

**UNIVERSIDAD CATÓLICA ANDRÉS BELLO
VICERRECTORADO ACADÉMICO
DIRECCIÓN GENERAL DE LOS ESTUDIOS DE POSTGRADO
ÁREA DE CIENCIAS ADMINISTRATIVAS Y DE GESTIÓN
POSTGRADO EN GERENCIA DE PROYECTOS**

**TRABAJO ESPECIAL DE GRADO
PROPUESTA DE UN MODELO DE GESTIÓN DE CALIDAD DEL
SOFTWARE PARA SELECCIONAR HERRAMIENTAS DEL
SOFTWARE**

Presentado por

ANDREINA DEL CARMEN CASTILLO ESPINOZA

Para optar al título de:

ESPECIALISTA EN GERENCIA DE PROYECTOS

Asesor

MARIA ANGÉLICA PÉREZ DE OVALLES

Caracas, Diciembre de 2007

DEDICATORIA

A Dios, que me das la oportunidad de vivir, brindándome la sabiduría, paciencia y perseverancia a lo largo de todo el desarrollo del presente trabajo, así como también la fortaleza que necesitaba en los momentos más difíciles; toda la felicidad y beneficios que he recibido en mi vida te los debo sin duda alguna a ti.

A mis padres, quienes han estado conmigo en todo momento. Gracias papá por darme un carrera para mi futuro y por creer en mi. Gracias mamá, por brindarme tu amor, cuidarme y recordarme en todo momento que no podía darme el lujo de desfallecer

A mis hermanos, gracias por estar conmigo y apoyarme siempre.

Andreína

.1

.2 AGRADECIMIENTOS

Quiero agradecer a todas aquellas personas que de una u otra forma han contribuido al logro de esta meta tan importante en mi vida, entre ellas quiero expresar mi más profundo agradecimiento a:

- **La Profesora María Angélica Ovalles, mi asesora, le doy las gracias por creer en mi y apoyarme desde un principio en la elaboración de este trabajo, y por brindarme un mejor panorama de lo que debía hacer, sin su ayuda no sería posible haber terminado.**

- *La Profesora Edumilis Méndez, quien me dio la oportunidad de guiarme en el desarrollo del presente Trabajo, prestándome su colaboración en todo momento.*

También quiero expresar mi agradecimiento a todos aquellos quienes han colaborado ya sea directa o indirectamente en la elaboración de este trabajo de grado.

ÍNDICE GENERAL

	Pág.
Índice de Tablas	vi
Índice de Figuras	vii
Resumen	viii
INTRODUCCIÓN	01
 CAPÍTULOS	
I. EL PROBLEMA	03
Planteamiento del Problema	03
Objetivos de la Investigación	07
<i>Objetivo General</i>	07
<i>Objetivos Específicos</i>	07
Limitaciones y Alcance	09
Justificación	09
Consideraciones Éticas	11
II. MARCO TEÓRICO	15
Antecedentes	
Modelo de Decisión para soportar la Selección de Herramientas CASE	15
Indicadores para la selección de herramientas de calidad del Software	17
Modelo Sistémico de Calidad (MOSCA)	20
Bases Teóricas	
Calidad del Software	24
Modelos de Calidad del Producto	28
Gestión de Calidad del Software	28
Herramientas del Software	45
III. MARCO METODOLÓGICO	
Framework Metodológico	53
Cronograma de Ejecución	60
IV. MARCO ORGANIZACIONAL	61
Reseña Histórica	61
Misión	61
Vinculación con el Sector Productivo	62

Adopción Tecnológica	65
Visión Futura	66
V DESARROLLO	67
Mapa conceptual asociado a las Herramientas de la Gestión de la Calidad del Software	67
Identificación de los elementos presentes en el Modelo Sistémico de Calidad MOSCA	73
Propuesta del Modelo de Gestión de Calidad del Software para seleccionar herramientas del software	79
VI ANÁLISIS DE LOS RESULTADOS	99
Análisis de los resultados obtenidos – Categoría Funcionalidad	99
Análisis de los resultados obtenidos – Categoría Usabilidad	100
Análisis de los resultados obtenidos – Categoría Fiabilidad	102
VII CONCLUSIONES Y RECOMENDACIONES	106
Conclusiones	106
Recomendaciones	107
REFERENCIAS BIBLIOGRÁFICAS	108
APÉNDICES	112
APÉNDICE I MODELO SISTÉMICO DE CALIDAD (MOSCA)	113
APÉNDICE II MODELOS DE CALIDAD DEL SOFTWARE	134

ÍNDICE DE TABLAS

TABLA	Pág
01 Tabla Semántica de Indicadores	16
02 Indicadores Tecnológicos.	18
03 Indicadores Organizacionales.	20
04 Categorías seleccionadas del Modelo MOSCA	24
05 Tabla Resumen de la Definición de Gestión de Calidad del Software propuesta por Sommerville, Pressman , SWEBOK y Unhelkar	39
06 Tabla Resumen de las Actividades de cada uno de los Procesos de la Gestión de Calidad del Software	43
07 Tabla Resumen de las Definiciones de Herramientas de Software propuesta por SWEBOK, Pressman ySommerville,	46
08 Fases del Método de Investigación Acción	54
09 Fases y Actividades de la Metodología de Investigación Acción Propuesta	57
10 Categorías y características de la perspectiva Producto del Modelo MOSCA	73
11 Propuesta del Modelo de Gestión de la Calidad del Software para seleccionar herramientas del software	79
12 Análisis de Resultados - Funcionalidad	99
13 Análisis de Resultados – Usabilidad	100
14 Análisis de los Resultados – Fiabilidad	102
15 Resumen	105

ÍNDICE DE FIGURAS

FIGURA		Pág
01	Diagrama de Ishikawa – Planteamiento del Problema	08
02	Diagrama de Ishikawa – Justificación del Problema	12
03	Modelo de Calidad Sistemica del Software (MOSCA)	22
04	Matriz de Calidad Global Sistémica	27
05	Análisis de los Tópicos para la Calidad del Software	33
06	Actividades de la Gestión de Calidad	36
07	Actividades del Aseguramiento de Calidad	38
08	Actividades del Control de Calidad	40
9	Tópicos de las HDS, como subárea de conocimiento de la IS	50
10	El ciclo de Investigación Acción	54
11	Framework Metodológico para el trabajo de grado	56
12	Mapa Conceptual: Calidad del Software	68
13	Mapa Conceptual: Herramientas del Software	70
14	Mapa Conceptual: Gestión de la Calidad del Software	71
14	Mapa Conceptual	72



UNIVERSIDAD CATÓLICA ANDRÉS BELLO
VICERRECTORADO ACADEMICO
DIRECCIÓN GENERAL DE LOS ESTUDIOS DE POSTGRADO
AREA DE CIENCIAS ADMINISTRATIVAS Y DE GESTION
POSTGRADO EN GERENCIA DE PROYECTOS

Título: Modelo De Gestión De Calidad Del Software para seleccionar Herramientas del Software.

Nombre del Autor: Andreina Castillo

Nombre del Asesor: María Angélica Ovalles

Año: 2008

El desarrollo de los Sistemas de Información como arte nuevo, ha ido sufriendo cambios que buscan asegurar que se alcancen los resultados esperados cuando se comienza un nuevo desarrollo. Si bien podría ser fácil definir los objetivos de un Sistema de Información, mantener el camino correcto para poder alcanzarlo y que este camino mantenga el equilibrio necesario entre la eficiencia y la efectividad es una tarea difícil. Esto es la búsqueda de Sistemas de Información de Calidad. En la actualidad, las organizaciones que desarrollan Sistemas de Información en búsqueda de lograr la Calidad y mantenerse competitivas se ven en la necesidad de hacer uso de las Herramientas del Software, ya que desempeñan un papel importante en el desarrollo de las aplicaciones y ayuda al equipo a valorar y mejorar la Calidad, sin embargo, tomar la decisión para seleccionarlas resulta una tarea compleja para las organizaciones desarrolladoras de software que hacen uso de ellas, dado a la extensiva variedad de las herramientas, el costo representativo de cada una de ellas, las características del proyecto y lo poco objetivos que son los vendedores al dar características de sus herramientas, reduciendo así la posibilidad de desarrollar trabajos de calidad y mantener a estas organizaciones en un alto nivel competitivo. Dada la diversidad de factores que afectan la adopción de una herramienta, lo complejas que son por la cantidad de componentes que pueden ofrecer, la presente propuesta de Trabajo de Grado tiene como objetivo principal dar

orientación para seleccionar las Herramientas del Software de la Gestión de la Calidad a través de la propuesta de un modelo.

Palabras Claves: Calidad, Software, Herramientas, Modelos, Organizaciones.

INTRODUCCIÓN

El proceso de Globalización que se vive en la actualidad, unido a la alta competencia experimentada para obtener mejores productos en los últimos años, exige que las empresas desarrollen productos de Calidad para poder mantenerse competitivas.

En la actualidad, las organizaciones que desarrollan Sistemas de Información como respuesta a esta presión se ven en la necesidad de hacer uso del enfoque de Herramientas del Software. Este enfoque busca aumentar la eficacia de los procesos de desarrollo de los Sistemas de Información, al soportar la realización de sus tareas con el uso de las tecnologías de información. Esta es la razón por la cual el mercado de las Herramientas del Software está en franca expansión.

Sin embargo, cuando una empresa toma la decisión de mejorar su proceso de desarrollo de Sistemas de Software, en especial los procesos correspondientes a la Gestión de la Calidad, a través del uso de Herramientas del Software, se enfrenta con la problemática de la selección de la tecnología más adecuada, dada la cantidad de aspectos que pueden afectar la adopción de la herramienta, y lo complejas que son por la cantidad de funciones que pueden ofrecer, por lo tanto, la selección de las herramientas del software es un proceso estratégico muy importante para la organización que desarrolla Sistemas de Software, dado el hecho de que la selección oportuna y acertada de éstas tiene gran valor competitivo.

En ese sentido, se propone la presente propuesta de Trabajo de Grado, el cual propone un Modelo de Gestión de Calidad para seleccionar Herramientas del Software, específicamente, las herramientas correspondientes a los procesos de la Gestión de la Calidad, logrando así

cubrir con una de las áreas de conocimiento de la Gerencia de Proyectos y limitar la investigación en cuanto al alcance de la misma.

Con la finalidad de mostrar claramente el contenido del trabajo desarrollado, se presenta a continuación la manera como está estructurado el mismo:

En el Capítulo I, se presenta el Planteamiento del Problema, con la finalidad de ofrecer un ámbito general sobre la necesidad de desarrollar la presente propuesta de Trabajo de Grado, los objetivos, las limitaciones y alcances del mismo.

En el Capítulo II, se presentan todos los conceptos que conforman el marco conceptual, con el objetivo de ubicar al lector en los fundamentos de dicha adaptación.

En el Capítulo III, se explican la metodología a utilizar, para desarrollar la Propuesta del Modelo de Gestión de la Calidad para la seleccionar herramientas del software.

En el Capítulo IV, se detalla el marco organizacional, con el cual se ofrecerá una breve descripción acerca de la empresa para la cual se desarrollará el Trabajo de Grado.

En el Capítulo V, se detalla el proceso de desarrollo de la propuesta del Modelo de Gestión de Calidad para seleccionar herramientas del software, indicando la instanciación del Modelo Sistémico de Calidad MOSCA.

En el Capítulo VI, se realiza un análisis de los resultados la propuesta obtenida.

Finalmente, en el Capítulo VII, se exponen las conclusiones y recomendaciones emanadas del desarrollo de toda la investigación realizada.

CAPITULO I

EL PROBLEMA

El objetivo principal de este capítulo es proporcionar una descripción clara y precisa sobre el contexto en el cual se desenvuelve la investigación, para lograr ciertos objetivos, y determinando límites razonables.

Planteamiento del Problema

En esta sección se hace referencia a los aspectos, que sirven de base para plantear el problema que motiva la realización de la investigación (Tamayo y Tamayo, 1997).

La Calidad debe estar presente en cualquier producto o servicio. Por esta razón las organizaciones han adoptado estándares y han mejorado muchos de sus procesos para lograr niveles de aceptación en el mercado internacional. Las organizaciones que desarrollan Sistemas de Información no son la excepción. Ellas deben ofrecer un alto nivel de calidad en sus productos para mantenerse competitivos en un mercado como el actual. Esta situación conduce a los desarrolladores a buscar soluciones para obtener calidad en sus productos. Ellos deben definir el atributo de calidad de los mismos y también determinar que el desarrollo de producto de software sea mejorado (Pérez et ad., 2001).

La Calidad del Software es un proceso complejo y muchas veces arroja como resultado la necesidad de mejorar los niveles de calidad encontrados. Para estructurar las causas, consecuencias y posibles soluciones, se hace necesario recurrir al Enfoque de Sistemas como una metodología de cambio que permita comprender que todos los elementos (internos y externos) están

interconectados y guardan una relación de afectación hacia la calidad. La calidad del software es un concepto multidimensional que no se puede definir de forma simple. Clásicamente, la noción de calidad implica que el producto desarrollado cumple su especificación. La calidad total no se debe entender como la suma de calidades parciales, no tiene ningún sentido diseñar un sistema altamente eficiente si no se utiliza; así como, no tiene ningún sentido diseñar un sistema muy efectivo si no es factible por los recursos y/o apremios del tiempo. Es por ello, que se deben considerar los recursos humanos, tecnológicos, financieros y de tiempo en cualquier diseño para el mínimo cumplimiento de un proceso eficaz sin la coerción de la eficacia del producto (Pérez., 2005).

Una forma de obtener calidad del software es a través del uso de herramientas de la gestión de calidad del software durante el proceso de desarrollo, las cuales contribuyen a la producción de software de alta calidad, a bajo costo y en el menos tiempo posible. Sin embargo, sus beneficios potenciales serán logrados sólo si estas herramientas son seleccionadas correctamente. Como consecuencia del avance tecnológico éstas han experimentado también continuos cambios aumentando la existencia de las mismas, garantizando así un alto nivel de variedad, cada una con sus costos respectivos (Rivas et al., 2007).

Kavi y Nahouraii (1996), plantean que la calidad del software puede verse fuertemente mejorada por una correcta selección de la herramienta que apoye cada fase del proceso de desarrollo, desde el análisis de requerimientos hasta las pruebas finales y la integración. Contrariamente, una inapropiada selección de la herramienta, puede actualmente impedir el desarrollo de un sistema.

La calidad del software puede verse fuertemente mejorada por una correcta selección de la herramienta que apoye cada fase del proceso de desarrollo, desde el análisis de requerimientos hasta las pruebas finales y la integración. Contrariamente, una inapropiada selección de la herramienta,

puede actualmente impedir el desarrollo de un sistema.

El impacto de una herramienta no es solamente debido a las propiedades inherentes a ella, sino también a las características del proyecto. Dos características del proyecto: su tamaño y las actividades a desarrollar, son particularmente importantes pues dan las pautas de las funciones que se requieren de la herramienta y la frecuencia de uso que se le dará (Bruckhaus et. al., 1996). La selección de una herramienta debe ser antecedida por la selección de la metodología, para Topper (Topper et. al., 1994) este hecho reduce los riesgos de seleccionar una herramienta inadecuada y por ende, los costos asociados se reducen cuando se selecciona una herramienta en base a una metodología ya entendida y asimilada por la organización. Un proceso formal de evaluación reduce la posibilidad de adquirir productos inadecuados o innecesarios, por lo que se requiere usar métodos de evaluación que permitan medir factores claves en el producto a adquirir (O'Brien, 1999). El proceso de selección que proponen Topper y sus colegas (Topper et. al., 1994), consta de 4 pasos: (1) realizar una revisión exhaustiva de las herramientas disponibles, (2) probar un pequeño grupo de una selección del paso anterior, probarlas a través de un proyecto piloto o de una evaluación más detallada, (3) presentar una puntuación de las herramientas y seleccionar la de puntuación más alta, (4) hacer un desarrollo con la herramienta y realizar un análisis costo beneficio de dicho desarrollo. Posiblemente esta propuesta de Topper y sus colegas, conlleve a una selección certera y acorde a las necesidades de la organización, pero la oportunidad estratégica de la implantación de un sistema se vería amenazada y serían elevados los recursos invertidos en el proceso de selección.

La extensiva variedad de las herramientas de la gestión de calidad del software existentes actualmente, aunado al costo representativo de cada una de ellas y lo poco objetivos que son los vendedores al dar características de sus herramientas no facilita la decisión para la selección de las mismas por

parte de las empresas que desarrollan software, de forma tal que le permitan desarrollar trabajos de calidad y por ende hacerlas competitivas.

Otros factores importantes que influyen en la adopción de una herramienta se refieren a la forma como la organización evalúa sus retornos de inversión, facilidad de compartir los datos entre las herramientas, el uso de un proceso formal de desarrollo con una herramienta que se ajuste a ello (Hoffer et. al., 1999). Además, para estos autores es importante considerar la presencia en la herramienta de los procesos de ingeniería de reverso y re-ingeniería, donde el primero se refiere a la creación de especificaciones de diseño para los sistemas a partir de un código de programa o una especificación de los datos y el segundo, es muy parecido a la ingeniería de reverso pero incluye características de análisis. Por otra parte presenta como factores de rechazo de las herramientas los altos costos de venta y entrenamiento, falta de estándares metodológicos, la visión de amenazas a la seguridad, y el desconocimiento de las herramientas.

Dada la diversidad de factores que afectan la adopción de una herramienta, lo complejas que son por la cantidad de componentes que pueden ofrecer, lo importante que puede ser para las organizaciones desde el punto de vista estratégico la selección oportuna y acertada de herramientas, es de gran importancia apoyar dicho proceso de selección. Por lo tanto, el principal problema presentado es que no es fácil seleccionar herramienta(s) adecuada(s), y es por ello que se hace necesario un modelo que permita evaluar y seleccionar las herramientas, específicamente, las herramientas utilizadas en el proceso de la gestión de Calidad, ofreciendo un soporte y garantizando el éxito de su implantación, el cual permitirá la eficiencia y la efectividad de este proceso. Esta propuesta, estará inspirada en el estudio de modelos de evaluación de la calidad del producto de software. Estos modelos ofrecen mediciones internacionalmente aceptadas tanto del proceso como del producto de desarrollo. Además, proveen un certificado del nivel de calidad logrado e incluso indican las potencialidades

existentes en la mejora de procesos y productos, lo cual es una condición necesaria en un mercado cada vez más dinámico y competitivo (Álvarez, 2000).

A continuación, se presenta el diagrama de Ishikawa para facilitar el análisis del problema presentado.

Objetivos de la Investigación

Los objetivos son fundamentales en una investigación, ya que sin ellos es imposible decidir sobre los medios de realización de la misma (Tamayo y Tamayo, 1997).

Objetivo General

Proponer un modelo de Gestión de Calidad del Software para seleccionar herramientas del software.

Objetivos Específicos

1. Formular un modelo conceptual.
2. Identificar las fortalezas en el estudio de los antecedentes.
3. Diagnosticar el Modelo Sistémico de Calidad (MOSCA), específicamente, la perspectiva de producto.
4. Diseñar una propuesta de un modelo que permita especificar la calidad de las herramientas que soporten el proceso de desarrollo del software.

Una vez hecha la formulación de los objetivos, se hace necesario delimitar el tema, es decir, colocar límites a la investigación y especificar el alcance de esos límites, de forma tal que se le otorgue importancia al tiempo que se le dedicará.

Limitaciones y Alcances

Al realizar una delimitación, se aclara el tipo de investigación, permitiendo así tener una visión general sobre la validez y el grado de confianza que puede tener como resultado. Esto supone determinar el alcance y los límites del tema (Tamayo y Tamayo, 1997).

Para lograr el diseño de la propuesta del modelo que permita especificar la calidad se utilizará el Framework Metodológico del Laboratorio de Investigación en Sistemas de Información (LISI), el cual se inspira en el método Investigación Acción. En este método se detalla un proceso cíclico de cinco fases las cuales son: Diagnosticar, Planificar la acción, Tomar la acción, Evaluar y Especificar el aprendizaje. Para efectos de esta investigación este proceso se llevará a cabo hasta la fase de “Tomar la acción” y dado a que el método de investigación seleccionado es cíclico sólo se realizará un ciclo de la misma.

A continuación se presenta la justificación del presente trabajo de investigación, en la cual se expone la necesidad del desarrollo de una metodología que soporte al Modelo Sistémico de Calidad (MOSCA).

Justificación

En este nuevo milenio, la competencia para obtener mejores productos que satisfagan los requerimientos de los clientes se ha convertido en el eje principal que mueve a las economías en la aldea global. Por esa razón, la calidad es considerada como una de los principales activos con los que cuenta un país para mejorar su competitividad. La clave de la supervivencia de muchas organizaciones a largo plazo es la calidad, no los bajos precios, de forma tal que se mantengan competitivos en un mercado como el actual.

El hecho inminente es que la calidad debe estar presente en cualquier producto, proceso o servicio que sea dirigido a los consumidores. Por lo

tanto, las organizaciones en busca de una aceptación global y para satisfacer las necesidades de sus clientes han tenido que adoptar estándares y mejorar muchos de sus procesos para lograr niveles de aceptación en el mercado internacional.

Una forma de que las empresas desarrolladoras de software obtengan la calidad es a través del uso de herramientas de gestión de calidad del software, las cuales permiten automatizar acciones bien definidas, reduciendo también la carga cognitiva del ingeniero de software, quien requiere libertad para concentrarse en los aspectos creativos del proceso. Este soporte se traduce en mejoras a la calidad y la productividad en el diseño y desarrollo (Rivas et al., 2007).

El soporte que brindan estas herramientas al proceso de desarrollo proporciona importantes ventajas para el equipo de trabajo de Ingeniería del Software. Estas mejoras se sintetizan en: a) Apoyan a las metodologías y métodos, integrando actividades y propiciando visión de continuidad entre fases metodológicas; b) Mejoran la comunicación entre los actores involucrados, facilitándoles compartir su trabajo [y desempeñarlo de forma dinámica e iterativa; c) Establecen métodos efectivos para almacenar y utilizar los datos, lo que permite organizar y correlacionar componentes, para accederlos a través de un repositorio; d) Agregan eficiencia al mantenimiento, ya que los programas son construidos sobre las mismas estructuras y estándares, facilitando la adherencia a la disciplina de diseño y facilitan también la conversión automática de programas a versiones más recientes de lenguajes de programación; y, e) Automatizan porciones del análisis y diseño tediosas y propensas a error, con influencia sobre la generación de código, las pruebas y el control (Rivas et al., 2007).

Sin embargo, una vez que se cuenta con una variada existencia de herramientas de gestión de calidad y conocer los soportes tangibles que brindan, parece lógico buscar un modelo que ayude a seleccionarlas correctamente, ya que los beneficios potenciales de estas herramientas sólo

pueden ser alcanzados si las mismas son seleccionadas y utilizadas de forma correcta. Actualmente, no hay una certeza de que los modelos actuales brinden esta alternativa al contrario es muy probable que se necesite un modelo que permita ofrecer este soporte. En este sentido, en la presente investigación se plantea proponer un modelo que facilite seleccionar las herramientas de la gestión de la calidad del software garantizando el uso correcto de las mismas y la especificación de la calidad.

Por otra parte, al proponer un modelo que ayude especificar la calidad de la gestión de la calidad del software en Venezuela, se tiene la particularidad de que es de origen nacional y en un futuro podría ser evaluado en organizaciones venezolanas, lo que permitiría al país posicionarse en el mercado competitivo del software. Además, el desarrollo de esta propuesta está inserta en una línea de investigación que está llevando adelante el Laboratorio de Investigación en Sistemas de Información (LISI), financiada por el Fondo Nacional de Ciencia, Tecnología e Investigación (FONACIT)(G-2005000165).

A continuación, se presenta el diagrama de Ishikawa para facilitar el análisis de la justificación presentada.

Consideraciones Éticas

Association Of Computer Machinery (Acm) Code Of Ethics, El Código De Ética Y Práctica Profesional De Ingeniería Del Software Y El "Ieee-Cs/Acm Joint Task Force On Software Engineering Ethics And Professional Practices

Los ingenieros de software debieran obligarse a hacer del análisis, especificación, diseño, desarrollo, pruebas y mantenimiento del software una profesión respetada y beneficiosa. En concordancia con la obligación con el bienestar, salud y seguridad de la sociedad y la aplicación de este proyecto,

los ingenieros del software debieran adherirse a los siguientes principios (<http://www.acm.org/serving/se/code.htm>, recuperado en Julio 2005):

- *Sociedad*: Los ingenieros de software actuarán de manera coherente con el interés social.

- *Cliente y Empresario*: los ingenieros de software actuarán de manera que produzca el mejor resultado para cliente y empresario, y de manera coherente con el interés social.

- *Producto*: garantizarán que sus productos y las modificaciones correspondientes cumplen los mayores estándares profesionales posibles.

- *Valoración*: mantendrán la integridad e independencia en sus valoraciones profesionales.

- *Personal*: los ingenieros del software participarán en el aprendizaje continuo referente a la práctica de su profesión y promoverán un enfoque ético en la práctica de la profesión.

Código Deontológico de la Asociación de Doctores, Licenciados e ingenieros en informática (ali)

La Asociación de Doctores, Licenciados e Ingenieros en Informática (ALI) es la asociación española más importante (junto con la ATI) en lo que se refiere al mundo profesional de los Informáticos y posee un código deontológico propio. Para la aplicación de esta investigación se referencian los siguientes principios (http://www.coiie.org/pdf/codigo_deontologico.pdf, recuperado en Julio 2005):

- Trabaja siempre de forma honesta y leal, y nunca participará de forma consciente en actividades ilegales o impropias de su labor profesional. Respetará el cumplimiento de todas las normas y leyes, actuales y futuras, que afecten al ejercicio de nuestra profesión y las actividades que de ellas se deriven.

- Efectuará su trabajo de forma objetiva e independiente, evitando ejercer actividades que puedan afectar su independencia de forma supuesta o real, especialmente cuando de su trabajo dependan decisiones de otros.

- Mantendrá la confidencialidad de la información a la que tenga acceso por razón de su cargo o desempeño profesional, y no podrá utilizarla en beneficio propio o de terceras personas.

- Respetará en todo momento la propiedad intelectual de terceros, garantizando la integridad de los productos y servicios bajo su responsabilidad.

- Procurará obtener y documentar de forma suficiente los estudios o trabajos en los que base sus conclusiones y recomendaciones.

- Garantizará que los sistemas de información a su cargo cumplan las normas, procedimientos y controles que aseguren su correcto funcionamiento.

- Mantendrá su nivel de competencia profesional con la participación en las actividades de desarrollo adecuadas que dedicará siempre para obtener la máxima calidad de su trabajo.

- Trabajaré desde su ejercicio para que el avance científico y profesional redunde en el progreso hacia una sociedad más justa y digna.

Después de haber concluido con el marco referencial, se hace necesario integrar la teoría con la investigación a través del marco teórico a fin de precisar y organizar los elementos contenidos en la descripción del problema.

CAPITULO II

MARCO TEÓRICO

El objetivo de este capítulo es describir los conceptos que servirán de base para realizar la propuesta del modelo de calidad. Se presentarán los antecedentes de la investigación, representados por el análisis de los modelos de evaluación de calidad existentes, como lo es el Modelo de Decisión para soportar la Selección de herramientas CASE, propuesto por Rojas et., al (1999); los Indicadores definidos por Luca et., al (2001) para la selección de herramientas de la Calidad del Software y el Modelo Sistemico de Calidad MOSCA por Pérez et., al (2005); con la finalidad de aprovechar los conocimientos teóricos ofrecidos por las mismas y obtener conclusiones existentes en torno a la presente investigación.

Modelo de Decisión para soportar la Selección de herramientas CASE

Este modelo de decisión propicia la eficiencia y la efectividad al seleccionar una herramienta CASE; dada la diversidad de factores que afectan la adopción de una herramienta CASE, lo complejas que son por la cantidad de componentes que pueden ofrecer, es muy importante para las organizaciones desde el punto de vista estratégico la selección oportuna y acertada de dichas herramientas. Para ello, Rojas et.,al, (1999), propone una serie de indicadores con sus respectivas variables de medición, clasificadas según su alcance, los cuales por medio de un modelo de decisión brindan una forma cuantitativa de comparar las diferentes herramientas CASE (Rojas et.,al, 1999).

A continuación se presenta un la Tabla N° 1 donde se visualiza cada uno de los factores, categorías, criterios, y subcriterios propuestos para este

modelo (Rojas et.,al, 1999).

Tabla N° 1
Tabla Semántica de Indicadores

Factores	Categoría	Criterio	Subcriterio
Tecnológicos	Internos	Alcance	<ul style="list-style-type: none"> • Fases del Ciclo de Vida. • Metodologías Soportadas. • Número de Usuarios. • Componentes. • Control de Proyecto. • Plataforma de Software Requerido. • Manejador de base de datos.
		Diseño	<ul style="list-style-type: none"> • Facilidad de uso. • Funcionalidad • Curva de Aprendizaje. • Ayuda en Línea. • Flexibilidad. • Tipo de Integración.
	Externos	Soporte	<ul style="list-style-type: none"> • Soporte técnico. • Entrenamiento. • Material de apoyo.
		Solidez	<ul style="list-style-type: none"> • Prestigio del desarrollador. • Costos.

Fuente: Adaptado de Pérez et al. (1999)

En base al modelo presentado, para efectos de esta investigación serán tomados en cuenta los cuatro indicadores: alcance, diseño, pertenecientes a la categoría Internos y soporte y solidez, pertenecientes

a la categoría Externos, de forma tal de obtener la primera versión de los enfoques bajo los cuales estará definido el modelo que especificará la calidad de herramientas que soportan la Gestión de Calidad del Software.

Indicadores para la selección de herramientas de calidad del Software

Luca et., al (2001), asegura que no existen indicadores que se utilizan para apoyar el desarrollo de software en organizaciones y su selección; razón por la cual ellos proponen un conjunto de indicadores que soporten la selección de herramientas del software de gestión de la calidad. Con estos indicadores, cualquier organización puede trazar un plan de aseguramiento de calidad y soportar el proceso de selección estas herramientas.

El conjunto de indicadores propuestos se ha clasificado en tecnológicos y organizacionales. Estos Indicadores apoyan al proceso de selección de una herramienta de gestión de calidad. Los indicadores tecnológicos se refieren directamente a la herramienta: diseño, usos y contexto. Los indicadores organizacionales están relacionados con el uso de este tipo de herramientas en las organizaciones (Luca et., al, 2001).

Sobre la base de los indicadores establecidos por (Rojas et.,al, 1999), los indicadores para la selección de herramientas del software que soportan la gestión de la calidad han sido clasificados (así como los tecnológicos y la organizacionales) en internos y externos.

A continuación se presentan las Tablas N°2 y N°3 que definen cada uno de los indicadores establecidos por Luca et., al (2001) para la selección de herramientas de calidad del software.

Tabla N° 2
Indicadores Tecnológicos

Enfoques	Indicadores
Metodología	<ul style="list-style-type: none"> • Apoya la calidad de las metodologías. • Apoya a las metodologías necesarias que requiere la organización. • Tasa de satisfacción.
Fases del ciclo de vida	<ul style="list-style-type: none"> • Número de fases del ciclo de vida que soporta. • Número de fases del ciclo de vida que requieren las organizaciones. • Tasa de Satisfacción.
Funcionalidad	<ul style="list-style-type: none"> • Adaptabilidad. • Interoperabilidad. • Seguridad.
Fiabilidad	<ul style="list-style-type: none"> • Madurez. • Tolerancia a fallas. • Recuperabilidad
Mantenibilidad	<ul style="list-style-type: none"> • Estabilidad
Evaluación y Certificación de Modelos	<ul style="list-style-type: none"> • Promueve el uso de evaluación y certificación de modelos de calidad del software. • Promueve el uso de evaluación y certificación de modelos de calidad del software que requiere la organización. • Tasa de Satisfacción.
Formas Estructurales	<ul style="list-style-type: none"> • Planificación de calidad • Control de calidad • Evaluación de calidad del software. • Procesos de documentación. • Desarrollo de procesos de análisis de calidad. • Análisis de calidad de los productos de software. • Medición del producto y el proceso de desarrollo de software de calidad. • Estimación de costo. • Estimación de recursos.

Tabla N° 2 (Cont.)

Enfoques	Indicadores
	<ul style="list-style-type: none"> • Estimación de defectos. • Análisis de los datos. • Datos de importación/exportación. • Reportes y gráficos.
Ayuda en línea	<ul style="list-style-type: none"> • Facilidad de ayuda de la herramienta. • Facilidad de uso. • Tasa de satisfacción.
Plataforma	<ul style="list-style-type: none"> • Disposición de hardware para operar la herramienta. • Disposición del software para operar la herramienta. • Tasa de Satisfacción.
Licencias	<ul style="list-style-type: none"> • Licencia de servidores y usuarios. • Sistema de comercialización
Costos	<ul style="list-style-type: none"> • Herramientas y costos de transferencias. • Costos de entrenamiento. • Costos de Mantenimiento. • Costos de soporte técnico. • Costos adicionales de software. • Costos adicionales de hardware.
Soporte	<ul style="list-style-type: none"> • Soporte técnico en el país. • Entrenamiento en el país. • Tipo de entrenamiento. • Actualización de versiones. • Manuales. • Base instalada. <p>Tasa de Satisfacción.</p>

Fuente: Adaptado de De Luca et al. (2001)

Tabla N° 3
Indicadores Organizacionales.

Categoría	Enfoque	Indicadores
Interno	Gestión del Proyecto	Aceptación. Mantenimiento. Estandarización. Plan de Calidad
	Desarrollo del personal	Entrenamiento. Aprendizaje. Habilidad
Externo	Imagen Institucional	Visión
	Relación Interinstitucional	Impacto

Fuente: Adaptado de De Luca et al. (2001)

Los aportes ofrecidos por De Luca et., al (2001), en cuanto a los indicadores propuestos para la selección de herramientas de calidad del software que serán tomados en cuenta para proponer el modelo que persigue el objetivo general de esta investigación son los siguientes:

1. De los indicadores tecnológicos, se tomarán los pertenecientes a los siguientes enfoques: a) Mantenibilidad; b) Fiabilidad y; c) Costos
2. De los indicadores organizacionales, se tomarán los pertenecientes a los siguientes enfoques: Gestión del Proyecto.

Con la selección de estos indicadores se pretende una orientación funcional en el modelo a proponer en base a algunas de las características más importantes que una herramienta de gestión de calidad del software deben poseer.

Modelo Sistémico de Calidad (MOSCA)

El Modelo Sistémico de Calidad (MOSCA) estima la calidad del proceso de desarrollo como el producto final, tomando en cuenta todas las variables

de calidad que están presentes en una organización desarrolladora de sistemas de software bajo el enfoque de Callaos y Callaos (Martínez, 2001).

Según Martínez (2001), el modelo trabaja sobre una plataforma que integra el modelo de calidad del producto del software y el modelo de calidad del desarrollo del software con un enfoque sistémico. En el Apéndice I, se ofrece una descripción detallada del modelo MOSCA.

MOSCA consta en su primera versión de cuatro (4) niveles, los cuales están indicados en la Figura 3 y se detallan a continuación:

- **Nivel 0: Dimensiones.** Las cuatro dimensiones propuestas por MOSCA son: Aspectos Internos y Contextuales del Proceso, Aspectos Internos y Contextuales del Producto.

- **Nivel 1: Categorías.** Este nivel contempla once (11) categorías, de las cuales seis (6) pertenecen al Producto y cinco (5) al Proceso de desarrollo.

- **Nivel 2: Características.** Cada Categoría tiene asociado un conjunto de Características, las cuales definen las áreas claves que se deben satisfacer para lograr, asegurar y para responder a las necesidades de las empresas de manufactura de contar con un instrumento que permita conocer la situación general de una empresa respecto a las “buenas prácticas de gestión” en los sistemas productivos, entendiendo que los resultados de una alta o baja calidad y productividad, así como las posibilidades de mejoras futuras, dependerá directamente de ellas.

- **Nivel 3: Métricas.** Este nivel corresponde a las métricas que permiten medir el cumplimiento de cada Característica de calidad tanto del proceso como del producto de software. Existe un total de 587 métricas para realizar la medición del software y/o el proceso de su desarrollo.

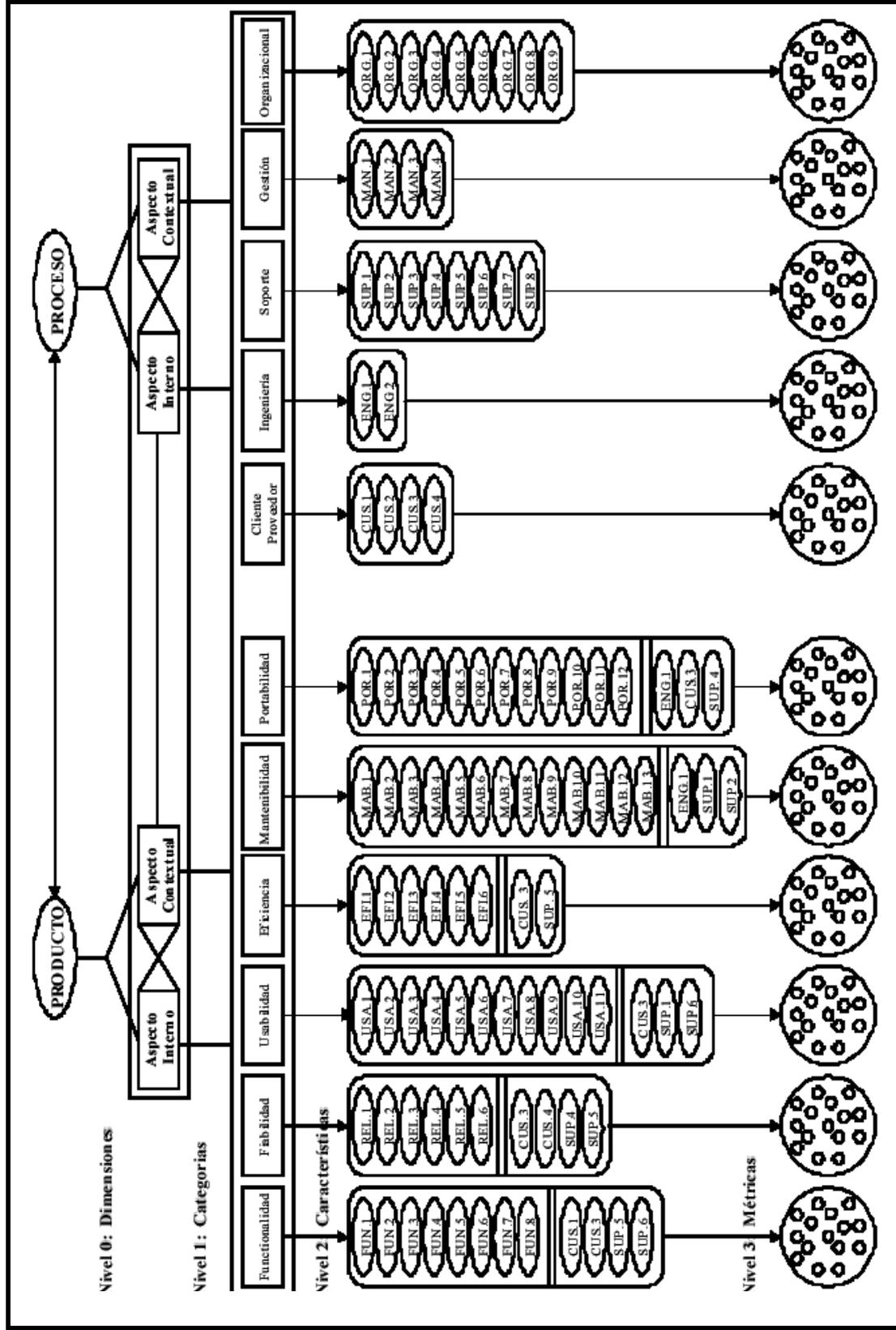


Figura 3. Modelo de Calidad Sistemática del Software (MOSCA)
Fuente: Pérez Et. Al. 2005

Este modelo no sólo arroja un valor de calidad asociado a la empresa evaluada sino que también, especifica los procesos que se deben mejorar en la misma y las características que no son satisfechas por el producto desarrollado, MOSCA tiene la particularidad de que es de origen nacional y de que la calidad sistémica ha sido evaluada ya en organizaciones venezolanas. Además, el modelo no sólo evalúa el nivel de calidad sino que, además, audita implícitamente a la empresa evaluada. MOSCA no se aplica personalmente a la empresa, por lo que el nivel de calidad se estima en torno al grado de sinceridad de los empleados encuestados, cuyo único objetivo es mejorar la calidad de la organización (Martínez, 2001).

MOSCA ha sido evaluado en diferentes organizaciones y las empresas evaluadas mostraron una alta aceptación cuando se aplicó el modelo en cada una de ellas y en las acciones que debían emprender para mejorar su Calidad Sistémica (Pérez, 2005).

En este sentido, para lograr proponer un modelo que ayude a seleccionar las herramientas de software, específicamente, las herramientas que soportan los procesos de la Gestión de Calidad, se tomará del Modelo MOSCA algunas categorías de la perspectiva producto con sus características y métricas, respectivamente. Las categorías seleccionadas se pueden observar en la Tabla N° 4.

Esta selección estuvo basada principalmente en el algoritmo sobre el cual se basa el modelo MOSCA, en el que se exige que además de seleccionar la categoría de Funcionalidad se deben seleccionar dos categorías adicionales. Se decidió seleccionar las categorías de Usabilidad y Fiabilidad como adicionales ya que esta primera versión de la propuesta del modelo está enfocada a medir la calidad de las herramientas en función de los resultados esperados por los usuarios que hacen uso de las mismas, la capacidad de la herramienta de software para mantener un nivel de rendimiento y la capacidad de la herramienta para ser entendida fácilmente por el usuario

Tabla N° 4
Categorías seleccionadas del Modelo MOSCA

PRODUCTO	
Categoría	Características
Funcionalidad (FUN)	FUN.1 Ajuste a los propósitos
	FUN.2 Precisión
	FUN.3 Interoperabilidad
Usabilidad (USA)	USA.1 Facilidad de comprensión
	USA.2 Capacidad de aprendizaje
	USA.3 Interfaz Grafica
Fiabilidad (FIA)	FIA.1 Tolerancia a fallas
	FIA.3 Recuperación

Fuente: Elaboración Propia, 2008

Una vez concluido con la presentación de los antecedentes se procederá a presentar las bases teóricas sobre las cuales se sustenta la investigación. Debido a que la propuesta del modelo se orientará al logro de productos de software de calidad, es importante conocer los conceptos relacionados con Calidad de Software. Posteriormente, se presentan los conceptos asociados a la gestión de calidad así como la importancia de las herramientas para la gestión de calidad.

Calidad del Software

La Calidad debe estar presente en cualquier producto o servicio. Por esta razón las organizaciones han adoptado estándares y han mejorado sus procesos para lograr niveles de aceptación en el mercado internacional (Rojas et ad., 2001).

Las organizaciones que desarrollan Sistemas de Información no son la excepción. Ellas deben ofrecer un alto nivel de calidad en sus productos para mantenerse competitivos en un mercado como el actual. Esta situación conduce a los desarrolladores a buscar soluciones para obtener calidad en sus productos. Ellos deben definir el atributo de Calidad de los mismos y también determinar que el desarrollo de producto de software sea mejorado (Rojas et ad., 2001).

La Calidad del Software es un factor de competitividad dentro de las compañías desarrolladoras de sistemas, ya que éstas tienen que satisfacer las necesidades de sus clientes para así poder mantenerse vigentes en el mercado, y convertirse en líderes dentro del mismo.

Una vez revisada la bibliografía, se encontró interesante plantear los enfoques acerca de la Calidad de Software presentados por Pressman y Callaos.

Pressman (2006), define la *Calidad del Software* como la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente.

Esta definición sirve para enfatizar los siguientes aspectos (Pressman, 2006):

- Los requerimientos del software son la base de las medidas de la calidad. La falta de concordancia con los requerimientos es una falta de calidad.
- Los estándares definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la Ingeniería de Software. Si no se siguen esos criterios, casi siempre ocurrirá una deficiencia en la calidad.
- Existe un conjunto de requisitos implícitos que no se mencionan frecuentemente. Si el software se ajusta a sus requisitos explícitos pero no alcanza los implícitos, la Calidad del Software queda en entredicho.

Pressman, dentro de su definición de Calidad del Software destaca la importancia de definir criterios para lograr desarrollos exitosos, estos criterios bien se podrían traducir en el uso de herramientas que orienten en el logro de productos de software de calidad.

No es suficiente resaltar que la Calidad del Software es importante, se debe crear un conjunto de actividades que ayuden a asegurar que todo producto de Ingeniería del Software presenta alta calidad y asegurar la misma en cada una de las fases de un proyecto del software involucrando a todos los miembros del equipo. Si el equipo trabaja en función de lograr calidad en todas las actividades de Ingeniería del Software, se logra reducir la reelaboración que se debe hacer disminuyendo el costo, logrando así disminuir el tiempo de llegada del producto al mercado (Pressman, 2006).

Además de esto, otra forma de lograr la Calidad de Software es a través de la combinación de procesos y del producto, lo cual se presenta como la Matriz de Calidad Global Sistémica propuesta por Callaos y Callaos (1996).

Callaos y Callaos (1996) proponen un concepto de calidad de los sistemas de software en el cual están involucrados tanto las características internas como el contexto organizacional, lo que genera un enfoque sistémico del concepto de calidad del software.

La definición de calidad sistémica en el desarrollo de software, contempla la Matriz Global Sistémica considerando las dimensiones del cliente y del usuario. Esta matriz se muestra en la Figura 4 y consta de cuatro (4) tipos de calidades: 1) Producto-eficiencia; 2) Producto-efectividad; 3) Proceso-eficiencia y; 4) Proceso-efectividad.

Esta división se justifica puesto que un proyecto incluye tanto la eficiencia como la efectividad, y además 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).



Figura 4. Matriz de Calidad Global Sistémica

Fuente: (Callaos y Callaos, 1996)

Nótese, que estos autores no se contradicen, por el contrario tienen semejanzas entre sus enfoques. Ambos toman en cuenta tanto el producto como el proceso de desarrollo del software, partiendo de la dimensión del usuario y del cliente. Pressman se centra en señalar que la calidad de software es posible obtenerla si se mantiene una vigilancia permanente durante todo el proceso de desarrollo, y el enfoque de Callaos y Callaos propone que la calidad del diseño se relaciona con la calidad del proceso, que la calidad es sistémica; es decir, que debe haber un balance entre todas sus dimensiones.

Una vez realizada la descripción de estos enfoques de Calidad del Software; se puede afirmar que los Sistemas de Software pueden ser considerados como productos y servicios. Esto ha motivado que a escala mundial se está generando un conjunto de modelos para estimar su calidad, los cuales responden a las necesidades de garantizar productos de calidad en una competencia abierta y mundial, sobre la base de cada tipo de Sistema de Software (Pérez et al., 2005).

Modelos de Calidad del Producto

Es importante señalar que para determinar la actual importancia de la calidad en un mundo en vías de la globalización, es necesario mencionar que los modelos de la calidad tanto del producto como del proceso de Sistemas de Software permiten controlar, mejorar y asegurar la calidad (Álvarez, 2000). Así como también, son un aporte para su certificación.

Actualmente, existen modelos de la calidad de producto de software que ofrecen mediciones internacionalmente aceptadas del producto de desarrollo. En el Apéndice II, se ofrece una descripción de cada uno los modelos de calidad del producto.

Se puede decir que estos modelos son útiles, pero sólo contemplan productos, ninguno de ellos está basado en un enfoque de calidad como el presentado por Callaos que combine de manera sistémica producto y proceso, es decir, ninguno de estos modelos presentados es sistémico. Sin embargo, es importante tomar en cuenta los aportes proporcionados por cada uno de ellos para obtener un sustento teórico que permita facilitar el diseño de la propuesta del modelo que se persigue como objetivo de la presente investigación.

Gestión de Calidad del Software

Es posible definir un *sistema de garantía de la calidad* como la estructura organizacional, responsabilidades, procedimientos, procesos y recursos para implementar la gestión de la calidad. Los sistemas de garantía de la calidad fueron creados para ayudar a las organizaciones a garantizar que sus productos y servicios satisfacen las expectativas de los clientes al cumplir sus especificaciones (Pressman, 2006).

El estándar ISO 9000:2000 describe un sistema de garantía de la calidad en términos genéricos que se aplican a cualquier negocio sin importar

los productos o servicios ofrecidos. El estándar de garantía de calidad que se aplica a la Ingeniería del Software es el ISO 9001:2000, el cual contiene 20 requisitos que deben estar presentes para un sistema eficiente de garantía de la calidad. Puesto que el estándar ISO 9001:2000 es aplicable a todas las disciplinas de ingeniería, se ha desarrollado un conjunto especial de directrices ISO (ISO 9000-3) que ayudan a interpretar el estándar para emplearlo en el proceso del software. Una de las cláusulas propuestas por el estándar ISO 9000-3 es la planificación de la Calidad que ayuda a establecer los objetivos de la misma. Algunas ideas básicas propuestas por el estándar ISO 900-3 son las siguientes:

- El control de calidad debe ser aplicado a todas las fases de la producción de software, incluido el mantenimiento y tareas posteriores a su implantación.
- Debe existir una estricta colaboración entre la organización que adquiere el software y el proveedor del mismo.
- El proveedor del software debe definir su sistema de calidad y asegurarse que toda la organización ponga en práctica este sistema.

Se puede decir que el ISO 9000-3 proporciona una guía útil que sirve para detectar y corregir problemas y/o errores en los productos de software, consiguiendo una mejora en la calidad de los mismos.

Una vez revisada la bibliografía, se encontraron cuatro enfoques acerca de los procesos subyacentes en la Gestión de la Calidad del Software, los presentados por Sommerville, Pressman, la guía de ingeniería del software (SWEBOK) y Unhelkar.

Sommerville (2002) plantea que la *administración de la calidad del software* se estructura en tres actividades principales:

1. *Aseguramiento de la Calidad*: Las actividades de aseguramiento de la calidad (QA) comprenden definir un marco de trabajo para lograr la calidad del software. Los procesos de QA comprenden definir o seleccionar estándares aplicables al proceso de desarrollo del software o a los productos

del software. Estos estándares pueden estar incrustados en procedimiento o procesos aplicables durante el desarrollo. Los procesos se pueden apoyar en herramientas que capturan el conocimiento de los estándares de calidad. Existen dos tipos de estándares que se establecen como parte del proceso de aseguramiento de la calidad: estándares del producto y estándares del proceso, estos poseen una relación muy cercana. Los estándares de producto se aplican a las salidas del proceso del software y, en muchos casos, los estándares del proceso incluyen actividades específicas del proceso que aseguran que se siguen los estándares del producto.

2. *Planeación de la Calidad:* La planeación de la calidad inicia en las primeras etapas del proceso del software. Un plan de calidad define la calidad del producto deseado. Define cómo valorar esta calidad. Por lo tanto, define lo que significaría software de “alta calidad”. El resultado de la planeación de la calidad es un plan de calidad del proyecto. El plan de calidad selecciona aquellos estándares internacionales apropiados para un producto particular y un proceso de desarrollo. Una parte importante de la planeación de la calidad es seleccionar los atributos de la calidad importantes y planear cómo alcanzarlos. El plan de calidad define los atributos de la calidad más importantes para el producto a desarrollar.

3. *Control de la Calidad:* La definición y promulgación de los procesos que aseguran que los procedimientos y estándares para la calidad del proyecto son seguidos por el equipo de desarrollo de software. El control de la calidad implica vigilar el proceso de desarrollo del software para asegurar que se sigan los procedimientos de aseguramiento y estándares de calidad. Los productos resultantes de un proceso de software se comprueban contra los estándares definidos del proyecto en el proceso del control de calidad. El proceso de control de calidad tiene su propio conjunto de procedimientos e informes a utilizar durante el desarrollo del software.

Estos procedimientos son directos y fácilmente comprensibles por los ingenieros que desarrollan software. Somerville (2002) plantea dos enfoques

complementarios para el control de la calidad: a) Revisiones de la calidad en las que el software, su documentación y los procesos utilizados para producir software son revisados por un grupo de personas. Son responsables de comprobar que se han seguido estancares del proyecto y que el software y los documentos están acordes a estos estándares. Toman las desviaciones de los estándares y las ponen en consideración de la administración del proyecto y; b) Validación automática del software en la que el software y los documentos producidos se procesan por algún programa y se comparan con los estándares que aplican a ese proyecto de desarrollo en particular. Esta valoración automática comprende una medida cuantitativa de algunos atributos del software. La medición del software se refiere a derivar un valor numérico para algún atributo de un producto de software o un proceso de software. Comparando estos valores entre ellos y con los estándares aplicados en la organización, es posible sacar conclusiones de la calidad del software o de los procesos del software.

Por otra parte, Pressman (2006) plantea que la gestión de calidad del software abarca:

1. Un proceso de garantía de la calidad del software, el cual consiste en un conjunto de funciones de auditoría e información que evalúan la efectividad y que tan completas son las actividades de control de calidad. La meta del aseguramiento de calidad es brindarle al gestor los datos necesarios para que este informado acerca de la calidad del producto, y por consiguiente que comprenda y confíe en que la calidad del producto esta satisfaciendo sus metas. Desde luego, si los datos que ofrece el aseguramiento de la calidad identifican problemas, es responsabilidad del gestor abordarlos y aplicar los recursos necesarios para resolver los conflictos de calidad.

2. Tareas específicas de aseguramiento y control de calidad, que incluyen revisiones técnicas formales y una estrategia de pruebas de varios niveles. Pressman (2006) plantea que el control de la calidad involucra una

serie de inspecciones, revisiones y pruebas empleadas a lo largo del proceso del software para garantizar que cada producto de trabajo satisfaga los requisitos que se le han asignado. Las revisiones sirven como filtros a través de todas las actividades de Ingeniería del Software, que eliminan los errores mientras son relativamente poco costoso de encontrar y corregir.

3. Control de todos los productos de trabajo del software y los cambios que generan.

4. Un procedimiento para garantizar la concordancia con los estándares de desarrollo del software.

5. Mecanismos de medición e informe: Ayuda a mejorar la calidad del producto y el proceso del software en sí mismo. Los modelos de fiabilidad del software extienden medidas, lo que permite recopilar datos de defecto para extrapolarlos en las tasas de falla proyectadas y las predicciones de fiabilidad.

Esta definición ayuda a destacar los siguientes aspectos (Pressman 2006):

- Creación de un conjunto de actividades que ayudarán a asegurar que todo producto de trabajo de ingeniería de software presentarán alta calidad.

- Realización de actividades de control y aseguramiento de la calidad que sirven como filtros a través de todas las actividades de ingeniería del software, descubriendo errores y defectos que luego puedan eliminarse mientras son relativamente poco costosos de encontrar y corregir.

- Incluir revisiones formales que incluyen inspecciones planificadas y revisiones cíclicas.

- Ayuda a mejorar la calidad del producto y el proceso de software en sí mismo mediante el aseguramiento de la calidad estadística, ya que a través de ella es posible informar sobre los defectos del software, determinar la causa subyacente de cada uno y corregir el problema que han provocado.

Una vez presentados los enfoques propuestos por Sommerville y Pressman, respectivamente, a continuación se presentará la perspectiva propuesta por la guía de ingeniería del software (SWEBOK) en relación a la gestión de la calidad del software.

El SWEBOK (2004) señala que existen tres tópicos relacionados a la Calidad del Software, los cuales son: a) Fundamentos de la Calidad del Software; b) Procesos de gestión de la Calidad del Software y; c) Consideraciones prácticas.

Cada uno de estos tópicos conlleva inmerso un conjunto de actividades que permiten garantizar la calidad de un producto resultante de un trabajo de Ingeniería del Software. En la Figura 5 se detalla el enfoque propuesto por el SWEBOK.

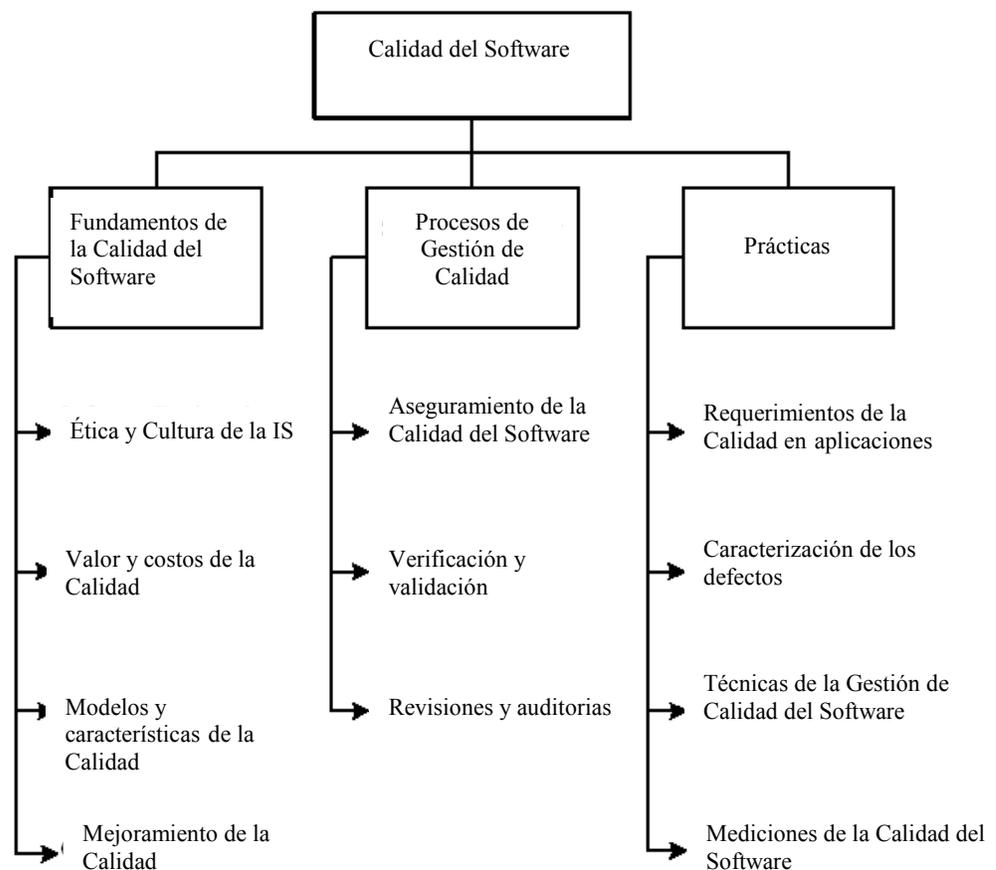


Figura 5. Análisis de los Tópicos para la Calidad del Software
Fuente: (SWEBOK, 2004)

En este sentido, el SWEBOK plantea que la gestión de la calidad del software se aplica en todas las perspectivas de los procesos del software,

productos y recursos y consiste en la ejecución de actividades, en las cuales algunas ayudan a encontrar errores y otras pueden indicar las posibles fallas. Además, presenta procesos subyacentes, los cuales deben tener en cuenta como los productos de software satisfacen al cliente y a todos los involucrados en los requisitos planteados, proporcionar valor a los clientes y proporcionar la calidad deseada. Además, fortalecen la calidad y encuentran posibles fallas, indicando como los proyectos de software están siendo utilizados en práctica y como los productos intermedios y finales se encuentran acordes con las exigencias especificadas. Los procesos inmersos en gestión de la calidad del software planteados por SWEBOK (2004) son los siguientes:

1. Aseguramiento de calidad del software: Garantiza que el producto del software y sus procesos se ajustan a los requisitos especificados por la planificación, además se realizan una serie de actividades que proporcionan la confianza adecuada para construir un software de calidad. Garantiza que los procesos previstos son los apropiados a la organización y que son llevados a cabo de acuerdo a la planificación.

2. Proceso de verificación y validación: El SWEBOK plantea que tanto la verificación como la validación es tratado como un único término y lo relaciona a un proceso disciplinado que evalúa productos de software durante todo el ciclo de vida de su desarrollo. Se esfuerza por asegurarse de que la calidad se basa en el software y el software satisface las necesidades de los usuarios.

3. Proceso de revisión y auditoria: El SWEBOK plantea que tanto la revisión como la auditoria es tratado como un único término y presenta cinco tipo de revisiones o auditorias, las cuales son: Manejo de comentarios, Revisiones Técnicas, Inspecciones, Walk-throughs y Auditorias.

Por último, Unhelkar (2003) plantea que los procesos de la Calidad del Software son los siguientes: gestión de la calidad, garantía de calidad y control de calidad.

A continuación se describirá el detalle de cada uno de los procesos planteados por Unhekar (2003):

1. Gestión De La Calidad: El proceso de gestión de la calidad se esfuerza por reunir a los interlocutores sociales y los aspectos metodológicos del proyecto con especial atención a la calidad. Algunas de las responsabilidades de este proceso son la planificación para el proyecto manteniendo la calidad en mente, la identificación de los estándares, la fijación de las expectativas de los usuarios, y, lo más importante, conseguir las personas adecuadas junto a la calidad del trabajo. Junto con la gestión de proyectos, se ocupa de enfoque global de la calidad de los proyectos. La creación de un ambiente de calidad con un buen factor ambiental, la creación y gestión de equipos, y el manejo general de la sociología en el marco del proyecto son algunas de las responsabilidades de la gestión de la calidad. Los entregables que Unhekar (2003) identifica para este proceso son:

- El plan de calidad: se encarga del enfoque global de la calidad dentro del proyecto.
- El plan de pruebas: se ocupa de la gestión de los aspectos de las pruebas dentro del proyecto.
- La calidad de proceso del software: describe el proceso del software que puede ser personalizado y seguido por los miembros del proyecto.
- Los estándares: existen en todos los niveles dentro del proyecto, incluido el diseño, documentación, codificación y pruebas.

figura 6

2. Aseguramiento de la calidad: Sigue el proceso de gestión de la calidad y se compromete el esfuerzo real de asegurar la calidad de los modelos y procesos en el proyecto. Como en la mayoría de los procesos, el proceso de aseguramiento de la calidad no es un proceso independiente, sino que más bien está relacionada con el proceso que producen los modelos. Su función es asegurar que los modelos y procesos de desarrollo de software son seguidos de manera que resulte en buen software. Siguiendo un proceso de calidad, se espera encontrar un menor número de errores y fallos y más satisfactoria las pruebas de aceptación, producto del hecho de que el desarrollo de software ha tenido lugar en respuesta a un problema bien definido. Proporciona las plantillas de producto terminado y verificados en función del cumplimiento de los estándares del proyecto. Esto se traduce en la calidad general de los modelos y procesos. Los entregables que Unhekar (2003) identifica para este proceso son:

- La Calidad del software publicada (QSP): proporciona la base para la publicación de la calidad de los aspectos del proyecto. Esto se compone de las plantillas, productos, actividades y tareas que se centran en la calidad.
- MOPS: el modelo de problema de espacio está sujeto a los rigores de actividades y tareas de la garantía de calidad.
- MOSS: El modelo de solución de espacios está sujeto a los rigores de actividades y tareas de la garantía de calidad.

3. Control de Calidad: Uno de los aspectos cruciales de aseguramiento de la calidad es el control de calidad. En el caso del software, control de calidad se refiere a si el producto de software funciona correctamente, de acuerdo a las especificaciones, y en combinación con otros productos de software y bases de datos. Los entregables que Unhekar (2003) identifica para este proceso son:

- El plan de pruebas: contiene el aspecto de organización de las pruebas. Esto incluye detalles de las personas y las listas de verificación, así como lo que se espera del proceso de pruebas.

- figura 7

- Diseño de pruebas: ofrece un diseño táctico de las pruebas. Prueba de diseños se pueden organizar en torno a los subsistemas o paquetes.
- El ambiente de pruebas: es el medio ambiente físico que necesita ser creado antes de que la prueba puede comenzar. Es también el entorno de software que debe ser creado para la prueba.
- Los datos de prueba: garantizan la correcta ejecución de las pruebas.
- Los resultados están documentados, recogidos y utilizados con fines de información. También pueden utilizarse para prever las áreas del sistema que necesita más correcciones, o incluso reescrituras.

Dado que estos autores enfocan el concepto de calidad del software desde dos perspectivas distintas, A continuación, la Tabla N° 5 presenta un resumen de las características más importantes de las tres definiciones de gestión de calidad del software estudiadas en este capítulo.

Tabla N° 5.
Tabla Resumen de la Definición de Gestión de Calidad del Software propuesta por Somerville, Pressman , SWEBOK y Unhelkar.

Enfoque	Procesos Subyacentes en la Gestión de Calidad del Software
SOMMERVILLE	Se estructura en tres actividades principales: <ul style="list-style-type: none"> • Planeación de la Calidad: La cual se caracteriza por la realización de un plan de calidad. • Aseguramiento de la Calidad: Para lo cual se definen o seleccionan estándares aplicables al proceso de desarrollo. • Control de la Calidad: En la cual se utilizan como técnicas el uso de métricas y revisiones.
PRESSMAN	La gestión de la Calidad abarca: <ul style="list-style-type: none"> • Un proceso de garantía de la calidad del software. • Tareas específicas de aseguramiento y control de la calidad (revisiones técnicas formales). • Prácticas efectivas de ingeniería del software (métodos y herramientas).

figura 8

Tabla N° 5 (Cont.)

PRESSMAN	<ul style="list-style-type: none"> • Un procedimiento para garantizar la concordancia con los estándares de desarrollo del software. • Mecanismos de medición e informes.
SWEBOK	<p>Los procesos manejados para la gestión de calidad del software son:</p> <ul style="list-style-type: none"> • Aseguramiento de la Calidad. • Proceso de verificación y validación. • Proceso de revisión y auditoria.
UNHEKAR	<p>Los procesos subyacentes en los procesos de la Calidad del Software son:</p> <ul style="list-style-type: none"> • Gestión de la Calidad. • Aseguramiento de la Calidad. • Control de Calidad • Proceso de revisión y auditoria.

Fuente: Elaboración Propia, 2008

A pesar de que estos autores enfocan el concepto de gestión de calidad del software desde perspectivas distintas, todos estos enfoques proporcionan suficientes argumentos para identificar los diferentes procesos que deben llevarse a cabo en la Gestión de la Calidad del Software. Sommerville, define tres procesos que están inmersos en la Gestión de la Calidad del Software, cada uno con actividades definidas para alcanzar los objetivos particulares de cada uno de ellos que garantizan la calidad total del proceso de desarrollo del software. Por otra parte, Pressman (2006) lo relaciona con la realización de actividades de control y aseguramiento de la calidad que sirven como filtros a través de todas las actividades de ingeniería del software, descubriendo errores y defectos que luego puedan eliminarse mientras son relativamente poco costosos de encontrar y corregir. Plantea que el aseguramiento de la calidad abarca procedimientos para la aplicación eficaz de métodos y herramientas, revisiones técnicas formales, estrategias y

técnicas de pruebas, procedimientos para el control de cambio, procedimientos para garantizar la concordancia con los estándares y mecanismos de medición y reporte.

Nótese, que estos autores no se contradicen, por el contrario tienen semejanzas entre sus enfoques. Pressman plantea actividades que se encuentran incluidos en cada uno de los procesos de la Gestión de Calidad del Software planteadas por Sommerville. Las revisiones técnicas formales y los procedimientos para garantizar la concordancia con los estándares planteadas por Pressman en el Aseguramiento de la Calidad, Sommerville lo asocia con los procesos de Control de Calidad y Aseguramiento de la Calidad, respectivamente.

Por otra parte, SWEBOK plantea que la Gestión de la Calidad del Software se aplica en todas las perspectivas de los procesos del software, productos y recursos y consiste en la ejecución de muchas actividades, en las cuales algunas ayudan a encontrar errores y otras pueden indicar las posibles fallas, entre algunas de esas actividades están: el Aseguramiento de la Calidad del Software, procesos de verificación y validación y procesos de auditoría. Este planteamiento proporcionado por SWEBOK difiere del enfoque presentado por Sommerville; ya que el Aseguramiento de la Calidad para Sommerville es un proceso y no una actividad. Por otra parte, los procesos de verificación y validación junto con las auditorías son actividades; sin embargo, Sommerville las identifica inmersas en los procesos de Control de Calidad y Aseguramiento de Calidad planteado por Sommerville.

Por último, el enfoque propuesto por Unhekar está alineado muy de cerca con el planteamiento sugerido por Sommerville. Ambos autores sugieren la ejecución de tres procesos para alcanzar la Calidad del Software y coinciden en la definición de cada uno de ellos, a pesar de cada uno de ellos les otorgue nombres diferentes. Estos enfoques proporcionan suficientes argumentos que justifican la necesidad de cumplir con procesos definidos que permitan lograr el objetivo propuesto, alcanzando de esta

manera los más altos niveles de la Calidad del Software. Por lo tanto, una vez estudiada la bibliografía y los aportes proporcionados por la misma, para efectos de la presente investigación, se tomará en cuenta el aporte sugerido por Sommerville para definir los procesos subyacentes en la Gestión de la Calidad del Software, los cuales son los siguientes: a) Planeación de la Calidad; b) Aseguramiento de la Calidad y; c) Control de la Calidad.

Además, se tomará en cuenta alguna de las actividades sugeridas por el aporte presentado por Unhekar para incorporarlas en cada uno de los procesos de la Gestión de la Calidad del Software mencionados anteriormente. La identificación de actividades por procesos será visualizada en la Tabla N°6.

Tabla N° 6.

Tabla Resumen de las Actividades de cada uno de los Procesos de la Gestión de Calidad del Software.

Procesos de la Gestión de Calidad del Software	Actividades
Planeación de la Calidad	<ul style="list-style-type: none"> • Realización de un plan de calidad: en el cual se definan las expectativas de calidad del usuario y los atributos de calidad. • Definición de estándares de calidad: lenguaje, modelado, base de datos, procesos de desarrollo, proyecto, organizacionales. • Formación del equipo de calidad,. • Definición del ambiente de calidad. • Definición de un plan de pruebas: organización de pruebas y recursos. • Definición de métricas de calidad.
Aseguramiento de la Calidad	<ul style="list-style-type: none"> • Aplicación de listas de chequeo para el cumplimiento de los procesos establecidos. • Aplicación de listas de chequeo para el cumplimiento de los estándares establecidos.

Tabla N° 6 (Cont.)

Procesos de la Gestión de Calidad del Software	Actividades
Aseguramiento de la Calidad	<ul style="list-style-type: none"> • Identificación de modelos de calidad. • Seleccionar las técnicas de calidad. • Revisiones Técnicas. • Realización de inspecciones. • Realización de procesos de validación y verificación. • Realización de estimaciones y métricas de calidad. • Documentación de reporte de errores. • Corrección de entregables. • Corrección de errores. • Auditorias
Control de la Calidad	<ul style="list-style-type: none"> • Verificación y /o seguimiento de la ejecución de pruebas: herramientas, ambiente. • Verificación y /o seguimiento de la corrección de errores. • Verificación y /o seguimiento de la aplicación de listas de chequeo para el cumplimiento de los procesos establecidos. • Verificación y /o seguimiento de la aplicación de listas de chequeo para el cumplimiento de los estándares establecidos. • Verificación y /o seguimiento de las revisiones Técnicas. • Verificación y /o seguimiento de las realización de inspecciones. • Verificación y /o seguimiento de la realización de procesos de validación y verificación. • Verificación y /o seguimiento de la documentación de reporte de errores. • Verificación y /o seguimiento de la corrección de entregables. • Verificación y /o seguimiento de las auditorias

Fuente: Elaboración Propia, 2008

Se debe destacar que la Gestión de Calidad es importante en el desarrollo de cualquier proyecto de Sistema de Software puesto que a través de las mismas se establecen los pasos y procedimientos que se deben seguir para lograr desarrollos exitosos. Cada uno de los procesos incluidos en la Gestión de la Calidad del Software lleva consigo la aplicación de técnicas y el uso de herramientas inmersas. De allí se deriva la necesidad de definir o identificar qué herramienta(s) puede(n) ayudar a evaluar la Gestión de la Calidad del Software. A continuación se presentan los conceptos asociados a herramientas del software.

Herramientas del Software

En los últimos años las empresas desarrolladoras de software han invertido grandes cantidades de dinero en herramientas de software bajo la convicción de que estas herramientas representan una ventaja competitiva para la organización, por el hecho de brindar soporte en casi todas las etapas del proceso de desarrollo de los sistemas de software. Ellas le brindan a las organizaciones capacidades para responder de una manera mejor y más rápida al entorno en el cual se desenvuelven (Rivas et al., 2007).

Para Mendoza (1999), el objetivo de todas las herramientas de software es soportar todo o parte del proceso de desarrollo de software. Para Rojas y Pérez (1995), las organizaciones deben buscar las herramientas que mejor se adapten a sus requerimientos, en función de la eficiencia del proceso de desarrollo y la efectividad esperada de sus resultados.

Las herramientas del software han desempeñado un importante papel en el desarrollo de proyectos de software. Como consecuencia del avance tecnológico éstas han experimentado también continuos cambios. Así como se cuenta en la actualidad con documentación sobre las numerosas herramientas de software disponibles, y con trabajos de investigación que revelan avances en

herramientas particulares, existe también en la literatura evidencia de que los escritos técnicos genéricos sobre herramientas de software son relativamente escasos (Rivas et al., 2004), identificándose la pertinencia de canalizar iniciativas para la formulación de modelos conceptuales, que apoyen la comprensión (y el uso posterior) de estas herramientas.

Actualmente, se considera a las herramientas de software como herramientas basadas en computadoras que asisten el proceso de ciclo de vida de software (IEEE, 2004), consolidadas en la literatura en la forma de Ingeniería de software asistida por computadora (CASE, por sus siglas en inglés). Esto es, software que se utiliza para ayudar a las actividades del proceso de software o software que es utilizado para diseñar y para implementar otro software (Sommerville, 2005). Permiten automatizar acciones bien definidas, reduciendo también la carga cognitiva del ingeniero de software, quien requiere libertad para concentrarse en los aspectos creativos del proceso. Este soporte se traduce en mejoras a la calidad y la productividad en el diseño y desarrollo.

A continuación, la Tabla N° 6 presenta un resumen de las características más importantes de las dos definiciones de herramientas de software estudiadas en este capítulo, las cuales son propuestas por SWEBOK (2004), Pressman (2006) y Sommerville (2005).

Tabla N° 7.

Tabla Resumen de las Definiciones de Herramientas de Software Propuestas por SWEBOK, Pressman y Sommerville.

Enfoque	Características
PRESSMAN	<ul style="list-style-type: none"> • Proporcionan el soporte automatizado o semiautomatizado para el proceso y los métodos. • Las herramientas pueden integrarse para reutilizar la información creada por una de ellas. • En la práctica los límites entre las clasificaciones de las herramientas son difusos.

Tabla N° 7 (Cont.)

Enfoque	Características
PRESSMAN	<ul style="list-style-type: none"> • Las herramientas del software se clasifican según su función específica dentro del proceso de desarrollo. Dentro de esta clasificación se definen dos categorías: herramientas (ayudan a tareas puntuales) y los bancos de trabajo (agrupan herramientas que mantienen algún grado de interacción y soportan a un método que incluye un modelo de proceso) • Plantea una arquitectura general de ingeniería del software como una tecnología multicapas, una perspectiva que la representa en un conjunto de cuatro grandes capas o niveles: enfoque de calidad, proceso, el método y herramientas.
SWEBOK	<ul style="list-style-type: none"> • Son consideradas como herramientas basadas en computadoras que asisten el proceso del ciclo de vida del software. • A menudo, son diseñadas para apoyar los métodos particulares de ingeniería de software, reduciendo las posibles cargas asociadas a la aplicación de los métodos manuales. • Al igual que los métodos, tienen la intención de hacer la ingeniería del software más sistemática y varían en alcance en cada una de las tareas de apoyo durante el ciclo de vida completo. • Existe poca documentación en relación a las herramientas disponibles, debido al cambio acelerado que experimentan, lo cual no permite proporcionar guías concretas acerca de las mismas. • Hacia lo interno, existen un conjunto de áreas de conocimiento que facilitan la comprensión del alcance y las limitaciones de diferentes temas en la ingeniería del software. Estas áreas son: Requerimientos, Diseño, Construcción, Pruebas y Mantenimiento de software, Gestión de la configuración, Gestión de la ingeniería y del proceso de la ingeniería de software, Herramientas y métodos de software, y Calidad de software. De este modo, las herramientas de software y los métodos, integran un área específica de la

Tabla N° 7 (Cont.)

Enfoque	Características
SWEBOK	ingeniería del software, y contribuyen a la producción de software de alta calidad, con bajo costo y en el menor tiempo posible
SOMMERVILLE	<ul style="list-style-type: none">• Es un software que se utiliza para ayudar a las actividades del proceso de software o software que es utilizado para diseñar y para implementar otro software

Fuente: Elaboración Propia, 2008

Una vez presentado el enfoque estos autores, es posible afirmar que las herramientas de software, junto con los métodos, constituyen un área de conocimiento específica y relevante de la ingeniería del software, que es a su vez parte de un conjunto de disciplinas relacionadas de la ingeniería. En este contexto, las herramientas de software contribuyen a la producción de software de alta calidad, a bajo costo y en el menor tiempo posible. Sin embargo, sus beneficios potenciales serán logrados sólo si se utilizan en forma correcta.

El soporte que brindan las herramientas de software al proceso de desarrollo proporciona importantes ventajas para el equipo de trabajo de Ingeniería del Software. Estas mejoras se sintetizan en (Rivas et.al, 2007):

- a) Apoyan a las metodologías y métodos, integrando actividades y propiciando visión de continuidad entre fases metodológicas;
- b) Mejoran la comunicación entre los actores involucrados, facilitándoles compartir su trabajo y desempeñarlo de forma dinámica e iterativa;
- c) Establecen métodos efectivos para almacenar y utilizar los datos, lo que permite organizar y correlacionar componentes, para accederlos a través de un repositorio;
- d) Agregan eficiencia al mantenimiento, ya que los programas son construidos sobre las mismas estructuras y estándares, facilitando la adherencia a la

disciplina de diseño y facilitan también la conversión automática de programas a versiones más recientes de lenguajes de programación; y, e) Automatizan porciones del análisis y diseño, tediosas y propensas a error, con influencia sobre la generación de código, las pruebas y el control. Resalta la consideración de que los beneficios potenciales sólo pueden ser alcanzados si las herramientas de software son utilizadas de forma correcta.

Por otra parte, establecer una clasificación resulta importante para comprender los tipos de herramientas disponibles y su soporte al proceso del software (Kendall et.,al, 2002). En adelante se exponen las principales perspectivas, o formas de clasificación de las herramientas de software, disponibles en la literatura. En la práctica, los límites entre estas clasificaciones son difusos (Pressman, 2006), lo cual contribuye al argumento de que no es fácil ubicar una herramienta utilizando una clasificación particular. Una vez revisada la bibliografía, a continuación se presentan diversas clasificaciones planteadas por diversos autores:

- De acuerdo con las funciones que la herramienta soporta: En esta perspectiva, las herramientas de software se clasifican según su función específica dentro del proceso de desarrollo (Pressman, 2006). En la literatura se identifican hasta 22 tipos de herramientas clasificadas funcionalmente (Pressman, 2006). Estas clasificaciones tienen una intención orientadora, apuntan a mostrar la amplitud de las herramientas y sus aportes a distintas etapas del desarrollo (Sommerville, 2005), más que a establecer taxonomías rigurosas.

- De acuerdo con el actor que es soportado por la herramienta. Esta perspectiva se basa en el sujeto a quien soporta la herramienta, y las define como: a) De Alto nivel, que ayudan principalmente a los analistas y diseñadores, permitiéndoles crear y modificar el sistema (Alter, 2002); y, b) De Bajo nivel, utilizadas por programadores y trabajadores quienes deben implementar los sistemas diseñados (Rivas et.,al, 2007).

- De acuerdo con el proceso. En esta clasificación se definen tres

categorías: a) las Herramientas, que ayudan a tareas puntuales y se utilizan a discreción del ingeniero de software; b) los Bancos de trabajo, que agrupan herramientas que mantienen algún grado de integración y soportan a un método que incluye un modelo del proceso (Pressman, 2006); y c) los Ambientes, que apoyan a los procesos de software, incluyen bancos de trabajo integrados y soportan a los datos, al control y a la integración (Fuggetta, 1993).

- Tópicos de las herramientas, como sub área de conocimiento de la IS. A modo de cierre, resulta pertinente considerar la definición de los tópicos relativos a las herramientas de software (IEEE, 2004). En esta clasificación existe un conjunto de herramientas dentro de cada tópico, cuyo campo de acción se asocia no sólo con las actividades propias de cada uno de estos, sino también se considera el caso de herramientas que prestan soporte con un carácter transversal, como el caso de las herramientas misceláneas.

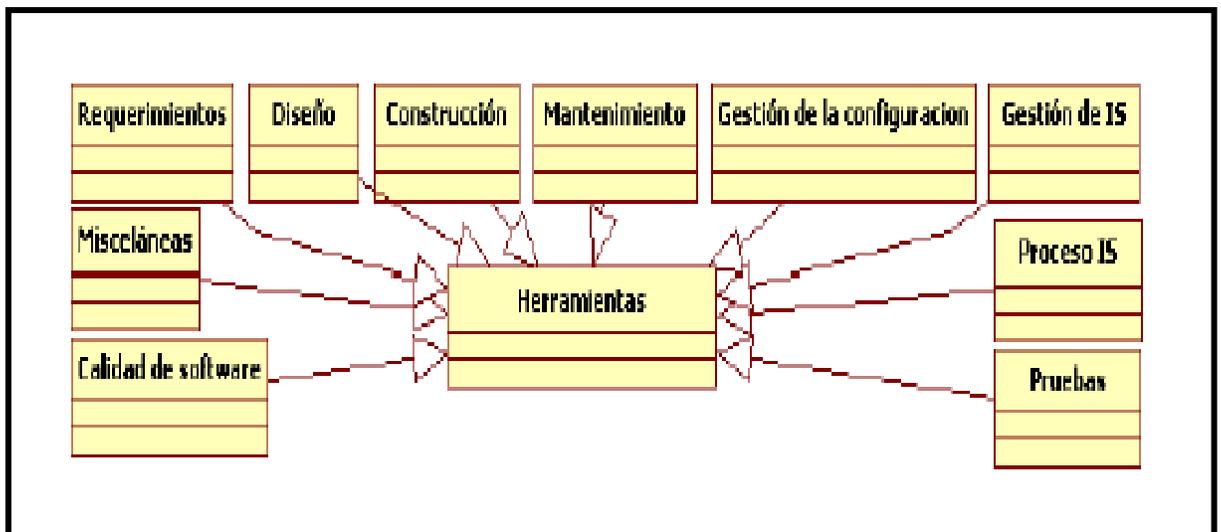


Figura 9. Tópicos de las HDS, como Subárea de Conocimiento de la IS

Fuente Rivas et, al, 2007

Una vez presentada una visión global acerca de las herramientas de software, es posible afirmar que su objetivo principal es auxiliar al equipo de

proyecto para valorar y mejorar la calidad del producto del trabajo de software.

La revisión bibliográfica muestra que estas herramientas, constituyen un área de conocimiento específica y relevante de la Ingeniería del Software, que es a su vez parte de un conjunto de disciplinas relacionadas de la ingeniería. Estas herramientas representan una ventaja competitiva para las organizaciones, por el hecho de brindar soporte en casi todas las etapas del proceso de desarrollo de los sistemas de software. Sin embargo, estas organizaciones se han encontrado con que las empresas desarrolladoras de herramientas no son lo suficientemente precisas a la hora de manifestar las capacidades y alcances de la herramienta que comercializan.

Por otro lado, las organizaciones en la actualidad están convencidas de que deben adquirir y/o producir productos de alta y reconocida Calidad, desarrollar proyectos exitosos, tomando en cuenta una adecuada gestión del desempeño en sus proyectos en virtud de la fuerte competencia que se ha generado en todas las áreas productivas donde los sistemas de software han ocupado un lugar preponderante dado la gran importancia que tienen desde el punto de vista estratégico. La gestión del desempeño de un proyecto completa procesos referentes a la gestión de calidad y gestión del riesgo, y esta investigación pretende brindar sustento en las actividades que deben realizarse en cada uno de los procesos referentes a la gestión de calidad (planeación de la calidad, aseguramiento de la calidad y control de la calidad) mediante la selección adecuada de las herramientas, específicamente, herramientas de la gestión de la calidad para garantizar que la Calidad esté debidamente instaurada en un proyecto, y evitar que el énfasis sea corregir insatisfacciones en lugar de prevenir la ocurrencia de lo indeseado y el resultado no corresponda con lo esperado, sin descartar las actividades de mejora continua para aumentar la capacidad de cumplir con los requisitos de calidad.

De allí se deriva la necesidad de proponer un modelo que identifique

cuál(es) criterio(s) deben tener las herramientas del software para soportar cada uno de los procesos de la Gestión de Calidad definidos en este capítulo, de forma tal que ofrezcan un soporte que facilite la decisión para la selección adecuada de la herramienta, garantizando el éxito de su implementación partiendo de los conceptos asociados a la Calidad, su aplicación en el proceso de desarrollo del software y el desarrollo de proyectos exitosos en el ámbito de desarrollos del software.

Siendo el marco teórico el que ayuda a precisar y a organizar los elementos contenidos en la descripción del problema, de tal forma que puedan ser manejados y convertidos en acciones concretas (Tamayo y Tamayo, 1997), se hace necesario exponer un procedimiento ordenado que se debe seguir para establecer lo significativo de estos elementos hacia los cuales está encaminada la investigación. Es por ello, que a continuación se presenta la metodología de investigación utilizada.

CAPÍTULO III

MARCO METODOLÓGICO

El objetivo de este capítulo es describir la metodología que se siguió para el desarrollo del modelo que permita evaluar la calidad de las herramientas que soportan la gestión de la calidad. Básicamente, este capítulo tiene como objetivo principal especificar el framework metodológico de Pérez y sus colegas (2002), utilizado para la realización de esta investigación.

Framework Metodológico

Para proponer un modelo que permita evaluar las herramientas de la gestión de calidad se utilizó el Framework Metodológico del Laboratorio de Investigación en Sistemas de Información (LISI), el cual se inspira en el método Investigación Acción y utiliza la metodología DESMET para la fase de evaluación.

El presente trabajo de investigación se encuentra orientado hacia el estudio de los Sistemas de Actividad Humana, por lo que resulta complejo adaptarse a las metodologías de investigación tradicionales. Este tipo de sistemas, según Checkland (1993), está formado por un número de actividades conectadas como resultado de algún principio de coherencia. Para este mismo autor, estos sistemas son complejos, multivariantes y poseen fenómenos del mundo real que no pueden ser pasados por alto. El método investigación acción está orientado a atacar este tipo de problemas en los que el componente humano juega un papel crucial.

El fundamento del método de Investigación Acción es que los procesos de los Sistemas de Actividad Humana, pueden ser estudiados mejor si se

introducen cambios dentro de estos procesos y se observan los efectos que producen estos cambios (Baskerville, 1999). La descripción más frecuente del método Investigación Acción es la propuesta por Evered y Susman (1978) donde se detalla un proceso cíclico de cinco fases las cuales se pueden observar a continuación en la Figura N° 10.

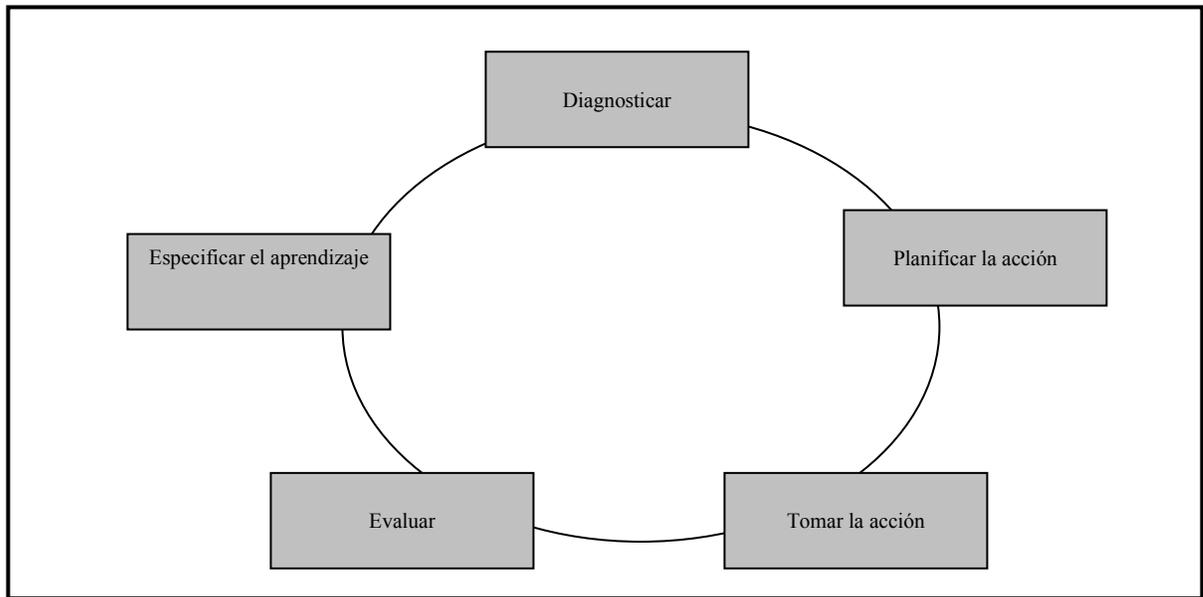


Figura 10. El ciclo de Investigación Acción
Fuente: [Baskerville, 1999]

Estas cinco fases identificadas por Baskerville (1999) interactúan entre sí, como se muestra en la Tabla N° 7.

Tabla N° 8.

Fases del Método Investigación Acción.

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.

Tabla N° 8 (Cont.)

Fase	Acciones
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 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 tácita.
Evaluar	Una vez completadas las acciones, los investigadores y demás participantes evalúan las salidas. 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 de investigación acción.
Especificar el aprendizaje	A partir del resultado de la evaluación, los investigadores especifican el conocimiento adquirido.

Fuente: (Baskerville, 1999)

Dada la naturaleza empírica de la fase de Evaluación del Método Investigación Acción, es necesario reforzarla con otras metodologías como DESMET, la cual según Kitchenham, (1996) ayuda al evaluador de una organización a planificar y a ejecutar la evaluación de manera imparcial y fiable.

El *Framework* Metodológico consta de 11 pasos o actividades agrupadas en 5 grandes fases, que se corresponden con las fases del Método Investigación Acción descrito anteriormente. A continuación, la

Figura N° 11 ilustra la adaptación del *Framework* Metodológico del LISI a la presente investigación.

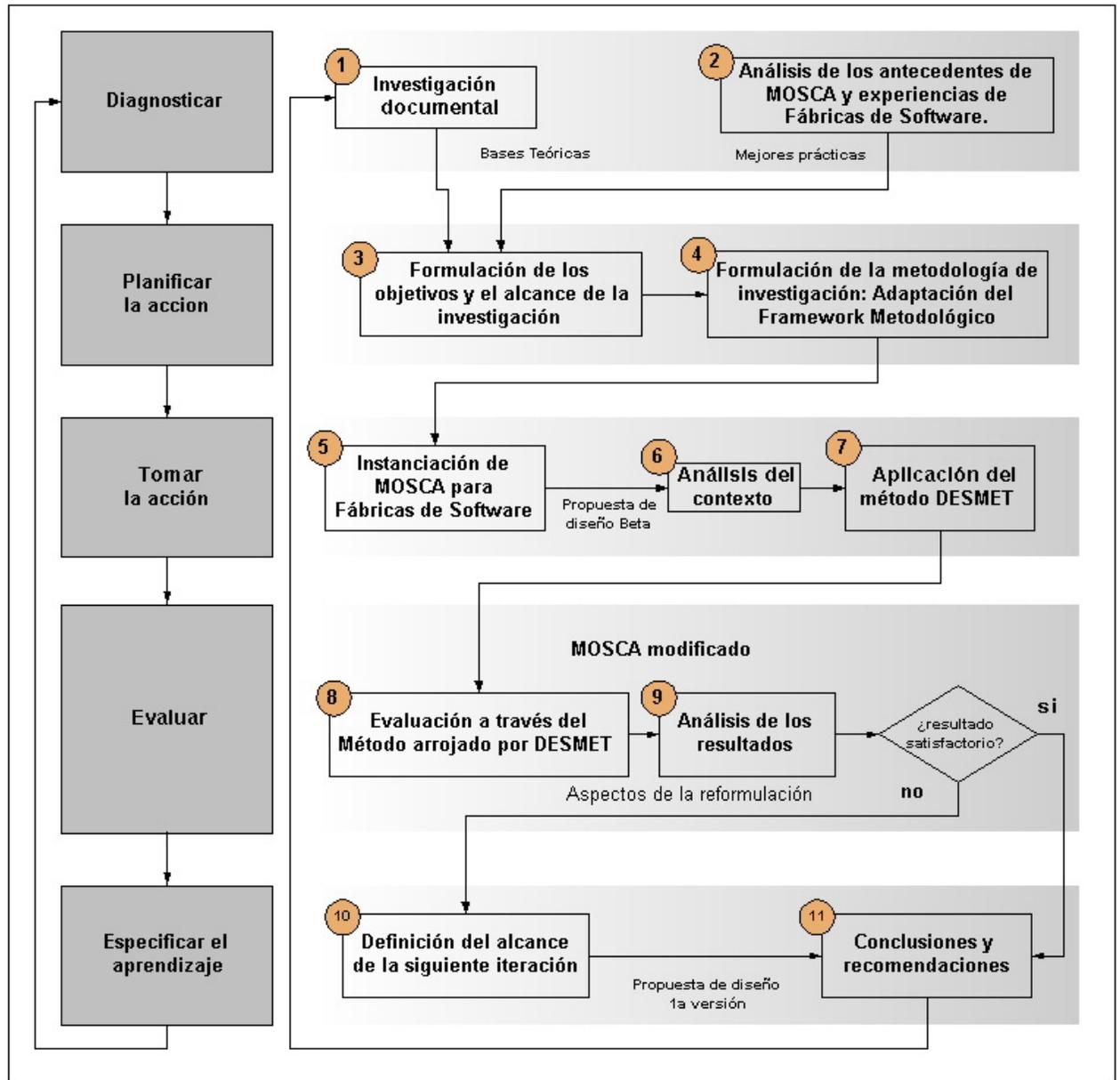


Figura N°11. Framework Metodológico para el trabajo de grado
Fuente: Adaptado de Pérez et al. (2002)

En la Tabla N° 9 se muestra el objetivo y una breve descripción de cada actividad.

Tabla N° 9.
Fases y Actividades de la Metodología de Investigación Acción Propuesta

Fases	Actividad	Descripción
Diagnosticar	1. Investigación documental.	Actividad de la fase de Diagnosticar, consiste en la revisión del material bibliográfico relacionado con las características, innovaciones y evolución histórica de las herramientas del software que soportan la gestión de calidad. El objetivo es concretar un marco conceptual que soporte el trabajo de investigación. El resultado de esta actividad fue presentado en el Capítulo II (Marco Teórico)
	2. Análisis de antecedentes de las herramientas de software.	Actividad de la fase de Diagnosticar, consiste en la descomposición de las herramientas del software y sus relaciones, a fin de evaluar la cobertura que posee actualmente. El resultado de esta actividad fue presentado en el Capítulo II (Antecedentes)
Planificar la Acción	3. Formulación de los objetivos y el alcance de la investigación.	Actividad de la fase Planificar la Acción donde se definen claramente los objetivos de la investigación. El objetivo es delimitar el área de investigación. El resultado de esta actividad está reflejado en la Justificación del Problema y la especificación de los Objetivos del Trabajo de Grado, presentados al inicio de la presente investigación.
	4. Formulación de la metodología de investigación: Adaptación de Investigación Acción.	Actividad de la fase Planificar la Acción donde se elabora la adaptación para esta investigación de la metodología de investigación acción. El objetivo es elaborar el marco metodológico que soporta el trabajo de investigación. El presente Capítulo es el resultado de esta actividad.

Tabla N° 9 (Cont.)

Fases	Actividad	Descripción
Tomar la Acción	5. Propuesta de un modelo para la gestión de Calidad.	Actividad de la fase Tomar la Acción. El objetivo es proponer un modelo que cubra los aspectos propios de la Gestión de Calidad.
	6. Análisis de Contexto.	Actividad de la fase Tomar la Acción, El objetivo es preparar el contexto donde será evaluada la propuesta. Se identifican criterios para ayudar a determinar el método de evaluación apropiado.
	7. Aplicación del método DESMET.	Actividad de la fase Tomar la Acción, donde se ejecuta la metodología DESMET. El objetivo es obtener un método fiable e imparcial para la evaluación.
Evaluar	8. Evaluación a través del método arrojado por DESMET.	Actividad de la fase Evaluar, que consiste en aplicar el método de evaluación arrojado por la actividad 7 al modelo propuesto. El objetivo es la evaluación del nuevo modelo mediante un método de evaluación apropiado.
	9. Análisis de los resultados.	Actividad de la fase Evaluar, que consiste en estudiar los resultados a partir de los objetivos planteados en el trabajo de investigación, en términos de: el modelo propuesto y los productos tangibles alcanzados. Al terminar esta fase se evalúa si el resultado es satisfactorio, en caso afirmativo se pasa al paso 11, en caso negativo se pasa al paso 10.
Especificar el Aprendizaje	10. Definición del alcance de la siguiente iteración.	Actividad de la fase Especificar el Aprendizaje, consiste en delimitar el alcance de las modificaciones que se deben realizar al modelo propuesto para aumentar la confiabilidad del mismo y sus posibilidades de éxito.

Tabla N° 9 (Cont.)

Fases	Actividad	Descripción
	11. Conclusiones y Recomendaciones.	Actividad de la fase Especificar el Aprendizaje, donde se establecen algunas conclusiones relativas al modelo propuesto aplicado y sus resultados. Finalmente, se sugieren algunas recomendaciones para futuros refinamientos del modelo propuesto y para investigaciones relacionadas.

Fuente: Elaboración propia, 2008

Debido a las limitaciones de tiempo, para la presente investigación se llevaran a cabo las siguientes fases: Diagnosticar, Planificar la acción y Tomar la acción y de esta fase será ejecutada sólo la actividad número 5, la cual consiste en proponer un modelo de gestión de calidad. Por otra parte, el ciclo metodológico usado soporta la investigación de problemas complejos, partiendo de prácticas exitosas y evaluando el modelo propuesto en versión Beta. Absorbe el aprendizaje y lo incorpora a una nueva propuesta mejorada. Este ciclo puede repetirse n veces; pero para efectos de esta investigación se consideró una sola iteración, dado que es un Trabajo de Grado con limitaciones de tiempo.

Cronograma de Ejecución

Actividades	Número de Semana												
	1	2	3	4	5	6	7	8	9	10	11	12	13
Vista Lógica de Sistemas													
<i>Diagnosticar</i>													
Investigación documental y análisis de los antecedentes													
Revisión de proyectos existentes y su forma de ejecución													
Identificación de riesgos													
<i>Planificar la acción</i>													
Formulación de los objetivos y alcance de la investigación													
Formulación de la metodología de investigación													
Selección del proyecto													
<i>Tomar la acción</i>													

Fuente: Elaboración Propia, 2008

CAPÍTULO IV MARCO ORGANIZACIONAL

Reseña Histórica

El Laboratorio de Investigación en Sistemas de Información **LISI** es un grupo de investigación de la sección de Sistemas de Información y Gestión del Departamento de Procesos y Sistemas de la Universidad Simón Bolívar, cuyo interés principal es el estudio de las tecnologías que apoyan el Proceso de Desarrollo de los Sistemas de Información.

Además, **LISI** busca crear una infraestructura de recursos físicos, tecnológicos y humanos que faciliten la investigación y la interacción eficiente y efectiva, con los últimos avances tecnológicos en el área de los Sistemas de Información.

El Laboratorio cuenta actualmente con un grupo de investigadores caracterizados por su profesionalismo y alta calidad humana. Nuestra oficina se encuentra ubicada en el Edificio de Laboratorios de Electrónica en la zona central del campus universitario.

Misión

LISI, se plantea como misión, el cumplimiento de un conjunto de objetivos que conforman el norte de todas las actividades que se realicen:

- Estudio de las Tecnologías de la Información que apoyan el proceso de desarrollo de los Sistemas de Información.
- Métodos y Metodologías para el desarrollo de los Sistemas de Información, utilizados en el mercado.
- Impacto de la Reingeniería de los Procesos de Negocio y su

- vinculación con las Tecnologías de la Información.
- Guías de Acción a las empresas del Sector Informático.
- Estructura Organizacional, Administración del Cambio y su relación con el proceso de desarrollo de los Sistemas de Información.
- Calidad en el Proceso de Desarrollo de los Sistemas de Información en Venezuela.

Vinculación con el Sector Productivo

Una de las actividades en las que el grupo de investigación **LISI** se ha esmerado y dedicado, es vincularse con el Sector Productivo en el área de la Informática en Venezuela, con la intención de relacionar los tópicos de investigación con las necesidades reales del país. Además, esto ha permitido comprobar hipótesis, y formular propuestas a través de estudios de campo en organizaciones, charlas, asesorías, entrenamientos, talleres y mentorías únicas en el país. Para cada caso se diseñan dinámicas muy específicas que permiten el logro del objetivo. Todas estas actividades se pueden llevar a cabo en la propia empresa o en las instalaciones de la USB. A través de FUNINDES se hacen los contactos administrativos y de planificación para contratar al **LISI** en la realización de estas actividades. La Universidad Simón Bolívar a través de FUNINDES entrega certificados que avalan asistencia a los cursos y cuando estos incluyen evaluaciones, certifica al participante.

Asesorías en Instanciación del Proceso Unificado 1999-2007

- Banco Andes
- Banco Mercantil
- Banco Banesco
- Banco Occidental de Descuento (BOD)
- Banco Provincial
- Banplus
- Business Ware V2
- CANTV
- CityBank PowerTeam
- Consis International

- Consltora Empresarial Eikos, CA
- Decanato de I&D- USB
- Digitel
- Grupo Distmah
- Grupo FJA, Feilmeier & Junker, Munich
- Grupo Pintacasa
- Hewlett-Packard
- Inelectra
- INPARQUES
- Intergraph Servicios
- INTESA
- Johnson & Johnson de Venezuela, C.A.
- Login C.A.
- Manapro Consultores C.A.
- MCT/CNTI
- Merlin Interactive Systems
- NETCOM de Venezuela
- Nokia de Venezuela
- PDVSA
- Policlínico Lucense SA (POLUSA) (España)
- Quimbiotec, C.A.
- Secretaría de Educación de la Alcaldía Mayor de Bogotá
- SENIAT
- Siqel Sistemas CA
- Sisorca
- Sun Microsystems
- Telcel
- Venclick.com
- Youth for Europe, Noruega

Asesorías en Certificación de Calidad 1996-2007

LISI en su compromiso por contribuir al continuo mejoramiento del proceso de desarrollo y en la preparación del sector informático venezolano para su participación en exigentes mercados, realiza asesorías dirigidas a empresas que desean conocer, definir y potenciar su proceso, en virtud de lograr mayor satisfacción de sus clientes o acreditaciones internacionales de calidad. Algunas empresas venezolanas con las que **LISI** ha compartido esta experiencia:

- PDVSA
- INTESA
- Corporación Systar
- Corpoven

Entrenamiento y Talleres 1995-2007

- Fundamentos de Rational Unified Process (RUP)
- Visión del Sistema - Analistas Sistemas
- Gestión de Requerimientos - Analistas Sistemas
- Diagramación con UML - Analistas Sistemas
- Visión de RUP para Gerentes/Líderes/Usuarios de Proyectos
- Fundamentos de RUP - Consultores Procesos
- Visión de RUP para Consultores de Procesos
- Modelado de Procesos de Negocio con UML - Consultores Procesos
- Formación de Consultores Internos
- Taller de Definición Metodológica
- Análisis y Diseño de Sistemas con Lenguaje de Modelo Unificado (UML)
- Gestión de Calidad
- Como alcanzar Nivel 2 CMM
- Estrategia de Pruebas

Mentorías 1995-2007

Se preparan mentorías de seguimiento a los Proyectos Pilotos que se van integrando con la Propuesta Metodológica. Esto permite ayudar a través de un seguimiento por parte de nuestros consultores, a que se logre un aprendizaje aplicado a la organización y así mismo a que se vaya formando el conocimiento necesario para asumir nuevos Proyectos dentro de la Organización, lo que se considera como Pivotes Metodológicos.

Estos Pivotes así como todos los que participan en la ejecución de los Proyectos Pilotos, deben ser entrenados antes y durante las actividades que se lleven a cabo, a través de Cursos diseñados para tal fin y que son ofertados en esta misma sección.

Para todo este proceso el **LISI** dispone de metodologías que permiten desde sensibilizar a la Alta Gerencia hasta facilitar al nivel operativo la ejecución de las actividades, lo que permite combinar apropiadamente el manejo tecnológico con consideraciones relacionadas con el Manejo del Cambio.

En sintonía con lo que estipula Extreme Programming en relación a la calidad a través del trabajo en parejas, el **LISI** tiene como filosofía atender sesiones de Mentorías y Proyectos Pilotos con duplas de consultores, lo cual le da calidad, eficiencia y multidisciplinaridad a los criterios y sugerencias manejados.

Adopción Tecnológica

Desde hace varios años a nivel mundial, se hacen esfuerzos por solucionar los problemas de las organizaciones a través de la adquisición y su posterior adopción tecnológica.

El grupo **LISI** se mantiene investigando sobre este tópico, ya que ha venido observando dentro y fuera del país, que la tecnología por si sola no conduce al éxito. Es por ello que las propuestas que se formulan en este grupo de investigación se fundamentan en la trilogía: Estructura, Tecnología y Gente.

En donde se demuestra que el éxito de emprender una adopción tecnológica depende de la interacción sistémica de los elementos: tecnología, estructura organizacional y la gente que utilizará y/o interactuará con la tecnología.

Visión Futura

La Universidad Simón Bolívar se ha caracterizado por exigir a sus profesores divulgación formal de los resultados alcanzados a través de las investigaciones que son financiadas y apoyadas con recursos del Decanato de Investigación y Desarrollo (DID), así como a través del CONICIT, empresas del sector productivo del país y convenios académicos-fuentes externas (Banco Interamericano de Desarrollo).

Dentro de este marco, los resultados deben ser presentados ante comunidades especialistas en el área, nacionales e internacionales, que sean arbitradas, las cuales se constituyen en diversas experiencias venezolanas que se realizan en la Universidad Simón Bolívar a través del laboratorio.

Todos estos resultados son a su vez, publicados en revistas, proceedings de Conferencias, en sitios Web, CD's, etc., para su difusión.

Una vista hacia el futuro, proyecta al **LISI** Como un centro especializado que, a través de los canales regulares que administra la Universidad Simón Bolívar, se convertirá en un ente imparcial capaz de asesorar al sector productivo del país en el área del desarrollo de los Sistemas de Información. Sobre la base de esta visión el **LISI** propone su Plan de Trabajo para el 2002-2003.

CAPITULO V

DESARROLLO

En este capítulo se presenta la Propuesta del Modelo que especifica la calidad de herramientas que soportan la Gestión de Calidad del Software. En principio se presenta y se describe el mapa conceptual; con la finalidad de aprovechar los conocimientos teóricos ofrecidos por las fuentes bibliográficas consultadas, las cuales ayudan a orientar el desarrollo de la propuesta como producto tangible de esta investigación. El mapa conceptual, el cual contiene todos los aspectos teóricos que envuelven la propuesta del modelo a presentar y la relación existente entre estos conceptos. Posteriormente, se presenta la instanciación del Modelo Sistémico de Calidad MOSCA, en el cual se presentan las categorías a nivel del producto que fueron tomados en cuenta para esta propuesta. Por último, se operacionaliza esta instanciación y se indican los cambios y aportes del propuesto con respecto al modelo MOSCA base.

Mapa Conceptual Asociado a las Herramientas de la Gestión de la Calidad del Software

La Calidad del Software representa un factor importante para las organizaciones desarrolladoras de software, ya que les permite mantenerse vigentes en el mercado convirtiéndose líderes en el mismo, partiendo de que persiguen altos niveles de aceptación en el mercado y la satisfacción de las necesidades de sus clientes.

En la Figura N° 12 se puede identificar lo anteriormente expuesto, lo que representa la primera parte del mapa conceptual global que se presenta al final de esta sección.

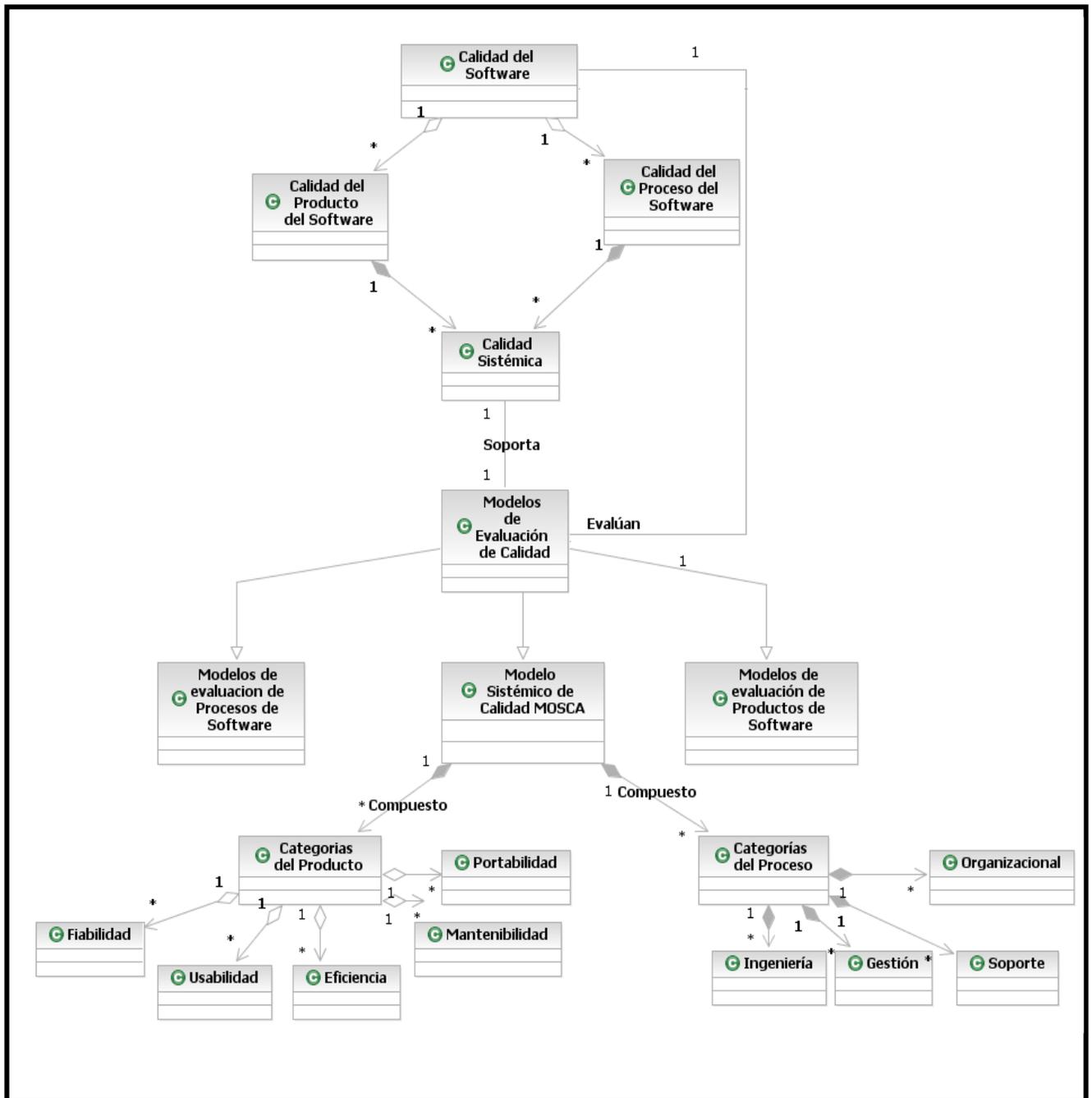


Figura 12. Mapa Conceptual: Calidad del Software

Fuente: Elaboración Propia, 2008

Como se observa en la Figura N° 12, la Calidad del Software puede orientarse al producto o al proceso, y la combinación de estos permite alcanzar la calidad sistémica. Por otra parte, para garantizar la Calidad del Software es importante realizar evaluaciones, esto ha motivado que existan

un conjunto de modelos que se utilizan para estimar calidad, los cuales responden a las necesidades de garantizar productos de calidad.

Existen modelos que evalúan la calidad del producto del software y modelos que evalúan la calidad de procesos del software. Sin embargo, en Venezuela se desarrolló el Modelo Sistémico de Calidad MOSCA, que evalúa la calidad tanto del proceso de desarrollo como del producto final, tomando en cuenta un enfoque sistémico. Por lo tanto, es posible afirmar que la calidad sistémica brinda soporte a los modelos que estiman la Calidad del Software, representando estos modelos las respuestas a las necesidades de garantizar productos de calidad en una competencia abierta y mundial, sobre la base de cada tipo de proceso de software.

El Modelo Sistémico de Calidad MOSCA posee dos dimensiones relacionadas al aspecto interno y al aspecto contextual tanto de la perspectiva del proceso de desarrollo como la perspectiva del producto. Cada una de estas dimensiones posee categorías asociadas. La perspectiva producto posee las siguientes categorías: Funcionalidad, Fiabilidad, Usabilidad, Eficiencia, Mantenibilidad y Portabilidad. Así mismo, cada una de las categorías posee un conjunto de características asociadas con sus respectivas métricas, lo que hace posible medir el cumplimiento de cada una de las características en el producto.

Por otra parte, otra forma de soportar la Calidad del Software es haciendo uso de Herramientas del Software, por el hecho de que estas brindan soporte en casi todas las etapas del desarrollo, mejorando la productividad en el diseño y en el desarrollo, contribuyendo así a la producción de software de alta calidad. Existe un conjunto de herramientas dentro de cada subárea de conocimiento de la Ingeniería del Software, cuya acción se asocia a las actividades propias de cada uno de estos. Los tópicos de las Herramientas del Software como subárea de conocimiento de la Ingeniería del Software, pueden visualizarse en la Figura N° 13.

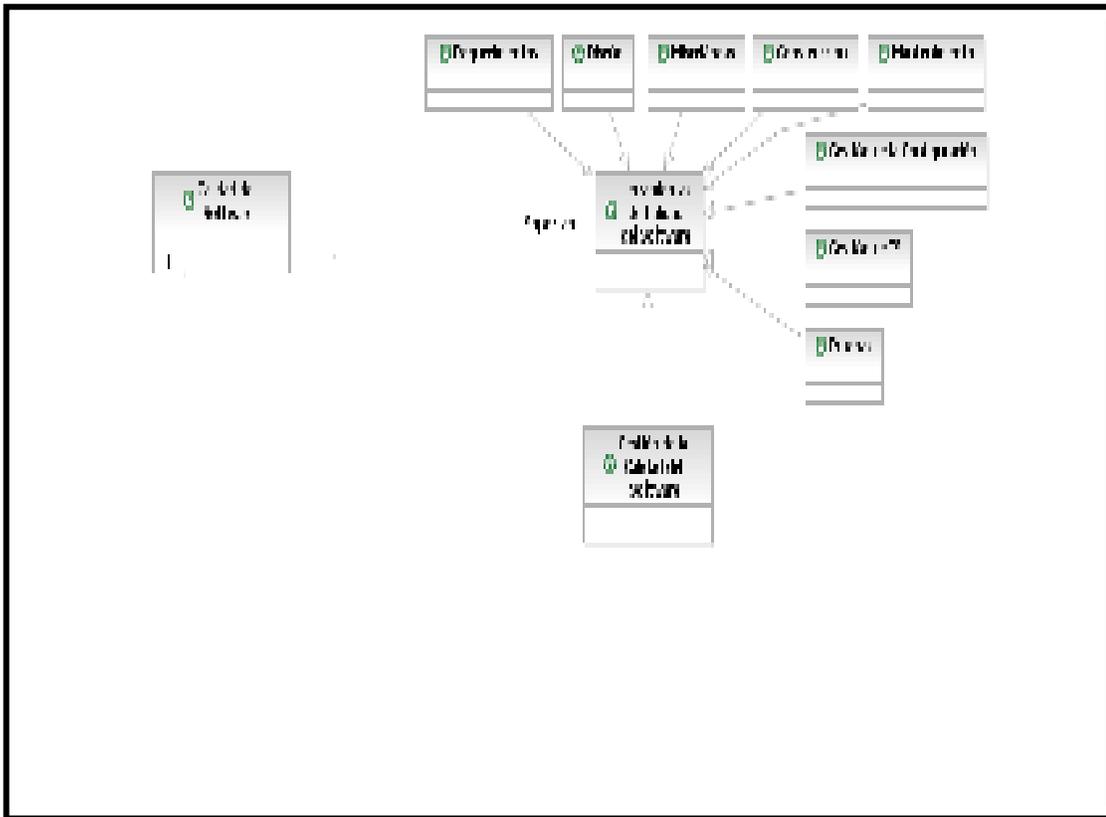


Figura 13. Mapa Conceptual: Herramientas del Software

Fuente: Elaboración Propia, 20008

Para efectos de esta investigación, será tomada en cuenta el área de conocimiento llamada Gestión de la Calidad del Software, la cual se muestra en la Figura N° 12 y se especifica en la Figura N° 13. Esta área de conocimiento incluye tres procesos los cuales son: planeación de la calidad del software, aseguramiento de la calidad del software y control de la calidad del software. Cada uno de estos procesos lleva consigo una serie de actividades que conlleva a aplicar técnicas y hacer uso de herramientas. En la Figura N° 14 se ilustra cada uno de los procesos subyacentes en la Gestión de la Calidad del Software identificados en la presente investigación con sus actividades respectivas.

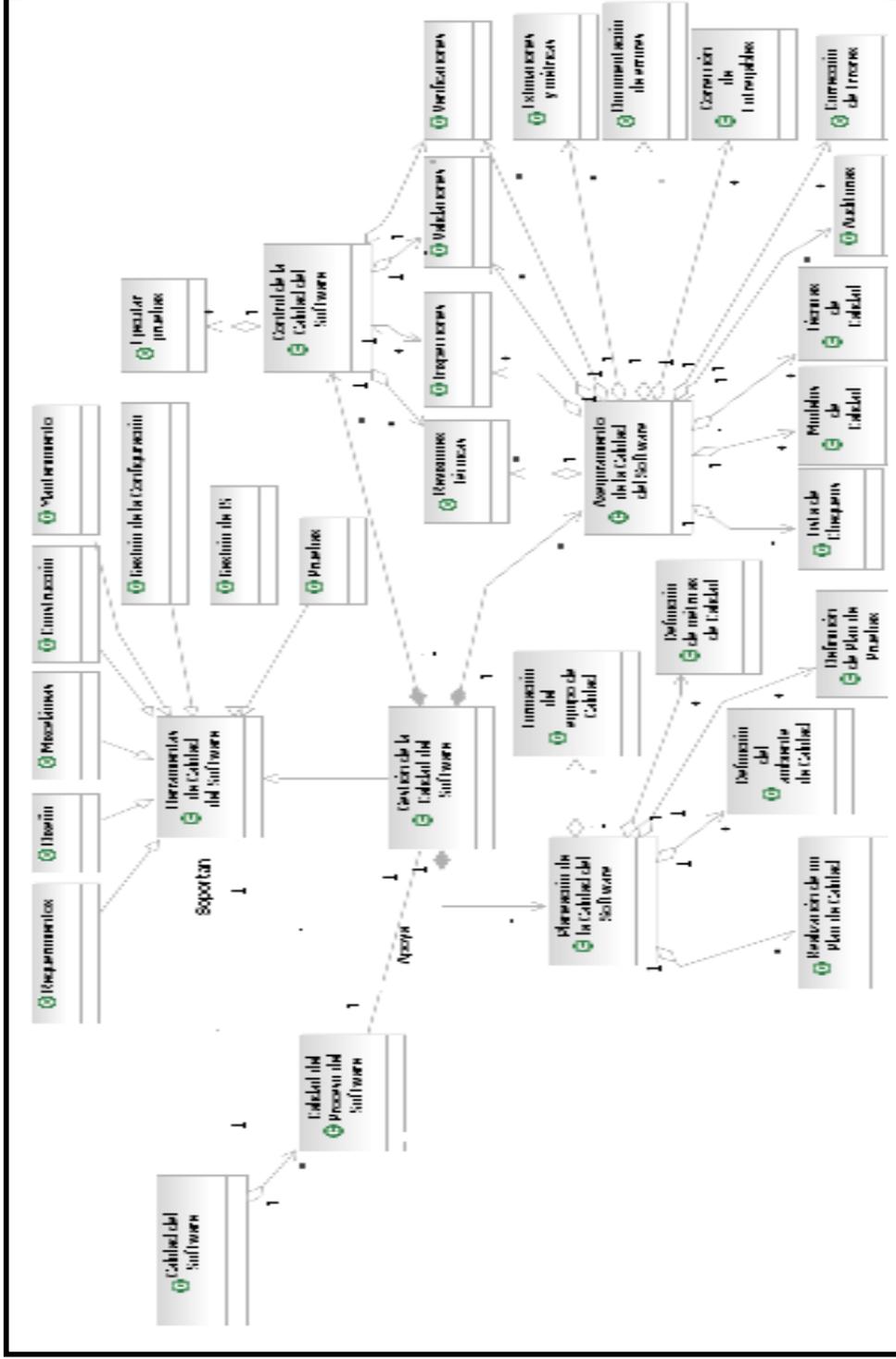


Figura 14. Mapa Conceptual: Gestión de la Calidad del Software
Fuente: Elaboración Propia, 2008

A continuación, se presenta la Figura N° 15, en la cual se visualiza el mapa conceptual desarrollado completamente.

Después de presentar el mapa conceptual, el cual brinda las bases de la investigación y los aspectos que deben considerarse para especificar la calidad de las herramientas de gestión de calidad, se presenta en la siguiente sección la identificación de los elementos presentes en el modelo sistémico de calidad MOSCA que servirán para desarrollar el modelo propuesto.

Identificación de los Elementos Presentes en el Modelo Sistémico de Calidad MOSCA

El Modelo Sistémico MOSCA trabaja sobre una plataforma que integra el modelo de calidad del producto de software y el modelo de calidad del proceso de desarrollo de software con un enfoque sistémico.

Para el modelo propuesto, se tomará en cuenta sólo la perspectiva de producto, debido a que las herramientas brindan un soporte para mejorar la calidad del producto y no es alcance de esta investigación evaluar o especificar la calidad para las mismas. Esta perspectiva posee seis (6) categorías asociadas, con sus respectivas características, las cuales se presentan en la Tabla N° 8.

Tabla N° 10

Categorías y Características de la Perspectiva Producto del Modelo MOSCA

PRODUCTO	
Categoría	Características
Funcionalidad	FUN.1 Ajuste a los propósitos
	FUN.2 Precisión
	FUN.3 Interoperabilidad
	FUN.4 Seguridad
	FUN.5 Correctitud
	FUN.6 Estructurado
	FUN.7 Encapsulado
	FUN.8 Especificado

Tabla N° 10 (Cont.)

PRODUCTO	
Categoría	Características
Fiabilidad	FIA.1 Madurez
	FIA.2 Tolerancia a Fallas
	FIA.3 Recuperación
	FIA.4 Correctitud
	FIA.5 Estructurado
	FIA.6 Encapsulado
Usabilidad	USA.1 Facilidad de comprensión
	USA.2 Capacidad de aprendizaje
	USA.3 Interfaz Grafica
	USA.4 Operabilidad
	USA.5 Conformidad con los estándares
	USA.6 Completo
	USA.7 Consistente
	USA.8 Efectivo
	USA.9 Especificado
	USA.10 Documentado
	USA.11 Auto-Descriptivo
Eficiencia	EFI.1 Comportamiento del tiempo
	EFI.2 Utilización de recursos
	EFI.3 Efectivo
	EFI.4 No-redundante
	EFI.5 Directo
	EFI.6 Utilizado
Mantenibilidad	MAB.1 Capacidad de análisis
	MAB.2 Capacidad de cambio
	MAB.3 Estabilidad
	MAB.4 Capacidad de prueba
	MAB.5 Acoplamiento
	MAB.6 Cohesión
	MAB.7 Encapsulamiento
	MAB.8 Atributos de Madurez del Software
	MAB.9 Sistemas de Estructura de Control
	MAB.10 Sistema de Estructura de Información
	MAB.11 Descriptivo
	MAB.12 Correctitud
	MAB.13 Estructural

Tabla N° 10 (Cont.)

PRODUCTO	
Categoría	Características
	MAB.2 Capacidad de cambio
	MAB.3 Estabilidad
	MAB.4 Capacidad de prueba
	MAB.5 Acoplamiento
	MAB.6 Cohesión
	MAB.7 Encapsulamiento
	MAB.8 Atributos de Madurez del Software
	MAB.9 Sistemas de Estructura de Control
	MAB.10 Sistema de Estructura de Información
	MAB.11 Descriptivo
	MAB.12 Correctitud
	MAB.13 Estructural
	Portabilidad
POR.2 Capacidad de instalación	
POR.3 Co-existencia	
POR.4 Capacidad de reemplazo	
POR.5 Consistencia	
POR.6 Parametrizado	
POR.7 Encapsulado	
POR.8 Cohesivo	
POR.9 Especificado	
POR.10 Documentado	
POR.11 Auto-descriptivo	
POR.12 No-redundante	

Fuente: Elaboración Propia, 2008

Por otra parte, para cada una de estas características existe un conjunto de subcaracterísticas asociadas, y éstas a su vez están relacionadas a una o más métricas, las cuales se utilizan para medir el grado de calidad sistémica presente en el producto de software.

MOSCA es un modelo genérico para especificar la calidad de las herramientas de gestión de la calidad del software, se deben identificar actividades inmersas en los procesos de planeación de calidad, aseguramiento de la calidad y control de la calidad, los cuales forman parte

de la Gestión de Calidad del Software, a fin de proponer subcaracterísticas y métricas acordes a las herramientas de este tipo. De esta manera, es posible evaluar la calidad de una herramienta de calidad del software.

El algoritmo de MOSCA, exige que además de la categoría Funcionalidad se seleccionen dos categorías adicionales, ya que pretender que una herramienta cumpla con las seis categorías no es factible, pues ellas entran en contradicción. Por lo tanto, las categorías seleccionadas para desarrollar la propuesta del modelo son las siguientes: Funcionalidad, Usabilidad y Fiabilidad. Dado a que esta primera versión de la propuesta está enfocada a medir la calidad de las herramientas en función de los resultados esperados por los usuarios que hacen uso de las mismas (funcionalidad), la capacidad de la herramienta de software para mantener un nivel especificado de rendimiento cuando es utilizado bajo condiciones especificadas (fiabilidad) y la capacidad de la herramienta para ser atractiva, entendida, aprendida y utilizada por el usuario bajo condiciones específicas (usabilidad) mientras que el resto de las categorías contienen características en las cuales sus métricas, se orientan a medir resultados esperados por los desarrolladores. A continuación, se presenta la justificación de la selección de las tres (3) categorías.

- Categoría Funcionalidad:

La categoría Funcionalidad es una categoría obligatoria a evaluar en todo software o herramienta. Se considera importante medir como una herramienta provee funciones que cumplan con necesidades específicas o implícitas, cuando estas son utilizadas, en este caso, en un contexto de investigación, con funciones propias de la gestión de la calidad.

La funcionalidad posee las siguientes características: el ajuste a los propósitos, la precisión, la interoperabilidad, la seguridad del producto de software, correctitud, estructurado, encapsulado y especificado. Sin embargo, sólo se seleccionaran tres (3) de ellas, las cuales son: Ajustes a los

propósitos, Precisión e Interoperabilidad, ya que las categorías de correctitud, estructurado, encapsulado y especificado se orientan a aspectos internos y para efectos de esta investigación la orientación es hacia los aspectos contextuales ya que no se realizó la revisión del código interno de las herramientas de software.

Esta selección estuvo basada principalmente en identificar como una herramienta de gestión de calidad del software provee un conjunto de funciones apropiadas de acuerdo al objetivo específico del usuario quien hace uso de ella.

El resto de las características de Funcionalidad que no fueron seleccionadas no escapan del alcance para medir la calidad que pueda presentar una herramienta de gestión de la calidad de software; sin embargo, la propuesta del modelo a desarrollar es presentada en una primera versión, para lo cual la selección de las características se orientó al resultado esperado por los usuarios que hacen uso de una herramienta de este tipo, mientras las características de seguridad del producto de software, correctitud, estructurado, encapsulado y especificado se enfocan hacia resultados buscados por los desarrolladores de software.

- Categoría Usabilidad:

La categoría de Usabilidad fue seleccionada porque es importante conocer la capacidad que posee una herramienta del software para ser atractiva, entendida, aprendida y utilizada por el usuario.

La usabilidad posee las siguientes características: facilidad de comprensión, capacidad de aprendizaje, interfaz gráfica, operabilidad, conformidad con los estándares, completo, consistente, efectivo, especificado, documentado, y auto-descriptivo. Sin embargo, sólo se seleccionaron cuatro (3) de ellas, las cuales son: Facilidad de comprensión, Capacidad de aprendizaje e Interfaz Gráfica.

La selección de estas características se fundamenta básicamente en la importancia que provee una herramienta para gestión de la calidad, facilitar al

usuario el entendimiento de la misma, la forma en que puede ser utilizada, la facilidad para guiar al usuario en el aprendizaje y los atributos que pueden hacerla más atractiva hacia el usuario.

Al igual que en la categoría de Funcionalidad, el resto de las características de Usabilidad no fueron seleccionadas porque en esta primera versión de la propuesta del modelo a desarrollar, la evaluación de las herramientas de la gestión de calidad del software se enfoca al resultado esperado por los usuarios y no hacia los resultados buscados por los desarrolladores de software, tal como se orientan el resto de las características.

- Categoría Fiabilidad:

La categoría de Fiabilidad fue seleccionada dada la importancia de la capacidad que las Herramientas de gestión de calidad pueden proporcionar en función de mantener un nivel de rendimiento con la precisión requerida cuando son utilizadas bajo ciertas condiciones

La fiabilidad posee las siguientes características: madurez, tolerancia a falla, recuperación, correctitud, estructurado y encapsulado. Sin embargo, sólo se seleccionaran dos (2) de ellas, las cuales son: Tolerancia a fallas y Recuperación.

La selección de estas características se basa en el interés de evaluar herramientas de gestión de calidad en cuanto al rendimiento específico en caso de errores en el software o de problemas en sus interfaces, así como también la capacidad que poseen en restablecer su rendimiento y recuperar datos afectados en caso de alguna falla, es decir, la disponibilidad para estar operativo en el momento de ejecutar una función en un momento determinado. El resto de las características no fueron tomadas en cuenta ya que se orientan a evaluar código de software y para efectos de esta propuesta no está al alcance el software de cada una de las herramientas a evaluar.

A continuación se presenta la propuesta del Modelo que especifica la

calidad de herramientas que soportan la Gestión de Calidad del Software.

Propuesta del Modelo de Gestión de Calidad del Software para Seleccionar Herramientas del Software

Como resultado de los elementos identificados en MOSCA, se logró obtener un conjunto de subcaracterísticas asociados a los procesos de planeación de la calidad, aseguramiento de la calidad y control de la calidad inmersos en la Gestión de la Calidad del Software, las cuales permiten evaluar las herramientas del software que soportan dichos procesos.

A continuación, se presenta en la Tabla N°9 la Propuesta del Modelo Gestión de Calidad del Software para seleccionar herramientas del software

Tabla N° 11
Propuesta del Modelo de Gestión de la Calidad del Software para
seleccionar herramientas del software
Métricas Nuevas: *
Métricas Modificadas: **

Característica	SubCaracterística	SubSubCaracterística	Pregunta	Formulación
Funcionalidad(FUN)				
FUN1. Ajustes a los propósitos	Planeación de la Calidad del Software	Cumplimiento de realización de plan de calidad.*	¿Existen funcionalidades asociadas a la herramienta que permitan identificar objetivos de calidad?	5=Completa def. 4=Casi todo def. 3= Med. Def. 2= Poco def. 1= No están def.
		Cumplimiento de realización de plan de calidad.*	¿Existen funcionalidades asociadas a la herramienta que permitan identificar atributos de calidad?	5=Completa def. 4=Casi todo def. 3= Med. Def. 2= Poco def. 1= No están def.
		Cumplimiento de realización de plan de calidad.*	¿Existen funcionalidades asociadas a la herramienta que permita la	5=Completa def. 4=Casi todo def. 3= Med. Def. 2= Poco def. 1= No están def.

			realización de un cronograma para definir la ejecución de pruebas?	
		Cumplimiento de realización de plan de calidad.*	¿Existen funcionalidades asociadas a la herramienta que permita enviar alertas para indicar al usuario la definición de las diferentes pruebas a realizar en las distintas fases del proyecto?	5=Completam. def. 4=Casi todo def. 3= Med. Def. 2= Poco def. 1= No están def.
		Cumplimiento de realización de plan de calidad.*	¿Existen funcionalidades asociadas a la herramienta que permita identificar el propósito del plan de calidad?	5=Completam. def. 4=Casi todo def. 3= Med. Def. 2= Poco def. 1= No están def.
		Cumplimiento de realización de plan de calidad.*	¿Existen funcionalidades asociadas a la herramienta que permita enviar alerta para diseñar casos de pruebas?	5=Completam. def. 4=Casi todo def. 3= Med. Def. 2= Poco def. 1= No están def.
		Cumplimiento de realización de plan de calidad.*	¿Existen funcionalidades asociadas a la herramienta que permita enviar alerta para indicar al usuario la ejecución de las diferentes pruebas a realizar en las distintas fases del proyecto?	5=Completam. def. 4=Casi todo def. 3= Med. Def. 2= Poco def. 1= No están def.
		Cumplimiento de realización de plan de calidad.*	¿La herramienta permite definir la estructura de una matriz en la cual se puedan reportar defectos?	5=Completam. def. 4=Casi todo def. 3= Med. Def. 2= Poco def. 1= No están def.
		Cumplimiento de realización de plan de calidad.*	¿La herramienta permite definir los diferentes roles que serán	5=Completam. def. 4=Casi todo def. 3= Med. Def. 2= Poco def.

			ejercidos durante la ejecución de las pruebas?	1= No están def.
		Cumplimiento de realización de plan de calidad.*	¿La herramienta envía alertas al usuario indicando que deben definirse los estándares de calidad?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Cumplimiento de realización de plan de calidad.*	¿La herramienta envía alertas al usuario indicando que deben definirse métricas de calidad?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Cumplimiento de realización de plan de calidad.*	¿La herramienta envía alertas al usuario para identificar recursos a utilizar?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Cumplimiento de realización de plan de calidad.*	¿La herramienta permite definir un plan de calidad completo?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Cumplimiento de realización de plan de calidad.*	¿La herramienta permite el manejo de versiones del plan de calidad?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Cumplimiento de realización de plan de calidad.*	¿La herramienta provee funcionalidades para identificar los roles y responsabilidades asociadas a la planeación de la calidad?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Definición de estándares de calidad.*	¿La herramienta envía alarmas indicando la definición de estándares de la documentación?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Definición de estándares de calidad.*	¿La herramienta envía alarmas indicando la definición de estándares de	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.

			codificación?	
Definición de estándares de calidad.*	¿La herramienta envía alarmas indicando la definición de estándares de elaboración de manuales?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.		
Definición de estándares de calidad.*	¿La herramienta envía alarmas indicando la definición de estándares de controles de cambio?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.		
Definición de estándares de calidad.*	¿La herramienta envía alarmas indicando la definición de estándares de documentación del código fuente?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.		
Definición de estándares de calidad.*	¿La herramienta envía alarmas indicando la definición de estándares de interfaz?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.		
Definición de estándares de calidad.*	¿La herramienta permite definir formularios para la revisión del diseño?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.		
Definición de estándares de calidad.*	¿La herramienta permite definir formatos para los controles de cambio?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.		
Definición de estándares de calidad.*	¿La herramienta envía alarmas indicando la definición de estándares registros de pruebas?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.		
Definición de estándares de calidad.*	¿La herramienta envía alarmas indicando la definición de documentos para el soporte a los	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.		

			usuarios?	
		Formación del equipo de calidad.*	¿La herramienta permite definir la estructura del equipo de calidad?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Formación del equipo de calidad.*	¿La herramienta permite definir los roles y sus actividades para cada una de las pruebas?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Formación del equipo de calidad.*	¿La herramienta envía alarmas al usuario para recordar la actividad de cada uno de los roles definidos?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Formación del equipo de calidad.*	¿La herramienta permite definir la el flujo de aprobación y revisiones de los documentos?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Formación del equipo de calidad.*	¿La herramienta envía alarmas para la realización de un cronograma de reuniones para revisiones de los objetivos de cada prueba?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Formación del equipo de calidad.*	¿La herramienta permite manejar agenda o calendario de actividades?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Definición del ambiente de calidad. *	¿La herramienta ofrece alternativas para configurar entornos tecnológicos de acuerdo a las pruebas definidas?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Definición del ambiente de calidad. *	¿La herramienta envía alarmas para incorporar los datos de pruebas a utilizarse en la ejecución de las mismas?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.

		Definición del ambiente de calidad. *	¿La herramienta posee funcionalidad que permita definir la estructura de matriz de reserva de ambientes?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Definición del ambiente de calidad. *	¿La herramienta ofrece reportes de las versiones del código que se encuentran compartidos y sus responsables?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Definición del ambiente de calidad. *	¿La herramienta ofrece reportes de los cambios que se ejecutan en el ambiente y su responsable?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Definición del ambiente de calidad. *	¿La herramienta permite definir formatos para el reporte de incidencias en los distintos ambientes?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Definición del ambiente de calidad. *	¿La herramienta permite registrar la dependencia entre aplicaciones ante una ejecución de pruebas?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Definición de un plan de pruebas. *	¿La herramienta permite determinar ítems a ser probados?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Definición de un plan de pruebas. *	¿La herramienta permite determinar características a ser probados?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Definición de un plan de pruebas. *	¿La herramienta permite determinar técnicas de pruebas?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Definición de un plan de pruebas. *	¿La herramienta permite criterios de aprobación?	5=Completam.. 4=Casi todo. 3= Med..

			2= Poco. 1= No.
Definición de un plan de pruebas. *	¿La herramienta permite identificar hitos de pruebas (entregas)?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Definición de un plan de pruebas. *	¿La herramienta permite identificar necesidades de ambiente de pruebas?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Definición de un plan de pruebas. *	¿La herramienta permite identificar recursos y/o entrenamientos necesarios para la ejecución de las pruebas?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Definición de un plan de pruebas. *	¿La herramienta permite identificar riesgos y planes de contingencias asociados a la ejecución de las pruebas?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Definición de un plan de pruebas. *	¿La herramienta permite identificar especificaciones de entrada?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Definición de un plan de pruebas. *	¿La herramienta permite identificar especificaciones de salida?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Definición de un plan de pruebas. *	¿La herramienta permite identificar dependencias entre los casos de pruebas definidos?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Definición de un plan de pruebas. *	¿La herramienta permite definir la secuencia de acciones a ser realizadas por el personal de prueba?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Definición de un plan de pruebas. *	¿La herramienta permite identificar los tipos de	5=Completam.. 4=Casi todo. 3= Med..	

			pruebas a incluir o excluir?	2= Poco. 1= No.
		Definición de métricas de calidad. *	¿La herramienta permite definir atributos de calidad para definir el propósito de las distintas métricas?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Definición de métricas de calidad. *	¿La herramienta permite establecer puntos de evaluación por métrica?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Definición de métricas de calidad. *	¿La herramienta permite definir valoración de cada pregunta por cada métrica?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Definición de métricas de calidad. *	¿La herramienta permite definir tablas de métricas por característica?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
	Aseguramiento de la Calidad	Cumplimiento de los procesos establecidos. *	¿La herramienta posee listas de chequeos para verificar el cumplimiento de los procesos establecidos?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Cumplimiento de los estándares establecidos. *	¿La herramienta posee listas de chequeos para verificar el cumplimiento de los estándares establecidos?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Identificación de los modelos de calidad *	¿La herramienta me permite cargar actividades asociadas a un modelo de calidad del proceso?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Identificación de los modelos de calidad *	¿La herramienta me permite cargar un reporte con respecto a las especificaciones que propone el modelo de calidad del proceso?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.

		cargado?	
Selección de técnicas de calidad. *	¿La herramienta ofrece alternativas de distintas técnicas de calidad para su elección?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Realización de revisiones técnicas. *	¿La herramienta permite puntos de revisión?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Realización de revisiones técnicas. *	¿La herramienta permite definir responsabilidades de cada participante en la revisión?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Realización de revisiones técnicas. *	¿La herramienta envía notificaciones automáticas en función del cronograma establecido de la revisión?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Realización de revisiones técnicas. *	¿La herramienta envía notificaciones automáticas de problemas que han sido identificados en las revisiones y no han sido resueltos?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Realización de revisiones técnicas. *	¿La herramienta permite definir una lista de chequeo por producto a revisar?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Realización de revisiones técnicas. *	¿La herramienta permite identificar los recursos necesarios para la revisión?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Realización de revisiones técnicas. *	¿La herramienta envía notificaciones para el entrenamiento a los participantes de la revisión?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Realización de	¿La herramienta	5=Completam..	

		revisiones técnicas. *	permite identificar un calendario de entrenamiento?	4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de revisiones técnicas. *	¿La herramienta permite estructurar la trazabilidad entre los requerimientos funcionales y los requerimientos del software?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de revisiones técnicas. *	¿La herramienta permite estructurar la trazabilidad entre pruebas y los requerimientos del software?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de revisiones técnicas. *	¿La herramienta permite definir formularios para realizar el informe como resultado de la revisión?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de revisiones técnicas. *	¿La herramienta permite manejar un calendario y/o agenda para las revisiones técnicas?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de revisiones técnicas. *	¿La herramienta ofrece técnicas de calidad?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de inspecciones. *	¿La herramienta permite definir criterios de inspección?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de inspecciones. *	¿La herramienta envía notificaciones automáticas para definir el grupo de inspección?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de inspecciones. *	¿La herramienta permite definir roles a los participantes que integran el grupo	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.

			de inspección?	
Realización de inspecciones. *	¿La herramienta permite identificar los productos a los cuales se les realizará la inspección?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.		
Realización de inspecciones. *	¿La herramienta permite determinar las fases en las que se realizaran las inspecciones?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.		
Realización de inspecciones. *	¿La herramienta define formatos para encontrar defectos?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.		
Realización de inspecciones. *	¿La herramienta permite reportar defectos y/o errores?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.		
Realización de inspecciones. *	¿La herramienta envía notificaciones automática para la realización de reuniones del grupo de inspección?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.		
Realización de inspecciones. *	¿La herramienta permite manejar calendario de inspecciones?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.		
Realización de inspecciones. *	¿La herramienta permite generar reportes de inspecciones (que se inspecciono, quien lo hizo, número de defectos encontrados)?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.		
Realización de inspecciones. *	¿La herramienta ofrece manuales de inspección?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.		
Cumplimiento de procesos de validación. *	¿La herramienta permite validar las	5=Completam.. 4=Casi todo.		

		características del producto de acuerdo a los requisitos establecidos?	3= Med.. 2= Poco. 1= No.
Cumplimiento de procesos de validación. *	¿La herramienta permite definir un calendario del proceso de validación?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Cumplimiento de procesos de validación. *	¿La herramienta permite manejar tickets para la asignación del personal asociado a este proceso?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Cumplimiento de procesos de validación. *	¿La herramienta permite definir las fases en las cuales el producto será validado?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Cumplimiento de procesos de validación. *	¿La herramienta envía notificaciones automáticas para ejecución de pruebas?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Cumplimiento de procesos de validación. *	¿La herramienta envía notificaciones para realizar análisis de fallas?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Cumplimiento de procesos de validación. *	¿La herramienta envía notificaciones para realizar análisis de defectos?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Cumplimiento de procesos de validación. *	¿La herramienta permite realizar casos de pruebas en función de los resultados esperados?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Cumplimiento de procesos de validación. *	¿La herramienta envía notificaciones automáticas para ejecución de pruebas?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Cumplimiento de procesos de validación. *	¿La herramienta envía	5=Completam.. 4=Casi todo.	

			notificaciones automáticas para ejecución de pruebas de regresión?	3= Med.. 2= Poco. 1= No.
		Cumplimiento de procesos de validación. *	¿La herramienta envía notificaciones automáticas para realizar análisis de pruebas estadísticas?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Cumplimiento de procesos de validación. *	¿La herramienta envía notificaciones automáticas para ejecución de pruebas de integración?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Cumplimiento de procesos de verificación. *	¿La herramienta permite definir criterios para realizar verificación?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Cumplimiento de procesos de verificación. *	¿La herramienta permite manejar cronogramas para las verificaciones?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Cumplimiento de procesos de verificación. *	¿La herramienta permite definir procedimientos de revisión?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Cumplimiento de procesos de verificación. *	¿La herramienta permite verificar la completitud de los entregables completados en cada una de las fases?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Cumplimiento de procesos de verificación. *	¿La herramienta envía notificaciones automáticas para el cumplimiento de todas las actividades definidas en cada una de las fases?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Cumplimiento de	¿La herramienta	5=Completam..

		procesos de verificación. *	permite verificar que la estructura de los documentos esté de acuerdo con los estándares establecidos?	4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de métricas de calidad. *	¿La herramienta permite definir métricas para medir complejidad lógica?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de métricas de calidad. *	¿La herramienta permite identificar el grado de granularidad?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de métricas de calidad. *	¿La herramienta permite definir métricas de evaluación de análisis de objetos, relaciones, diagramas?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de métricas de calidad. *	¿La herramienta permite definir métricas de especificación (número de requisitos funcionales)?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de métricas de calidad. *	¿La herramienta permite definir métricas de especificación (número de requisitos funcionales)?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de métricas de calidad. *	¿La herramienta permite definir métricas de diseño (complejidad de los datos, complejidad del sistema)?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de métricas de calidad. *	¿La herramienta permite definir métricas para medir aplicaciones desde la perspectiva del usuario: calidad y productividad de	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.

		los programas, beneficios de implementación, valor comercial del sistema?	
Realización de métricas de calidad. *	¿La herramienta permite definir métricas para medir el tamaño del procesamiento de la información?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Realización de métricas de calidad. *	¿La herramienta permite definir métricas para medir el tamaño la especificación de los requisitos del software?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Realización de métricas de calidad. *	¿La herramienta permite definir métricas para evaluar modelos técnicamente correctos?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Realización de métricas de calidad. *	¿La herramienta permite definir métricas para evaluar el grado en que los requisitos cumplen con las necesidades del usuario?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Realización de métricas de calidad. *	¿La herramienta permite definir métricas para medir la calidad de los entregables?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Documentación de reporte de errores *	¿La herramienta permite documentar los errores reportados?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Documentación de reporte de errores *	¿La herramienta permite generar reportes de errores?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Realización de auditorías.*	¿La herramienta ofrece alternativas para elegir el tipo de auditoría a	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco.	

		realizar?	1= No.
Realización de auditorias.*	¿La herramienta permite identificar el producto a auditar?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Realización de auditorias.*	¿La herramienta permite identificar los requisitos de la auditoria?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Realización de auditorias.*	¿La herramienta permite evaluar evidencias para determinar si el producto satisface los requerimientos?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Realización de auditorias.*	¿La herramienta permite corroborar la conformidad con los estándares establecidos mediante listas de chequeos?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Realización de auditorias.*	¿La herramienta permite verificar la evaluación del grado de cumplimiento de normas, políticas y procedimientos de la organización?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Realización de auditorias.*	¿La herramienta permite identificar riesgos ante debilidades significativas de control?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Realización de auditorias.*	¿La herramienta permite identificar revisiones del modelo de calidad adoptado?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Realización de auditorias.*	¿La herramienta envía notificaciones automáticas de auditorías internas?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.	
Realización de	¿La herramienta	5=Completam..	

		auditorías.*	realiza un documento en donde se defina el plan de auditoría (objeto, alcance, asistente, auditores, fecha de la auditoría?)	4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de auditorías.*	¿La herramienta permite realizar reportes posteriores a la auditoría en donde se establezcan acciones correctivas?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de auditorías.*	¿La herramienta permite verificar la existencia de manuales procedimientos?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de auditorías.*	¿La herramienta permite identificar la realización de auditorías sobre las políticas de calidad?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de auditorías.*	¿La herramienta envía notificaciones automáticas para realizar un cronograma de auditorías?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de auditorías.*	¿La herramienta permite definir los participantes de la auditoría?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de auditorías.*	¿La herramienta permite verificar que todos los ítems que forman parte de la configuración se consideren en las pruebas?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Realización de auditorías.*	¿La herramienta permite verificar el registro de las pruebas?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco.

				1= No.
		Realización de auditorias.*	¿La herramienta permite verificar que los procesos y desarrollo del trabajo se ajustan a los procedimientos especificados?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
	Control de la Calidad del Software	Ejecución de pruebas.*	¿La herramienta envía notificaciones automáticas para realizar las pruebas especificadas?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Ejecución de pruebas.*	¿La herramienta permite verificar la ejecución de pruebas de regresión?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
		Ejecución de pruebas.*	¿La herramienta permite llevar a cabo todo el proceso de gestión de calidad?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
FUN.2 Precisión USAD =3/10		Frecuencia de las pruebas	¿Cómo fue la frecuencia de las pruebas?	5= Semestral o más 4= Trimestral 3= Mensual 2= Semanal 1= Diario
		Criterios de validación	¿Existen criterios de validación para el producto?	5=Completam. def. 4=Casi todo def. 3= Med. Def. 2= Poco def. 1= No están def.
		Actividades de validación	¿Existen actividades de validación?	5=Completam. def. 4=Casi todo def. 3= Med. Def. 2= Poco def. 1= No están def.
FUN.3 Interoperabilidad USA = 1/6		Complejidad al pasar a una funcionalidad de otro sistema	¿Cómo es la complejidad al pasar a una funcionalidad en otro sistema?	5=Muy alta 4=Alta 3= Mediana 2= Básica 1= No tiene.
Usabilidad (USA)				
USA.1 Facilidad de comprensión USAD 3/5 =	Aseguramiento de la Calidad	¿Cuál es el tiempo en que un usuario adquiere las destrezas necesarias	¿Cuál es el tiempo en que un usuario adquiere las	5= 6 días o menos 4= 7-14 días

1MODF		para usar el sistema?	destrezas necesarias para usar el sistema?	3= 15-30 días 2= 2-5 meses 1= 6 meses o más
		Nivel de dificultad de la herramienta.**	¿Cuál es el nivel de dificultad de la herramienta?	5=Muy difícil 4=Difícil 3=Promedio 2=Fácil 1=Muy fácil
		Facilidad para ubicar funcionalidades	¿Es fácil ubicar las funcionalidades del sistema?	5= Muy fácilmente 4= Fácilmente 3= Dificultad media 2= Difícil 1= Muy difícil
USA.2 Capacidad de aprendizaje USA = 1/7	Aseguramiento de la Calidad	Calidad en la clasificación de los temas*	¿Es fácil utilizar el material de apoyo, por una buena clasificación de los temas?	5=Excelente 4=Buena 3= Promedio 2=Debajo del Prom. 1= Inaceptable.
USA.3 Interfaz Gráfica USA = 7/8 = 2 MODF	Aseguramiento de la Calidad	Interfaz personalizable de la herramienta.**	¿La interfaz de la herramienta es personalizable?	5=Completam.. 4=Casi todo. 3= Med.. 2= Poco. 1= No.
	Aseguramiento de la Calidad	Localización rápida de opciones en la herramienta.**	¿Se localizan rápidamente las opciones de la herramienta, es consistente la ubicación de utensilios?	5=Muy difícil 4=Difícil 3=Promedio 2=Fácil 1=Muy fácil
	Aseguramiento de la Calidad	Satisfacción del Diseño visual	¿Cómo es el diseño visual de las pantallas?	5=Excelente 4=Buena 3=Promedio 2=Debajo del Prom. 1= Inaceptable
	Aseguramiento de la Calidad	Versatilidad de la navegación	¿Cómo es la versatilidad de la navegación, entre pantallas?	5=Excelente 4=Buena 3=Promedio 2=Debajo del Prom. 1= Inaceptable
	Aseguramiento de la Calidad	Consistencia en el uso del color	¿Es Consistente el uso del color?	5=Excelente 4=Buena 3=Promedio 2=Debajo del

				Prom. 1= Inaceptable
	Aseguramiento de la Calidad	Contraste entre los colores	¿Cómo es el contraste con los colores?	5=Excelente 4=Buena 3=Promedio 2=Debajo del Prom. 1= Inaceptable
	Aseguramiento de la Calidad	Frecuencia de colores oscuros en fondos de pantalla (background) de texto para lectura.	¿Cómo es la frecuencia de colores oscuros en fondos de pantalla (background) de texto para lectura?	5=Excelente 4=Buena 3=Promedio 2=Debajo del Prom. 1= Inaceptable
Fiabilidad (FIA)				
FIA.2 Tolerancia a Fallas USA = 1/1	Aseguramiento de la Calidad	Anulación de operación incorrecta	¿Qué tan frecuente puede el producto de software evitar fallas, aún si el usuario lo opera incorrectamente?	5=Siempre 4=Casi siempre 3= Algunas veces 2= Pocas veces 1= Nunca
FIA.3 Recuperación USA = 4/4 = 1 MOD	Aseguramiento de la Calidad	Capacidad de reiniciar	¿Puede el producto de software recuperarse fácilmente después de una caída?	5=Muy fácilmente 4=Fácilmente 3=Dificultad Media 2= Difícil 1= Muy difícil
	Aseguramiento de la Calidad	Velocidad de reinicialización	¿Cuál es la velocidad de la reinicialización.	5=Muy rápida 4=Rápida 3= Promedio 2=Lenta 1= Muy lenta
	Aseguramiento de la Calidad	Disponibilidad del producto de software	¿Pueden los usuarios trabajar con el producto de software el tiempo necesario?	5=Siempre 4=Casi siempre 3= Algunas veces 2= Pocas veces 1= Nunca
	Aseguramiento de la Calidad	Existencia de procesos que disminuyan el tiempo de caída de la herramienta.**	¿Existen procesos que minimicen el tiempo de caída del producto de software?	5=Siempre 4=Casi siempre 3= Algunas veces 2= Pocas veces 1= Nunca

Fuente: Elaboración Propia, 2008

CAPÍTULO VI

ANÁLISIS DE RESULTADOS

Este capítulo tiene como objetivo exponer un análisis de los resultados obtenidos de la propuesta del Modelo de Gestión de Calidad del Software para la selección de herramientas de software, y sobre base de esta experiencia. Este análisis es de tipo cuantitativo dado la naturaleza de los resultados obtenidos. A continuación se presenta el resultado para cada una de las categorías seleccionadas en el capítulo V para el desarrollo de la propuesta del modelo.

Análisis de los Resultados Obtenidos – Categoría Funcionalidad

A continuación, se presenta en la Tabla N° 12 los resultados obtenidos para la categoría Funcionalidad.

Tabla N° 12
Análisis de Resultados – Funcionalidad

	Característica de Ajuste a los propósitos	Característica de Precisión	Característica de Interoperabilidad	Total
Cantidad de Métricas propuestas	17	0	0	17
Cantidad de Métricas modificadas Vs Cantidad de Métricas utilizadas de MOSCA	0	0	0	0
Cantidad de Métricas utilizadas Vs Cantidad Total de Métricas de MOSCA	0	3/10	1/6	4
Total	17	3	1	21

Fuente: Elaboración Propia, 2008

Como se puede observar en la Tabla N° 10, la mayor cantidad de métricas obtenidas fueron las métricas propuestas., representado por un total de diecisiete (17) métricas de las veintiún (21) métricas propuestas para esta categoría. Este resultado se refleja específicamente en la característica de ajustes a los propósitos con un total de diecisiete (17) métricas. Al utilizarlas para seleccionar herramientas del software, en base a la escala propuesta para cada una de ellas, es posible medir la capacidad de las herramientas del software para proveer funciones que cumplan con las necesidades específicas al ser utilizadas por los distintos usuarios, proveer resultados correctos y medir la capacidad de interacción con otros sistemas.

De igual forma la mayor cantidad de métricas utilizadas de MOSCA se encuentran en la característica de precisión, indicando así que aplicando tres (3) métricas de las diez (10) propuestas por MOSCA en la característica de precisión, es posible medir la capacidad de las herramientas del software para proveer los resultados correctos. Así mismo, para la característica de interoperabilidad es posible medir la capacidad de la herramienta para interactuar con otros sistemas aplicando una (1) métrica de las seis (6) propuestas por MOSCA.

Análisis de los Resultados Obtenidos – Categoría Usabilidad

A continuación, se presenta en la Tabla N° 13 los resultados obtenidos para la categoría Usabilidad.

Tabla N° 13
Análisis de Resultados – Usabilidad

	Característica de Facilidad de comprensión	Característica de Capacidad de aprendizaje	Característica de Interfaz Gráfica	Total
Cantidad de Métricas propuestas	0	0	0	0

Tabla N° 13 (Cont.)

	Característica de Facilidad de comprensión	Característica de Capacidad de aprendizaje	Característica de Interfaz Gráfica	Total
Cantidad de Métricas propuestas	0	0	0	0
Cantidad de Métricas modificadas Vs Cantidad de Métricas utilizadas de MOSCA	1/3	0	2/7	3
Cantidad de Métricas utilizadas Vs Cantidad Total de Métricas de MOSCA	3/5	3/5	7/8	13
Total	4	3	9	16

Fuente: Elaboración Propia, 2008

Como se muestra en la Tabla N° 13, la mayor cantidad de métricas obtenidas son las referentes a las métricas utilizadas del Modelo Sistémico de Calidad Mosca, la cual está representada por un total de trece métricas (13) de las dieciséis (16) métricas propuestas para esta categoría. Este resultado es producto del impacto de la cantidad de métricas utilizadas en la característica de interfaz gráfica, las cuales representan un total siete (7) métricas aunadas a las métricas utilizadas en las características de facilidad de comprensión y capacidad de aprendizaje, respectivamente. Este resultado indica que la aplicación de siete (7) métricas de las ocho (8) propuestas por MOSCA en la categoría del producto, específicamente, la característica de interfaz gráfica permite medir los atributos de las herramientas para conocer que tan atractivas son al usuario, en cuanto al uso del color y la naturaleza del diseño gráfico.

Por otra parte, se tiene que aplicando tres (3) métricas de las cinco (5) propuestas por MOSCA para la característica de facilidad de comprensión y facilidad de aprendizaje respectivamente, es posible conocer la capacidad que tienen las herramientas del software para facilitar al usuario el

entendimiento de la misma y la forma en que pueden ser utilizadas. Así como también, es posible medir la dificultad al entender las funcionalidades, operaciones y conceptos de las herramientas del software y la capacidad de habilitar al usuario el aprendizaje del uso de la herramienta.

A este resultado le sigue la cantidad de métricas modificadas, el cual está representado por un total de tres métricas (3) de las dieciséis (16) propuestas para esta categoría, las cuales fueron adaptadas para medir el nivel de dificultad de la herramienta, la interfaz y la localización de las opciones de la herramienta.

Análisis de los Resultados Obtenidos – Categoría Fiabilidad

A continuación, se presenta en la Tabla N° 14 los resultados obtenidos para la categoría Fiabilidad.

Tabla N° 14
Análisis de Resultados – Fiabilidad

	Característica de tolerancia a fallas	Característica de recuperación	Total
Cantidad de Métricas propuestas	0	0	0
Cantidad de Métricas modificadas Vs Cantidad de Métricas utilizadas de MOSCA	0	1/4	1
Cantidad de Métricas utilizadas Vs Cantidad Total de Métricas de MOSCA	1/1	4/4	5
Total	1	5	6

Fuente: Elaboración Propia, 2008

En la Tabla N° 14 se puede observar que la mayor cantidad de métricas obtenidas corresponden al renglón de métricas utilizadas del Modelo Mosca con una representación de cinco (5) métricas de seis (6) propuestas para

esta categoría. Este resultado se ve afectado directamente por la cantidad de métricas utilizadas para la característica de recuperación, en la cual de cuatro (4) métricas propuestas por MOSCA se utilizaron todas, lo que indica que mediante la aplicación de estas métricas es posible medir la capacidad de las herramientas del software para restablecer un nivel específico de rendimiento y recuperar datos afectados en caso de una falla. Así mismo, ocurre con la categoría de tolerancia a fallas, en la cual aplicando la única métrica propuesta por MOSCA para esta categoría es posible medir la capacidad de las herramientas para mantener un nivel de rendimiento específico en caso de errores de las mismas.

A este resultado le sigue la cantidad de métricas modificadas, el cual está representado por un total de una (1) de cuatro (4) métricas utilizadas para la categoría de recuperación en la cual se realizó una adaptación para medir la existencia de procesos que disminuyan el tiempo de caída de la herramienta.

A continuación, la Tabla N° 15 presenta un resumen de los resultados obtenidos para las tres categorías seleccionadas: Funcionalidad, Usabilidad y Fiabilidad.

Tabla N° 15
Resumen Análisis de Resultados

	Funcionalidad			Usabilidad				Fiabilidad		Total
	Característica de Ajuste a los Propósitos	Característica de Precisión	Característica de interoperabilidad	Característica de Facilidad de Comprensión	Característica de Capacidad de Aprendizaje	Característica de Interfaz Gráfica	Característica de Tolerancia a Fallos	Característica de Recuperación		
Cantidad de métricas propuestas	17	0	0	0	0	0	0	0	17	
Cantidad de Métricas modificadas Vs utilizadas de MOSCA Cantidad de Métricas	0	0	0	1/3	0	2/7	0	1/4	4	
Cantidad de Métricas utilizadas Vs Cantidad Total de Métricas de MOSCA	0	3/10	1/6	3/5	3/5	7/8	1/1	4/4	22	
Total	17	3	1	4	3	9	1	5	43	

Fuente: Elaboración Propia, 2008

Como se observa en la Tabla N° 15, la propuesta del Modelo de Gestión de Calidad del Software para seleccionar herramientas del software, se conforma por un total de cuarenta y tres (43) métricas que permiten especificar la calidad de las herramientas del software, en cuanto a la funcionalidad, usabilidad y fiabilidad de las mismas, facilitando la decisión para la selección adecuada y garantizando así el éxito de su implantación. Cada evaluación parcial aplicando la propuesta del modelo arrojará resultados que pueden ser utilizados por el decidor para mejorar su proceso de adopción: (1) herramientas del software que satisfacen los requerimientos, y (2) madurez de la organización para adoptar esta tecnología. Sin embargo, sólo los resultados que se obtienen de la aplicación integral del modelo reflejan la calidad de las herramientas de software que satisfacen sus requerimientos, brindando el apoyo a las organizaciones en la toma de decisiones para adoptar herramientas del software. Aunque es evidente que la decisión final queda a cargo de quien ejecuta el modelo.

Por otra parte, la propuesta del Modelo de Gestión de Calidad del Software para seleccionar herramientas del software permite aumentar el grado de confiabilidad del Modelo de Calidad Sistémico MOSCA para evaluar la calidad de productos, específicamente, herramientas de la gestión de calidad del software, brindando un soporte en los procesos inmersos de la gestión de la calidad, como los son: Planeación de la Calidad, Aseguramiento de la Calidad y Control de la Calidad, dichos procesos están incluidos en el área de la gestión de calidad para la gerencia de proyectos, garantizando que la calidad esté debidamente instaurada en un proyecto, específicamente, proyectos de desarrollos de software, y evitar así que el énfasis sea corregir insatisfacciones en lugar de prevenir la ocurrencia de lo indeseado y el resultado no corresponda con lo esperado. De esta manera, las organizaciones ofrecen altos niveles de calidad, manteniéndose competitivas en un mercado como el actual, siendo la calidad uno de los principales activos con los que cuenta un país para mejorar.

CAPÍTULO VII

CONCLUSIONES Y RECOMENDACIONES

Este capítulo tiene por finalidad, presentar las conclusiones y recomendaciones derivadas de la propuesta del Modelo de Gestión de Calidad para seleccionar Herramientas de Software.

Conclusiones

La propuesta del modelo desarrollado en este Trabajo de Investigación está fundamentada en una instanciación de modelo sistémico de calidad MOSCA., es decir, este modelo ayudó a proporcionar mayor documentación para soportar la propuesta del Modelo de Gestión de Calidad para seleccionar herramientas del software. Esta propuesta representa una forma más completa de obtener un ordenamiento para las herramientas del software, tomando en cuenta los requerimientos a cumplir. Además, amplía y refuerza el análisis de estas herramientas para apoyar de mejor forma la escogencia al adoptarlas.

También, se pudo lograr el cumplimiento de cada uno de los objetivos del presente trabajo lo que garantiza su completitud, y representa el alcance de la meta personal como investigadores, al lograr proponer un modelo que permite prestar apoyo a las organizaciones en la toma de decisiones para adoptar herramientas del software, ofreciendo una evaluación cuantitativa de las mismas mediante cálculos. Además, permite especificar la calidad de las herramientas del software, tomando en cuanto los procesos inmersos en la gestión de la calidad, como lo son la planeación de la calidad, el aseguramiento de la calidad y el control de la calidad, para así ir hacia una implantación efectiva de las mismas y hacia una gerencia de proyectos

efectiva al tomar en cuenta los tres procesos principales del área de gestión de calidad, representando esta área una de las nueve (9) áreas de conocimiento de la dirección de proyectos.

Por otra parte es innegable que constituye un método extremadamente útil para el desarrollo de proyectos, dado a que brinda soporte en la fase de inicio, planificación y control, específicamente, en el área de calidad.

Por último, se tiene que esta innovación en cuanto a la evaluación de herramientas de software permitirá a las empresas venezolanas un análisis integral para ayudar a mejorar la efectividad que se espera lograr con la adquisición de una herramienta de software, generando mayor satisfacción y ayuda más confiable en la compleja decisión de adopción de las mismas.

Recomendaciones

Una vez culminado con todo el desarrollo de la investigación, se hace necesario hacer las siguientes recomendaciones para futuros refinamientos.

- Refinar el modelo a través de su aplicación en herramientas del software disponibles en el mercado, de acuerdo a un criterio de clasificación de las herramientas.
- Definir un proceso que pueda apoyar a las organizaciones a aplicar el modelo propuesto, en el cual se contemple los requerimientos de las organizaciones, identificación de las herramientas, clasificación de las herramientas y fuentes de adquisición de las herramientas.
- Proponer un modelo que permite evaluar a los proveedores de la(s) herramienta(s) del software seleccionada(s), ya que una vez determinada cual(es) herramienta(s) por sus particularidades es la mejor alternativa, se evalúan las características de los proveedores de la misma, de esta manera es posible considerar varios proveedores a la vez (si los hay) y una baja o alta calificación de alguno de ellos influencia sobre la selección de la herramienta.

REFERENCIAS BIBLIOGRÁFICAS

ACM. Software Engineering Code of Ethics and Professional Practice. Fuente disponible: <http://www.acm.org/serving/se/code.htm>

Álvarez, A. (2000). *Modelo de Evaluación de la Calidad del Proceso de Desarrollo de Sistemas de Información*. Trabajo Especial de Grado presentado y publicado en la Universidad Simón Bolívar, Caracas.

Alter, S (2002). *Information systems. The foundation of e-business*. (4th. ed.). USA: Prentice Hall

Asociación Latinoamericana del QFD. (2008). ¿Qué es el QFD?. Tomado el 30 de Enero, 2008. Publicado en: http://www.qfdlat.com/_Que_es_el_QFD_/que_es_el_qfd_.html.

Baskerville, R. (1999). Communications of the AIS. *Investigating information system with action research.*, Vol .2, art. 19 Octubre.

Bruckhaus, T.; Madhavji, N.; Janssen, I.; Henshaw, J. (1996). *The impact of tool on software productivity*. *IEEE Software* .Sep, 1996.

Callaos, N. (1996). *Sistemas de Información: Conceptos y Definiciones*. Venezuela.

Chekland, P. (1993). *Pensamientos de sistemas, prácticas de sistemas*. (Grupo Noriega Editores).

Código Ético y Deontológico para Ingenieros en Informática. Fuente disponible: http://www.coiie.org/pdf/codigo_deontologico.pdf

De Luca, L. (2001). *Indicadores para la selección de Herramientas que Soportan la Gerencia de la Calidad del Software*. Trabajo Especial de Grado presentado y publicado en la Universidad Simón Bolívar, Caracas.

Evered, R y Susman, G. (1978). *Administrative Science Quaterly* (23) 4 pp. 582-603. *And assessment of the scientific merits of action research*.

Fuggetta , A. (1993). *A classification of CASE technology*. *Computer. Publication*, 26 (12), , 25-38.

- Dromey, G. (1994). *A Model for Software Product Quality*. Publicado en: <http://www.sqi.gu.edu.au/>
- Gillies, A. (1997). *Software Quality: Theory and management*. (Segunda Edición).
- Hoffer, J; George, J. y Valacich, J. *Modern Systems Analysis and Design*. Benjamin/Cummings Publishing Company, inc. 1996.
- Herman et al. (1995). Horacio Helmas; Pereira Paulo. Manual AMEF. (Primera Edición). Brasil.
- (Hordago, 2007). Hordago. AMFE: Análisis Modal de Fallos y Efectos. (Primera Edición). USA.
- IEEE (2004). *Guide to the software engineering body of knowledge SWEBOK. A project of the IEEE Computer Society Professional Practices Committee*. USA: IEEE computer Society
- Kavi y Nahoraii, (1996). Software tool Assessment IEEE Software.Sep, 1996.
- Kendall; Kendall (2002).. *Systems analysis and design* (5th ed.) USA: Prentice Hall .
- Kitchenham B.. (1996). *Evaluating Software Engineering Methods and Tools*. *Software Engineering Notes*:
 Part 1: The Evaluation Context and Evaluation Methods. 21(1), 11-15, 1996.
 Part 2: Selecting an Appropriate Evaluation Method - Technical Criteria. 21(2), 11-15, 1996.
 Part 3: Selecting an Appropriate Evaluation Method - Practical Issues. 21(4), 9-12, 1996.
- Martínez, J. (2001). *MOSCA Modelo Sistémico de Calidad(Integración del Modelo de Calidad de Software y el Modelo de Calidad del Proceso de Desarrollo con un Enfoque Sistémico)*. Trabajo Especial de Grado presentado y publicado en la Universidad Simón Bolívar, Caracas.
- Mendoza, L (1999), *Indicadores organizacionales para evaluar herramientas CASE en organizaciones venezolanas*. (Universidad Simón Bolívar).
- Monsalve, L. (1998). Calidad de los Productos de Software. *Revista Ingeniería Informática*. Disponible en:<http://sol.inf.udec.cl/revista/edicion1-1/monsalve.html>
- O'Brien, (1999). *Introduction to Information Systems*. 9th Edition, Erwin 1997.

- O'Connor, D. (1998). *ISO 9126.A General Quality Model*. Department of Physical and Quantitative Sciences in Waterford Institute of Technology. Disponible en: <http://emhain.wit.ie/doconnor/lectures/se3/metrics/iso9126/>
- Ortega, M. (2001). *Modelo de Calidad del Producto de Software con un Enfoque Sistémico*. Trabajo Especial de Grado presentado y publicado en la Universidad Simón Bolívar, Caracas.
- Pérez, M., (2005). *Modelo Sistémico de Calidad, MOSCA (Certificación de Calidad)*. Publicado en: <http://www.lisi.usb.ve/publicaciones>.
- Pérez, M; Rojas, T; Mendoza, L y Grimán, A (2001) . *Systemic Quality Model for System Development Process: Case Study*. Publicado en: <http://www.lisi.usb.ve/publicaciones>.
- PMBOK (2004). *Guía de los fundamentos de la dirección de proyectos*. (Tercera Edición). EEUU.
- Pressman, R. (2006). *Ingeniería de Software: Un Enfoque Práctico*. (Sexta Edición). España.
- Rivas, L., Pérez, M., Mendoza, L. & Grimán, A. (2007). *Herramientas de Desarrollo de Software: Hacia la Construcción de una Ontología*. Encuentro Venezolano sobre Tecnologías de la Información e Ingeniería del Software, EVETIS 2007, Margarita, Venezuela.
- Rojas, T, Pérez, M & Grimán, A (1999). *Modelo de Decisión para soportar la selección de herramientas CASE*.
- Pérez, M y T. Rojas (1995) , *Improvement in the development of information systems by increasing its process effectiveness*.(Universidad Simón Bolívar).
- SWEBOK(2004). IEEE. *Guide to the software engineering body of knowledge SWEBOK. A project of the IEEE Computer Society Professional Practices Committee*. USA: IEEE computer Society (2004)
- Somerville (2002). *Ingeniería del Software (Sexta Edición)* México.
- Tague, Nancy, R. (1995) *The Quality Toolbox*. EUU
- Tamayo, M. (1997). *El Proceso de la Investigación Científica*. (Tercera Edición). México.
- Topper, A.; Ouellette, D.; Jorgensen, P. *Structure Methods Mergin Models, Techniques an CASE* Mc Graw Hill, 1994.

Unhekar, B (2003). Process Quality Assurance for UML – Based Projects. (Primera Edición). USA.

(UVM, 2006). Universidad del Valle de México. Análisis Modo Efecto de Fallas. Tomado el 21 de Febrero, publicado en: <http://digitalia.uvmnet.edu/portalestudiantil/memorias/iis/disexp/analisismodoefectofallas16-02-06.doc>.

APÉNDICES

APÉNDICE I

MODELO SISTÉMICO DE CALIDAD (MOSCA)

Este Modelo integra tanto la calidad del producto de software como la calidad del proceso de desarrollo con un enfoque sistémico, tomando en cuenta todas las variables de calidad que están presentes en una organización desarrolladora de Sistemas de Software. A continuación se presenta en detalle cada uno de los 4 niveles que conforman el Modelo Sistémico de Calidad (MOSCA).

3.1 Nivel 0: Dimensiones

Eficiencia del proceso, Efectividad del proceso, Eficiencia del producto y Efectividad del producto son las cuatro dimensiones propuestas en el prototipo de modelo. Solo una buena interrelación entre ellas permitirá garantizar la calidad Sistémica global de una organización.

3.2 Nivel 1: Categorías

Se contemplan 11 categorías: 6 pertenecientes al producto y las otras 5 al proceso de desarrollo. Esta división no implica un desligamiento entre ellas, simplemente se realiza para identificar a que sector o sub-modelo pertenecen.

3.2.1 Producto (Ortega, 2000):

- Funcionalidad (FUN)
- Fiabilidad (FIA)
- Usabilidad (USA)
- Eficiencia (EFI)
- Mantenibilidad (MAB)
- Portabilidad

3.2.2 Proceso (Álvarez, 2000):

- Cliente - Proveedor (CUS)
- Ingeniería (ENG)
- Soporte (SUP)
- Gestión (MAN)
- Organizacional (ORG)

3.3 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.

Entre las características asociadas a cada categoría del producto, aparecen, en el modelo MOSCA, una serie de características del proceso. Lo mismo ocurre con cada una de las categorías del 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.

En el caso de la calidad del producto, si una vez medidas las características asociadas a una categoría en particular, resulta arrojar resultados defectuosos, entonces, se deben analizar las características de la calidad del proceso asociadas a esa categoría del producto para encontrar las causas del porque no cumplió con todos los requerimientos solicitados.

En caso de que las características del proceso asociadas a dicha categoría del producto hayan sido cumplidas a cabalidad, ocasionará entonces, que el resultado de la medición de la misma mejore notablemente.

Lo mismo ocurre midiendo la calidad del proceso. Si las mediciones realizadas sobre las características asociadas a una categoría en particular del sub-modelo del proceso, resultan arrojar resultados defectuosos, se debe analizar las características de la calidad del producto asociadas a esa

categoría del proceso para ver el impacto que se generó sobre ellas.

En el caso de que las características de calidad del producto asociada a dicha categoría del proceso, cumplan con todos los requerimientos, entonces, el resultado de la medición de esa categoría del sub-modelo del proceso mejorará.

En el siguiente punto, donde se describen las categorías en detalle, se explica la relación que tienen algunas características del producto con las categorías del proceso y viceversa.

A continuación se muestra en la tabla N° 1, todas las características junto con una clave única que las identifica.

Tabla N° 1
Características del Prototipo de Modelo Final

PROCESO	
Categoría	Características
Cliente-Proveedor	CUS.1 Proceso de Adquisición del Sistema o producto de Software
	CUS.2 Proceso de Suministro
	CUS.3 Proceso de Elicitación de Requerimientos
	CUS.4 Proceso de Operación
Ingeniería	ENG.1 Proceso de Mantenimiento de software y Sistemas
	ENG.2 Proceso de Desarrollo
Soporte	SUP.1 Proceso de Documentación
	SUP.2 Proceso de Gestión de Configuración
	SUP.3 Proceso de Aseguramiento de la Calidad
	SUP.4 Proceso de Verificación
	SUP.5 Proceso de Validación
	SUP.6 Proceso de Revisión Conjunta
	SUP.7 Proceso de Auditoria
	SUP.8 Proceso de Resolución de Problemas
Gestión	MAN.1 Proceso de Gestión
	MAN.2 Proceso de Gestión de Proyecto
	MAN.3 Proceso de Gestión de Calidad
	MAN.4 Proceso de Gestión del Riesgo
Organizacional	ORG.1 Proceso de Alineación Organizacional
	ORG.2 Proceso de Gestión de Cambio
	ORG.3 Proceso de Establecimiento del Proceso
	ORG.4 Proceso de Evaluación del Proceso
	ORG.5 Proceso de Mejoramiento
	ORG.6 Proceso de Gestión de RRHH
	ORG.7 Proceso de Infraestructura
	ORG.8 Proceso de Medición
	ORG.9 Proceso de Reuso

PRODUCTO	
Categoría	Características
Funcionalidad	FUN.1 Ajuste a los propósitos
	FUN.2 Precisión
	FUN.3 Interoperabilidad
	FUN.4 Seguridad
	FUN.5 Correctitud
	FUN.6 Estructurado
	FUN.7 Encapsulado
	FUN.8 Especificado
Fiabilidad	FIA.1 Madurez
	FIA.2 Tolerancia a Fallas
	FIA.3 Recuperación
	FIA.4 Correctitud
	FIA.5 Estructurado
	FIA.6 Encapsulado
Usabilidad	USA.1 Facilidad de comprensión
	USA.2 Capacidad de aprendizaje
	USA.3 Interfaz Grafica
	USA.4 Operabilidad
	USA.5 Conformidad con los estándares
	USA.6 Completo
	USA.7 Consistente
	USA.8 Efectivo
	USA.9 Especificado
	USA.10 Documentado
	USA.11 Auto-Descriptivo
Eficiencia	EFI.1 Comportamiento del tiempo
	EFI.2 Utilización de recursos
	EFI.3 Efectivo
	EFI.4 No-redundante
	EFI.5 Directo
	EFI.6 Utilizado
Mantenibilidad	MAB.1 Capacidad de análisis
	MAB.2 Capacidad de cambio
	MAB.3 Estabilidad
	MAB.4 Capacidad de prueba
	MAB.5 Acoplamiento
	MAB.6 Cohesión
	MAB.7 Encapsulamiento
	MAB.8 Atributos de Madurez del Software
	MAB.9 Sistemas de Estructura de Control
	MAB.10 Sistema de Estructura de Información
	MAB.11 Descriptivo
	MAB.12 Correctitud
	MAB.13 Estructural
Portabilidad	POR.1 Adaptabilidad
	POR.2 Capacidad de instalación
	POR.3 Co-existencia
	POR.4 Capacidad de reemplazo
	POR.5 Consistencia
	POR.6 Parametrizado
	POR.7 Encapsulado
	POR.8 Cohesivo
	POR.9 Especificado

	POR.10 Documentado
	POR.11 Auto-descriptivo
	POR.12 No-redundante

3.4 Nivel 3: Métricas

Cada característica posee una serie de métricas utilizadas para medir la calidad sistémica de una organización.

A continuación se explicarán cada una de las categorías con sus respectivas características agrupadas por dimensión. Las métricas asociadas a cada característica se encuentran detalladas al final del capítulo.

3.4.1 Funcionalidad (FUN):

Es la capacidad del producto del software para proveer funciones que cumplan con necesidades específicas o implícitas, cuando el software es utilizado bajo ciertas condiciones. La funcionalidad toma en cuenta el ajuste a los propósitos, la precisión, la interoperabilidad, y la seguridad del producto de software. La tabla N° 2 muestra las características de la Funcionalidad agrupadas por dimensión.

Tabla N° 2

Características de la Funcionalidad Agrupadas por Dimensión

EFFECTIVIDAD DEL PRODUCTO	EFICIENCIA DEL PRODUCTO
FUN.1 Ajuste a los propósitos	FUN.5 Correctitud
FUN.2 Precisión	FUN.6 Estructurado
FUN.3 Interoperabilidad	FUN.7 Encapsulado
FUN.4 Seguridad	FUN.8 Especificado

Fuente: Adaptado de Ortega, 2000

FUN.1 Ajuste a los propósitos: Es la capacidad del producto de software

para proveer un conjunto de funciones apropiado según tareas y objetivos específicos del usuario.

FUN.2 Precisión: Es la capacidad del producto de software para proveer los resultados correctos. Esto incluye el grado de precisión de los valores calculados.

FUN.3 Interoperabilidad: Es la capacidad del producto de software para interactuar con uno o más sistemas.

FUN.4 Seguridad: Es la capacidad del producto de software para proteger información y datos de manera que las personas que no estén autorizadas no puedan leerlos o modificarlos y a las personas o sistemas autorizados se les permita el acceso a los mismos.

FUN.5 Correctitud: Está dividida en tres propiedades relacionadas con la capacidad de cómputo, completitud y consistencia. Alguna violación de una de estas propiedades puede significar que el software no tendrá la funcionalidad esperada.

FUN.6 Estructurado: Una forma estructural tiene la propiedad de ser estructurada si sigue las reglas de la programación estructurada.

FUN.7 Encapsulado: El concepto de encapsulamiento se refiere a que las variables, las constantes y los tipos deben ser utilizados sólo en el contexto en el que son definidos. La manera en que son utilizadas puede tener un impacto significativo sobre la modularidad y por lo tanto sobre la calidad de los módulos y programas.

FUN.8 Especificado: Un módulo, programa u otra forma estructural están especificados si su funcionalidad es descrita con pre-condiciones y post-condiciones.

3.4.2 Fiabilidad (FIA):

La fiabilidad es la capacidad del producto de software para mantener un nivel especificado de rendimiento cuando es utilizado bajo condiciones especificadas. La tabla N° 3 muestra las características asociadas a la

Fiabilidad agrupadas por dimensión.

Tabla N° 3.

Características de la Fiabilidad Agrupadas por Dimensión

EFFECTIVIDAD DEL PRODUCTO	EFICIENCIA DEL PRODUCTO
FIA.1 Madurez FIA.2 Tolerancia a fallas FIA.3 Recuperación	FIA.4 Correctitud FIA.5 Estructurado FIA.6 Encapsulado

Fuente: Adaptado de Ortega, 2000

FIA.1 Madurez: Es la capacidad del producto de software para evitar fallas como resultado de errores en el mismo.

FIA.2 Tolerancia a fallas: Es la capacidad del producto de software para mantener un nivel de rendimiento específico en caso de errores en el software o de infracciones sobre sus interfaces.

FIA.3 Recuperación: Es la capacidad del producto de software para restablecer un nivel especificado de rendimiento y recuperar los datos afectados en el caso de una falla. Después de una falla, el producto de software algunas veces se encuentra no operativo por un período de tiempo, dicho lapso de tiempo es medido como su recuperación. Una definición asociada a la recuperación es la *Disponibilidad* que consiste en la capacidad de un producto de software para estar en estado operativo en el momento de ejecutar una función en un momento determinado, bajo condiciones específicas.

FIA.4 Correctitud: Esta propiedad ya fue considerada y explicada en la característica de funcionalidad. Esta propiedad afecta tanto a la funcionalidad como a la fiabilidad.

FIA.5 Estructurado: Igual que la propiedad anterior, fue considerada en la funcionalidad.

FIA.6 Encapsulado: También fue considerada en la funcionalidad y afecta de la misma manera sobre la fiabilidad.

3.4.3 Usabilidad(USA):

Esta categoría se refiere a la capacidad del producto de software para ser atractivo, entendido, aprendido y utilizado por el usuario bajo condiciones específicas. La tabla N° 4, muestra las características asociadas a la usabilidad agrupadas por dimensión.

Tabla N° 4

Características de la Usabilidad Agrupadas por Dimensión

EFFECTIVIDAD DEL PRODUCTO	EFICIENCIA DEL PRODUCTO
USA.1 Facilidad de comprensión USA.2 Capacidad de Aprendizaje USA.3 Interfaz Gráfica USA.4 Operabilidad USA.5 Conformidad con los estándares	USA.6 Completo USA.7 Consistente USA.8 Efectivo USA.9 Especificado USA.10 Documentado USA.11 Auto-descriptivo

Fuente: Adaptado de Ortega, 2000

USA.1 Facilidad de comprensión: Es la capacidad que tiene el producto de software para facilitar al usuario el entendimiento del software y la forma en que puede ser utilizado. Las métricas asociadas a esta categoría deben ser capaces de evaluar el comportamiento de los usuarios sin previo conocimiento de la operación del software y medir la dificultad al entender las funciones, operaciones y conceptos del software. Pueden ser consideradas entidades como documentación (en todos los formatos disponibles, en línea o impreso), interfaces de software y vocabulario.

USA.2 Capacidad de Aprendizaje: Es la capacidad del producto de

software para habilitar al usuario el aprendizaje de la aplicación.

USA.3 Interfaz Grafica: Este concepto está asociado los atributos del software que lo hacen más atractivo al usuario, tales como el uso del color y la naturaleza del diseño gráfico.

USA.4 Operabilidad: Es la capacidad del producto de software para habilitar al usuario a operarlo y controlarlo.

USA.5 Conformidad con los estándares:

USA.6 Completo: Una forma estructural es completa si tiene todos los elementos necesarios para definir e implementar la forma estructural.

USA.7 Consistente: Una forma estructural es consistente si su uso mantiene sus propiedades o funcionalidad y si todos sus elementos contribuyen a reforzar su intención o efecto.

USA.8 Efectivo: Una forma estructural es efectiva cuando presenta sólo los elementos necesarios para definir e implementar la forma estructural.

USA.9 Especificado: Un módulo, programa u otra forma estructural están especificados si su funcionalidad es descrita con pre-condiciones y post-condiciones.

USA.10 Documentado: Una forma estructural está documentada cuando su propósito y propiedades están explícitamente definidas en el contexto de la forma estructural. Aplicable a módulos, ciclos, estructuras de datos, variables, constantes, tipos, etc.

USA.11 Auto-descriptivo: Una forma estructural es auto-descriptiva si su propósito es evidente en los nombres de los módulos y los identificadores tienen significado en el contexto de la aplicación. Es aplicado a módulos, variables, constantes, estructuras de datos, etc.

3.4.4 Eficiencia (EFI):

Es la capacidad del producto de software para proveer un rendimiento apropiado, relativo a la cantidad de recursos utilizados, bajo condiciones específicas. La tabla N° 5, muestra las características asociadas a la

Eficiencia agrupadas por dimensión.

Tabla N° 5

Características de la Eficiencia Agrupadas por Dimensión

EFFECTIVIDAD DEL PRODUCTO	EFICIENCIA DEL PRODUCTO
EFI.1 Comportamiento del tiempo EFI.2 Utilización de recursos	EFI.3 Efectivo EFI.4 No-redundante EFI.5 Directo EFI.6 Utilizado

Fuente: Adaptado de Ortega, 2000

EFI.1 Comportamiento del tiempo: Es la capacidad del producto de software para proveer respuestas y tiempos de procesamiento apropiados en tiempo de ejecución bajo condiciones específicas.

EFI.2 Utilización de recursos: Es la capacidad del producto de software para utilizar cantidades apropiadas de los recursos cuando el mismo ejecuta sus funciones bajo condiciones específicas.

EFI.3 Efectivo: Una forma estructural es efectiva cuando presenta sólo los elementos necesarios para definir e implementar la forma estructural.

EFI.4 No-redundante: Una forma estructural es No-redundante cuando tiene sólo los elementos lógicos necesarios para definir una forma estructural. Esta propiedad se diferencia de la Efectividad en que la No-redundancia involucra redundancia lógica, en vez de redundancia computacional. En otras palabras, aplica a las condiciones lógicas y no a las instrucciones de asignación.

EFI.5 Directo: Una expresión es directa cuando la abstracción, representación, y la estructura del cálculo es congruente con el problema original. Una forma indirecta de representación hace que sea más difícil la comprensión. Aplicado a instrucciones, expresiones, variables, constantes, tipos.

EFI.6 Utilizado: Una forma estructural es utilizada si ha sido definida y luego utilizada en su alcance.

3.4.5 Mantenibilidad(MAB):

La mantenibilidad es una de las categorías más importantes a ser medidas en un producto de software porque es la capacidad del mismo para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptaciones del software ante cambios del ambiente, en requerimientos y especificaciones funcionales. La tabla N° 6 muestra las características asociadas a la mantenibilidad agrupadas por dimensión.

Tabla N° 6

Características de la Mantenibilidad Agrupadas por Dimensión

EFFECTIVIDAD DEL PRODUCTO	EFICIENCIA DEL PRODUCTO
MAB.1 Capacidad de análisis	MAB.5 Acoplamiento
MAB.2 Capacidad de Cambio	MAB.6 Cohesión
MAB.3 Estabilidad	MAB.7 Encapsulamiento
MAB.4 Capacidad de prueba	MAB.8 Atributos de Madurez del Software
	MAB.9 Sistema de Estructura de Control
	MAB.10 Sistema de Estructura de Información
	MAB.11 Descriptivo
	MAB.12 Correctitud
	MAB.13 Estructural

Fuente: Adaptado de Ortega, 2000

MAB.1 Capacidad de análisis: Es la capacidad del producto de software para ser diagnosticado por deficiencias o fallas en el software.

MAB.2 Capacidad de Cambio: Es la capacidad del producto de software para facilitar una modificación específica a ser implementada.

MAB.3 Estabilidad: Es la capacidad del producto de software para evitar efectos inesperados después de modificaciones en el software.

MAB.4 Capacidad de prueba: Es la capacidad del producto de software para permitir que el software modificado sea validado.

MAB.5 Acoplamiento: Es una métrica que mide la interconexión entre los módulos de una estructura de programa. El acoplamiento depende de la complejidad de la interfaz entre los módulos, el punto en el que se entra o se hace referencia al módulo y qué datos pasan a través de la interfaz.

MAB.6 Cohesión: Una forma estructural es cohesiva si todos sus elementos están enlazados uno a los otros y contribuyen a llevar a cabo un simple objetivo o función.

MAB.7 Encapsulamiento: Las variables, las constantes y los tipos deben ser utilizados sólo en el contexto en el que son definidos. La manera en que son utilizadas puede tener un impacto significativo sobre la modularidad y por lo tanto sobre la calidad de los módulos y programas.

MAB.8 Atributos de Madurez del Software: Son las características físicas y las medidas asociadas a la edad y uso del sistema de software objetivo.

MAB.9 Sistema de Estructura de Control: Son las propiedades relacionadas a la modularidad, los atributos de control inter-modular, la selección y uso de construcciones de flujo de control, la manera en la cual el programa o sistema es descompuesto en algoritmos, y el método con el cual los algoritmos son implementados.

MAB.10 Sistema de Estructura de Información: Son aquellas propiedades asociadas al almacenamiento y flujo de información en un programa o sistema, incluyendo definiciones de datos y las características de las entradas y salidas del sistema.

MAB.11 Descriptivo: Para que un software sea útil, debe ser fácil de entender y usar. A continuación se muestra una clasificación de estas propiedades para ser utilizadas en la evaluación de la eficiencia del producto.

Esta clasificación incluye: Sistema de tipografía, nombres y comentarios; Componentes de tipografía, nombre y comentarios; y Abstracción de la documentación de soporte.

MAB.12 Correctitud: Esta característica ya fue considerada y explicada en la característica de funcionalidad. Esta propiedad afecta tanto a la funcionalidad como a la fiabilidad.

MAB.13 Estructural: Igual que la propiedad anterior, fue considerada en la funcionalidad

3.4.6 Portabilidad(POR):

La portabilidad es la capacidad del producto de software para ser transferido de un ambiente a otro. La tabla N° 7 muestra las características asociadas a la Portabilidad agrupadas por dimensión.

Tabla N° 7

Características de la Mantenibilidad Agrupadas por Dimensión

EFFECTIVIDAD DEL PRODUCTO	EFICIENCIA DEL PRODUCTO
POR.1 Adaptabilidad POR.2 Capacidad de Instalación POR.3 Co-existencia POR.4 Capacidad de remplazo	POR.5 Consistente POR.6 Parametrizado POR.7 Encapsulado POR.8 Cohesivo POR.9 Especificado POR.10 Documentado POR.11 Auto-descriptivo POR.12 No-redundante

Fuente: Adaptado de Ortega, 2000

POR.1 Adaptabilidad: Es la capacidad del producto de software para ser adaptado a diferentes ambientes especificados sin aplicar acciones u otros medios que no sean los provistos para este propósito en el software considerado.

POR.2 Capacidad de Instalación: Es la capacidad del producto de software para ser instalado en un ambiente especificado.

POR.3 Co-existencia: La Co-existencia es la capacidad del producto de software para co-existir con otro software independiente en un ambiente común compartiendo recursos comunes.

POR.4 Capacidad de remplazo: Es la capacidad del producto de software para ser usado en lugar de otro producto de software especificado para el mismo propósito en un mismo ambiente. Por ejemplo, la capacidad para el reemplazo de una nueva versión de un producto es importante para el usuario, cuando ésta se actualiza.

POR.5 Consistente: Esta propiedad ya fue considerada y explicada en la característica de usabilidad. Esta propiedad afecta tanto a la usabilidad como a la portabilidad.

POR.6 Parametrizado: Un módulo está parametrizado si contiene sólo los parámetros necesarios para caracterizar una función / procedimiento.

POR.7 Encapsulado: Esta propiedad ya fue considerada y explicada en la característica de funcionalidad. Esta propiedad afecta tanto a la funcionalidad como a la portabilidad.

POR.8 Cohesivo: Una forma estructural es cohesiva si todos sus elementos están enlazados uno a los otros y contribuyen a llevar a cabo un simple objetivo o función. Las instrucciones en un componente cohesivo deben estar organizadas a partir de la menos a la más dependiente, es decir, la última instrucción, en una secuencia depende de todas sus predecesoras. Cualquier instrucción independiente, va en contra de la cohesión. Lo importante es lograr una alta cohesión y reconocer cuando hay poca cohesión para modificar el diseño del software y conseguir una mayor independencia funcional

POR.9 Especificado: Esta propiedad ya fue considerada también y explicada en la característica de funcionalidad. Esta propiedad afecta tanto a la funcionalidad como a la portabilidad.

POR.10 Documentado: Esta propiedad ya fue considerada y explicada en la característica de usabilidad. Esta propiedad afecta tanto a la usabilidad como a la portabilidad.

POR.11 Auto-descriptivo: Esta propiedad, como la anterior, ya fue considerada y explicada en la característica de usabilidad. Esta propiedad afecta tanto a la usabilidad como a la portabilidad.

POR.12 No-redundante: Esta propiedad ya fue considerada y explicada en la característica de eficiencia. Esta propiedad afecta tanto a la eficiencia como a la portabilidad.

Todas las categorías descritas anteriormente están asociadas a la calidad del producto. A continuación se explican detalladamente las categorías pertinentes a la calidad del proceso.

3.4.7 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. La tabla N° 8 muestra las características asociadas a la categoría Cliente-Proveedor agrupadas por dimensión.

Tabla N° 8.

Características de la Categoría Cliente-Proveedor Agrupadas por Dimensión

EFFECTIVIDAD DEL PROCESO	EFICIENCIA DEL PROCESO
CUS.1 Proceso de Adquisición del Sistema o producto de Software CUS.3 Proceso de Licitación de Requerimientos	CUS.2 Proceso de Suministro

Fuente: Elaboración Propia

CUS.1 Proceso de Adquisición del Sistema o producto de Software: Obtener un producto y/o servicio que satisfaga las necesidades del cliente garantizando la aceptación del mismo.

CUS.2 Proceso de Suministro: Proveer un sistema o software al cliente que satisfaga los requerimientos acordados.

CUS.3 Proceso de Elicitación de Requerimientos: Reunir, procesar y monitorear la evolución de las necesidades y requerimientos del cliente a través de la vida del producto y/o servicio.

CUS.4 Proceso de Operación: Asegurar la correctitud y eficiencia de la operación del software durante el uso al que fue destinado y en el ambiente donde fue instalado.

3.4.8 Ingeniería (ENG):

Consiste en procesos que directamente especifican, implementan o mantienen el producto de software, su relación con el Sistema y su documentación. La tabla N° .9 muestra las características asociadas a la categoría Ingeniería agrupadas por dimensión.

Tabla N° 9

Características de la Categoría Ingeniería Agrupadas por Dimensión

EFFECTIVIDAD DEL PROCESO	EFICIENCIA DEL PROCESO
ENG.1 Proceso de Mantenimiento de software y Sistemas	ENG.2 Proceso de Desarrollo

Fuente: Elaboración Propia

ENG.1 Proceso de Desarrollo: se refiere al conjunto de requerimientos de un producto de software funcional o sistema basado en software que

deben ser transformados para que se satisfagan las necesidades expresadas por el cliente.

ENG.2 Proceso de Mantenimiento de software y Sistemas: se refiere a la gestión de la modificación, migración y retiro de determinados componentes del sistema en respuesta a una petición del cliente.

3.4.9 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. La tabla N° 10 muestra las características asociadas a la categoría de Soporte agrupadas por dimensión.

Tabla N° 10

Características de la Categoría Soporte Agrupadas por Dimensión

EFFECTIVIDAD DEL PROCESO	EFICIENCIA DEL PROCESO
SUP.3 Proceso de Aseguramiento de la Calidad	SUP.1 Proceso de Documentación
SUP.6 Proceso de Revisión Conjunta	SUP.2 Proceso de Gestión de Configuración
SUP.7 Proceso de Auditoria	SUP.4 Proceso de Verificación
SUP.8 Proceso de Resolución de Problemas	SUP.5 Proceso de Validación
	SUP.6 Proceso de Revisión Conjunta
	SUP.7 Proceso de Auditoria
	SUP.8 Proceso de Resolución de Problemas

Fuente: Elaboración Propia

SUP.1 Proceso de Documentación: Esta basado en el desarrollo y mantenimiento de documentos que registran información producida por un proceso o actividad.

SUP.2 Proceso de Gestión de Configuración: Se basa en establecer y mantener la integridad de todos los productos elaborados en un proceso o actividad.

SUP.3 Proceso de Aseguramiento de la Calidad: Se debe proporcionar seguridad con el objeto de que los procesos de un proyecto cumplan con los requerimientos específicos y se adhiera a los planes establecidos.

SUP.4 Proceso de Verificación: Confirmar que cada producto trabajado y/o servicio de un proceso o proyecto refleje apropiadamente los requerimientos especificados.

SUP.5 Proceso de Validación: Se deben confirmar que los requerimientos para uso específico del producto del sistema (software) sean satisfechos.

SUP.6 Proceso de Revisión Conjunta: 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.

SUP.7 Proceso de Auditoria: Determinar independientemente la complacencia de los productos seleccionados y los procesos con requerimientos, planes y contratos pertinentes.

SUP.8 Proceso de Resolución de Problemas: Se debe asegurar que todos los problemas descubiertos sean analizados y solucionados.

3.4.10 Gestión (MAN):

Consiste en procesos que contienen prácticas de naturaleza genérica, que pueden ser utilizadas por cualquier personaje que dirija algún tipo de proyecto o proceso, dentro de un Ciclo de Vida de Primario. La tabla N° 11 muestra las características asociadas a la categoría de Gestión agrupadas por dimensión.

Tabla N° 11

Características de la Categoría Gestión Agrupadas por Dimensión

MAN.1 Proceso de Gestión MAN.3 Proceso de Gestión de Calidad MAN.4 Proceso de Gestión del Riesgo	MAN.1 Proceso de Gestión MAN.2 Proceso de Gestión de Proyecto MAN.3 Proceso de Gestión de Calidad MAN.4 Proceso de Gestión del Riesgo

Fuente: Elaboración Propia

MAN.1 Proceso de Gestión: Esta basado en organizar, monitorear y controlar la inicialización y ejecución de cualquier proceso o función dentro de la organización para alcanzar las metas del negocio de manera efectiva.

MAN.2 Proceso de Gestión de Proyecto: Se refiere a la Identificación, establecimiento, coordinación y monitoreo de actitudes, tareas y recursos para un proyecto produciendo un producto o servicio que satisfaga los requerimientos.

MAN.3 Proceso de Gestión de Calidad: Se basa en monitorear la calidad de los productos y/o servicios del proyecto a través del ciclo de vida del proyecto.

MAN.4 Proceso de Gestión del Riesgo: Esta basado en identificar y mitigar continuamente los riesgos del proyecto a través del ciclo de vida del proyecto.

3.4.11 Organizacional (ORG):

Está relacionado con 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. La tabla N° 12 muestra las características asociadas a la categoría Organizacional agrupadas por dimensión.

Tabla N° 12

Características de la Categoría Gestión Agrupadas por Dimensión

EFFECTIVIDAD DEL PROCESO	EFICIENCIA DEL PROCESO
ORG.1 Proceso de Alineación Organizacional	ORG.3 Proceso de Establecimiento del Proceso
ORG.2 Proceso de Gestión de Cambio	ORG.4 Proceso de Evaluación del Proceso
ORG.5 Proceso de Mejoramiento	ORG.5 Proceso de Mejoramiento
ORG.8 Proceso de Medición	ORG.6 Proceso de Gestión de RRHH
ORG.9 Proceso de Reuso	ORG.7 Proceso de Infraestructura

Fuente: Elaboración Propia

ORG.1 Proceso de Alineación Organizacional: Se debe garantizar que los individuos en la organización compartan una visión común, cultura y entendimiento de las metas del negocio para autorizarlos a funcionar efectivamente. No obstante, la Reingeniería de los negocios y Gerencia de Calidad del Sistema ocurren en un contexto de negocios y, para tener éxito, debe ser dirigido hacia las metas del negocio.

ORG.2 Proceso de Gestión de Cambio: Planificar, gestionar e implementar el cambio, cuando sea necesaria una adaptación al medio externo de la organización o una mejora recurrida por exigencias internas.

ORG.3 Proceso de Establecimiento del Proceso: Se refiere a establecer una serie de procesos organizacionales para todos los ciclos de vida de los procesos del Sistema, y cómo ellos se aplican a las actividades del negocio.

ORG.4 Proceso de Evaluación del Proceso: Se debe determinar la extensión de los procesos estándares del software de la organización contribuyendo con la realización de las metas del negocio; y ayudar a la organización a centralizar sus necesidades para un mejoramiento continuo del proceso.

ORG.5 Proceso de Mejoramiento: Esta basado en promover una mejora continua de la efectividad y la eficiencia en los procesos usados por la organización, de acuerdo con sus necesidades de negocio.

ORG.6 Proceso de Gestión de RRHH: Proporcionar, tanto a la organización como a los proyectos, individuos que posean capacidades y conocimientos para realizar sus roles efectivamente y trabajar juntos como un grupo cohesionado.

ORG.7 Proceso de Infraestructura: Se refiere al mantenimiento de una infraestructura estable y segura, necesaria para apoyar la realización de cualquier otro proceso.

ORG.8 Proceso de Medición: Se debe recopilar y analizar los datos relacionados con los productos desarrollados y procesos implementados dentro de la unidad organizacional, apoya una gestión efectiva de los procesos, y prueba objetivamente productos de calidad.

ORG.9 Proceso de Reuso: Promover y facilitar la reusabilidad de los productos del nuevo sistema y el existente desde una perspectiva organizacional y producto/ proyecto.

APÉNDICE II

MODELOS DE LA CALIDAD DEL SOFTWARE

Los modelos serán estudiados brevemente considerando sus características más resaltantes, de esta forma se podrán identificar sus fortalezas y sus debilidades

2.1 Modelos de Calidad del Producto

2.1.1 Mc Call

El modelo fue propuesto por McCall en 1977 y está orientado a los desarrolladores de Sistemas, para ser utilizado durante el proceso de desarrollo. Abarca tres (3) áreas de trabajo:

Tabla N° 1

Áreas de Trabajo del Modelo de McCall

ÁREA	DESCRIPCIÓN
OPERACIÓN DEL PRODUCTO	Requiere que pueda ser comprendida rápidamente, operada eficientemente y que los resultados sean aquellos requeridos por el usuario.
REVISIÓN DEL PRODUCTO	Está relacionada con la corrección de errores y la adaptación de los sistemas. Esto es importante porque es generalmente considerada como la parte más costosa en el desarrollo de software.
TRANSICIÓN DEL PRODUCTO	Puede que no sea muy importante en todas las aplicaciones. Sin embargo, la orientación a procesamiento distribuido y el rápido cambio en el hardware es probable que incremente su importancia.

Fuente: (Gillies, 1997)

2.1.2 FURPS (Functionality, Usability, Reliability, Performance, Supportability)

Hewlett-Packard lo presenta en 1987, desarrollando un conjunto de factores de calidad de software (funcionalidad, facilidad de empleo, fiabilidad, rendimiento y capacidad de soporte) con sus respectivos atributos (Pressman, 2002). Los factores de calidad FURPS y sus atributos, pueden usarse para establecer métricas de la calidad para todas las actividades del proceso de desarrollo de Sistemas de Información. A continuación en la tabla N° 2 se describen los factores de este modelo con sus respectivos atributos.

Tabla N° 2.
Factores y Atributos del Modelo de FURPS

FACTORES	ATRIBUTOS
FUNCIONALIDAD	Conjunto de características Capacidades Generalidad Seguridad Facilidad de uso Factores humanos Aesthetic Consistencia Documentación
FIABILIDAD	Frecuencia/ Severidad de falla Recuperabilidad Predictabilidad Precisión Tiempo promedio de falla
RENDIMIENTO	Velocidad Eficiencia Consumo de recursos Entradas Tiempo de respuesta

CAPACIDAD DE SOPORTE	Capacidad de prueba Extensibilidad Adaptabilidad Mantenibilidad Compatibilidad Configurabilidad Capacidad de Servicio Capacidad de Instalación
-------------------------------------	---

Fuente: (Pressman, 2002)

2.1.3 Modelo de Dromey

Para Dromey (1994) la forma más común de formular modelos para la calidad de los productos de software es identificar primero un conjunto pequeño de atributos de calidad de alto nivel y luego, siguiendo un proceso “top-down”, de descomposición de esos atributos en subconjuntos de atributos.

En el trabajo de Dromey (1994) se describe primero un modelo genérico de la calidad que consiste en tres entidades primarias: un conjunto de componentes (o formas estructurales), un conjunto de propiedades de calidad de los componentes y un conjunto de atributos de calidad de alto nivel, como base al modelo para la calidad de los productos de software que este investigador propone posteriormente.

Adicionalmente, Dromey (1994) propone cuatro propiedades de la Calidad del Software, medidas a través de atributos: *Propiedad de correctitud*, *Propiedad estructural*, *Propiedad de modularidad* y *Propiedad descriptiva*.

Otras de las normas o estándares internacionales que está basada en un modelo de calidad de producto y define la calidad como un grupo de características del producto es la norma ISO 9126 (Ortega et al., 2000). A continuación se hace una descripción más detallada acerca de este modelo.

2.1.4 Estándar ISO 9126

Presentado en 1998, los factores de calidad de los productos de software y sus atributos son (O'Connor, 1998):

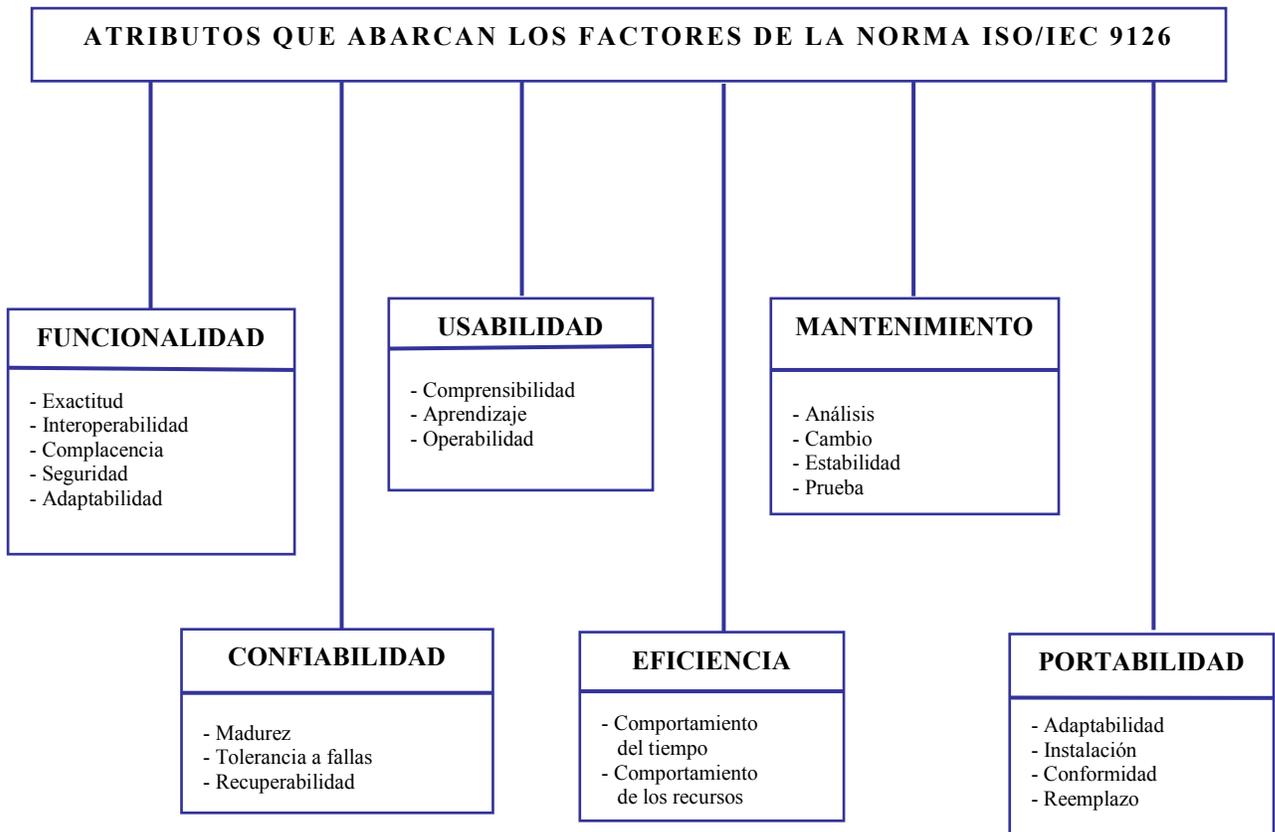


Figura N° 1. Atributos de las Características de la Norma ISO/IEC 9126

Fuente: (O'Connor, 1998)

- *Funcionalidad:* Este factor comprende un conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y propiedades específicas. Las funciones son aquellas que satisfacen las necesidades establecidas.

- *Confiabilidad:* Este factor comprende un conjunto de atributos que se relacionan con la capacidad del software para mantener su nivel de actuación bajo ciertas condiciones en un período de tiempo dado.

- *Usabilidad*: Este factor comprende un conjunto de atributos que se relacionan con el esfuerzo necesario para el uso y el valor de uso, por un conjunto determinado de usuarios.

- *Eficiencia*: Este factor comprende un conjunto de atributos asociados a la relación entre el nivel de desempeño del software y la cantidad de recursos usados bajo determinadas condiciones.

- *Mantenibilidad*: Este factor comprende un conjunto de atributos asociados al esfuerzo necesario para hacer modificaciones específicas.

- *Portabilidad*: Este factor comprende un conjunto de atributos asociados a la habilidad del software para ser transferido de un ambiente a otro.

Como resumen, a continuación se presenta la tabla N° .3 que describe las características principales de cada uno de estos modelos.

Tabla N° 3.

Tabla Resumen de los Modelos de la Calidad del Producto de Software

MODELO	CARACTERÍSTICAS
McCALL	<ul style="list-style-type: none"> • Es usado durante el proceso de desarrollo. • Abarca tres áreas de trabajo: operación, revisión y transición del producto. • No incluye características de rendimiento de hardware.
FURPS	<ul style="list-style-type: none"> • Posee factores de calidad denominados atributos y sub-atributos respectivos. • Está orientado a un modelo de calidad del producto de software. • Sus factores de calidad y atributos pueden usarse para establecer métricas de calidad para todo el proceso de desarrollo de sistemas de software.
	<ul style="list-style-type: none"> • Identifica un conjunto pequeño de atributos de calidad de alto nivel y sigue la dirección top-down. • Refleja que se puede estudiar la calidad de un producto

DROMEY	<p>desde diferentes puntos de vista.</p> <ul style="list-style-type: none"> • Incluye formas estructurales en el modelo. • Propone cuatro propiedades de la calidad de software medidas a través de atributos.
ISO 9126	<ul style="list-style-type: none"> • Se orienta directamente hacia la calidad de producto de software. • Aspectos como funcionalidad, usabilidad, eficiencia, confiabilidad, mantenimiento y portabilidad están relacionados con la calidad del producto final que se le entrega al usuario. • Define la calidad como un grupo de características del producto. • Soporta la construcción de calidad en el software y la definición de normas por medio de codificación con lenguajes estándares.

Fuente: Propia

Cada uno de estos modelos presenta aportes importantes para evaluar la calidad del producto, así como también debilidades.

El modelo de McCall proporciona un mecanismo para que el gestor del proyecto identifique lo que considera importante. Entre otros factores, como son facilidad de mantenimiento y transportabilidad, se ha demostrado que tienen un impacto significativo en el costo del ciclo de vida. Pero, McCall obvia el atributo del entorno personal que bien podría corresponder al factor de operación del producto, y tampoco incluye el atributo de rendimiento de hardware, que podría corresponder al factor de transición del producto. Por otra parte Dromey, señala que no es posible construir la fiabilidad y el mantenimiento, pero resalta que si es posible construir un conjunto de propiedades que ayudan a garantizar dichas propiedades.