Proceso de Pruebas de Software (MOSCP).

Figura 33: Resultados de las Sub-Características de la Característica 137  
Documentación(SUP1) del Modelo Sistémico de Calidad para el Proceso de  
Pruebas de Software (MOSCP).

Figura 34: Resultados de las Sub-Características de la Característica 137  
Aseguramiento de la Calidad (SUP2) del Modelo Sistémico de Calidad para  
el Proceso de Pruebas de Software (MOSCP).

Figura 35: Resultados de las Sub-Características de la Característica 138  
Revisión Conjunta (SUP3) del Modelo Sistémico de Calidad para el Proceso  
de Pruebas de Software (MOSCP)

Figura 36: Resultados de las Sub-Características de la Característica 138  
Auditoría (SUP4) del Modelo Sistémico de Calidad para el Proceso de  
Pruebas de Software (MOSCP)

Figura 37: Resultados de las Sub-Características de la Característica Gestión 139  
(MAN1) del Modelo Sistémico de Calidad para el Proceso de Pruebas de  
Software (MOSCP).

Figura 38: Resultados de las Sub-Características de la Gestión de Calidad 139  
(MAN2) del Modelo Sistémico de Calidad para el Proceso de Pruebas de  
Software (MOSCP).

Figura 39: Resultados Métricas del Modelo Sistémico de Calidad para el 141  
Proceso de Pruebas de Software (MOSCP)

Figura 40: Resultados Sub-Características de la Característica Determinación 143  
de Requerimientos(CUS1) del Modelo Sistémico de Calidad para el Proceso  
de Pruebas de Software (MOSCP).

- Figura 41: Resultados Sub-Características de la Característica Planificación y Control (ENG1) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP). 144
- Figura 42: Resultados Sub-Características de la Característica Análisis y Diseño (ENG2) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP). 145
- Figura 43: Resultados Sub-Características de la Característica Desarrollo (ENG3) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP). 146
- Figura 44: Resultados Sub-Características de la Característica Criterios de Terminación (ENG4) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP). 147
- Figura 45: Resultados Sub-Características de la Característica Reportes (ENG5) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP). 148
- Figura 46: Resultados Sub-Características de la Característica Actividades de Cierre (ENG6) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP). 149
- Figura 47: Resultados Sub-Características de la Característica Documentación (SUP1) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP). 150
- Figura 48: Resultados Sub-Características de la Característica Aseguramiento de la Calidad (SUP2) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP). 151
- Figura 49: Resultados Sub-Característica de la Característica Revisión Conjunta (SUP3) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP). 151
- Figura 50: Resultados Sub-Característica de la Característica Auditoría (SUP4) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP). 152
- Figura 51: Resultados Sub-Características de la Característica Gestión (MAN1) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP). 153
- Figura 52: Resultados Sub-Característica de la Característica Gestión de Calidad (MAN2) del Modelo Sistémico de Calidad para el Proceso de 154

Pruebas de Software (MOSCP).

Figura 53: Resultados Característica Determinación de Requerimientos (ORG1) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP). 155

Figura 54: Resultados de las Características de la Categoría Ingeniería (ENG) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP). 156

Figura 55: Resultados de las Características de la Categoría Soporte (SUP) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP). 156

Figura 56: Resultados de las Características de la Categoría Gestión (MAN) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP). 157

Figura 57: Resultados Características de la Categoría Organizacional (ORG) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP). 157

Figura 58: Resultados Características de la Categoría Entrenamiento (TRA) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP). 158

Figura 59: Resultados Categorías del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP). 159

## ÍNDICE DE TABLAS

<b>TABLA</b>		<b>Página</b>
01	Uso de modelo de Mejoras	5
02	Uso de modelos de técnicas de apoyo en el proceso de desarrollo	7
03	Metas y subprácticas relacionadas a las áreas de Validación y Verificación del Modelo Integración de Modelos de Capacidad de Madurez (CMMI, por sus siglas en inglés).	16
04	Criterios definidos para realizar el análisis de los modelos de referencia	24
05	Descripción de las características de los modelos de mejora del proceso de pruebas.	25
06	Definición de las características para realizar el análisis de los modelos en estudio.	31
07	Tabla comparativa de los modelos de mejora de procesos de pruebas.	32
08	Categorías del submodelo del Producto.	35
09	Categorías del submodelo del Proceso.	36
10	Distribución de las Características y métricas para medir la calidad sistémica del producto de software.	37
11	Distribución de las Características y métricas para medir la calidad sistémica del proceso de desarrollo.	38
12	Nivel de calidad del producto con respecto a las categorías satisfechas para el producto.	44
13	Nivel de Calidad Sistémica Global a partir del nivel de Calidad del Producto.	45
14	Definiciones de Proceso de Pruebas de Software.	61
15	Fases del Método Investigación-Acción.	68
16	Nuevas sub_ características, sub-sub-características y métricas para la categoría Cliente-Proveedor (CUS).	88

17	Nueva característica Planificación y Control (ENG1), sub_ características y métricas para la categoría Ingeniería (ENG).	91
18	Nueva característica Análisis y Diseño (ENG2), sub_ características y métricas para la categoría Ingeniería (ENG).	93
19	Nueva característica Desarrollo (ENG3), sub_ características y métricas para la categoría Ingeniería (ENG).	94
20	Nueva característica Criterios de Terminación (ENG4), sub_ características y métricas para la categoría Ingeniería (ENG).	96
21	Nueva característica Reportes (ENG 5), sub_ características y métricas para la categoría Ingeniería (ENG).	97
22	Nueva característica Actividades de Cierre (ENG 6), sub_ características y métricas para la categoría Ingeniería (ENG).	97
23	Nuevas sub_ características y métricas para la categoría Soporte (SUP) y la característica Documentación (SUP 1).	100
24	Nuevas sub_ características y métricas para la categoría Soporte (SUP) y la característica Aseguramiento de la Calidad (SUP 2).	102
25	Nuevas sub_ características y métricas para la categoría Soporte (SUP) y la característica Revisión Conjunta (SUP 3).	103
26	Nuevas sub_ características y métricas para la categoría Soporte (SUP) y la característica Auditoría (SUP 4)	104
27	Nuevas sub_ características y métricas para la categoría Gestión (MAN) y la característica Gestión (MAN 1)	106
28	Nuevas sub_ características y métricas para la categoría Gestión (MAN) y la característica Gestión de Calidad (MAN2)	107
29	Nuevas métricas para la categoría Organizacional (ORG) y la característica Lineamientos Organizacionales (ORG 1)	108
30	Nuevas métricas para la categoría Organizacional (ORG) y la Característica Evaluación del Proceso (ORG2).	109
31	Nuevas métricas para la categoría Organizacional (ORG) y la Característica Reuso (ORG3).	110

32	Nueva característica Adiestramiento (TRA) y métricas para el Modelo Sistémico de Calidad para el proceso de pruebas de Software (MOSCP).	112
33	Condiciones favorables presentes y no favorables presentes en cada uno de los métodos DESMET	121
34	Definición de la escala utilizada para evaluar los criterios generales y específicos.	126
35	Criterios generales a evaluar en el Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).	126
36	Criterios específicos a evaluar en el Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).	127
37	Resultados evaluación Categorías del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).	160

**UNIVERSIDAD CENTROCCIDENTAL “LISANDRO ALVARADO”**

**DECANATO DE CIENCIAS Y TECNOLOGÍA**

**PROPUESTA DE UN MODELO SISTÉMICO DE CALIDAD PARA EL  
PROCESO DE PRUEBAS DE SOFTWARE (MOSCP)**

Autor(a): Ing. Isabel Cristina Coronel Méndez

Tutor(a): Msc. Ing. Ana M. Díaz

**RESUMEN**

El presente trabajo de investigación está orientado a estudiar un área vital de la Ingeniería de software que contribuye al aseguramiento de la calidad del software, a saber, las pruebas. Actualmente, las empresas están demandando modelos que les permitan mejorar el proceso de pruebas que están implementando, es por esta razón, que existen una serie de modelos de referencia para determinar las debilidades y fortalezas del proceso de pruebas, sin embargo, no lo abordan desde el punto de vista sistémico. Es por ello que el objetivo general de esta investigación es: Proponer un Modelo Sistémico de Calidad para el proceso de pruebas de software, que permita integrar la calidad del producto y la calidad del proceso de desarrollo, de esta manera contribuir a obtener productos libres de errores y fallas, aumentando la productividad en las organizaciones tomando en cuenta las relaciones existentes entre los diversos elementos que intervienen en el proceso de pruebas con una visión holística o en otras palabras, sistémica. El estudio está concebido bajo la modalidad de Investigación cualitativa, apoyado en la metodología Investigación-Acción, la cual permite organizar la investigación en el análisis, diseño y elaboración de la propuesta de un modelo de calidad y así aportar un marco de referencia en el desarrollo del proceso de pruebas de software bajo un enfoque sistémico.

**Palabras Clave:** Prueba de Software, Proceso de Prueba, Modelos de Calidad del Proceso de Prueba, Calidad Sistémica, Modelo Sistémico de Calidad.

## INTRODUCCIÓN

Hoy en día, debido al aumento del tamaño y la complejidad del software, el proceso de pruebas se ha convertido en una tarea vital en el desarrollo de cualquier sistema informático. Gutiérrez, et.al (2005). Al respecto Pérez et.al (2008), señala lo siguiente, las pruebas de software, es un proceso que busca proporcionar confianza en el producto desarrollado tanto desde el punto de vista de los desarrolladores del sistema como del punto de vista de los clientes, ya que éste debe satisfacer tanto los requisitos funcionales como los no funcionales para su uso operacional o puesta en producción, así, la confiabilidad de las pruebas impacta la confiabilidad del producto software desarrollado. Por lo anteriormente planteado, se debe garantizar la calidad esperada por los clientes según los criterios de aceptación acordados.

En este sentido, Sanz y otros (2008), afirman que cada vez hay más pequeñas y medianas empresas que están adquiriendo conciencia sobre la importancia del proceso de prueba y la mejora del mismo. Muchas de ellas están demandando un marco de trabajo que les proporcione las herramientas necesarias para realizar una mejora del proceso de pruebas de un modo poco costoso y sencillo.

Significa entonces, que debido a la necesidad de la calidad del software generado, se debe tomar en cuenta la calidad del proceso de pruebas que se sigue, por esto actualmente existen marcos de referencia o modelos que buscan determinar la madurez del proceso de pruebas implementado en el desarrollo de software, entre los cuales destacan: Modelo de Madurez de Prueba (TMM, por sus siglas en inglés,1996), Modelo de Mejora del Proceso de Pruebas (TPI, por sus siglas en inglés,1999), Modelo de Madurez Integrado de Prueba (TMMI, por sus siglas en inglés,2008), Área de Proceso de Pruebas Integrado con CMMI (TESPAI, por sus siglas en inglés,2008), y el estándar en desarrollo ISO/IEC 29119 Software Testing (2007, aún en desarrollo), los cuales, incluyen todas las actividades relacionadas al proceso de pruebas.

Sin embargo, dado la complejidad del proceso de desarrollo de software, es importante considerar una visión sistémica, la cual manifieste la organización como un todo que busca la obtención de los objetivos organizacionales, a través de productos de software de alta calidad, lo cual proporciona sin duda alguna la satisfacción del cliente, es importante trasladar la visión sistémica al proceso de pruebas y así garantizar que se está desarrollando buscando la calidad en cada una de las actividades involucradas y ejecutadas en el mismo, y de esta manera garantizar la calidad sistémica del producto desarrollado.

En el presente trabajo, se propone un modelo sistémico de calidad para el proceso de pruebas de software, el cual tomando en cuenta cada una de las actividades implicadas en el mismo, desde una perspectiva sistémica busca obtener un software de calidad, mediante un proceso definido y organizado.

Este proyecto de investigación se organiza en capítulos. En el primero, El Problema, se exponen los diferentes elementos que conforman la situación planteada con respecto a la importancia de la calidad del software y el papel del proceso de pruebas como mecanismo para el aseguramiento de la misma. Igualmente se presentan los objetivos del mismo, la justificación e importancia, alcance y limitaciones, los cuales representan las bases que conducen la investigación.

En el Capítulo II (Marco Teórico), se presentan los antecedentes relacionados, así como las bases teóricas que apoyan el conocimiento del tema en estudio; se describen los modelos de referencia relacionados a la calidad del proceso de pruebas de software, conceptos asociados a las pruebas de software, calidad del software, teorías relacionadas a la calidad sistémica, así como los modelos de calidad sistémicos, entre los que destaca: Modelo Sistémico de Calidad del software (MOSCA)

El Capítulo III, está relacionado con el Marco Metodológico, en él se describe la naturaleza del tema en estudio, el procedimiento de la investigación, así como el estudio de factibilidad.

El Capítulo IV Propuesta del Estudio, dado el estudio del capítulo anterior, se presenta el desarrollo de las fases del framework metodológico, que fueron implementadas para formular la propuesta del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

Finalmente, el Capítulo V Conclusiones y Recomendaciones, ofrece un conjunto de conclusiones obtenidas del estudio y algunas recomendaciones enfocadas al Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

## **CAPÍTULO I**

### **EL PROBLEMA**

#### **1.1 Planteamiento del Problema**

La presencia imperante del software en la vida cotidiana, trae como consecuencia la demanda del desarrollo de numerosas aplicaciones de software, debido a las necesidades de la sociedad en general el diseño e implantación del software es un proceso cada vez más complejo, en el cual cualquier fallo puede traer gravísimas consecuencias. No obstante, Prins et.al (2007), afirman que muchas organizaciones siguen subestimando los riesgos y posibles consecuencias del impacto económico de los defectos del software. Señalan que según cálculos del Instituto Nacional de Estándares y Tecnologías (NITS, por sus siglas en inglés) de los Estados Unidos, en el año 2002 las pérdidas por errores de software en el mercado norteamericano rondaron los 67,5 millones de euros, esto sin incluir daños adicionales ocasionados por: Pérdida de ventas, imagen deteriorada, pérdida de productividad y reclamos por daños y perjuicios, cada uno de estos factores puede llegar a poner en riesgo la continuidad de la organización.

En este sentido, Casallas et al. (2007), asegura que el mejoramiento continuo de los procesos de desarrollo de software es cada día más importante para las empresas que pretenden competir abiertamente en los mercados internacionales especialmente aquellas que poseen como meta la calidad del software. En relación con éste último, Rivero et.al (2009) afirman que la madurez

y capacidad de una empresa de software para desarrollar productos de alta calidad se mide a través de modelos de mejoras, tales como los modelos Integración de Modelos de Capacidad de Madurez (CMMI, por sus siglas en inglés), SPICE e ISO 9000. El modelo CMMI-SW del Instituto de Ingeniería de Software es el modelo empleado por la industria mundial del software, como un estándar de facto, en la evaluación de la capacidad y madurez que tienen estas empresas para desarrollar software de alta calidad.

En este orden de ideas, como conclusión del análisis del desarrollo de software en empresas venezolanas en el área de modelos de mejoras de proceso, Rivero et.al (2009), se obtuvo que sólo el 45% del total de las empresas consultadas usan el modelo ISO frecuentemente u ocasionalmente, mientras que el 36% usa el modelo CMMI. Los otros modelos ITIL Y PMP son poco conocidos o utilizados. Los resultados del análisis se pueden apreciar en la Tabla 01.

Uso de modelos de calidad y buenas prácticas	Opciones de Respuesta	Micro-empresas	Pequeñas	Medianas	Grandes	Total
ISO (Normas de calidad de la International Organization for Standardization)	Uso Frecuente	4,29%	7,14%	2,86%	10,00%	24,29%
	Uso ocasional	2,86%	1,43%	12,86%	4,29%	21,43%
	Conoce y no lo usa	12,86%	17,14%	7,14%	8,57%	45,71%
	No conoce	1,43%	2,86%	1,43%	2,86%	8,57%
CMMI (Capability Maturity Model Integration del Software Engineering Institute de Carnegie Mellon University)	Uso Frecuente	1,43%	7,14%	4,29%	7,14%	20,00%
	Uso ocasional	1,43%	2,86%	5,71%	5,71%	15,71%
	Conoce y no lo usa	11,43%	10,00%	7,14%	10,00%	38,57%
	No conoce	7,14%	8,57%	7,14%	2,86%	25,71%
Uso de modelos de calidad y buenas prácticas	Opciones de Respuesta	Micro-empresas	Pequeñas	Medianas	Grandes	Total
ITIL (Information Technology Infrastructure Library)	Uso Frecuente	0,00%	0,00%	0,00%	1,43%	1,43%
	Uso ocasional	0,00%	4,29%	1,43%	8,57%	14,29%
	Conoce y no lo usa	10,00%	12,86%	8,57%	8,57%	40,00%
	No conoce	11,43%	11,43%	14,29%	7,14%	44,29%
PMP (Project Management Professional basado en la PMBOK Guide)	Uso Frecuente	0,00%	2,86%	1,43%	1,43%	5,71%
	Uso ocasional	1,43%	1,43%	0,00%	7,14%	10,00%
	Conoce y no lo usa	7,14%	12,86%	11,43%	10,00%	41,43%
	No conoce	12,86%	11,43%	11,43%	7,14%	42,86%

**Tabla 01:** Uso de modelo de Mejoras

**Fuente:** 7th Latin American and Caribbean Conference for Engineering and Technology. San Cristobal, Venezuela.

**Autor:** Rivero y otros (2009)

Ahora bien, en el proceso de desarrollo del software se destaca la fase de pruebas, la cual Grimán et al. (2003), la consideran como una *disciplina*<sup>1</sup> importante en el proceso de desarrollo de los sistemas de software, esta disciplina generalmente no se implementa de forma organizada y sistemática.

A su vez, Fernández et al. (2008), afirman que las pruebas de software constituyen la técnica de verificación y validación más utilizada en los proyectos de desarrollo, ya que está presente sistemáticamente en todos ellos en un grado u otro de madurez o eficacia, el resto de técnicas de aseguramiento de la calidad del software suele aplicarse en porcentajes relativamente bajos en organizaciones y desarrollos.

Por su parte Calvo et al. (2009), considera que la prueba de software debe ser una actividad sistemática y planificada, siguiendo una serie de normas, modelos, estándares o guías que aseguren mediante evidencia el correcto desarrollo del proceso de pruebas.

De lo anteriormente expuesto, se puede concluir que la fase de pruebas es una actividad en el proceso de desarrollo de software de vital importancia para el aseguramiento de la calidad del producto desarrollado, si se realiza de manera sistemática y planificada. Sin embargo, en el estudio realizado por Rivero y otros (2009), se evidencia que el empleo de herramientas en lo que respecta a las actividades de prueba son las menos utilizadas por las empresas venezolanas. (Ver Tabla 02)

---

<sup>1</sup> *disciplina* es una colección de actividades relacionadas con un área de atención dentro de todo el proyecto.

CRITERIOS DE INTERES		Tamaño de la empresa				
Herramientas de apoyo en la ejecución del proyecto	Opciones de Respuesta	Micro-empresas	Pequeñas	Medianas	Grandes	Total
Análisis y Diseño de Software asistido por herramientas de diagramación	Uso Frecuente	9,86%	16,40%	14,08%	16,00%	56,34%
	Uso ocasional	5,63%	7,04%	4,23%	1,41%	18,31%
	Conoce y no lo usa	5,63%	7,04%	5,63%	5,63%	23,94%
	No conoce	0,00%	0,00%	0,00%	1,41%	1,41%
Debugger	Uso Frecuente	11,27%	16,90%	9,06%	12,66%	50,70%
	Uso ocasional	5,63%	8,45%	4,23%	9,86%	28,17%
	Conoce y no lo usa	4,23%	4,23%	7,04%	2,82%	18,31%
	No conoce	0,00%	0,00%	2,82%	0,00%	2,82%
H. de Control de Versiones	Uso Frecuente	14,08%	12,66%	12,66%	19,72%	59,15%
	Uso ocasional	0,00%	9,86%	5,63%	5,63%	21,13%
	Conoce y no lo usa	5,63%	5,63%	2,82%	0,00%	14,08%
	No conoce	1,41%	1,41%	2,82%	0,00%	5,63%
Testing Automático	Uso Frecuente	4,23%	2,82%	1,41%	2,82%	11,27%
	Uso ocasional	8,45%	7,04%	5,63%	8,45%	29,56%
	Conoce y no lo usa	5,63%	14,08%	14,08%	11,27%	45,07%
	No conoce	2,82%	5,63%	2,82%	2,82%	14,08%
Gestión de Requerimientos	Uso Frecuente	11,27%	9,86%	16,90%	18,31%	56,34%
	Uso ocasional	2,82%	7,04%	1,41%	2,82%	14,08%
	Conoce y no lo usa	4,23%	7,04%	4,23%	2,82%	18,31%
	No conoce	2,82%	5,63%	1,41%	1,41%	11,27%

**Tabla 02:** Uso de modelos de técnicas de apoyo en el proceso de desarrollo.

**Fuente:** 7th Latin American and Caribbean Conference for Engineering and Technology. San Cristobal, Venezuela.

**Autor:** Rivero y otros (2009).

Contrariamente a lo que se ha expresado, los resultados del World Quality Report 2010-2011, elaborado por Sogeti et. al (2011) señalan que las economías emergentes, lideradas por China, son más dinámicas a la hora de asegurar la calidad de las aplicaciones de negocio. Así, el 83% de las empresas chinas y el 56% de las brasileñas declaran incrementos notables de sus inversiones en Aseguramiento de la Calidad (QA, por sus siglas en inglés) y Testing. De nuevo aparecen en contraste la estabilización de las inversiones en estas mismas áreas por parte de Europa y Estados Unidos. Australia también aparece en el reporte como una de las zonas geográficas con mayor dinamismo en Aseguramiento de Calidad (QA, por sus siglas en inglés) y Testing. El estudio señala que el 37% de las empresas del país declaran que también incrementarán sus presupuestos en estas áreas. Estas inversiones, tal y como se detalla en el estudio, estarán focalizadas en asegurar la eficiencia y la estandarización de las aplicaciones mediante la creación de Centros de Excelencia de Testing para centralizar y consolidar las prácticas que aseguren la calidad.

Adicionalmente, cada vez hay más pequeñas y medianas empresas que están adquiriendo conciencia acerca de la importancia del proceso de pruebas y la mejora

del mismo, muchas de ellas están demandando un marco de trabajo que les proporcione las herramientas necesarias para realizar una mejora del proceso de un modo poco costoso y sencillo. Sanz et al. (2008).

En la actualidad existen numerosos modelos y estándares para la mejora del proceso de pruebas, entre los cuales destacan: Modelo de Madurez de Prueba (TMM, por sus siglas en inglés,1996), Modelo de Mejora del Proceso de Pruebas (TPI, por sus siglas en inglés,1999), Modelo de Madurez Integrado de Prueba (TMMI, por sus siglas en inglés,2008), Área de Proceso de Pruebas Integrado con CMMI (TESPAI, por sus siglas en inglés,2008), y el estándar en desarrollo ISO/IEC 29119 Software Testing(2007-aún en desarrollo), los cuales mediante una serie de actividades buscan evaluar la madurez del proceso de pruebas que se sigue en el proceso de desarrollo de software, dichos modelos, poseen como característica común la cobertura completa del proceso de pruebas, sin embargo, según lo señalado por Callaos y Callaos (1996) citado por (Méndoza et al., 2005), dichos modelos no engloban las dos tendencias actuales de modelos de calidad; estas son calidad del producto(software) y calidad del proceso, con un enfoque sistémico.

En este orden de ideas surge la interrogante, ¿Cuándo se puede decir que un proceso de aseguramiento de calidad del software se está desarrollando con una visión sistémica? Callaos y Callaos (1996) citado por (Méndoza et al., 2005), propone un concepto de calidad del software en el cual están involucrados tanto características internas como el contexto organizacional, lo que genera un enfoque sistémico del concepto de calidad del software. Además aseguran que, en todos los enfoques de calidad, incluyendo el de la calidad del software, la organización necesita ser considerada como organización orgánica más bien que mecánica, ya que ésta usa sus recursos, las ideas y los esfuerzos para transformar las entradas en mercancías y servicios de calidad. Así se puede indicar que, la calidad del producto de software está fuertemente determinada con la calidad del proceso que se lleve a cabo en el desarrollo del mismo, lo que se puede extrapolar a la calidad de las pruebas, es decir,

la calidad de las pruebas de software ejecutadas, estarán determinadas por la calidad del proceso que se esté utilizando.

Ante lo planteado surgen las siguientes interrogantes a ser resueltas durante el desarrollo de la investigación:

1. ¿Es posible diagnosticar la necesidad de un modelo sistémico de calidad para el proceso de pruebas de software?
2. ¿Existe un modelo sistémico de calidad orientado al proceso de pruebas de software?
3. ¿Se logrará proponer un modelo sistémico de calidad para el proceso de pruebas de software?
4. ¿Cómo sería el mecanismo más apropiado para evaluar el aporte de la investigación?

El presente trabajo describe la propuesta de un modelo sistémico de calidad para el proceso de pruebas de software, diseñado y parametrizado para responder las interrogantes anteriormente planteadas, dando como resultado un proceso de pruebas evaluado bajo el enfoque sistémico.

## **Objetivos**

### **1.2.1 Objetivo General**

Proponer un modelo sistémico de calidad para el proceso de pruebas de software.

### **1.2.2 Objetivos Específicos**

- (a) Diagnosticar la necesidad de un modelo sistémico de calidad para el proceso de pruebas.
- (b) Determinar la visión sistémica de un conjunto de estándares y modelos de calidad del proceso de pruebas de software con el propósito de comprobar el aporte de la investigación.
- (c) Proponer un modelo para especificar la calidad del proceso de pruebas desde el punto de vista sistémico.
- (d) Evaluar el modelo propuesto en el proceso de pruebas de software de una organización.

### **1.3 Justificación e Importancia**

Actualmente el proceso de pruebas está tomando auge, debido a que cada día las empresas desarrolladoras de software están apuntando hacia la calidad del producto desarrollado. Sin importar la metodología utilizada en el proceso de desarrollo de software entre las cuales destacan RUP, XP, UP, ICONIX, entre otras, las pruebas deben manifestar si los objetivos planteados a nivel de requisitos son satisfechos por el producto desarrollado.

La propuesta de un modelo de calidad para el proceso de pruebas considerando la organización como un ente abierto, tomando en cuenta aspectos internos y aspectos contextuales del producto y del proceso, es decir, desde una visión sistémica, en aras de garantizar la calidad del producto entregado al cliente, sería un aporte interesante para las entidades desarrolladoras de software, ya que en la actualidad existe una gran madurez del mercado con respecto a la garantía de la calidad del software siguiendo el concepto de calidad total sistémica, esto se debe a la estrecha relación que existe con el aumento de la productividad, dado que una organización no puede alcanzar niveles óptimos de calidad sin tomar en cuenta las relaciones existentes entre cada una de los elementos que la conforman, es decir, como un ente holístico.

Al respecto, Méndez (2007)<sup>a</sup>, señala lo siguiente: Al incrementar la calidad del proceso de desarrollo se mejoran los resultados obtenidos en la productividad; ya que la organización, como ente abierto, no puede alcanzar niveles óptimos de calidad en sus productos de software y en sus procesos de desarrollo sin tomar en cuenta factores internos y externos que influyen en todo el sistema.

Paralelamente García et al. (2007) citado por Sanz et al. (2008), la mejora del proceso de pruebas generará un conjunto de beneficios para la industria del software, entre los más destacables se encuentran los siguientes:

- Se produce un incremento en la satisfacción del cliente al utilizar un software con una cantidad de errores inferior.
- Se incrementa la eficiencia del proceso de desarrollo.
- Se facilita la definición y cumplimiento de los objetivos de calidad.
- Se incrementa la satisfacción de los trabajadores debido a que se proporcionan herramientas y recursos apropiados para la realización eficiente del trabajo.

En este mismo sentido, De Rojas et.al (2009) señalan los beneficios que obtuvo una PYME en la mejora del proceso de pruebas, entre los cuales destacan:

- Aumenta la seguridad y confianza de los desarrolladores, ya que saben que sus resultados están verificados y validados por el analista de pruebas.
- Mejora la confianza de los consultores y el personal de soporte, ya que saben que los procesos de prueba están permitiendo liberar software con menos errores.
- Mejorar la satisfacción del cliente, ya que se les está dando lo que necesitan y sin errores.

En tal sentido, para las organizaciones desarrolladoras de software, la presente propuesta constituirá una guía para llevar a cabo el proceso de pruebas tomando en cuenta la importancia de garantizar la calidad sistémica en cada una de las actividades ejecutadas en el mismo, lo cual contribuye al lanzamiento y entrega de productos al mercado certificados, libres de errores y fallas, logrando así obtener la permanencia en el tan competitivo mundo del software.

#### **1.4 Alcance y Limitaciones**

La presente investigación está orientada a proponer un Modelo Sistémico de Calidad para el Proceso de Pruebas de Software, que sea capaz de determinar la calidad del software focalizado en la visión sistémica, para ello se hace necesario en primera instancia enfocar el análisis en definir la calidad del software, además se analizará el proceso de pruebas de software como mecanismo para aseguramiento de calidad por excelencia, al mismo tiempo se realizará un análisis de cada uno de los modelos de calidad para el proceso de pruebas existentes, entre los cuales se puede mencionar: Modelo de Madurez de Prueba (TMM, por sus siglas en inglés), Modelo de Mejora del Proceso de Pruebas (TPI, por sus siglas en inglés), Modelo de Madurez Integrado de Prueba (TMMI, por sus siglas en inglés), Área de Proceso de Pruebas Integrado con CMMI (TESPAI, por sus siglas en inglés), y el estándar en desarrollo

ISO/IEC 29119 Software Testing, con la finalidad de determinar las características, fortalezas y debilidades de cada uno de ellos. Seguidamente, se realizará un estudio acerca de la teoría sistémica, así como los modelos existentes basados en este enfoque, en este caso el Modelo Sistémico de Calidad (MOSCA) del Software. En tal sentido, el modelo a plantear en la investigación debe satisfacer todas las características de madurez relacionados a la calidad sistémica para considerar que un proceso de pruebas se está desarrollando bajo este principio de calidad.

## CAPÍTULO II

### MARCO TEÓRICO

En este capítulo se encuentran los antecedentes y bases teóricas que establecen el conocimiento de la presente Tesis de Maestría. En la primera sección, se describen los trabajos existentes en relación a la mejora del proceso de pruebas que conformaran los antecedentes de la investigación, a continuación se expondrán las bases teóricas las cuales contribuirán a dilucidar los distintos conceptos relacionados con las pruebas del software se tomarán en cuenta la conceptualización, objetivos, proceso de pruebas. Seguidamente, se inicia la exposición de los trabajos existentes en relación a la mejora del proceso de pruebas.

#### **2.1 Antecedentes de la Investigación**

A continuación se expondrá lo relacionado a los modelos existentes en el área de pruebas de software, con la finalidad de determinar los aspectos relevantes para el presente estudio.

##### **2.1.1 Integración de Modelos de Capacidad de Madurez (CMMI, por sus siglas en inglés).**

El Instituto de Ingeniería de Estados Unidos (SEI, por sus siglas en inglés), a saber, Integración de Modelos de Capacidad de Madurez (CMMI, por sus siglas en inglés), propuso un modelo de madurez de mejora de los procesos para el desarrollo de productos y de servicios, el cual consiste en las mejores prácticas

que tratan las actividades de desarrollo y mantenimiento que cubren el ciclo de vida del producto, desde la concepción a la entrega y el mantenimiento. Existen 22 áreas de procesos de las cuales sólo dos cubren el proceso de pruebas, para efectos de esta investigación fue importante indagar como enfoca el CMMI el proceso de pruebas, lo cual se puede observar en la Tabla 03, según lo expuesto en la guía para la integración de procesos (2009).

Área	Descripción	Metas	Prácticas Específicas
Validación (VAL)	Es un área de proceso de ingeniería en el nivel de madurez 3. El propósito de la validación (VAL), es demostrar que un producto o componente de producto se ajusta a su uso previsto cuando se sitúa en su entorno previsto.	SG1 Preparar la Validación	SP1.1 Seleccionar los productos a validar  SP 1.2 Establecer el entorno de validación  SP 1.3 Establecer los procedimientos y los criterios de validación.
		SG 2 Validar el producto o los componentes de producto	SP 2.1 Realizar la validación  SP 2.2 Analizar los resultados de la validación.
Verificación (VER)	Es un área de proceso de ingeniería en el nivel de madurez 3. El propósito de la Verificación (VER), es asegurar que los productos de trabajo seleccionados cumplen sus requisitos especificados.	SG 1 Preparar la Verificación	SP 1.1 Seleccionar los productos de trabajo a verificar  SP 1.2 Establecer el entorno de verificación  SP 1.3 Establecer los procedimientos y los criterios de verificación.

		SG 2 Realizar las revisiones entre pares	SP 2.1 Preparar las revisiones entre pares SP 2.2 Llevar a cabo las revisiones entre pares SP 2.3 Analizar los datos de la revisión entre pares
		SG 3 Verificar los productos de trabajo seleccionados	SP 3.1 Realizar la verificación SP 3.2 Analizar los resultados de la verificación

**Tabla 03:** Metas y subprácticas relacionadas a las áreas de Validación y Verificación del Modelo Integración de Modelos de Capacidad de Madurez (CMMI, por sus siglas en inglés).

**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2011)

Se puede observar que las áreas de proceso de Validación y Verificación son similares para CMMI, pero tratan aspectos diferentes. La validación demuestra que el producto, tal como se proporcionará, se ajustará a su uso previsto, mientras que la verificación trata sobre si el producto de trabajo refleja apropiadamente los requisitos especificados.

El presente modelo constituye un antecedente para el estudio, debido a que busca ayudar a las organizaciones a la mejora de la capacidad de madurez de sus procesos durante todo el ciclo de vida del producto. Sin embargo, las áreas

definidas para el proceso de pruebas Validación (VAL) y Verificación (VER), no cubren todas las necesidades del proceso de pruebas, ya que no se encuentran definidas todas las mejores prácticas relacionadas con las pruebas software ni el conjunto de los productos de trabajo que son necesarios. Sanz et.al (2009).

En este sentido, existen una serie de modelos y estándares de madurez específicos para el proceso de prueba de software, debido a lo crítico del proceso de pruebas para garantizar la calidad del software, es importante analizar los aportes que cada uno de estos modelos y estándares proporcionan a la presente investigación, los cuales permitirán disponer de los fundamentos teóricos necesarios para proponer el modelo objeto de estudio de la presente investigación.

Para Sanz et al. (2008), un modelo de referencia orientado al proceso de pruebas define el marco de referencia necesario para poder determinar las fortalezas y debilidades del proceso de pruebas implementado en la organización, es decir, para poder determinar el estado actual del proceso y establecer el plan de mejora, se recogerán las prácticas existentes en la organización y se compararán con las propuestas en el modelo de referencia, por tanto el primer requisito de un modelo de referencia dirigido al proceso de pruebas será contener todas las prácticas relativas a las pruebas.

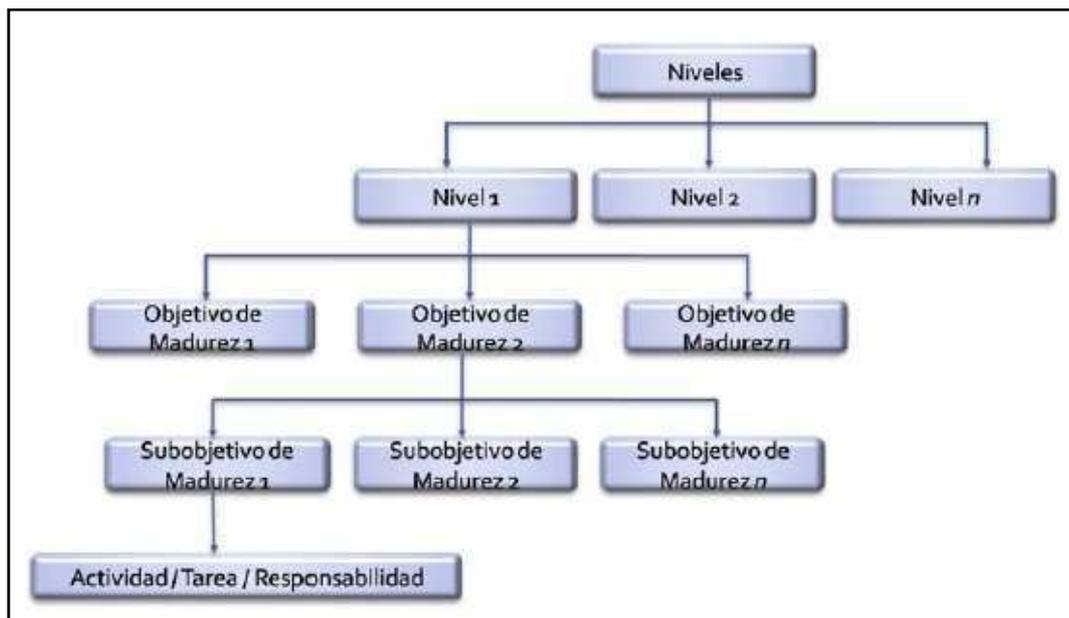
En la actualidad, los modelos de referencia relacionados con la mejora del proceso de pruebas más destacables son los siguientes: Modelo de Madurez de Prueba (TMM, por sus siglas en inglés,1996), Modelo de Mejora del Proceso de Pruebas (TPI, por sus siglas en inglés,1999), Modelo de Madurez Integrado de Prueba (TMMI, por sus siglas en inglés,2008), Área de Proceso de Pruebas Integrado con CMMI (TESPAI, por sus siglas en inglés,2008), y el estándar en desarrollo ISO/IEC 29119 Software Testing(2007-aún en desarrollo), los cuales están íntimamente relacionados con el tema a desarrollar.

### 2.1.2 Modelo de Madurez de Prueba (TMM, por sus siglas en inglés)

Según Swinkels (2000), fue creado por el Instituto de Tecnología de Illinois en 1996. El propósito del modelo es apoyar la mejora del proceso de pruebas dentro de la organización.

Por su parte, Sanz et.al (2008) lo define de la siguiente manera, un modelo de referencia orientado al proceso de pruebas. La mejora del proceso de pruebas es soportada por una única área de proceso que define un conjunto de niveles y objetivos de madurez en el TMM.

El modelo de referencia propone 5 niveles de madurez y describe una estructura interna implementada en cada uno de ellos, tal como se puede observar en la Figura 01.



**Figura 01:** Estructura de TMM

**Fuente:** Revista Española de Innovación, Calidad e Ingeniería del Software, Vol.4, No. 4, 2008

**Autor:** Sanz et al. (2008)

Así, el modelo TMM, propone lo siguiente para lograr un proceso de pruebas maduro:

- a) Niveles de Madurez, cada nivel de madurez representa la madurez específica de prueba, en evolución hacia un proceso de pruebas maduro.
- b) Objetivos de Madurez, para alcanzar un cierto nivel de madurez, se han identificado metas que deben seguirse para mejorar las pruebas.
- c) Subobjetivo de Madurez, los cuales son más concretos y están definidos para cada objetivo de madurez
- d) Actividad/Tarea/Responsabilidad, las actividades y tareas son definidas en términos de acciones y deben ser realizadas para mejorar la capacidad de las pruebas, dichas actividades y tareas están asignadas a un grupo clave de participantes en el proceso de pruebas.

Entre las contribuciones que brinda este trabajo a la presente investigación se pueden mencionar las siguientes:

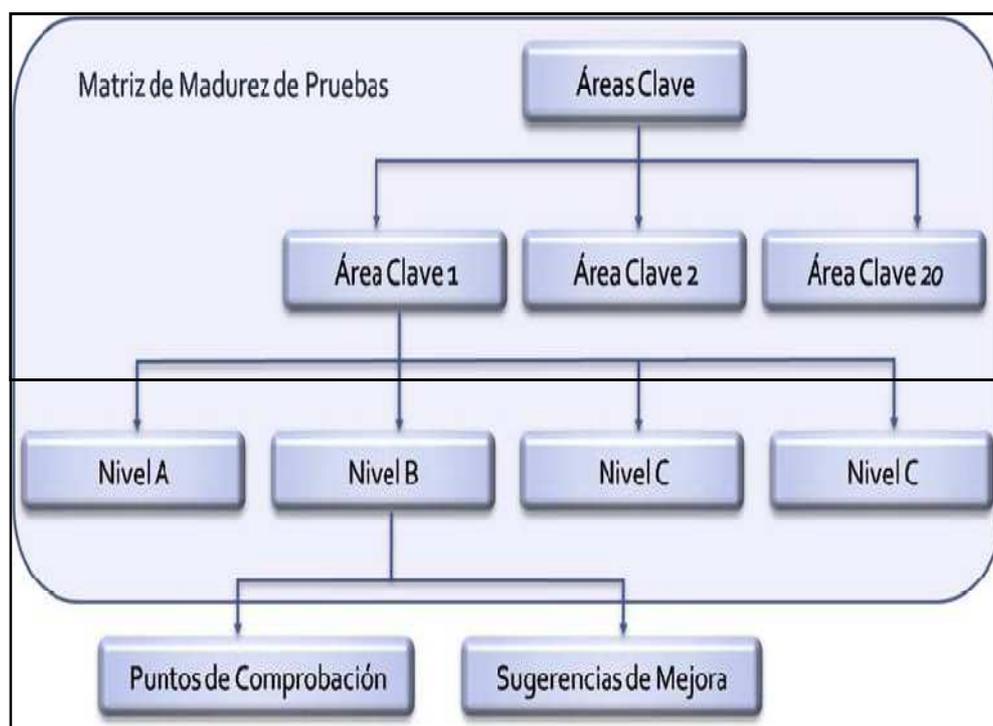
- Incorporación de participantes claves en el proceso de pruebas: Dirección superior, desarrolladores, probadores y por último clientes/usuarios.
- Planificación y organización de las pruebas, implica establecer objetivos, analizar riesgos, delinear estrategias, realizar el diseño de las pruebas y los casos de prueba.
- Evaluar la calidad del software, lo cual incluye definir atributos de calidad que puedan medirse.

Para dar continuidad al estado del arte con respecto a los modelos de mejora para el proceso de pruebas, seguidamente se analizará el Modelo de Mejora del Proceso de Pruebas (TPI, por sus siglas en inglés).

### 2.1.3 Modelo de Mejora del Proceso de Pruebas (TPI, por sus siglas en inglés)

Según Swinkels (2000), el modelo TPI fue desarrollado por Koomen y Pol en el año 1997, la realización de este modelo se basó en considerar las pruebas de software, como importante pero difícil e incontrolado proceso.

Recientemente, Sanz et al. (2008) lo describe como un modelo de referencia dirigido al proceso de pruebas, el cual propone un conjunto de 20 áreas claves con diferentes niveles de madurez. Además, define una Matriz de Madurez de Pruebas para establecer las relaciones existentes entre las diferentes áreas, y determinar la relevancia de cada uno de los niveles. Por último, establece un conjunto de Puntos de Comprobación y Sugerencias de Mejora. La estructura de TPI, se muestra en la Figura 02.



**Figura 02:** Estructura de TPI.

**Fuente:** Revista Española de Innovación, Calidad e Ingeniería del Software, Vol.4, No. 4, 2008.

**Autor:** Sanz et al. (2008).

Seguidamente, se describe cada uno de los aportes del modelo en estudio para la presente investigación.

- Administración del proceso de prueba, para cada actividad deben definirse propósito, entradas, proceso, salida, dependencias, técnicas aplicables, herramientas, documentación.
- Incorporación de métricas, sobre el progreso del proceso y la calidad del sistema.
- Entorno de pruebas, haciendo énfasis en la configuración del mismo, así como en la definición de responsabilidades, administración y disponibilidad.
- Compromiso y motivación, los cuales son pre requisitos para ejecutar el proceso de pruebas sin problemas.
- Comunicación, dentro del equipo de pruebas y con el resto de las partes como son desarrolladores, cliente, usuarios, entre otros.
- Reportes, informar al cliente sobre el producto y el proceso de desarrollo.

A continuación, se analizan los aspectos relacionados al Modelo de Madurez Integrado de Prueba.

#### **2.1.4 Modelo de Madurez Integrado de Prueba (TMMI, por sus siglas en inglés)**

Es un marco de referencia publicado en el 2008, desarrollado por la fundación TMMi, como guía para el mejoramiento de procesos de pruebas. Está estructurado de forma detallada para la mejora de procesos de pruebas y alineado como complemento al modelo CMMI (TMMi Foundation, 2009).

En el mismo orden de ideas, Sanz et.al (2008), define TMMI como un modelo de referencia dirigido al proceso de pruebas, el cual utiliza el concepto de niveles de madurez para la evaluación y mejora del proceso de pruebas. Dicho modelo, define cinco niveles de madurez con sus correspondientes áreas de proceso, así como los objetivos específicos y genéricos, así como prácticas específicas y genéricas.

En la Figura 03 se puede apreciar la estructura del modelo en estudio.



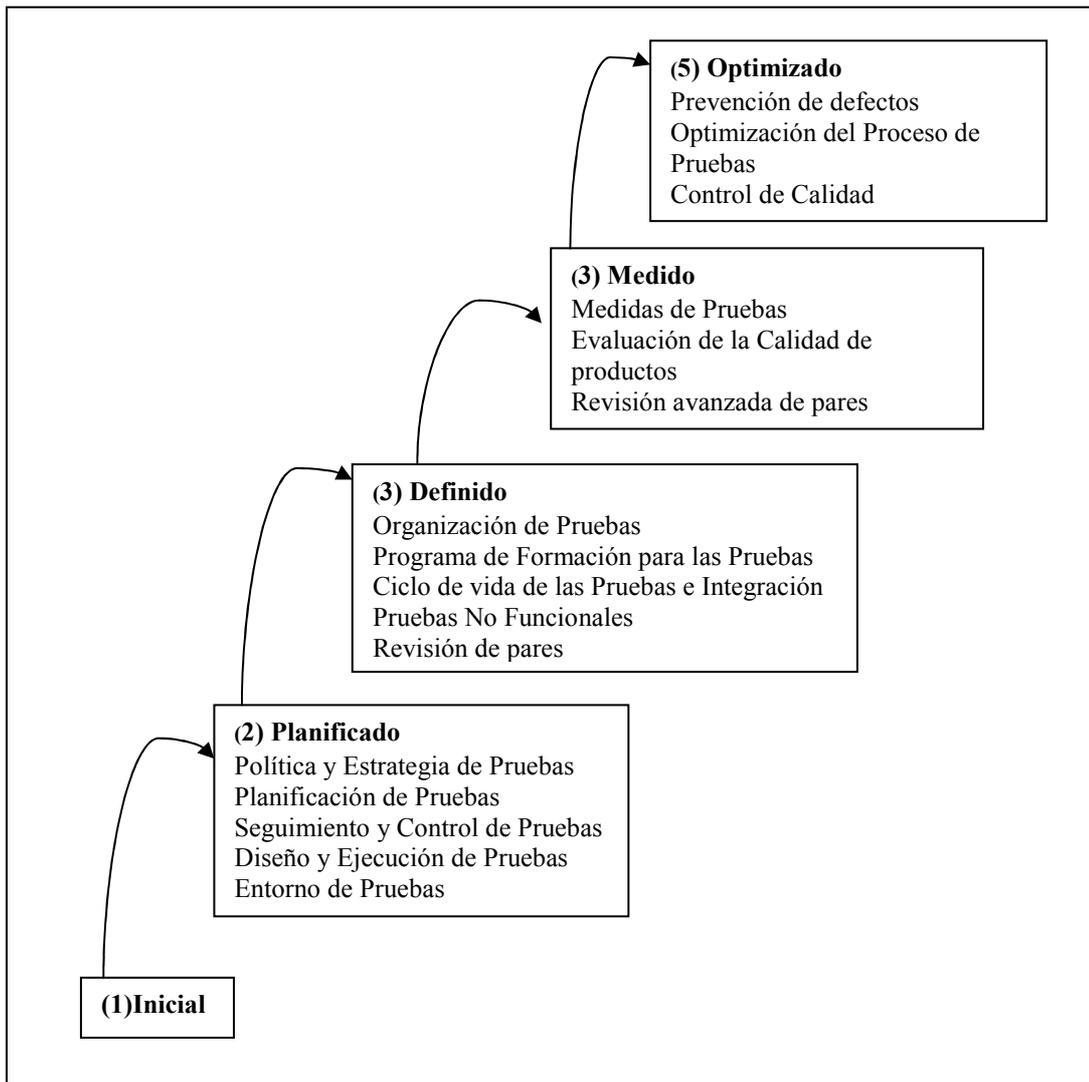
**Figura 03:** Estructura de TMMI

**Fuente:** Revista Española de Innovación, Calidad e Ingeniería del Software, Vol.4, No. 4, 2008

**Autor:** Sanz et al. (2008)

De la misma manera que CMMI, TMMI propone cinco niveles, los cuales muestran una jerarquía de madurez y un camino evolutivo para la mejora del proceso de pruebas, cada nivel está formado por un conjunto de áreas claves las cuales la organización debe implementar para lograr el nivel de madurez correspondiente.

Las áreas claves para cada nivel de madurez se pueden observar en la Figura 04.



**Figura 04:** Niveles y áreas Claves de TMMI

**Fuente:** <http://www.tmmifoundation.org/downloads/tmmi/TMMi%20Framework.pdf>

**Autor:** TMMi Foundation (2008).

En el contexto del trabajo de investigación, se describen cada uno de los aportes del modelo TMMI:

- Planificación de las pruebas, en una etapa temprana del proyecto.
- Evaluación de los productos según los criterios cuantitativos de los atributos de calidad tales como fiabilidad, facilidad de uso y mantenimiento.
- Procedimiento detallado para la selección de herramientas de pruebas que apoyen el proceso de pruebas durante el diseño, ejecución, gestión de casos de pruebas, entre otros.
- Prevención de defectos y control de calidad.

Es importante destacar que, la investigación realizada por Sanz et al. (2008), presentó un análisis de los modelos antes descritos para ello definieron seis criterios: Tipo, Cobertura completa, Completamente definido, Representación por etapas, Representación continua, Conformidad con CMMI. (Ver Tabla 04).

Criterio	Definición
Tipo	Tipo de modelo de mejora. Puede ser un modelo de mejora de procesos si cubre los diferentes procesos existentes en la organización, o un modelo de mejora del proceso de pruebas si solamente cubre el proceso de pruebas.
Cobertura Completa	Un modelo de referencia proporciona Cobertura Completa si incluye todas las prácticas relativas a las pruebas.
Completamente Definido	Un modelo de referencia está completamente definido si todas las prácticas definidas en el modelo están completamente detalladas.
Representación por Etapas	Un modelo de referencia proporciona la representación por etapas si describe todos los objetivos y prácticas necesarias para establecer el nivel de madurez de la organización.
Representación Continua	Un modelo de referencia proporciona representación continua si describe todos los objetivos y prácticas necesarias para establecer el nivel de capacidad de la organización.
Conformidad con CMMI	Un modelo de referencia está en Conformidad con CMMI si tiene una estructura compatible con la estructura de CMMI y la integración entre ambos modelos puede realizarse de forma completa y sencilla.

**Tabla 04:** Criterios definidos para realizar el análisis de los modelos de referencia.

**Fuente:** Revista Española de Innovación, Calidad e Ingeniería del Software, Vol.4, No. 4, 2008

**Autor:** Sanz y otros (2008)

Los resultados obtenidos sobre los modelos de referencia y teniendo en cuenta los criterios definidos se presentan en la Tabla 05.

Criterio/Modelo	CMMI	TMM	TMMi	TPI
Tipo	Modelo de Mejora de procesos	Modelo de Mejora del proceso de pruebas	Modelo de Mejora del proceso de pruebas	Modelo de Mejora del proceso de pruebas.
Cobertura Completa	No	Si	Si	Si
Completamente Definido	Si	Si	Si	No
Representación por Etapas	Si	Si	Si	No
Representación Continua	Si	No	No	Si
Conformidad con CMMI	Si	No	No	No

**Tabla 05:** Descripción de las características de los modelos de mejora del proceso de pruebas.

**Fuente:** Revista Española de Innovación, Calidad e Ingeniería del Software, Vol.4, No. 4, 2008

**Autor:** Sanz y otros (2008).

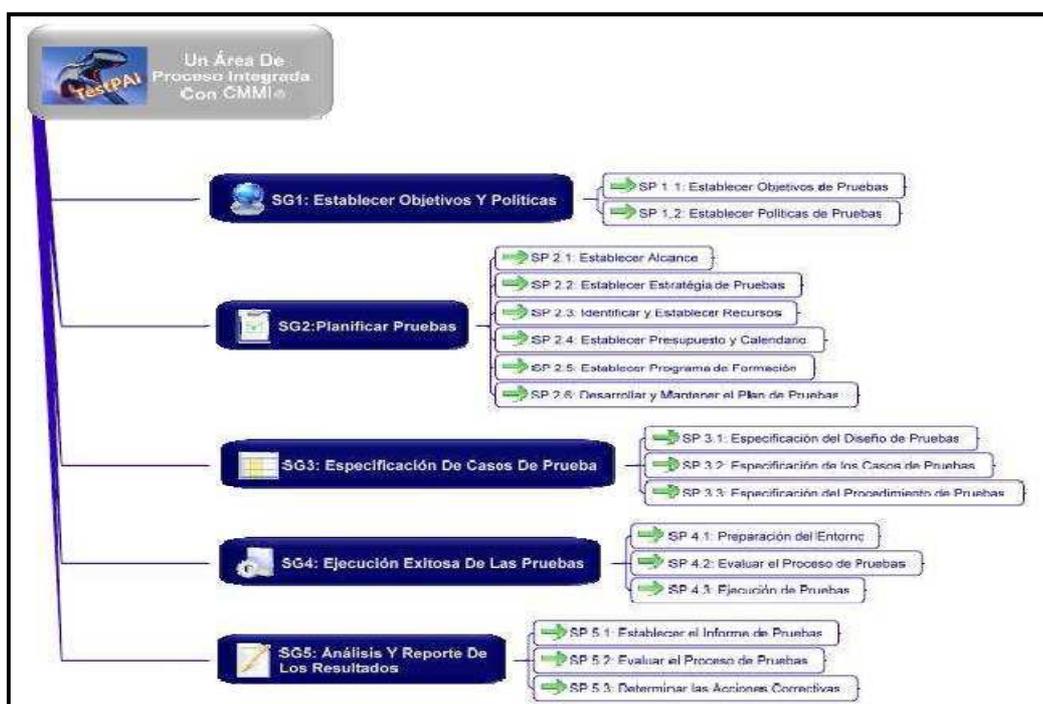
Tras el análisis de los resultados, Sanz y otros (2008), determinaron que ninguno de los modelos de referencia existentes soluciona el problema de mejorar el proceso de pruebas mientras se realiza la mejora de otros procesos de la organización integrado totalmente y de una forma sencilla con CMMI.

A continuación, se procede a analizar el Área de Proceso de Pruebas Integrados con CMMI, el cual surge con la hipótesis de desarrollar un área de proceso de pruebas que contenga todas las prácticas relativas a las pruebas y se integre totalmente y de una forma sencilla con CMMI.

### 2.1.5 Área de Proceso de Pruebas Integrado con CMMI (TESPAI, por sus siglas en inglés)

Sanz et al. (2008), define TESPAI como un área de proceso de pruebas integrado con CMMI. Se encuentra situada en el nivel 3 con los procesos de Ingeniería, ha sido desarrollado para proporcionar el marco de trabajo necesario para que la mejora del proceso de pruebas se desarrolle de forma paralela a la mejora de otros procesos implementados en la organización. TESPAI, incluye y define todas las prácticas relativas a las pruebas, además, contempla tanto la representación continua como la representación por etapas.

Además, define cinco objetivos específicos (SG) y sus correspondientes prácticas específicas (SP), las cuales se pueden apreciar en la Figura 05.



**Figura 05:** Objetivos y Prácticas Específicas de TestPAI.

**Fuente:** Revista Española de Innovación, Calidad e Ingeniería del Software, Vol.4, No. 4, 2008

**Autor:** Sanz et al. (2008).

A continuación se procede a detallar cada uno de los aportes de TESTPAI al presente trabajo de investigación:

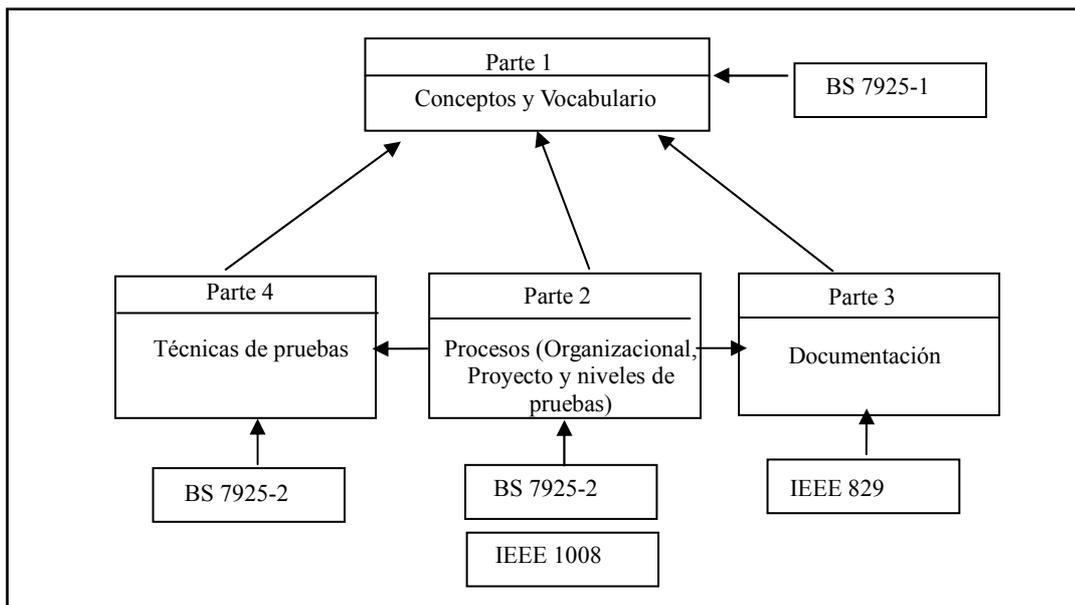
- Planificación de las pruebas, implica establecer y analizar los diferentes elementos que afectan la gestión y desarrollo de las pruebas, tales como definir alcance, analizar riesgos, definir la estrategia o asignar recursos entre otros.
- Especificación de casos de pruebas, en los cuales se describen características tales como entradas, salidas o resultados esperados de una prueba.
- Establecer la infraestructura adecuada para el desarrollo de las pruebas de software.
- Analizar, evaluar y comunicar los resultados obtenidos.

Luego del análisis de TESTPAI se procede a conocer el estándar en desarrollo ISO/IEC 29119 Software Testing, el cual se viene gestando desde el año 2007.

#### **2.1.6 ISO/IEC 29119 Software Testing**

Según Tuya (2009)<sup>a</sup>, este estándar comenzó su camino en 2007, año en que el ISO aprobó la constitución del grupo de trabajo 26(WG26, por sus siglas en inglés) dentro del subcomité ISO/IEC JTC1/SC7. Posteriormente, AENOR constituyó el correspondiente grupo de trabajo AEN/CTN 71/SC7/GT26, para colaborar en la elaboración del estándar.

La estructura de ISO/IEC 29119 consta de cuatro partes, las cuales se pueden observar en la Figura 06.



**Figura 06:** Estructura ISO/IEC 29119(Traducida)

**Fuente:** IV Taller sobre Pruebas en Ingeniería del Software

**Autor:** Tuya (2009)<sup>b</sup>

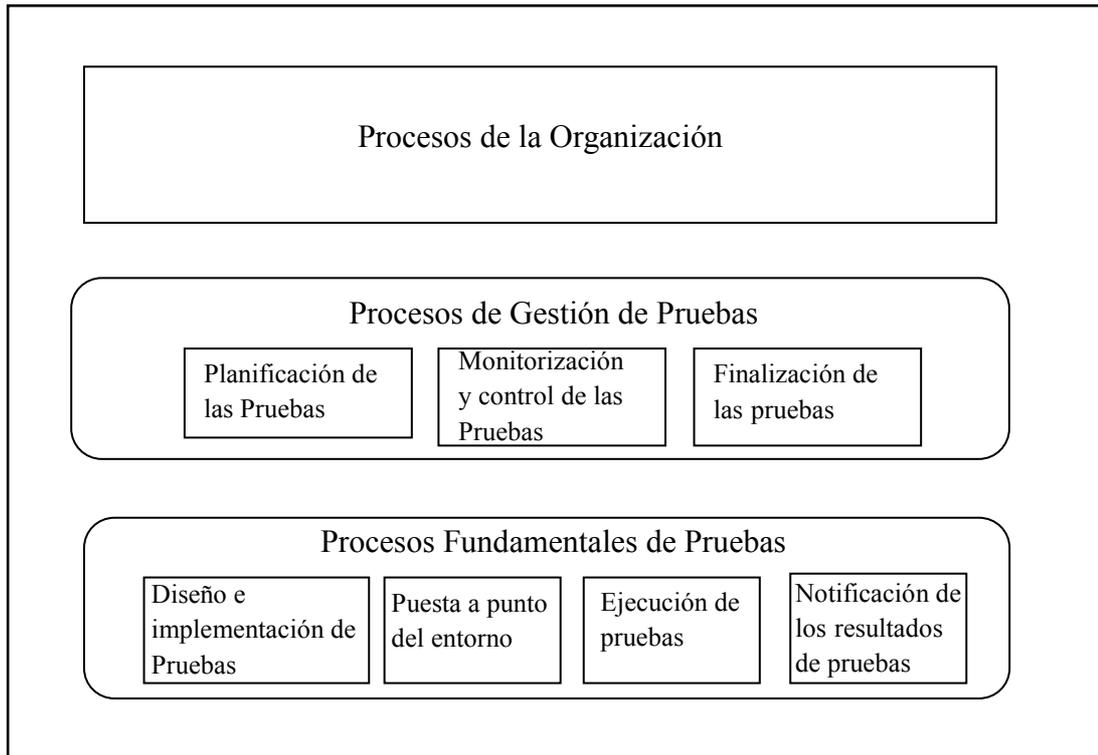
A continuación se procede a describir cada uno de los elementos que conforman la estructura del estándar en estudio.

### **Parte 1. Conceptos y Vocabulario**

El objetivo de este componente, es proporcionar una visión general de la norma y de los conceptos generales relacionados al proceso de pruebas de software.

### **Parte 2. Proceso de Pruebas**

Este elemento está orientado hacia un proceso genérico en lo que respecta a las políticas y estrategias, gestión, diseño y ejecución de un proceso de pruebas. Para ilustrar el modelo de procesos tal y como figura en el borrador de trabajo se dispone de la Figura 07, en la cual se puede apreciar los elementos que conforman este componente del estándar.



**Figura 07:** Parte 2-Procesos (Versión Revisada y Traducida)

**Fuente:** IV Taller sobre Pruebas en Ingeniería del Software

**Autor:** Tuya (2009)<sup>b</sup>

En particular, se observa que el modelo está formado por tres niveles. En el nivel superior, se encuentran los procesos de la organización que no son específicos de un determinado proyecto de pruebas. Permiten definir las políticas y estrategias aplicables a toda la organización o a una línea de proyectos.

Por otra parte, para un proyecto de pruebas se tienen los niveles de procesos de gestión y procesos fundamentales. Con respecto a los procesos de gestión, definen un conjunto de procesos de gestión genéricos para permitir flexibilidad y adaptación a diferentes contextos. Estos procesos son la planificación, monitorización y control, y finalización de las pruebas.

Finalmente se encuentra, el nivel correspondiente a los procesos fundamentales, los cuales abarcan los aspectos técnicos de las pruebas: Diseño e implementación, puesta a punto del entorno, ejecución de pruebas y la notificación de los resultados de pruebas. También se incluirán variantes de los procesos para contemplar tanto pruebas dinámicas como estáticas.

### **Parte 3 - Documentación de pruebas**

Se centra en la documentación de las pruebas a lo largo de todo ciclo de vida del proceso de pruebas de software. Esto contendrá plantillas de todas las capas del modelo de proceso 29119, incluyendo:

- Política organizacional del Proceso de prueba.
- Estrategia organizacional del Proceso de prueba.
- Proyecto de Gestión del Proceso de prueba.
- Proceso de pruebas fundamentales.

Es importante acotar que, IEEE ha dado el permiso de ISO para utilizar el conocido Estándar de Documentación de Pruebas IEEE 829 como base para esta parte de la norma. ISO 29119 eventualmente reemplazará IEEE 829.

### **Parte 4 - Técnicas de pruebas**

Esta parte del estándar cubrirá las técnicas de pruebas del software a través de todos los tipos de pruebas, incluyendo estáticas (por ejemplo: revisiones, inspecciones, tutoriales), funcionales (por ejemplo: caja negra, caja blanca), no funcionales (por ejemplo funcionamiento, seguridad, utilidad) y basadas en experiencia (por ejemplo: cálculo de error, experimental).

El estándar ISO/IEC 29119 puede aportar ventajas significativas a la mejora del proceso de pruebas del software ya que proveerá a los profesionales de Ingeniería de Software una guía sobre pruebas la cual abordará todos los aspectos del ciclo de vida (conceptos, vocabulario, proceso, documentación, técnicas), no obstante aún se encuentra en fase de desarrollo por lo cual se debe esperar para obtener mayor detalle acerca del mismo.

A manera de resumen, se presenta la Tabla 06 y Tabla 07, las cuales contienen la descripción de las características de los modelos en estudio, así como una

sinopsis de las características de los modelos y estándares analizados en esta sección respectivamente.

Característica	Definición
Tipo	Tipo de modelo de mejora. Puede ser un modelo de mejora de procesos si cubre los diferentes procesos existentes en la organización, o un modelo de mejora del proceso de pruebas si solamente cubre el proceso de pruebas.
Año de Desarrollo/Publicación	Se refiere al año en el cual se desarrolló o año de publicación del modelo en estudio.
Número de Niveles	Se utilizan para describir el camino evolutivo para la mejora del proceso de pruebas en una organización.
Número de áreas claves	Se refieren al conjunto de prácticas que cuando se implementan de forma conjunta, satisfacen un grupo de objetivos considerados importantes para mejorar esa área.
Cobertura de las actividades de Pruebas	Un modelo de referencia provee cobertura de las actividades de pruebas si todas las prácticas definidas en el modelo están completamente definidas.
Elementos de Evaluación	Se refiere a los métodos o instrumentos utilizados por el modelo para determinar la madurez del proceso de pruebas.
Información sobre el Modelo/Estándar	Especifica la fuente en donde se puede ubicar información acerca del modelo.
Atributos de Calidad	Enumera características de calidad para el proceso de pruebas de software.

**Tabla 06:** Definición de las características para realizar el análisis de los modelos en estudio.

**Fuente:** Elaboración Propia.

**Autor:** Autor de la investigación (2012).

Característica	TMM	TPI	TMMI	TESPAI	ISO/IEC 29119
Tipo	Modelo de Mejora del proceso de pruebas	Modelo de Mejora del proceso de pruebas	Modelo de Mejora del Proceso de Pruebas	Modelo de Mejora del proceso de pruebas	Estándar ISO/IEC
Año de Desarrollo/Publicación	1996	1999	2008	2008	2007- Aún en desarrollo
Número de Niveles	5	14	5	5	3
Número de áreas claves	13	20	18	17	No aplica
Cobertura de las actividades de Pruebas	Completa	Completa	Completa	Completa	Completa
Elementos de Evaluación	Cuestionario	* Lista de Comprobación. *Matriz de madurez de prueba.	*Cuestionario. *Entrevistas. *Método de evaluación TAMAR	Lista de Verificación	No aplica
Información sobre el Modelo/Estándar	Artículos	Artículos	*Página Oficial TMMi Foundation *Artículos	Artículos	Artículos
Atributos de Calidad	No	No	Si	No	No

**Tabla 07:** Tabla comparativa de los modelos de mejora de procesos de pruebas.

**Fuente:** Elaboración Propia.

**Autor:** Autor de la investigación (2012).

Ahora bien, es necesario agregar que los modelos en estudio se orientan a la madurez del proceso de pruebas, en otras palabras, la mejora del proceso de pruebas, ya que se toma en cuenta categorías dirigidas al mismo, entre las cuales se puede mencionar: Ingeniería, Gestión, Aspectos organizacionales, sin embargo, sólo el TMMI menciona características de calidad como: Fiabilidad, Facilidad de uso y Mantenimiento, descartando la Funcionalidad, Fiabilidad, Eficiencia, Portabilidad, las cuales son vitales para garantizar la calidad del producto. En este sentido, el objetivo de la presente investigación es: Proponer un modelo sistémico de calidad para el proceso de pruebas de software, por lo cual se hace necesario considerar aspectos relacionados a la calidad del proceso de pruebas, así como, las características de calidad relacionadas al producto, es decir, el modelo propuesto debe poseer una visión de calidad sistémica.

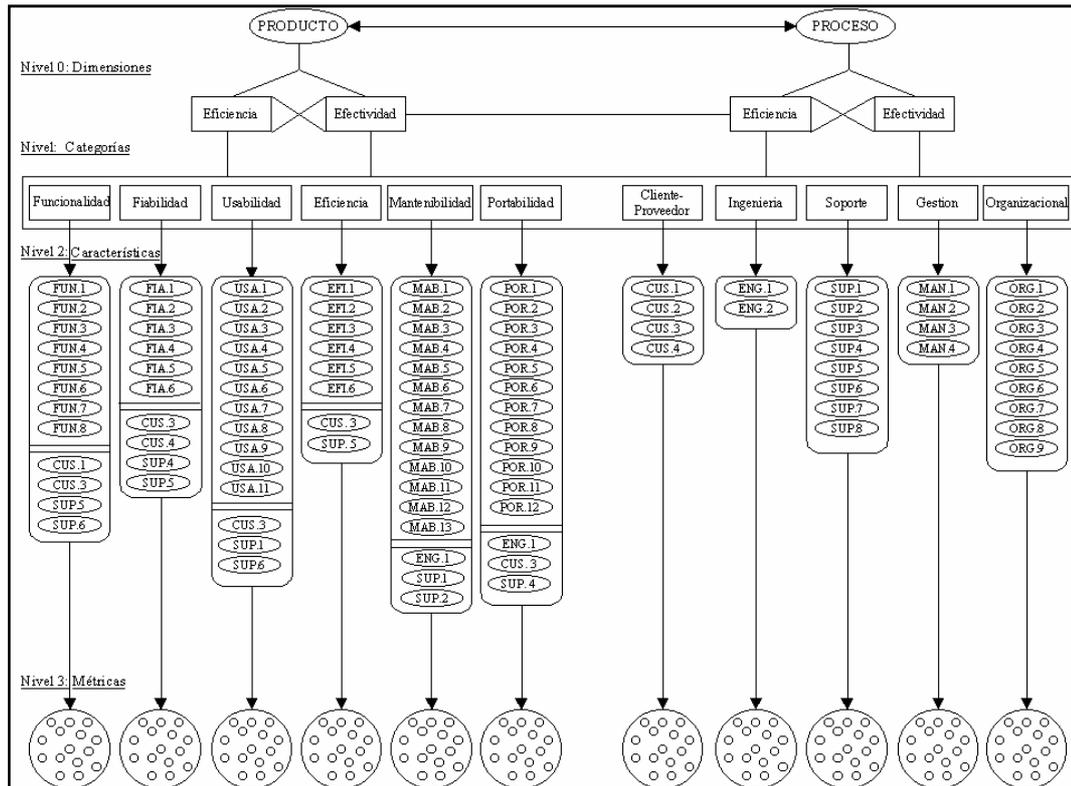
En relación con lo anteriormente expuesto, la presente investigación se apoyará en el Modelo Sistémico de Calidad (MOSCA) del Software Mendoza et.al (2005), desarrollado por el LISI\_USB, el cual se analizará a continuación.

### **2.1.7 Modelo Sistémico de Calidad del Software (MOSCA)**

El Modelo Sistémico de Calidad del Software (MOSCA), es un modelo desarrollado por LISI-USB que se basa en la calidad sistémica para la medición de la calidad de los sistemas de software, integra el modelo de calidad del producto (Ortega et al., 2002) y el modelo de calidad del proceso de desarrollo (Pérez et al., 2001).

En cuanto al producto, este modelo plantea, sobre la base de las seis características de calidad del estándar internacional ISO/IEC 9126(JTC 1/SC 7,1991), un conjunto de subcaracterísticas y métricas asociadas, que miden la calidad de un producto software con un enfoque sistémico, las cuales hacen del modelo un instrumento de medición de gran valor ya que cubre todos los aspectos necesarios e imprescindibles para medir directamente la calidad del producto de software. Mendoza et.al (2005).

Ahora bien, en lo que respecta al proceso el modelo basa sus características en la adaptación del modelo de procesos SPICE conocida también como ISO/IEC 9126 al modelo de calidad sistémica. Pérez et.al (2001) citado por Mendoza et.al (2005). Seguidamente, se presenta la estructura del modelo que se basa en el enfoque sistémico (Ver Figura 08).



**Figura 08:** Estructura de MOSCA.  
**Fuente:** Universidad Simón Bolívar Venezuela.  
**Autor:** Mendoza et.al (2005).

Como se puede apreciar en la Figura 08, la estructura del Modelo Sistémico de Calidad (MOSCA) del Software consta de cuatro niveles, los cuales son explicados a continuación:

**(a) Nivel 0 Dimensiones**

Este nivel contempla las cuatro dimensiones propuestas por el modelo a saber: Aspectos Internos del Proceso, Aspectos Contextuales del proceso, Aspectos Internos del Producto y Aspectos Contextuales del Producto. Sólo un balance y

una buena interrelación entre ellas garantizan la calidad sistémica global de una organización.

**(b) Nivel 1 Categorías**

En total el modelo contempla once categorías: seis pertenecientes al producto y las otras cinco al proceso de desarrollo. Las categorías asociadas al producto se ilustran en la Tabla 08.

<b>Categoría del Producto</b>	<b>Definición</b>
Funcionalidad (FUN)	Capacidad del producto de software para proveer funciones que cumplan con necesidades específicas o implícitas, cuando el software es utilizado bajo ciertas condiciones
Fiabilidad (FIA)	Capacidad del producto de software para mantener un nivel especificado de rendimiento cuando es utilizado bajo ciertas condiciones.
Usabilidad (USA)	Capacidad del producto de software para ser atractivo, entendido, aprendido y utilizado por el usuario bajo condiciones específicas.
Eficiencia (EFI)	Capacidad del producto de software para proveer un rendimiento apropiado, relativo a la cantidad de recursos utilizados, bajo condiciones específicas
Mantenibilidad (MAB)	Capacidad del producto para ser modificado.
Portabilidad (POR)	Capacidad del producto de software para ser transferido de un ambiente a otro.

**Tabla 08:** Categorías del submodelo del Producto.

**Fuente:** Universidad Simón Bolívar Venezuela

**Autor:** Mendoza, L. y otros (2005)

De igual manera, se presentan las categorías asociadas al proceso (Ver Tabla 09)

Categoría del Proceso	Definición
Cliente-Proveedor (CUS)	Está conformada por procesos que impactan directamente al cliente, apoya el desarrollo y la transición del Software hasta el cliente, y provee la correcta operación y uso del producto o servicio de software.
Ingeniería (ENG)	Consiste en procesos que directamente especifican, implementan o mantienen el producto de software, su relación con el Sistema y su documentación
Soporte (SUP)	Consta de procesos que pueden ser empleados por cualquiera de los procesos (incluyendo a los de soporte) en varios niveles del ciclo de vida de adquisición.
Gestión (MAN)	Abarca los procesos que contienen prácticas genéricas, que pueden ser utilizadas por cualquier personal que dirija algún tipo de proyecto o proceso.
Organizacional (ORG)	Agrupa los procesos que establecen las metas comerciales de la organización y desarrollan bienes (valores) de proceso, producto y recurso, que ayudarán a la organización a alcanzar sus metas en los proyectos.

**Tabla 09:** Categorías del submodelo del Proceso.

**Fuente:** Universidad Simón Bolívar Venezuela.

**Autor:** Mendoza, L. y otros (2005).

Es importante destacar que esta división no implica un desligamiento entre ellas, simplemente se realiza para identificar a qué sector o submodelo pertenecen.

### (c) Nivel 2 Características

Cada categoría tiene asociado un conjunto de características, las cuales definen las áreas claves a satisfacer para lograr, asegurar y controlar la calidad tanto en el producto como en el proceso. Esto se debe a que algunas características de la calidad del proceso, impactan directamente en las categorías del producto al igual que ciertas características de la calidad del producto definen categorías del proceso.

Lo anteriormente planteado, ayuda a precisar que si una vez medidas las características asociadas a una característica en particular del producto, arroja resultados no deseados, se pueden analizar las características de la calidad del proceso asociadas a esa categoría del producto para encontrar las posibles causas.

En las Tablas 10 y 11 se muestran todas las características del modelo acompañadas con la clave única que las identifica.

Categoría	Características	
	Aspectos Contextuales del Producto	Aspectos Internos del Producto
<b>Funcionalidad (FUN)</b> Total de métricas: 46	FUN 1. Ajuste a los propósitos (16) FUN 2. Precisión (10) FUN 3. Interoperabilidad (7) FUN 4. Seguridad (2)	FUN 5. Correctitud (8) FUN 6. Estructurado (1) FUN 7. Encapsulado (1) FUN 8. Especificado (1)
	Sub-total de métricas: 35	Sub-total de métricas: 11
<b>Fiabilidad (FIA)</b> Total de métricas: 32	FIA 1. Madurez (17) FIA 2. Tolerancia a fallas (1) FIA 3. Recuperación (4)	FIA 4. Correctitud (8) FIA 5. Estructurado (1) FIA 6. Encapsulado (1)
	Sub-total de métricas: 22	Sub-total de métricas: 10
<b>Usabilidad (USA)</b> Total de métricas: 38	USA 1. Facilidad de comprensión (5) USA 2. Capacidad de Aprendizaje (9) USA 3. Interfaz Gráfica (5) USA 4. Operabilidad (13) USA 5. Conformidad con los estándares	USA 6. Completo (1) USA 7. Consistente (1) USA 8. Efectivo (1) USA 9. Especificado (1) USA 10. Documentado (1) USA 11. Auto-descriptivo (1)
	Sub-total de métricas: 37	Sub-total de métricas: 6
<b>Eficiencia (EFI)</b> Total de métricas: 10	EFI 1. Comportamiento del tiempo (2) EFI 2. Utilización de recursos (4)	EFI 3. Efectivo (1) EFI 4. No redundante (1) EFI 5. Directo (1) EFI 6. Utilizado (1)
	Sub-total de métricas: 6	Sub-total de métricas: 4
<b>Mantenibilidad (MAB)</b> Total de métricas: 79	MAB 1. Capacidad de análisis (2) MAB 2. Facilidad de Cambio (7) MAB 3. Estabilidad (4) MAB 4. Capacidad de prueba (3)	MAB 5. Acoplamiento (1) MAB 6. Cohesión (1) MAB 7. Encapsulado (1) MAB 8. Madurez del Software (17) MAB 9. Estructura de Control (4) MAB 10. Estructura de Información (9) MAB 11. Descriptivo (14) MAB 12. Correctitud (8) MAB 13. Estructural (5) MAB 14. Modularidad (3)
	Sub-total de métricas: 16	Sub-total de métricas: 63
<b>Portabilidad (POR)</b> Total de métricas: 44	POR 1. Adaptabilidad (9) POR 2. Capacidad de Instalación (4) POR 3. Co-existencia (2) POR 4. Capacidad de reemplazo (2)	POR 5. Consistente (1) POR 6. Parametrizado (3) POR 7. Encapsulado (1) POR 8. Cohesivo (1) POR 9. Especificado (1) POR 10. Documentado (1) POR 11. Auto-descriptivo (1) POR 12. No redundante (1) POR 13. Auditoria (6) POR 14. Manejo de la Calidad (3)
	Sub-total de métricas: 17	Sub-total de métricas: 19
	Calidad de los Datos (8) -abarca las dos dimensiones-	

**Tabla 10:** Distribución de las Características y métricas para medir la calidad sistémica del producto de software.

**Fuente:** Universidad Simón Bolívar Venezuela.

**Autor:** Mendoza, L. y otros (2005).

Categoría	Características	
	Aspectos Contextuales del Proceso	Aspectos Internos del Proceso
<b>Cliente – Proveedor (CUS)</b> Total de métricas: 57	CUS 1. Adquisición del Sistema o producto de Software (24) CUS 3. Determinación de Requerimientos (20) Sub-total de métricas: 44	CUS 2. Suministro (8) CUS 4. Operación (5) Sub-total de métricas: 13
<b>Ingeniería (ING)</b> Total de métricas: 29	ENG 1. Desarrollo (12) Sub-total de métricas: 12	ENG 2. Mantenimiento de Software y Sistemas (17) Sub-total de métricas: 17
<b>Soporte (SUP)</b> Total de métricas: 130	SUP 3. Aseguramiento de la Calidad (17) SUP 6. Revisión Conjunta (14) SUP 7. Auditoría (15) SUP 8. Resolución de Problemas (11) Sub-total de métricas: 57	SUP 1. Documentación (9) SUP 2. Gestión de Configuración (12) SUP 4. Verificación (6) SUP 5. Validación (6) SUP 6. Revisión Conjunta (14) SUP 7. Auditoría (15) SUP 8. Resolución de Problemas (11) Sub-total de métricas: 73
<b>Gestión (MAN)</b> Total de métricas: 91	MAN 1. Gestión (14) MAN 3. Gestión de Calidad (10) MAN 4. Gestión del Riesgo (12) Sub-total de métricas: 36	MAN 1. Gestión (14) MAN 2. Gestión de Proyecto (19) MAN 3. Gestión de Calidad (10) MAN 4. Gestión del Riesgo (12) Sub-total de métricas: 55
<b>Organizacional (ORG)</b> Total de métricas: 123	ORG 1. Lineam. Organizacionales (14) ORG 2. Gestión de Cambio (10) ORG 5. Mejoramiento del Proceso (16) ORG 8. Medición (11) ORG 9. Reuso (12) Sub-total de métricas: 63	ORG 3. Establecimiento del Proceso (11) ORG 4. Evaluación del Proceso (9) ORG 5. Mejoramiento del Proceso (16) ORG 6. Gestión de RRHH (16) ORG 7. Infraestructura (8) Sub-total de métricas: 60

**Tabla 11:** Distribución de las Características y métricas para medir la calidad sistémica del proceso de desarrollo.

**Fuente:** Universidad Simón Bolívar Venezuela

**Autor:** Mendoza, L. y otros (2005)

#### (d) Métricas

Para cada característica se propone una serie de métricas utilizadas para medir la calidad sistémica. Dada la cantidad de métricas asociadas a MOSCA no serán tratadas en su totalidad en la presente investigación. Un total de 679 métricas asociadas. (Tablas 10 y 11).

El estudio realizado por Mendoza et.al (2005), analizó la influencia que tienen las características del proceso en las categorías del sub-modelo del producto de software, a continuación se describen cada una de éstas características.

#### **Características del proceso de desarrollo que impactan directamente en las categorías del producto.**

Según Mendoza et.al (2005), cada categoría del modelo del producto pareciera tener una dependencia específica de ciertas características del modelo del proceso, lo que implica que si estas características se cumplieron a cabalidad durante el proceso de desarrollo entonces la categoría del producto debiera poseer los requerimientos mínimos de calidad. Siguiendo la línea del trabajo de

investigación, se seleccionan las relacionadas al proceso de pruebas de software. A continuación, se explican cada una de las características.

#### **2.1.7.1 Características del proceso que influyen en la Funcionalidad del producto de Software**

- **SUP 5 Proceso de Validación:** En el proceso de validación se debe confirmar que los requerimientos para un uso específico del producto del sistema (software) sean satisfechos, es decir, este proceso garantiza que la Funcionalidad del producto de software se cumpla.
- **SUP 6 Proceso de Revisión Conjunta:** Este proceso afecta directamente a la Funcionalidad del producto, ya que se refiere al mantenimiento de un común entendimiento con el cliente sobre el progreso del proceso o del proyecto, en contraste con los objetivos del contrato. De esa manera, si se realiza correctamente dicha característica, el producto resultante se aproximará bastante a los requerimientos planteados por el cliente.

#### **2.1.7.2 Características del proceso que influyen en la Fiabilidad del producto de Software**

- **SUP 4 Proceso de Verificación:** Este proceso afecta a la Fiabilidad del producto ya que se desea confirmar que cada producto trabajado y/o servicio de un proceso o proyecto refleje apropiadamente los requerimientos especificados. El proceso de verificación asegura que el producto de software tenga todos los componentes necesarios para cumplir con los requerimientos para los cuales fue diseñado.
- **SUP 5 Proceso de Validación:** Para que un producto sea fiable, se debe confirmar que los requerimientos para el uso específico del producto del sistema (software) sean satisfechos. Si este proceso se realiza adecuadamente, se contará con un punto de control que constate que el

producto de software funcione correctamente para lo que fue diseñado, ya que se confirma la satisfacción de los requerimientos en el proceso de desarrollo.

#### **2.1.7.3 Características del proceso que influyen en la Usabilidad del producto de Software**

- **SUP 6 Proceso de Revisión Conjunta:** Esta característica se refiere al mantenimiento de un común entendimiento con el cliente sobre el progreso del proceso o proyecto, en contraste con los objetivos del contrato. Influye en la usabilidad ya que el cliente siempre estará al tanto de cómo es la interacción con el producto.

#### **2.1.7.4 Características del proceso que influyen en la Eficiencia del producto de Software**

- **SUP 5 Proceso de Validación:** Para que un producto sea Eficiente, se deben confirmar los requerimientos específicos que debería tener el producto. Si este proceso se realiza adecuadamente, el producto de software no utilizará más recursos de los necesarios, ya que los mismos fueron confirmados en el proceso de desarrollo.

#### **2.1.7.5 Características del proceso que influyen en la Mantenibilidad del producto de Software**

- **SUP 1 Proceso de Documentación:** Este proceso influye en la Mantenibilidad de un producto de software ya que tiene como fin desarrollar, mantener y actualizar, durante todo el desarrollo, los documentos que registran información sobre su evolución.

### 2.1.7.6 Características del proceso que influyen en la Portabilidad del producto de Software

- **SUP 4 Proceso de Verificación:** Esta característica afecta a la Portabilidad del producto de software ya que la misma debe confirmar que el producto refleje apropiadamente los requerimientos especificados, en particular que tenga la capacidad de ser transferido de un ambiente al otro.

Ahora bien, existen características del proceso que impactan en todas las categorías del producto de software, las cuales se describen a continuación

- **MAN 3 Proceso de Gestión de Calidad:** Esta característica del proceso influye en todas las categorías del producto ya que tiene como finalidad monitorear la calidad de los productos y/o servicios a lo largo del proceso de desarrollo. Si no se cumple esta característica, el producto resultante no garantizará un nivel de calidad adecuado y por ende el mismo no se adaptará a las condiciones para las cuales fue diseñado.
- **SUP 3 Proceso de Aseguramiento de la Calidad:** Esta característica también afecta a todas las categorías del producto porque proporciona seguridad con el objeto de que los procesos de un proyecto cumplan con los requerimientos específicos y se adhiera a los planes establecidos. Si no se cumple adecuadamente esta característica, no se garantizará la calidad en el producto de software debido a que el mismo no cumplirá con los requerimientos necesarios para funcionar bajo condiciones específicas.
- **SUP 7 Proceso de Auditoria:** Esta característica también afecta a todas las categorías del producto ya que tiene como finalidad determinar independientemente la complacencia de los productos seleccionados y los procesos con requerimientos, planes y contratos pertinentes. Esta característica afecta a cada una de las categorías ya que garantiza que el

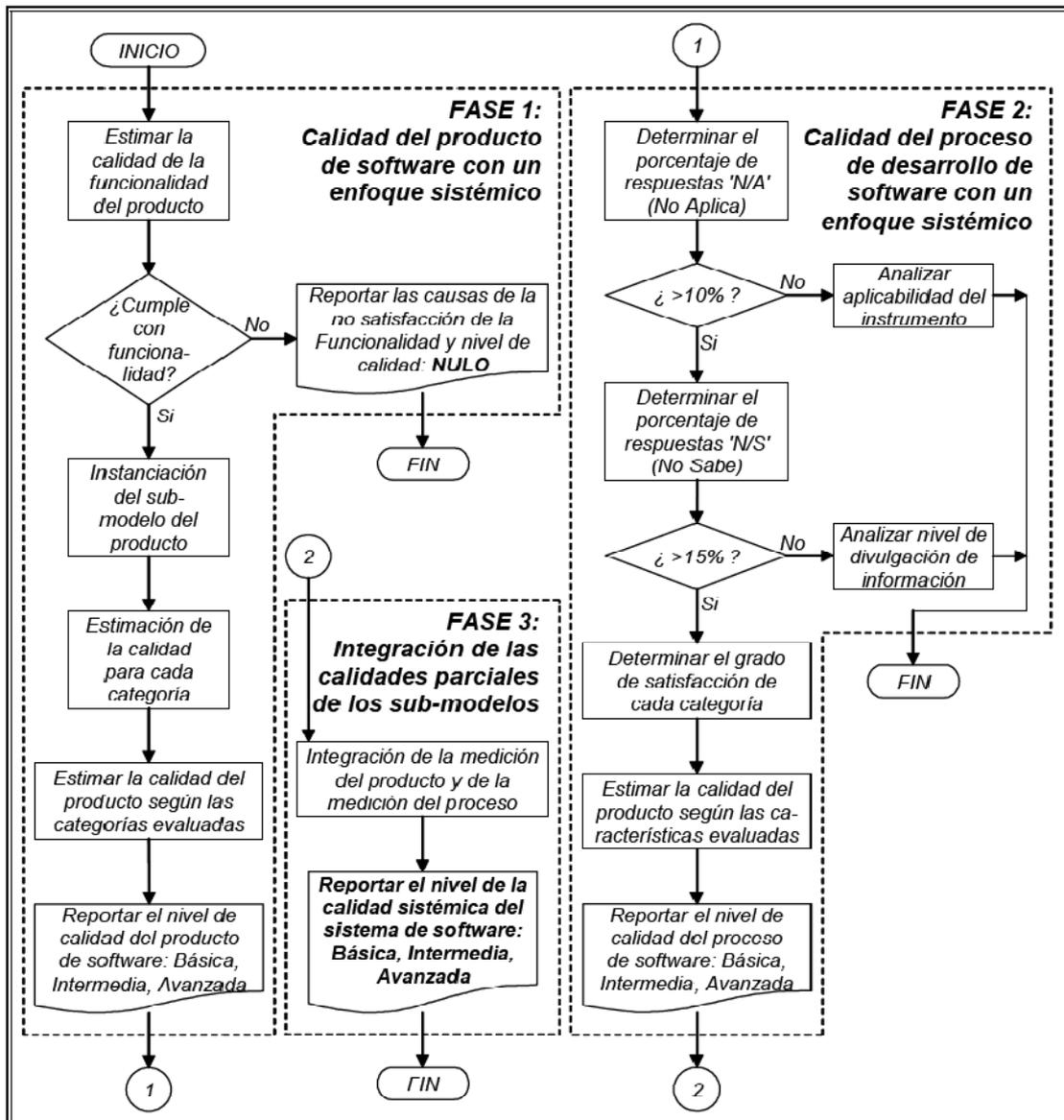
producto cumpla con los requerimientos establecidos, tomando en cuenta las especificaciones del cliente.

Adicionalmente MOSCA dispone de un algoritmo de aplicación del modelo, el cual constituirá una guía para la presente investigación.

### **2.7.1.2 Algoritmo de aplicación del Modelo MOSCA**

En la Figura 09, se puede observar gráficamente los pasos a seguir para medir la calidad sistémica dentro de una organización a través de la aplicación del modelo MOSCA. Para ello, se deberá medir primero la calidad del producto de software y luego la calidad del proceso de desarrollo del mismo. Mendoza et.al (2005).

Tal como se ilustra en la Figura 09, siempre y en todos los casos se debe medir primero la categoría Funcionalidad del producto. Si cumple con todas las características necesarias que se proponen para esta categoría, entonces se deberá proceder a adaptar el submodelo del producto según las especificaciones del cliente. Si el producto no cumple con la categoría Funcionalidad, la evaluación finaliza; es decir, el resto del sub-modelo de producto y el sub-modelo del proceso no deberá ser evaluado. Esto se debe a que la categoría Funcionalidad es la más importante dentro de la medición de la calidad, ya que identifica la capacidad del mismo para cumplir las funciones para las que fue fabricado. Además, como aporte importante, se brinda al cliente las causas del por qué la Funcionalidad no pudo ser satisfecha y el nivel de calidad resultó ser nulo. Mendoza et.al (2005).



**Figura 09:** Algoritmo de aplicación de MOSCA.  
**Fuente:** Universidad Simón Bolívar Venezuela.  
**Autor:** Mendoza, L. y otros (2005).

Ahora bien, para poder medir la calidad del producto de software se presenta la Tabla 12, en la cual se relacionan el nivel de calidad con las categorías satisfechas. En este punto es preciso recordar que si no se satisface la categoría Funcionalidad el algoritmo finaliza y la calidad del producto de software será nula.

Funcionalidad	Segunda categoría evaluada	Tercera categoría evaluada	Nivel de calidad del producto de software
Satisfecha	No Satisfecha	No Satisfecha	Básico
Satisfecha	Satisfecha	No Satisfecha	Intermedio
Satisfecha	No Satisfecha	Satisfecha	Intermedio
Satisfecha	Satisfecha	Satisfecha	Avanzado

**Tabla 12:** Nivel de calidad del producto con respecto a las categorías satisfechas para el producto.

**Fuente:** Universidad Simón Bolívar Venezuela.

**Autor:** Mendoza, L. y otros (2005).

Una vez terminada la evaluación del producto y sólo en caso de que este obtenga al menos un nivel de calidad básico, se procederá a medir la calidad del proceso a través del sub-modelo del mismo. Partiendo de las categorías evaluadas en el sub-modelo del Proceso, se estima la calidad de este según las categorías satisfechas Mendoza et.al (2005):

- (a) **Calidad básica.** Es la mínima calidad requerida. Se satisfacen las características: Cliente-Proveedor e Ingeniería.
- (b) **Calidad intermedia.** Esta no sólo satisface las características de Calidad básica, sino que, además, satisface las características de Soporte y Gestión.
- (c) **Calidad avanzada:** Satisface todas las características.

Por último, se debe realizar una “integración” de la medición del producto y de la medición del proceso para obtener la medición de la calidad sistémica. Los niveles de calidad sistémica se proponen en la Tabla 13 Mendoza et.al (2005)

Nivel de Calidad Producto	Nivel de Calidad Proceso	Calidad Sistémica
Básico	-	Nulo
Básico	Básico	Básico
Intermedio	-	Nulo
Intermedio	Básico	Básico
Avanzado	-	Nulo

Avanzado	Básico	Intermedio
Básico	Intermedio	Básico
Intermedio	Intermedio	Intermedio
Avanzado	Intermedio	Intermedio
Básico	Avanzado	Intermedio
Intermedio	Avanzado	Intermedio
Avanzado	Avanzado	Avanzado

**Tabla 13:** Nivel de Calidad Sistémica Global a partir del nivel de Calidad del Producto.

**Fuente:** Universidad Simón Bolívar Venezuela.

**Autor:** Mendoza, L. y otros (2005).

Como puede observarse en la Tabla 13, esta propuesta obedece a la necesidad de mantener un equilibrio entre las distintas dimensiones de la Calidad de los SS; es por ello que, la Calidad del Producto de Software tiene igual peso que la Calidad del Proceso de Desarrollo de Software. Se considera que la aplicación del modelo permitirá ajustar con mayor precisión este “equilibrio”. Finalmente, se tiene que la integración de las medidas de calidad de los sub-modelos mide la calidad sistémica como una balanza; es decir, si el nivel de calidad de uno de los sub-modelos es menor que el nivel del otro sub-modelo, entonces la balanza no estará estable y por ello se inclinará hacia el nivel de menor calidad. Esto se debe a la sencilla razón de que **si la Calidad del Producto de Software o la Calidad del Proceso de Desarrollo no cumplen con las características necesarias para tener un nivel más alto de calidad, implicará directamente que la calidad sistémica tampoco cumpla con las características necesarias para tener un nivel de calidad superior.**

Los aportes que brindará el modelo MOSCA a la presente investigación constituirán las bases para el diseño del Modelo Sistémico de Calidad (MOSCP), ya que se apoyará en la estructura presentada por MOSCA: Dimensiones, Categorías, Características y Métricas. Además, MOSCA determina los criterios necesarios para garantizar la calidad sistémica del modelo, así como también provee el algoritmo necesario para su evaluación.

Una vez analizado el Modelo MOSCA, se hace necesario exponer los aportes de cada una de las investigaciones realizados al presente trabajo de investigación:

Del modelo CMMI, el creciente interés en brindar ayuda a las organizaciones en mejorar la capacidad de madurez de sus procesos durante todo el ciclo de vida del producto, en cuanto al proceso de pruebas se puede resaltar dos áreas claves: Validación (VAL) y Verificación (VER), estableciéndolas como vitales para garantizar la calidad del producto.

Por otro lado, el aporte del Modelo TMM está relacionado a la incorporación de participantes claves en el proceso de pruebas, se realiza la acotación que es importante incluir directivos, desarrolladores, probadores, clientes/usuarios, es decir, involucrar a todos los interesados en el éxito del proceso. Además, indica la necesidad de la planificación y organización de las pruebas, así como el establecimiento de métricas para evaluar la calidad del software.

El modelo TMMI, menciona criterios cuantitativos para la evaluación de los productos basados en atributos de calidad tales como: fiabilidad, facilidad de uso y mantenimiento, a su vez menciona aspectos relevantes en el proceso de pruebas tales como: Planificación de las pruebas, selección de herramientas de pruebas y Prevención de defectos y control de calidad.

Por su parte, TESTPAI destaca la importancia de cuatro aspectos fundamentales para el proceso de prueba como lo son: Planificación de las pruebas, especificación de casos de pruebas, establecer la infraestructura y analizar, evaluar y comunicar los resultados obtenidos.

Finalmente, el estándar en desarrollo ISO/IEC 29119, establece tres niveles particulares que se deben tener en cuenta: Procesos de la Organización, Procesos de Gestión de Pruebas y Procesos Fundamentales de Pruebas, además señala la importancia de la documentación de las pruebas durante todo el ciclo de vida del proceso de pruebas de software.

Se observa claramente que cada uno de los aportes mencionados anteriormente establecen la base para seleccionar los aspectos relacionados al proceso de pruebas, así como las características de calidad según MOSCA y de esta manera proponer el Modelo sistémico de calidad para el proceso de pruebas de software (MOSCP).

Seguidamente, se presentan las bases teóricas las cuales constituirán los fundamentos de la investigación.

## **2.2 Bases Teóricas**

Hasta el momento se ha presentado distintos trabajos que forman los antecedentes de la presente investigación. En esta sección se tratará un conjunto de conceptos y definiciones que conformarán las bases teóricas para el desarrollo de la actual investigación. Se exponen los aspectos concernientes al proceso de software, así como al proceso de pruebas de software partiendo de la conceptualización, características, objetivos, fases del proceso, tipos, niveles y técnicas de pruebas. Posteriormente, se muestran los tópicos asociados a la calidad del software, revisando aspectos como calidad del producto, calidad del proceso hasta llegar al concepto de calidad sistémica, temas íntimamente relacionados al objetivo del presente proyecto de investigación.

### **2.2.1 Proceso de software**

Para comprender el concepto de Proceso de software, se hace necesario definir en primer lugar proceso. ISO 9000, define *proceso* como un conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados. Por otro lado, SWEBOK (2004) señala que una definición de un proceso puede ser un procedimiento, una política o un estándar.

Seguidamente se define proceso de software tomando en cuenta diversos autores.

Según Fugetta (2000), un proceso de software es un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, instalar y mantener un producto software. El autor señala que existen diversos elementos a abordar, las cuales se agrupan en las siguientes categorías:

- Tecnología de desarrollo de software.
- Métodos y Técnicas de Desarrollo Software.
- Comportamiento Organizacional.
- Economía y Marketing.

Pressman (2005), lo define como un marco de trabajo para las tareas que se requieren en la construcción de software de alta calidad.

Por otro lado Sommerville (2005), define proceso de software como un conjunto de actividades y resultados asociados que producen un producto software. El autor distingue cuatro actividades fundamentales de procesos, las cuales son comunes para todos los procesos de software, dichas actividades se listan a continuación:

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

A su vez SWEBOK (2004), señala que un proceso de ciclo de vida de software se definen por muchas razones, las cuales incluyen: incrementar la calidad del producto, facilitar el entendimiento y la comunicación humana, apoyar la mejora de procesos, apoyar la gestión de los procesos, suministrar una guía automatizada para los procesos y suministrar un apoyo para ejecuciones automatizadas.

Evidentemente, un proceso de software se beneficia de las tecnologías de desarrollo de software, los procedimientos y técnicas de desarrollo de software y o

el comportamiento organizacional. Para efectos de la presente investigación se usará el siguiente concepto de proceso de software:

El proceso de software, es un marco de trabajo (Pressman, 2005), conformado por cuatro actividades fundamentales: Especificación del software, desarrollo del software, validación del software, evolución del software (Sommerville, 2005), las cuales deben tomar en cuenta la interrelación entre diversos elementos: Tecnologías de desarrollo de software, métodos y técnicas de desarrollo software, comportamiento organizacional, economía y marketing (Fugetta, 2000), para la construcción de software de alta calidad, así como facilitar la comunicación y la mejora de procesos. SWEBOK (2004).

### **2.2.2 Modelo de Procesos**

En primera instancia se define *modelo*, Para Gomes (2002), citado por León (2006), la representación de un sistema en cualquier forma distinta a la propia entidad se llama Modelo.

Con miras al logro del objetivo de la presente investigación, se hace necesario definir modelo de procesos tomando en cuenta el aporte de diferentes autores.

SWEBOK (2004) asegura que los, *modelo de procesos*, sirven como definiciones de alto nivel de las fases que tienen lugar durante el desarrollo. No están enfocados a ofrecer definiciones detalladas sino más bien a sobresaltar las actividades clave y sus interdependencias.

Por otro lado, Pressman (2005), señala que un modelo de procesos, es un marco de trabajo genérico para el proceso, el cual incluye las siguientes actividades dentro del marco: comunicación, planeación, modelado, construcción y desarrollo.

En el mismo orden de ideas, Sommerville (2005), lo define de la siguiente manera, es una representación abstracta de un proceso de software. Cada modelo de proceso, representa un proceso desde una perspectiva particular, y así proporciona sólo información parcial sobre ese proceso.

Recientemente, García et.al (2010), lo definen de la siguiente manera: El modelo de procesos, define un catálogo o colección estructurada de buenas prácticas que describen las características de un proceso efectivo.

De lo anteriormente expuesto, se observa que definitivamente la importancia de los modelos y específicamente de modelo de procesos en el área del software, radica en representar la interrelación existente entre los diversos elementos que intervienen en el proceso de software modelado, así como las actividades claves, permitiendo de esta manera el estudio de las mismas, para lograr un conocimiento completo del proceso.

Seguidamente, se procede a conceptualizar las pruebas de software, el cual es un punto relevante de acuerdo al tema de investigación planteado.

### **2.2.3 Pruebas de Software**

Al hablar de las pruebas de software debe considerarse un tema más amplio que suele denominarse Verificación y Validación, ya que éstas son consideradas como un elemento de dichos procesos.

La *Verificación*, es el conjunto de actividades que aseguran que el software implemente correctamente una función específica. Por otro lado la *Validación*, es un conjunto de diferentes actividades que aseguran que el software construido corresponde con los requisitos del cliente. Pressman (2005).

Con respecto a este punto, Sommerville (2005) indica, que el papel de la verificación implica comprobar que el software está de acuerdo con su especificación. La Validación, sin embargo, es un proceso más general, el

objetivo de la validación es asegurar que el sistema software satisface las expectativas del cliente. Así mismo, el autor señala que el objetivo último del proceso de Verificación y Validación, es establecer la seguridad de que el sistema software está hecho para un propósito.

Por su parte, Polo (2006), señala que el proceso de Verificación intenta determinar si los productos software de una actividad se ajustan a los requisitos o a las condiciones impuestas en actividades anteriores. Así mismo, señala que el objetivo del proceso de Validación, es determinar si los requisitos y el sistema final cumplen con los objetivos para lo que se construyó el producto.

Por otro lado, la guía para la certificación CMMI (2009), expone lo siguiente, el propósito de la Validación (VAL), es demostrar que un producto o componente de producto se ajusta a su uso previsto cuando se sitúa en su entorno previsto. A su vez, señala que, el propósito de la Verificación (VER), es asegurar que los productos de trabajo seleccionados cumplen sus requerimientos especificados.

De lo anteriormente expuesto, se observa la importancia de los procesos de y Validación y Verificación; por un lado el proceso de Verificación, señala la importancia de comprobar cada uno de los productos que se van construyendo, bajo la premisa que si lo que se está construyendo es correcto, el producto software final cumplirá con los requerimientos. Del mismo modo, el proceso de Validación, valora comprobar el cumplimiento de los objetivos del producto software desarrollado, ajustado a los requerimientos del cliente.

Una vez expuestos los conceptos relacionados a los procesos de Verificación y Validación, según la óptica de diversos autores, se considera las *pruebas del software* como una aproximación de dichos procesos. Por lo cual, se hace necesario conceptualizar, desde el punto de vista de diversos autores.

Según lo señalado por Sommerville (2005), son una técnica dinámica de Verificación y Validación, las mismas implican ejecutar una implementación del

software con datos de prueba, se examinan las salidas y su entorno operacional para comprobar que funciona tal y como se requiere.

En el mismo orden de ideas, SWEBOOK (2004), define prueba de software desde el punto de vista dinámico de la siguiente manera, consisten en verificar dinámicamente a través de un grupo finito de casos de prueba el comportamiento de un programa contra el comportamiento esperado, seleccionados de manera adecuada desde el dominio infinito de ejecución.

Fernández (2005), señala que las pruebas del software se definen como una actividad en el cual un sistema o uno de sus componentes se ejecuta en circunstancias previamente especificadas, los resultados se observan y registran y se realiza una evaluación de algún aspecto.

En este mismo sentido, Pressman (2005), define las pruebas de software como un conjunto de actividades que se planean con anticipación y se realizan de manera sistemática.

De igual manera, Pérez (2006), señala que las pruebas, son una actividad en la que un sistema o componente es ejecutado bajo condiciones especificadas, los resultados son observados o registrados y una evaluación es hecha de algún aspecto del sistema o componente.

Por otro lado, Casallas et al. (2007), asegura que sin importar cual metodología sea utilizada por un equipo de desarrollo, las pruebas realizadas sobre el producto, ya sean unitarias, de integración o de sistema, reflejan si los objetivos planteados a nivel de requerimientos son satisfechos por el producto desarrollado.

Por su parte Calvo et al. (2009) plantea lo siguiente, son un elemento imprescindible y crítico para la validación de un producto o de un componente de producto de software. Sin embargo, se tiende a confundir prueba de software con validación, siendo la prueba un factor primordial del proceso de validación. Las pruebas son las últimas barreras para detectar errores, puesto que permiten encontrar fallos como consecuencia de defectos ocasionados por errores.

Para efectos de esta investigación, se utilizará el siguiente concepto de pruebas de software:

Las pruebas de software, son una técnica dinámica de verificación y validación (Sommerville, 2005), las cuales comprenden un conjunto de actividades que deben ser planeadas y ejecutadas de manera sistemática (Pressman, 2005), donde el sistema o componente es ejecutado bajo condiciones específicas, a través de un grupo finito de casos de pruebas (SWEBOK, 2004), los resultados son observados o registrados y una evaluación es hecha de algún aspecto del sistema o componente. (Pérez, 2006).

La definición antes descrita, servirá de punto de partida para el presente trabajo de investigación, no obstante, en secciones posteriores, se explicará en detalle los objetivos de las pruebas de software, características, así como los artefactos involucrados.

Sin duda alguna, las pruebas del software constituyen un proceso fundamental en el aseguramiento de la calidad del producto elaborado, es por ello que se hace necesario mencionar los objetivos de las mismas de acuerdo a opiniones coincidentes de distintos autores.

### **2.2.3.1 Objetivos de las pruebas de Software**

- (a) SWEBOK (2004), indica que el proceso de pruebas es una actividad que tiene el objetivo de evaluar y mejorar la calidad del producto, identificando defectos y problemas.
- (b) Somerville (2005), señala que generalmente el objetivo de las pruebas del software es convencer a los desarrolladores del sistema y a los clientes de que el software es lo suficientemente bueno para su uso operacional.
- (c) Uno de los objetivos de la fase de pruebas del sistema es verificar que el comportamiento externo del sistema software satisface los requisitos establecidos por los clientes y futuros usuarios del mismo Gutiérrez et al. (2006).

(d) El propósito general de las pruebas es identificar los errores que se encuentran en el código y comprobar el correcto funcionamiento de los diferentes componentes que forman el sistema, y el sistema como tal. Sanz et al. (2008).

(e) Determinar que el producto esté libre de defectos y que su funcionamiento es el esperado. Calvo et al. (2009).

Ahora bien, de acuerdo con los planteamientos anteriores se observa que el propósito de las pruebas de software es identificar defectos en el código y comprobar el correcto funcionamiento del software elaborado de acuerdo a los requisitos establecidos por el cliente, mejorando la calidad del producto.

Debido al creciente interés en el proceso de pruebas como un mecanismo de aseguramiento de la calidad del software, es importante discutir las principales características.

### **2.2.3.2 Características de las pruebas**

Para cumplir con el objetivo de las pruebas de software, éstas deben poseer un conjunto de características, las cuales se describen a continuación según la perspectiva de diversos autores.

Según Kaner et.al citado por Pressman (2005), sugieren los siguientes atributos para una buena prueba.

- Tiene una elevada probabilidad de encontrar un error. Alcanzar este objetivo requiere que la persona que aplica la prueba comprenda el software y trate de desarrollar una imagen mental de la manera en que puede fallar.
- No es redundante, el tiempo y los recursos destinados a las pruebas son limitados. No hay razón para realizar una prueba que tenga el mismo propósito que otra. Cada prueba debe tener un propósito diferente.

- Debe ser la mejor de su clase, en un grupo de pruebas con un objetivo similar y recursos limitados podría optarse por la ejecución de un sólo subconjunto de ellas.
- No debe ser ni muy simple ni demasiado compleja, aunque a veces es posible combinar una serie de pruebas en un caso de prueba, los posibles efectos colaterales asociados con este enfoque podrían enmascarar errores. En general, cada prueba debe ejecutarse por separado.

En el mismo orden de ideas, SWEBOK (2004), respecto a las características de las pruebas señala lo siguiente:

- Presentes durante todo el proceso de desarrollo y mantenimiento.
- Dinámicas, para la ejecución dinámica del programa, se utilizan datos de entrada.
- Finitas, ya que realizar pruebas exhaustivas podría llevarse meses y años.
- Selectivas, es necesario identificar un criterio de selección de pruebas apropiado para un conjunto de condiciones particulares de un problema complejo.
- Comprobables, ya que se debe comparar los resultados obtenidos con los resultados esperados por el usuario.

En este mismo sentido, Gutiérrez (2006), señala lo siguiente:

- Constan de tres elementos: interacciones entre el sistema y la prueba, valores de prueba y resultados esperados.
- Está constituida generalmente de cuatro fases: Diseño de pruebas, codificación, ejecución y análisis de resultados.

Por otro lado, Prado (2007), señala lo siguiente:

- Las pruebas deben estar presentes a lo largo de todo el ciclo de desarrollo.
- Probar exhaustivamente el software es imposible.
- No es recomendable que el programador pruebe sus propios programas.
- Se puede disponer de diversas herramientas.

Observe que las características de las pruebas de software propuestas por los diferentes autores se centran en los siguientes aspectos: Las pruebas de software deben estar presentes a lo largo de todo el ciclo de desarrollo, son dinámicas, poseen elementos esenciales, entre los cuales destacan: Interacciones entre el sistema y la prueba, valores de prueba y resultados esperados; además es importante establecer criterios de selección al momento de ejecutar las pruebas, es decir, las pruebas no deben ser redundantes ni demasiado complejas. Seguidamente, se definen los principales artefactos o documentos involucrados en el proceso de pruebas.

### **2.2.3.3 Artefactos de las Pruebas**

#### **(a) Plan de Pruebas**

Según Prado (2007), el propósito del plan de pruebas es explicitar el alcance, enfoque, recursos requeridos, calendario, responsables y la gestión de riesgos de un proceso de pruebas.

Para la definición del plan de pruebas, se debe valorar los siguientes aspectos:

- El alcance de la aplicación.
- La complejidad de sus procesos.
- Plataforma(s) en la que se debe probar.
- Conocimientos y formación de quienes ejecutarán las pruebas.
- Normativas legales aplicables.

En el mismo orden de ideas, Casallas et.al (2007) señala que un plan de pruebas es la unidad básica de ejecución dentro del sistema de pruebas y está constituido por un conjunto de pruebas y ejecuciones.

### (b) Caso de Prueba

Jacobson (2004) citado por Chávez et.al (2009), señala que un caso de prueba específica una forma de probar el sistema, incluyendo la entrada o resultado con la que se ha de probar y las condiciones bajo las que ha de probarse.

En el mismo sentido [IEEE 610.12-1990] citado por Pérez(2006), afirma que, es un conjunto de valores de entrada, precondiciones de ejecución, resultados esperados y poscondiciones de ejecución, desarrollados con un objetivo particular o condición de prueba, tal como ejercitar un camino de un programa particular o para verificar que se cumple un requerimiento específico.

Por su parte Leffingwell et al. (2006),citado por Méndez et.al (2007)<sup>b</sup>, señalan que un caso de prueba es una especificación, usualmente formal, de un conjunto de entradas de prueba, condiciones de ejecución y resultados esperados, identificados con el propósito de hacer una evaluación de aspectos particulares de un elemento objeto de prueba.

En este orden de ideas, Méndez et.al(2007)<sup>b</sup>, proponen un modelo conceptual asociado a los casos de prueba, el cual se puede observar en la Figura 10.



**Figura 10:** Modelo Conceptual asociado a los Casos de Prueba.

**Fuente:** Actas de Talleres de Ingeniería del Software y Bases de Datos, Vol. 1, No. 4

**Autor:** Méndez y otros (2007)<sup>b</sup>

Se puede indicar entonces, que los casos de prueba son esenciales para todas las actividades de pruebas, y están íntimamente relacionados al proceso de pruebas debido a lo siguiente:

- Son la base para diseñar y ejecutar los procedimientos de pruebas.
- La profundidad de las pruebas es proporcional al número de casos de pruebas.
- El diseño y desarrollo, y los recursos necesarios son gobernados por los casos de pruebas requeridos.

#### **(c) Procedimientos de Prueba**

Definidos en Jacobson (2004) citado por Chávez et.al (2009), donde se dice que éstos especifican como realizar varios casos de prueba o parte de estos. A su vez, Pérez (2006), indica que los procedimientos de prueba, son instrucciones detalladas para la configuración, ejecución y evaluación de los resultados para un caso de prueba determinado.

#### **(d) Suite de pruebas**

Pérez (2006), lo define de la siguiente manera, es un conjunto de una o más pruebas, con un propósito y base de datos común que usualmente se ejecutan en conjunto.

#### **(e) Script de Pruebas**

Según lo señalado en Beizer (1990), citado por Pérez (2006), es un programa que especifica para la prueba: elemento a ser probado, requerimiento, estado inicial, entradas, resultados esperados y criterio de validación.

#### (f) Resultado Esperado

Según el International Software Testing Qualifications Board(2005),es el comportamiento predicho por la especificación u otra fuente, del componente o sistema a ser probado bajo condiciones especificadas.

#### (g) Resultado Obtenido

Es el comportamiento producido u observado cuando un componente o sistema es probado. International Software Testing Qualifications Board(2005).

Como se puede observar, los artefactos o documentos de las pruebas, se deben tener en cuenta a lo largo de todo el proceso de las pruebas de software, como un mecanismo de control para garantizar el éxito, el cual se reflejará en la eficiencia del proceso ejecutado y posteriormente en la calidad del producto obtenido.

En la siguiente sección se procede a conceptualizar el proceso de pruebas, así como la visión de diversos autores para la elaboración de un proceso de pruebas.

#### 2.2.3.4 El Proceso de Pruebas

Es importante delimitar el concepto de Proceso de pruebas que se adoptará en la presente investigación, ya que diversos autores lo definen de diversas formas.

En la Tabla 14, se pueden visualizar las diferentes definiciones que los autores han dado al proceso de pruebas de software.

Autor(Año)	Definición
Krutchén, Pressman, Pfléger, Cardoso y Sommerville citado por Grimán et al. (2003)	El proceso de ejecución de pruebas debe ser considerado durante todo el ciclo de vida de un proyecto, para así obtener un producto de alta calidad. Su éxito dependerá de la estrategia de prueba adecuada.
Swebok(2004)	Es un proceso definido y controlado, que debe ser gestionado por personas. El proceso de pruebas soporta actividades y sirve de guía a los equipos de prueba, desde la planificación hasta la evaluación de los resultados, de tal manera que se puede proporcionar una garantía justificada de que los objetivos de las pruebas se conseguirán de manera económica. Además señala las siguientes actividades: <ul style="list-style-type: none"><li data-bbox="890 1980 1086 2002">• Planificación.</li></ul>

	<ul style="list-style-type: none"> <li>• Generación de casos de prueba.</li> <li>• Desarrollo del entorno de pruebas.</li> <li>• Ejecución.</li> <li>• Evaluación de los resultados.</li> <li>• Notificación de problemas/diario de las pruebas.</li> <li>• Seguimiento de los defectos.</li> </ul>
Sommerville(2005)	<p>Señala un modelo general del proceso de pruebas, en el que se puede apreciar las siguientes fases con sus respectivos artefactos:</p> <ul style="list-style-type: none"> <li>• Diseñar casos de Prueba: Casos de Prueba.</li> <li>• Preparar datos de Prueba: datos de prueba.</li> <li>• Ejecutar el programa con los datos de prueba: Resultados de la prueba.</li> <li>• Comparar resultados con los datos de prueba: Informe de la prueba.</li> </ul>
Testing Qualifications(ISTQB, por sus siglas en Inglés), citado por Pérez (2006)	<p>El proceso de pruebas está constituido de las siguientes actividades principales:</p> <ul style="list-style-type: none"> <li>• Planificación y Control: se definen los objetivos de la prueba y las actividades de prueba, así como comparar el estado real con respecto a lo planificado y reportar el estado.</li> <li>• Análisis y Diseño de las Pruebas: Los objetivos generales se transforman en pruebas tangibles y diseño de las pruebas.</li> <li>• Implementación y Ejecución: Se ejecutan las pruebas, a su vez, se registran los resultados para compararlos con los resultados esperados.</li> <li>• Evaluar el criterio de terminación y Reportes: En esta fase, se evalúan las pruebas contra los objetivos definidos, además se elaboran los reportes correspondientes.</li> <li>• Actividades de Cierre: Almacenan los datos de las actividades realizadas para consolidar experiencia.</li> </ul>
Casallas et al. (2007)	<p>El proceso de pruebas involucra el diseño e implementación, su ejecución, el reporte de los defectos encontrados, la planeación de las correcciones y la implementación de dichas correcciones.</p>
Sanz et al. (2008)	<p>El proceso de pruebas comienza con la fase de definición de requisitos y se desarrollará de forma paralela al proceso de desarrollo de software. Además indica que, un proceso de pruebas bien definido implica:</p> <ul style="list-style-type: none"> <li>• Identificar y establecer los objetivos, políticas y estrategia de prueba que</li> </ul>

	<p>guiaran al proceso.</p> <ul style="list-style-type: none"> <li>• Desarrollar y mantener un plan de pruebas. Se debe definir las especificaciones necesarias para generar los casos de prueba adecuados.</li> <li>• Analizar e informar de los resultados obtenidos durante la ejecución de las pruebas.</li> <li>• Identificar y establecer las medidas adecuadas para solucionar los posibles problemas encontrados.</li> </ul>
Myers (2004) citado por Chávez et.al( 2009)	El proceso de prueba de software, es la búsqueda de fallos mediante la ejecución del sistema o de partes del mismo en un entorno y con valores de entrada bien definidos.

**Tabla 14:** Definiciones de Proceso de Pruebas de Software.

**Fuente:** Elaboración Propia.

**Autor:** Autor de la investigación (2012).

Para efectos de la presente investigación se usará el siguiente concepto de Proceso de pruebas de software:

El proceso de pruebas de software, es la búsqueda de fallos mediante la ejecución del sistema Myers (2004) citado por Chávez et.al( 2009), siguiendo una serie de actividades o fases: Planificación y Control, Análisis y Diseño de las Pruebas, Implementación y Ejecución, Evaluar el criterio de terminación y Reportes, Actividades de Cierre Testing Qualifications(ISTQB, por sus siglas en Inglés), citado por Pérez (2006), cada una de las cuales producirá un artefacto o documento como(Casos de prueba, datos de pruebas, resultados e informe de las pruebas) Sommerville (2005). El proceso de pruebas, es un proceso definido y controlado, gestionado por personas SWEBOK (2004), debe ser considerado durante todo el ciclo de vida de un proyecto, para así obtener un producto de alta calidad. Krutchen, Pressman, Pfleger, Cardoso y Sommerville citado por Grimán et al. (2003).

Es evidente entonces, que la complejidad intrínseca del software, causa que los procesos de pruebas sean determinantes para aumentar la calidad del producto final; análogamente, la calidad de las pruebas de software vendrá determinada por la calidad del proceso de pruebas que se utilice.

Así, independientemente de la estrategia de prueba a utilizar se debe considerar la planeación de las pruebas de software en su totalidad, iniciando con el plan de pruebas, efectuar un correcto diseño de los casos de pruebas, ejecutar las pruebas adecuadas y ajustadas al plan de pruebas, informar las incidencias que se produzcan, evaluar los resultados obtenidos; todas estas fases deben considerar la concordancia con los requisitos del sistema con miras al aseguramiento de la calidad tanto a nivel del producto como a nivel del proceso que se está implementando.

En relación al tema de investigación planteado se procede a indagar en los conceptos relacionados a calidad del software y calidad del software desde el punto de vista sistémico.

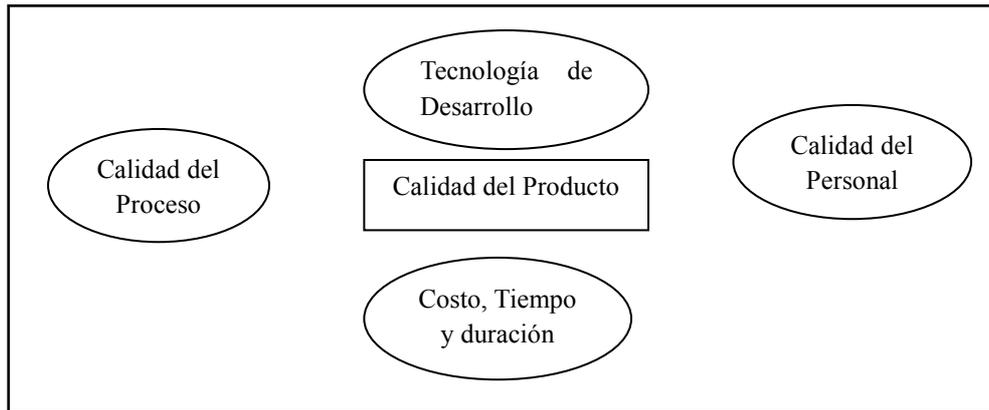
#### **2.2.4 Calidad del Software**

Pressman (2005), define *calidad de software* de la siguiente manera, concordancia con los requisitos funcionales y de desempeños explícitamente establecidos, estándares de desarrollo explícitamente documentados y características implícitas que se esperan de cualquier software desarrollado profesionalmente.

La calidad según ISO/IEC 9126, es la totalidad de las características de una entidad que refieren su capacidad de satisfacer necesidades establecidas o implícitas.

En este sentido, a la hora de definir la calidad del software se debe diferenciar entre la calidad del producto software y la calidad del proceso de desarrollo de éste.

En cuanto a la calidad del producto Sommerville (2005), indica que existen cuatro factores que afectan la calidad en este aspecto, los cuales se pueden observar en la Figura 11.



**Figura 11:** Principales factores de calidad del producto software.

**Fuente:** Ingeniería del Software: séptima edición.

**Autor:** Sommerville, Ian (2005).

La calidad del producto se ve afectada si un proyecto, independientemente de su tamaño, está mal presupuestado o planificado con un tiempo de entrega irreal. Un buen proceso requiere recursos para su implementación efectiva. Si estos recursos son insuficientes, el proceso no puede desarrollarse de forma efectiva. Si los recursos no son adecuados, sólo las personas excelentes pueden salvar el proyecto. Más aún, si el déficit es demasiado grande, la calidad del producto se degradará.

No obstante, las metas que se establezcan para la calidad del producto van a determinar los objetivos del proceso de desarrollo, ya que la calidad del primero va a depender, entre otros aspectos de éstos. Sin un buen proceso de desarrollo es casi imposible obtener un buen producto.

En este sentido, Callaos y Callaos (1996), citado por Mendoza et.al (2005), proponen un concepto de calidad del software en el cual están involucrados tanto las características internas del producto como el contexto organizacional, lo que genera un enfoque sistémico del concepto de calidad del software.

En esta investigación, se hará énfasis en la definición de calidad sistémica en el desarrollo de los sistemas de software, el cual contempla la matriz de Calidad Global Sistémica, mostrada en la Figura 12, la cual consta de cuatro tipos de calidad: Eficiencia del Producto, Efectividad del Producto, Eficiencia del Proceso y Efectividad del Proceso, considerando las dimensiones del cliente y del usuario.

Esta división se justifica en un sentido, porque un proyecto incluye tanto la eficiencia como la efectividad y en el otro, porque el Sistema concebido (el producto) es diferente al Sistema de las actividades humanas (el proceso), mediante el cual el Sistema-Producto es diseñado. Callaos y Callaos (1996), citado por Mendoza et. al (2005).



**Figura 12:** Matriz de Calidad Global Sistémica.

**Fuente:** Universidad Simón Bolívar Venezuela.

**Autor:** Callaos y Callaos (1991) citado por Mendoza, L. y otros (2005).

Rojas et.al (1995) citado por Mendoza et.al (2005), definen cada uno de los componentes de la matriz de la siguiente manera:

- **Eficiencia del Producto:** Es determinada por actividades de diseño interno y programación, ya que un producto eficiente es conseguido cuando se aplican las prácticas correctas de diseño físico y programación.
- **Efectividad del Producto:** Es determinada por las actividades de identificación de requerimientos, diseño de interfaces y diseño general de la red (ubicación de puntos), debido a que la misma está relacionada con la adecuación y confort del usuario.
- **Eficiencia del Proceso:** Está asociada con las actividades de gerencia de proyectos, las cuales incluyen el cumplimiento de fechas de entrega, aumento de la productividad y ahorro de recursos.
- **Efectividad del Proceso:** Se relaciona con las actividades generales de gerencia, tales como liderazgo, administración de cambio, relaciones

humanas y grupales, ya que las mismas conducen a establecer buenas relaciones entre los integrantes del equipo responsable del desarrollo de Sistemas de Información.

Según Callaos (1996) citado por Mendoza et. al (2005), la calidad global no es la suma de las calidades parciales, sino el compromiso entre todo el conjunto de calidades que conlleve un óptimo global con cierto sacrificio de los óptimos parciales. En consecuencia, se tiene que la calidad del software no es algo que depende de una sola característica en particular, sino que obedece al compromiso de todas sus partes.

Como resultado de la investigación realizada por Méndez et.al (2007)<sup>a</sup> se comprobó la relación existente entre la calidad sistémica y el aumento de la productividad en las organizaciones, ya que al incrementar la calidad del proceso de desarrollo se mejoran los resultados obtenidos en la productividad; ya que la organización como un ente abierto, no puede alcanzar niveles óptimos de calidad en sus productos de software y en sus procesos de desarrollo sin tomar en cuenta factores internos y externos que influyen en todo el sistema.

En el contexto del tema de estudio es importante indagar si el mejoramiento de las pruebas del software aporta algún beneficio a las organizaciones.

### **2.3 Bases Legales**

La Constitución de la República Bolivariana de Venezuela (1999), indica en su artículo 98, lo siguiente:

La creación cultural es libre. Esta libertad comprende el derecho a la inversión, producción y divulgación de la obra creativa, científica, tecnológica y humanística incluyendo la protección legal de los derechos del autor o de la autora sobre sus obras. El estado reconocerá y protegerá la propiedad intelectual sobre las obras científicas, literarias y artísticas, invenciones, innovaciones, denominaciones, patentes, marcas y lemas de acuerdo con las condiciones y excepciones

que establezcan la ley y los tratados internacionales suscritos y ratificados por la República en esta materia. (p. 81)

En el artículo anterior queda expresado la libertad y protección que tiene el autor sobre sus creaciones, dando de esta forma un respaldo a la propuesta de investigación planteada.

Así mismo, la Ley Orgánica de Ciencia Tecnología e Innovación (2005) nos indica en su artículo numero 2, “Las actividades científicas, tecnológicas, de innovación y sus aplicaciones son de interés público y de interés general.” (p. 1), con lo cual se asevera aun más el compromiso del estado con las actividades tecnológicas, brindando de esta manera más apoyo al desarrollo intelectual de los individuos en la sociedad que pertenecen.

A continuación se procederá a describir el Marco Metodológico, en el cual se expone el tipo de investigación, se realiza el estudio de la factibilidad y se procede a detallar el procedimiento de la investigación.

## CAPÍTULO III

### MARCO METODOLÓGICO

Todo estudio requiere de la definición de la metodología a ser empleada para su desarrollo, debido a que se considera que es la base de todo proyecto de investigación y exponer los métodos, técnicas y procedimientos empleados, de forma tal, que se pueda mostrar una visión más amplia de la investigación que se elaboró, todo esto constituye la validez de la investigación.

#### 3.1 Tipo de Investigación

El presente trabajo se encuentra enmarcado en la modalidad de Investigación Cualitativa tal como lo señala el Manual para la Presentación de Trabajo de Grado UCLA (2002); ya que un modelo sistémico de calidad para el proceso de pruebas, permite abordar el escenario de pruebas como un todo, es decir, desde una perspectiva holística. Adicionalmente, se encuentra apoyado en el tipo de diseño de Investigación-Acción. Según el Manual para la Presentación de Trabajo de Grado de la Universidad Centroccidental Lisandro Alvarado UCLA (2002), se entiende por Investigación Cualitativa:

Aquella que permite abordar problemas de investigación de manera inductiva, holística y tiene una intervención interpretativa de la experiencia humana.

La investigación cualitativa comprende diversos tipos y diseños de investigación entre ellos se encuentran: Investigación-Acción, etnografía, etnometodología, interaccionismo simbólico, fenomenología, u otros enfoques emergentes. Estos diseños de investigación se caracterizan por ser abiertos, flexibles y dialecticos. (p. 3).

Con relación al diseño tipo Investigación-Acción, Baskerville (1999), identifica cinco fases las cuales interactúan entre sí, las cuales se pueden observar en la Tabla 15.

Fase	Acciones
Diagnosticar	Corresponde a la identificación de los principales problemas que atañen a la organización y que motivan su deseo de cambiar. Envuelve todo lo referente a la interpretación del problema complejo de la organización de forma holística.
Planificar la Acción	Esta actividad especifica las acciones organizacionales que deberían tomarse para relevar o mejorar los problemas detectados al diagnosticar. El descubrimiento de los planes de acción es guiado por el marco teórico, el cual indica el estado futuro deseado por la organización, y los cambios requeridos para alcanzar ese estado. El plan establece el objetivo del cambio y el enfoque para cambiar.
Tomar la Acción	Implementa el plan de acción. Los investigadores y los participantes colaboran en la intervención activa dentro de la organización cliente, provocando ciertos cambios. Diversas estrategias de intervención pueden ser adoptadas: Directiva(los investigadores dirigen el cambio), no directiva y táctica.
Evaluar	Una vez completada las acciones, los investigadores y demás participantes evalúan la salida. La evaluación incluye determinar si los efectos teóricos de la acción fueron alcanzados, y si estos efectos relevaron a los problemas. Si los cambios fueron exitosos, la evaluación se pregunta si los cambios propuestos fueron los únicos causantes de este éxito. Si los cambios no fueron exitosos, es necesario establecer un marco para la próxima iteración del ciclo investigación acción.
Especificar el Aprendizaje	A partir de los resultados de la investigación los investigadores especifican el conocimiento adquirido.

**Tabla 15:** Fases del Método Investigación-Acción.

**Fuente:** Volume 2 Issue 3es, Nov. 1999 Association for Information Systems Atlanta, GA, USA.

**Autor:** Baskerville (1999).

El modelo propuesto se tomará como referencia para valorar la calidad sistémica del proceso de pruebas de software, con los resultados obtenidos se evaluará el nivel de calidad sistémico en el proceso de pruebas que ejecuta una

determinada organización, y en base a éstos, se especificaran los procesos que deben ser mejorados en la organización y las características de calidad que no son satisfechas por el producto de software desarrollado.

### **3.2 Fases del Estudio**

Debido a la modalidad de investigación en la cual está enmarcado el proyecto, se presentará las fases de estudio, con la finalidad de detallar las etapas en el proceso de elaboración del modelo propuesto.

Es importante destacar que para lograr el diseño del Modelo Sistémico de Calidad para el proceso de pruebas (MOSCP) enmarcado en MOSCA, se utilizará el *framework*<sup>2</sup> metodológico del Laboratorio de Investigación en Sistemas de Información (LISI) propuesto por Pérez et.al (2004), el cual se inspira en el método Investigación-Acción y utiliza la metodología DESMET para la evaluación, dichos métodos se detallaran en secciones posteriores.

#### **3.2.1 Fase Diagnóstica**

En esta fase se recolectó toda la documentación e información relacionada con el objeto de la investigación, en tal sentido la revisión bibliográfica permitió conocer los conceptos, objetivos e importancia del proceso de pruebas de software como un elemento clave en el aseguramiento de la calidad. Asimismo, se realizó un análisis de los modelos y estándares de madurez para el proceso de pruebas existentes, identificando las características principales, así como las ventajas y desventajas de los mismos, de acuerdo con los conceptos relacionados a la teoría sistémica y el estudio del Modelo Sistémico de Calidad del Software (MOSCA), se propone diseñar un Modelo Sistémico de Calidad del Proceso de Pruebas de Software.

---

*framework: una estructura conceptual y tecnológica, formada por un conjunto de bloques predefinidos de software, cuya utilización permite la organización y el desarrollo de proyectos software de forma mucho más ágil. Morcillo, Eduardo(2010)*

### **3.2.2 Población y Muestra**

Para garantizar el éxito de un proyecto de investigación se debe considerar la elección de la población y la muestra acorde a los objetivos de la investigación.

En este sentido se hace necesario definir dichos conceptos, La población, según Hernández, y otros (2000), es el conjunto de todos los casos que concuerdan con una serie de especificaciones. En otras palabras, cuando se tiene definido cual es la situación que se va analizar es donde se procede a delimitar la población para así estudiar y darle origen a los datos de la investigación. Con respecto a la muestra, suele ser definida como un subgrupo de la población. Hernández, y otros (2000).

En el caso del presente proyecto, se plantea un enfoque sistémico, por lo cual es importante considerar, tanto aspectos relacionados al producto así como aspectos relacionados al proceso, por esta razón, se considera necesario, elegir como población personal de dos empresas desarrolladoras de software y como muestra tomar en consideración los recursos involucrados, en el producto: Líder del Proyecto, Desarrolladores, Líder de Pruebas, Analista de Pruebas, Clientes; así como los recursos involucrados en el proceso de pruebas: Líder del Proyecto, Desarrolladores, Líder de Pruebas, Analista de Pruebas.

### **3.2.3 Técnicas de Recolección de Datos**

Las técnicas de recolección de información permiten la expresión operativa del diseño de investigación y la especificación concreta de cómo se hará la investigación. Al respecto, Salazar y Trejo (2004) definen los instrumentos de recolección de información como: recursos que utiliza el investigador para registrar información o datos sobre las variables que tiene en mente. De allí que, las técnicas que permitirán al investigador diseñar instrumentos de recolección de información en esta investigación se presentan a continuación.

- Revisión Bibliográfica, es el proceso a través del cual se indaga en diversas fuentes de información existentes sobre el tema, lo cual permite obtener toda la base teórica relacionada necesaria para la comprensión del problema objeto de la investigación.

- Cuestionarios, en opinión de Tamayo (2003), es un cuestionario que se lee al interrogado, contiene una serie de ítems o preguntas estructuradas formuladas y llenadas por un empadronador frente a quien responde.

En la presente investigación, se aplicó un cuestionario destinado al proceso de pruebas de software, y dirigido al personal involucrado en el mismo, a saber: Líder de Proyecto, Desarrollador, Líder de Pruebas, Analista de Pruebas.

### **3.2.4 Estudio de Factibilidad**

#### **3.2.4.1 Factibilidad Técnica**

Realizar la propuesta del modelo sistémico de calidad para el proceso de pruebas de Software (MOSCP), implica examinar los modelos y estándares relacionados a la calidad en el proceso de pruebas, así como las definiciones, características y marco de trabajo del proceso de pruebas de software. Además, se hace necesario revisar los conceptos involucrados en la calidad del software, bajo el enfoque sistémico, relacionado a este concepto, es importante analizar y describir el Modelo Sistémico de Calidad (MOSCA) del Software, ya que será el marco de trabajo para realizar la propuesta.

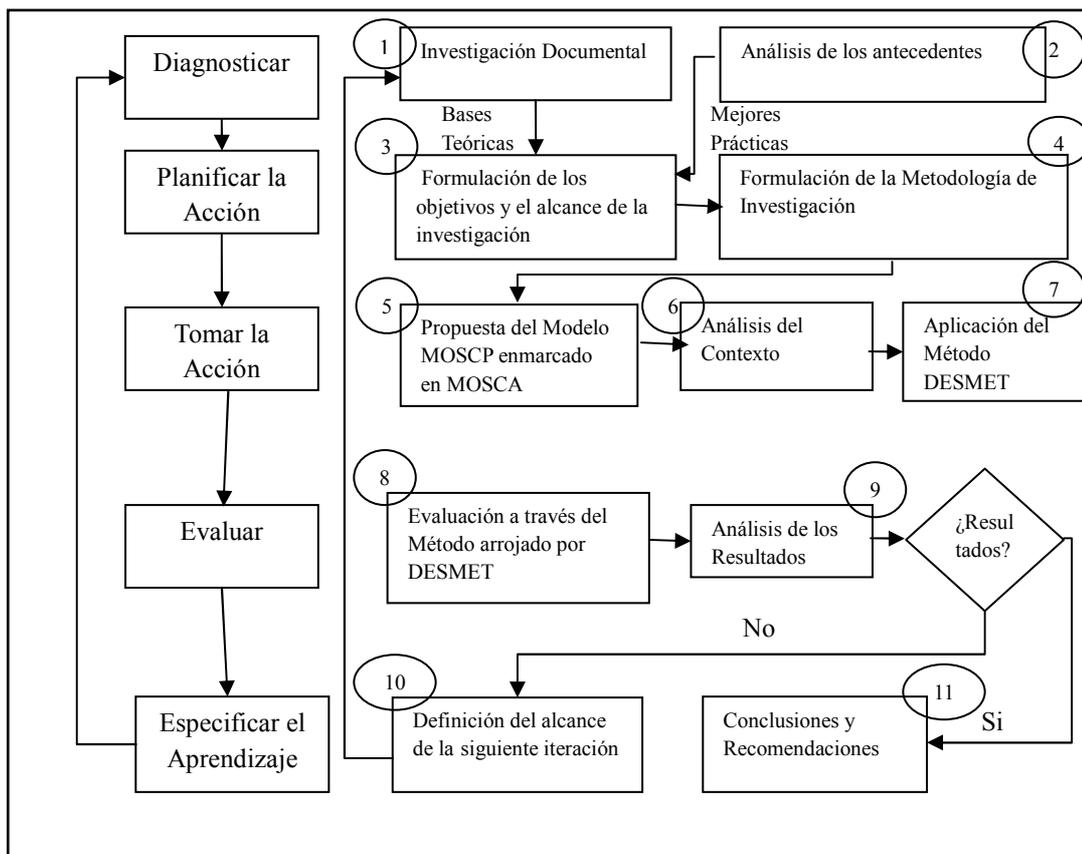
Para consolidar estos conceptos que enmarcan la investigación, se hace necesario la revisión de fuentes documentales: libros, artículos de índole científico, asesoría de expertos en la materia de calidad del software, así como, en el proceso de pruebas.

#### **3.2.4.2 Factibilidad Económica**

Actualmente se cuenta con el apoyo de expertos en calidad del software y proceso de pruebas de dos casas de estudio venezolanas, a saber, Universidad Centroccidental Lisandro Alvarado (UCLA), además de la Universidad Simón Bolívar (USB), específicamente del Laboratorio de Investigación en Sistemas de Información (LISI). Además, de los artículos publicados en revistas científicas.

### 3.2.5 Procedimiento de la Investigación

Como se mencionó en la sección 3.2 se utilizó el Framework Metodológico del Laboratorio de Investigación en Sistemas de Información (LISI), el cual consta de 11 fases o actividades agrupadas en cinco grandes fases, que se corresponden con las fases de Investigación-Acción (Ver Tabla 15). La Figura 13 muestra la adaptación del Framework Metodológico del LISI para la presente investigación.



**Figura 13:** Framework Metodológico de la Investigación.

**Fuente:** Universidad Simón Bolívar.

**Autor** Adaptado de Pérez et.al (2004)

Como se puede observar en la Figura 13, el Framework metodológico establece los pasos para la presente investigación, a continuación se describe cada uno de ellos.

### **3.2.5.1 Investigación Documental**

Se encuentra en la fase de Diagnosticar, en ella se realizó una revisión de los términos relacionados a la conceptualización, características, elementos y proceso de las pruebas de software; así como lo relacionado a la calidad del software bajo la visión sistémica. El propósito de esta etapa es determinar un marco teórico que avale la presente investigación, el resultado de ésta actividad se encuentra en el Capítulo II (Marco Teórico).

### **3.2.5.2 Análisis de los Antecedentes**

Pertenece a la fase de Diagnosticar. En primer lugar, se realizó un análisis de cada uno de los modelos de referencia de pruebas de software existentes para determinar características, fortalezas, debilidades de cada uno de ellos y así determinar cómo abordan dichos modelos el proceso de pruebas. En el marco de la investigación, se realizó una exploración del Modelo Sistémico de Calidad del Software (MOSCA), con la finalidad de evaluar la cobertura que posee actualmente del proceso de pruebas de software. Todo lo anterior fue el insumo para determinar posibles características del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP). El resultado de ésta actividad se encuentra en el Capítulo II (Antecedentes).

### **3.2.5.3 Formulación de los objetivos y el alcance de la investigación**

Etapa de la fase Planificar la acción donde se definen claramente los objetivos de la investigación, con el propósito de establecer el alcance de la investigación. El objetivo de esta actividad está plasmado en la justificación del problema y la especificación de los objetivos del Trabajo de Grado, los cuales se pueden ubicar en el Capítulo I (El Problema).

### **3.2.5.4 Formulación de la Metodología de Investigación**

Etapa de la fase Planificar la acción donde se construyó la adaptación para esta investigación de la metodología Investigación-Acción. El presente capítulo es el resultado de esta actividad.

### **3.2.5.5 Propuesta del Modelo MOSCP enmarcado en MOSCA**

Se encuentra ubicado en la fase de Tomar la acción, el objetivo es un Modelo enmarcado en MOSCA que cubra los aspectos relacionados al proceso de pruebas.

Para lograr la propuesta del Modelo Sistémico de Calidad para el Proceso de Pruebas (MOSCP) es necesario realizar los siguientes pasos:

#### **3.2.5.5.1 Análisis de los niveles que conformarán el Modelo**

En este paso se realizará el diagrama del árbol jerárquico del modelo, para esto se realizarán las siguientes actividades

- **Determinar las dimensiones del modelo**, estas dimensiones consideran aspectos internos y contextuales del proceso de pruebas.
- **Establecer las categorías del modelo**, se definen las categorías relacionadas al proceso de pruebas de software.
- **Determinar las características del modelo**, cada categoría tiene asociadas un conjunto de características, las cuales definen las áreas claves a satisfacer para lograr, asegurar y controlar la calidad del proceso de pruebas de software.
- **Definir las métricas**, para cada categoría se definió un conjunto de métricas asociadas, las cuales fueron utilizadas para medir el nivel de calidad sistémica.

#### **3.2.5.5.2 Definición del Algoritmo de Procesamiento de la Información**

En él se presenta los pasos a seguir para medir la calidad sistémica del proceso de pruebas en una organización a través de la aplicación del Modelo Sistémico de Calidad para el Proceso de Pruebas (MOSCP).

### 3.2.5.6 Análisis del Contexto

Etapa de la fase Tomar la acción, donde se determinan las especificaciones para implementar MOSCP. El objetivo es preparar el entorno donde fue evaluada la propuesta.

### 3.2.5.7 Aplicación del Método DESMET

Se encuentra en la fase Tomar la acción, en ella se ejecuta la metodología DESMET, con el fin de obtener un método fiable e imparcial para la evaluación. A continuación se presenta una descripción del método, sus objetivos, así como los criterios que se utilizan para la selección del criterio a utilizar.

Según Kitchenham et.al (1997), la metodología está compuesta por un conjunto de métodos con sus respectivas herramientas que pueden ser utilizadas o aplicadas por una organización en particular, dicha metodología tiene por objeto la evaluación de métodos y herramientas de Ingeniería del software.

En el mismo orden de ideas, se exponen los criterios de selección, los cuales permitirán escoger uno de los nueve métodos de evaluación propuestos por Kitchenham et.al (1997):

- **Experimento cuantitativo**, conforma el método científico básico para la determinación de diferencias entre métodos o herramientas.
- **Estudio de caso cuantitativo**, consiste en la evaluación de un método o herramienta después que éste ha sido utilizado en un proyecto real.
- **Encuesta cuantitativa**, es utilizada cuando varias herramientas o métodos han sido usados en la organización, y consiste en interrogar a los usuarios para que suministren información acerca de alguna característica de interés.
- **Análisis de características por proyección**, por lo general se basa en la documentación de la herramienta o método a evaluar.
- **Análisis de características por estudio de caso**, consiste en el análisis de características del método o herramienta luego que el mismo ha sido aplicado en un proyecto real.

- **Análisis de características por experimento**, en este caso se sugiere que un conjunto de usuarios potenciales pruebe el objeto de evaluación antes de realizar sus evaluaciones.
- **Análisis de características por encuesta**, es realizada por personas con experiencia en el uso de la herramienta o con conocimientos teóricos de la misma.
- **Análisis cualitativos de efectos**, se refiere a la utilización de juicios de expertos para determinar los efectos cuantitativos de diferentes métodos y herramientas.
- **Benchmarking**, consiste en evaluar el comportamiento de la herramienta o método en relación con otros ya establecidos.

Este mismo autor, identifica seis criterios que influyen en la selección del método de evaluación, los cuales se listan a continuación:

- El contexto de la evaluación.
- La naturaleza del impacto esperado de la utilización del método o herramienta.
- La naturaleza del objeto de evaluación, es decir, si es un método, una herramienta o un método genérico.
- El alcance del impacto del método o herramienta.
- La madurez del método o herramienta.
- La curva de aprendizaje asociada al método o herramienta.
- Las habilidades de medición que posee la organización encargada de evaluar.

Particularmente, en el caso de esta investigación DESMET, fue utilizada para la planificación y posterior aplicación del modelo propuesto y así evaluar la calidad sistémica del proceso de pruebas de una determinada empresa.

### 3.2.5.8 Evaluación a través del Método arrojado por DESMET

Esta actividad se ubica en la fase Evaluar, consiste en aplicar el método de evaluación arrojado por DESMET, con el propósito de evaluar el modelo siguiendo un criterio de evaluación apropiado.

En este propósito, dada la naturaleza del proyecto, se eligió el método de evaluación Análisis de características por estudio de caso, propuesto por la metodología, el cual consiste en la evaluación de las características de calidad en un proyecto de software real, es decir, la evaluación en una organización, específicamente en la presente investigación se puede indicar las siguientes actividades:

- **Identificar el modelo a Evaluar**, el objeto de la evaluación fueron las relaciones establecidas para las características de calidad del proceso de pruebas. Específicamente, el modelo a evaluar fue el Modelo Sistémico de Calidad para el Proceso de Pruebas (MOSCP).
- **Identificar un conjunto de criterios a evaluar**, los criterios que deben ser evaluados fueron los relacionadas al proceso de pruebas.
- **Evaluar el modelo contra los criterios identificados**, en este sentido, es importante resaltar que se realizó enfocado en el proceso de pruebas, el mismo estuvo dirigido a cuatro grupos: Líderes de Pruebas, Desarrolladores, Analistas de Pruebas, Clientes.
- **Evaluar un proyecto piloto**, en este punto del proceso, se seleccionó una empresa, en la cual, se eligió un proyecto de desarrollo de software, en este caso, se seleccionó el proceso de pruebas para aplicar el modelo propuesto.
- **Analizar los valores resultantes y realizar un reporte de evaluación**, en este paso se analizó los resultados correspondientes, a nivel del proceso de pruebas, para finalmente indicar el nivel de calidad sistémico en el proceso de pruebas de software.

### **3.2.5.9 Análisis de los Resultados**

Se ubica en la fase Evaluar, consiste en examinar los resultados de acuerdo a los objetivos planteados en la investigación, en lo que se refiere al modelo y los beneficios obtenidos. Al culminar esta fase se valoró si el resultado se encuentra dentro de lo esperado, se pasa al paso 11, en caso contrario, al paso 10.

### **3.2.5.10 Definición del alcance de la siguiente iteración**

Etapa de la fase Especificar el aprendizaje, consiste en realizar las modificaciones al modelo propuesto para aumentar la confiabilidad del mismo y sus posibilidades de éxito.

### **3.2.5.11 Conclusiones y Recomendaciones**

Se ubica en la fase Especificar el aprendizaje, en esta actividad se organizan las conclusiones relativas al modelo propuesto. Por último, se dictaron las recomendaciones relacionadas a trabajos futuros e investigaciones relacionadas.

Como se puede observar, el marco metodológico propuesto se puede realizar n cantidad de iteraciones, pero para efectos de la presente investigación se realizará sólo una iteración debido al factor tiempo involucrado en la Tesis de Maestría. Para dar continuidad al presente trabajo, a continuación se procede a desarrollar la propuesta de la investigación.

## **CAPITULO IV**

### **PROPUESTA DEL MODELO SISTÉMICO DE CALIDAD PARA EL PROCESO DE PRUEBAS DE SOFTWARE (MOSCP)**

Para dar continuidad al ciclo metodológico y luego de haber examinado los antecedentes y el marco teórico relacionados con esta investigación, en esta sección se presenta el modelo conceptual para el proceso de pruebas de software, dicho modelo representa la sinopsis de los conceptos y relaciones que avalan el Modelo propuesto en la presente investigación. Finalmente, se presenta la propuesta del Modelo Sistemico de Calidad para el proceso de pruebas de Software (MOSCP).

#### **4.1 Modelo Conceptual**

Según Moreira et.al (2002), los modelos conceptuales son representaciones externas, compartidas por una determinada comunidad y consistentes con el conocimiento científico que esa comunidad posee. Con relación a este concepto, Gea et.al (2003), señalan que, estos modelos, en algunos casos, se pueden representar mediante una ontología de conceptos y relaciones que suceden en el sistema. Otras veces, se puede optar por métodos formales o notaciones ampliamente usadas y conocidas para representar los conceptos más relevantes.

De lo anteriormente expuesto, se puede decir que un modelo conceptual muestra los conceptos presentes en el dominio del problema; en este modelo se puede mostrar además, los conceptos, los atributos asociados a los conceptos (opcionalmente) y la

relación o asociación entre ellos. Para efectos de la presente investigación el modelo conceptual fue construido utilizando notación UML (Ver Figura 14).

A continuación, se procede a explicar brevemente el modelo conceptual de la presente investigación.

Las pruebas de software, son un proceso dinámico el cual comprende un conjunto de actividades que deben ser planeadas y ejecutadas de manera sistemática. El objetivo de las pruebas de software es identificar defectos en el código y comprobar el correcto funcionamiento del software elaborado de acuerdo a los requisitos del cliente, mejorando la calidad del producto.

Las pruebas de software poseen las siguientes características:

- Presente durante el todo el ciclo de desarrollo.
- Dinámicas
- Finitas.
- Selectivas.
- Comprobables.

A su vez, las pruebas de software generan los siguientes artefactos o documentos:

- Plan de Pruebas
- Caso de Prueba
- Procedimientos de Prueba
- Suite de pruebas
- Script de Pruebas
- Resultado Esperado
- Resultado Obtenido

Las pruebas de software se realizan siguiendo un proceso, el cual debe ser definido, controlado y gestionado por personas, dicho proceso contempla la ejecución de las siguientes fases:

- Planificación y Control
- Análisis y Diseño de las Pruebas
- Implementación y Ejecución
- Evaluar el criterio de terminación y Reportes
- Actividades de Cierre

Existen diversos modelos cuyo objetivo es la mejora del proceso de pruebas de software, entre los cuales se puede mencionar: CMMI, TMM, TPI, TMMI, TestPAI, ISO/IEC 29119, los cuales aportan las mejores prácticas en cuanto a la mejora del proceso de pruebas, entre las cuales destacan las siguientes:

- Incorporación de participantes claves en el proceso de pruebas.
- Planificación y organización de las pruebas.
- Evaluar la calidad del software.
- Incorporación de métricas.
- Entorno de pruebas.
- Comunicación.
- Procedimiento detallado para la selección de herramientas de pruebas.
- Analizar, evaluar y comunicar los resultados obtenidos.
- Documentación de las pruebas.

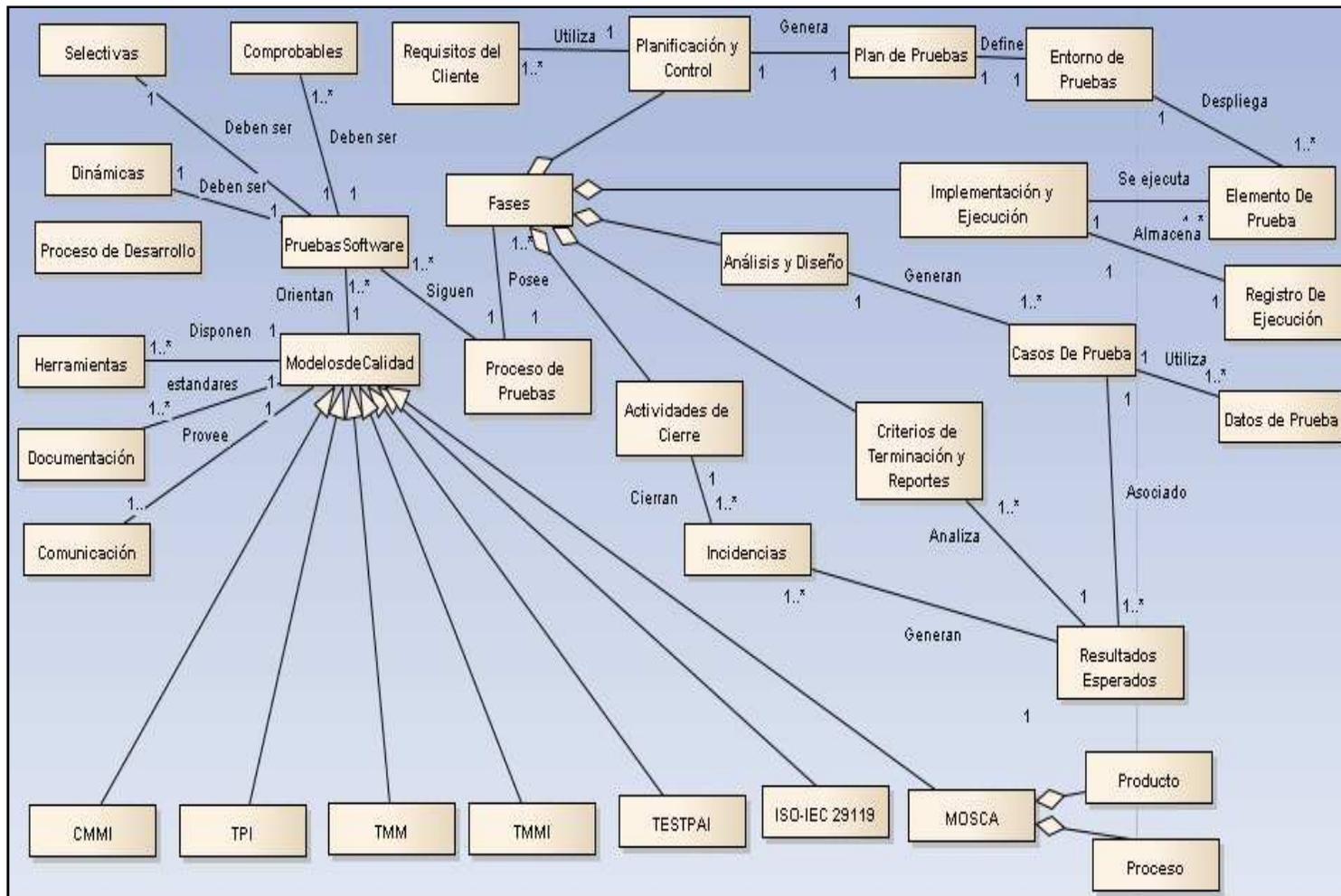
Por otro lado, la calidad del software se debe garantizar de manera holística, es decir, se debe tomar en cuenta tanto la calidad del proceso como la calidad del

producto, en este sentido, se dispone del Modelo Sistémico Calidad del Software (MOSCA), el cual fue considerado punto de referencia en la presente investigación.

La presente propuesta utilizará los niveles 0, 1, 2,3, según MOSCA a saber: dimensión, categoría y métricas.

En la Figura 15, se presenta el diagrama con la estructura de MOSCA que se usará en la propuesta y los niveles que lo conforman son los siguientes:

- a) **Nivel 0 Dimensión:** Está conformado por los aspectos contextuales e internos del proceso.
- b) **Nivel 1 Categorías:** Debido a la naturaleza de la investigación se seleccionaron las siguientes categorías:
  - **Cliente-Proveedor (CUS),** está conformada por procesos que impactan directamente al cliente, apoya el desarrollo y la transición del software hasta el cliente, y provee la correcta operación y uso del producto o servicio de software. Es de vital importancia, que el proceso de pruebas de software tenga presente la visión del cliente para asegurar la calidad del proceso de pruebas y a su vez la calidad del producto.
  - **Ingeniería (ENG),** consiste en procesos que directamente especifican, implementan o mantienen el producto de software, su relación con el sistema y su documentación. Para el proceso de pruebas, se hace necesario tener las especificaciones a seguir durante la ejecución del proceso de pruebas de software, además es significativo documentar las evidencias de la ejecución del proceso.



**Figura 14:** Modelo Conceptual Pruebas de Software de la presente investigación.

**Fuente:** Autor de la investigación.

**Autor:** Elaboración Propia (2012).

- **SopORTE (SUP)**, consta de procesos que pueden ser empleados por cualquiera de los procesos (incluyendo a los de soporte) en varios niveles del ciclo de vida de adquisición. Los procesos de soporte deben estar presentes en el proceso de pruebas de software, debido a que contribuyen a certificar cada una de las actividades realizadas durante el proceso de pruebas.
- **Gestión (MAN)**, abarca los procesos que contienen prácticas genéricas, que pueden ser utilizadas por cualquier personal que dirija algún tipo de proyecto o proceso. Para el proceso de pruebas es primordial establecer mecanismos que permitan gestionar el correcto desarrollo del mismo.
- **Organizacional (ORG)**, agrupa los procesos que establecen las metas comerciales de la organización y desarrollan bienes (valores) de proceso, producto y recurso, que ayudarán a la organización a alcanzar sus metas en los proyectos. En el caso del proceso de prueba, también se toma en cuenta los objetivos a alcanzar en el mismo en términos de tiempo, recursos involucrados, resultados esperados, entre otros, los cuales tendrán un impacto en la consecución de los objetivos de la organización.

c) **Nivel 2 Características:** Una vez definidas las categorías a ser evaluadas, se analizaron las características asociadas a cada categoría y se descartaron aquellas que no se ajustan al contexto de la investigación, dichas características puntualizan la áreas claves que se deben alcanzar, asegurar y controlar la calidad de cada una de las categorías a la que se encuentran vinculadas y la del proceso de pruebas. A continuación, se describe cada una de las características

Características a ser evaluada en la categoría **Cliente-Proveedor (CUS)**:

- **Determinación de Requisitos (CUS1)**, evalúa si en el proceso de pruebas se realizó una correcta determinación de requisitos del software, dichos

requisitos serán los insumos para definir todos los recursos necesarios para la ejecución de las pruebas, entre las cuales destacan: herramientas, roles, tiempo, entorno de pruebas, entre otros.

Características a ser evaluada en la categoría **Ingeniería (ENG)**:

- **Desarrollo (ENG3)**, esta característica evalúa todos los procesos que se llevaran a cabo durante la ejecución de las pruebas de software, dichos procesos influyen directamente en el producto software.

Características a ser evaluada en la categoría **Soporte (SUP)**:

- **Documentación (SUP1)**, evalúa la generación de los soportes necesarios que avalan el correcto funcionamiento de los desarrollos que son entregados al área de pruebas de software.
- **Aseguramiento de la calidad (SUP2)**, evalúa todas las actividades que se realizaran para garantizar la calidad del proceso de pruebas.
- **Revisión Conjunta (SUP3)**, evalúa el estado y los productos del proceso de pruebas.
- **Auditoria (SUP4)**, evalúa el cumplimiento de requerimientos, así como también la ejecución del proceso de pruebas enmarcado en el plan de pruebas.

Características a ser evaluada en la categoría **Gestión (MAN)**:

- **Gestión (MAN1)**, evalúa la gestión general del proceso de pruebas de software en cada una de sus fases.

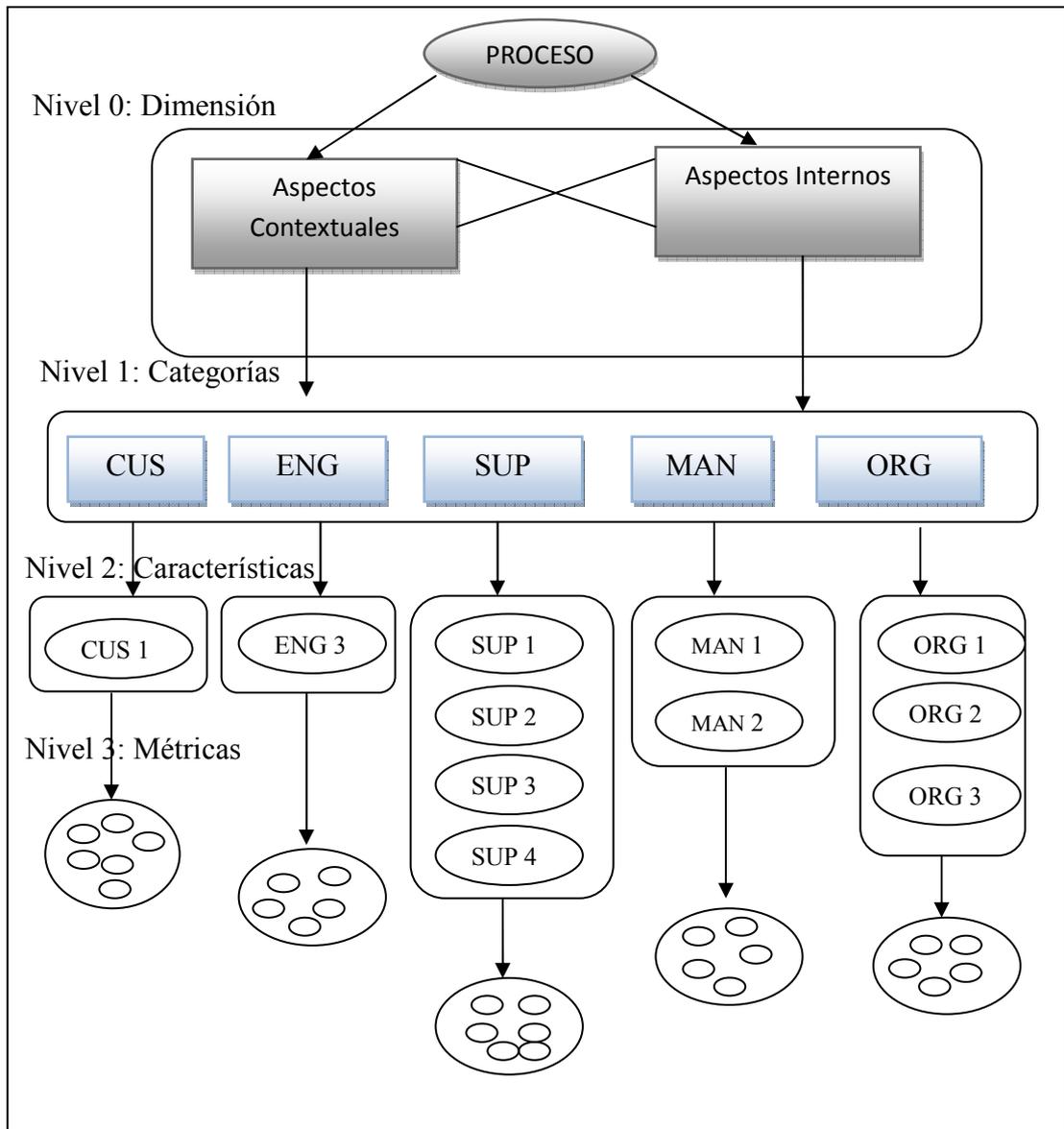
- **Gestión de Calidad (MAN2)**, evalúa los niveles de cumplimiento en la ejecución de las pruebas de software, según lo establecido en el plan de pruebas.

Características a ser evaluada en la categoría **Organizacional (ORG)**:

- **Lineamientos Organizacionales (ORG1)**, evalúa las directrices diseñadas en la organización para la ejecución del proceso de pruebas.
- **Evaluación del Proceso (ORG2)**, evalúa el grado de eficiencia del proceso de pruebas.
- **Reuso (ORG3)**, evalúa la posibilidad de utilizar los artefactos generados (plan de pruebas, casos de prueba, datos de prueba, documentación), en el proceso de pruebas se puede utilizar en pruebas posteriores que posean características comunes.

d) **Nivel 3 Métricas:** Luego de definir las características a ser evaluadas, se examinaron las métricas asociadas a cada característica y debido a la naturaleza de la investigación se procedió a definir métricas pertinentes a nivel del contexto del presente trabajo de investigación.

En la Figura 15, se observa la dimensión, categorías, características y métricas tomadas de MOSCA, sin embargo la presente investigación va a proponer características y métricas adicionales relacionadas al proceso de pruebas de software. Para alcanzar este objetivo, la propuesta está apoyada en los conceptos expuestos en el marco teórico y en los trabajos realizados por otros autores.



**Figura 15:** Diagrama del Modelo Sistémico de Calidad MOSCA-Consideraciones de MOSCA

**Fuente:** Adaptado de Mendoza et. al (2005).

**Autor:** Elaboración Propia (2012).

En este orden de ideas, para cada una de las características seleccionadas, fueron elaboradas sus correspondientes métricas, además fueron creadas nuevas características y nuevas sub-características, así como nuevas métricas.

Seguidamente, se describen cada una de las características, sub-características y métricas nuevas, las cuales se encuentran apoyadas en los diversos conceptos e investigaciones expuestas en el marco teórico.

Para la categoría **Cliente-Proveedor (CUS)** y la característica **Determinación de Requerimientos (CUS1)**, se crearon 2 sub-características y 4 métricas, las cuales se pueden apreciar en la Tabla 16.

- Documento Funcional (CUS 1.1): Posee dos (2) métricas.
- Comunicación (CUS 1.2): Posee dos (2) métricas.

Sub-Característica	Métrica	Pregunta	Formulación	Dirigido a
Documento Funcional (CUS 1.1)	Documento Funcional	¿Recibe el documento funcional de requisitos de software?	0= no 1=si	Analista de Pruebas
	Revisión del Documento Funcional	¿Realiza la revisión del documento funcional antes de elaborar los casos de prueba?	0= no 1=si	Analista de Pruebas
Definición del Alcance (CUS 1.2)	Validación de requisitos	¿Se valida que los requisitos están correctamente definidos?	0= no 1=si	Analista de Pruebas, Usuario
	Comunicación	¿Existe comunicación, entre las partes involucradas?	0= no 1=si	Analista de Pruebas, Usuario

**Tabla 16:** Nuevas sub\_características, sub-sub-características y métricas para la categoría Cliente-Proveedor (CUS).

**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2012).

Para la categoría **Ingeniería (ENG)**, se crearon 5 nuevas características, con sus respectivas sub-características y métricas; así mismo, para la característica Desarrollo (ENG3), se crearon 3 sub-características y 6 métricas. A continuación, se describen cada una de las características.

**Planificación y Control (ENG1)**, está formada por 3 sub-características y 9 métricas, las cuales se pueden apreciar en la Tabla 17.

- Determinar el alcance de las pruebas (ENG 1.1), posee una (1) métrica.
- Desarrollar el plan de Pruebas (ENG 1.2), posee siete (7) métricas.
- Entorno de Pruebas (ENG 1.3), posee una (1) métrica.

**Análisis y Diseño (ENG2)**, consta de 2 sub- características y 12 métricas, en la Tabla 18 se muestra la constitución de esta característica.

- Definición de los elementos a probar (ENG 2.1), posee tres (3) métricas.
- Generación de casos de Prueba (ENG 2.2), posee nueve (9) métricas.

**Desarrollo (ENG3)**, se crearon 3 sub-características y 6 métricas, las cuales se pueden observar en la Tabla 19.

- Ejecución de Casos de Prueba (ENG 3.1), posee dos (2) métricas.
- Ejecución de la plataforma (ENG 3.2), posee una (1) métrica.
- Ejecución del Sistema (ENG 3.3), posee tres (3) métricas.

**Criterios de Terminación (ENG4)**, está formada por 3 sub-características y 9 métricas, las mismas se pueden observar en la Tabla 20:

- Análisis de Resultados (ENG 4.1), posee siete (7) métricas.
- Determinar Riesgos (ENG 4.2), posee una (1) métrica.
- Determinar Costos (ENG 4.3), posee una (1) métrica.

**Reportes (ENG 5)**, está formada por 3 sub-características y 5 métricas, lo cual se muestra en la Tabla 21:

- Incidencias (ENG 5.1), posee dos (2) métricas.
- Culminación de Pruebas (ENG 5.2), posee una (1) métrica.

- Certificación (ENG 5.3), posee dos (2) métricas.

**Actividades de Cierre (ENG 6)**, posee 3 sub-características y 3 métricas, las cuales se pueden observar en la Tabla 22:

- Comprobar Entregables (ENG 6.1), posee una (1) métrica.
- Cierre de Incidencias (ENG 6.2), posee una (1) métrica.
- Ambiente de Prueba (ENG 6.3), posee una (1) métrica.

Características	Sub- Características	Métrica	Pregunta	Formulación	Dirigido a
Planificación y Control (ENG 1)	Determinar el alcance de las pruebas (ENG 1.1)	Alcance de las Pruebas	¿Se establece el alcance de las pruebas antes de la elaboración del plan de pruebas?	0= no 1=si	Líder de Pruebas.
	Desarrollar el plan de Pruebas (ENG 1.2)	Definición de Actividades	¿Se definen las actividades a ser realizadas?	0= no 1=si	Líder de Pruebas
		Determinar Roles	¿Se determina el personal que estará involucrado en las pruebas?	0= no 1=si	Líder de Pruebas
		Tiempo Requerido	¿Se establece el tiempo requerido para la ejecución de las pruebas?	0= no 1=si	Líder de Pruebas
		Riesgos	¿Se estudian los posibles riesgos en la ejecución de las pruebas?	0= no 1=si	Líder de Pruebas
		Niveles de Cobertura	¿Se establecen los niveles de	0= no 1=si	Líder de Pruebas

			cobertura de las pruebas de software?		
		Plan de Pruebas	¿Se genera el plan de pruebas?	0= no 1=si	Líder de Pruebas
		Documentación	¿Se realiza la documentación del plan de pruebas?	0= no 1=si	Líder de Pruebas
	Entorno de Pruebas (ENG 1.3)	Disponibilidad del Entorno de Pruebas	¿Se verifica la disponibilidad del entorno para la ejecución de las pruebas?	0= no 1=si	Líder de Pruebas

**Tabla 17:** Nueva característica Planificación y Control (ENG1), sub\_ características y métricas para la categoría Ingeniería (ENG).

**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2012).

Características	Sub- Característica	Métrica	Pregunta	Formulación	Dirigido a
Análisis y Diseño (ENG 2)	Definición de los elementos a probar (ENG 2.1)	Selección de elementos a probar	¿Se seleccionan los elementos a probar?	0= no 1=si	Analista de Pruebas
		Riesgos de fallo	¿Se toma en cuenta los riesgos de fallo en la selección de los elementos a probar?	0= no 1=si	Analista de Pruebas
		Aporte del Usuario	¿Se toma en cuenta el aporte del usuario en la selección de los elementos a probar?	0= no 1=si	Analista de Pruebas
	Generación de casos de Prueba (ENG 2.2)	Nivel y técnicas de pruebas	¿Se establece el nivel y técnicas de pruebas a utilizar?	0= no 1=si	Analista de Pruebas
		Número de casos de prueba	¿Se especifica un número de caso de prueba?	0= no 1=si	Analista de Pruebas
		Entradas del caso de pruebas	¿Se indican las entradas del caso de prueba?	0= no 1=si	Analista de Pruebas
		Descripción de pasos de ejecución	¿Se describe detalladamente los pasos a ejecutar en el caso de prueba?	0= no 1=si	Analista de Pruebas
		Descripción de resultados esperados	¿Se describen los resultados esperados para cada prueba?	0= no 1=si	Analista de Pruebas

		Aplicación a probar	¿Se indica que aplicación o componente se está probando?	0= no 1=si	Analista de Pruebas
		Relación con otros componentes	¿Se indica la relación del caso de prueba con otra aplicación o componente?	0= no 1=si	Analista de Pruebas
		Apoyo de Software de Gestión	¿Se encuentran bajo el control de un software de gestión?	0= no 1=si	Analista de Pruebas
		Documentación	¿Se documentan los casos de prueba?	0= no 1=si	Analista de Pruebas

**Tabla 18:** Nueva característica Análisis y Diseño (ENG2), sub\_ características y métricas para la categoría Ingeniería (ENG).

**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2012).

Sub-Característica	Métrica	Pregunta	Formulación	Dirigido a
Ejecución de Casos de Prueba (ENG 3.1)	Ejecución de forma manual	¿Se ejecutan los casos de prueba de forma manual?	0= no 1=si	Analista de Pruebas
	Ejecución de forma automática	¿Se ejecutan los casos de prueba de forma automática?	0= no 1=si	Analista de Pruebas
Ejecución de la plataforma (ENG 3.2)	Ejecución exitosa de la plataforma	¿Se ejecutan exitosamente la plataforma necesaria para desplegar el sistema bajo prueba?	0= no 1=si	Analista de Pruebas
Ejecución del Sistema (ENG 3.3)	Ejecución de acuerdo a los casos de prueba	¿Se ejecuta el sistema de acuerdo a los casos de prueba?	0= no 1=si	Analista de Pruebas
	Utilización de los datos de prueba	¿Se utilizan los datos de prueba especificados en el caso de prueba?	0= no 1=si	Analista de Pruebas
	Herramientas para la ejecución	¿Se dispone de herramientas para la ejecución de la prueba?	0= no 1=si	Analista de Pruebas

**Tabla 19:** Nueva característica Desarrollo (ENG3), sub\_ características y métricas para la categoría Ingeniería (ENG).

**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2012).

Característica	Sub- Característica	Métrica	Pregunta	Formulación	Dirigido a
Criterios de Terminación (ENG 4)	Análisis de Resultados (ENG 4.1)	Existencia de errores	¿El software probado tiene errores?	0= no 1=si	Analista de Pruebas
		Generación de incidencias	¿En caso de existir errores se genera una incidencia?	0= no 1=si	Analista de Pruebas
		Comunicación de incidencias	¿Se comunica al desarrollador la incidencia?	0= no 1=si	Analista de Pruebas
		Suspensión de la prueba	¿La prueba fue suspendida por problemas del entorno?	0= no 1=si	Analista de Pruebas
		Cumplimiento de los objetivos	¿La prueba cumple con las funcionalidades de acuerdo a los objetivos de las pruebas?	0= no 1=si	Analista de Pruebas.
		Certificación de las pruebas	¿Se solicita al usuario la certificación de las pruebas realizadas?	0= no 1=si	Analista de Pruebas. Desarrollador. Usuario. Líder de Pruebas.
		Comunicación de éxito	¿Se comunica al desarrollador el éxito del proceso de pruebas?	0= no 1=si	Analista de Pruebas. Desarrollador.
	Determinar Riesgos (ENG 4.2)	Estudio de riesgos de	¿Se estudian los riesgos de	0= no 1=si	Líder de Pruebas. Analista de

		fallos potenciales	fallos potenciales		Pruebas
	Determinar Costos (ENG 4.3)	Costos de continuación de las pruebas	¿Se consideran los costos de continuación de las pruebas?	0= no 1=si	Líder de Pruebas. Analista de Prueba.

**Tabla 20:** Nueva característica Criterios de Terminación (ENG4), sub\_ características y métricas para la categoría Ingeniería (ENG).

**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2012).

Característica	Sub- Característica	Métrica	Preguntas	Formulación	Dirigido a
Reportes (ENG 5)	Incidencias (ENG 5.1)	Generación de código para las incidencias	¿Se genera un id por cada una de las incidencias detectadas?	0= no 1=si	Analista de Pruebas.
		Comunicación de incidencias al equipo de desarrollador	¿Se comunican las incidencias detectadas al responsable del desarrollo?	0= no 1=si	Analista de Pruebas Desarrolladores.
	Culminación de Pruebas (ENG 5.2)	Generación de reportes de culminación de las pruebas	¿Se generan reportes una vez finalizada las pruebas de software?	0= no 1=si	Analista de Pruebas
	Certificación (ENG 5.3)	Generación de reportes de certificación	¿Se generan reportes de certificación de	0= no 1=si	Analista de Prueba.

		de las pruebas	las pruebas del elemento de software?		
--	--	----------------	---------------------------------------	--	--

**Tabla 21:** Nueva característica Reportes (ENG 5), sub\_ características y métricas para la categoría Ingeniería (ENG).

**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2012).

Característica	Sub- Característica	Métrica	Pregunta	Formulación	Dirigido a
Actividades de Cierre (ENG 6)	Comprobar Entregables (ENG 6.1)	Transmisión de entregables	¿Los entregables fueron transmitidos según lo planificado?	0= no 1=si	Analista de Prueba. Líder de Pruebas. Desarrolladores
	Cierre de Incidencias (ENG 6.2)	Cierre de Incidencias	¿Se realizó el cierre de incidencias?	0= no 1=si	Analista de Prueba, Líder de Pruebas, Desarrolladores
	Ambiente de Prueba (ENG 6.3)	Almacenamiento del ambiente de pruebas	¿Se guarda el ambiente de prueba para futuras pruebas?	0= no 1=si	Analista de Prueba.

**Tabla 22:** Nueva característica Actividades de Cierre (ENG 6), sub\_ características y métricas para la categoría Ingeniería (ENG).

**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2012).

En el mismo orden de ideas, para la categoría **Soporte (SUP)** y la característica **Documentación (SUP1)**, se crearon 6 sub-características y 13 métricas; las cuales se pueden apreciar en la Tabla 23.

- Solicitud de Pruebas (SUP 1.1), posee tres (3) métricas.
- Soporte de Ejecución (SUP 1.2), posee tres (3) métricas.
- Soporte de Incidencias (SUP 1.3), posee tres (3) métricas.

- Documentos Recibidos (SUP 1.4), posee una (1) métrica.
- Soporte de Certificación (SUP 1.5), posee dos (2) métricas
- Repositorio de Documentación (SUP 1.6), posee una (1) métrica.

Sub- Característica	Métrica	Pregunta	Formulación	Dirigido a
Solicitud de Pruebas (SUP 1.1)	Generación de solicitud de pruebas	¿Se genera la solicitud de pruebas del elemento a probar?	0= no 1=si	Desarrolladores
	Generación de código para cada solicitud de pruebas	¿Se genera un identificador de prueba para cada solicitud de pruebas recibida?	0= no 1=si	Líder de Pruebas
	Asignación de especialista de pruebas	¿Se asigna un especialista de pruebas para cada ticket de prueba generado?	0= no 1=si	Líder de Pruebas
Soporte de Ejecución (SUP 1.2)	Estándar de documentación de la ejecución	¿Se sigue un estándar en la documentación durante la ejecución de las pruebas?	0= no 1=si	Líder de Pruebas Analista de Pruebas
	Estándar en el repositorio de documentación de ejecución de las pruebas	¿El repositorio de documentación de ejecución de pruebas posee una estructura de directorios estándar?	0= no 1=si	Líder de Pruebas Analista de Pruebas
	Almacenamiento	¿Se guardan las evidencias	0= no 1=si	Analista de Pruebas

	de evidencias	(pantallas, scripts con sus respectivas salidas, logs, entre otros) de la ejecución de los casos de prueba?		
Soporte de Incidencias (SUP 1.3)	Estándar en el repositorio de documentación de las incidencias.	¿El repositorio de documentación de las incidencias detectadas durante la ejecución de pruebas posee una estructura de directorios estándar?	0= no 1=si	Líder de Pruebas Analista de Pruebas
	Almacenamiento de incidencias	¿Se almacenan las incidencias (pantallas, scripts con sus respectivas salidas, logs, entre otros) de la incidencia detectada?	0= no 1=si	Líder de Pruebas Analista de Pruebas
	Asociación de número de incidencia con el número de caso de prueba	¿El número de incidencia está asociado con el número del caso de prueba ejecutado?	0= no 1=si	Líder de Pruebas Analista de Pruebas
Documentos Recibidos (SUP 1.4)	Almacenamiento de documentos recibidos	¿Se almacenan los documentos recibidos para la ejecución de las pruebas?	0= no 1=si	Líder de Pruebas Analista de Pruebas
	Formato de			

Soporte de Certificación (SUP 1.5)	certificación	¿Se cuenta con un formato para la certificación de las pruebas realizadas?	0= no 1=si	Líder de Pruebas Analista de Pruebas
	Almacenamiento de la certificación de las pruebas	¿Se almacena la certificación de las pruebas realizadas?	0= no 1=si	Líder de Pruebas Analista de Pruebas
Repositorio de Documentación (SUP 1.6)	Almacenamiento en el repositorio de la documentación generada durante todo el proceso de pruebas	¿La documentación generada durante el proceso de pruebas se almacena en el repositorio de documentación?	0= no 1=si	Líder de Pruebas Analista de Pruebas

**Tabla 23:** Nuevas sub\_características y métricas para la categoría Soporte (SUP) y la característica Documentación (SUP 1).

**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2012).

En este mismo orden de ideas, para la categoría **Soporte (SUP)** y la característica **Aseguramiento de la calidad (SUP2)**, se crearon 3 sub-características y 10 métricas, las cuales se pueden observar en la Tabla 24.

- Plan de calidad (SUP 2.1), posee tres (3) métricas.
- Seguimiento de defectos (SUP 2.2), posee tres (3) métricas.
- Seguimiento de actividades (SUP 2.3), posee tres (4) métricas.

Sub- Característica	Métrica	Pregunta	Formulación	Dirigido a
Plan de calidad	Elaboración del plan de calidad	¿Se elabora el plan de calidad para el proceso de pruebas?	0= no 1=si	Líder de Pruebas

(SUP 2.1)	Asignación de roles	¿Se asignan roles a las personas encargadas de la ejecución de las pruebas?	0= no 1=si	Líder de Pruebas
	Comunicación del plan de calidad	¿Se comunica el plan de calidad a seguir a todos los involucrados?	0= no 1=si	Líder de Pruebas, Analista de Pruebas, Desarrolladores
Seguimiento de defectos (SUP 2.2)	Fallos por requisitos mal definidos	¿Los fallos observados, se deben a requisitos mal definidos?	0= no 1=si	Analista de Pruebas
	Fallos por declaración de variables incorrectas	¿Los fallos observados, se deben a declaraciones incorrectas de variables?	0= no 1=si	Analista de Pruebas
	Fallos por errores de programación	¿Los fallos observados, se deben a errores de programación?	0= no 1=si	Analista de Pruebas
Seguimiento de Actividades (SUP 2.3)	Actividades dentro del plan de pruebas	¿Se verifica que las actividades estén dentro del plan de pruebas?	0= no 1=si	Líder de Pruebas
	Artefactos en conformidad con los estándares.	¿Se realizan revisiones a los artefactos a verificar su conformidad con los estándares?	0= no 1=si	Líder de Pruebas
	Registro de auditorías	¿Se registran las auditorías realizadas?	0= no 1=si	Líder de Pruebas
	Control de documentación	¿Se realiza un control de la documentación de	0= no 1=si	Líder de Pruebas

		pruebas?		
--	--	----------	--	--

**Tabla 24:** Nuevas sub\_ características y métricas para la categoría Soporte (SUP) y la característica Aseguramiento de la Calidad (SUP 2).

**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2012).

Para la categoría **Soporte (SUP)** y la característica **Revisión Conjunta (SUP 3)**, se crearon una (1) sub- característica y cinco (5) métricas, las cuales se pueden observar en la Tabla 25.

- Ejecución del Proceso (SUP 3.1), posee cinco (5) métricas.

Sub- Característica	Métrica	Pregunta	Formulación	Dirigido a
Ejecución del Proceso(SUP 3.1)	Solicitud de revisión conjunta	¿Se solicitan revisiones periódicas de las actividades contempladas en el plan de pruebas de software?	0= no 1=si	Analista de Pruebas Líder de Pruebas Usuario
	Definición de recursos a utilizar	¿Se establecen los recursos a utilizar en las revisiones del proceso de pruebas de software?	0= no 1=si	Líder de Pruebas Analista de Pruebas Usuario
	Especificación del alcance	¿Se especifica el alcance de las revisiones del proceso de pruebas de software?	0= no 1=si	Líder de Pruebas Analista de Pruebas Usuario
	Documentación de problemas detectados	¿Se documentan los problemas detectados durante el proceso de revisión?	0= no 1=si	Líder de Pruebas Analista de Pruebas Usuario
	Comunicación de resultados	¿Se comunican los resultados del proceso de revisión	0= no 1=si	Líder de Pruebas Analista de Pruebas

		conjunta?		Usuario
--	--	-----------	--	---------

**Tabla 25:** Nuevas sub- características y métricas para la categoría Soporte (SUP) y la característica Revisión Conjunta (SUP 3).

**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2012).

Para la categoría **Soporte (SUP)** y la característica Auditoría (**SUP4**), se crearon una 1 sub- característica y 5 métricas, las cuales se pueden observar en la Tabla 26.

- Ejecución del Proceso (SUP 4.1), posee cinco (5) métricas.

Sub- Característica	Métrica	Pregunta	Formulación	Dirigido a
Ejecución del Proceso (SUP 4.1)	Solicitud de auditorias	¿Se solicitan auditorias periódicas de las actividades desarrolladas en el proceso de pruebas de software?	0= no 1=si	Analista de Pruebas Líder de Pruebas Usuario
	Especificación de los responsables de la ejecución de la auditoria.	¿Se establecen los responsables de la ejecución de la auditoría del proceso de pruebas de software?	0= no 1=si	Líder de Pruebas
	Alcance de la auditoria	¿Se especifica el alcance de las auditorías del proceso de pruebas de software?	0= no 1=si	Líder de Pruebas Analista de Pruebas Usuario
	Documentación de la auditoria	¿Se documentan los resultados de la evaluación del proceso de pruebas de software?	0= no 1=si	Líder de Pruebas Analista de Pruebas

	Comunicación de los resultados de la auditoría	¿Se comunican los resultados del proceso de auditoría del proceso de pruebas de software?	0= no 1=si	Líder de Pruebas Analista de Pruebas Usuario
--	--	---	---------------	--

**Tabla 26:** Nuevas sub\_ características y métricas para la categoría Soporte (SUP) y la característica Auditoría (SUP 4)

**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2012).

De la misma manera, para la categoría **Gestión (MAN)** y la característica **Gestión (MAN1)**, se crearon 4 sub- características y 16 métricas, las cuales se pueden observar en la Tabla 27.

- Gestión de Planificación (MAN 1.1), la cual posee cuatro (4) métricas.
- Ejecución del Proceso (MAN 1.2), está constituida por seis (6) métricas.
- Seguimiento de las Pruebas (MAN 1.3), se encuentra formada por cuatro (4) métricas.
- Cierre del Proceso (MAN 1.4), posee dos (2) métricas.

Sub-Característica	Métrica	Pregunta	Formulación	Dirigido a
Gestión de Planificación(MAN 1.1)	Definición de alcance de las pruebas	¿Se define el alcance de las pruebas?	0= no 1=si	Analista de Pruebas Líder de Pruebas
	Definición de responsables de la ejecución	¿Se definen los responsables de la ejecución de las pruebas de software?	0= no 1=si	Líder de Pruebas
	Identificación	¿Se identifican	0= no	Líder de Pruebas

	de los requisitos de las pruebas	los requisitos de las pruebas?	1=si	Analista de Pruebas Usuario
	Definición de escenarios de prueba	¿Se definen los escenarios de prueba?	0= no 1=si	Líder de Pruebas Analista de Pruebas
Ejecución del Proceso(MAN 1.2)	Asignación de recursos	¿Se asignan los recursos para la ejecución de las pruebas?	0= no 1=si	Líder de Pruebas Analista de Pruebas
	Configuración el entorno de pruebas	¿Se realiza la configuración del entorno de pruebas?	0= no 1=si	Analista de Pruebas
	Ejecución del plan de pruebas	¿Se ejecuta el plan de pruebas?	0= no 1=si	Analista de Pruebas
	Elaboración de los casos de prueba	¿Se elaboran los casos de prueba?	0= no 1=si	Analista de Pruebas
	Descripción de las actividades	¿Se describen las actividades de prueba a realizar?	0= no 1=si	Analista de Pruebas
	Información acerca de las incidencias detectadas	¿Se informa acerca de las incidencias detectadas?	0= no 1=si	Líder de Pruebas Analista de Pruebas Desarrolladores
	Definición de puntos de control	¿Se definen puntos de control durante la ejecución del proceso de	0= no 1=si	Líder de Pruebas Analista de Pruebas

Seguimiento de las Pruebas (MAN 1.3)		pruebas?		
	Seguimiento de la ejecución de las pruebas	¿Se realiza un seguimiento de la ejecución de las pruebas?	0= no 1=si	Líder de Pruebas Analista de Pruebas
	Informe acerca del avance de las pruebas	¿Se informa acerca del avance del proceso de pruebas?	0= no 1=si	Líder de Pruebas Analista de Pruebas
	Gestión de la calidad del proceso de pruebas	¿Se gestiona la calidad en el proceso de pruebas?	0= no 1=si	Líder de Pruebas Analista de Pruebas
Cierre del Proceso (MAN 1.4)	Verificación de la ejecución de los casos de prueba	¿Se verifica la ejecución de los casos de prueba elaborados?	0= no 1=si	Líder de Pruebas Analista de Pruebas Desarrolladores
	Documentación elaborado cumple los estándares	¿Se verifica si la documentación elaborada cumple con los estándares establecidos?	0= no 1=si	Líder de Pruebas Analista de Pruebas Desarrolladores

**Tabla 27:** Nuevas sub\_características y métricas para la categoría Gestión (MAN) y la característica Gestión (MAN 1).

**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2012).

A su vez, para la categoría **Gestión (MAN)** y la característica **Gestión de Calidad (MAN2)**, se crearon una 1 sub- característica y tres (3) métricas, las cuales se pueden observar en la Tabla 28.

Sub- Característica	Métrica	Pregunta	Formulación	Dirigido a
Niveles de Cobertura(MAN 2.1)	Verificación de los niveles de cobertura	¿Se verifican los niveles de cobertura de las pruebas establecidos en el plan de pruebas?	0= no 1=si	Analista de Pruebas Líder de Pruebas
	Rango de fallos detectados	¿Se establecen rango de fallos detectados?	0= no 1=si	Analista de Pruebas
	Verificación de los fallos detectados	¿Se verifica que los fallos detectados no se presentan en una iteración posterior?	0= no 1=si	Analista de Pruebas Desarrolladores

**Tabla 28:** Nuevas sub\_características y métricas para la categoría Gestión (MAN) y la característica Gestión de Calidad (MAN2).

**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2012).

Adicionalmente, para la categoría **Organizacional (ORG)** y la característica **Lineamientos Organizacionales (ORG1)**, se crearon seis (6) métricas, las cuales se pueden observar en la Tabla 29.

Característica	Métrica	Pregunta	Formulación	Dirigido a
Lineamientos Organizacionales	Selección de metodología	¿Se selecciona alguna metodología para el proceso de pruebas?	0= no 1=si	Líder de Pruebas
	Estándar para la ejecución de las pruebas	¿Se incluye algún estándar para la ejecución del proceso de	0= no 1=si	Líder de Pruebas

(ORG1)		pruebas?		
	Definición de fases	¿Se definen las fases a seguir en las pruebas de software?	0= no 1=si	Líder de Pruebas
	Duración de las fases	¿Se definen la duración de cada una de las fases del proceso de pruebas?	0= no 1=si	Líder de Pruebas
	Definición de artefactos	¿Se definen los principales artefactos del proceso de pruebas?	0= no 1=si	Líder de Pruebas
	Definición de canales de comunicación	¿Se definen los canales de comunicación que se deben seguir durante el proceso de pruebas?	0= no 1=si	Líder de Pruebas

**Tabla 29:** Nuevas métricas para la categoría Organizacional (ORG) y la característica Lineamientos Organizacionales (ORG 1)

**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2012).

Ahora bien, para la categoría **Organizacional (ORG)** y la característica **Evaluación del proceso (ORG2)**, se crearon cuatro (4) métricas, las cuales se pueden observar en la Tabla 30.

Característica	Métrica	Pregunta	Formulación	Dirigido a
Evaluación del Proceso (ORG 2)	Verificación del número de casos de prueba especificados	¿Se verifica el número de casos de prueba especificados?	0= no 1=si	Líder de Pruebas Analista de Pruebas
	Verificación del número de casos de prueba ejecutados	¿Se verifica el número de casos de prueba ejecutados?	0= no 1=si	Líder de Pruebas Analista de Pruebas
	Verificación del número de casos de prueba con incidencias	¿Se verifica el número de casos de prueba con incidencias?	0= no 1=si	Líder de Pruebas Analista de Pruebas
	Verificación del número de casos de prueba exitosos	¿Se verifica el número de casos de prueba exitosos?	0= no 1=si	Líder de Pruebas Analista de Pruebas

**Tabla 30:** Nuevas métricas para la categoría Organizacional (ORG) y la característica Evaluación del Proceso (ORG2).

**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2012).

Además, para la categoría **Organizacional (ORG)** y la característica **Reuso (ORG3)**, se crearon cinco (5) métricas, las cuales se pueden observar en la Tabla 31.

Característica	Métrica	Pregunta	Formulación	Dirigido a
Reuso (ORG3)	Documentación de las soluciones	¿Se documentan las soluciones adaptadas en determinados tipos de aplicaciones?	0= no 1=si	Analista de Pruebas
	Determinar artefactos reutilizables	¿Se establecen los artefactos reutilizables?	0= no 1=si	Líder de Pruebas
	Repositorio de material de pruebas	¿Se cuenta con un repositorio de material de pruebas?	0= no 1=si	Analista de Pruebas
	Garantía de permanencia de la información en el repositorio de pruebas	¿Se garantiza la permanencia de la información en el repositorio de pruebas?	0= no 1=si	Líder de Pruebas
	Garantía de accesibilidad de la información del repositorio de pruebas	¿Se garantiza accesibilidad a la información contenida en el repositorio de pruebas?	0= no 1=si	Líder de Pruebas

**Tabla 31:** Nuevas métricas para la categoría Organizacional (ORG) y la característica Reuso (ORG3).

**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2012).

Por otro lado, se propuso una nueva categoría **Adiestramiento (TRA)**, la cual evalúa los procesos a nivel de capacitación del personal en cada uno de las procesos actividades desarrolladas durante la ejecución del proceso de pruebas en la

organización. Dicha categoría está formada por 3 características y 10 métricas, las cuales se pueden observar en la Tabla 32.

- **Planificación (TRA1)**, formada por tres (3) métricas.
- **Ejecución (TRA2)**, formada por tres (3) métricas.
- **Actividades de Cierre (TRA3)**, formada por cuatro (4) métricas.

Sub- Característica	Métricas	Pregunta	Formulación	Dirigido a
Planificación (TRA1)	Análisis de necesidades de entrenamiento del personal	¿Se analizan las necesidades de entrenamiento del personal del área de pruebas?	0= no 1=si	Líder de Pruebas
	Diseño de planes de capacitación	¿Se diseñan planes de capacitación para el personal del área de pruebas?	0= no 1=si	Líder de Pruebas
	Determinación de recursos necesarios para la implementación del plan	¿Se determinan los recursos necesarios para implementar el plan de capacitación?	0= no 1=si	Líder de Pruebas
Ejecución (TRA 2)	Desarrollo de manuales	¿Se desarrolla manuales de entrenamiento?	0= no 1=si	Líder de Pruebas

	Contenidos ajustados a las actividades realizadas en el proceso de pruebas	¿Los contenidos impartidos, se ajustan a las actividades realizadas en el proceso de pruebas?	0= no 1=si	Analista de Pruebas
	Personal capacitado	¿El personal encargado de impartir el entrenamiento posee experiencia en el área de pruebas?	0= no 1=si	Analista de Pruebas
Actividades de Cierre (TRA 3)	Información del personal capacitado	¿Se almacena la información del personal adiestrado?	0= no 1=si	Líder de Pruebas
	Registro de cursos impartidos	¿Se registra los cursos impartidos?	0= no 1=si	Líder de Pruebas
	Entrega de certificados	¿Se entrega certificado al participante en la capacitación?	0= no 1=si	Líder de Pruebas
	Evaluación del beneficio del adiestramiento	¿Se evalúa el beneficio del adiestramiento al personal?	0= no 1=si	Líder de Pruebas

**Tabla 32:** Nueva característica Adiestramiento (TRA) y métricas para el Modelo Sistémico de Calidad para el proceso de pruebas de software (MOSCP).

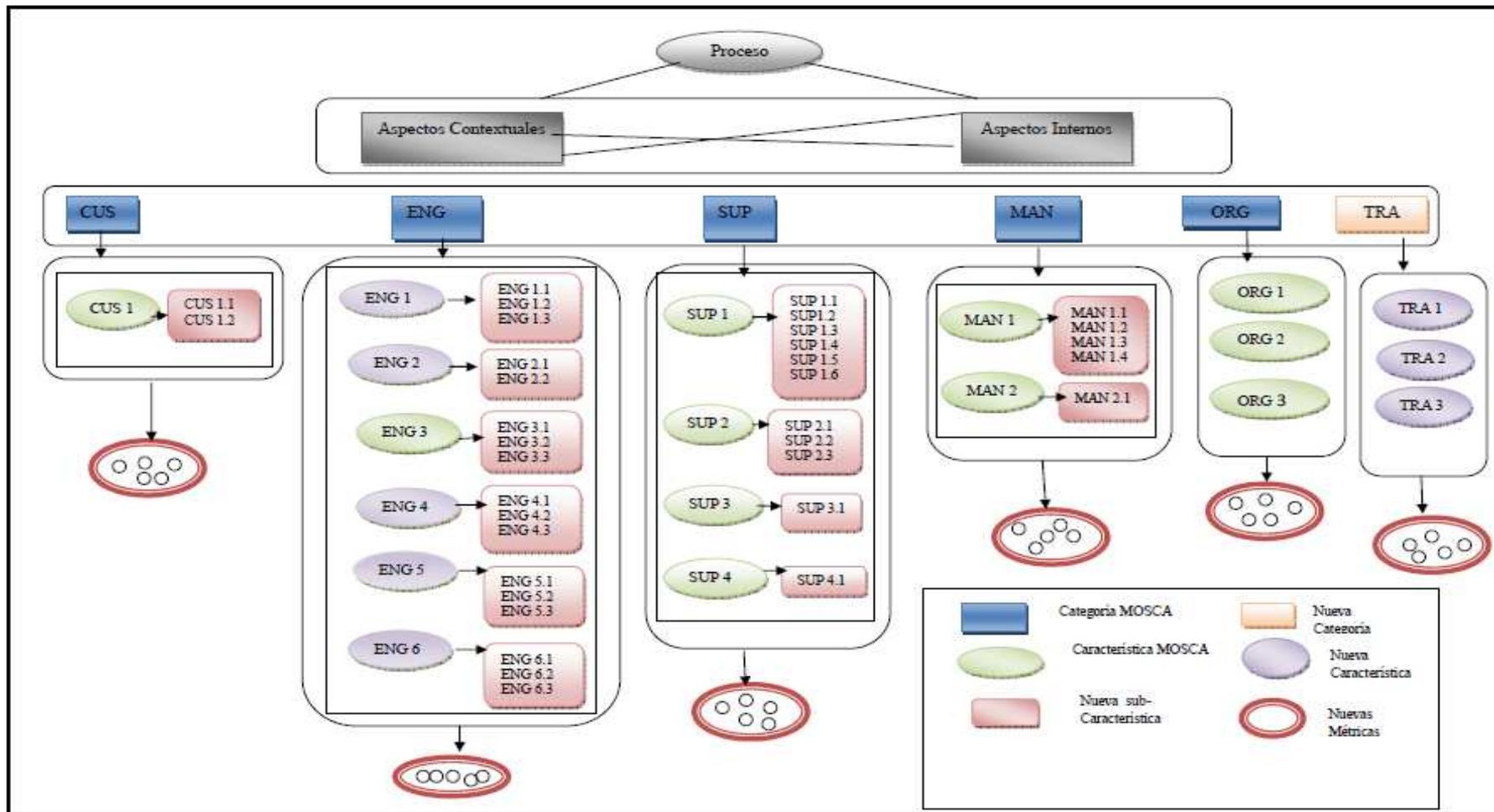
**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2012).

Como puede observarse, en el modelo propuesto se incluyó una nueva categoría, así como un conjunto de características, sub características y nuevas métricas, las cuales permiten evaluar si un proceso de pruebas de software se está realizando tomando en cuenta la calidad, y lo que es aún más importante teniendo en cuenta la visión sistémica.

Resumiendo, se tiene que la propuesta del Modelo Sistémico de Calidad para el Proceso de Pruebas (MOSCP), consta de los siguientes elementos (Ver Figura 16):

- 6 Categorías.
- 19 Características.
- 35 Sub-Características.
- 125 Métricas.



**Figura 16:** Propuesta del Modelo Sistemico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

Seguidamente, se describe otro paso de la fase Tomar la acción de la metodología empleada, la cual consiste en preparar el entorno para evaluar la propuesta.

#### **4.2 Análisis del Contexto**

En esta fase se determinan las especificaciones para implementar el Modelo Sistémico de calidad para el proceso de pruebas (MOSCP).

Según Kitchenham (1996), el análisis de contexto se realiza siguiendo los siguientes criterios: el contexto de la evaluación, la naturaleza del impacto esperado, la naturaleza del objeto de evaluación, el alcance del impacto del objeto de la evaluación, la madurez del objeto de la evaluación, el tiempo de aprendizaje y la madurez de la organización evaluadora.

Seguidamente, se describen cada uno de los criterios anteriormente mencionados:

- a) **Contexto de la Evaluación:** El presente criterio está conformado por los siguientes elementos:
  - Modelo Sistémico de calidad para el proceso de pruebas de software: es el objeto a evaluar.
  - Investigador: Es el responsable de recolectar y analizar los resultados de la evaluación.
  - IUTY: Instituto Universitario de Tecnología de Yaracuy, es la organización donde se evaluará el modelo propuesto, utilizando como referencia, el proceso de pruebas de software implementado en la actualidad.
  
- b) **Naturaleza del impacto esperado:** En el presente trabajo de investigación la naturaleza del impacto esperado es cuantitativa ya que con la evaluación del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software

(MOSCP), se busca mejorar la calidad del proceso de pruebas en las organizaciones.

c) **Naturaleza del objeto de evaluación:** El Modelo Sistémico de calidad para el Proceso de Pruebas de Software (MOSCP), es un modelo que permitirá a las organizaciones garantizar la calidad sistémica del proceso de pruebas de software.

d) **Análisis del Impacto del Objeto de Evaluación:**

Kitchenham (1996), indica que el alcance del impacto del método o herramienta tiene dos dimensiones principales:

- **El detalle del producto:** El cual identifica si el método se aplica al desarrollo o mantenimiento del producto software en su conjunto o a las piezas individuales del mismo.
- **El grado de impacto:** Identifica la probabilidad que el efecto del método se siente sobre el ciclo de vida del producto o proyecto.

En este orden de ideas, con la determinación de la calidad sistémica del proceso de pruebas de software, a través del Modelo Sistémico de Calidad para el proceso de pruebas (MOSCP), se consideró la dimensión grado de impacto, debido a que se obtiene una visión holística de la calidad del proceso de pruebas de software utilizado en la organización, lo cual se debe emplear para introducir cambios que permitan elevar el nivel de calidad en dicho proceso.

e) **Madurez del objeto de la evaluación:** El Modelo Sistémico de Calidad para el proceso de pruebas se encuentra en la fase de propuesta, por lo tanto no está en uso en proyectos comerciales.

f) **Tiempo de Aprendizaje:** Kitchenham (1996), clasifica el tiempo de aprendizaje del objeto de evaluación de la siguiente manera:

- **Tiempo requerido para comprender los principios del Modelo:** el tiempo requerido para comprender los principios básicos del modelo es medio, ya que los participantes en la evaluación deben poseer conocimientos en los conceptos relacionados al proceso de pruebas de software, con especial énfasis en los siguientes aspectos: cliente, ingeniería, soporte, mantenibilidad, organizacionales y entrenamiento, así como entender la importancia de la calidad sistémica.
  - **Tiempo para ser competente en su uso:** Después de comprender los principios básicos del modelo y valorar la importancia de la calidad desde el punto de vista sistémico, el tiempo requerido para ser un experto en el uso del modelo es medio.
- g) **Madurez de la organización:** La organización evaluadora del modelo, en el caso de este estudio, posee un equipo altamente capacitado en el proceso de desarrollo de software, se contó con el apoyo de los actores fundamentales en el proceso de pruebas (Líder de pruebas, analista de pruebas, desarrolladores y clientes), los cuales brindaron su amplia experiencia para la evaluación del modelo.

Una vez desarrollados cada uno de los elementos que integran el Análisis del Contexto, se da por finalizada la fase seis del marco metodológico, en ella se preparó el entorno para la evaluación de la propuesta.

Seguidamente se procede a describir la fase siete del ciclo metodológico, la cual consiste en la aplicación del método DESMET con la finalidad de obtener un método fiable e imparcial para la evaluación del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

### 4.3 Aplicación del Método DESMET

La aplicación del método DESMET, es la última actividad de la fase tomar la acción, la finalidad de esta actividad es obtener un método fiable e imparcial para la evaluación del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

En este sentido, en el Marco Metodológico se describieron los nueve métodos propuestos por Kitchenham (1996), así como un conjunto de criterios que favorecen el uso de cada método de evaluación. Seguidamente, se presenta la Tabla 33, en la cual se puede observar las condiciones favorables y no favorables de cada uno de los métodos de DESMET para realizar la evaluación del modelo propuesto.

Método de Evaluación	Condiciones Favorables	Formulación		Porcentaje (%)
		Si	No	
Experimento Cuantitativo	Beneficios claramente Cuantificables		x	33.33%
	Personal disponible para participar en el experimento	x		
	Método relacionado a una sola tarea o actividad		x	
	Beneficios directamente medibles desde las salidas obtenidas	x		
	Tiempo de aprendizaje relativamente corto		x	
	Deseo de hacer independiente del contexto a MOSCP		x	
	Beneficios cuantificables en un sólo proyecto.	x		
	Beneficios cuantificables		x	

Estudio de Caso Cuantitativo	antes del retiro del producto.			60%
	Procedimientos de desarrollo estables.	x		
	Personal con experiencia en medición.	x		
	Plazos para la evaluación acordes con el tiempo de desarrollo del proyecto		x	
Encuestas Cuantitativas	Beneficios no cuantificables en un solo proyecto		x	0%
	Existencia de una base de datos que almacene la siguiente información: productividad y calidad medida por MOSCP		x	
	Experiencias con proyectos utilizando MOSCP		X	
Análisis de características por proyección	Gran número de características del Modelo (MOSCP)		x	50%
	Plazos cortos para realizar la evaluación	x		
Análisis de características por estudio de caso	Beneficios difíciles de cuantificar		x	60%
	Beneficios observables en un solo proyecto	x		
	Procedimientos de desarrollo estables.	x		
	Población de usuarios del modelo MOSCP limitada		x	
	Plazos para la evaluación acordes con el tiempo de desarrollo del proyecto	x		
Análisis de	Beneficios difíciles de cuantificar.	x		

características por experimento	Beneficios directamente medibles desde las salidas obtenidas		x	50%
	Tiempo de aprendizaje relativamente corto		x	
	Población de usuarios del Modelo MOSCP variada	x		
Análisis de características por encuesta	Beneficios difíciles de cuantificar.	x		50%
	Población de usuarios del Modelo MOSCP variada	x		
	Beneficios no observables en un solo proyecto		x	
	Experiencias con proyectos utilizando MOSCP o proyectos preparados para aprender sobre el modelo		X	
Análisis de efectos cualitativos	Disponibilidad de opiniones de expertos en evaluaciones similares.	x		25%
	Ausencia de procedimientos de desarrollo estables.		x	
	Requisitos para mezclar y combinar el método (MOSCP)		x	
	Interés en la evaluación de métodos genéricos		x	
Benchmarking	En la aplicación del modelo MOSCP se requiere la intervención humana de manera	x		

	intensiva			50%
	Las salidas del método pueden ser clasificadas de acuerdo a algunos criterios de bondad presentes en el modelo		x	

**Tabla 33:** Condiciones favorables presentes y no favorables en cada uno de los métodos de DESMET.

**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2012).

Como se observa en la Tabla 33, de acuerdo con los porcentajes obtenidos se visualiza que dos métodos de evaluación alcanzaron en 60%, no obstante se elige como método de evaluación para el modelo propuesto: Análisis de Características por estudio de caso, ya que el mismo analiza las características del método luego que el mismo se ha aplicado en un proyecto real, así se aplicará el Modelo Sistemico de Calidad para el Proceso de Pruebas de Software (MOSCP), en el proceso de pruebas del Instituto Universitario de Tecnología del Estado Yaracuy (IUTY).

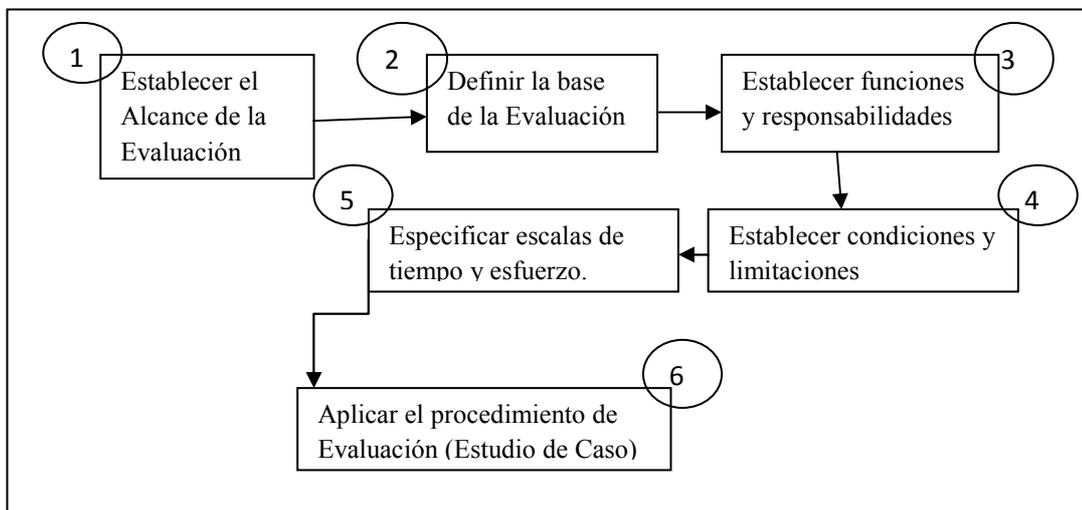
Para continuar con las fases del ciclo metodológico se procede a describir la Fase Evaluar.

#### **4.4 Evaluación a través del método arrojado por DESMET**

Esta es la primera actividad de la fase Evaluar, la cual consiste en evaluar el modelo siguiendo un criterio apropiado. Así, de acuerdo al resultado obtenido en la sección anterior el método de evaluación sugerido por DESMET, es el Análisis de Características por estudio de caso.

En este sentido, Kitchenham (1996) propone en la evaluación de Análisis de características por caso de estudio, se debe realizar el proceso de planificación, para ello se deben realizar las siguientes actividades (Ver Figura 17):

- Establecer el Alcance de la Evaluación.
- Definir la base de la Evaluación.
- Establecer funciones y responsabilidades.
- Establecer condiciones y limitaciones.
- Especificar escalas de tiempo y esfuerzo.
- Aplicar el procedimiento de Evaluación (Estudio de Caso).



**Figura 17:** Actividades en el proceso de planificación del Análisis de características.

**Fuente:** Kitchenham (1996).

**Autor:** Elaboración Propia (2012).

Para continuar, se describen cada uno de los pasos correspondientes al proceso de planificación de Análisis de Características.

- a) Establecer el Alcance de la Evaluación:** El propósito de la evaluación de la propuesta del Modelo Sistémico de Calidad para el proceso de pruebas(MOSCP), en el proceso de pruebas seguido en el Instituto Universitario de Tecnología del Estado Yaracuy(IUTY), es determinar si las actividades realizadas en dicho proceso se realiza siguiendo una visión sistémica.

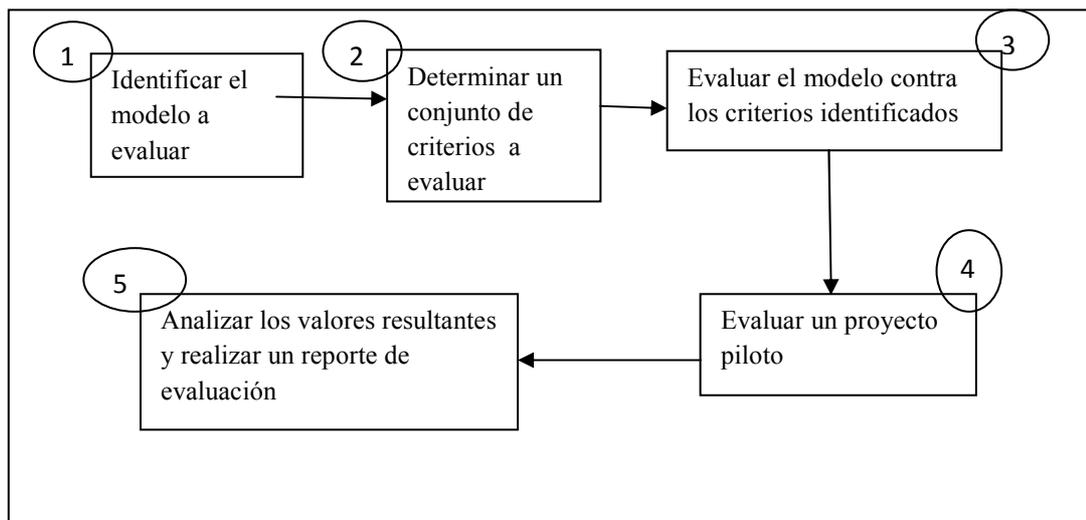
- b) Definir la base de la Evaluación:** Con respecto a este punto, Kitchenham (1996), señala que la profundidad de la investigación requerida por los participantes está relacionada con el nivel de confianza requerido en las evaluaciones. En este sentido, la evaluación del presente proyecto está fundamentada en la investigación documental y en el marco metodológico, los cuales se presentaron en capítulos anteriores.
- c) Establecer funciones y responsabilidades:** En cuanto a los roles relacionados con la evaluación, Kitchenham (1996) indica que son los siguientes:
- **El Patrocinante:** La evaluación del Modelo se logró gracias al apoyo y experiencia del equipo de la unidad de Informática del Instituto Universitario de Tecnología del Estado Yaracuy (IUTY).
  - **El Evaluador:** Asumido por el autor de la investigación, en este caso realizó las siguientes actividades: Preparación del plan de evaluación, identificación de las características candidatas a evaluar, recopilación y análisis de los resultados y preparación del informe final.
  - **Los Evaluadores del Método:** representado por las personas que califican cada característica del modelo propuesto, en el caso de la presente investigación este rol lo asumió el equipo de la unidad de Informática del Instituto Universitario de Tecnología del Estado Yaracuy (IUTY).
- d) Establecer condiciones y limitaciones:** Las limitaciones presentes en el proceso de evaluación están relacionadas al factor tiempo, en primer lugar la disponibilidad de tiempo por parte del equipo evaluador, por otro lado, el tiempo estipulado para la culminación del trabajo de grado, lo cual limita la duración del proceso de evaluación.

e) **Especificar escalas de tiempo y esfuerzo:** A continuación se lista cada una de las actividades necesarias para cumplir con el proceso de evaluación, así como el tiempo dedicado en cada una de ellas.

- Preparación del plan de evaluación: 4 días
- Elaboración de los instrumentos de evaluación: 8 días
- Solicitud de entrevista con el equipo de la Unidad de Informática del Instituto Universitario de Tecnología del Estado Yaracuy (IUTY) para proceder a la evaluación: 3 días
- Evaluación de las características: 10 días
- Recopilación y análisis de los resultados: 6 días
- Elaboración de las conclusiones y Recomendaciones: 3 días.

f) **Aplicar el procedimiento de Evaluación (Estudio de Caso)**

El procedimiento de evaluación seleccionado fue el estudio de Caso, de acuerdo a las actividades señaladas en el marco metodológico un estudio de caso implica las siguientes actividades (Ver Figura 18).



**Figura 18:** Actividades en el proceso de un estudio de caso.

**Fuente:** Kitchenham (1996).

**Autor:** Elaboración Propia (2012).

A continuación, se describen cada una de las actividades que se llevaron a cabo en el estudio de caso de la presente investigación:

- a) **Identificar el modelo a evaluar:** El objeto de la evaluación, es el Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP), el cual tiene por objetivo proveer una herramienta a las organizaciones que buscan la calidad sistémica en el proceso de pruebas.
- b) **Determinar un conjunto de criterios a evaluar:** Para evaluar de manera segura el modelo propuesto incluyendo cada una de las categorías, características, sub-características y métricas, se hace necesario establecer criterios de evaluación de las características que van de lo general: El Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP), a lo específico las métricas del modelo. En este sentido, Kitchenham (1996), señala que existen dos tipos de criterios, los cuales se describen a continuación:
  - **Simples:** Se utilizan cuando el criterio está presente o ausente dentro del contexto de la evaluación. Éstas son evaluadas utilizando una escala nominal; es decir haciendo uso de un SI/NO.
  - **Compuestas:** Donde el grado de soporte que ofrece el método debe ser medido utilizando una escala ordinal.

En la presente investigación se utilizaron los criterios simples, ya que permiten al evaluador obtener un criterio claro de aceptación. (Ver Tabla 34).

Criterio	Descripción
1	Significa que el modelo o métrica posee el criterio establecido.
0	Significa que el modelo o métrica no posee el criterio establecido.

**Tabla 34:** Definición de la escala utilizada para evaluar los criterios generales y específicos.

**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2012)

Los criterios más generales evalúan el modelo en un nivel macro, a saber en el subárbol Proceso como un todo y sus categorías, su definición conceptual se puede visualizar en la Tabla 35.

Criterio General	Descripción
Pertinencia del modelo propuesto	Se refiere a si el modelo de calidad propuesto es pertinente o no dentro del proceso de especificación de la calidad del proceso de pruebas de software.
Complejidad de las categorías involucradas	Se refiere a si las Categorías: <ul style="list-style-type: none"> <li>• Cliente-Proveedor (CUS).</li> <li>• Ingeniería (ENG).</li> <li>• Soporte (SUP).</li> <li>• Gestión (MAN).</li> <li>• Organizacional (ORG).</li> <li>• Entrenamiento (TRA).</li> </ul> Dan cobertura total a los aspectos de calidad del proceso de pruebas de software.
Adecuación al contexto	Se refiere a si la especificación de calidad del modelo propuesto es adecuada en el contexto de la evaluación.
Precisión del nivel de calidad especificado en el modelo	Se refiere a si la calidad especificada por el modelo propuesto en el proyecto piloto fue precisa.

**Tabla 35:** Criterios generales a evaluar en el Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2012).

Una vez especificados los criterios generales fue necesario establecer un conjunto de criterios que permitan evaluar las métricas del modelo. Los criterios a considerar se describen en la Tabla 36.

Criterio Especifico	Descripción
Pertinencia de la métrica	Se refiere a si una métrica es adecuada para medir la existencia o no de la característica donde se encuentra.
Factibilidad de la métrica	Se refiere a si es factible medir la característica propuesta en la métrica dentro del contexto de evaluación.
Nivel de profundidad	Se refiere a si la métrica a verificar tiene el nivel de profundidad adecuado para que el resultado sea relevante.
Escala de la métrica	Se refiere a si la escala propuesta es adecuada para medir la métrica.

**Tabla 36:** Criterios específicos a evaluar en el Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Elaboración Propia.

**Autor:** Autor de la Investigación (2012).

Ahora bien, una vez establecidos los criterios a aplicar para evaluar el Modelo propuesto, se necesitó precisar el criterio de aceptación a utilizar, el cual se describe a continuación:

- Para que el Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP), se considere aceptable todas sus categorías deben poseer un nivel de aceptación mayor o igual a 75 %.
- Para que una característica dentro de una categoría, se considere aceptable, debe poseer un nivel de aceptación mayor o igual a 75 %.
- Para que una métrica se considere aceptable, debe poseer un nivel de aceptación mayor o igual a 75 %.

Es importante mencionar que el nivel de aceptación mayor o igual a 75% fue seleccionado por el autor de la investigación considerando que es una práctica común en la evaluación de la mayoría de los modelos de calidad.

Así, con la definición de los criterios de evaluación generales y específicos, los cuales permiten la evaluación y establecido el nivel de aceptación a utilizar, se da por culminada la actividad número dos del estudio de caso.

### **c) Evaluar el modelo contra las características identificadas**

La evaluación del modelo se dividió de la siguiente manera:

- Evaluación de los criterios generales.
- Evaluación de las métricas del modelo.

Seguidamente cada una de las evaluaciones realizadas el modelo

#### **Evaluación de los criterios generales**

Los criterios generales que se evaluaron están relacionados a las Categorías, Características y Sub-Características del modelo. En el Anexo I, se presenta en detalle el resultado de los criterios generales.

La evaluación de los criterios generales se presentará de la siguiente manera:

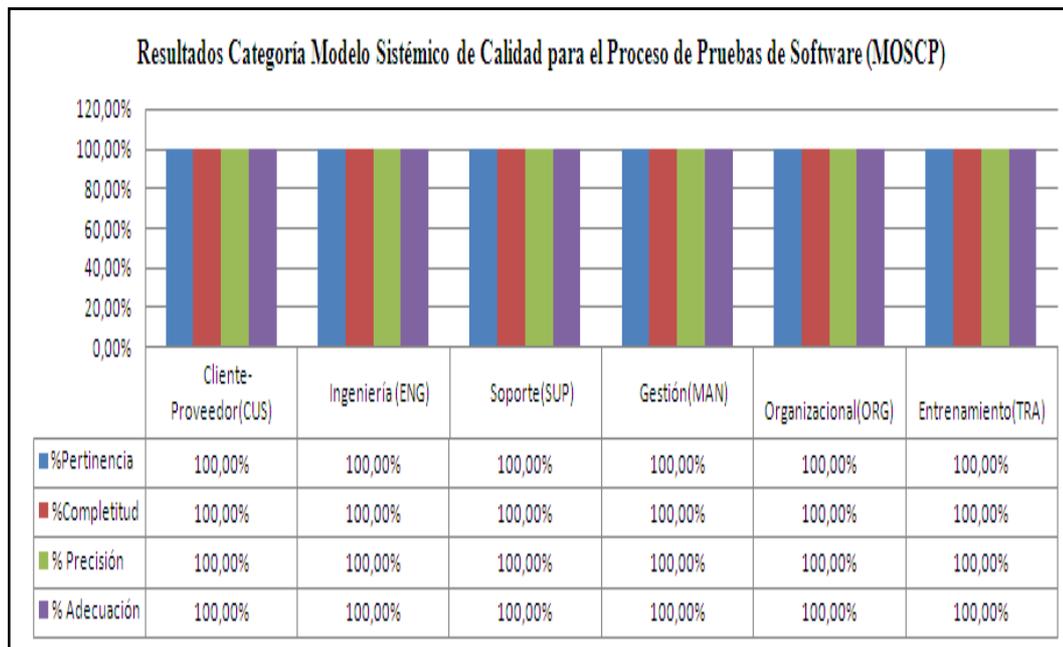
- Evaluación de las Categorías del Modelo.
- Evaluación de las Características del Modelo.
- Evaluación de las Sub-Características del Modelo.

Por otro la evaluación de los criterios específicos estará formada por:

- Evaluación de las Métricas del Modelo.

A continuación, se describen cada una de las evaluaciones realizadas.

- **Evaluación de las Categorías del Modelo:** La primera evaluación está relacionada a las Categorías del Modelo; seguidamente se presentan los resultados de dicha evaluación



**Figura 19:** Resultados Categoría Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

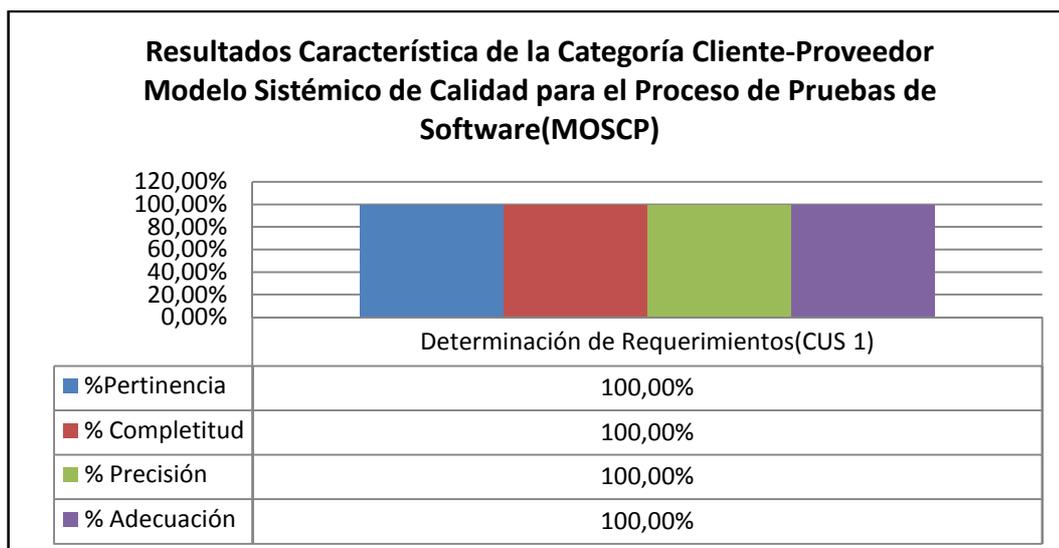
**Autor:** Elaboración Propia (2012).

En la Figura 19, se muestra los resultados de la evaluación en cuanto a las categorías del modelo propuesto. Como se puede observar, todas las categorías alcanzaron el porcentaje máximo de la evaluación, a saber 100 %.

Significa entonces, que los evaluadores consideraron que las Categorías del modelo propuesto son **precisas** dentro del proceso de especificación de la calidad del proceso de pruebas de software, son **completas** en la cobertura de calidad del proceso de pruebas, son **adecuadas** en el contexto de la evaluación, son **precisas** en el resultado alcanzado.

- **Evaluación a las Características del Modelo:** La segunda evaluación está relacionada a las Características del modelo propuesto, la cual se presentará de la siguiente manera:

**Cliente Proveedor (CUS):** Determinación de Requerimientos (CUS1), los resultados de la evaluación se pueden observar en la Figura 20.

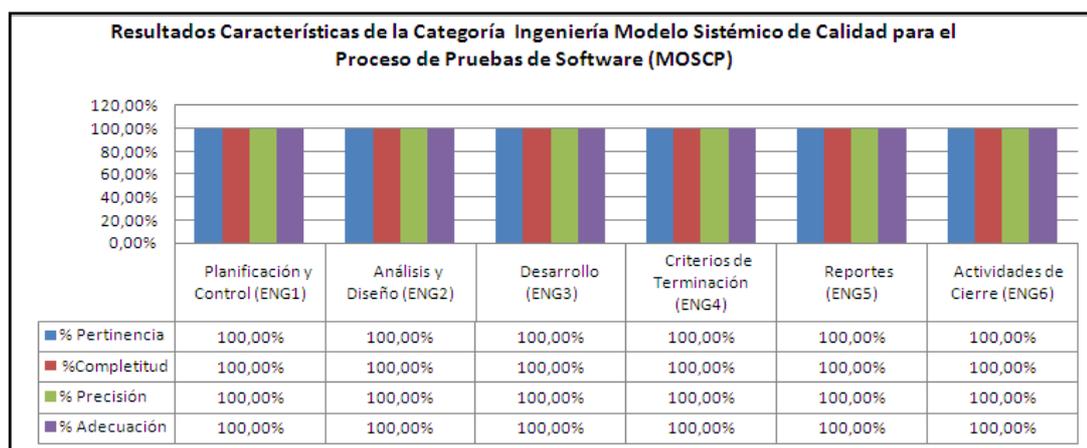


**Figura 20:** Resultados Característica Determinación de Requerimientos(CUS1) de la Categoría Cliente Proveedor (CUS) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP)

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

**Ingeniería (ENG):** Planificación y Control (ENG1), Análisis y Diseño (ENG2), Desarrollo (ENG3), Criterios de Terminación (ENG4), Reportes (ENG5), Actividades de Cierre (ENG6), en la Figura 21 se pueden apreciar los resultados de la evaluación.

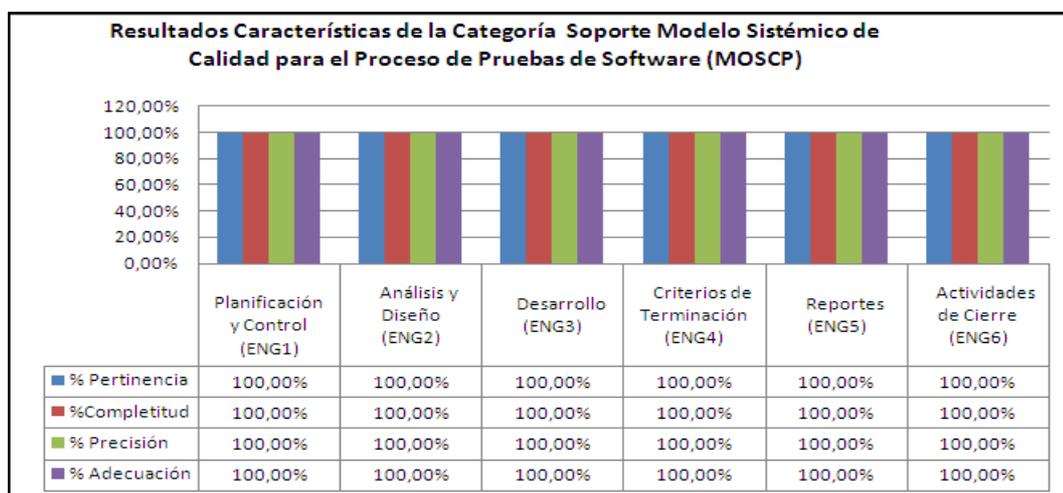


**Figura 21:** Resultados de las Características de la Categoría Ingeniería (ENG) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP)

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

**Soporte (SUP):** Documentación (SUP1), Aseguramiento de la Calidad (SUP2), Revisión Conjunta (SUP3), Auditoría (SUP4), los resultados de la evaluación se pueden apreciar en la Figura 22.

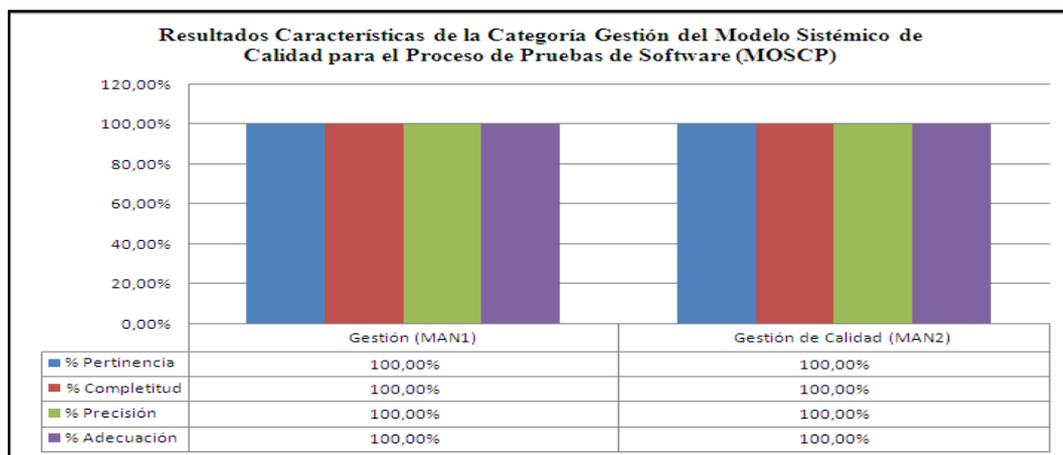


**Figura 22:** Resultados de las Características de la Categoría Soporte (SUP) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

**Gestión:** Gestión (MAN1), Gestión de Calidad (MAN2), en la Figura 23 se puede apreciar los resultados de la evaluación.

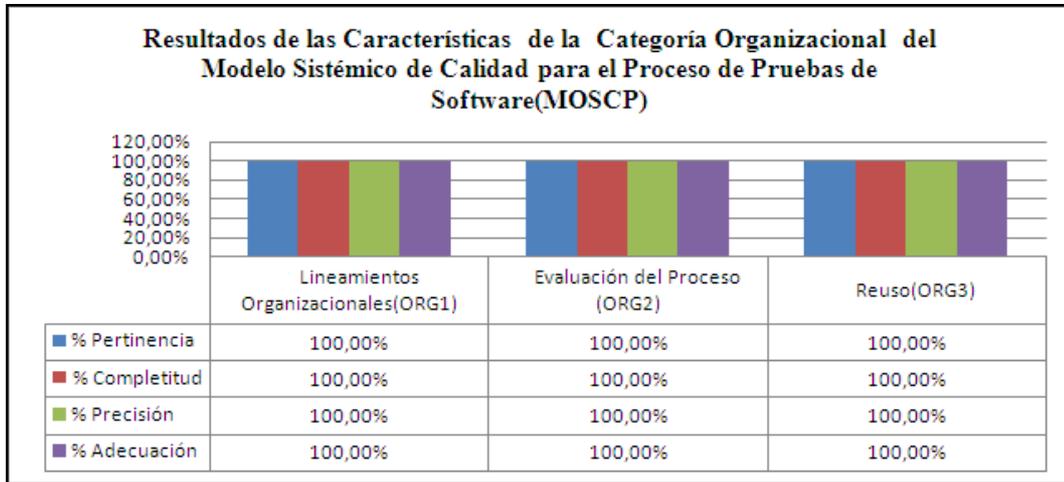


**Figura 23:** Resultados de las Características de la Categoría Gestión (MAN) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

**Organizacional (ORG):** Lineamientos Organizacionales (ORG1), Evaluación del Proceso (ORG2), Reuso (ORG3). En la Figura 24, se puede observar los resultados de la evaluación.

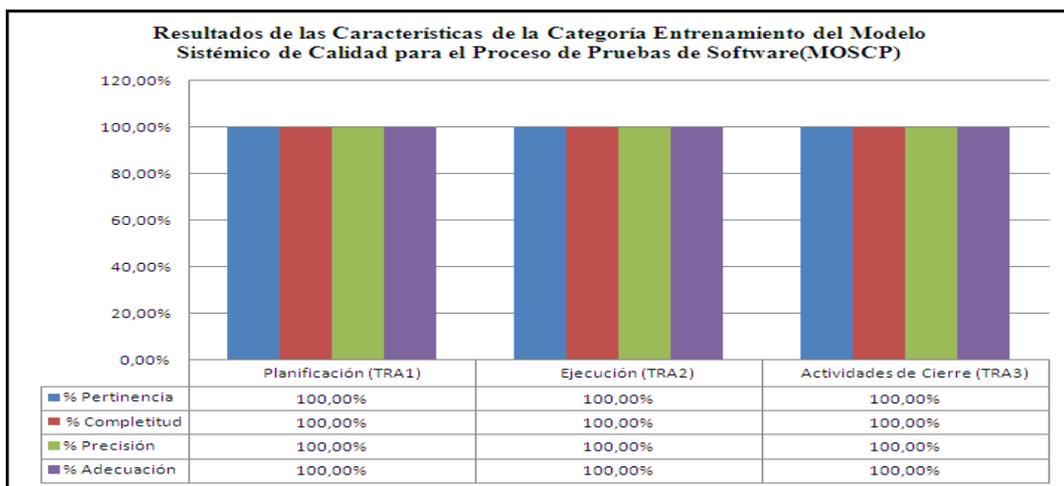


**Figura 24:** Resultados de las Características de la Categoría Organizacional (ORG) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

**Entrenamiento (TRA):** Planificación (TRA1), Ejecución (TRA2), Actividades de Cierre (TRA3), los resultados de la evaluación se pueden observar en la Figura 25.



**Figura 25:** Resultados de las Características de la Categoría Entrenamiento (TRA) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

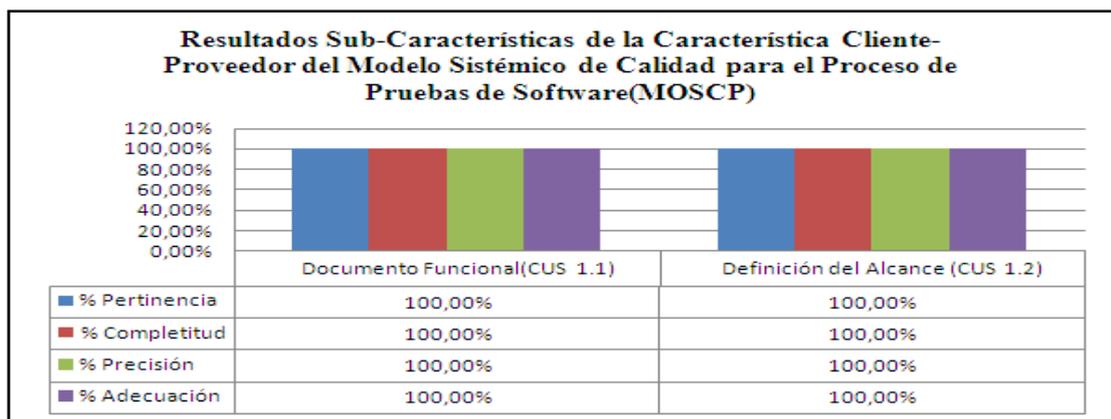
De acuerdo a los resultados presentados obtenidos en la evaluación de las características del modelo propuesto, se puede observar que las características obtuvieron un porcentaje de aceptación del 100% por parte del equipo evaluador, lo que indica que el modelo propuesto es **preciso** en el nivel de calidad especificado, **adecuado** en el contexto de la evaluación, **completo** ya que aportan una cobertura completa a los aspectos de calidad del proceso de pruebas de software y **pertinente** en su especificación de la calidad del proceso de pruebas de software.

Seguidamente, se presenta la evaluación de las sub-características del modelo propuesto, la cual constituye la tercera etapa de la evaluación.

- **Evaluación de las Sub-características del Modelo:**

En esta sección se presentarán los resultados de la evaluación de las sub-características correspondientes a cada una de las características del modelo propuesto, las cuales se listan a continuación:

**Determinación de Requerimientos (CUS1):** Documento Funcional (CUS 1.1), Definición del Alcance (CUS 1.2), los resultados de la evaluación se pueden observar en la Figura 26.

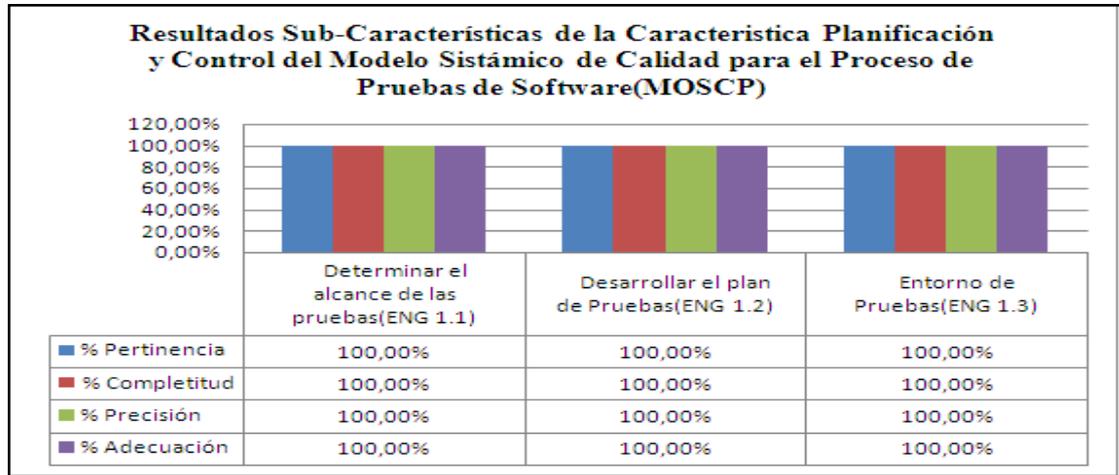


**Figura 26:** Resultados de las Sub-Características de la Característica Cliente-Proveedor(CUS) del Modelo Sistemico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

**Planificación y Control (ENG1):** Determinar el alcance de las pruebas (ENG 1.1), Desarrollar el plan de Pruebas (ENG 1.2), Entorno de Pruebas (ENG 1.3), en la Figura 27, se puede apreciar los resultados de la evaluación.

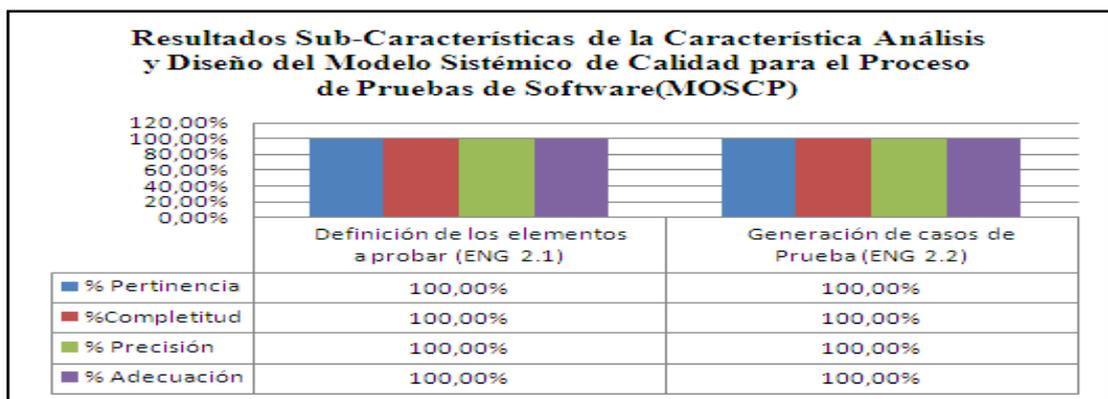


**Figura 27:** Resultados de las Sub-Características de la Característica Planificación y Control (ENG1) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

**Análisis y Diseño (ENG2):** Definición de los elementos a probar (ENG 2.1), Generación de casos de Prueba (ENG 2.2), los resultados de la evaluación se muestran en la Figura 28.

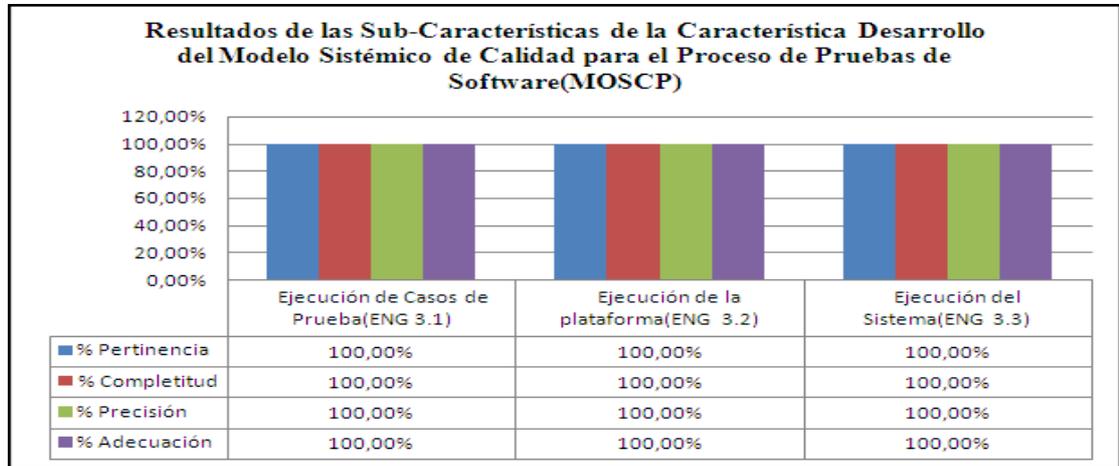


**Figura 28:** Resultados de las Sub-Características de la Característica Análisis y Diseño (ENG2) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

**Desarrollo (ENG3):** Ejecución de Casos de Prueba (ENG 3.1), Ejecución de la plataforma (ENG 3.2), Ejecución del Sistema (ENG 3.3). En la Figura 29, se muestran los resultados de la evaluación.

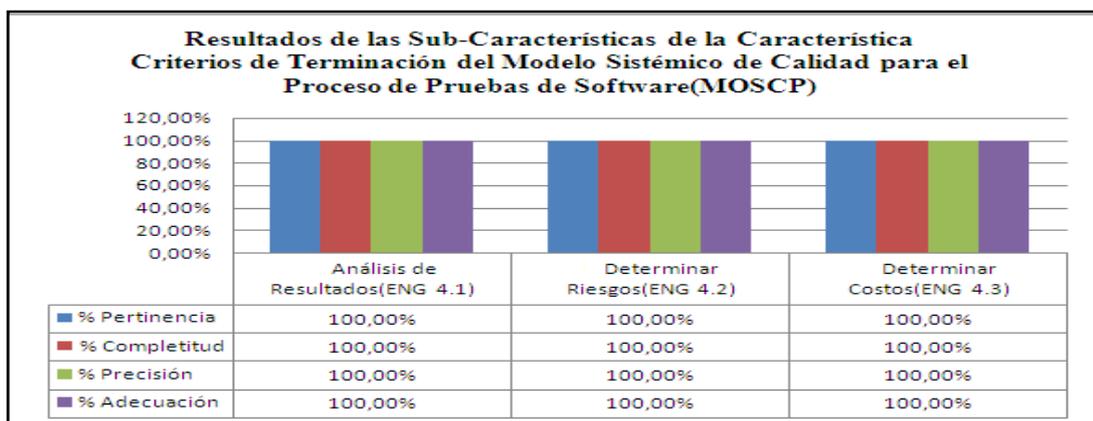


**Figura 29:** Resultados de las Sub-Características de la Característica Desarrollo (ENG3) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

**Criterios de Terminación (ENG4):** Análisis de Resultados (ENG 4.1), Determinar Riesgos (ENG 4.2), Determinar Costos (ENG 4.3). Los resultados de la evaluación se pueden apreciar en la Figura 30.

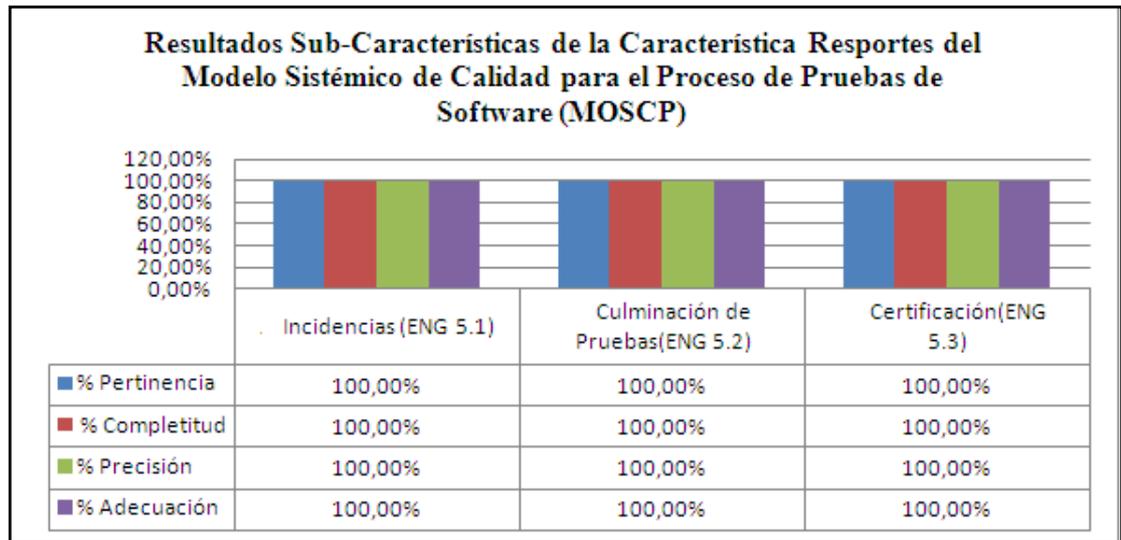


**Figura 30:** Resultados de las Sub-Características de la Característica Criterios de Terminación (ENG4) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

**Reportes (ENG5):** Incidencias (ENG 5.1), Culminación de Pruebas (ENG 5.2), Certificación (ENG 5.3). Los resultados de la evaluación se pueden apreciar en la Figura 31.

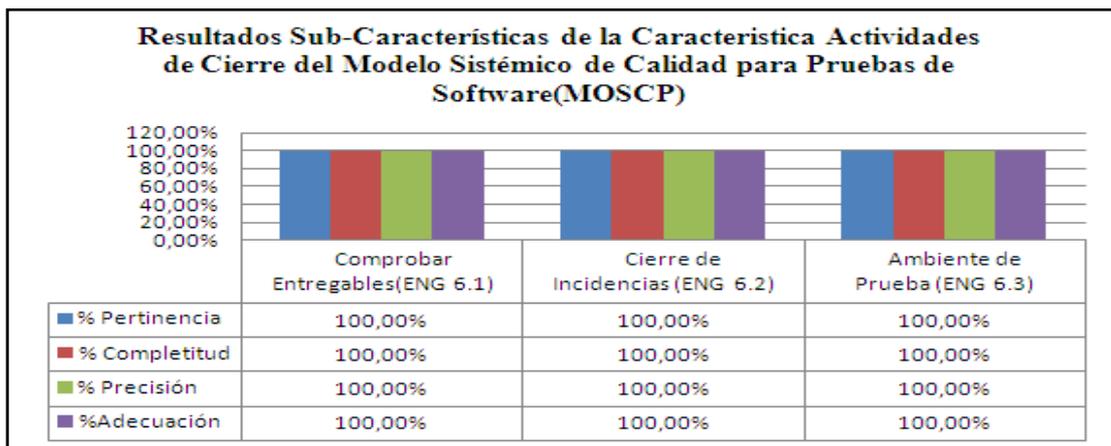


**Figura 31:** Resultados de las Sub-Características de la Característica Reportes (ENG) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

**Actividades de Cierre (ENG6):** Comprobar Entregables (ENG 6.1), Cierre de Incidencias (ENG 6.2), Ambiente de Prueba (ENG 6.3). En la Figura 32, se muestran los resultados de la evaluación.

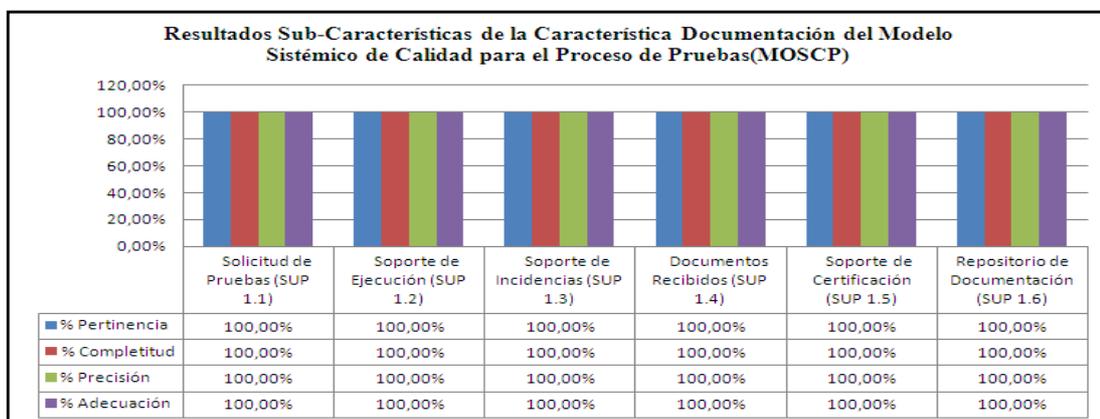


**Figura 32:** Resultados de las Sub-Características de la Característica Actividades de Cierre (ENG6) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

**Documentación (SUP1):** Solicitud de Pruebas (SUP 1.1), Soporte de Ejecución (SUP 1.2), Soporte de Incidencias (SUP 1.3), Documentos Recibidos (SUP 1.4), Soporte de Certificación (SUP 1.5), Repositorio de Documentación (SUP 1.6). Los resultados de la evaluación se presentan en la Figura 33.

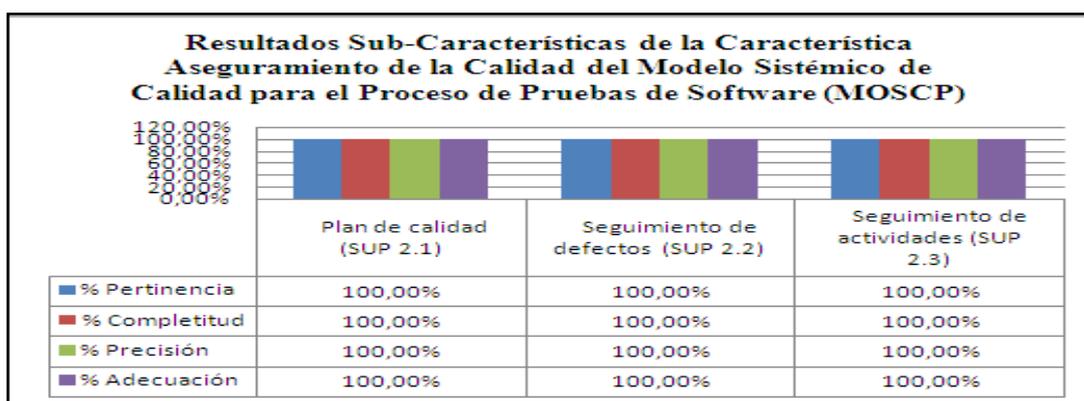


**Figura 33:** Resultados de las Sub-Características de la Característica Documentación (SUP1) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

**Aseguramiento de la Calidad (SUP2):** Plan de calidad (SUP 2.1), Seguimiento de defectos (SUP 2.2), Seguimiento de actividades (SUP 2.3). En la Figura 34, se pueden apreciar los resultados de la evaluación.

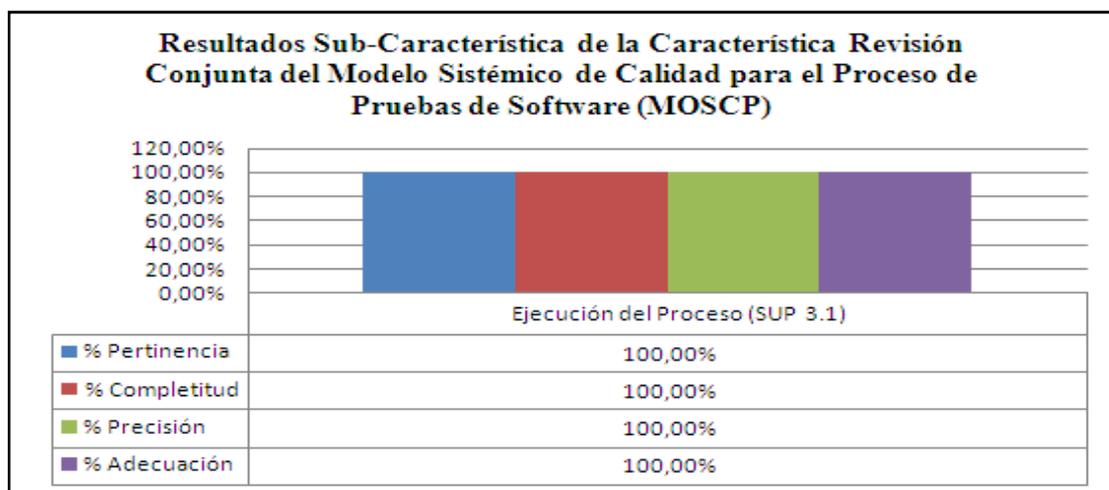


**Figura 34:** Resultados de las Sub-Características de la Característica Aseguramiento de la Calidad (SUP2) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

**Revisión Conjunta (SUP3):** Ejecución del Proceso (SUP 3.1). En la Figura 35, se pueden visualizar los resultados de la evaluación.

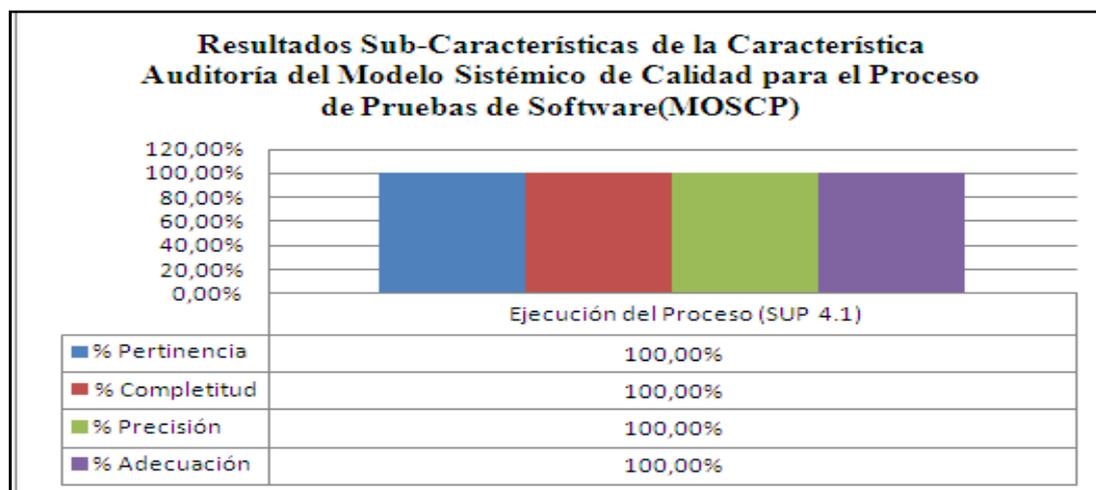


**Figura 35:** Resultados de las Sub-Características de la Característica Revisión Conjunta (SUP3) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

**Auditoría (SUP4):** Ejecución del Proceso (SUP 4.1). Los resultados de la Evaluación se pueden visualizar en la Figura 36.

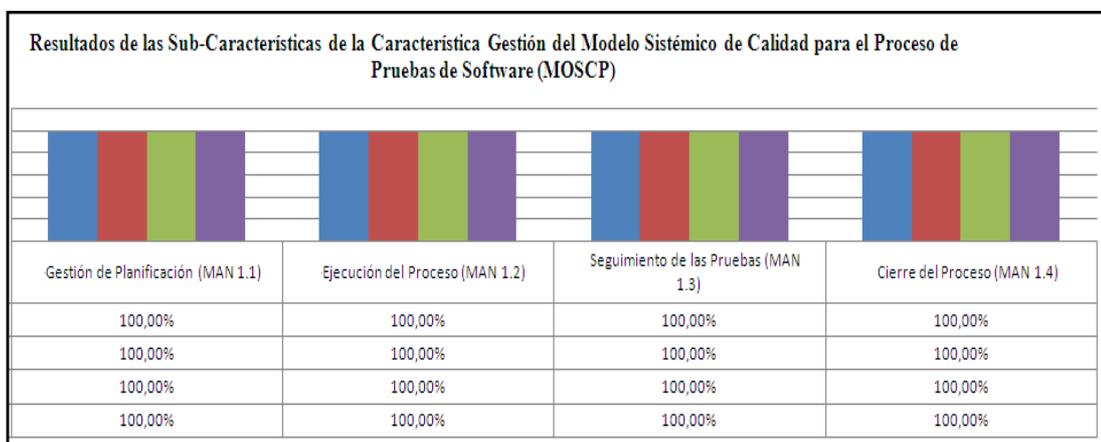


**Figura 36:** Resultados de las Sub-Características de la Característica Auditoría (SUP4) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP)

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

**Gestión (MAN1):** Gestión de Planificación (MAN 1.1), Ejecución del Proceso (MAN 1.2), Seguimiento de las Pruebas (MAN 1.3), Cierre del Proceso (MAN 1.4). En la Figura 37, se pueden apreciar los resultados de la evaluación

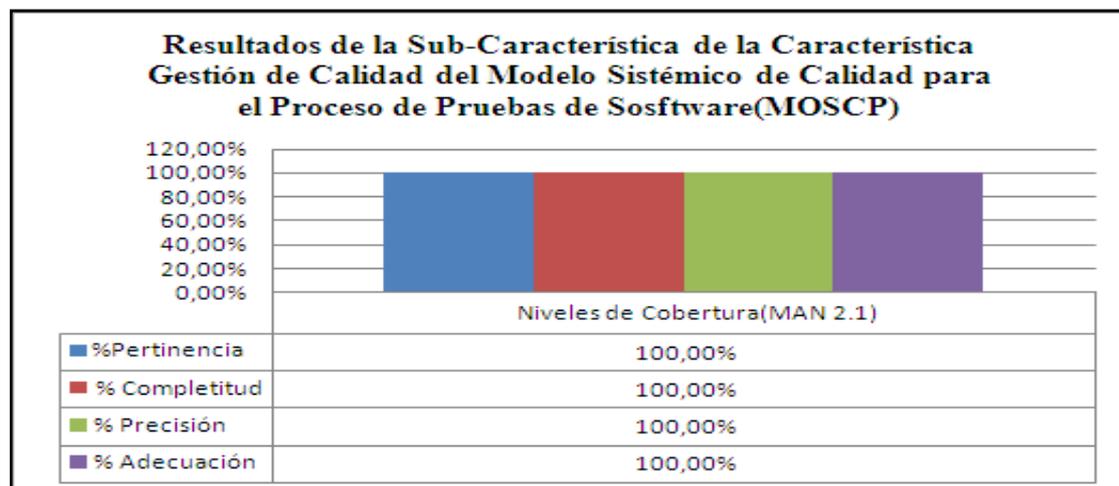


**Figura 37:** Resultados de las Sub-Características de la Característica Gestión (MAN1) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

**Gestión de Calidad (MAN2):** Niveles de Cobertura (MAN 2.1). Los resultados de la evaluación se muestran en la Figura 38.



**Figura 38:** Resultados de las Sub-Características de la Gestión de Calidad (MAN2) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

Por otro lado, es importante destacar que para las Características correspondientes a las Categorías Organizacional (ORG) y Entrenamiento (TRA), no se crearon sub-características.

Como se pudo observar en las figuras presentadas (26,27,28,29,30,31,32,33,34,35,36,37,38); las Sub-características del modelo propuesto obtuvieron un 100% de la evaluación, lo que significa que son **Pertinentes** en el proceso de especificación de calidad del software, **Completas** en cuanto a su especificación, **Adecuadas** en el contexto de la evaluación y **Precisas** dentro del nivel de calidad alcanzado.

Así, en vista que cada Categoría, Característica, Sub-Característica obtuvo un 100 % en la evaluación y de acuerdo a lo establecido en el criterio de aceptación, se considera que el Modelo Sistémico de Calidad para el Proceso de Pruebas de Software posee un nivel de calidad aceptable en lo relacionado a los criterios generales del modelo.

Seguidamente, se presenta la evaluación de los criterios específicos del modelo, a saber, evaluación de las métricas.

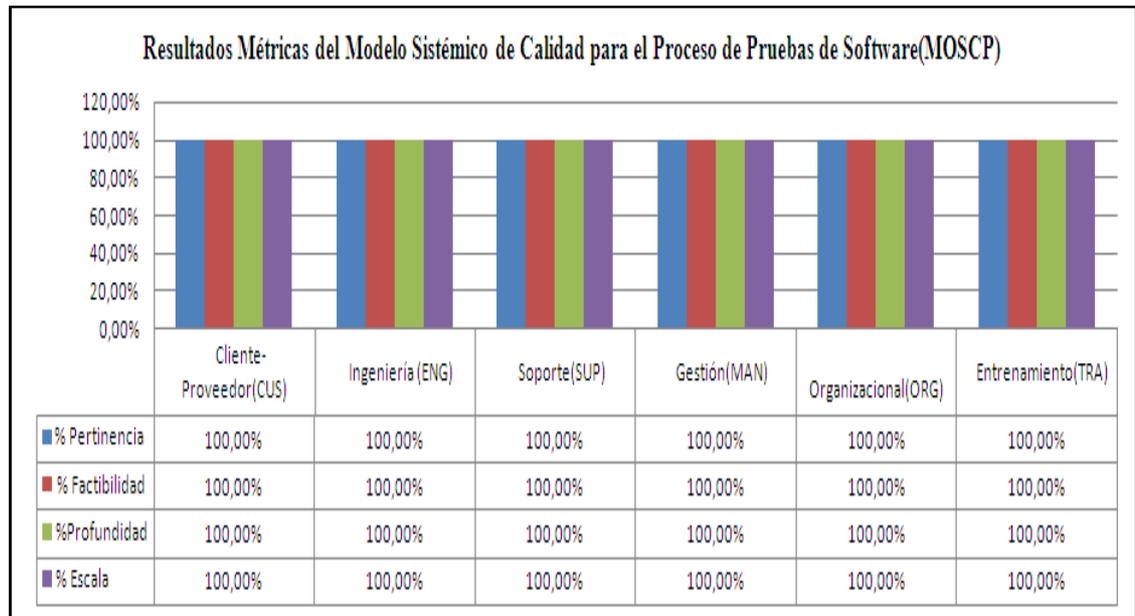
- **Evaluación de las Métricas del Modelo**

En este apartado, se presenta la evaluación de las métricas del modelo. Los resultados de dicha evaluación se pueden apreciar en la Figura 39.

En el Anexo II, se presenta en detalle el resultado de la evaluación de las métricas del modelo propuesto.

Como se puede, observar en la Figura 37, los evaluadores consideraron que las métricas de las categorías del modelo poseen 100% en los criterios establecidos: **Pertinentes**, es decir adecuadas para medir la existencia de la característica **Factibles**, de evaluar en el contexto que se aplicaron, **Profundidad**, adecuada para que el resultado sea relevante, **Escala**, apropiada para medir la métrica.

De acuerdo a los resultados de la evaluación y según el criterio de aceptación establecido, se considera que el modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP), posee un nivel de calidad aceptable.



**Figura 39:** Resultados Métricas del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

Seguidamente se procede a describir la actividad número cuatro en el proceso de un estudio de caso (Ver Figura 18).

#### **d) Evaluar un proyecto piloto**

Para dar inicio a esta actividad, en primer lugar se debe seleccionar un proyecto piloto para la evaluación.

Para la selección del proyecto piloto, se consideraron los siguientes criterios:

- **Tamaño de la Organización de media a grande**

La organización seleccionada fue el Instituto Universitario de Tecnología del Estado Yaracuy (IUTY), la cual presenta un tamaño medio debido a que actualmente se cuenta con una población estudiantil de 35000, incluyendo

estudiantes regulares y egresados. Además, el Instituto cuenta con 600 empleados interesados en la calidad del sistema.

- **El proyecto debe ser de alta relevancia para la organización.**

La organización seleccionó para la evaluación el proceso que se sigue para migrar el Sistema de Inscripción para el Control de Actividades Académicas (SICAA), el cual permite la gestión de las actividades académicas: Incorporación de estudiantes, Egresos de estudiantes de la Institución, Control de Notas. Es importante destacar, que la institución busca contar con una referencia en materia de calidad del software, específicamente en el proceso de pruebas de software.

- **Desarrollo del proyecto por dos o más personas**

El proceso de desarrollo del Sistema de Inscripción para el Control de Actividades Académicas (SICAA), intervienen aproximadamente 9 personas. Los roles involucrados dentro del proceso son: Desarrolladores, Cliente, Líder, Analista de Pruebas.

- **Colaboración por parte de la Organización**

La organización debe estar dispuesta a colaborar durante la aplicación del modelo y posterior a su aplicación, en este sentido, los integrantes del equipo de la unidad de informática del IUTY, están dispuestos a prestar su experiencia en todos los aspectos relacionados al proceso de evaluación.

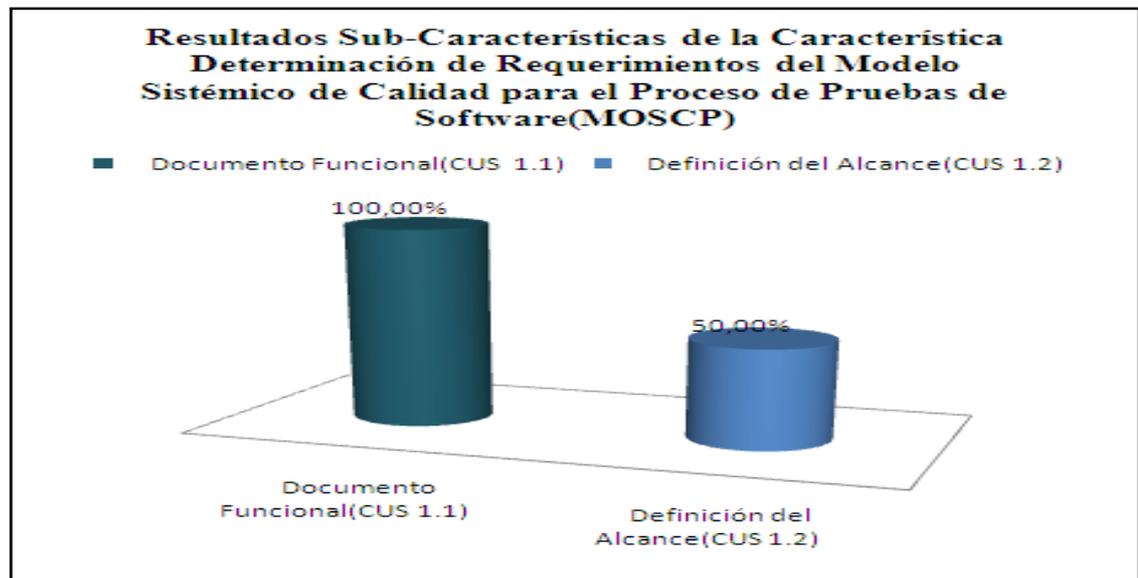
Una vez seleccionado el proyecto piloto, se procede a presentar la evaluación del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP), en el proceso de pruebas seguido por el equipo de la unidad de informática del Instituto Universitario de Tecnología del Estado Yaracuy (IUTY), en el desarrollo del Sistema de Inscripción para el Control de Actividades Académicas (SICAA).

Los encargados de la evaluación fueron los integrantes del equipo mencionados anteriormente, los resultados de la evaluación se pueden apreciar en detalle en el Anexo II.

Seguidamente se presentan las evaluaciones realizadas.

- **Evaluación de las Sub-características del Modelo en el Proceso de Pruebas de Software del desarrollo del Sistema de Inscripción para el Control de Actividades Académicas (SICAA)**

En las Figuras 40,41,42,43,44,45,46,47,48,48,49,50 se muestran los resultados de la evaluación del modelo por sub-características.



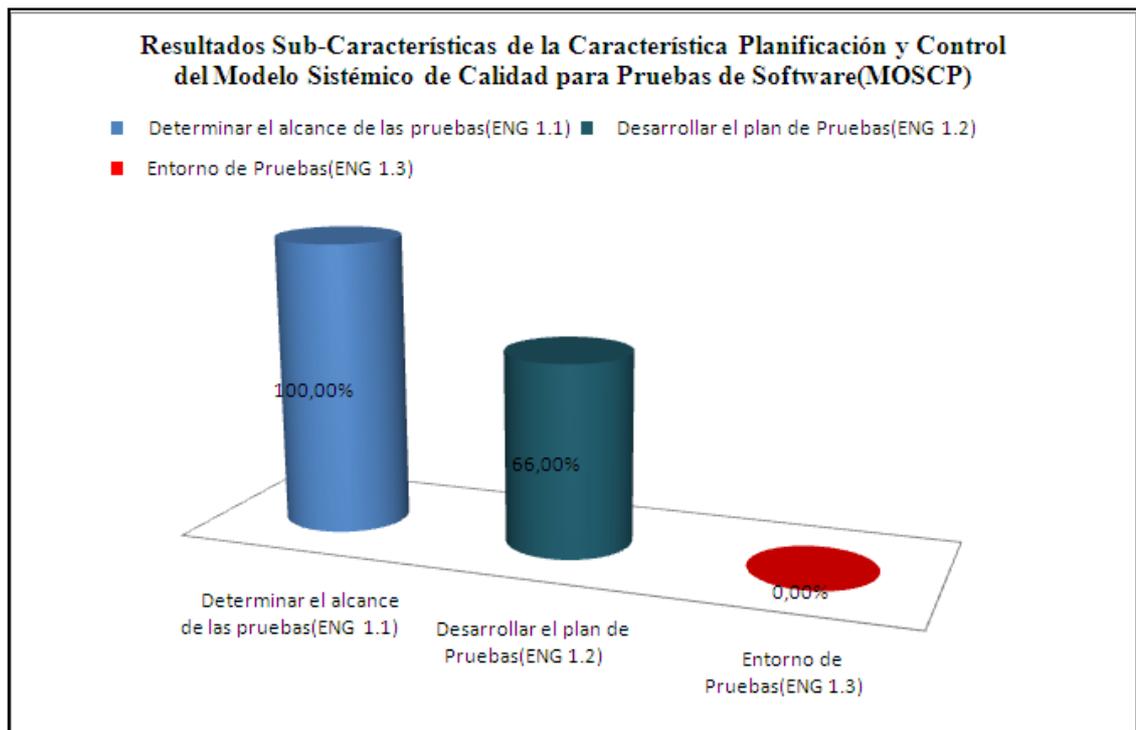
**Figura 40:** Resultados Sub-Características de la Característica Determinación de Requerimientos (CUS1) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

En la Figura 40, se observa la evaluación de las sub-características correspondientes a la Característica Determinación de Requerimientos (CUS1), como elementos resaltantes se pueden indicar los siguientes:

- La Subcaracterística Documento Funcional (CUS 1.1), obtuvo un 100% del porcentaje de evaluación, lo que significa que está dentro del criterio de aceptación (75%).
- La Subcaracterística Definición del Alcance (CUS 1.2), alcanzó el puntaje de 50%, lo cual es menor a 75%, por lo tanto esta subcaracterística no se encuentra dentro del criterio de aceptación.



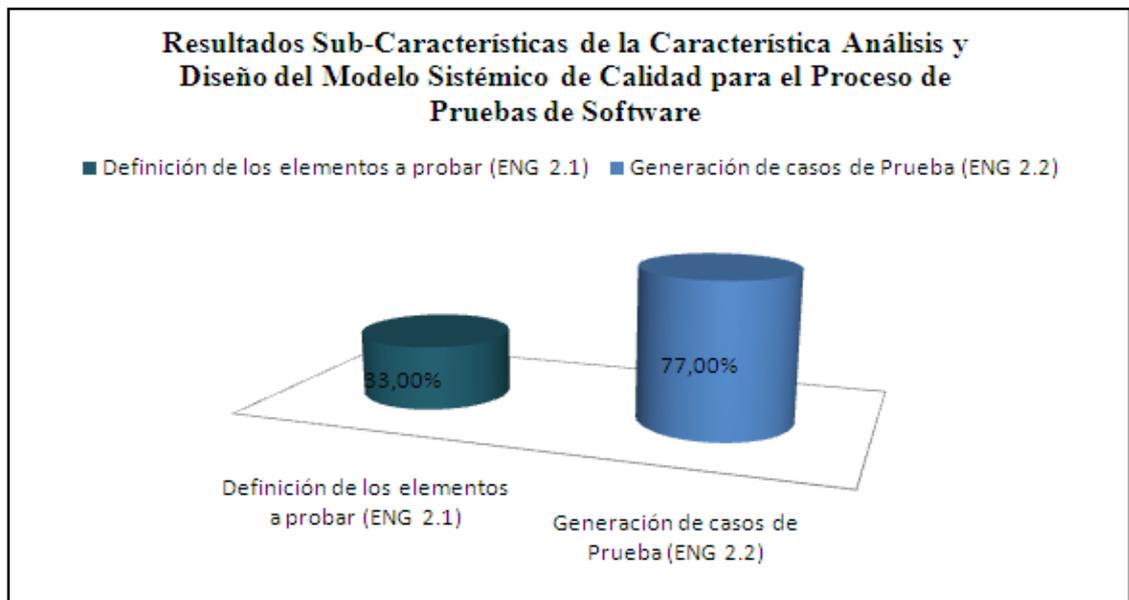
**Figura 41:** Resultados Sub-Características de la Característica Planificación y Control (ENG1) del Modelo Sistemático de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

La Figura 41, muestra los resultados de la evaluación de las sub-características de la característica Planificación y Control (ENG1), en ella se observa que el mayor porcentaje lo obtuvo la sub-característica: Determinar el alcance de las pruebas (ENG 1.1) con un 100% dicho valor es mayor a 75%, por ende está dentro del criterio de aceptación. Sin embargo, las sub-características: Desarrollar el plan de Pruebas (ENG 1.2) y Entorno de Pruebas(ENG 1.3) obtuvieron 66 % y

0% respectivamente, lo cual implica que no se encuentran dentro del nivel de aceptación establecido

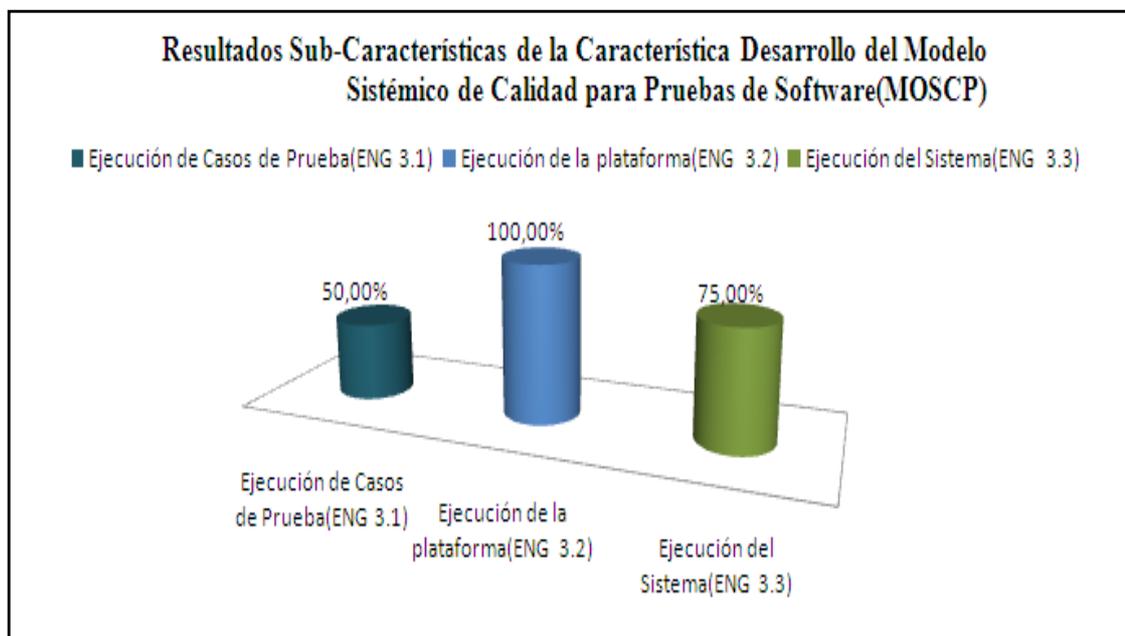


**Figura 42:** Resultados Sub-Características de la Característica Análisis y Diseño (ENG2) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

En la Figura 42, se muestra el gráfico con los resultados obtenidos, para las sub-características asociadas a la característica Análisis y Diseño (ENG2), se puede observar que el mayor porcentaje lo obtuvo la sub-característica: Generación de casos de Prueba (ENG 2.2) con un 77% dicho valor es mayor a 75%, por ende está dentro del criterio de aceptación, le sigue la sub-característica Definición de los elementos a probar (ENG 2.1), con un 33%, esto indica que no se establecen criterios claros en la definición de los elementos a probar durante el proceso de pruebas de software, este resultado está por debajo del 75% por lo tanto no está dentro del criterio de aceptación.

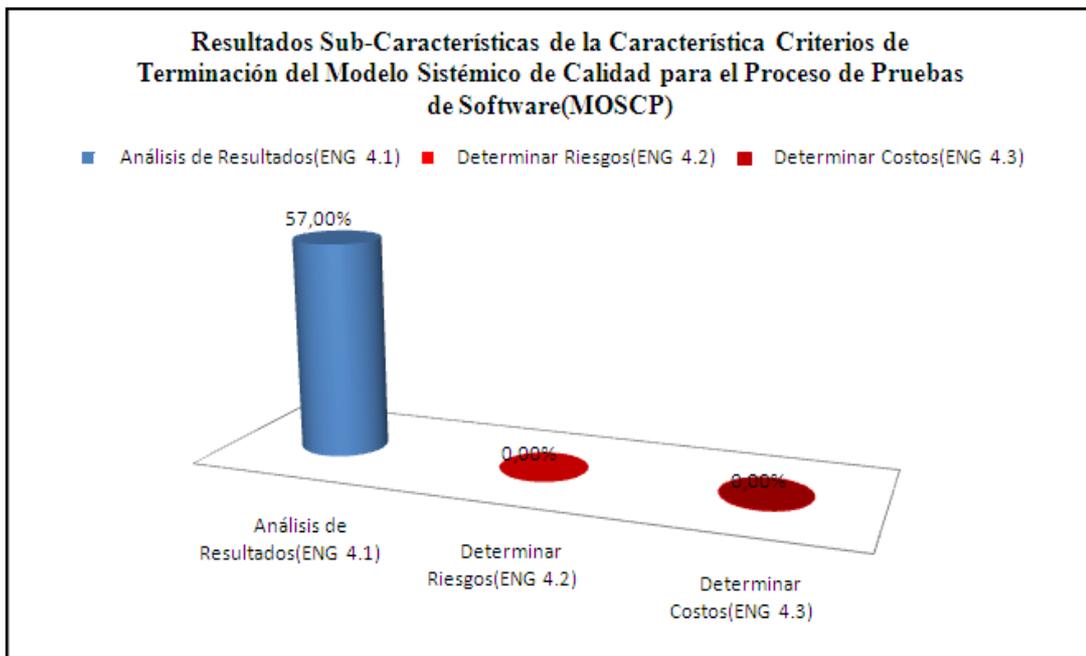


**Figura 43:** Resultados Sub-Características de la Característica Desarrollo (ENG3) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

La Figura 43, muestra los Resultados Sub-Características de la Característica Desarrollo (ENG3), se observa que la sub-característica Ejecución de la Plataforma (ENG 3.2) y Ejecución del Sistema lograron alcanzar un puntaje de 100% y 75% respectivamente, dichos valores se encuentran dentro del porcentaje de aceptación. Sin embargo, la sub-característica Ejecución de Casos de Prueba obtuvo un porcentaje de 50%, lo cual no cumple con el nivel de aceptación.



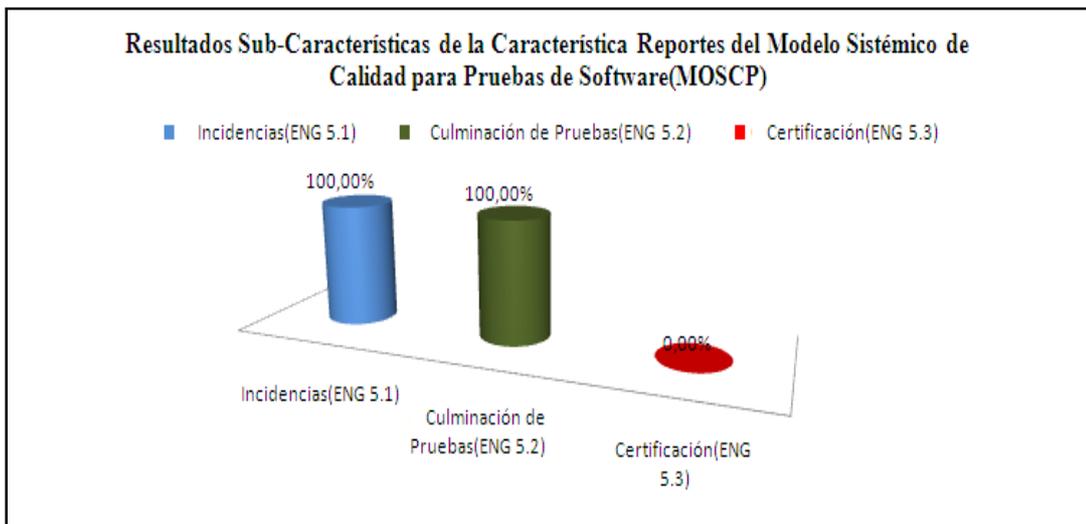
**Figura 44:** Resultados Sub-Características de la Característica Criterios de Terminación (ENG4) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

Los resultados de las Sub-Características de la Característica Criterios de Terminación (ENG4), se pueden visualizar en la Figura 44, en ésta se observa que los resultados de las sub-características están muy por debajo del criterio de aceptación, esto significa que el equipo involucrado en el proceso de pruebas de software no toman en cuenta criterios importantes como los riesgos de fallos potenciales, los costos de continuar con las pruebas de software, así como el análisis de resultados para la culminación de las pruebas.

Seguidamente se presenta la Figura 45, en la cual se pueden observar los resultados de la evaluación, correspondiente a las sub-características de la Característica Reportes, en la cual se visualiza claramente que las sub-características Incidencias (5.1) y Culminación de las Pruebas(5.2), obtuvieron el mayor puntaje 100% quedando de esta manera satisfecho el criterio de aceptación. No obstante, la sub-característica Certificación obtuvo un porcentaje de 0%, lo cual significa que no se generan reportes de certificación de las pruebas de los elementos de software.

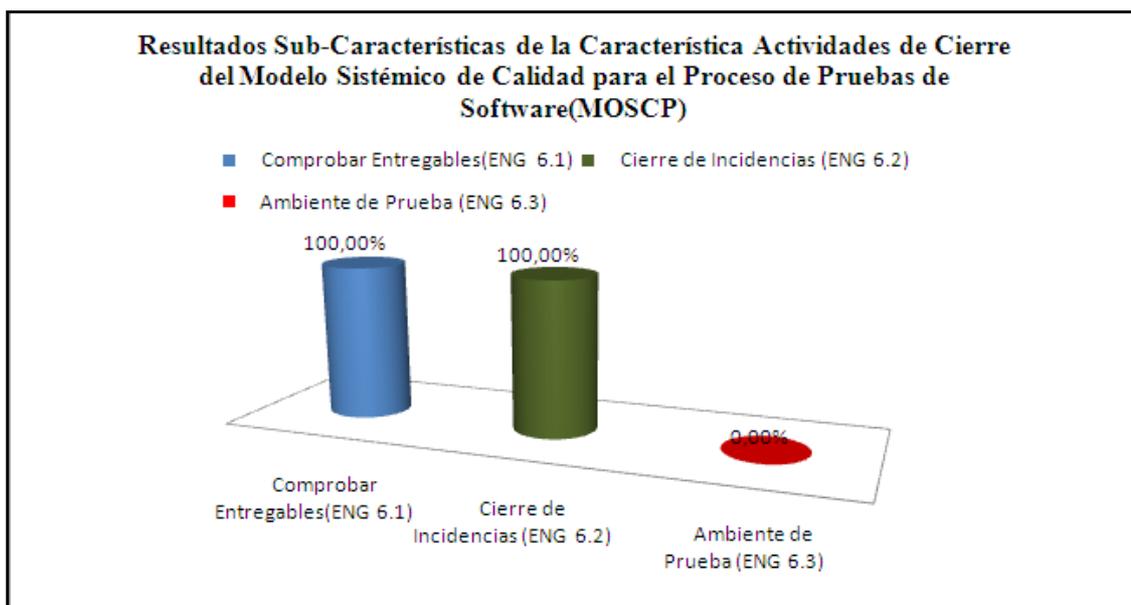


**Figura 45:** Resultados Sub-Características de la Característica Reportes (ENG5) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

De igual manera, se presentan los resultados de la evaluación de las Sub-Características de la Característica Actividades de Cierre (ENG6), dichos resultados se pueden observar en la Figura 46, en la cual se observa claramente que las sub-características; Comprobar Entregables (ENG 6.1) y Cierre de Incidencias (ENG 6.2) alcanzaron el mayor porcentaje(100%), lo cual cumple con el criterio de aceptación; en cambio la sub-característica correspondiente al Ambiente de Pruebas(ENG 6.3), obtuvo un 0%, esto significa que no se guarda el ambiente de pruebas para futuras pruebas de software. Es importante destacar, que debido a este resultado no se logró el nivel de aceptación establecido, a saber 75 % de aceptación.

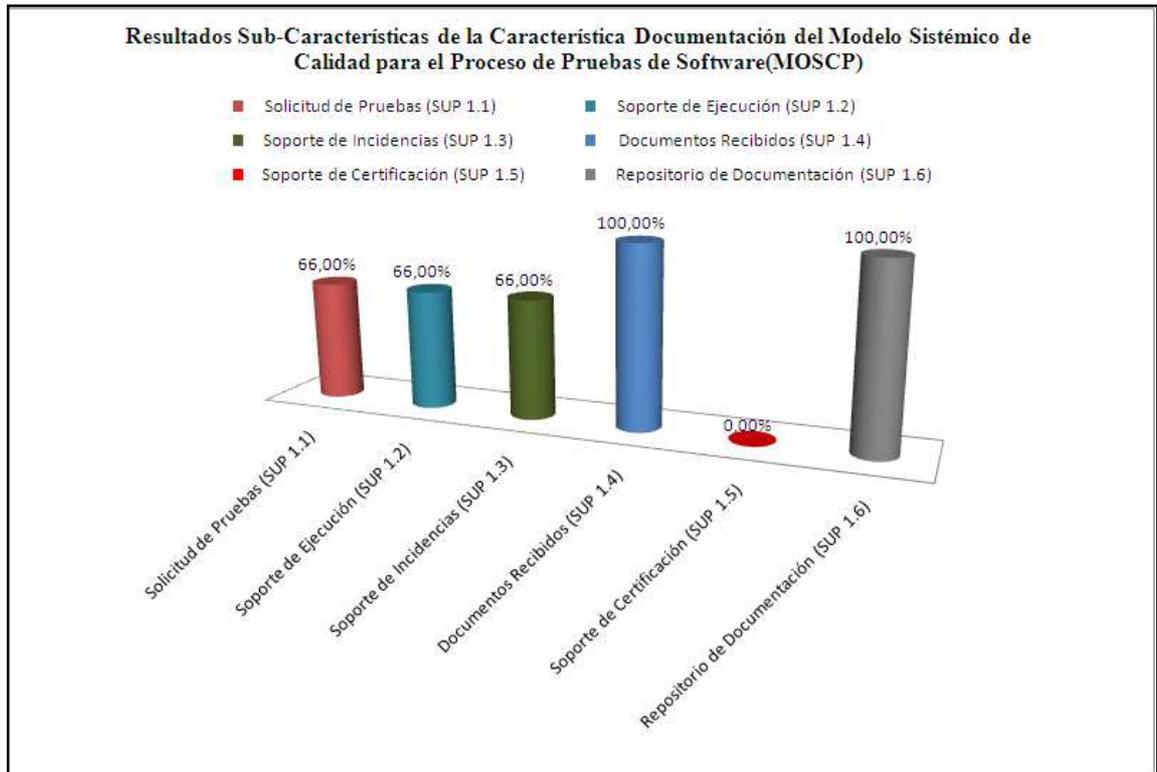


**Figura 46:** Resultados Sub-Características de la Característica Actividades de Cierre (ENG6) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

En igual forma, se muestra la Figura 47, en donde se pueden visualizar los resultados correspondientes Sub-Características de la Característica Documentación (SUP1), donde Documentos Recibidos (SUP 1.4) y Repositorio de Documentación (SUP 1.6) poseen un puntaje de 100% cumpliendo con el nivel de aceptación. Por otro lado, las sub-características: Solicitud de Pruebas (SUP 1.1), Soporte de Ejecución (SUP 1.2), Soporte de Incidencias (SUP 1.3) obtuvieron un porcentaje de 66%; sin embargo no cumplen con el nivel de aceptación. Por último, la sub-característica Soporte de Certificación (SUP 1.5), tiene un puntaje de 0%, esto señala que no se cuenta con un formato de certificación de las pruebas realizadas.

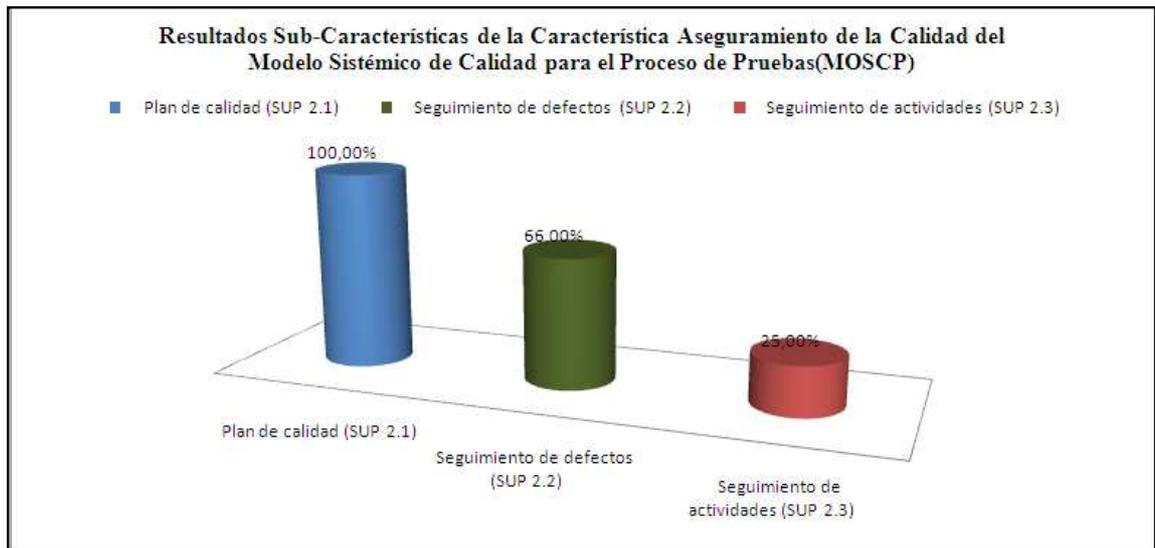


**Figura 47:** Resultados Sub-Características de la Característica Documentación (SUP1) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

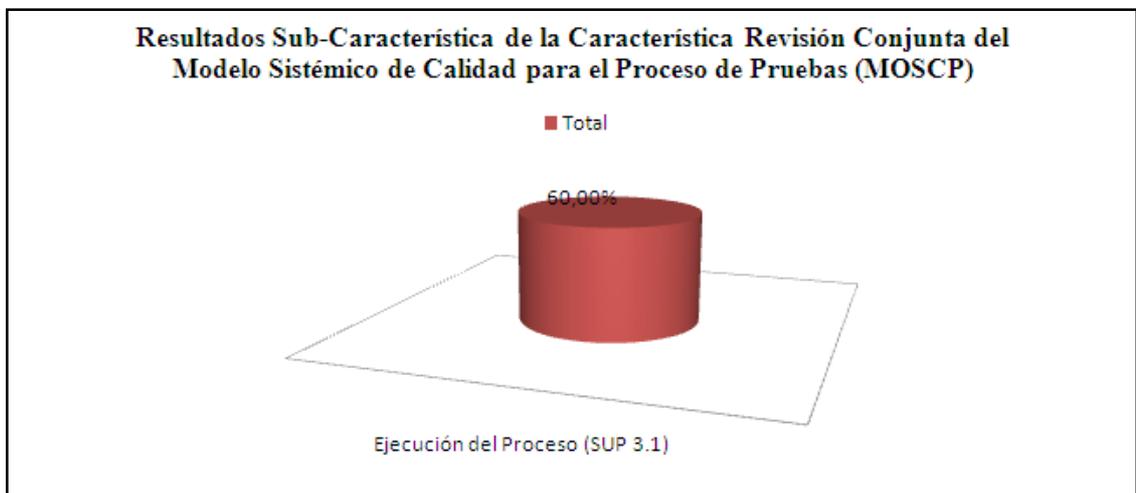
Seguidamente se presenta la Figura 48, en la cual se observan los resultados de la evaluación de las Sub-Características de la Característica Aseguramiento de la Calidad (SUP2). El mayor porcentaje fue alcanzado por Plan de Calidad (SUP 2.1), logrando satisfacer el criterio de aceptación, seguido por la puntuación de la sub-característica: Seguimiento de defectos (SUP 2.2), Seguimiento de Actividades (SUP 2.3) con un 66% y 25% respectivamente, lo cual indica que no se realiza un correcto seguimiento de la ejecución de las actividades desarrolladas durante el proceso de pruebas de software, dichos porcentajes no cumplen con el criterio de aceptación establecido.



**Figura 48:** Resultados Sub-Características de la Característica Aseguramiento de la Calidad(SUP2) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).



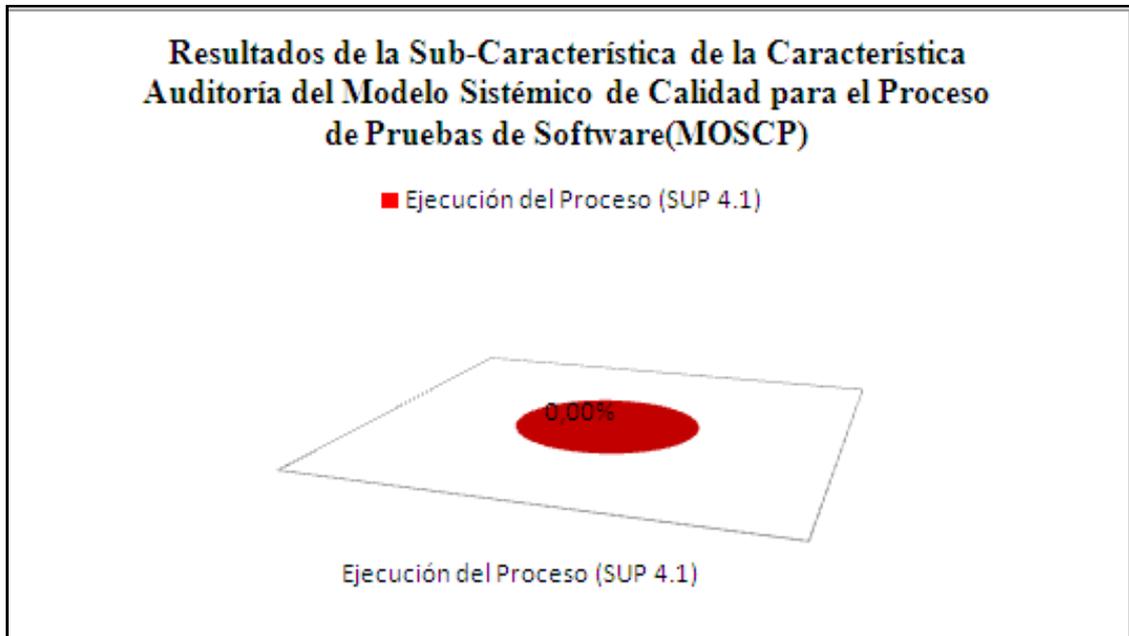
**Figura 49:** Resultados Sub-Característica de la Característica Revisión Conjunta (SUP3) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

La Figura 49, contiene los resultados de la evaluación de las Sub-Characterísticas de la Característica Revisión Conjunta (SUP3), la cual refleja que

no se lleva a cabo de manera controlada el proceso de revisión conjunta debido a que logró una puntuación de 60%, lo cual no está dentro del nivel de aceptación.



**Figura 50:** Resultados Sub-Característica de la Característica Auditoría (SUP4) del Modelo Sistemico de Calidad para el Proceso de Pruebas de Software (MOSCP).

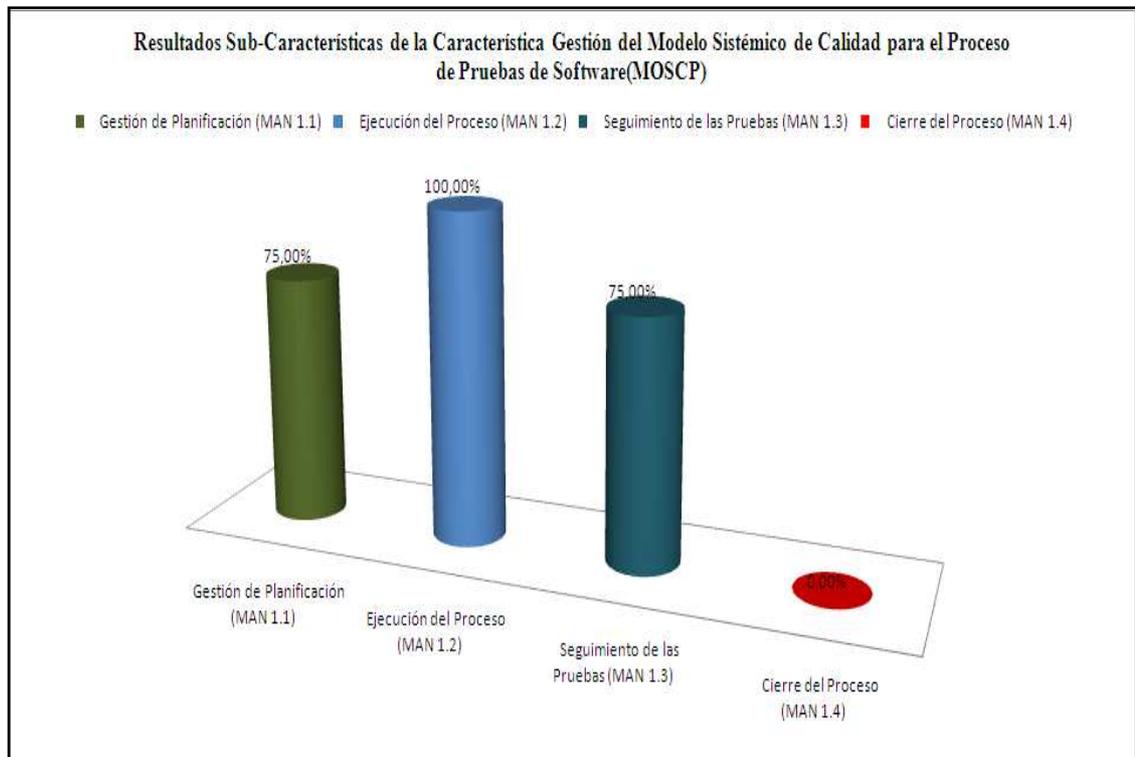
**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

En la Figura 50, se muestran los resultados Sub-Característica de la Característica Auditoría (SUP4), la cual arrojó un puntaje de 0%, esto significa que en el proceso de pruebas no se lleva a cabo el proceso de Auditoria.

Seguidamente se presenta la Figura 51, la cual contiene los resultados de la evaluación de la Sub-Características de la Característica Gestión (MAN1), en ella se observa que la sub-característica que obtuvo el 100% fue Ejecución del Proceso (MAN 1.2), seguida por las características Gestión de Planificación (MAN 1.1) y Seguimiento de las Pruebas (MAN 1.3) ambas con un 75%; éstas sub-características se encuentran dentro del nivel de aceptación. Sin embargo, la sub-característica Cierre del Proceso (MAN 1.4), obtuvo un 0 %, lo cual indica que no se lleva a cabo una correcta verificación en cuanto al cierre del proceso de pruebas

de software, tomando en cuenta aspectos como la correcta documentación y ejecución de los casos de prueba.

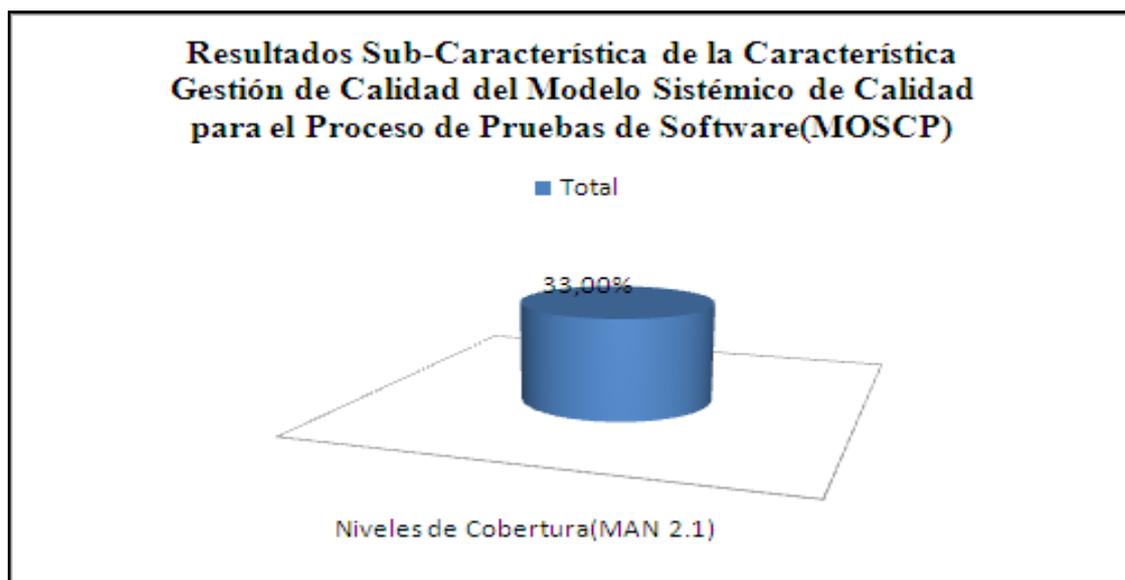


**Figura 51:** Resultados Sub-Características de la Característica Gestión (MAN1) del Modelo Sistemático de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

Para finalizar con la evaluación de las sub-características, se presenta la Figura 52, la cual contiene la información correspondiente a los resultados de la Sub-Característica de la Característica Gestión de Calidad (MAN2), donde se observa que Niveles de Cobertura (MAN 2.1) posee un puntaje de 33%, este resultado refleja que no se verifica los niveles de cobertura del proceso de pruebas de software. Debido a que el resultado es menor a 75%, se puede indicar que no cumple con el criterio de aceptación.



**Figura 52:** Resultados Sub-Característica de la Característica Gestión de Calidad(MAN2) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

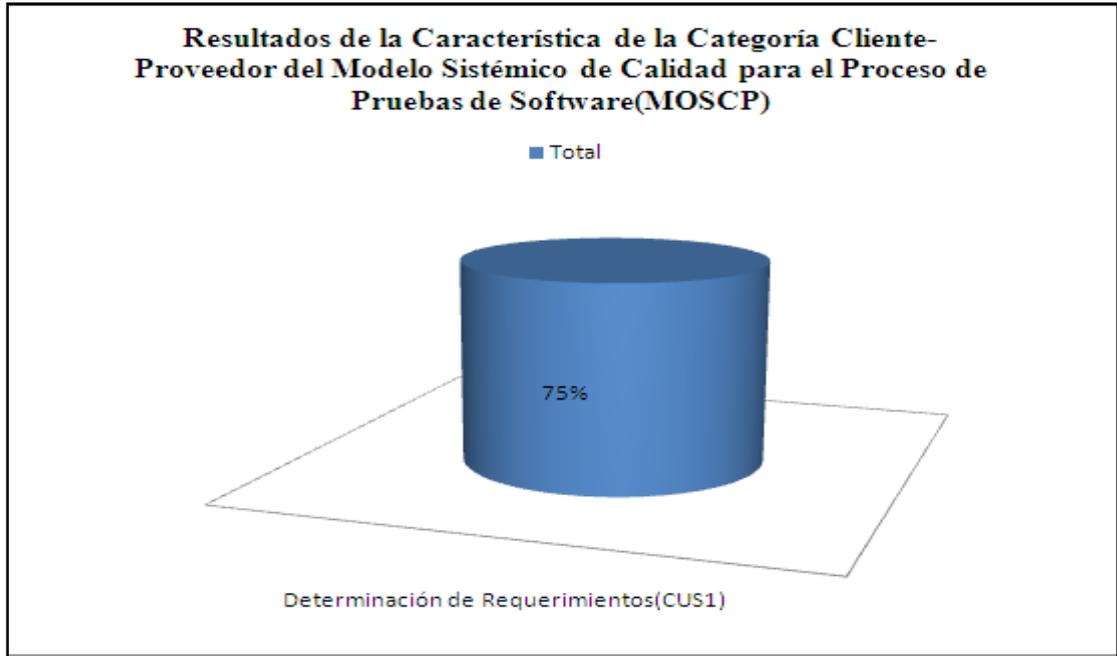
**Autor:** Elaboración Propia (2012).

A continuación, se presentan los resultados correspondientes a evaluación de las Características del modelo.

- **Evaluación de las Características del Modelo en el Proceso de Pruebas de Software del desarrollo del Sistema de Inscripción para el Control de Actividades Académicas (SICAA)**

En las Figuras 53, 54, 55, 56, 57,58 se presentan los resultados de la evaluación por Características

En la Figura 53, se pueden apreciar los resultados de la evaluación de la Característica Determinación de Requerimientos (CUS1), en la cual se observa que obtuvo un puntaje de 75%, dicho resultado se encuentra dentro del criterio de aceptación.

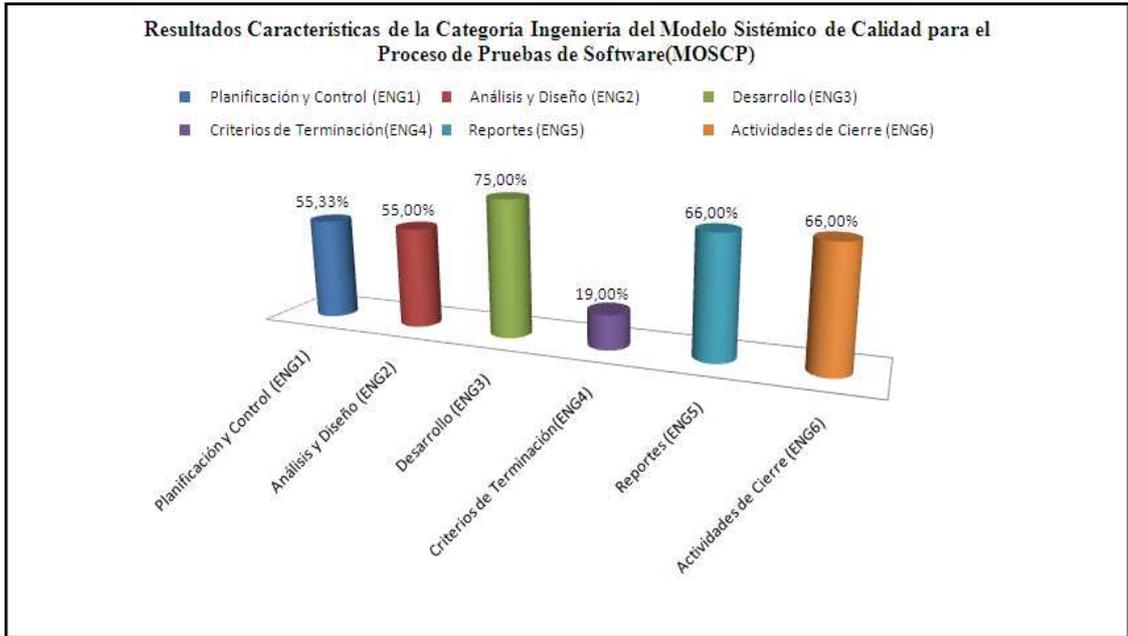


**Figura 53:** Resultados Característica Determinación de Requerimientos (ORG1) del Modelo Sistemico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

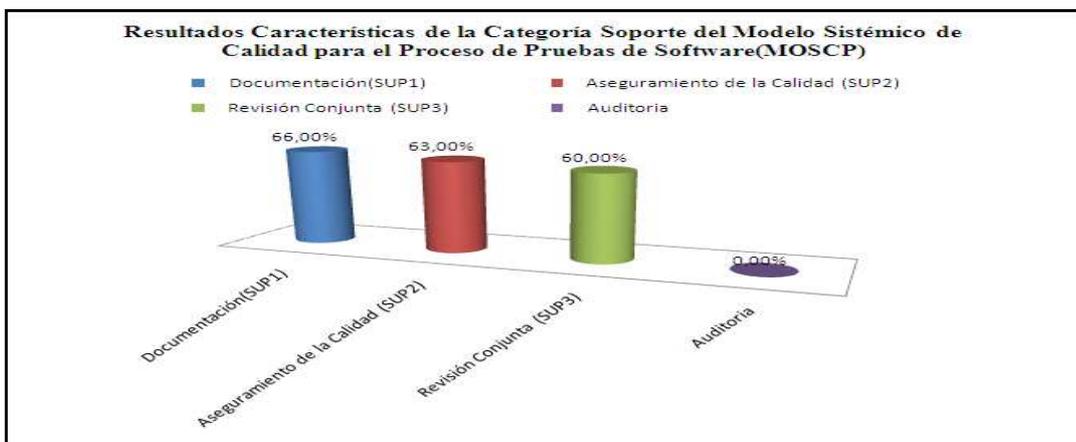
**Autor:** Elaboración Propia (2012).

Seguidamente, se presenta los resultados de la evaluación correspondiente a Características de la Categoría Ingeniería (ENG), en ella se puede observar que la Característica Desarrollo (ENG 3) obtuvo un porcentaje de 75% cumpliendo con el nivel de aceptación. Por el contrario, las características: Planificación y Control (ENG1), Análisis y Diseño (ENG 2), Criterios de Terminación (ENG 4), Reportes (ENG5) Actividades de Cierre (ENG 6) obtuvieron un 55.33%,55%,19%,66%,66% respectivamente, lo cual no es menor a 75%; por lo tanto dichas características no entran en el criterio de aceptación. (Ver Figura 54)



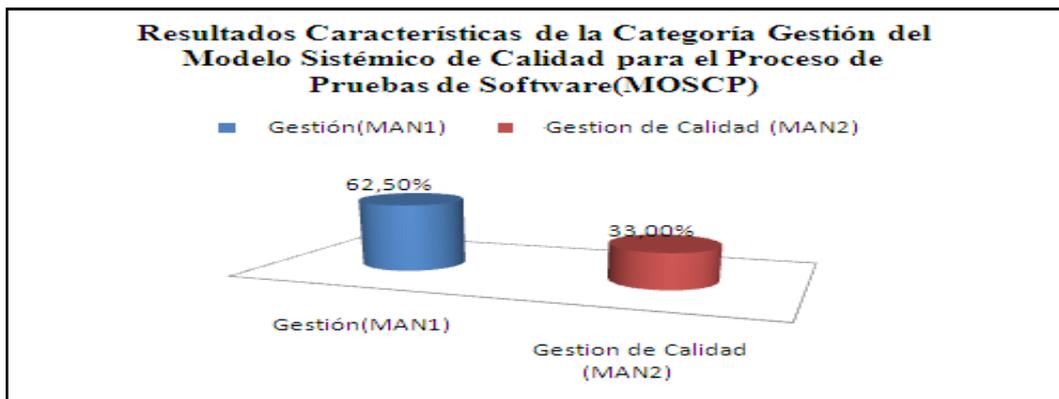
**Figura 54:** Resultados de las Características de la Categoría Ingeniería (ENG) del Modelo Sistemático de Calidad para el Proceso de Pruebas de Software (MOSCP).  
**Fuente:** Autor de la Investigación (2012).  
**Autor:** Elaboración Propia (2012).

Los resultados correspondientes a la evaluación de las Características de la Categoría Soporte (SUP), se presentan en la Figura 55, los porcentajes obtenidos en esta evaluación son los siguientes: Documentación (SUP1) 66%, Aseguramiento de la Calidad (SUP2) 63%, Revisión Conjunta (SUP3) 60%, Auditoría (SUP4) 0%, dado que el nivel de aceptación es de 75%, ninguna de las características están dentro del mismo.



**Figura 55:** Resultados de las Características de la Categoría Soporte (SUP) del Modelo Sistemático de Calidad para el Proceso de Pruebas de Software (MOSCP).  
**Fuente:** Autor de la Investigación (2012).  
**Autor:** Elaboración Propia (2012).

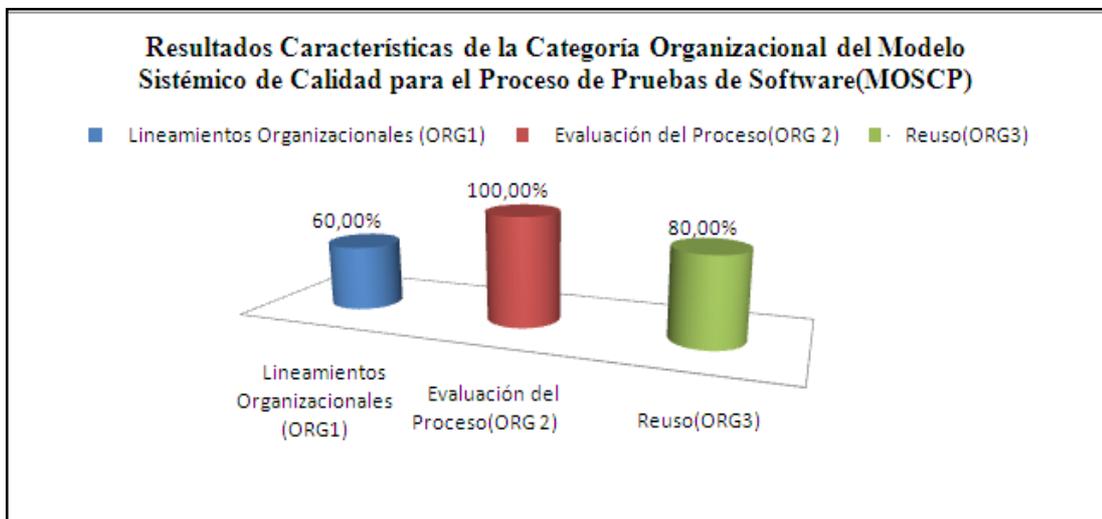
Asimismo se presenta la Figura 56, la cual contiene los resultados de la evaluación de las Características de la Categoría Gestión (MAN), en ella se observa claramente que las características no alcanzaron el nivel de aceptación, ya que la Característica Gestión (MAN1) obtuvo un 62,50% y Gestión de Calidad (MAN 2) un 33%.



**Figura 56:** Resultados de las Características de la Categoría Gestión (MAN) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).



**Figura 57:** Resultados Características de la Categoría Organizacional (ORG) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

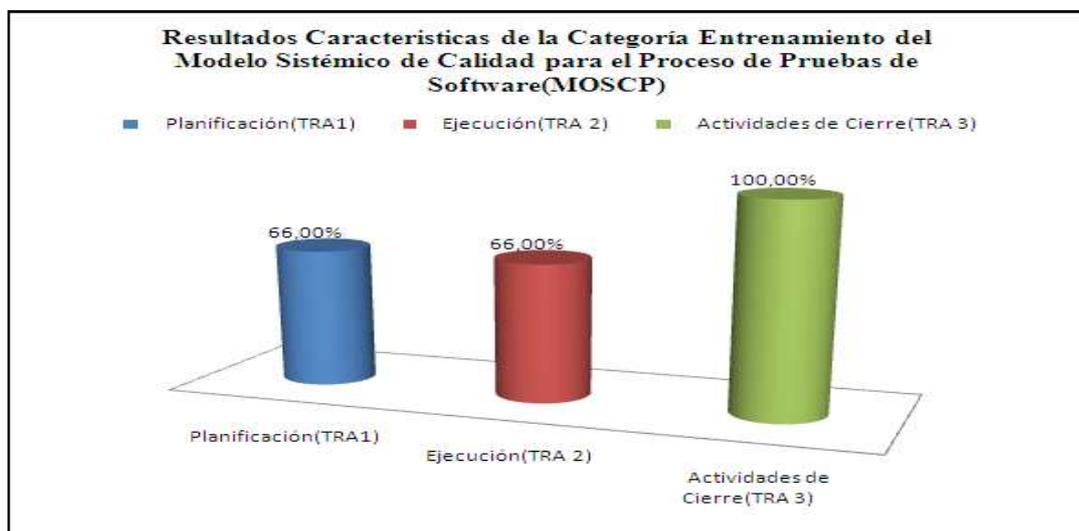
**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

En la Figura 57, se pueden observar los resultados de la evaluación de las Características de la Categoría Organizacional (ORG), en ella se observa que la característica Evaluación del Proceso (ORG2), obtuvo un 100%, luego le sigue

Reuso(ORG3) con un 80%, ambos puntajes están dentro del nivel de aceptación. Sin embargo, la característica Lineamientos Organizacionales (ORG1), obtuvo un 60%, lo cual no cumple con el criterio de aceptación.

Por otro lado, en la Figura 58 se muestran los resultados de las Características correspondientes a la Categoría Entrenamiento (TRA), se puede observar que la característica Actividades de Cierre obtuvo un 100% lo cual satisface el criterio de aceptación, pero Planificación (TRA1) y Ejecución(TRA2) obtuvieron 66% , en este caso no cumplen con el criterio de aceptación.



**Figura 58:** Resultados Características de la Categoría Entrenamiento (TRA) del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

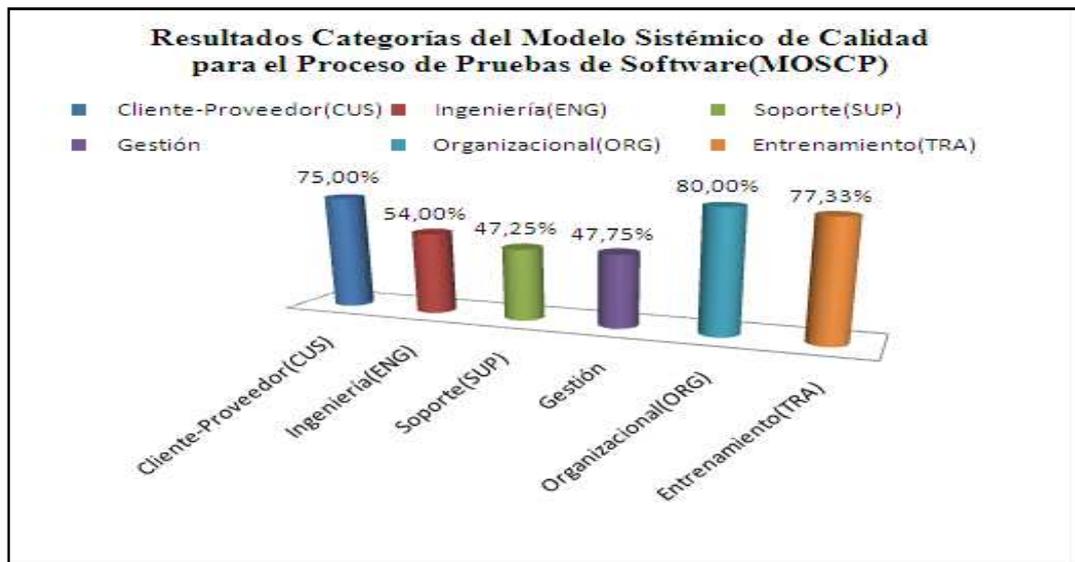
**Autor:** Elaboración Propia (2012).

Finalmente, se presentan los resultados correspondientes a evaluación de las Categorías del modelo.

### **Evaluación de las Categorías del Modelo en el Proceso de Pruebas de Software del desarrollo del Sistema de Inscripción para el Control de Actividades Académicas (SICAA)**

Los resultados de la evaluación se encuentran de manera detallada en el Anexo III. A continuación se describen los elementos resaltantes de la evaluación (Ver Figura 59):

La Categoría Organizacional (ORG), obtuvo el mayor porcentaje con un 80%, luego le sigue Entrenamiento (TRA) 77.33%, Cliente-Proveedor (CUS) con un 75%, éstas categorías obtuvieron un valor mayor o igual a 75% y según lo establecido en el criterio de aceptación presentan un nivel de calidad aceptable. Sin embargo, las Categorías Ingeniería (ENG), Soporte (SUP), Gestión (MAN) obtuvieron 54%, 47.25, 47.75 respectivamente, dichos porcentajes no cumplen con un nivel de calidad aceptable.



**Figura 59:** Resultados Categorías del Modelo Sistemico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

Seguidamente, se presenta un resumen de la evaluación correspondiente a las categorías del Modelo en el Proceso de Pruebas de Software del desarrollo del Sistema de Inscripción para el Control de Actividades Académicas (SICAA), así como el nivel de calidad alcanzado en cada una de las categorías, dichos resultados se pueden apreciar en la Tabla 37

Categoría	Porcentaje	Nivel de Calidad
Cliente-Proveedor (CUS)	75%	Satisfecha
Ingeniería (ENG)	54%	No Satisfecha
Soporte (SUP)	47.25	No Satisfecha
Gestión (MAN)	47.75	No Satisfecha
Organizacional (ORG)	80%	Satisfecha
Entrenamiento (TRA)	77.33	Satisfecha

**Tabla 37:** Resultados evaluación Categorías del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

**Fuente:** Autor de la Investigación (2012).

**Autor:** Elaboración Propia (2012).

Según lo señalado por Mendoza et.al(2005), para obtener una calidad básica del proceso se deben satisfacer las Categorías Cliente-Proveedor(CUS) e Ingeniería(ENG), de acuerdo a los resultados obtenidos(Ver Tabla 38) se puede concluir que el **Proceso de Pruebas de Software del desarrollo del Sistema de Inscripción para el Control de Actividades Académicas (SICAA) posee un nivel de calidad nulo.**

## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

El presente capítulo tiene como finalidad presentar las conclusiones que el autor puede establecer al desarrollar el presente trabajo de investigación, así como las recomendaciones para futuros refinamientos e investigaciones. Las conclusiones que se presentan a continuación se basan en el marco teórico desarrollado, los antecedentes de la investigación, la metodología, los objetivos específicos, así como la propuesta, a continuación se exponen las conclusiones:

- El proyecto constituye el contexto que debe servir de referencia para proponer un modelo de calidad sistémico para el proceso de pruebas de software, basado en el Modelo Sistémico de Calidad (MOSCA), el cual se basa en el concepto de calidad global sistémica, integrando la calidad del proceso y el producto.
- El concepto de proceso de pruebas de software, involucra cada una de las fases que lo integran (planificación, análisis y diseño de las pruebas, implementación y ejecución, evaluar el criterio de terminación y las actividades de cierre), así como los elementos que la conforman.
- El marco metodológico utilizado en la presente investigación, orientó la investigación y a su vez sirvió para evidenciar que el método investigación- acción, ofrece numerosas ventajas en el desarrollo de trabajos de investigación.
- Se analizaron cada uno de los antecedentes de la investigación relacionadas al proceso de pruebas de software, tomándose de las mismas las características más resaltantes que debe poseer el proceso de pruebas,

dicho análisis sirvió de base para la selección y propuestas de las características del modelo propuesto.

- Se elaboró el modelo conceptual del proceso de pruebas de software, en él se puede evidenciar los aportes de los diversos antecedentes analizados, así como la definición del proceso de pruebas y los elementos que la integran. Es importante destacar, que el modelo conceptual presentado en el trabajo de investigación, constituye un gran aporte en el área de Ingeniería del software, puesto que las pruebas se investigaron realizando especial énfasis en la calidad sistémica.
- Se presentó la descripción del Modelo Sistémico de Calidad (MOSCA), el cual brindó la estructura (Dimensiones, categorías, características y métricas) y la visión sistémica al Modelo Sistémico de Calidad para el proceso de pruebas (MOSCP).
- Se propuso un modelo que incluye 6 categorías, 19 características, 35 Sub-características, 125 Métricas, definidas en el contexto del proceso de pruebas de software.
- El resultado de la evaluación de las Categorías, Características, Subcaracterísticas y Métricas, fueron consideradas aceptables por los evaluadores, ya que las consideraron pertinentes, completas, adecuadas y precisas.
- El análisis de los resultados de la evaluación en el proceso de pruebas de software del desarrollo de SICAA, mostró que tiene un nivel de calidad nulo.

Así, tomando en cuenta las conclusiones planteadas, se han considerado las siguientes recomendaciones para futuros trabajos de investigación:

- Extender el estudio del proceso de pruebas de software, como un proceso que apoya la calidad del desarrollo de software.
- Usar el modelo propuesto para ofrecer a las organizaciones un marco de referencia en lo que respecta a la calidad del proceso de pruebas software desde el punto de vista sistémico.

- Plantear un trabajo que implemente a nivel de software la propuesta del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP).

## GLOSARIO DE TÉRMINOS

**Actividad:** Una actividad es un conjunto de acciones que se realizan con el objetivo de crear o actualizar uno o varios artefactos.

**Área de Proceso:** Un grupo de prácticas relacionadas en un área que, cuando se implementan colectivamente, satisfacen un conjunto de metas consideradas importantes para hacer mejoras en esa área. Todas las áreas de proceso CMMI son comunes tanto a la representación continua como a la representación por etapas.

**Artefacto:** Un artefacto es todo tipo de información creada, producida, cambiada o usada por el proceso, es un producto del proceso. Los artefactos pueden tener asociado una plantilla, como guía de su contenido. Los artefactos podrán asociarse a las actividades como entrada o salida de las mismas.

**COTS:** Elementos que pueden comprarse a un vendedor comercial (COTS equivale a “Commercial off the shelf”, es decir producto comercial).

**Disciplina:** Una disciplina permite realizar un agrupamiento lógico de las diferentes actividades.

**Entregables:** Se especifican los artefactos que serán producidos durante el proyecto de prueba y serán entregados al cliente.

**Herramientas:** Las herramientas asisten en la realización de las diversas actividades del proceso.

Las herramientas podrán ser asociadas o desasociadas a una actividad. Esto indica que para realizar la actividad se usa la herramienta.

**Holismo:** Considera que los sistemas constituyen totalidades compuestas de partes.

**Nivel de Madurez:** Grado de mejora de procesos a través de un conjunto predefinido de áreas de proceso en las que se alcanzan todas las metas del conjunto.

**Parte Interesada Relevante:** Una parte interesada que se identifica por su involucración en actividades específicas y que se incluye en un plan.

**Práctica Específica:** Un componente esperado del modelo CMMI que se considera importante para alcanzar la meta específica asociada. Las prácticas específicas describen las actividades esperadas para dar como resultado el logro de las metas específicas de un área de proceso.

**Productos de trabajo típicos:** Un componente informativo del modelo CMMI que proporciona ejemplos de resultados de una práctica específica. Estos ejemplos se denominan productos de trabajo típicos porque a menudo hay otros productos de trabajo que son igual de eficaces pero no están enumerados.

**Representación Continua:** Una estructura del modelo de capacidad y de madurez (CMMI), donde los niveles de capacidad proporcionan un orden recomendado para abordar la mejora de procesos dentro de cada área de proceso específica

**Representación por Etapas:** Una estructura del modelo CMMI en la que el alcance de las metas de un conjunto de áreas de proceso establece un nivel de madurez; cada nivel construye una base para los niveles siguientes.

**Rol:** Un rol define el comportamiento y responsabilidades de un individuo, o un conjunto de individuos que trabajan juntos como un equipo.

**Sinergia:** Sinergia viene del griego syn, con y ergos, trabajo, es decir, trabajo conjunto.

**SOA:** Es una arquitectura de software que comienza con una definición de interfaz y construye toda la topología de la aplicación como una topología de interfaces, implementaciones y llamados a interfaces. SOA es una relación de servicios y consumidores de servicios, ambos suficientemente amplios para representar una función de negocios completa.

**Subpráctica:** Un componente informativo del modelo CMMI que proporciona guías para interpretar e implementar una práctica específica o genérica. Las subprácticas pueden redactarse como si fueran obligatorias, pero de hecho sólo pretenden proporcionar ideas que pueden ser útiles para la mejora de procesos.

**Testware:** Es el producto resultante de las actividades de prueba.

**TMap:** Es una metodología desarrollada por Sogeti de aproximación al testing, y abierta al resto del mercado. Esta metodología permite una mayor rapidez en la puesta en explotación de un software de excelente calidad.

## REFERENCIAS BIBLIOGRÁFICAS

Baskerville, Richard (1999). A critical perspective on action research as a method for information systems research Volume 2 Issue 3es, Nov. 1999 Association for Information SystemsAtlanta. GA, USA.

Calvo-Manzano, J.A; Cuevas, G; Jimenez-Puello, J; San Feliu, T (2009). Comparativa entre modelos y estándares de madurez para pruebas de software enfocados en la práctica de validación del CMMI-DEV v1.2 como referencia. Actas de los Talleres de las Jornadas de Ingeniería de Software y base de datos, Vol.3, No 4.Madrid, España.

Casallas, Rubby; Correal, Darío; López, Nicolás. (2007). Mejoramiento del Proceso de Pruebas en un Contexto de Desarrollo de Software Globalizado. Universidad de Los Andes. Bogotá, Colombia.

Chávez, Indira;López,Yucely;Delgado Martha.(2009). Propuesta de generación de casos de prueba teniendo en cuenta indicadores de cubrimiento. 7th Latin American and Caribbean Conference for Engineering and Technology. San Cristóbal, Venezuela.

Chrissis, Mary Beth; Konrad,Mike; Shrum, Sandy (2009) .CMMI, Guía para la integración de procesos y la mejora de productos. Pearson Educación. Segunda edición. Madrid.

CMMI. Guía para la integración de procesos. (2009) [Documento en Línea]  
Disponible en:

<http://www.sei.cmu.edu/library/assets/cmmi-dev-v12-spanish.pdf>

Constitución de la República Bolivariana de Venezuela (1999). (p.81)

De Rojas, Antonio; Vos Tanja; Marín Beatriz. (2009). Experiencias de una PYME en la mejora del proceso de pruebas. Revista Española de Innovación, Calidad e Ingeniería del Software, Vol.5, No.2. Madrid, España.

Fernández, Luis (2005). Un sondeo sobre la práctica actual de pruebas de software en España. Revista Española de Innovación, Calidad e Ingeniería del Software, Vol.1, No.2. Madrid, España.

Fernández, Luis; Lara, Pedro; Villalba, María; Vos, Tanja (2008). Factores que afectan negativamente a la aplicación práctica de las pruebas de software. Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos, Vol. 2, No. 4.

Fuggetta, A. (2000): Software Process: A Roadmap. International Conference on Software Engineering.

García, María; Irrazabal Emanuel; Garzás, Javier (2010). Implantación de las normas ISO/IEC 15504 e ISO/IEC 12207 con métodos ágiles y SCRUM. [Documento en Línea] Disponible en: [www.kybeleconsulting.com](http://www.kybeleconsulting.com).

Gutiérrez, Javier; Escalona, María; Mejías, Manuel; Torres, Jesús. (2005). Estudio comparativo de propuestas para la generación de casos de prueba a partir de requisitos funcionales. Universidad de Sevilla, España.

Gutiérrez, Javier; Escalona, María; Mejías, Manuel; Reina, Antonia. (2006)<sup>a</sup>. Modelos de Pruebas Para Pruebas del Sistema. Universidad de Sevilla, España.

Gutiérrez, Javier; Escalona, María; Mejías Manuel; Reina, Antonia. (2006)<sup>b</sup>. Modelos y Algoritmos para la generación de objetivos de Pruebas. Universidad de Sevilla, España.

Grimán, Anna C; Pérez María, Mendoza, Luis E. (2003). Estrategia de Pruebas para Software OO que garantiza requerimientos no funcionales. Laboratorio de Investigación de Sistemas de Información (LISI). Caracas, Venezuela.

Hernández, R; Fernández, C; Baptista, L. (2000). Metodología de la Investigación. Mc Graw Hill Interamericana. México.

Instituto Nacional de Tecnologías de la Comunicación (2009). Guía para la certificación. [Documento en Línea] Disponible en: [www.inteco.es/file/nnpNKZ2X\\_tV506ZWcHTKRg](http://www.inteco.es/file/nnpNKZ2X_tV506ZWcHTKRg)

Kitchenham, B (1996).DESMET: A method for evaluating Software Engineering methods and tools. Department of Computer Science University of Keele.

Kitchenham, B.,Linkman, S; Law, D (1997). “DESMET: A methodology for evaluating software engineering methods and tools,” *IEEE Computing & Control Engineering Journal*, Vol. 8, No. 3.

ISO/TEC 9126 1.2. Information Technology–Software Product Quality, Part 1, Quality Model, ISO/IEC JTC1/SC7/WG6.

León, Tulio (2006). Diseño de un modelo de software educativo para la enseñanza de estructuras de datos en programación. Universidad Centroccidental Lisandro Alvarado. Barquisimeto, Venezuela.

Ley Orgánica de Ciencia Tecnología e Innovación (2005). (p.1)

López, Memphis; Romero, Manuel. (2006). Nacimiento y Desarrollo del Enfoque Sistémico como uno de los métodos generales de investigación. Revista Clepsidra. Universidad Autónoma de Colombia.

International Software Testing Qualifications Board, Certified Tester Foundation Level Syllabus, Versión 2005. [Documento en Línea] Disponible en: <http://www.istqb.org/fileadmin/media/SyllabusFoundation.pdf>

ISO/IEC 29119 Software Testing: Part 2 - Test Process. Working Group 26. Reference: WG26-NO43 (2008). [Página Web en Línea] Disponible en: <http://www.softwaretestingstandard.org/> (Consulta: 7/11/2010).

Méndez, Edumilis; Pérez, María; Mendoza, Luis E (2007)<sup>a</sup>. ¿Cómo se relacionan la Calidad Sistémica y la Productividad en el Proceso de Desarrollo de Software? Actas de Talleres de Ingeniería del Software y Bases de Datos, Vol. 1, No. 4.

Méndez, Edumilis; Pérez, María; Mendoza, Luis E (2007)<sup>b</sup>. Aplicación de un Método para Especificar Casos de Prueba de Software en la Administración Pública. Actas de Talleres de Ingeniería del Software y Bases de Datos, Vol. 1, No. 4.

Mendoza, Luis E; Pérez, María; Grimán, Anna C (2005). Prototipo de Modelo Sistémico de Calidad (MOSCA) del Software. Computación y Sistemas, Vol.8, No.3. México.

Morcillo, Eduardo (2010). Frameworks de desarrollo: un método ágil para el desarrollo de software. Taller digital de la Universidad de Alicante.

Ortega, M., Pérez, M; Rojas, T. (2000) "A Model for Software Product Quality with a Systemic Focus," in 4th World Multiconference on Systemics, Cybernetics and Informatics SCI 2000 and The 6th International Conference on Information Systems.

Pérez, Beatriz. Proceso de Testing Funcional Independiente (2006). PEDECIBA Informática. Montevideo, Uruguay. 148 p.

Pérez, M., Rojas, T, Mendoza, L; Grimán, A. (2001). "Systemic Quality Model for System Development Process: Case Study".Universidad Simón Bolívar.

Pérez, M., Rojas, T, Mendoza, L; Grimán, A. (2004). Prototipo de Modelo Sistémico de Calidad (MOSCA) del Software.Universidad Simón Bolívar.

Pérez, María; Méndez, Edumilis; Domínguez Kenyer; Mendoza, Luis (2008). Propuesta de modelo para especificar la calidad de herramientas de pruebas. Actas de Talleres de Ingeniería del Software y Bases de Datos, Vol. 2, No. 4.

Polo,Macario(2006). Pruebas del Software. Curso de Doctorado sobre proceso software y gestión del conocimiento. Universidad Ciudad Real.

Pressman, Roger S (2005). Ingeniería del Software un enfoque práctico.Mc Graw Hill.Sexta Edición. España.

Prins, Casper; Blokpoel, Martin (2007). Los costos de subestimar los defectos del software. [Documento en Línea] Disponible en: [http://www.es.sogeti.com/Global/ART%<sup>c3</sup>%8dCULOS/ART\\_MARTIN\\_CASPER.pdf](http://www.es.sogeti.com/Global/ART%c3%8dCULOS/ART_MARTIN_CASPER.pdf)

Raja, Elena (2007). Casi todas las pruebas del software. Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos, Vol. 1, No. 4 Madrid,España.

Rivero, Milagro; Montilva, Jonás; Barrios, Judith; Murúa, Mario; Granados, Gladys (2009).Un análisis del desarrollo de software en empresas venezolanas. 7th Latin American and Caribbean Conference for Engineering and Technology. San Cristóbal, Venezuela.

Rivero, Ramón; Ochoa, Elizabeth; Abon, Luis; Abon, Leevan (2010). Propuesta de un modelo de pruebas para una arquitectura orientada a servicios. 8th Latin American and Caribbean Conference for Engineering and Technology. Arequipa, Perú.

Salazar y Trejo (2004). Instrumentos de Recolección de Información Prentice Hall Hispanoamericana S.A. Madrid.

Sanz, A., Saldaña, J., García, J., Gaitero, D. (2008). TestPAI Un área de proceso de pruebas integrada con CMMI. Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos, Vol. 2, No. 4 Madrid, España.

Sanz, A., García, J., Saldaña, J., Amescua, Antonio., Velasco, Manuel (2009). Definición de un Modelo Organizativo para una Factoría de Pruebas de Software. Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos, Vol. 3, No. 4 Madrid, España.

Sogeti y otros. (2011). World Quality Report 2010-2011.

Sommerville, Ian (2005). Ingeniería del Software. Pearson Educación. Séptima edición. Madrid.

Swinkels, Ron (2000). A comparison of TMM and other Test Process Improvement Models. Frits Philips Institute.

Tamayo T, M. (2003). El Proceso de la Investigación Científica. Limusa. México.

Test Maturity Model Integration (TMMi): Version 2.0. TMMi Foundation (2009). [Documento en Línea] Disponible en: <http://www.tmmifoundation.org/downloads/tmmi/TMMi%20Framework.pdf>

Tuya, Javier (2009)<sup>a</sup>. El futuro estándar ISO/IEC 29119-Software Testing. Revista Española de Innovación, Calidad e Ingeniería del Software, Vol.5, No.4. Madrid, España.

Tuya, Javier (2009)<sup>b</sup>. Estado Actual del estándar ISO/IEC 29119 -Software Testing. IV Taller sobre Pruebas en Ingeniería del Software. San Sebastian.17 p.

Universidad Centroccidental Lisandro Alvarado (UCLA) (2002). Manual para la Presentación de Trabajo de Grado. Barquisimeto. Material Mimeografiado.

## **ANEXOS**

**Anexo I.** Resultados de la Evaluación de los **criterios generales** del Modelo.

Categoría a Evaluar(Criterio General)	Pertinencia	Complejidad	Precisión	Adecuación
Cliente-Proveedor(CUS)	1	1	1	1
Ingeniería (ENG)	1	1	1	1
Soporte(SUP)	1	1	1	1
Gestión(MAN)	1	1	1	1
Organizacional(ORG)	1	1	1	1
Entrenamiento(TRA)	1	1	1	1

Como se observa, la primera evaluación corresponde a las Categorías del modelo propuesto, a saber: Cliente-Proveedor (CUS), Ingeniería (ENG), Soporte (SUP), Gestión (MAN), Organizacional (ORG), Entrenamiento (TRA).

Por otro lado, la segunda evaluación está relacionada con las características específicas de las categorías: Cliente-Proveedor (CUS), Ingeniería (ENG), Soporte (SUP), Gestión (MAN), Organizacional (ORG), Entrenamiento (TRA)

Categoría a Evaluar(Criterio General)	Característica a Evaluar	Pertinencia	Complejidad	Precisión	Adecuación
Cliente-Proveedor(CUS)	Determinación de Requerimientos (CUS1)	1	1	1	1
Ingeniería (ENG)	Planificación y Control (ENG1)	1	1	1	1
	Análisis y Diseño (ENG2)	1	1	1	1
	Desarrollo (ENG3)	1	1	1	1
	Criterios de Terminación (ENG4)	1	1	1	1
	Reportes (ENG5)	1	1	1	1
	Actividades de Cierre (ENG6)	1	1	1	1
Soporte(SUP)	Documentación (SUP1)	1	1	1	1
	Aseguramiento de la Calidad (SUP2)	1	1	1	1
	Revisión Conjunta (SUP3)	1	1	1	1
	Auditoría (SUP4)	1	1	1	1
Gestión (MAN)	Gestión (MAN1)	1	1	1	1
	Gestión de Calidad (MAN2)	1	1	1	1
Organizacional(ORG)	Lineamientos Organizacionales(ORG1)	1	1	1	1
	Evaluación del Proceso (ORG2)	1	1	1	1
	Reuso(ORG3)	1	1	1	1
Adiestramiento (TRA)	Planificación (TRA1)	1	1	1	1
	Ejecución (TRA2)	1	1	1	1

	Actividades de Cierre (TRA3)	1	1	1	1
--	------------------------------	---	---	---	---

Por otro lado, la tercera evaluación está relacionada con las subcaracterísticas específicas de las categorías: Cliente-Proveedor (CUS), Ingeniería (ENG), Soporte (SUP), Gestión (MAN), Organizacional (ORG), Entrenamiento (TRA).

Características	Subcaracterísticas	Pertinencia	Compleitud	Precisión	Adecuación
Determinación de Requerimientos (CUS1)	Documento Funcional(CUS 1.1)	1	1	1	1
	Definición del Alcance (CUS 1.2)	1	1	1	1
Planificación y Control (ENG1)	Determinar el alcance de las pruebas (ENG 1.1)	1	1	1	1
	Desarrollar el plan de Pruebas (ENG 1.2)	1	1	1	1
	Entorno de Pruebas (ENG 1.3)	1	1	1	1
Análisis y Diseño (ENG2)	Definición de los elementos a probar (ENG 2.1)	1	1	1	1
	Generación de casos de Prueba (ENG 2.2)	1	1	1	1
Desarrollo (ENG3)	Ejecución de Casos de Prueba (ENG 3.1)	1	1	1	1

	Ejecución de la plataforma (ENG 3.2)	1	1	1	1
	Ejecución del Sistema (ENG 3.3)	1	1	1	1
Criterios de Terminación (ENG4)	Análisis de Resultados (ENG 4.1)	1	1	1	1
	Determinar Riesgos (ENG 4.2)	1	1	1	1
	Determinar Costos (ENG 4.3)	1	1	1	1
Reportes (ENG5)	Incidencias (ENG 5.1)	1	1	1	1
	Culminación de Pruebas (ENG 5.2)	1	1	1	1
	Certificación (ENG 5.3)	1	1	1	1
Actividades de Cierre (ENG6)	Comprobar Entregables (ENG 6.1)	1	1	1	1
	Cierre de Incidencias (ENG 6.2)	1	1	1	1
	Ambiente de Prueba (ENG 6.3)	1	1	1	1
Documentación(SUP1)	Solicitud de Pruebas (SUP 1.1)	1	1	1	1
	Soporte de Ejecución (SUP 1.2)	1	1	1	1
	Soporte de Incidencias (SUP 1.3)	1	1	1	1
	Documentos Recibidos (SUP 1.4)	1	1	1	1
	Soporte de Certificación (SUP 1.5)	1	1	1	1

	Repositorio de Documentación (SUP 1.6)	1	1	1	1
Aseguramiento de la Calidad (SUP2)	Plan de calidad (SUP 2.1)	1	1	1	1
	Seguimiento de defectos (SUP 2.2)	1	1	1	1
	Seguimiento de actividades (SUP 2.3)	1	1	1	1
Revisión Conjunta (SUP3)	Ejecución del Proceso (SUP 3.1)	1	1	1	1
Auditoría (SUP4)	Ejecución del Proceso (SUP 4.1)	1	1	1	1
Gestión (MAN1)	Gestión de Planificación (MAN 1.1)	1	1	1	1
	Ejecución del Proceso (MAN 1.2)	1	1	1	1
	Seguimiento de las Pruebas (MAN 1.3)	1	1	1	1
	Cierre del Proceso (MAN 1.4)	1	1	1	1
Gestión de Calidad (MAN2)	Niveles de Cobertura(MAN 2.1)	1	1	1	1

**Anexo II.** Resultados de la Evaluación de las Métricas del Modelo Sistémico de Calidad para el Proceso de Pruebas de Software (MOSCP) y del modelo en el Proceso de Pruebas de Software del desarrollo del Sistema de Inscripción para el Control de Actividades Académicas (SICAA) del Instituto Universitario de Tecnología del Estado Yaracuy (IUTY).

El cuestionario está dividido en dos secciones. La sección en azul corresponde a la evaluación de las métricas del modelo y la sección en gris es la evaluación del modelo en el proceso de pruebas del IUTY. Las Columnas: Pertinencia, Factibilidad, Profundidad, Escala, se deben completar con los valores 0 o 1. Por otro lado, la columna Respuesta se debe responder con los valores indicados en la columna Formulación.

Características	Subcaracterísticas	Métrica	Pregunta	Formulación	Dirigido a	Pertinencia	Factibilidad	Profundidad	Escala	Respuesta
Determinación de Requerimientos (CUS1)	Documento Funcional (CUS 1.1)	Documento Funcional	¿Recibe el documento funcional de requisitos de software?	0= no 1=si	Analista de Pruebas	1	1	1	1	1
		Revisión del Documento Funcional	¿Realiza la revisión del documento funcional antes de elaborar los casos de	0= no 1=si	Analista de Pruebas	1	1	1	1	1

			prueba?							
	Definición del Alcance(CUS 1.2)	Validación de requisitos	¿Se valida que los requisitos están correctamente definidos?	0= no 1=si	Analista de Pruebas, Usuario	1	1	1	1	0
		Comunicación	¿Existe comunicación, entre las partes involucradas?	0= no 1=si	Analista de Pruebas, Usuario	1	1	1	1	1
Planificación y Control (ENG1)	Determinar el alcance de las pruebas (ENG 1.1)	Alcance de las Pruebas	¿Se establece el alcance de las pruebas antes de la elaboración del plan de pruebas?	0= no 1=si	Líder de Pruebas.	1	1	1	1	1
	Desarrollar el plan de Pruebas (ENG 1.2)	Definición de Actividades	¿Se definen las actividades a ser realizadas?	0= no 1=si	Líder de Pruebas	1	1	1	1	0
		Determinar Roles	¿Se determina el personal que estará involucrado en	0= no 1=si	Líder de Pruebas	1	1	1	1	1

			las pruebas?							
		Tiempo Requerido	¿Se establece el tiempo requerido para la ejecución de las pruebas?	0= no 1=si	Líder de Pruebas	1	1	1	1	1
		Riesgos	¿Se estudian los posibles riesgos en la ejecución de las pruebas?	0= no 1=si	Líder de Pruebas	1	1	1	1	0
		Niveles de Cobertura	¿Se establecen los niveles de cobertura de las pruebas de software?	0= no 1=si	Líder de Pruebas	1	1	1	1	0
		Plan de Pruebas	¿Se genera el plan de pruebas?	0= no 1=si	Líder de Pruebas	1	1	1	1	1
		Documentación	¿Se realiza la documentación del plan de	0= no 1=si	Líder de Pruebas	1	1	1	1	1

			pruebas?							
	Entorno de Pruebas (ENG 1.3)	Disponibilidad del Entorno de Pruebas	¿Se verifica la disponibilidad del entorno para la ejecución de las pruebas?	0= no 1=si	Líder de Pruebas	1	1	1	1	0
Análisis y Diseño (ENG2)	Definición de elementos a probar (ENG 2.1)	Selección de elementos a probar	¿Se seleccionan los elementos a probar?	0= no 1=si	Analista de Pruebas	1	1	1	1	1
		Riesgos de fallo	¿Se toma en cuenta los riesgos de fallo en la selección de los elementos a probar?	0= no 1=si	Analista de Pruebas	1	1	1	1	0
		Aporte del Usuario	¿Se toma en cuenta el aporte del usuario en la selección de	0= no 1=si	Analista de Pruebas	1	1	1	1	0

			los elementos a probar?							
Generación de casos de Prueba (ENG 2.2)	Nivel y técnicas de pruebas	¿Se establece el nivel y técnicas de pruebas a utilizar?	0= no 1=si	Analista de Pruebas	1	1	1	1	0	
	Número de casos de prueba	¿Se especifica un número de caso de prueba?	0= no 1=si	Analista de Pruebas	1	1	1	1	1	
	Entradas del caso de pruebas	¿Se indican las entradas del caso de prueba?	0= no 1=si	Analista de Pruebas	1	1	1	1	1	
	Descripción de pasos de ejecución	¿Se describe detalladamente los pasos a ejecutar en el caso de prueba?	0= no 1=si	Analista de Pruebas	1	1	1	1	1	
	Descripción de resultados esperados	¿Se describen los resultados esperados para	0= no 1=si	Analista de Pruebas	1	1	1	1	1	

			cada prueba?							
		Aplicación a probar	¿Se indica que aplicación o componente se está probando?	0= no 1=si	Analista de Pruebas	1	1	1	1	1
		Relación con otros componentes	¿Se indica la relación del caso de prueba con otra aplicación o componente?	0= no 1=si	Analista de Pruebas	1	1	1	1	0
		Apoyo de Software de Gestión	¿Se encuentran bajo el control de un software de gestión?	0= no 1=si	Analista de Pruebas	1	1		1	1
		Documentación	¿Se documentan los casos de prueba?	0= no 1=si	Analista de Pruebas	1	1	1	1	1
	Ejecución de Casos de Prueba	Ejecución de forma manual	¿Se ejecutan los casos de	0= no 1=si	Analista de Pruebas	1	1	1	1	1

Desarrollo (ENG3)	(ENG 3.1)		prueba de forma manual?							
		Ejecución de forma automática	¿Se ejecutan los casos de prueba de forma automática?	0= no 1=si	Analista de Pruebas	1	1	1	1	0
	Ejecución de la plataforma (ENG 3.2)	Ejecución exitosa de la plataforma	¿Se ejecutan exitosamente la plataforma necesaria para desplegar el sistema bajo prueba?	0= no 1=si	Analista de Pruebas	1	1	1	1	1
		Ejecución del Sistema (ENG 3.3)	Ejecución de acuerdo a los casos de prueba	¿Se ejecuta el sistema de acuerdo a los casos de prueba?	0= no 1=si	Analista de Pruebas	1	1	1	1
	Utilización de los datos de prueba		¿Se utilizan los datos de prueba especificados en el caso de	0= no 1=si	Analista de Pruebas	1	1	1	1	1

			prueba?							
		Herramientas para la ejecución	¿Se dispone de herramientas para la ejecución de la prueba?	0= no 1=si	Analista de Pruebas	1	1	1	1	0
Criterios de Terminación (ENG4)	Análisis de Resultados (ENG 4.1)	Existencia de errores	¿El software probado tiene errores?	0= no 1=si	Analista de Pruebas	1	1	1	1	1
		Generación de incidencias	¿En caso de existir errores se genera una incidencia?	0= no 1=si	Analista de Pruebas	1	1	1	1	1
		Comunicación de incidencias	¿Se comunica al desarrollador la incidencia?	0= no 1=si	Analista de Pruebas	1	1	1	1	1
		Suspensión de la prueba	¿La prueba fue suspendida por problemas del entorno?	0= no 1=si	Analista de Pruebas	1	1	1	1	0
		Cumplimiento de los objetivos	¿La prueba cumple con las funcionalidad	0= no 1=si	Analista de Pruebas.	1	1	1	1	1

			s de acuerdo a los objetivos de las pruebas?							
		Certificación de las pruebas	¿Se solicita al usuario la certificación de las pruebas realizadas?	0= no 1=si	Analista de Pruebas. Desarrollador. Usuario. Líder de Pruebas.	1	1	1	1	0
		Comunicación de éxito	¿Se comunica al desarrollador el éxito del proceso de pruebas?	0= no 1=si	Analista de Pruebas. Desarrollador.	1	1	1	1	0
	Determinar Riesgos (ENG 4.2)	Estudio de riesgos de fallos potenciales	¿Se estudian los riesgos de fallos potenciales	0= no 1=si	Líder de Pruebas. Analista de Pruebas	1	1	1	1	0
	Determinar Costos (ENG 4.3)	Costos de continuación de las pruebas	¿Se consideran los costos de continuación de las pruebas?	0= no 1=si	Líder de Pruebas. Analista de Prueba.	1	1	1	1	0

Reportes (ENG5)	Incidencias (ENG 5.1)	Generación de código para las incidencias	¿Se genera un id por cada una de las incidencias detectadas?	0= no 1=si	Analista de Pruebas.	1	1	1	1	1
		Comunicación de incidencias al equipo de desarrollador	¿Se comunican las incidencias detectadas al responsable del desarrollo?	0= no 1=si	Analista de Pruebas Desarrolladores.	1	1	1	1	1
	Culminación de Pruebas (ENG 5.2)	Generación de reportes de culminación de las pruebas	¿Se generan reportes una vez finalizada las pruebas de software?	0= no 1=si	Analista de Pruebas	1	1	1	1	0
		Certificación (ENG 5.3)	Generación de reportes de certificación de las pruebas	¿Se generan reportes de certificación de las pruebas del	0= no 1=si	Analista de Prueba.	1	1	1	1

			elemento de software?							
Actividades de Cierre (ENG6)	Comprobar Entregables (ENG 6.1)	Transmisión de entregables	¿Los entregables fueron transmitidos según lo planificado?	0= no 1=si	Analista de Prueba. Líder de Pruebas. Desarrolladores	1	1	1	1	1
	Cierre de Incidencias (ENG 6.2)	Cierre de Incidencias	¿Se realizó el cierre de incidencias?	0= no 1=si	Analista de Prueba, Líder de Pruebas, Desarrolladores	1	1	1	1	1
	Ambiente de Prueba (ENG 6.3)	Almacenamiento del ambiente de pruebas	¿Se guarda el ambiente de prueba para futuras pruebas?	0= no 1=si	Analista de Prueba.	1	1	1	1	1
Documentación(SUP1)	Solicitud de Pruebas (SUP 1.1)	Generación de solicitud de pruebas	¿Se genera la solicitud de pruebas del elemento a probar?	0= no 1=si	Desarrolladores	1	1	1	1	1

		Generación de código para cada solicitud de pruebas	¿Se genera un identificador de prueba para cada solicitud de pruebas recibida?	0= no 1=si	Líder de Pruebas	1	1	1	1	0
		Asignación de especialista de pruebas	¿Se asigna un especialista de pruebas para cada ticket de prueba generado?	0= no 1=si	Líder de Pruebas	1	1	1	1	1
	Soporte de Ejecución (SUP 1.2)	Estándar de documentación de la ejecución	¿Se sigue un estándar en la documentación durante la ejecución de las pruebas?	0= no 1=si	Líder de Pruebas Analista de Pruebas	1	1	1	1	1
		Estándar en el repositorio de documentación de ejecución de las pruebas	¿El repositorio de documentación de ejecución de pruebas posee una estructura	0= no 1=si	Líder de Pruebas Analista de Pruebas	1	1	1	1	0

			de directorios estándar?							
		Almacenamiento de evidencias	¿Se guardan las evidencias (pantallas, scripts con sus respectivas salidas, logs, entre otros) de la ejecución de los casos de prueba?	0= no 1=si	Analista de Pruebas	1	1	1	1	1
	Soporte de Incidencias (SUP 1.3)	Estándar en el repositorio de documentación de las incidencias.	¿El repositorio de documentación de las incidencias detectadas durante la ejecución de pruebas posee una estructura de directorios estándar?	0= no 1=si	Líder de Pruebas Analista de Pruebas	1	1	1	1	0

		Almacenamiento de incidencias	¿Se almacenan las incidencias (pantallas, scripts con sus respectivas salidas, logs, entre otros) de la incidencia detectada?	0= no 1=si	Líder de Pruebas Analista de Pruebas	1	1	1	1	1
		Asociación de número de incidencia con el número de caso de prueba	¿El número de incidencia está asociado con el número del caso de prueba ejecutado?	0= no 1=si	Líder de Pruebas Analista de Pruebas	1	1	1	1	1
	Documentos Recibidos (SUP 1.4)	Almacenamiento de documentos recibidos	¿Se almacenan los documentos recibidos para la ejecución de las pruebas?	0= no 1=si	Líder de Pruebas Analista de Pruebas	1	1	1	1	1
	Soporte de	Formato de				1	1	1	1	0

	Certificación (SUP 1.5)	certificación	¿Se cuenta con un formato para la certificación de las pruebas realizadas?	0= no 1=si	Líder de Pruebas Analista de Pruebas					
		Almacenamiento de la certificación de las pruebas	¿Se almacena la certificación de las pruebas realizadas?	0= no 1=si	Líder de Pruebas Analista de Pruebas	1	1	1	1	0
	Repositorio de Documentación (SUP 1.6)	Almacenamiento en el repositorio de la documentación generada durante todo el proceso de pruebas	¿La documentación generada durante el proceso de pruebas se almacena en el repositorio de documentación ?	0= no 1=si	Líder de Pruebas Analista de Pruebas	1	1	1	1	1
Aseguramiento de la Calidad (SUP2)	Plan de calidad (SUP 2.1)	Elaboración del plan de calidad	¿Se elabora el plan de calidad para el proceso de pruebas?	0= no 1=si	Líder de Pruebas	1	1	1	1	1

		Asignación de roles	¿Se asignan roles a las personas encargadas de la ejecución de las pruebas?	0= no 1=si	Líder de Pruebas	1	1	1	1	1
		Comunicación del plan de calidad	¿Se comunica el plan de calidad a seguir a todos los involucrados?	0= no 1=si	Líder de Pruebas, Analista de Pruebas, Desarrolladores	1	1	1	1	1
	Seguimiento de defectos (SUP 2.2)	Fallos por requisitos mal definidos	¿Los fallos observados, se deben a requisitos mal definidos?	Fallos por requisitos mal definidos	¿Los fallos observados, se deben a requisitos mal definidos?	1	1	1	1	1
		Fallos por declaración de variables incorrectas	¿Los fallos observados, se deben a declaraciones incorrectas de variables?	Fallos por declaración de variables incorrectas	¿Los fallos observados, se deben a declaraciones incorrectas de variables?	1	1	1	1	0

					incorrectas de variables?					
		Fallos por errores de programación	¿Los fallos observados, se deben a errores de programación?	Fallos por errores de programación	¿Los fallos observados, se deben a errores de programación?	1	1	1	1	1
	Seguimiento de actividades (SUP 2.3)	Actividades dentro del plan de pruebas	¿Se verifica que las actividades estén dentro del plan de pruebas?	0= no 1=si	Líder de Pruebas	1	1	1	1	1
		Artefactos en conformidad con los estándares.	¿Se realizan revisiones a los artefactos a verificar su conformidad con los estándares?	0= no 1=si	Líder de Pruebas	1	1	1	1	0
		Registro de	¿Se registran	0= no	Líder de	1	1	1	1	0

		auditorias	las auditorías realizadas?	1=si	Pruebas					
		Control de documentación	¿Se realiza un control de la documentación de pruebas?	0= no 1=si	Líder de Pruebas	1	1	1	1	0
Revisión Conjunta (SUP3)	Ejecución del Proceso (SUP 3.1)	Solicitud de revisión conjunta	¿Se solicitan revisiones periódicas de las actividades contempladas en el plan de pruebas de software?	0= no 1=si	Analista de Pruebas Líder de Pruebas Usuario	1	1	1	1	1
		Definición de recursos a utilizar	¿Se establecen los recursos a utilizar en las revisiones del proceso de pruebas de software?	0= no 1=si	Líder de Pruebas Analista de Pruebas Usuario	1	1	1	1	0
		Especificación del alcance	¿Se especifica el alcance de las revisiones	0= no 1=si	Líder de Pruebas Analista de	1	1	1	1	0

			del proceso de pruebas de software?		Pruebas Usuario					
		Documentación de problemas detectados	¿Se documentan los problemas detectados durante el proceso de revisión?	0= no 1=si	Líder de Pruebas Analista de Pruebas Usuario	1	1	1	1	0
		Comunicación de resultados	¿Se comunican los resultados del proceso de revisión conjunta?	0= no 1=si	Líder de Pruebas Analista de Pruebas Usuario	1	1	1	1	1
Auditoría (SUP4)	Ejecución del Proceso (SUP 4.1)	Solicitud de auditorías.	¿Se solicitan auditorías periódicas de las actividades desarrolladas en el proceso de pruebas de software?	0= no 1=si	Analista de Pruebas Líder de Pruebas Usuario	1	1	1	1	0
			¿Se establecen	0= no	Líder de	1	1	1	1	0

		Especificación de los responsables de la ejecución de la auditoria.	los responsables de la ejecución de la auditoría del proceso de pruebas de software?	1=si	Pruebas					
		Alcance de la auditoria	¿Se especifica el alcance de las auditorías del proceso de pruebas de software?	0= no 1=si	Líder de Pruebas Analista de Pruebas Usuario	1	1	1	1	0
		Documentación de la auditoria	¿Se documentan los resultados de la evaluación del proceso de pruebas de software?	0= no 1=si	Líder de Pruebas Analista de Pruebas	1	1	1	1	0
		Comunicación de los resultados de la auditoria	¿Se comunican los resultados del proceso de	0= no 1=si	Líder de Pruebas Analista de	1	1	1	1	0

			auditoría del proceso de pruebas de software?		Pruebas Usuario					
Gestión (MAN1)	Gestión de Planificación (MAN 1.1)	Definición de alcance de las pruebas	¿Se define el alcance de las pruebas?	0= no 1=si	Analista de Pruebas Líder de Pruebas	1	1	1	1	1
		Definición de responsables de la ejecución	¿Se definen los responsables de la ejecución de las pruebas de software?	0= no 1=si	Líder de Pruebas	1	1	1	1	1
		Identificación de los requisitos de las pruebas	¿Se identifican los requisitos de las pruebas?	0= no 1=si	Líder de Pruebas Analista de Pruebas Usuario	1	1	1	1	1
		Definición de escenarios de	¿Se definen los escenarios de	0= no 1=si	Líder de Pruebas	1	1	1	1	0

	prueba	prueba?		Analista de Pruebas					
Ejecución del Proceso (MAN 1.2)	Asignación de recursos	¿Se asignan los recursos para la ejecución de las pruebas?	0= no 1=si	Líder de Pruebas Analista de Pruebas	1	1	1	1	1
	Configuración el entorno de pruebas	¿Se realiza la configuración del entorno de pruebas?	0= no 1=si	Analista de Pruebas	1	1	1	1	1
	Ejecución del plan de pruebas	¿Se ejecuta el plan de pruebas?	0= no 1=si	Analista de Pruebas	1	1	1	1	1
	Elaboración de los casos de prueba	¿Se elaboran los casos de prueba?	0= no 1=si	Analista de Pruebas	1	1	1	1	1
	Descripción de las actividades	¿Se describen las actividades de prueba a realizar?	0= no 1=si	Analista de Pruebas	1	1	1	1	1
	Información	¿Se informa	0= no	Líder de	1	1	1	1	1

		acerca de las incidencias detectadas	acerca de las incidencias detectadas?	1=si	Pruebas Analista de Pruebas Desarrolladores					
Seguimiento de las Pruebas (MAN 1.3)	Definición de puntos de control	¿Se definen puntos de control durante la ejecución del proceso de pruebas?	0= no 1=si	Líder de Pruebas Analista de Pruebas	1	1	1	1	1	1
	Seguimiento de la ejecución de las pruebas	¿Se realiza un seguimiento de la ejecución de las pruebas?	0= no 1=si	Líder de Pruebas Analista de Pruebas	1	1	1	1	1	1
	Informe acerca del avance de las pruebas	¿Se informa acerca del avance del proceso de pruebas?	0= no 1=si	Líder de Pruebas Analista de Pruebas	1	1	1	1	1	0
	Gestión de la	¿Se gestiona la	0= no	Líder de	1	1	1	1	1	1

		calidad del proceso de pruebas	calidad en el proceso de pruebas?	1=si	Pruebas Analista de Pruebas					
		Verificación de la ejecución de los casos de prueba	¿Se verifica la ejecución de los casos de prueba elaborados?	0= no 1=si	Líder de Pruebas Analista de Pruebas Desarrolladores	1	1	1	1	0
	Cierre del Proceso (MAN 1.4)	Documentación elaborado cumple los estándares	¿Se verifica si la documentación elaborada cumple con los estándares establecidos?	0= no 1=si	Líder de Pruebas Analista de Pruebas Desarrolladores	1	1	1	1	0
Gestión de Calidad (MAN2)	Niveles de Cobertura(MAN 2.1)	Verificación de los niveles de cobertura	¿Se verifican los niveles de cobertura de las pruebas establecidos en	0= no 1=si	Analista de Pruebas Líder de Pruebas	1	1	1	1	0

			el plan de pruebas?							
		Rango de fallos detectados	¿Se establecen rango de fallos detectados?	0= no 1=si	Analista de Pruebas	1	1	1	1	0
		Verificación de los fallos detectados	¿Se verifica que los fallos detectados no se presentan en una iteración posterior?	0= no 1=si	Analista de Pruebas Desarrolladores	1	1	1	1	1
Lineamientos Organizacionales (ORG1)		Selección de metodología	¿Se selecciona alguna metodología para el proceso de pruebas?	0= no 1=si	Líder de Pruebas	1	1	1	1	0
		Estándar para la ejecución de las pruebas	¿Se incluye algún estándar para la ejecución del proceso de pruebas?	0= no 1=si	Líder de Pruebas	1	1	1	1	1

		Definición de fases	¿Se definen las fases a seguir en las pruebas de software?	0= no 1=si	Líder de Pruebas	1	1	1	1	0
		Duración de las fases	¿Se definen la duración de cada una de las fases del proceso de pruebas?	0= no 1=si	Líder de Pruebas	1	1	1	1	0
		Definición de artefactos	¿Se definen los principales artefactos del proceso de pruebas?	0= no 1=si	Líder de Pruebas	1	1	1	1	0
		Definición de canales de comunicación	¿Se definen los canales de comunicación que se deben seguir durante	0= no 1=si	Líder de Pruebas	1	1	1	1	1

			el proceso de pruebas?							
Evaluación del Proceso (ORG 2)		Verificación del número de casos de prueba especificados	¿Se verifica el número de casos de prueba especificados?	0= no 1=si	Líder de Pruebas Analista de Pruebas	1	1	1	1	1
		Verificación del número de casos de prueba ejecutados	¿Se verifica el número de casos de prueba ejecutados?	0= no 1=si	Líder de Pruebas Analista de Pruebas	1	1	1	1	1
		Verificación del número de casos de prueba con incidencias	¿Se verifica el número de casos de prueba con incidencias?	0= no 1=si	Líder de Pruebas Analista de Pruebas	1	1	1	1	1
		Verificación del número de casos	¿Se verifica el número de	0= no 1=si	Líder de Pruebas	1	1	1	1	1

		de prueba exitosos	casos de prueba exitosos?		Analista de Pruebas					
Reuso (ORG3)		Documentación de las soluciones	¿Se documentan las soluciones adaptadas en determinados tipos de aplicaciones?	0= no 1=si	Analista de Pruebas	1	1	1	1	1
		Determinar artefactos reutilizables	¿Se establecen los artefactos reutilizables?	0= no 1=si	Líder de Pruebas	1	1	1	1	0
		Repositorio de material de pruebas	¿Se cuenta con un repositorio de material de pruebas?	0= no 1=si	Analista de Pruebas	1	1	1	1	1

		Garantía de permanencia de la información en el repositorio de pruebas	¿Se garantiza la permanencia de la información en el repositorio de pruebas?	0= no 1=si	Líder de Pruebas	1	1	1	1	1
		Garantía de accesibilidad de la información del repositorio de pruebas	¿Se garantiza accesibilidad a la información contenida en el repositorio de pruebas?	0= no 1=si	Líder de Pruebas	1	1	1	1	1
Planificación (TRA1)		Análisis de necesidades de entrenamiento del personal  Diseño de planes de capacitación	¿Se analizan las necesidades de entrenamiento del personal del área de pruebas?	0= no 1=si	Líder de Pruebas	1	1	1	1	1

		Determinación de recursos necesarios para la implementación del plan	¿Se diseñan planes de capacitación para el personal del área de pruebas?	0= no 1=si	Líder de Pruebas	1	1	1	1	0
			¿Se determinan los recursos necesarios para implementar el plan de capacitación?	0= no 1=si	Líder de Pruebas	1	1	1	1	1
Ejecución (TRA 2)		Desarrollo de manuales	¿Se desarrolla manuales de entrenamiento?	0= no 1=si	Líder de Pruebas	1	1	1	1	0
						1	1	1	1	1

		Contenidos ajustados a las actividades realizadas en el proceso de pruebas	¿Los contenidos impartidos, se ajustan a las actividades realizadas en el proceso de pruebas?	0= no 1=si	Analista de Pruebas					
		Personal capacitado	¿El personal encargado de impartir el entrenamiento posee experiencia en el área de pruebas?	0= no 1=si	Analista de Pruebas	1	1	1	1	1
Actividades de Cierre (TRA 3)		Información del personal capacitado	¿Se almacena la información del personal adiestrado?	0= no 1=si	Líder de Pruebas	1	1	1	1	1

		Registro de cursos impartidos	¿Se registra los cursos impartidos?	0= no 1=si	Líder de Pruebas	1	1	1	1	1
		Entrega de certificados	¿Se entrega certificado al participante en la capacitación?	0= no 1=si	Líder de Pruebas	1	1	1	1	1
		Evaluación del beneficio del adiestramiento	¿Se evalúa el beneficio del adiestramiento al personal?	0= no 1=si	Líder de Pruebas	1	1	1	1	1

**Anexo III.** Resultados de la Evaluación del Modelo en el Proceso de Pruebas de Software del desarrollo del Sistema de Inscripción para el Control de Actividades Académicas (SICAA)

Categoría	Características	Sub-características	Porcentaje Sub- Característica	Porcentaje Característica	Porcentaje Categoría
Cliente- Proveedor(CUS)	Determinación de Requerimientos (CUS1)	Documento Funcional(CUS 1.1)	100%	75%	75%
		Definición del Alcance (CUS 1.2)	50%		
Ingeniería (ENG)	Planificación y Control (ENG1)	Determinar el alcance de las pruebas (ENG 1.1)	100%	55.33%	54%
		Desarrollar el plan de Pruebas (ENG 1.2)	66%		
		Entorno de Pruebas (ENG 1.3)	0%		
	Análisis y Diseño (ENG2)	Definición de los elementos a probar (ENG 2.1)	33%	55%	
		Generación de casos de Prueba (ENG 2.2)	77%		
	Desarrollo (ENG3)	Ejecución de Casos de Prueba	50%		

		(ENG 3.1)		75%	
		Ejecución de la plataforma (ENG 3.2)	100%		
		Ejecución del Sistema (ENG 3.3)	75%		
	Criterios de Terminación (ENG4)	Análisis de Resultados (ENG 4.1)	57%	19%	
		Determinar Riesgos (ENG 4.2)	0%		
		Determinar Costos (ENG 4.3)	0%		
	Reportes (ENG5)	Incidencias (ENG 5.1)	100%	66%	
		Culminación de Pruebas (ENG 5.2)	100%		
		Certificación (ENG 5.3)	0%		
	Actividades de Cierre (ENG6)	Comprobar Entregables (ENG 6.1)	100%	66%	
		Cierre de Incidencias (ENG 6.2)	100%		
		Ambiente de Prueba (ENG 6.3)	0%		
Soporte(SUP)	Documentación(SUP1)	Soporte de Ejecución (SUP 1.2)	66%		
		Soporte de Incidencias (SUP 1.3)	66%		
		Documentos Recibidos (SUP 1.4)	100%		
		Soporte de Certificación (SUP	0%		

		1.5)		66%	47.25
		Repositorio de Documentación (SUP 1.6)	100%		
	Aseguramiento de la Calidad (SUP2)	Plan de calidad (SUP 2.1)	100%	63%	
		Seguimiento de defectos (SUP 2.2)	66%		
		Seguimiento de actividades (SUP 2.3)	25%		
	Revisión Conjunta (SUP3)	Ejecución del Proceso (SUP 3.1)	60%	60%	
	Auditoría (SUP4)	Ejecución del Proceso (SUP 4.1)	0%	0%	
Gestión(MAN)	Gestión (MAN1)	Gestión de Planificación (MAN 1.1)	75%	62.5%	47,75
		Ejecución del Proceso (MAN 1.2)	100%		
		Seguimiento de las Pruebas (MAN 1.3)	75%		
		Cierre del Proceso (MAN 1.4)	0%		
	Gestión de Calidad (MAN2)	Niveles de Cobertura(MAN 2.1)	33%	33%	
Organizacional(ORG)	Lineamientos Organizacionales (ORG1)			60%	80%
	Evaluación del Proceso(ORG 2)			100%	

	Reuso(ORG3)			80%	
Entrenamiento(TRA)	Planificación(TRA1)			66%	77.33
	Ejecución(TRA 2)			66%	
	Actividades de Cierre(TRA 3)			100%	

