

**INSTITUTO SUPERIOR MINERO-METALÚRGICO**  
**Dr. Antonio Núñez Jiménez**  
**METALURGIA-ELECTROMECAÁNICA**



**CALIDAD DE PRODUCTOS SOFTWARE**  
**MÉTRICAS APLICADAS**

**Trabajo de diploma para optar por el título de**  
**Ingeniero Informático**

**Autor:**

**Eliet Peña Núñez**

**Tutores:**

**Dra. Yiezenia Rosario Ferrer**  
**Dr. Arístides Alejandro Legrá Lobaina**

**Holguín, junio, 2009**  
**Año del 50 Aniversario del Triunfo de la Revolución**

*“Cuando puedas medir lo que estás diciendo y expresarlo con números, ya conoces algo sobre ello; pero cuando no puedas medirlo, cuando no puedas expresarlo con números, su conocimiento es precario e insatisfactorio: puede ser el comienzo del conocimiento, pero apenas estás avanzando hacia el escenario de la ciencia”*

*Lord Kelvin*

## **Declaración de Autoría**

Declaro ser autor de la presente tesis y reconozco al ISMM los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de Junio del año 2009.

Eliet Peña Núñez

---

Firma del Autor

Dra. Yiezenia Rosario Ferrer

Dr. Arístides Alejandro Legrá Lobaina

---

Firma del Tutor

---

Firma del Tutor

## **Dedicatoria**

*A mis padres, hermano y demás familiares, con los que tengo una eterna deuda de gratitud.*

*A la Revolución cubana, por hacer realidad los sueños de tantos jóvenes universitarios.*

## **Agradecimientos**

*A nuestro Comandante en Jefe, Fidel Castro Ruz, por ofrecerme la gran oportunidad de ser partícipe del proceso social más extraordinario de la historia y alcanzar las metas trazadas en mi vida.*

*A mis tutores, Yiezenia Rosario Ferrer y Arístides Alejandro Legrá Lobaina por sus consejos y enseñanzas.*

*A todos aquellos, que de una forma u otra han contribuido al desarrollo de este trabajo, es imposible realizar la culminación del mismo sin significar mi más profundo agradecimiento.*

## Resumen

Actualmente las entidades en Cuba dedicadas a la producción de software tienen entre sus principales objetivos desarrollar productos y servicios informáticos de alta calidad. Para poder obtener estos resultados y lograr un posicionamiento y un reconocimiento en el mercado, es necesaria la implantación de Modelos o Estándares de Calidad que garanticen la comprobación objetiva de la evaluación de la calidad de los productos de software, desarrollados en cada una de las entidades. La premisa principal del presente trabajo de investigación es: proponer un conjunto de métricas definidas internacionalmente, con el objetivo de que se apliquen durante el proceso de evaluación de la calidad de los productos de software, definido. Además con el desarrollo del mismo se persigue crear una conciencia en la utilización de las métricas en la Industria del Software. Para lograr lo antes expuesto, se realizó un estudio de las métricas estandarizadas internacionalmente, de las cuales se seleccionaron, las que conforman la propuesta y luego se efectuó la validación de la misma, obteniendo como resultado que la propuesta sea aplicable en entidades dedicadas a la evaluación de los productos de software.

# Índice de Contenidos

Introducción .....	11
1.1 Introducción .....	15
1.2 Situación actual .....	16
1.3 La Industria Cubana del Software .....	17
1.4 ¿Que es Calidad? .....	19
1.4.1 ¿Qué es Calidad de Software? .....	19
1.5 Métricas de Software .....	20
1.5.1 Características de las Métricas .....	21
1.5.2 Clasificación de las Métricas .....	22
1.5.3 ¿Por qué es importante medir? .....	23
1.5.4 Proceso de Medición .....	24
1.5.5 Establecimiento de una línea base .....	25
1.5.6 Métricas del Proceso .....	26
1.5.7 Métricas del Proyecto .....	27
1.5.8 Métricas del Producto .....	28
1.6 Métricas y Calidad .....	28
1.6.1 Modelos de Calidad de Software a Nivel Producto .....	29
1.6.2 La Evaluación de los Productos de Software .....	31
1.6.3 Estándar Internacional ISO/IEC 9126 .....	32
1.6.3.1 Modelo de calidad .....	33
1.6.3.2 Modelo para la calidad interna y externa .....	36
1.6.3.3 Métricas Externas .....	37
1.7 Conclusiones .....	43
2.1 Introducción .....	44
2.2 Objetivos de la Propuesta de Solución .....	44
2.3 Roles y responsabilidad de los participantes en el proceso de evaluación .....	45
2.4 Selección de las métricas externas .....	45
2.4.1 Criterios utilizados para la selección de las métricas externas .....	46
2.4.2 Métricas externas seleccionadas .....	47
2.4.2.1 Métricas de Funcionalidad .....	47
2.4.2.2 Métricas de Confiabilidad .....	50
2.4.2.3 Métricas de Usabilidad .....	58

2.5 Descripción de las fases del Proceso de evaluación.....	60
2.5.1 Recopilación de Datos.....	60
2.5.2 Establecimiento de los pesos .....	61
2.5.3 Cálculo de Métricas .....	62
2.5.4 Valoración de los Resultados de las Métricas .....	64
2.5.5 Veredicto conclusivo.....	66
2.6 Conclusiones .....	68
3.1 Introducción .....	69
3.2 Guía para la validación .....	69
3.3 Conclusiones .....	77
Conclusiones Generales.....	78
Recomendaciones .....	79
Referencias Bibliográficas .....	80
Glosario de Términos y Siglas.....	83
Anexos.....	85
Anexo 1 - Definiciones de las Características y Sub-características de calidad propuestas por el estándar ISO/IEC 9126.....	85
Anexo 2 - Tablas contentivas de las métricas .....	88
Anexo 3 - Plantilla # 1 .....	101
Anexo 4 - Plantilla # 2.....	105
Anexo 5 - Plantilla # 3.....	107
Anexo 6 - Plantilla # 4.....	113
Anexo 7 - Modelo para la recogida de información referente al peso de los criterios .....	115
Anexo 8 - Modelo para la recogida de información referente a la calificación de los criterios .....	117

## Índice de Figuras

<b>Figura 1.</b> Concepto de Métricas.....	21
<b>Figura 2.</b> Proceso de recopilación de Métricas del Software.....	25
<b>Figura 3.</b> Proceso de evaluación .....	32
<b>Figura 4.</b> Relación entre las normas de las series ISO/IEC 9126 e ISO/IEC 14598 .....	34
<b>Figura 5.</b> La Calidad en el Ciclo de Vida.....	35
<b>Figura 6.</b> Modelo para la Calidad Externa e Interna .....	36
<b>Figura 7.</b> Niveles de Maduración .....	38
<b>Figura 8.</b> Fases del Proceso de Evaluación.....	60
<b>Figura 9.</b> Estructura para el Peso .....	62

## Índice de Tablas

<b>Tabla 1.</b> Datos de las variables contenidas en las métricas .....	61
<b>Tabla 2.</b> Establecimiento del peso .....	62
<b>Tabla 3.</b> Cálculo de métricas.....	63
<b>Tabla 4.</b> Puntuación a las sub-características.....	65
<b>Tabla 5.</b> Resultado del trabajo de expertos.....	71
<b>Tabla 6.</b> Cálculo de la Dispersión (S) para hallar la concordancia entre los expertos .....	72
<b>Tabla 7.</b> Tabla para el cálculo de concordancia de Kendall.....	73
<b>Tabla 8.</b> Tabla de calificación de cada criterio .....	74

## Introducción

En la actualidad el desarrollo del software ha aumentado considerablemente, lo que hace que sea muy difícil lograr un posicionamiento y un reconocimiento en el mercado internacional. Esto trae consigo que sea un reto para la Industria del Software desarrollar las estrategias que le permitan alcanzar un nivel de calidad realmente alto, por lo que se hace necesario todo un estudio para la elección e implantación del Modelo o Estándar de Calidad indicado. [ALLIANCE 2007]

En Cuba en los últimos años ha habido un auge en el desarrollo de software, principalmente las empresas han comenzado la automatización e informatización de varias áreas donde el trabajo con las tecnologías se desempeña con mayor calidad y a corto plazo. Es por ello que el Gobierno Cubano tiene como una de sus tareas principales desarrollar la Industria Cubana del Software con el objetivo de informatizar la sociedad e insertarnos en el mercado de software a nivel mundial. [La informatización en Cuba, 2005]

Con los avances científico-técnicos, evidenciados mayormente en la creación de productos o servicios para satisfacer la demanda del mercado o de determinados clientes, aumenta la competencia y por tanto la necesidad de lograr una buena aceptación por parte de los usuarios.

Este nivel de aceptación está estrechamente ligado a la definición de criterios que guíen el proceso de concepción y creación del producto. Partiendo de esto surge la necesidad de definir cuándo un producto, proceso o servicio tiene calidad.

La calidad es un activo estratégico clave, del que dependen la mayor parte de las organizaciones, no sólo para mantener su posición en el mercado sino incluso para asegurar su supervivencia. El desarrollo de un producto que satisfaga, en la mayor medida posible, los requerimientos del cliente, es la medida de calidad buscada en la producción.

Una especificación y evaluación integral y detallada de la calidad de los productos de software es un factor clave para asegurar que la calidad sea la adecuada. Esto se puede lograr definiendo de manera apropiada las características de calidad, teniendo en cuenta el propósito del uso del producto de software en la institución.

Es importante especificar y evaluar cada característica relevante de la calidad de los productos de software, cuando esto sea posible, utilizando mediciones validadas o de amplia aceptación, que hagan técnicamente transparente esta actividad.

En el mundo actualmente existen normas o estándares internacionales para medir la calidad de los productos de software. En Cuba a pesar de la existencia de estos, no se han consolidado los conocimientos acerca de las posibles métricas a utilizar en el proceso de evaluación, para medir la calidad de los productos de software; además estas no se aplican adecuadamente ni se hace en correspondencia con la criticidad o nivel de riesgo del producto que se está evaluando. La no utilización correcta de las métricas para evaluar la calidad de los productos de software puede provocar la ocurrencia de desastres tecnológicos como ha ocurrido a través de la historia en diferentes partes del mundo.

A raíz del análisis de todo lo anterior surge la interrogante: ¿Cómo perfeccionar el proceso de evaluación de la calidad de los productos de software?

Entonces, el **Problema Científico** a resolver es el siguiente: El conjunto de métricas que se utilizan en la evaluación de la Calidad de los Productos de Software no garantizan un nivel satisfactorio de la calidad de los mismos.

El **Objeto de estudio** de la investigación es la Calidad de los Productos de Software y el **Campo de Acción** en el que se enfoca la investigación son Las Métricas a aplicar para evaluar la Calidad de los Productos de Software.

Para proporcionarle una solución al problema analizado se define como **Objetivo General**: Proponer un sistema de métricas que al ser aplicadas durante el proceso de evaluación de la calidad de los productos de software, garanticen un nivel satisfactorio de la calidad de los productos de software.

Del objetivo global se derivan los siguientes **Objetivos específicos**:

- Realizar un estudio de las métricas estandarizadas internacionalmente en vista a la selección.
- Definir la propuesta de las métricas en un proceso de evaluación de productos de software.
- Proponer métodos de evaluación con la ayuda del empleo de las métricas.
- Validar la propuesta.

Para el desarrollo de la investigación se propone la siguiente **Idea a Defender**: Si se utilizan las métricas propuestas que constituyen un sistema funcional, confiable y usable para evaluar la Calidad de los Productos de Software, entonces será posible aumentar la calidad del producto final.

Con el objetivo de guiar, controlar y evaluar la investigación se definieron las siguientes **Tareas de investigación**:

- Estudio y análisis de métricas estandarizadas internacionalmente posibles a aplicar.
- Selección de las métricas para el desarrollo de la propuesta.
- Diseño de una guía y estrategia de aplicación de las métricas seleccionadas.
- Análisis del resultado de los estudios realizados para la aplicación de la guía de métricas, con el objetivo de su validación.

Esta propuesta traerá consigo un mejoramiento de la calidad del producto final, además permitirá ir perfeccionando el proceso de evaluación de la calidad de los productos de software. Actualmente, las entidades en Cuba dedicadas a evaluar los productos de software no hacen uso de las métricas propuestas, de ahí la importancia de la investigación, la novedad y que el resultado obtenido sea aplicado en las empresas dedicadas a la evaluación de la calidad de los productos de software.

La estrategia planteada para conducir la investigación es del tipo **Descriptiva**, ya que se persigue establecer la caracterización estructural de un cuerpo de métricas y su correlación para dictaminar un veredicto de aceptación de un producto de software.

El presente trabajo de diploma consta de una Introducción, 3 capítulos con la intención de realizar una división por los contenidos que serán tratados, las conclusiones generales, recomendaciones, referencias bibliográficas utilizadas durante el desarrollo del trabajo, un glosario de términos y siglas y por último los anexos que complementan el cuerpo del trabajo y que son necesarios para su entendimiento.

**Capítulo 1:** Fundamentación Teórica. En este capítulo se hace un análisis de la actualidad internacional y nacional sobre el tema Calidad del Software y las Métricas del Software profundizando en los términos referentes al mismo, además se abordan las principales definiciones que se tienen en cuenta durante todo el trabajo.

**Capítulo 2:** Descripción de la Propuesta de Solución. En este capítulo se presentan las métricas propuestas, el proceso de evaluación para la aplicación de las métricas, en el cual se describen cada uno de los elementos que componen el proceso y se detallan los pasos que se realizan en la evaluación.

**Capítulo 3:** Validación de la Propuesta de Solución. En este capítulo se determina la probabilidad de éxito que tiene la propuesta, para esto se realizan un conjunto de cálculos los cuales se detallan con el objetivo de lograr un mayor entendimiento.

# 1

## CAPÍTULO FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción

La empresa de software cubana aún siendo joven reúne sus esfuerzos para desarrollar software de calidad con los que pueda insertarse en el mercado mundial. Un factor decisivo en el logro de este objetivo es el aseguramiento (*garantía*) de calidad de un producto de software, determinado por el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto (*software*) satisfará los requisitos dados de calidad.

Actividades como la Gestión de la Configuración del Software, la Verificación y Validación del Software a lo largo del ciclo de vida y el uso de las Métricas de Software en el control de Proyecto como herramienta fundamental a la hora de garantizar calidad del producto. Actualmente el término métricas de software está siendo ampliamente difundido y utilizado. En el presente capítulo, se hace un análisis crítico del concepto de métrica y todo lo referente a este tema en cuanto al estado del arte en el mundo, y en Cuba, además se abordan los términos que sirven de soporte teórico a la investigación desarrollada y que estarán presentes a lo largo de este trabajo.

## **1.2 Situación actual**

En la actualidad el uso de las métricas se está poniendo en práctica con éxito en el amplio mercado del software pues las empresas productoras están reconociendo la importancia que tienen las mediciones para cuantificar y por consiguiente gestionar de forma más efectiva la calidad de los productos de software.

Es válido aclarar que en ocasiones los resultados de los procesos de medición no son interpretados de la mejor manera, pues aún existen compañías que no tienen una cultura adecuada sobre la medición, desconociendo el alcance de calidad que pudiera alcanzar el producto final.

Varios estándares han incluido a las métricas de software, a continuación se mencionan algunos de ellos.

### **ISO 15504**

Incluye dentro de la categoría de los procesos organizacionales (en la segunda parte de la norma) al proceso de medición, incluyendo la definición de métricas, la gestión de los datos (incluidos los históricos), y el uso de las métricas en la organización.

### **Familia ISO 9000**

Se establece la necesidad de implementar el proceso de medición con el objetivo de controlar la calidad del producto, la capacidad del proceso y la satisfacción del cliente, la gestión usa métricas como una entrada fundamental para la planificación, control y gestión del proyecto.

## **ISO/IEC 9126**

Las normas de las familias ISO/IEC 9126 (Calidad de los productos de software) y de la ISO/IEC 14598 (Evaluación de los productos de software). Las características de calidad de los productos de software definidas en esta parte de la ISO/IEC 9126 pueden ser utilizadas para especificar tanto los requisitos funcionales como no funcionales de los clientes y usuarios.

A pesar de la existencia de métricas para evaluar la calidad de los productos de software, en una encuesta realizada sobre el nivel de conocimiento en el tema, en el sitio Web [www.CalidaddelSoftware.com](http://www.CalidaddelSoftware.com) integrado por más de 900 miembros de los cuales el 83% son de España y 14% de Iberoamérica, cuyo objetivo es mantener en contacto a personas y organizaciones interesadas en la Calidad y la Mejora del Proceso Software, se preguntó: ¿Sobre qué cree que hay mayor carencia de conocimientos? El mayor porcentaje de votos entre las nueve áreas propuestas fue para la respuesta estimación y métricas con un 26.84%. [Encuestas]

Es un problema la gran variedad de métricas que existen y que en la mayoría de los casos no se tiene un conocimiento explícito de sus objetivos y la manera en que puedan aplicarse. Cada empresa es responsable de definir las métricas que va a implantar, teniendo en cuenta sus objetivos organizacionales y necesidades de información, las métricas deben estar alineadas con el calendario, costo y niveles de calidad propuestos.

### **1.3 La Industria Cubana del Software**

La Industria Cubana del Software (ICSW) está llamada a convertirse en una significativa fuente de ingresos nacional, como resultado del correcto aprovechamiento de las ventajas del considerable capital humano disponible. Las Universidades cubanas juegan un papel importante en el desarrollo de la ICSW, y en la materialización de los proyectos asociados al programa cubano de informatización. [La informatización en Cuba, 2005]

La preparación de los recursos humanos especializados para las Tecnologías de la Informática y las Comunicaciones (TIC) es un factor clave de la estrategia cubana de Informatización. Además de las especialidades afines a la Informática en 17 de las universidades y 16 Institutos Superiores Pedagógico (ISP), también se cuenta con la Universidad de las Ciencias Informáticas (UCI), con los 26 Institutos Politécnicos de Informática (IPI) donde estudian ya más de 40 000 estudiantes y la preparación en los 600 Joven Club de Computación presentes en todos los municipios del país que constituyen un intensivo para todos los trabajadores y estudiantes que deseen incursionar en el mundo de las computadoras, facilitando el desarrollo y aplicación de la informática en esferas específicas correspondientes a un municipio o localidad determinada. [La informatización en Cuba, 2005]

Sin embargo a pesar del capital humano y desarrollo tecnológico con que cuentan las empresas de software, en muchas ocasiones no se realizan actividades relacionadas a la gestión del proyecto, la gestión de la configuración, la gestión de la calidad y las mediciones.

En estudios realizados por Centros de Estudios de Ingeniería de Sistemas (CEIS) en empresas nacionales se detectaron problemas entre los que se encuentran: los resultados alcanzados no cubren las expectativas, la productividad es baja, la cantidad real de recursos a consumir (en tiempo principalmente) es casi impredecible, el trabajo realizado casi nunca tiene la calidad y profesionalidad requerida, los proyectos sufren atrasos excesivos y no existen antecedentes de datos históricos sobre la calidad de los productos que han sido elaborados. [Acosta 2007]

Entre tantos problemas identificados las empresas de software cubanas están dando los primeros pasos en el perfeccionamiento de la gestión de la calidad con el objetivo de poder certificarse o al menos alcanzar algún nivel de madurez para evolucionar y mejorar sus procesos desde procesos inmaduros a procesos disciplinados, maduros con calidad, eficiencia mejorada y probada.

## **1.4 ¿Que es Calidad?**

El Diccionario de la Real Academia Española, conceptúa la Calidad como la "propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor", y es sinónimo de "buena calidad" la "superioridad o excelencia". [Infocalidad, 2005]

Por su parte, la ISO (International Organization for Standardization – Organización Internacional para la Estandarización), agrega otros aspectos importantes a tener en cuenta cuando define la calidad como el conjunto de propiedades y de características de un producto o servicio, que le confieren aptitud para satisfacer unas necesidades explícitas o implícitas.

Las definiciones anteriores conducen a la siguiente conclusión:

Calidad, resume las características, propiedades, cualidades y en general atributos propios de un producto, que determinan sobre este la ausencia de defectos y la conformidad de todo el personal que de una forma y otra se vinculan con él, (productores, clientes, usuarios, etc.). La calidad consiste en no tener deficiencias.

### **1.4.1 ¿Qué es Calidad de Software?**

Se define la calidad de software como la ausencia de errores de funcionamiento, la adecuación a las necesidades del usuario, y el alcance de un desempeño apropiado (tiempo, volumen, espacio), además del cumplimiento de los estándares. Los objetivos que la calidad persigue son: La aceptación (utilización real por parte del usuario) y la Mantenibilidad (posibilidad y facilidad de corrección, ajuste y modificación durante largo tiempo). [Ingeniería del software, 2005]

La Calidad del Software es la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente. Es el conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas. [Pressman, 1998]

Las definiciones anteriores conducen a las siguientes conclusiones:

- La principal muestra de calidad es que el usuario quede satisfecho con el producto.
- La aplicación de estándares es un factor importante en la obtención de productos con calidad.
- Existen un conjunto de características que tienen que estar implícitas en productos que tengan calidad.

## 1.5 Métricas de Software

Todas las organizaciones de software exitosas implementan mediciones como parte de sus actividades cotidianas pues estas brindan la información objetiva necesaria para la toma de decisiones y que tendrá un impacto efectivo en el negocio y desempeño en la ingeniería.

Para poder asegurar que un proceso o sus productos resultantes son de calidad o poder compararlos, es necesario asignar valores, descriptores, indicadores o algún otro mecanismo mediante el cual se pueda llevar a cabo dicha comparación.

Para entender mejor el concepto de métrica es necesario aclarar que los términos, métricas, medición y medida no tienen el mismo significado.

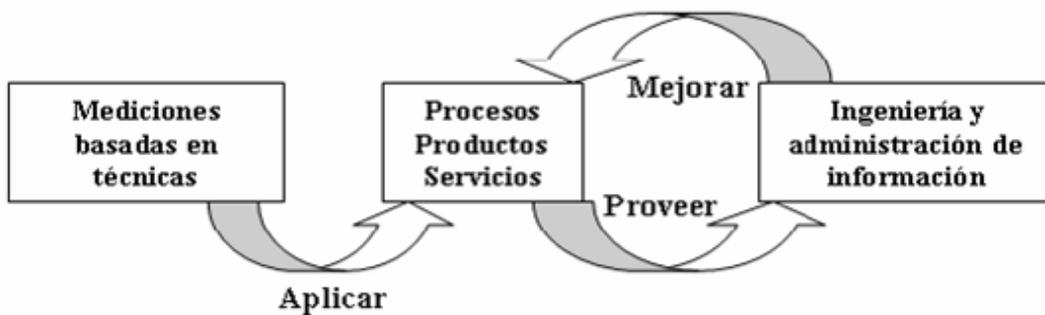
Medida: Proporciona una indicación cuantitativa de la extensión, cantidad, dimensiones, capacidad o tamaño de algunos atributos de un proceso o producto. [Pressman, 1998]

Medición: La medición es el acto de determinar una medida. [Pressman, 1998]

Métrica: Es una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. [Pressman, 1998]

Se definen las métricas de software como “La aplicación continua de mediciones basadas en técnicas para el proceso de desarrollo del software y sus productos, para suministrar información relevante a tiempo, así el administrador con el empleo de estas técnicas mejorará el proceso y sus productos”.

Para una definición más completa deben incluirse los servicios relacionados al software como la respuesta a los resultados del cliente (Ver Figura 1).



**Figura 1. Concepto de Métricas**

### **1.5.1 Características de las Métricas**

Para que sea útil en el contexto del mundo real, una métrica del software debe ser objetiva, simple y calculable, consistente en el empleo de unidades y tamaños, persuasiva, además debería ser independiente del lenguaje de programación y proporcionar una realimentación eficaz para el desarrollador de software. [Pressman, 1998]

¿Por qué asegurarnos de que las métricas cumplen estas condiciones?

Las métricas deben ser un instrumento que ayude a mejorar el proceso, producto o proyecto de software, no tiene mucho sentido aplicar métricas que lejos de ayudar a los desarrolladores constituyan un problema; bien por ser demasiado complejas, porque no se entiendan correctamente los objetivos que persiguen o porque arrojen resultados imprecisos que no puedan ser interpretados por los ingenieros de software.

Es importante entonces que una métrica pueda obtenerse fácilmente, que se entienda por qué y para qué se utiliza, que los cálculos no produzcan resultados ambiguos o en los que existan extrañas combinaciones de unidades, y que la interpretación de valores obtenidos esté acorde a las nociones intuitivas del ingeniero de software o especialista. Por otra parte las métricas no deben ser específicas para ningún lenguaje de programación o metodología de desarrollo.

### **1.5.2 ¿Por qué es importante medir?**

Una de las razones principales del incremento masivo en el interés por la medición de software ha sido la percepción de que las métricas son necesarias para la mejora de la calidad del proceso.

Hay cuatro razones para medir los procesos del software, los productos y los recursos [Pressman, 1998]:

1• Caracterizar: Para comprender mejor los procesos, los productos, los recursos y los entornos y para establecer las líneas base para las comparaciones con evaluaciones futuras.

2• Evaluar: Para determinar el estado con respecto al diseño. Las medidas permiten conocer cuándo los proyectos y procesos están perdiendo la pista, de modo que puedan ponerse bajo control. Además para valorar si se cumplen o no los objetivos de calidad trazados y para evaluar el impacto de la tecnología y las mejoras en los productos y procesos.

3• Predecir: Para poder planificar. Los valores que se observan para algunos atributos pueden ser utilizados para predecir otros, lo que contribuye a establecer objetivos alcanzables para el coste, planificación y calidad, de manera que se puedan aplicar los recursos apropiados, además permite analizar los riesgos y realizar intercambios diseño - coste.

4• Mejorar: Se mide para mejorar cuando se recoge la información cuantitativa que ayuda a identificar obstáculos, problemas de raíz, ineficiencias y otras oportunidades para mejorar la calidad del producto y el rendimiento del proceso.

### **1.5.3 Clasificación de las Métricas**

Existen innumerables métricas con propósitos diferentes que reflejan o describen la conducta del software, estas pueden medir entre otros aspectos la competencia, calidad, desempeño y la complejidad del software contribuyendo a establecer de una manera sistemática y objetiva una visión interna del trabajo mejorando así la calidad del producto.

A continuación se muestra la clasificación de las mismas:

Métricas de complejidad: Son todas las métricas de software que definen de una u otra forma la medición de la complejidad; tales como volumen, tamaño, anidaciones, costo (estimación) y configuración. Estas son los puntos críticos de la concepción, viabilidad, análisis, y diseño de software.

Métricas de competencia: Son todas las métricas que intentan valorar o medir las actividades de productividad de los programadores o practicantes con respecto a su certeza, rapidez, eficiencia y competencia.

Métricas de desempeño: Corresponden a las métricas que miden la conducta de módulos y sistemas de un software, bajo la supervisión del sistema operativo o hardware. Generalmente tienen que ver con la eficiencia de

ejecución, tiempo, almacenamiento, complejidad de algoritmos computacionales, etc.

Métricas estilizadas: Son las métricas de experimentación y de preferencia; por ejemplo: estilo de código, las limitaciones, etc. Pero estas no se deben confundir con las métricas de calidad o complejidad.

Métricas de calidad: Son todas las métricas de software que definen de una u otra forma la calidad del software; tales como exactitud, estructuración o modularidad, pruebas, mantenimiento, reusabilidad, entre otras. Estas son los puntos críticos en el diseño, codificación, pruebas y mantenimiento.

Estas clasificaciones de métricas fortalecen la idea, de que más de una métrica puede ser deseable para valorar la complejidad y la calidad del software, teniendo en cuenta que para ello es necesario medir los atributos del software.

#### **1.5.4 Proceso de Medición**

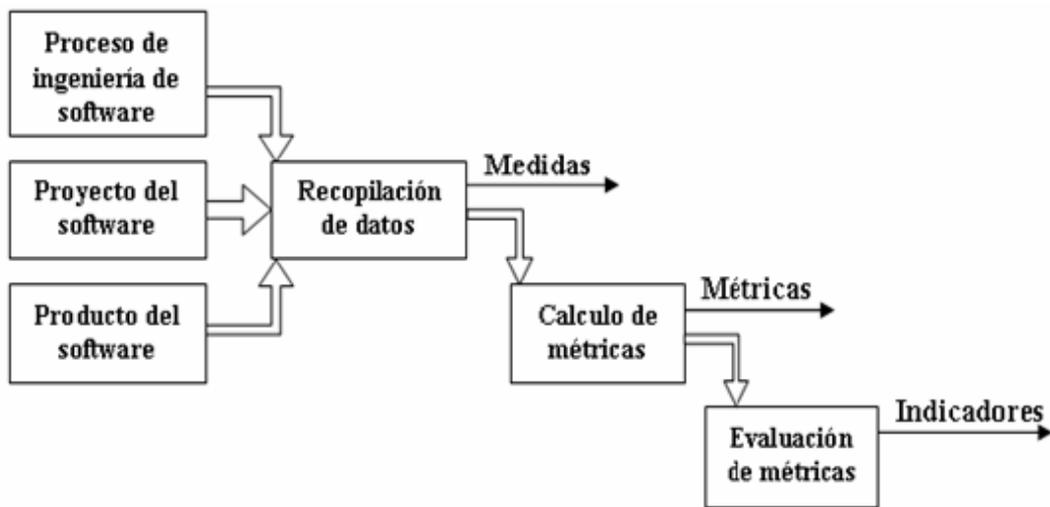
Todo proceso de medición del software tiene como objetivo fundamental satisfacer necesidades de información a partir de las cuales se deben identificar las entidades y los atributos que deben ser medidos.

El proceso de medición, se caracteriza en cinco actividades [Pressman, 2002]:

1. Formulación: Obtención de medidas y métricas del software apropiadas para la presentación del software en cuestión.
2. Colección: Mecanismo empleado para acumular datos necesarios para obtener las métricas formuladas.
3. Análisis: Cálculo de las métricas y la aplicación de herramientas matemáticas.
4. Interpretación: La evaluación de los resultados de las métricas en un esfuerzo por conseguir una visión interna de la calidad de la presentación.
5. Retroalimentación: Recomendaciones obtenidas de la interpretación de métricas y técnicas transmitidas al equipo de desarrollo de software.

### 1.5.5 Establecimiento de una línea base

Un punto de partida para realizar estimaciones es establecer una línea base de métricas que permita a una organización sintonizar su proceso de ingeniería del software para eliminar las causas de los defectos que tienen el mayor impacto en el desarrollo del software, es fundamental que una línea base contenga datos recopilados de proyectos desarrollados anteriormente lo que requiere una investigación histórica de los mismos, la línea base no es más que la recopilación de medidas, métricas e indicadores que guíen el proyecto o el proceso.



**Figura 2. Proceso de recopilación de Métricas del Software**

Por lo general la información reunida no necesariamente tiene que ser diferente. Las mismas métricas pueden obtener beneficios a nivel de proceso, proyecto y producto.

## 1.5.6 Métricas del Proceso

Las métricas del proceso se recopilan de todos los proyectos y durante un largo período de tiempo. Su intento es proporcionar indicadores que lleven a mejoras de los procesos de software a largo plazo. [Pressman, 1998] Un indicador es una métrica o una combinación de métricas que proporcionan una visión profunda del proceso del software, del proyecto de software o del producto en sí.

La medición del proceso implica las mediciones de las actividades relacionadas con el software siendo algunos de sus atributos típicos el esfuerzo, el coste y los defectos encontrados. Las métricas permiten tener una visión profunda del proceso de software que ayudará a tomar decisiones más fundamentadas, ayudan a analizar el trabajo desarrollado, conocer si se ha mejorado o no con respecto a proyectos anteriores, ayudan a detectar áreas con problemas para poder remediarlos a tiempo y a realizar mejores estimaciones.

Para mejorar un proceso se deben medir los atributos del mismo, desarrollar métricas de acuerdo a estos atributos y utilizarlas para proporcionar indicadores que conduzcan la mejora del proceso. Los errores detectados antes de la entrega del software, la productividad, recursos y tiempo consumido y ajuste con la planificación son algunos de los resultados que pueden medirse en el proceso, así como las tareas específicas de la ingeniería del software. Actualmente existen muchas métricas, y éstas deben usarse conforme se ajusten al proceso.

Las métricas del proceso se caracterizan por:

- 1• El control y ejecución del proyecto.
- 2• Medición de tiempos del análisis, diseño, implementación, implantación y postimplantación.
- 3• Medición de las pruebas (errores, cubrimiento, resultado en número de defectos y número de éxito).
- 4• Medición de la transformación o evolución del producto.

### **1.5.7 Métricas del Proyecto**

Dado que el proyecto engloba todos los recursos, actividades y artefactos, que se organizan para lograr un producto de software es de vital importancia definir algunas mediciones que ayuden al mejoramiento del mismo. A nivel de proyecto se minimiza la planificación de desarrollo haciendo los ajustes necesarios para evitar retrasos o riesgos potenciales, minimizar los defectos, y por tanto la cantidad de trabajo que ha de rehacerse, lo que ocasiona una reducción del coste global del proyecto, además puede evaluarse la calidad de los productos en el momento actual y cuando sea necesario.

La primera aplicación de métricas de proyectos en la mayoría de los proyectos de software ocurre durante la estimación. Las métricas recopiladas de proyectos anteriores se utilizan como una base desde la que se realizan las estimaciones del esfuerzo y del tiempo para el actual trabajo del software. A medida que avanza un proyecto, las medidas del esfuerzo y del tiempo consumido se comparan con las estimaciones originales (y la planificación de proyectos). El gestor de proyectos utiliza estos datos para supervisar y controlar el avance. A medida que comienza el trabajo técnico, otras métricas de proyectos comienzan a tener significado. Se miden los índices de producción representados mediante páginas de documentación, las horas de revisión, los puntos de función y las líneas fuentes entregadas, en el proyecto se sigue la pista de los errores detectados durante todas las tareas de ingeniería del software. Cuando va evolucionando el software desde la especificación del diseño, se recopilan las métricas técnicas para evaluar la calidad del mismo y para proporcionar indicadores que influirán en el enfoque tomado para la generación y prueba del código. [Pressman, 1997]

Las métricas del proyecto se caracterizan por:

- 1• Evaluar el estado del proyecto en curso.
- 2• Seguir la pista de los riesgos potenciales.

3• Detectar las Áreas de problemas antes de que se conviertan en “críticas”.

4• Ajustar el flujo y las tareas del trabajo.

5• Evaluar la habilidad del equipo del proyecto en controlar la calidad de los productos de trabajo del software.

### **1.5.8 Métricas del Producto**

Las métricas del producto se centran en las características del software y no en cómo fue producido. Un producto no es solo el software o sistema funcionando sino también los artefactos, documentos, modelos, módulos, o componentes que lo conforman, por tanto, las métricas del producto deben hacerse sobre la base de medir cada uno de estos.

Las métricas del producto son mediciones del producto software. Esta definición incluye el tamaño del producto, la complejidad de la estructura lógica y la complejidad de estructuras de datos, entre otros. Además está estrechamente relacionada con las mediciones del proceso. [Rossi 2003]

### **1.6 Métricas y Calidad**

El principal objetivo de los ingenieros del software es producir un sistema, aplicación o producto de alta calidad, para lo cual emplean métodos y herramientas efectivas dentro del contexto de un proceso maduro de desarrollo del software y además deben desarrollar mediciones que den como resultado sistemas de alta calidad. Para obtener esta evaluación, el ingeniero debe utilizar medidas técnicas, que evalúan la calidad con objetividad, no con subjetividad.

[Acosta 2007]

## 1.6.1 Modelos de Calidad del Software a Nivel Producto

- **Modelo FURPS**

El modelo FURPS propuesto por Robert Grady y Hewlett Packard Co (HP) cuenta con 5 características de calidad del software: (1) Funcionalidad, (2) Facilidad de uso, (3) Confiabilidad, (4) Performance y (5) Facilidad de soporte. Además plantea 2 categorías de requerimientos, las cuales son:

1- requerimientos funcionales (F): especifican funciones que el sistema debe ser capaz de realizar, sin tomar restricciones físicas a consideración, y se definen a través de las entradas y salidas esperadas.

2- requerimientos no funcionales (URPS): Usability (Facilidad de uso), Reliability(Confiabilidad), Performance y Supportability (Facilidad de soporte). Describen atributos del sistema o atributos del ambiente del sistema. [Scalone 2006]

- **Modelo GQM (Goal – Question - Metric)**

El modelo GQM (objetivo-pregunta-métrica /goal – question - metric) de Basili y Rombach es una propuesta de objetivos / metas orientado a la definición de modelos de calidad. Se propone el paradigma GQM para evaluar la calidad de cada proyecto. Este modelo utiliza una propuesta para definir un modelo de calidad hasta obtener las métricas respectivas con el análisis e interpretación de los datos de las mediciones respectivas. Plantea el enfoque de medición para evaluar la calidad del software basado en la identificación de objetivos a lograr.[Scalone 2006]

- **Modelo de McCall**

Hace más de 25 años se definieron factores de calidad como los primeros pasos hacia el desarrollo de métricas de calidad del software. El modelo de McCall organiza los factores en tres ejes o puntos de vista desde los cuales el usuario puede contemplar la calidad de un producto (1) Operación del producto, (2) Revisión del producto y (3) Transición del producto. Cada punto de vista se descompone en una serie de factores que determinan la calidad de cada una de ellos. Cada factor determinante de la calidad, se descompone, a su vez, en una serie de criterios o propiedades que determinan su calidad. Los criterios pueden ser evaluados mediante un conjunto de métricas y para cada criterio deben fijarse unos valores máximo y mínimo aceptables.[Scalone 2006]

- **Modelo de BOEHM**

El modelo de Boehm (1978) agrega algunas características a las existentes en el modelo de McCall y representa una estructura jerárquica de características, cada una de las cuales contribuye a la calidad total. Consiste en un modelo de descomposición de características de calidad del software en 3 niveles (usos principales, componentes intermedios y componentes primitivos) previos a la aplicación de métricas. Este modelo plantea factores de calidad formados por criterios de calidad y métricas respectivas.[Scalone 2006]

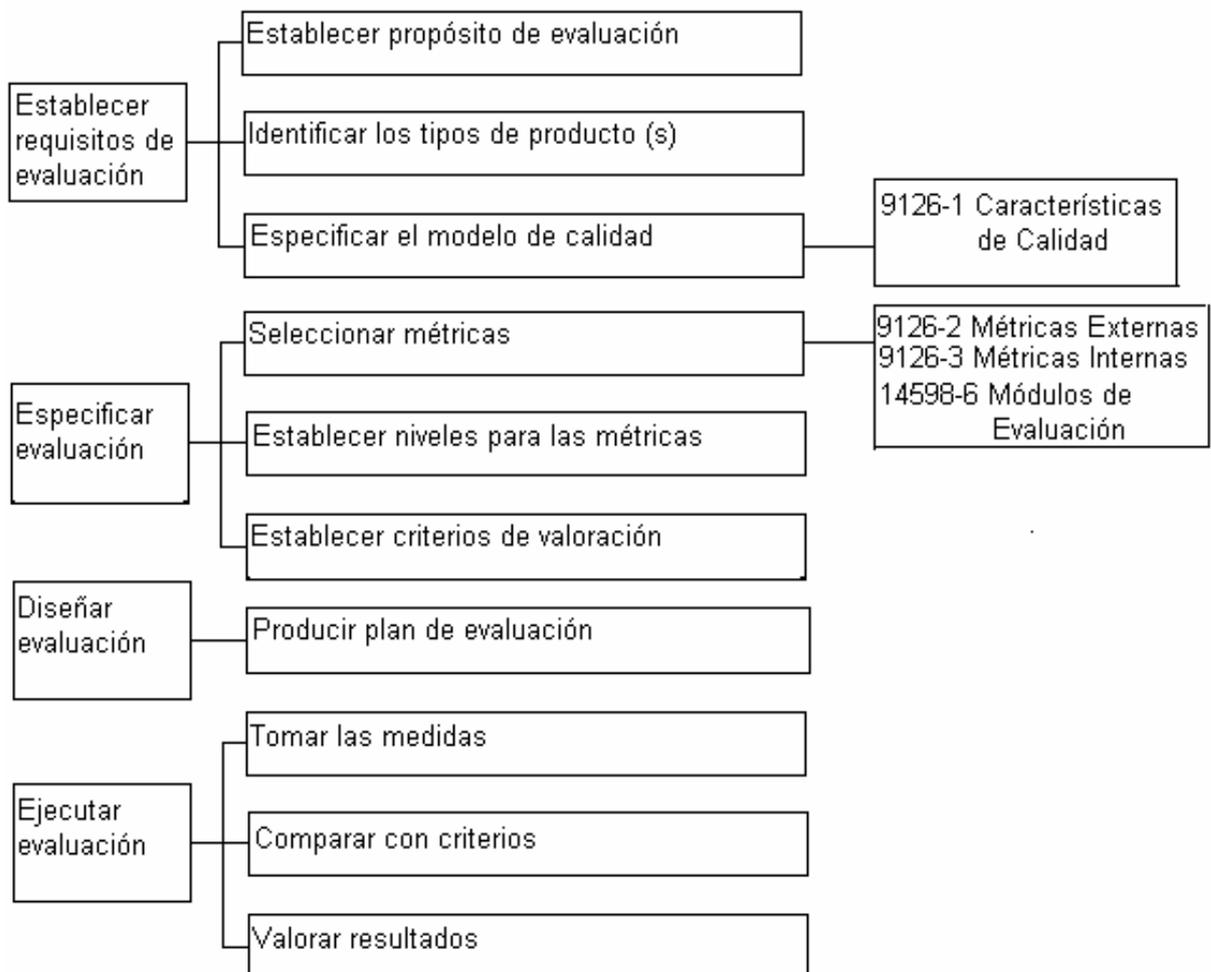
La calidad de un sistema, aplicación o producto es tan buena como los requisitos que describen el problema, el diseño que modela la solución, el código que conduce a un programa ejecutable y las pruebas que ejercitan el software para detectar errores, en este sentido los desarrolladores deben realizar y utilizar mediciones. Para alcanzar esta evaluación de la calidad en tiempo real, se deben utilizar medidas técnicas que evalúan la calidad con objetividad, no con subjetividad.

A pesar de que se pueden recolectar varias medidas de calidad, el primero de los objetivos en el proyecto es medir los errores y defectos. Las métricas que provienen de estas medidas proporcionan una indicación de la efectividad de las actividades de control y de aseguramiento de la calidad en grupos o en particulares. [Pressman, 2002]

## **1.6.2 La Evaluación de los Productos de Software**

La norma ISO/IEC 14598 (Evaluación de los productos de software) establece un proceso de evaluación, que se muestra simplificada en la estructura de la Figura 3, a partir de cuatro subprocesos básicos:

- Establecer los requisitos de la evaluación.
- Especificar la evaluación.
- Diseñar la evaluación.
- Ejecutar la evaluación.



**Figura 3. Proceso de evaluación**

Aunque la norma brinda estos amplios conceptos, no quedan bien definidos los elementos relacionados con Establecer criterios de valoración y Valorar resultados, lo cual se sustentará más detalladamente en la propuesta.

### 1.6.3 Estándar Internacional ISO/IEC 9126

La ISO/IEC 9126 es un estándar internacional para la evaluación del software. Es supervisado por el proyecto SquaRE (Security Quality Requirements Engineering - Ingeniería de Requisitos de Calidad de Seguridad) y la ISO 25000:2005, que siguen los mismos conceptos generales.

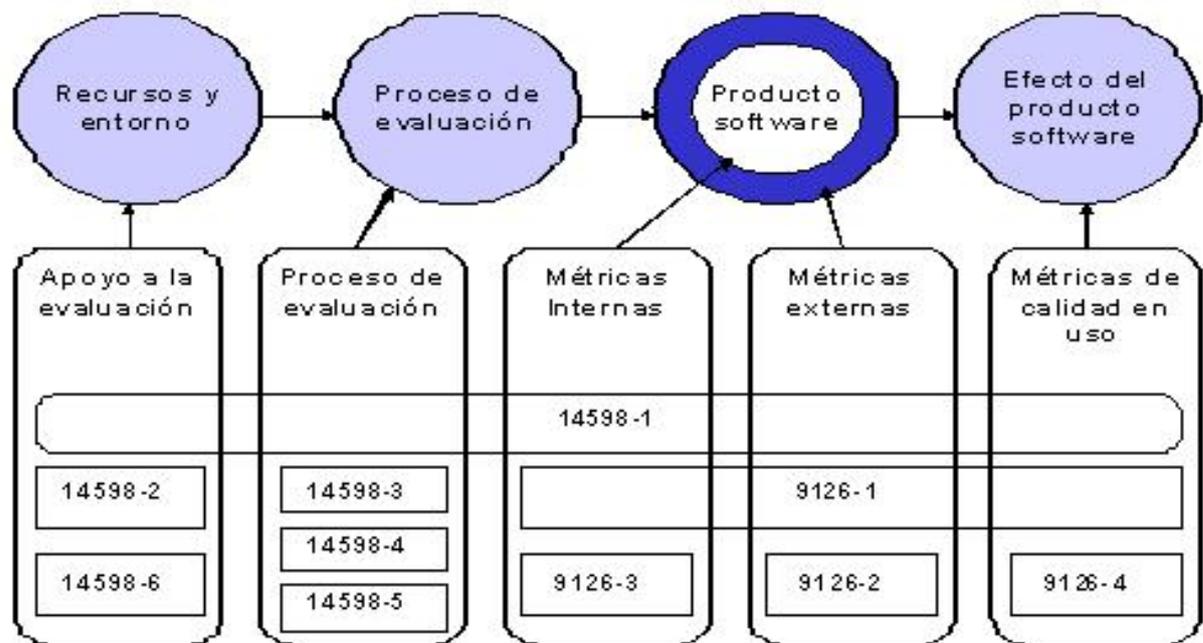
El estándar se divide en cuatro partes que trata, respectivamente, los temas siguientes: modelo de la calidad interna y externa; métricas externas; métricas internas; y modelo de la calidad durante el uso.

Este estándar proviene del modelo establecido en 1977 por McCall y sus colegas, propuesto para especificar la calidad del software. A pesar de que es uno de los más antiguos se ha extendido en todo el mundo y de él han derivado muchos otros como el de Boehm (1978) o el SQM (Software Quality Management) de Murine (1988) [Antonio, 2002].

### **1.6.3.1 Modelo de calidad**

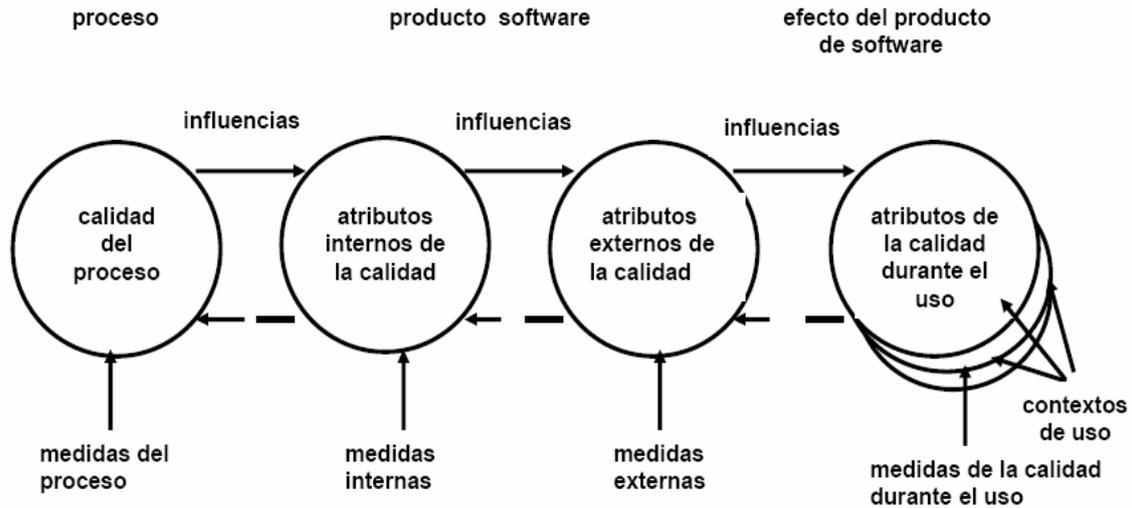
El modelo que describe para la calidad de los productos software se divide en dos partes: a) calidad interna y externa, b) calidad durante el uso. Se especifican seis características para la calidad interna y externa, que son además divididas en sub-características que se manifiestan externamente cuando el software se usa como una parte del sistema computarizado, y son un resultado de los atributos o cualidades internos del software. Para la calidad durante el uso se definen cuatro características. La calidad durante el uso es el efecto combinado que percibe el usuario de la calidad interna y externa del software. [NC-ISO/IEC 9126-1, 2005]

La interrelación entre las normas ISO/IEC 9126 y las normas de la serie ISO/IEC 14598 se muestra en la Figura 4.



**Figura 4. Relación entre las normas de las series ISO/IEC 9126 e ISO/IEC 14598**

La calidad de cualquiera de los procesos del ciclo de vida, contribuye a mejorar la calidad del producto, y esta a su vez contribuye a mejorar la calidad en el uso. Por consiguiente, evaluar y mejorar un proceso es un medio para mejorar la calidad del producto; la evaluación y mejora de la calidad del producto son una vía para mejorar la calidad durante el uso (Ver Figura 5). De igual modo, la evaluación de la calidad durante el uso permite la retroalimentación para mejorar un producto, y cuando se produce la evaluación permite la retroalimentación para mejorar un proceso. [NC-ISO/IEC 9126-1, 2005]



**Figura 5. La calidad en el ciclo de vida**

Como muestra la Figura 5, las métricas internas pueden ser aplicadas a los productos intermedios que se desarrollan a lo largo del ciclo de vida de desarrollo de un producto software, tales como solicitud de propuesta, especificación de requisitos, especificaciones de diseño o código fuente. Las métricas internas le proporcionan a los desarrolladores la habilidad de medir la calidad de estos productos intermedios, con lo cual se puede predecir la calidad del producto final. Esto le permite a los desarrolladores identificar los problemas que afecten la calidad e iniciar las acciones correctivas en las etapas tempranas del ciclo de vida de desarrollo del producto. [ISO/IEC 9126-2, 2003]

Por su lado, las métricas externas pueden ser usadas para medir la calidad del producto software a través de la medición del comportamiento del sistema del cual el software forma parte (Figura5). Las métricas externas solo pueden ser usadas durante las etapas de pruebas del proceso ciclo de vida y durante cualquier otra etapa operacional. [ISO/IEC 9126-2, 2003]

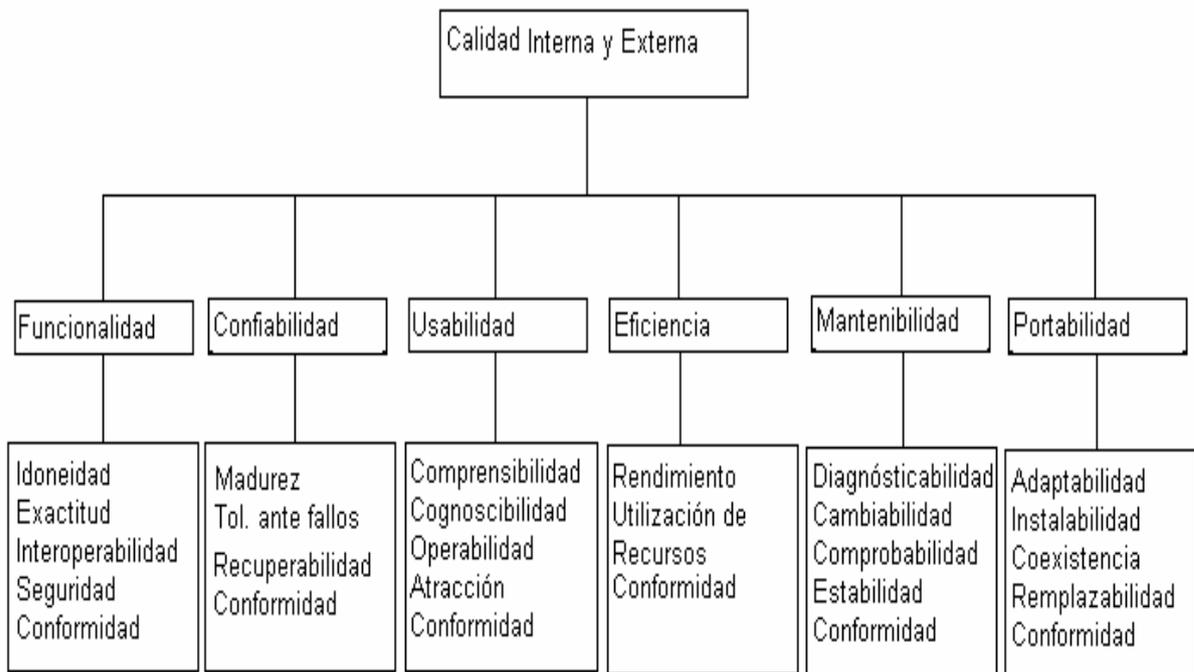
Por último, las métricas de calidad en uso (Figura 5) miden si un producto resuelve las necesidades de usuarios específicos para alcanzar metas específicas con eficacia, productividad, seguridad y satisfacción en un contexto específico de uso. Esto solo puede lograrse en un entorno real del sistema. [ISO/IEC 9126-2, 2003]

Las necesidades de calidad del usuario pueden ser especificadas como requisitos de calidad a través de las métricas de calidad en uso, las métricas externas y algunas veces las métricas internas. Estos requisitos especificados por las métricas deben ser usados como criterios cuando el producto se evalúa. [ISO/IEC 9126-2, 2003]

Es recomendable usar métricas internas que tengan una relación lo más fuerte posible con los objetivos de las métricas externas, así ellas pueden ser usadas para predecir los valores de las métricas externas. Sin embargo, a menudo es difícil diseñar un modelo teórico riguroso que proporcione una relación fuerte entre las métricas internas y externas. [ISO/IEC 9126-2, 2003]

### 1.6.3.2 Modelo para la calidad interna y externa

Este modelo se ha desarrollado en un intento de identificar los atributos más importantes para la calidad interna y externa en un producto software. El modelo identifica seis características claves de calidad [NC-ISO/IEC 9126-1, 2005] donde cada una de ellas se descomponen en un conjunto de sub-características como se muestra en la Figura 6.



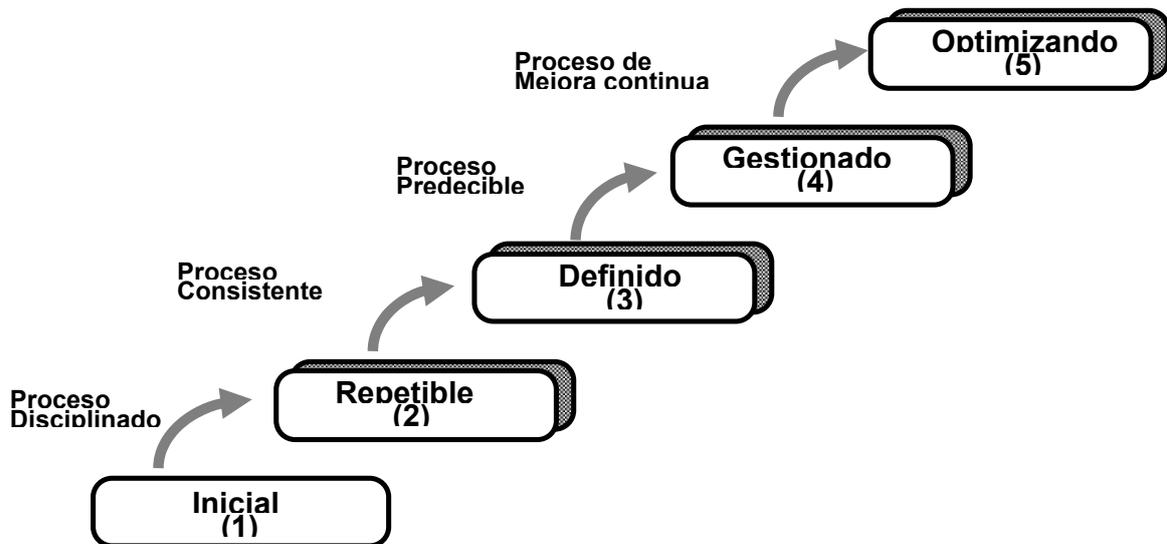
**Figura 6. Modelo para la calidad Interna y Externa**

### 1.6.3.3 Métricas Externas

Las métricas externas usan valores de un producto del software derivados de las medidas del comportamiento del sistema del que es parte al probar, operar u observar el software o sistema ejecutable. Estos valores se emplean como base de la medición para la posterior evaluación del software. Antes de adquirir o usar un producto del software, el mismo debe evaluarse usando métricas basadas en objetivos comerciales relacionados al uso, explotación y gestión del producto en un ambiente organizacional y técnico especificado. Estas son las métricas externas primarias y de ellas se da una relación en las Tablas contentivas de las métricas de la ISO/IEC TR 9126-2. Las mismas constituyen una ventaja para los usuarios, evaluadores, verificadores, y diseñadores pues le permiten medir la calidad del producto de software a través de la medición del comportamiento del sistema del cual él forma parte, así como evaluar la calidad del producto de software durante las pruebas o la operación. [NC ISO/IEC 9126-1, 2005]

Se recomienda utilizar métricas internas que tengan una relación tan fuerte como sea posible con las métricas externas planificadas, para que aquellas puedan usarse para predecir los valores de las métricas externas. Por supuesto es generalmente difícil de diseñar un modelo teórico riguroso que proporcione una relación sólida entre las métricas internas y las externas. [NC-ISO/IEC 9126-1, 2005]

En el presente trabajo se adoptan las métricas externas correspondientes a las características funcionalidad, confiabilidad y usabilidad, fundamentales para un **primer nivel de madurez**, se arriba a una Guía para la aplicación de métricas y la interpretación de las mismas con los fines de la evaluación de los productos de software.



**Fig 7. Niveles de maduración**

Un nivel de madurez es un sistema evolutivo y bien definido para alcanzar el proceso de madurez de software. Cada nivel de madurez tiene dentro de si mismo parámetros que permiten la mejora continua. Alcanzar cierto nivel significa seguir en busca de mejores prácticas, lograr un producto controlado, verificable, validado, medido y a la vez mantener los logros alcanzados. [Cueva 1999]

## **1. Las métricas para la medición de la característica funcionalidad**

Las métricas externas de funcionalidad deben ser capaces de medir un atributo como es el comportamiento funcional del sistema en el cual el software está presente. Estas son:

### **1.1 Métricas de idoneidad**

Las métricas externas de idoneidad deben ser capaces de medir un atributo como es la ocurrencia de un funcionamiento insatisfactorio o la ocurrencia de una operación insatisfactoria.

Un funcionamiento u operación insatisfactoria puede ser:

- Funcionamiento u operación que no se desempeña de la forma especificada en el Manual de usuario o la especificación de requisitos.
- Funcionamiento u operación que no provee una salida aceptable o razonable al tomar en consideración un objetivo específico de las tareas del usuario.

## **1.2 Métricas de exactitud**

Las métricas externas de precisión deben ser capaces de medir un atributo como es la frecuencia con que los usuarios se encuentren con la ocurrencia de una falta de exactitud o de precisión, como pueden ser:

- Resultados incorrecto o imprecisos causados por datos inadecuados; por ejemplo, un dato con pocos dígitos significativos para un cálculo de precisión.
- Inconsistencia entre el procedimiento de operación actual y el descrito en el manual de operación.
- Diferencias entre el resultado actual y el razonablemente esperado producto de una tarea ejecutada durante la operación.

## **1.3 Métricas de interoperabilidad**

Las métricas externas de interoperabilidad deben ser capaces de medir un atributo como es el número de funciones o la ocurrencia de la menor incomunicación que involucre a datos y comandos o instrucciones que sean transferidos entre el producto de software y otros sistemas, otros productos de software u otros equipos a los cuales está conectado.

## **1.4 Métricas de seguridad (informática)**

Las métricas externas de seguridad (de la información o informática) no se han incluido para el primer nivel de maduración objeto del presente trabajo.

## **1.5 Métricas de conformidad de la funcionalidad**

Las métricas externas de conformidad de la funcionalidad deben ser capaces de medir un atributo como lo es el número de funciones con dificultades en la conformidad (o la ocurrencia de problemas de conformidad) con las regulaciones, normas u otras convenciones relacionadas, lo cual haga que el producto de software falle en adherirse a las mismas. Estas métricas no se incluyen para el primer nivel de maduración objeto de este trabajo.

## **2. Las métricas para la medición de la característica confiabilidad**

Las métricas externas de confiabilidad deben ser capaces de medir atributos relacionados con el comportamiento del sistema del cual el software forma parte durante la ejecución de las pruebas para indicar la magnitud de la confiabilidad, o sea, seguridad de funcionamiento del software durante la operación del sistema, con las que en la mayor parte de los casos no se distingue entre el software y el sistema. Ellas son:

### **2.1 Métricas de madurez**

Las métricas externas de madurez deben ser capaces de medir un atributo como la exención de fallas en el software, causados por la ocurrencia de fallos existentes en el propio software.

### **2.2 Métricas de tolerancia ante fallos**

Las métricas externas de tolerancia ante fallos deben estar relacionadas con la capacidad del software de mantener un nivel de ejecución específico en casos de fallos de operación, o se infrinjan las interfaces especificadas.

### **2.3 Métricas de recuperabilidad**

Las métricas externas de recuperabilidad deben ser capaces de medir aquellos atributos como son los software y sistemas capaces de reestablecer su nivel adecuado de ejecución y recuperar los datos directamente afectados en casos de fallos totales.

## **2.4 Métricas de conformidad de la confiabilidad**

Las métricas externas de conformidad de la confiabilidad deben ser capaces de medir un atributo como lo es la cantidad de funciones con dificultades en la conformidad, o la ocurrencia de problemas de conformidad con las regulaciones, normas u otras convenciones relacionadas con la confiabilidad o seguridad de funcionamiento. Dichas métricas no se incluyen para un primer nivel de maduración del presente trabajo.

## **3. Las métricas para la medición de la característica usabilidad**

Las métricas externas de usabilidad miden la dimensión con que el software puede ser comprendido, estudiado, operado, atractivo y concordante con las regulaciones y guías relativas a la usabilidad.

Resulta recomendable que la evaluación de estas métricas se haga por un grupo (7, 8, aunque menores pueden obtener información de utilidad) de usuarios o evaluadores usuarios simulados o clonados (pero representativos de un rango de usuarios) sin que reciban asistencia externa alguna. A continuación se brindan las que en una primera etapa serán objeto de utilización.

### **3.1 Métricas de comprensibilidad**

Las métricas externas de comprensibilidad deben ser capaces de valorar cómo un nuevo usuario podría comprender:

- Si el software es idóneo para la aplicación a la cual lo destina.
- Cómo el software puede ser usado para una tarea en particular.

### **3.2 Métricas de cognoscibilidad**

Las métricas externas de cognoscibilidad (para medir el grado en que puede ser estudiado) y las de operabilidad (para medir el grado en que puede ser implementado y operado) emplean métodos de aplicación eminentemente de usuarios y no son idóneas para el empleo por terceros en una evaluación de certificación, por lo que no se abordan en el presente trabajo.

### **3.3 Métricas de atracción**

Las métricas externas de atracción deben ser capaces de evaluar la apariencia del software, y van a estar influenciadas por factores tales como el color en la pantalla y su diseño.

### **3.4 Métricas de conformidad de la usabilidad**

Las métricas externas de conformidad de la usabilidad deben ser capaces de evaluar la adherencia del software a las regulaciones, normas, convenciones, guías y estilos relativos a la usabilidad. Estas métricas no se incluyen en este trabajo.

## 1.7 Conclusiones

- A nivel mundial la Industria de Software ha potenciado un gran desarrollo, siendo una de las prioridades lograr la calidad en sus productos y servicios, para lo cual centran sus esfuerzos en la mejora de los procesos de desarrollo teniendo en cuenta modelos, normas, estándares.
- Uno de los factores que influyen en la calidad del software es la utilización de mediciones que permitan analizar, evaluar, predecir y mejorar los procesos y productos, proporcionando información valiosa para la toma de decisiones y la evaluación del estado en que se encuentran los objetivos fijados.
- La **ISO/IEC TR 9126-2** proporciona un conjunto de métricas a través de las cuales se podrá comenzar a evaluar los productos de software a partir de elementos cuantitativos y no solos con elementos cualitativos o por detección de defectos como se ha venido haciendo hasta el presente.

# 2

## CAPÍTULO DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

### 2.1 Introducción

En este capítulo se realiza la descripción de la propuesta que tiene como objetivo el presente trabajo, para ello se describen los roles y responsabilidad de las personas implicadas en el proceso de evaluación, se definen las fases necesarias para establecer una evaluación con calidad, dentro de estos se adaptan las métricas estandarizadas internacionalmente seleccionadas para el proceso de evaluación y se establecen los artefactos que se utilizarán para la recogida de información.

### 2.2 Objetivos de la Propuesta de Solución

Proporcionar una guía para un proceso de evaluación que facilite:

- Una selección de métricas, tanto preliminar para un primer nivel de madurez del grupo evaluador o entidad de software, como dinámica durante el proceso de evaluación, y orientar su mejor aplicación.
- Interpretar el resultado de la medición de cada una de ellas.
- A partir de los resultados de todas las métricas que se hayan aplicado, desarrollar un método de valoración de resultados para el producto de software en su conjunto, obtenido con una técnica de trabajo en grupo.

## 2.3 Roles y responsabilidad de los participantes en el proceso de evaluación

Teniendo en cuenta las funciones a realizar en todo el proceso de evaluación y la necesidad participativa de especialistas en el mismo, con el objetivo de optimizar las tareas se definen los siguientes roles y responsabilidades involucradas:

- Líder del Equipo de Evaluación y Calidad del Software: Responsable de la evaluación, debe estar capacitado y autorizado por el Asesor de Calidad de la Entidad y debe asegurar que:
  - ✓ El proceso de evaluación sea realizado conforme a la Guía propuesta y que los resultados de ésta sean representativos.
  - ✓ Los datos de la evaluación sean registrados en el formato adecuado para la evaluación y conforme con lo indicado en la guía.
- Equipo Evaluador: Personas con experiencias en Ingeniería de Software, sólidos conocimientos de la guía, el número de integrantes dependerá de la complejidad del proyecto o la urgencia con que se realice la evaluación, estos realizan las siguientes actividades, recolección de información, realización de la evaluación e información del estado actual del proyecto. Los integrantes serán:
  - ✓ Líder evaluador de la característica Funcionalidad.
  - ✓ Líder evaluador de la característica Confiabilidad.
  - ✓ Líder evaluador de la característica Usabilidad.
- Coordinador del proyecto: Miembro del proyecto encargado de representarlo en la evaluación.

## 2.4 Selección de las métricas externas

Luego de realizar un detallado análisis de las métricas externas primarias contenidas en las Tablas contentivas de métricas de la ISO/IEC TR 9126-2 se ha procedido a realizar una selección para propiciar su aplicación en la

práctica, de modo escalonado y por niveles de madurez de la organización o entidad donde se implanten.

### **2.4.1 Criterios utilizados para la selección de las métricas externas**

Para seleccionar las métricas se efectuaron varios pasos los cuales se especifican a continuación:

- **1er Paso Seleccionar las características a evaluar:**
  - ✓ Para este primer nivel de madurez fueron seleccionadas las características Funcionalidad, Confiabilidad y Usabilidad ya que la NC ISO/IEC 12119 Tecnología de la Información-Paquetes de Software-Requisitos de Calidad y Pruebas (ISO 12119: 1994, IDT), plantea que la evaluación de las 3 características mencionadas anteriormente no se puede dejar de realizar por ser esenciales en el proceso de evaluación de Calidad.
- **2do Paso Seleccionar las sub-características de las características seleccionadas:**

Fueron seleccionadas de la característica Funcionalidad las sub-características:

- ✓ Idoneidad
- ✓ Exactitud
- ✓ Interoperabilidad

Fueron elegidas de la característica Confiabilidad las sub-características:

- ✓ Madurez
- ✓ Tolerancia ante fallos
- ✓ Recuperabilidad

Fueron escogidas de la característica Usabilidad las sub-características:

- ✓ Comprensibilidad
- ✓ Atracción

•**3er Paso Seleccionar las Métricas de las sub-características seleccionadas:**

✓ Se escogieron la gran mayoría de las métricas contenidas en cada una de las sub-características seleccionadas, las cuales se describen en el sub-epígrafe 2.4.2; algunas métricas no fueron elegidas por su complejidad. En total se hizo una selección de 26 métricas.

## **2.4.2 Métricas externas seleccionadas**

A continuación se detallan las métricas que han sido seleccionadas del conjunto de métricas estandarizadas internacionalmente definidas en la ISO/IEC TR 9126-2. En el Anexo 2 se presenta una tabla resumen con las métricas propuestas.

### **2.4.2.1 Métricas de Funcionalidad**

#### **1.1 Idoneidad**

##### **a) Adecuación funcional**

Se propone la métrica **(1.1.a)** que permite hacer un análisis de cuán adecuada es la función evaluada.

Se define la métrica como:

$$X = 1 - A/B$$

Donde:

**A** -Número de funciones en las cuales se detectaron problemas en la evaluación.

**B** -Número de funciones evaluadas.

## **b) Completitud de la implementación funcional**

Del conjunto de métricas que presenta la sub-característica **Idoneidad**, también se propone la métrica **(1.1.b)** que permite realizar una evaluación de cuán completa ha sido la implementación y su conformidad con la especificación de requisitos.

Se define la métrica como:

$$X = 1 - A/B$$

Donde:

**A** - Número de funciones perdidas detectadas en la evaluación.

**B** - Número de funciones descritas en la especificación de requisitos.

## **c) Cobertura de la implementación funcional**

Además se propone la métrica **(1.1.c)** que permite hacer un análisis de cuán correcta ha sido la implementación funcional.

Se define la métrica como:

$$X = 1 - A/B$$

Donde:

**A** - Número de funciones incorrectamente implementadas o funciones perdidas detectadas.

**B** - Número de funciones descritas en la especificación de requisitos.

## **1.2 Exactitud**

### **a) Exactitud esperada**

Se incluye en la guía la métrica **(1.2.a)** que permite hacer un análisis de las diferencias entre los resultados actuales y los razonablemente esperados.

Se define la métrica como:

$$X = A/T$$

Donde:

**A** - Número de casos encontrados con diferencias entre los resultados razonablemente esperados y aquellos resultantes más allá de lo permisible.

**T** - Tiempo de operación.

### **1.3 Interoperabilidad**

#### **a) Intercambiabilidad de datos, en base a su formato**

También se propone emplear la métrica **(1.3.a)** que permite realizar un análisis de cuán correctamente ha sido implementado el intercambio de funciones de interfaces para una transferencia de datos específica.

Se define la métrica como:

$$X = A / B$$

Donde:

**A** - Número de formatos de datos intercambiados exitosamente con otros software o sistemas durante las pruebas del intercambio de datos.

**B** - Número total de formatos de datos a intercambiar.

#### **b) Intercambiabilidad de datos, en base al éxito del intento**

Se propone la métrica **(1.3.b.1)** que permite hacer un análisis de cuán frecuentemente falló el intento de intercambio de datos entre el software objeto de la prueba y otro software.

Se define la métrica como:

$$1) X = 1 - A/B$$

Donde:

**A** - Número de casos en que se falló al proceder a un intercambio de datos con otros software o sistemas.

**B** - Número de casos en que se intentó proceder a un intercambio de datos.

Además se propone la métrica **(1.3.b.2)** que permite hacer un análisis de cuán frecuentemente es satisfactoria la transferencia de datos entre el software objeto de la prueba y otro.

Se define la métrica como:

$$2) Y = A / T$$

Donde:

**A** - Número de casos en que se falló al proceder a un intercambio de datos con otros software o sistemas.

**T** - Período de tiempo de operación.

## **2.4.2.2 Métricas de Confiabilidad**

### **2.1 Madurez**

#### **a) Latencia estimada de la intensidad de fallos**

Se propone la métrica **(2.1.a)** que permite determinar cuántos problemas aún existen que pueden emerger como futuros fallos.

Se define la métrica como:

$$X = (ABS (A1- A2)) / B$$

Donde:

**ABS ()** - Valor absoluto.

**A1** - Número total de fallos latentes predecibles en el producto de software.

**A2** - Número total de fallos detectados realmente.

**B** - Tamaño del producto.

### **b) Intensidad de fallos totales contra casos de prueba**

También se propone la métrica **(2.1.b)** que permite determinar cuántos fallos totales fueron detectados durante un período de pruebas definido.

Se define la métrica como:

$$X = A1 / A2$$

Donde:

**A1** - Número total de fallos totales detectados.

**A2** - Número de casos de pruebas ejecutados.

### **c) Grado de solución ante fallos totales**

Se propone la métrica **(2.1.c)** que permite determinar cuántas condiciones de fallo total están resueltas.

Se define la métrica como:

$$X = A1 / A2$$

Donde:

**A1** - Número de fallos totales solucionados.

**A2** - Número total de problemas reales detectados.

#### **d) Intensidad de fallos**

Se incluye la métrica **(2.1.d)** que permite determinar cuántos fallos fueron detectados durante un período de pruebas definido.

Se define la métrica como:

$$X = A / B$$

Donde:

**A** - Número total de fallos detectados.

**B** - Tamaño del producto.

#### **e) Erradicación de fallos**

Se propone la métrica **(2.1.e.1)** que permite determinar cuántos fallos han sido corregidos.

Se define la métrica como:

$$1) X = A1 / A2$$

Donde:

**A1** - Número de fallos solucionados.

**A2** - Número total de fallos reales detectados.

También se propone la métrica **(2.1.e.2)**.

Se define la métrica como:

$$2) Y = A1 / A3$$

Donde:

**A1** - Número de fallos solucionados.

**A3** - Número total de fallos latentes pronosticados.

#### **f) Tiempo medio entre fallos totales**

Se propone la métrica **(2.1.f.1)** que permite realizar un análisis de cuán frecuentemente el software fracasa en su operación.

Se define la métrica como:

$$1) X = T1 / A$$

Donde:

**T1** – Tiempo de operación.

**A** - Número total de fallos realmente detectados (fallos totales ocurridos durante el tiempo de operación observado).

Además se propone la métrica **(2.1.f.2)**.

Se define la métrica como:

$$2) Y = T2 / A$$

Donde:

**T2** – Suma de los intervalos de tiempo entre los fallos totales consecutivos producidos.

**A** - Número total de fallos realmente detectados (fallos totales ocurridos durante el tiempo de operación observado).

#### **g) Cobertura de las pruebas**

Se propone la métrica **(2.1.g)** que permite hacer un análisis de cuántos casos de pruebas requeridos han sido ejecutados, detectados durante las pruebas.

Se define la métrica como:

$$X = A / B$$

Donde:

**A** – Número de casos de pruebas que han sido realmente ejecutados, y que representan el escenario de operación durante las pruebas.

**B** – Número de casos de pruebas a ejecutar requeridos para cubrir los requisitos.

#### **h) Madurez de las pruebas**

Se propone la métrica **(2.1.h)** que permite determinar si está bien probado el producto.

Se define la métrica como:

$$X = A / B$$

Donde:

**A** – Número de casos de pruebas que han obtenido un resultado satisfactorio al ser ejecutados o durante su operación.

**B** – Número de casos de pruebas a ejecutar para cubrir los requisitos.

## **2.2 Tolerancia ante fallos**

### **a) Evitación de desastre**

Se propone la métrica **(2.2.a)** que permite hacer un análisis de cuán frecuentemente el producto de software causa un desastre en el ambiente de la producción total.

Se define la métrica como:

$$X = 1 - A / B$$

Donde:

**A** - Número de desastres.

**B** - Número de fallos totales.

## **b) Evitación de operaciones incorrectas**

Se propone la métrica **(2.2.b)** que permite determinar cuántas funciones están implementadas con capacidad de evitación de operaciones incorrectas.

Se define la métrica como:

$$X = A / B$$

Donde:

**A** - Número de ocurrencia de fallos totales críticos o serios evitada.

**B** - Número de casos de pruebas ejecutados a los patrones de operaciones incorrectas (casi causantes de fallos) durante las pruebas.

## **2.3 Recuperabilidad**

### **a) Grado de disponibilidad**

Se propone la métrica **(2.3.a)** que permite determinar cuán disponible está el sistema para su uso durante.

Se define la métrica como:

$$1) X = (T_o / T_o + T_r)$$

Donde:

**T<sub>o</sub>** – Tiempo de operación.

**T<sub>r</sub>** – Tiempo de reparación.

### **b) Tiempo medio de inactividad**

La métrica **(2.3.b)** también se propone, la cual permite determinar cuál es el tiempo promedio en que el sistema se mantiene no disponible cuando ocurre un fallo total y antes de la arrancada gradual.

Se define la métrica como:

$$X = T / N$$

Donde:

**T** - Tiempo total de inactividad.

**N** - Número de desastres observados.

### **c) Tiempo medio de recuperación**

Se incluye la métrica **(2.3.c)**, esta permite determinar cuál es el tiempo promedio que toma el sistema para completar la recuperación desde el inicio de la recuperación parcial.

Se define la métrica como:

$$X = \text{SUM}(T_n) / N$$

Donde:

**T** - Tiempo de recuperación de la inactividad en cada n oportunidad.

**N** - Número de oportunidades en que el sistema entró en recuperación.

### **d) Recargabilidad**

Se propone la métrica **(2.3.d)** que permite hacer un análisis de cuán frecuentemente el sistema logra recargar luego de una ejecución perdida proveyendo nuevamente los servicios a los usuarios en el plazo de tiempo previsto.

Se define la métrica como:

$$X = A / B$$

Donde:

**A** - Número de veces que se provocó el reinicio o recarga **en el plazo de tiempo previsto** en la prueba especificada o en la implantación.

**B** - Número total de veces que se provocó el reinicio o recarga durante la prueba especificada o la implantación.

#### **e) Restaurabilidad**

Se propone la métrica **(2.3.e)** que permite hacer un análisis de cuán capaz es el producto de auto restaurarse luego de un evento anormal o una solicitud.

Se define la métrica como:

$$X = A / B$$

Donde:

**A** – Número de casos de restauración exitosos.

**B** – Número de casos de restauración probados por los requisitos.

#### **f) Efectividad de la restauración**

Se propone la métrica **(2.3.f)** que permite hacer un análisis de cuán efectiva es la capacidad de restauración.

Se define la métrica como:

$$X = A / B$$

Donde:

**A** – Número de casos de restauración exitosos en el plazo de tiempo previsto.

**B** – Número de casos de restauración ejecutados.

## 2.4.2.3 Métricas de Usabilidad

### 3.1 Comprensibilidad

#### a) Integridad de la descripción de producto

Se propone la métrica **(3.1.a)** que permite determinar qué proporción de funciones (o tipos de funciones) son comprendidas después de leer la descripción del producto.

Se define la métrica como:

$$X = A / B$$

Donde:

**A** - Número de funciones comprendidas.

**B** - Número total de funciones.

#### b) Accesibilidad a demos

Se propone la métrica **(3.1.b)** que permite hacer un análisis de a qué proporción de demos/tutoriales pueden acceder los usuarios.

Se define la métrica como:

$$X = A / B$$

Donde:

**A** - Número de demos/tutoriales a los que pueden acceder los usuarios exitosamente.

**B** - Número total de demos/tutoriales a los que se puede acceder.

### **c) Comprensibilidad de entradas y salidas**

También se propone la métrica **(3.1.c)** que permite hacer un análisis si pueden los usuarios comprender lo que se requiere como entrada y lo que suministra el sistema de software como salida.

Se define la métrica como:

$$X = A / B$$

Donde:

**A** - Número de elementos de entrada y que suministra el sistema de software como salida comprendidas correctamente.

**B** - Número total de elementos de entrada y que suministra el sistema de software como salida proporcionadas por el interfaz.

### **3.2 Atracción**

#### **a) Adaptabilidad de la apariencia de la interfaz**

Se propone la métrica **(3.2.b)** que permite hacer un análisis de qué proporción de los elementos de la interfaz puede ser, por su apariencia, adaptado por el usuario para la satisfacción del mismo.

Se define la métrica como:

$$X = A / B$$

Donde:

**A** - Número de elementos de la interfaz del sistema cuya apariencia puede ser adaptada por el usuario.

**B** - Número de elementos de la interfaz del sistema cuya apariencia querría adaptar el usuario.

## 2.5 Descripción de las fases del Proceso de evaluación

En este sub-epígrafe se definen las diferentes fases que componen el proceso de evaluación sugerido para la aplicación de las métricas, el cual consta de 5 fases (Ver Figura 7), a continuación se detallan cada una de ellas:

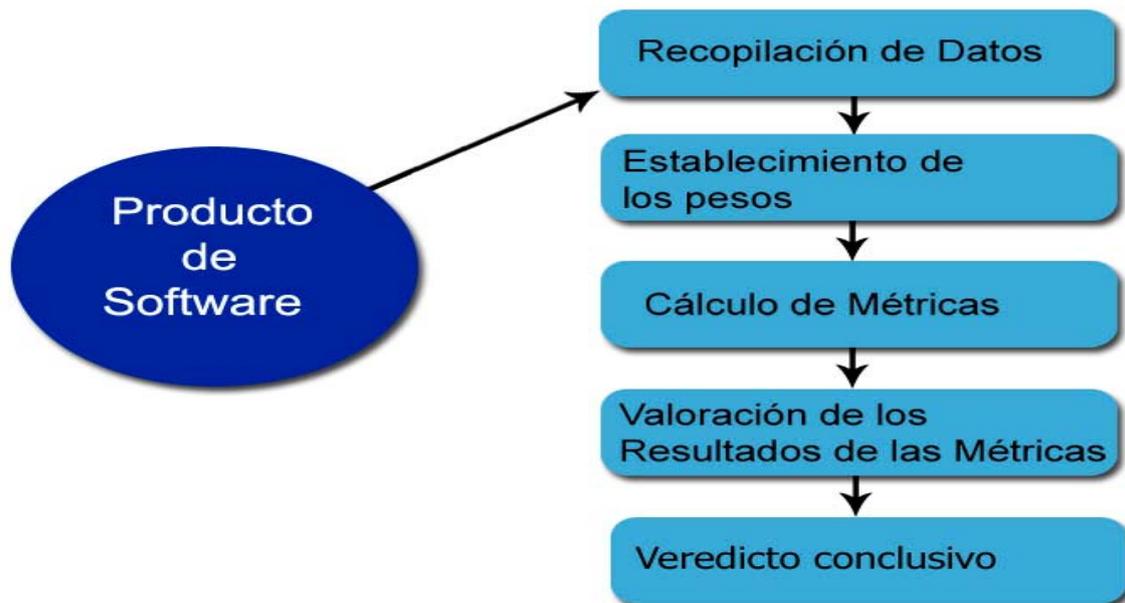


Figura 8. Fases del Proceso de evaluación

### 2.5.1 Recopilación de Datos

En esta fase se recopilan los datos de las variables implicadas en las fórmulas de medición de las métricas, para esto se brinda una **plantilla # 1** (Ver Anexo 3) que es una tabla con el nombre de las métricas y el significado de sus correspondientes variables, que deben ser obtenidas en los procesos de pruebas.

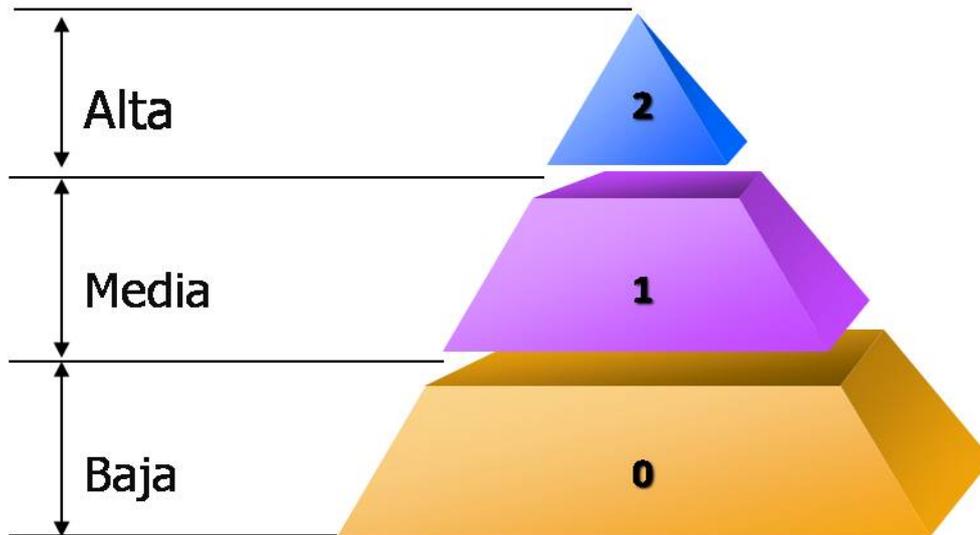
A continuación se muestra un ejemplo de un fragmento de la plantilla # 1:

**Tabla 1. Datos de las variables contenidas en las métricas**

<b>Métricas de Idoneidad</b>		
<b>Nombre de la métrica</b>	<b>Variables</b>	<b>Datos</b>
<b>a) Adecuación funcional</b>	<b>A</b> - Número de funciones en las cuales se detectaron problemas en la evaluación. <b>B</b> - Número de funciones evaluadas.	<b>A = 5</b> <b>B = 8</b>
<b>Métrica de Exactitud</b>		
<b>Nombre de la métrica</b>	<b>Variables</b>	<b>Datos</b>
<b>a) Exactitud esperada</b>	<b>A</b> - Número de casos encontrados con diferencias entre los resultados razonablemente esperados y aquellos resultantes más allá de lo permisible. <b>T</b> - Tiempo de operación.	<b>A = 6</b> <b>T = 9</b>
<b>Métricas de Interoperabilidad</b>		
<b>Nombre de la métrica</b>	<b>Variables</b>	<b>Datos</b>
<b>a) Intercambiabilidad de datos, en base a su formato</b>	<b>A</b> - Número de formatos de datos intercambiados exitosamente con otros software o sistemas durante las pruebas del intercambio de datos. <b>B</b> - Número total de formatos de datos a intercambiar.	<b>A = 4</b> <b>B = 5</b>

## 2.5.2 Establecimiento de los pesos

Después de haber recopilado los datos de las variables se procede al establecimiento de los pesos que posee cada sub-característica en el proyecto de software sometido a evaluación, para ello se ofrece una **plantilla # 2** (Ver Anexo 4) para reflejar los resultados de este análisis. Este peso lo establece el evaluador antes de realizar los cálculos de las métricas, en dependencia de la importancia que tenga la sub-característica que se está evaluando sobre el producto, se sugiere para el peso la siguiente estructura: 2 si la relevancia es alta, 1 si es media y 0 si es baja (Ver Figura 8).



**Figura 9. Estructura para el peso**

A continuación se muestra un ejemplo de un fragmento de la plantilla # 2 para el establecimiento del peso:

**Tabla 2. Establecimiento del peso**

CARACTERÍSTICA	SUBCARACTERÍSTICA	PESOS	MÉTRICAS	NIVEL REQUERIDO	RESULTADO REAL
Funcionalidad	Idoneidad	2	1.1.a) $X = 1 - A/B$	1 ( $0 \leq X \leq 1$ )	
	Exactitud	2	1.2.a) $X = A/T$	0 ( $0 \leq X$ )	
	Interoperabilidad	2	1.3.a) $X = A / B$	1 ( $0 \leq X \leq 1$ )	

### 2.5.3 Cálculo de Métricas

Una vez recopilados los datos y establecido el peso se prosigue con el cálculo de las métricas. En esta fase se utiliza la **plantilla # 2** (Ver Anexo 4) para guardar los resultados del cálculo de métricas.

A continuación se muestra un ejemplo de un fragmento de la plantilla # 2 para el cálculo de las métricas:

Tabla 3. Cálculo de métricas

CARACTERISTICA	SUBCARACTERISTICA	PESOS	METRICAS	NIVEL REQUE- RIDO	RESUL- TADO REAL
Funcionalidad	Idoneidad	2	1.1.a) $X = 1 - A/B$	1 ( $0 \leq X \leq 1$ )	0.375
	Exactitud	2	1.2.a) $X = A/T$	0 ( $0 \leq X$ )	0.66
	Interoperabilidad	2	1.3.a) $X = A / B$	1 ( $0 \leq X \leq 1$ )	0.8

**Escala para cuando el nivel requerido es 1**

Rango	Evaluación	Puntuación
0 - 0,2	Mal	0
0,3 – 0,5	Regular	1
0,6 – 0,7	Bien	2
0,8- 1	Muy Bien	3

**Escala para cuando el nivel requerido es 0**

Rango	Evaluación	Puntuación
0 - 0,2	Muy Bien	3
0,3 – 0,5	Bien	2
0,6 – 0,7	Regular	1
0,8- 1	Mal	0

El resultado del cálculo de la métrica 1.1.a) es 0.375 y el nivel requerido es 1, este resultado del cálculo se ubica en la Tabla de la Escala para cuando el nivel requerido es 1 y la puntuación que se le daría a la sub-característica Idoneidad en la plantilla # 3 es 1 (Regular).

El resultado del cálculo de la métrica 1.2.a) es 0.66 y el nivel requerido es 0, este resultado del cálculo se ubica en la Tabla de la Escala para cuando el nivel requerido es 0 y la puntuación que se le daría a la sub-característica Exactitud en la plantilla # 3 es 1 (Regular).

El resultado del cálculo de la métrica 1.3.a) es 0.8 y el nivel requerido es 1, este resultado del cálculo se ubica en la Tabla de la Escala para cuando el nivel

requerido es 1 y la puntuación que se le daría a la sub-característica Interoperabilidad en la plantilla # 3 es 3 (Muy Bien).

**Nota:** Esta puntuación que se le da a las sub-características se hace teniendo en cuenta el peso de la misma. En el caso de que el peso de una sub-característica sea 0 (Baja Relevancia) entonces se le da una puntuación directa de 3 (Muy Bien). En el caso de que el peso sea 1 o 2 entonces la puntuación será la obtenida después de llevar a las Tablas de las Escalas. En el caso de que a una sub-característica se le evalúe más de una métrica entonces el resultado del cálculo de las métricas evaluadas se ubica en las Tablas de Escalas, según su nivel requerido, se halla el promedio entre los resultados obtenidos después de haber llevado a la escala y este valor del promedio será la puntuación que se le dará a esa sub-característica.

#### **2.5.4 Valoración de los Resultados de las Métricas**

Después de haberse hecho el cálculo de las métricas se hace una valoración de los resultados de las pruebas y el comportamiento de las características de calidad. Es decir se le da una puntuación a cada una de las sub-características según el resultado del cálculo de las métricas de esa sub-característica y el peso que tenga dicha sub-característica en el producto de software que se está evaluando, este peso lo establece el evaluador antes de realizar los cálculos, el resultado del cálculo de las métricas después de haberse llevado a las Tablas de la Escala, según el nivel requerido y el peso se verifican en la **plantilla # 2** (Ver Anexo 4). Luego se hace un resumen individual del resultado de cada característica, esto sería hallar el promedio entre las puntuaciones de las sub-característica correspondientes a cada característica, en caso del promedio haber dado un número decimal se redondea a un número entero. Una vez realizado este procedimiento se llega a una conclusión sobre el grado de conformidad que tiene cada característica sobre el producto que se está evaluando, éste grado de conformidad se determina por el resultado del promedio; y finalmente según el grado de conformidad se emite un criterio de evaluación para dicha característica. Todo este proceso lo hace el evaluador de cada característica a evaluar. Se provee una **plantilla # 3** (Ver Anexo 5) a utilizar en esta fase.

A continuación se muestra un ejemplo de un fragmento de la plantilla # 3 para darle una puntuación a cada una de las sub-características de la característica funcionalidad:

Tabla 4. Puntuación a las sub-características

CARACTERISTICAS Y SUBCARACTERISTICAS DE CALIDAD	PUNTUACION			
	3	2	1	0
<b>1. FUNCIONALIDAD</b>				
<b>1.1 Idoneidad.</b> Capacidad del software para mantener un conjunto apropiado de funciones para las tareas y los objetivos del usuario especificados.			X	
a) Funcionamiento correcto (ausencia de fallos totales – ciclos infinitos, interrupción de la ejecución o salidas abruptas-, errores críticos, errores de ejecución, resultados incorrectos, correspondencia de las descripciones con los objetos, por ejemplo en el nivel de ayuda solicitado o los mensajes de error según el fallo detectado )				
b) Correspondencia de las funciones con los requisitos funcionales de la Descripción del software (especificaciones)				
<b>1.2 Exactitud.</b> Capacidad del software para proporcionar efectos o resultados correctos o convenidos con el grado de exactitud necesario.			X	
a) Exactitud de los cálculos				
<b>1.3 Interoperabilidad.</b> Capacidad del producto de software para interactuar recíprocamente con uno o más sistemas especificados.	X			
a) Funcionamiento correcto al interactuar (ausencia de fallos totales – ciclos infinitos, interrupción de la ejecución o salidas abruptas-, errores críticos, errores de ejecución, resultados incorrectos,)				

Esto es otro ejemplo de la plantilla # 3 para hacer el resumen individual del resultado de la característica funcionalidad:

Después de darle una puntuación a cada sub-característica de la característica Funcionalidad se halla el promedio entre ellas, sería:

$$\text{Promedio} = (\text{Puntuación Sub-característica Idoneidad} + \text{Puntuación Sub-característica Exactitud} + \text{Puntuación Sub-característica Interoperabilidad})/3 = 1 + 1 + 3 = 5/3 = 1.66$$

El resultado del promedio es 1.66 redondeado es 2, por tanto el grado de conformidad es Suficientemente conforme (2) y el criterio de evaluación es Pequeñas modificaciones.

PUNTUACION		GRADO DE CONFORMIDAD	
		3	Conforme
Funcionalidad	1.66	2	Suficientemente conforme
VALOR	2	1	Medianamente conforme
		0	No conforme

CRITERIO DE EVALUACION
<input type="checkbox"/> Sin modificaciones
<input checked="" type="checkbox"/> Pequeñas modificaciones
<input type="checkbox"/> Grandes modificaciones
<input type="checkbox"/> Nueva elaboración

### 2.5.5 Veredicto conclusivo

En esta fase ya se llega a una conclusión final sobre el grado de aceptación que tiene el producto que se está evaluando. Es decir después de haberse hecho un resumen individual del grado de conformidad de cada característica evaluada se llega a un veredicto conclusivo. En dicha fase se halla una puntuación promedia a partir de la individual de cada característica evaluada, en caso que el promedio sea un número decimal se lleva a un número entero. Luego se toma el grado de conformidad por cada característica, es decir por cada sección. Finalmente se llega a un veredicto sobre el grado de aceptación que tiene el producto, este grado puede ser aceptado, diferido o no aceptado. Para esta fase se hará uso de la **plantilla # 4** (Ver Anexo 6).

A continuación se muestra un ejemplo de un fragmento de la plantilla # 4 para hallar una Puntuación promedio a partir de la individual de cada característica:

Suponiendo que se hizo el mismo procedimiento de la característica Funcionalidad para las características Confiabilidad y Usabilidad, cada una con un promedio individual de 3 y 2 respectivamente. Una vez obtenidos los

promedios individuales entonces se puede hallar el promedio de las 3 características, sería:

$$\text{Promedio} = (\text{Puntación Característica Funcionalidad} + \text{Puntación Característica Confiabilidad} + \text{Puntación Característica Usabilidad})/3 = 2 + 3 + 2 = 7/3 = 2.33$$

El resultado del promedio es 2.33 redondeado es 2.

### PUNTUACION PROMEDIADA A PARTIR DE LA INDIVIDUAL

<b>Funcionalidad</b>	2
<b>Confiabilidad</b>	3
<b>Usabilidad</b>	2
<b>VALOR*</b>	2

<b>GRADOS DE CONFORMIDAD POR SECCIONES (poner una X)</b>		
<b>Sección A</b>		<input type="checkbox"/> Conforme <input checked="" type="checkbox"/> Suficientemente conforme <input type="checkbox"/> Medianamente conforme <input type="checkbox"/> No conforme
<b>Sección B</b>		<input checked="" type="checkbox"/> Conforme <input type="checkbox"/> Suficientemente conforme <input type="checkbox"/> Medianamente conforme <input type="checkbox"/> No conforme
<b>Sección C</b>		<input type="checkbox"/> Conforme <input checked="" type="checkbox"/> Suficientemente conforme <input type="checkbox"/> Medianamente conforme <input type="checkbox"/> No conforme

<b>VEREDICTO</b>	
	<b>ACEPTADO</b>
	<b>DIFERIDO</b>
	<b>NO ACEPTADO</b>

## 2.6 Conclusiones

En este capítulo se desarrolló la propuesta de solución, obteniéndose una guía para la aplicación de las métricas propuestas. Para su creación se definieron los objetivos de la misma, los cuales se enfocaron en la obtención de calidad del producto final; se identificaron los roles y responsabilidades de cada rol, los cuales se adaptaron a las características claves que debe poseer un producto de software con calidad; se hizo la selección de las métricas; se definieron las fases que componen la guía del proceso de evaluación para la aplicación de las métricas seleccionadas, en las cuales se especificaron los artefactos que se utilizarán para la recogida de información.

El desarrollo de la propuesta fue guiada por la ISO/IEC TR 9126-2 existente en el mundo y que tiene resultados a nivel internacional, por lo que se espera que si se aplica la misma se obtengan las siguientes ventajas:

- Fácil comprensión.
- Aumento de la calidad del producto final.
- Determinación del nivel de calidad real del producto.

# 3

## CAPÍTULO VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

### 3.1 Introducción

Para realizar la validación de la propuesta de solución, se utilizó el Método de Experto el cuál se basa en la evaluación cuantitativa de criterios definidos, que permite realizar un estudio por expertos para determinar si se acepta o no la propuesta analizada.

### 3.2 Guía para la validación

Para llevar a cabo el desarrollo de la validación se efectuaron un conjunto de pasos, los cuales se detallan a continuación:

1. Se elaboran los criterios de evaluación que fueron utilizados en la validación, estos se agrupan por categorías.

#### **Grupo No 1: Criterios de mérito científico.**

1. Valor científico de la propuesta.
2. Calidad de la investigación.
3. Contribución científica.
4. Responsabilidad científica y profesionalidad de los investigadores.

#### **Grupo No 2: Criterios de implantación.**

5. Necesidad de empleo de la propuesta.
6. Posibilidades de aplicación.

### **Grupo No 3: Criterios de flexibilidad.**

7. Adaptabilidad a entidades dedicadas a evaluar la calidad de los productos de software.
8. Capacidad del proceso de evaluación para la admisión de cambios que impliquen mejoras.

### **Grupo No 4. Criterios de impacto.**

9. Impacto en el área para la cual está destinada la guía.
10. Organización en el proceso de desarrollo.

2. Se determina el peso relativo de cada grupo de criterios de acuerdo al por ciento que representa cada grupo del total y los intereses a evaluar.

Grupo No.1.....	40
Grupo No.2.....	20
Grupo No.3.....	20
Grupo No.4.....	20

3. Se realiza una selección de 7 expertos en la cual se tiene en cuenta su especialidad, grado científico y currículum.

4. Se le solicita a los expertos que den una evaluación de cada uno de los criterios, teniendo en cuenta que la suma del valor dado por parte de los expertos a cada criterio de un grupo no exceda del peso relativo asignado a este. Para recoger la información anterior se definió un modelo, el cual se expone en el anexo 7 del trabajo (Ver Anexo 7).

5. Después de recibir los valores del peso relativo de cada criterio se construye la Tabla donde se guarda el resultado del trabajo de los expertos (Ver Tabla 5).

**Tabla 5. Resultado del trabajo de expertos**

G	C/E	E1	E2	E3	E4	E5	E6	E7	Ep
40	C1	8	10	10	5	8	10	10	8.71
	C2	15	10	10	15	11	9	11	11.6
	C3	8	10	10	5	10	10	8	8.71
	C4	9	10	10	15	11	11	11	11
20	C5	15	10	10	10	10	11	10	10.8
	C6	5	10	10	10	10	9	10	9.14
20	C7	10	15	15	15	11	10	11	12.4
	C8	10	5	5	5	9	10	9	7.57
20	C9	10	10	10	10	10	10	10	10
	C10	10	10	10	10	10	10	10	10
T		100	100	100	100	100	100	100	100

6. Se verifica la consistencia en el trabajo de los expertos, para lo que se utiliza el coeficiente de concordancia de Kendall y el estadígrafo Chi cuadrado ( $\chi^2$ ).

Para esto se sigue el procedimiento siguiente:

- Sea C el número de criterios que van a evaluarse y (E) el número de expertos que realizan la evaluación.
- Para cada criterio se determina:
  - ✓  $\sum E$ : Sumatoria del peso dado por cada experto.
  - ✓ Ep: Puntuación promedio del peso dado por cada experto.
  - ✓  $M\sum E$ : media de los  $\sum E$ .
  - ✓  $\Delta C$ : Diferencia entre  $\sum E$  y  $M\sum E$ .

- Se determina la desviación de la media, que posteriormente se eleva al cuadrado para obtener la dispersión (S) por la expresión:

$$S = \sum (\sum E - \sum \sum E / C)^2$$

**Tabla 6. Cálculo de la Dispersión (S) para hallar la concordancia entre los expertos**

	$\sum E$	$\sum E / C$	$\sum E - \sum \sum E / C$	$(\sum E - \sum \sum E / C)^2$
<b>C1</b>	61	6.1	-9	81
<b>C2</b>	81	8.1	11	121
<b>C3</b>	61	6.1	-9	81
<b>C4</b>	77	7.7	7	49
<b>C5</b>	76	7.6	6	36
<b>C6</b>	64	6.4	-6	36
<b>C7</b>	87	8.7	17	289
<b>C8</b>	53	5.3	-17	289
<b>C9</b>	70	7	0	0
<b>C10</b>	70	7	0	0
$\sum \sum E / C$	-	70	-	-
$S = \sum (\sum E - \sum \sum E / C)^2$	-	-	-	982

- Conociendo la dispersión se puede calcular el coeficiente de concordancia de Kendall (W):

$$W = S / E^2 (C^3 - C) / 12$$

- El coeficiente de concordancia de Kendall permite calcular el Chi cuadrado real:

$$X^2 = E (C-1) W$$

- Los valores obtenidos se muestran en la Tabla 7.

**Tabla 7. Tabla para el cálculo de concordancia de Kendall**

Expertos/Criterios	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	E <sub>4</sub>	E <sub>5</sub>	E <sub>6</sub>	E <sub>7</sub>	∑E	E <sub>p</sub>	ΔC	ΔC <sup>2</sup>
C <sub>1</sub>	8	10	10	5	8	10	10	61	8.71	9	81
C <sub>2</sub>	15	10	10	15	11	9	11	81	11.6	11	121
C <sub>3</sub>	8	10	10	5	10	10	8	61	8.71	9	81
C <sub>4</sub>	9	10	10	15	11	11	11	77	11	7	49
C <sub>5</sub>	15	10	10	10	10	11	10	76	10.8	6	36
C <sub>6</sub>	5	10	10	10	10	9	10	64	9.14	6	36
C <sub>7</sub>	10	15	15	15	11	10	11	87	12.4	17	289
C <sub>8</sub>	10	5	5	5	9	10	9	53	7.57	17	289
C <sub>9</sub>	10	10	10	10	10	10	10	70	10	0	0
C <sub>10</sub>	10	10	10	10	10	10	10	70	10	0	0
<b>DC</b>	100	100	100	100	100	100	100	700	99.93	82	982
<b>M ∑E</b>	70										
<b>W</b>	0.24										
<b>X<sup>2</sup></b>	15.12										
<b>X<sup>2</sup><sub>(α, c-1)</sub></b>	21.6660										

- El Chi cuadrado calculado se compara con el obtenido de las tablas estadísticas y de esta forma se obtiene la siguiente conclusión.

Como se cumple la siguiente condición:

$$X^2_{\text{real}} < X^2_{(\alpha, c-1)}$$

$$15.12 < 21.6660$$

Se puede decir que existe concordancia en el trabajo de expertos.

7. Después de comprobar la consistencia del trabajo de expertos se puede definir el peso relativo de cada criterio (P).

8. Conociendo el peso de cada criterio (P) y la calificación dada por los evaluadores en una escala de 1-5 se puede construir la Tabla 8, para obtener el valor de  $P \times c$ , donde (c), es el criterio promedio concebido por los expertos.

Para recoger la calificación dada por los expertos a cada uno de los criterios se definió un modelo el cual se expone en el anexo 8 del trabajo. (Ver Anexo 8)

**Tabla 8. Tabla de calificación de cada criterio**

Criterios	Calificación (c)					P	P × c
	1	2	3	4	5		
C <sub>1</sub>				X		0.0871	0.3484
C <sub>2</sub>				X		0.116	0.464
C <sub>3</sub>				X		0.0871	0.3484
C <sub>4</sub>					X	0.11	0.55
C <sub>5</sub>					X	0.108	0.54
C <sub>6</sub>					X	0.0914	0.457
C <sub>7</sub>					X	0.124	0.62
C <sub>8</sub>				X		0.0757	0.3008
C <sub>9</sub>					X	0.1	0.5
C <sub>10</sub>				X		0.1	0.4
<b>Total</b>							4.5286

9. Se calcula el Índice de Aceptación del proyecto (IA).

$$IA = \sum (P \times c) / 5$$

$$IA = 0.905$$

10. Por último se determina la probabilidad de éxito de la propuesta, para esto se ubica el Índice de Aceptación (IA) calculado anteriormente en rangos que están ya predefinidos, en dependencia de donde se ubique, será la probabilidad de éxito que tenga la propuesta.

El Índice de Aceptación calculado es 0.905.

**Rangos predefinidos de Índice de Aceptación.**

$IA > 0,7$  Existe alta probabilidad de éxito

$0,7 > IA > 0,5$  Existe probabilidad media de éxito

$0,5 > IA > 0,3$  Probabilidad de éxito baja

$0,3 > IA$  Fracaso seguro

Por lo que existe alta probabilidad de éxito.

## **Principales recomendaciones manifestadas por los expertos**

- ✓ Que se utilicen las métricas dentro del proceso de ingeniería para iniciar las acciones correctivas en las etapas tempranas del ciclo de vida de desarrollo del producto.
- ✓ Seguir trabajando en base a un aumento del número de métricas seleccionadas con sus respectivos criterios sometidos a medición.
- ✓ Continuar explotando los estándares internacionales a fin de lograr perfeccionar la propuesta de solución para un mayor nivel de madurez.
- ✓ Que se prepare esta guía de evaluación con el objetivo de ser aplicada por una determinada entidad evaluadora de calidad de software correspondiente a una específica esfera de trabajo en nuestra sociedad, como bien podría ser el campo de las Ciencias Médicas.

### **3.3 Conclusiones**

En este capítulo se evaluó la guía obtenida mediante el Método de Expertos; el cual permitió analizar los criterios de cada uno de los expertos y determinar el índice de aceptación que tiene la propuesta del presente trabajo, obteniéndose concordancia en el trabajo de expertos y una alta probabilidad de éxito de ser aplicada en las entidades que se dedican a evaluar la calidad de los productos de software.

## **Conclusiones Generales**

En el presente trabajo se obtuvo una guía con los pasos imprescindibles para la aplicación de las métricas propuestas en el proceso de evaluación de la calidad de los productos de software.

Se hizo un estudio de la situación actual en el mundo y en Cuba, respecto a la utilización de las métricas en los procesos de evaluación de la calidad de los productos de software.

Se analizaron métricas estandarizadas internacionalmente, definidas en la ISO/IEC TR 9126-2, seleccionando las métricas a proponer para este primer nivel de madurez.

A partir de todo el trabajo realizado, se obtuvo la propuesta, la cual describe las métricas seleccionadas y el proceso de evaluación para la aplicación de las mismas, facilitando que en futuros análisis sea perfeccionada y aplicada en diversas esferas debido a su carácter dinámico.

Finalmente se hizo la validación de la propuesta, arrojando que la probabilidad de éxito es alta, lo que implica desde el punto de vista teórico, el cumplimiento de la idea a defender planteada y desde el punto de vista práctico, la aplicabilidad de la propuesta en entidades dedicadas a la evaluación de los productos de software.

## Recomendaciones

Los objetivos planteados en el presente trabajo se alcanzaron de manera general, pero se considera que para seguir solucionando los problemas respecto a la calidad del software, se necesita de un proceso investigativo mucho más amplio y este trabajo es sólo el primer nivel de madurez del mismo, teniendo en cuenta esto se pueden hacer las siguientes recomendaciones:

- Hacer talleres, seminarios, conferencias, etc., donde se deje clara la importancia de la aplicación de las métricas en el proceso de evaluación de productos de software.
- Seguir trabajando en la selección de las métricas.
- Que la guía propuesta sea tomada en cuenta y aplicada en las entidades dedicadas a evaluar la calidad de los productos de software, principalmente en el Ministerio de Salud Pública (MINSAP).

## Referencias Bibliográficas

ALLIANCE, B. S. Fuerte crecimiento del mercado del software, 2007. [2007].  
Disponible en: <http://www.bsa.org/espana/press/newsreleases/BSA-destaca-el-fuerte-crecimiento-del-mercado-del-software-europeo-pero-advierte-sobre-nuevos-riesgos-y-amenazas-potenciales.cfm>

de Antonio J, Angélica. Gestión, Control y Garantía de la Calidad del Software, 2002. Disponible en:  
[http://www.inf.uach.cl/rvega/asignaturas/info265/G\\_Calidad.pdf](http://www.inf.uach.cl/rvega/asignaturas/info265/G_Calidad.pdf)

Encuestas. CAELUM 2007. [Disponible en:  
<http://www.calidaddelsoftware.com/modules.php?name=Surveys&op=results&pollID=3>

ESTRADA, A. F. Case Corporativo para el proceso de control de cambios. La Habana, Instituto Superior Politécnico “José Antonio Echeverría”, Facultad de Ingeniería Industrial, Centro de Estudios de Ingeniería y Sistemas, 2000. p.

ESTRADA, A. F. Un modelo de Referencia para la Gestión de Configuración en la PYME de Software. La Habana, Instituto Superior Politécnico “José Antonio Echeverría”, Facultad de Ingeniería Industrial, Centro de Estudios de Ingeniería y Sistemas, 2003. 16. p.

GARCÍA, F. Proceso Software y Gestión del Conocimiento, Departamento de Tecnologías y Sistemas de Información. Escuela Superior de Informática. 2007. [Disponible en: <http://alarcos.inf-cr.uclm.es/doc/psgc/doc/psgc-4a.pdf>

GONZÁLEZ, D. Las Métricas de Software y su Uso en la Región. México, Universidad de las Américas, Puebla, 2001. p.

Infocalidad 2005. [Disponible en:

[http://www.infocalidad.net/gest\\_calidad\\_def/definicion.asp](http://www.infocalidad.net/gest_calidad_def/definicion.asp)

Ingeniería del software. 2005. [Disponible en:

<http://www.monografias.com/trabajos15/ingenieria-software/ingenieria-software.shtml>

La informatización en Cuba. Sitio del Ministerio de Relaciones Exteriores de Cuba 2005. [Disponible en:

[http://www.cubaminrex.cu/Sociedad\\_Informacion/Cuba\\_SI/Informatizacion.htm](http://www.cubaminrex.cu/Sociedad_Informacion/Cuba_SI/Informatizacion.htm)

ISO/IEC\_9126-1 (2001). Ingeniería de software. Calidad del producto. Parte 1. Modelo de la calidad.

ISO/IEC TR 9126-2:2003 Software engineering – Product quality – Part 2: External metrics. CH-1211 Ginebra, Suiza, 2003. 94 p.

NC ISO/IEC 12119 Tecnología de la Información-Paquetes de Software-Requisitos de Calidad y Pruebas (ISO 12119: 1994, IDT)

ISO/IEC 176/SC1N322 (2007) Organización Internacional para la Estandarización. Concepts and Terminology. Revision of ISO 9000 Quality management systems- Fundamentals and vocabulary Preliminary working draft. 2007. 45 p.

MARTÍNEZ, R. D. ConfigCASE 3.0 Herramienta de apoyo a la Gestión de apoyo a la gestión de configuración. Propuesta Arquitectónica., Instituto Superior Politécnico “José Antonio Echeverría”, 2006. p.

NC\_ISO/IEC\_GUIA\_60 (2005). EVALUACION DE LA CONFORMIDAD—CODIGO DE BUENA PRÁCTICA, Oficina Nacional de Normalización.

PRESSMAN, R. S. Ingeniería del Software. Un Enfoque Práctico. 5ta Edición. 1998. p.

Pressman, Roger S. Ingeniería de Software, un enfoque práctico. 5ta edición. McGraw-Hill, 2002.

ROGER S. PRESSMAN. Ingeniería de Software. Un Enfoque Práctico. 4ª edición. 1997. 51-61 p.

Rossi, Bibiana. Métricas de Software, 2003. Disponible en:  
<http://www.itba.edu.ar/capis/rtis/articulosdeloscuadernoseta previa/ROS SI-METRICAS.pdf>

Cueva Lovelle. J Manuel. Calidad de Software. Universidad de Oviedo, España, 1999. Disponible en:  
<http://www.uniovi.es>

Scalone. Fernanda. Estudio comparativo de los Modelos y Estándares de Calidad del Software. Universidad Tecnológica Nacional. Buenos Aires, 2006.

Acosta Montolla. Lizzet. Calidad y Estándares Internacionales. Universidad de las Ciencias Informáticas, 2007.

## Glosario de Términos y Siglas

**Artefactos:** Son los elementos de entrada y salida de las actividades. Son productos tangibles del proyecto. Las cosas que el proyecto produce o usa para componer el producto final (modelos, documentos, código, ejecutables...).

**Atributos:** Una unidad con nombre de un clasificador que describe el rango de valores que las instancias de una propiedad pueden tomar.

**CEIS:** Centro de Estudios de Ingeniería de Sistemas.

**Hardware:** Se refiere a toda la infraestructura tecnológica, componentes físicos (que se pueden tocar) de la computadora: discos, unidades de disco, monitor, teclado, mouse, impresora, placas, chips, unidades de almacenamiento externo, scanners, entre otros.

**Gestión de Calidad:** Determinación y aplicación de las políticas de calidad de la empresa (objetivos y directrices generales).

**Gestión de configuración:** Es el proceso de identificar y definir los elementos en el sistema, controlando el cambio de estos elementos a lo largo de su ciclo de vida, registrando y reportando el estado de los elementos y las solicitudes de cambio, y verificando que los elementos estén completos y que sean los correctos.

**Gestión de proyecto:** La gestión de proyectos es la disciplina de organizar y administrar recursos de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, el tiempo, y coste definidos.

**ICSW:** La Industria Cubana del Software.

**IEC:** Comisión Internacional para la Electrónica.

**ISO:** Organización Internacional para la Estandarización.

**ISP:** Instituto Superior Pedagógico.

**IPI:** Instituto Politécnico de Informática.

**Metodología:** Son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software.

**NC:** Norma Cubana.

**Proceso:** Conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados, utilizando recursos.

**Producto:** Artefactos que se crean durante la vida del proyecto, como los modelos, código fuente, ejecutables, y documentación.

**Proyecto:** Elemento organizativo a través del cual se gestiona el desarrollo de software. El resultado de un proyecto es una versión de un producto.

**Requisitos:** Son las funciones, servicios y restricciones operativas del sistema.

**Requisitos funcionales:** Son aquellos que describen lo que debe hacer el sistema.

**Requisitos no funcionales:** Son aquellos que describen las facilidades que debe proporcionar el sistema.

**Sistema operativo:** Programa que una vez cargado, normalmente al encender el ordenador, maneja y controla los procesos y los programas (llamados aplicaciones).

**Software:** Se refiere a los programas y datos almacenados en un ordenador.

**SQM:** Gestión de Calidad del Software.

**SquaRE:** Ingeniería de Requisitos de Calidad de Seguridad.

**TIC:** Tecnologías de la Informática y las Comunicaciones.

**TR:** Reporte Técnico.

**UCI:** Universidad de las Ciencias Informáticas.

## Anexos

### Anexo 1 - Definiciones de las Características y Sub-características de calidad propuestas por el estándar ISO/IEC 9126

La siguiente tabla muestra las definiciones de las características y sub-características de calidad del estándar ISO/IEC 9126 [Pressman, 2002] [NC-ISO/IEC 9126-1, 2005]:

1. Funcionalidad	
	Es la capacidad del software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas cuándo el software se usa bajo las condiciones especificadas. Es el grado en que el software satisface las necesidades indicadas por las siguientes sub-características:
a. <b>Idoneidad</b>	Capacidad del software para mantener un conjunto apropiado de funciones para las tareas y los objetivos del usuario especificados.
b. <b>Exactitud</b>	Capacidad del software para proporcionar efectos o resultados correctos o convenidos con el grado de exactitud necesario.
c. <b>Interoperabilidad</b>	Capacidad del software para interactuar recíprocamente con uno o más sistemas especificados.
d. <b>Seguridad (informática)</b>	Capacidad del software para proteger información y los datos, para que personas o sistemas desautorizados no puedan leer o pueden modificar los mismos, y las personas o sistemas autorizados tenga el acceso a ellos.
e. <b>Conformidad con la funcionalidad</b>	Capacidad del software para adherirse a las normas que se le apliquen, convenciones, regulaciones, leyes y las prescripciones similares relativas a la funcionalidad.
2. Confiabilidad o Fiabilidad	
	Es la capacidad del software para mantener un nivel de ejecución especificado cuando se usa bajo las condiciones especificadas. Cantidad de tiempo que el software está disponible para su uso. Se refiere a las sub-características:
a. <b>Madurez</b>	Capacidad del software de evitar un fallo total como resultado de haberse producido un fallo del software.
b. <b>Tolerancia ante fallos</b>	Capacidad del software de mantener un nivel de ejecución o desempeño especificado en caso de fallos del software o de infracción

	de su interfaz especificada.
<b>c. Recuperabilidad</b>	Capacidad del software de restablecer un nivel de ejecución especificado y recuperar los datos directamente afectados en caso de fallo total.
<b>d. Conformidad con la confiabilidad</b>	Capacidad del software para adherirse a las normas que se le apliquen, convenciones, regulaciones, leyes y las prescripciones similares relativas a la confiabilidad.
<b>3. Usabilidad</b>	
	Grado con que el software es fácil de usar. Viene reflejado por las siguientes sub-características:
<b>a. Comprensibilidad</b>	Capacidad del software para permitirle al usuario entender si el software es idóneo, y cómo puede usarse para las tareas y condiciones de uso particulares.
<b>b. Cognoscibilidad</b>	Capacidad del software para permitirle al usuario aprender su aplicación.
<b>c. Operabilidad</b>	Capacidad del software para permitirle al usuario operarlo y controlarlo.
<b>d. Atracción</b>	Capacidad del software de ser atractivo o amigable para el usuario.
<b>e. Conformidad con la usabilidad</b>	Capacidad del software para adherirse a las normas, convenciones, guías de estilo o regulaciones relativas a la usabilidad.
<b>4. Eficiencia</b>	
	Capacidad del software para proporcionar una ejecución o desempeño apropiado. Grado con que el software hace óptimo el uso de los recursos del sistema. Esta característica está indicada por las siguientes sub-características:
<b>a. Rendimiento</b>	Capacidad del software para proporcionar apropiados tiempos de respuesta y procesamiento, así como tasas de producción de resultados, al realizar su función bajo condiciones establecidas.
<b>b. Utilización de recursos</b>	Capacidad del software para utilizar la cantidad y el tipo apropiado de recursos cuando el software realiza su función bajo las condiciones establecidas.
<b>c. Conformidad con la eficiencia</b>	Capacidad del software de adherirse a las normas o convenciones que se relacionan con la eficiencia.
<b>5. Mantenibilidad</b>	
	Capacidad del software de ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptaciones del software a cambios en el ambiente, así como en los requisitos y las especificaciones

	funcionales. Está reflejada por las siguientes sub-características:
a. <b>Diagnósticabilidad</b>	Capacidad del software de ser objeto de un diagnóstico para detectar deficiencias o causas de los fallos totales en el software, o para identificar las partes que van a ser modificadas.
b. <b>Cambiabilidad</b>	Capacidad del software para permitir la aplicación de una modificación especificada.
c. <b>Estabilidad</b>	Capacidad del software para minimizar los efectos inesperados de las modificaciones realizadas al software.
d. <b>Comprobabilidad</b>	Capacidad del software para permitir la validación de un software modificado.
e. <b>Conformidad con la mantenibilidad</b>	Capacidad del software para adherirse a las normas o convenciones que se relacionan con la mantenibilidad.
<b>6. Portabilidad</b>	
	La facilidad con que el software puede ser llevado de un entorno a otro. Se refiere a las sub-características:
a. <b>Adaptabilidad</b>	Capacidad del software de ser adaptado a los ambientes especificados sin aplicar acciones o medios de otra manera que aquellos suministrados con el propósito de que el software cumpla sus fines.
b. <b>Instalabilidad</b>	Capacidad del software de ser instalado en un ambiente especificado.
c. <b>Coexistencia</b>	Capacidad del software de coexistir con otro software independiente en un ambiente común y compartir los recursos comunes.
d. <b>Remplazabilidad</b>	Capacidad del software de ser usado en lugar de otro producto software especificado para los mismos fines y en el mismo ambiente.
e. <b>Conformidad con la portabilidad</b>	Capacidad del software de adherirse a las normas o convenciones relativas a la portabilidad.

## Anexo 2 - Tablas contentivas de las métricas

### 1 - Tablas contentivas de las métricas de funcionalidad

Tabla 1.1 Métricas de idoneidad

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
a) <b>Adecuación funcional</b>	¿Cuán adecuada es la función evaluada?	Número de funciones idóneas para ejecutar funciones específicas en comparación con el número de funciones evaluadas.	$X = 1 - A/B$ A - Número de funciones en las cuales se detectaron problemas en la evaluación B - Número de funciones evaluadas	$0 \leq X \leq 1$ A mayor cercanía al 1 resultará más adecuada	X = contab/ Contable  A = Contab  B = Contab
b) <b>Compleitud de la implementación funcional</b>	¿Cuán completa ha sido la implementación y su conformidad con la especificación de requisitos?	Ejecutar las pruebas (de caja negra) funcionales de acuerdo con la especificación de requisitos. Cuento el número de funciones perdidas detectadas y compare el resultado con el número de funciones descritas en la especificación de requisitos.	$X = 1 - A/B$ A - Número de funciones perdidas detectadas en la evaluación B - Número de funciones descritas en especificación de requisitos.	$0 \leq X \leq 1$ A mayor cercanía al 1 resultará mejor	X = contab/ Contable  A = contabl  B = contabl
c) <b>Coertura de la implementación funcional</b>	¿Cuán correcta ha sido la implementación funcional?	Ejecutar las pruebas funcionales (de caja negra) de acuerdo con la especificación de requisitos. Cuento el número de funciones incorrectamente implementadas o funciones perdidas detectadas y compare el resultado con el número total de funciones descritas en la especificación de requisitos. Cuento el número de funciones que están completas en relación con las que no lo están.	$X = 1 - A/B$ A - Número de funciones incorrectamente implementadas o funciones perdidas detectadas.  B - Número de funciones descritas en la especificación de requisitos.	$0 \leq X \leq 1$ A mayor cercanía al 1 resultará mejor	X = contab/ Contable  A = contabl  B = contabl

**Tabla 1.2 Métricas de exactitud**

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
a) <b>Exactitud esperada</b>	¿Existen diferencias entre los resultados actuales y los razonablemente esperados?	<p>Ejecutar los casos de pruebas de entrada versus salida y comparar los resultados actuales y los razonablemente esperados.</p> <p>Cuenta el número de casos encontrados con diferencias inaceptables en relación con los resultados razonablemente esperados.</p>	<p><math>X = A/T</math></p> <p>A - Número de casos encontrados con diferencias entre los resultados razonablemente esperados y aquellos resultantes más allá de lo permisible.</p> <p>T - Tiempo de operación</p>	<p><math>0 \leq X</math></p> <p>A mayor cercanía al 0 resultará mejor</p>	<p>X = contab/ tiempo</p> <p>A = contabl</p> <p>T = tiempo</p>

**Tabla 1.3 Métricas de interoperabilidad**

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
a) <b>Intercambiabilidad de datos, en base su formato</b>	¿Cuán correctamente ha sido implementado el intercambio de funciones de interfaces para una transferencia de datos específica?	<p>Número Ejecutar las pruebas a cada registro de salida de las funciones de interfaces de acuerdo con la especificación de los campos de datos. Cuenta el número de formatos de datos que deben ser intercambiados con otros software o sistemas durante las pruebas en comparación con el número total</p>	<p><math>X = A / B</math></p> <p>A- Número de formatos de datos intercambiados exitosamente con otros software o sistemas durante las pruebas del intercambio de datos.</p> <p>B - Número total de formatos de datos a intercambiar</p>	<p><math>0 \leq X \leq 1</math></p> <p>A mayor cercanía al 1 resultará mayor interoperabilidad</p>	<p>X = contab/ Contable</p> <p>A = Contab</p> <p>B = Contab</p>

**Tabla 1.3 Métricas de interoperabilidad (continuación)**

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
<p><b>b) Intercambiabilidad de datos, en base éxito del intento</b></p>	<p>¿Cuán frecuentemente falló el intento de intercambio de datos entre el software objeto de la prueba y otro software? ¿Cuán frecuentemente ES SATISFACTORI A la transferencia de datos entre el software objeto de la prueba y otro?</p>	<p>Ejecutar las pruebas. Cuenta el número de casos en que las funciones de interfaces fueron usadas y fallaron.</p>	<p><b>1) <math>X = 1 - \frac{A}{B}</math></b>  A- Número de casos en que se falló al proceder a un intercambio de datos con otros software o sistemas. B - Número de casos en que se intentó proceder a un intercambio de datos  <b>2) <math>Y = \frac{A}{T}</math></b>  T - Período de tiempo de operación</p>	<p><b><math>0 \leq X \leq 1</math></b>  A mayor cercanía al 1 resultará mejor  <b><math>0 \leq Y</math></b>  A mayor cercanía al 0 resultará mejor</p>	<p>1) <math>X = \frac{\text{cont}}{\text{cont}}</math> A = contabl B = contabl 2) <math>Y = \frac{\text{cont}}{\text{tiempo}}</math> T - tiempo</p>

## 2 - Tablas contentivas de las métricas de confiabilidad

Tabla 2.1. Métricas de madurez

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
a) Latencia estimada de la intensidad de fallos	¿Cuántos problemas aún existen que pueden emerger como futuros fallos?	Cuenta el número de fallos detectados durante el período definido de pruebas y pronostique el número potencial de futuros fallos utilizando un modelo de estimación de la fiabilidad.	$X = (ABS(A1 - A2)) / B$ ABS() - Valor absoluto A1 - Número total de fallos latentes predecibles en el producto de software A2 - Número total de fallos detectados realmente B - Tamaño del producto.	$0 \leq X$ En dependencia del estadio de las pruebas, En etapas más avanzadas, mientras más pequeño, mejor	X = Contabl / Tamaño A1 = Cont A2 = Cont B = Tamaño

Tabla 2.1. Métricas de madurez (continuación)

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
b) Intensidad de fallos totales contra casos de prueba	¿Cuántos fallos totales fueron detectados durante un	Cuenta el número de fallos totales detectados y el número de casos de pruebas.	$X = A1 / A2$ A1 - Número total de fallos totales	$0 \leq X$ En dependencia del estadio de las pruebas, En etapas más avanzadas,	X = Contabl / contabl e

	período de pruebas definido?		detectados A2 - Número de casos de pruebas ejecutados	mientras más pequeño, mejor	A1 = Cont  A2 = Cont
<b>c) Grado de solución ante fallos totales</b>	¿Cuántas condiciones de fallo total están resueltas?	Cuente el número de fallos totales que no se repitieron en determinado período de pruebas bajo condiciones similares. Mantenga un reporte de solución de problemas describiendo la situación de todos los fallos totales.	$X = A1 / A2$  A1 - Número de fallos totales solucionados  A2 - Número total de problemas reales detectados.	$0 \leq X \leq 1$  A mayor cercanía al 1 resultará mejor, cuanto más fallos totales estén resueltos	X = Contabl / contabl e  A1 = Cont  A2 = Cont
<b>d) Intensidad de fallos</b>	¿Cuántos fallos fueron detectados durante un período de pruebas definido?	Cuente el número de fallos detectados y compute su intensidad.	$X = A / B$  A - Número total de fallos detectados  B - Tamaño del producto	$0 \leq X$ Depende del estadio de las pruebas, En etapas más avanzadas, mientras más pequeño, mejor	X = Contabl / tamaño  A = Cont  B = Tamaño

**Tabla 2 .1. Métricas de madurez (continuación)**

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
e) <b>Erradicación de fallos</b>	¿Cuántos fallos han sido corregidos?	Cuenta el número de fallos resueltos durante el período de pruebas y compárelos con el número total de fallos detectados y el número total de fallos pronosticados.	<b>1) <math>X = A1 / A2</math></b> A1 - Número de fallos solucionados A2 - Número total de fallos reales detectados. <b>2) <math>Y = A1 / A3</math></b> A3 - Número total de fallos latentes pronosticados	<b><math>0 \leq X \leq 1</math></b>  A mayor cercanía al 1 resultará mejor (cuanto menos fallos queden)  <b><math>0 \leq Y</math></b>  A mayor cercanía al 0 resultará mejor (cuanto menos fallos queden)	1) $X = \text{cont} / \text{cont}$ $A1 = \text{cont}$ $A2 = \text{cont}$ 2) $Y = \text{cont} / \text{cont}$  contable $A3 = \text{cont}$
f) <b>Tiempo medio entre fallos totales</b>	¿Cuán frecuentemente el software fracasa en su operación.?	. Cuenta el número de fallos totales ocurridos durante el período de operación definido y compute el intervalo promedio entre fallos totales.	<b>1) <math>X = T1 / A</math></b> <b>2) <math>Y = T2 / A</math></b>  T1 – Tiempo de operación T2 – suma de los intervalos de tiempo entre los fallos totales consecutivos producidos. A - Número total de fallos realmente detectados (fallos totales ocurridos durante el tiempo de operación observado)	<b><math>0 &lt; X, Y</math></b>  A mayor resultará mejor, cuanto mayor sea el período de tiempo entre fallos  <b><math>0 \leq Y</math></b>  A mayor cercanía al 0 resultará mejor (cuanto menos fallos queden)	1) $X = \text{tiempo} / \text{cont}$ 2) $Y = \text{tiempo} / \text{cont}$  $A = \text{cont}$  $T1 = \text{tiempo}$ $T2 = \text{tiempo}$  $A3 = \text{cont}$

**Tabla 2.1. Métricas de madurez (continuación)**

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
<b>g) Cobertura de las pruebas</b>	¿Cuántos casos de pruebas requeridos han sido ejecutados detectados durante las pruebas?	Cuenta el número de casos de pruebas que han sido ejecutados detectados durante las pruebas y compárelo con el número de casos de pruebas requeridos para obtener una adecuada cobertura de pruebas.	$X = A / B$  A – Número de casos de pruebas que han sido realmente ejecutados, y que representan el escenario de operación durante las pruebas.  B – Número de casos de pruebas a ejecutar requeridos para cubrir los requisitos	$0 \leq X \leq 1$  Mientras más cercano al 1, mejor cobertura.	$X = \frac{\text{Contable}}{\text{Cont}} = \frac{A}{B}$
<b>h) Madurez de las pruebas</b>	¿Está bien probado el producto?	Cuenta el número de casos de pruebas que han obtenido un resultado satisfactorio de los casos realmente ejecutados y compárelo con el número total de casos de pruebas requeridos para cubrir los requisitos.	$X = A / B$  A – Número de casos de pruebas que han obtenido un resultado satisfactorio al ser ejecutados o durante su operación.  B – Número de casos de pruebas a ejecutar para cubrir los requisitos	$0 \leq X \leq 1$  Mientras más cercano al 1, mejor.	$X = \frac{\text{Contable}}{\text{Cont}} = \frac{A}{B}$

**Tabla 2.2. Métricas de tolerancia ante fallos**

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
a) <b>Evitación de desastre</b>	¿Cuán frecuentemente el producto de software causa un desastre en el ambiente de la producción total?	Cuenta el número de desastres detectados con respecto al número de fallos totales.	$X = 1 - A / B$  A - Número de desastres  B - Número de fallos totales	$0 \leq X \leq 1$  A mayor cercanía al 1 resultará mejor,	X = Contabl/ contable  A = Cont B = Cont
b) <b>Evitación de operaciones incorrectas</b>	¿Cuántas funciones están implementadas con capacidad de evitación de operaciones incorrectas?	Cuenta el número de casos de prueba de operaciones incorrectas que fueron evitadas para que no causaran fallos totales críticos o serios, y compárelo con el número de casos de pruebas ejecutados a los patrones de operaciones incorrectas considerados período de pruebas bajo condiciones similares.	$X = A / B$  A - Número de ocurrencia de fallos totales críticos o serios evitada  B - Número de casos de pruebas ejecutados a los patrones de operaciones incorrectas (casi causantes de fallos) durante las pruebas.	$0 \leq X \leq 1$  A mayor cercanía al 1 resultará mejor, cuanto más operaciones incorrectas del usuario sean evitadas	X = Contabl/ contable  A = Cont B = Cont

**Tabla 2.3. Métricas de recuperabilidad**

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
a) <b>Grado de disponibilidad</b>	¿Cuán disponible está el sistema para su uso durante?	Pruebe el sistema en la producción como ambiente durante un período de tiempo especificado ejecutando todas las operaciones del usuario.	<p>1) <math>X = (T_o / (T_o + T_r))</math></p> <p>2) <math>Y = A1 / A2</math></p> <p><math>T_o</math> – Tiempo de operación  <math>T_r</math> – Tiempo de reparación  <math>A1</math> – Total de casos de uso del software disponibles exitosamente, cuando se han intentado utilizar. <math>A2</math> – Total de casos de uso del software que se han intentado utilizar durante el tiempo de observación.</p>	<p><math>0 \leq X \leq 1</math></p> <p>A mayor cercanía al 1 resultará mejor</p>	<p><math>X = \text{Tiempo} / \text{Tiempo}</math></p> <p><math>Y = \text{Contabl} / \text{contable}</math></p> <p><math>A1 = \text{Cont}</math></p> <p><math>A2 = \text{Cont}</math></p>

**Tabla 2.3. Métricas de recuperabilidad (continuación)**

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
b) <b>Tiempo medio de inactividad</b>	¿Cuál es el tiempo promedio en que el sistema se mantiene no disponible	Mida el tiempo de inactividad cada vez que el sistema se encuentre no disponible durante un período de prueba especificado y compute el tiempo medio.	<p><math>X = T / N</math></p> <p><math>T</math> - tiempo total de inactividad  <math>N</math> - Número de</p>	<p><math>0 &lt; X</math></p> <p>Cuanto menor sea mejor, el sistema estará inactivo por menos tiempo.</p>	<p><math>X = \text{tiempo} / \text{contable}</math></p> <p><math>T = \text{tiempo}</math></p>

	cuando ocurre un fallo total y antes de la arrancada gradual?		desastres observados.  El peor caso o la distribución del tiempo de inactividad debe ser medido.		$N = \text{Cont}$
<b>c) Tiempo medio de recuperación</b>	¿Cuál es el tiempo promedio que toma el sistema para completar la recuperación desde el inicio de la recuperación parcial?	Mida el tiempo total de recuperación cada vez que haya pasado a la inactividad (caído el sistema) y calcule el tiempo promedio invertido en la recuperación.	$X = \text{SUM}(T_n) / N$ T - tiempo de recuperación de la inactividad en cada oportunidad N - Número de oportunidades en que el sistema entró en recuperación	$0 < X$  Cuanto menor sea mejor.	$X = \text{tiempo} / \text{contable}$  $T = \text{tiempo}$  $N = \text{Cont}$

**Tabla 2.3. Métricas de recuperabilidad (continuación)**

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
<b>d) Recargabilidad</b>	¿Cuán frecuentemente el sistema logra recargar luego de una ejecución perdida proveyendo nuevamente los servicios a los usuarios en el plazo de tiempo previsto?	Contar el número de veces que el sistema se reinicia o se recarga y ofrece exitosamente sus servicios a los usuarios en el plazo de tiempo previsto, y compárelo con el número de veces que el sistema se reinició como resultado de una caída en inactividad durante un período de prueba especificado.	$X = A / B$  A - Número de veces que se provocó el reinicio o recarga en el plazo de tiempo previsto en la prueba especificada o en la implantación.  B - Número	$0 \leq X \leq 1$ A mayor cercanía al 1 mejor, el usuario puede reiniciar o recargar más fácilmente.	$X = \text{contab} / \text{contab}$  A = contabl  B = contabl

			total de veces que se provocó el reinicio o recarga durante la prueba especificada o la implantación.		
--	--	--	---	--	--

**Tabla 2.3. Métricas de recuperabilidad (continuación)**

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
<b>e) Restaurabilidad</b>	¿Cuán capaz es el producto de auto restaurarse luego de un evento anormal o una solicitud?	Cuenta el número de restauraciones exitosas y compárelo con el número de restauraciones probadas requeridas por las especificaciones.  Ejemplos de requisitos de restauración son:  - función deshacer, - función rehacer	$X = A / B$  A – Número de casos de restauración exitosos  B – Número de casos de restauración probados por los requisitos	$0 \leq X \leq 1$ A mayor cercanía al 1 mejor y el producto es más capaz de restaurarse en casos definidos.	X = contab/ contab A = contabl B = contabl
<b>f) Efectividad de la restauración</b>	¿Cuán efectiva es la capacidad de restauración?	Cuenta el número de restauraciones que cumplen con el plazo de tiempo previsto y compárelo con el número de restauraciones y compárelo con el número de restauraciones requeridas en el plazo de tiempo especificado.**	$X = A / B$  A – Número de casos de restauración exitosos en el plazo de tiempo previsto  B – Número de casos de restauración ejecutados	$0 \leq X \leq 1$ A mayor cercanía al 1 mejor, y los procesos de restauración del producto son más efectivos.	X = contab/ contab A = contabl B = contabl

### 3 - Tablas contentivas de las métricas de usabilidad

**Tabla 3.1 Métricas de comprensibilidad**

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
a) <b>Integridad de la descripción de producto</b>	¿Qué proporción de funciones (o tipos de funciones) son comprendidas después de leer la descripción del producto?	Conduzca las pruebas de usuario. Realice entrevistas a usuarios con cuestionarios preparados al efecto. Observe el comportamiento del usuario. Cuente el número de funciones que fueron adecuadamente comprendidas en comparación con el número total de funciones en el producto.	<b>1) <math>X = A / B</math></b> A - número de funciones comprendidas B - número total de funciones	<b><math>0 \leq X \leq 1</math></b> A mayor cercanía al 1 resultará mejor	X = cont/cont A = Contable B = Contable
b) <b>Accesibilidad a demos</b>	- ¿A qué proporción de demos/tutoriales pueden acceder los usuarios?	Conduzca las pruebas de usuario. Observe el comportamiento del usuario.	<b><math>X = A / B</math></b> A - número de demos/tutoriales a los que pueden acceder los usuarios exitosamente. B - número total de demos/tutoriales a los que se puede acceder	<b><math>0 &lt; X \leq 1</math></b> mientras cercano al 1, mejor	X = Contable /contable A = Contable B = Contable

**Tabla 3.1 Métricas de comprensibilidad (continuación)**

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
c) <b>Comprensibilidad de entradas y salidas</b>	¿Pueden los usuarios comprender lo que se requiere como	Conduzca las pruebas de usuario. Realice entrevistas a usuarios con cuestionarios preparados al efecto. Observe el	<b><math>X = A / B</math></b> A - número de elementos de entrada y que	<b><math>0 \leq X \leq 1</math></b> Mientras cercano al 1, mejor	X = Cont/co nt

	entrada y lo que suministra el sistema de software como salida?	comportamiento del usuario. Cuento el número de elementos de entrada .	<p>suministra el sistema de software como salida comprendidas correctamente</p> <p>B - número total de elementos de entrada y que suministra el sistema de software como salida proporcionados por el interfaz.</p>		<p>A = Contable</p> <p>B = Contable</p>
--	---	--	---	--	---

**Tabla 3.2 Métricas de atracción**

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
b) <b>Adaptabilidad de la apariencia de la interfaz</b>	¿Qué proporción de los elementos de la interfaz puede ser, por su apariencia, adaptado por el usuario para la satisfacción del mismo?	Conduzca las pruebas de usabilidad. Observe el comportamiento del usuario.	<p><math>X = A / B</math></p> <p>A - número de elementos de la interfaz del sistema cuya apariencia puede ser adaptada por el usuario</p> <p>B - número de elementos de la interfaz del sistema cuya apariencia querría adaptar el usuario</p>	<p><math>0 \leq X \leq 1</math></p> <p>mientras cercano al 1, mejor</p>	<p>X = con/con A = Contable B = Contable</p>

## Anexo 3 - Plantilla # 1

VARIABLES QUE DEBEN SER OBTENIDAS EN LOS PROCESOS DE PRUEBAS POR ESTAR IMPLICADAS EN LAS FÓRMULAS DE MEDICIÓN DE LAS TABLAS CONTENIDAS EN LAS MÉTRICAS.

### 1 Tabla para determinar las variables contenidas en las métricas de Funcionalidad

<b>Métricas de idoneidad</b>		
<b>Nombre de la métrica</b>	<b>Variables</b>	<b>Datos</b>
<b>a) Adecuación funcional</b>	<b>A</b> - Número de funciones en las cuales se detectaron problemas en la evaluación. <b>B</b> - Número de funciones evaluadas.	
<b>b) Completitud de la implementación funcional</b>	<b>A</b> - Número de funciones perdidas detectadas en la evaluación. <b>B</b> - Número de funciones descritas en especificación de requisitos.	
<b>c) Cobertura de la implementación funcional</b>	<b>A</b> - Número de funciones incorrectamente implementadas o funciones perdidas detectadas. <b>B</b> - Número de funciones descritas en la especificación de requisitos.	
<b>Métrica de exactitud</b>		
<b>Nombre de la métrica</b>	<b>Variables</b>	<b>Datos</b>
<b>a) Exactitud esperada</b>	<b>A</b> - Número de casos encontrados con diferencias entre los resultados razonable-mente esperados y aquellos resultantes más allá de lo permisible. <b>T</b> - Tiempo de operación.	
<b>Métricas de interoperabilidad</b>		
<b>Nombre de la métrica</b>	<b>Variables</b>	<b>Datos</b>
<b>a) Intercambiabilidad de datos, en base su formato</b>	<b>A</b> - Número de formatos de datos intercambiados exitosamente con otros software o sistemas durante las pruebas del intercambio de datos. <b>B</b> - Número total de formatos de datos a intercambiar.	
<b>b) Intercambiabilidad de datos, en base éxito del intento</b>	<b>A</b> - Número de casos en que se falló al proceder a un intercambio de datos con otros software o sistemas. <b>B</b> - Número de casos en que se intentó proceder a un intercambio de datos. <b>T</b> - Período de tiempo de operación.	

## 2 Tabla para determinar las variables contenidas en las métricas de Confiabilidad

<b>Métricas de madurez</b>		
<b>Nombre de la métrica</b>	<b>Variables</b>	<b>Datos</b>
<b>a) Latencia estimada de la intensidad de fallos</b>	<b>ABS ()</b> - Valor absoluto. <b>A1</b> - Número total de fallos latentes predecibles en el producto de software. <b>A2</b> - Número total de fallos detectados realmente. <b>B</b> - Tamaño del producto.	
<b>b) Intensidad de fallos totales contra casos de prueba</b>	<b>A1</b> - Número total de fallos totales detectados. <b>A2</b> - Número de casos de pruebas ejecutados.	
<b>c) Grado de solución ante fallos totales</b>	<b>A1</b> - Número de fallos totales solucionados. <b>A2</b> - Número total de problemas reales detectados.	
<b>d) Intensidad de fallos</b>	<b>A</b> - Número total de fallos detectados. <b>B</b> - Tamaño del producto.	
<b>e) Erradicación de fallos</b>	<b>A1</b> - Número de fallos solucionados. <b>A2</b> - Número total de fallos reales detectados. <b>A3</b> - Número total de fallos latentes pronosticados.	
<b>f) Tiempo medio entre fallos totales</b>	<b>T1</b> - Tiempo de operación. <b>T2</b> - Suma de los intervalos de tiempo entre los fallos totales consecutivos producidos. <b>A</b> - Número total de fallos realmente detectados (fallos totales ocurridos durante el tiempo de operación observado).	
<b>g) Cobertura de las pruebas</b>	<b>A</b> - Número de casos de pruebas que han sido realmente ejecutados, y que representan el escenario de operación durante las pruebas. <b>B</b> - Número de casos de pruebas a ejecutar requeridos para cubrir los requisitos.	
<b>h) Madurez de las pruebas</b>	<b>A</b> - Número de casos de pruebas que han obtenido un resultado satisfactorio al ser ejecutados o durante su operación. <b>B</b> - Número de casos de pruebas a ejecutar para cubrir los requisitos.	
<b>Métricas de tolerancia ante fallos</b>		
<b>Nombre de la métrica</b>	<b>Variables</b>	<b>Datos</b>
<b>a) Evitación de</b>	<b>A</b> - Número de desastres.	

<b>desastre</b>	<b>B</b> - Número de fallos totales.	
<b>b) Evitación de operaciones incorrectas</b>	<b>A1</b> - Número de ocurrencia de fallos totales críticos o serios evitada. <b>A2</b> - Número de casos de pruebas ejecutados a los patrones de operaciones incorrectas (casi causantes de fallos) durante las pruebas.	
	<b>Métricas de recuperabilidad</b>	
<b>Nombre de la métrica</b>	<b>Variables</b>	<b>Datos</b>
<b>a) Grado de disponibilidad</b>	<b>To</b> - Tiempo de operación. <b>Tr</b> - Tiempo de reparación.	
<b>b) Tiempo medio de inactividad</b>	<b>T</b> - Tiempo total de inactividad. <b>N</b> - Número de desastres observados.	
<b>c) Tiempo medio de recuperación</b>	<b>T</b> - Tiempo de recuperación de la inactividad en cada n oportunidad. <b>N</b> - Número de oportunidades en que el sistema entró en recuperación.	
<b>d) Recargabilidad</b>	<b>A</b> - Número de veces que se provocó el reinicio o recarga en el plazo de tiempo previsto en la prueba especificada o en la implantación. <b>B</b> - Número total de veces que se provocó el reinicio o recarga durante la prueba especificada o la implantación.	
<b>e) Restaurabilidad</b>	<b>A</b> - Número de casos de restauración exitosos. <b>B</b> - Número de casos de restauración probados por los requisitos.	
<b>f) Efectividad de la restauración</b>	<b>A</b> - Número de casos de restauración exitosos en el plazo de tiempo previsto. <b>B</b> - Número de casos de restauración ejecutados.	

**3 Tabla para determinar las variables contenidas en las métricas de Usabilidad**

<b>Métricas de comprensibilidad</b>		
<b>Nombre de la métrica</b>	<b>Variables</b>	<b>Datos</b>
<b>a) Integridad de la descripción de producto</b>	<b>A</b> - Número de funciones comprendidas. <b>B</b> - Número total de funciones.	
<b>b) Accesibilidad a demos</b>	<b>A</b> - Número de demos/tutoriales a los que pueden acceder los usuarios exitosamente. <b>B</b> - Número total de demos/tutoriales a los que se puede acceder.	
<b>c) Comprensibilidad de entradas y salidas</b>	<b>A</b> - Número de elementos de entrada y que suministra el sistema de software como salida comprendidas correctamente. <b>B</b> - Número total de elementos de entrada y que suministra el sistema de software como salida proporcionados por el interfaz.	
<b>Métrica de atracción</b>		
<b>Nombre de la métrica</b>	<b>Variables</b>	<b>Datos</b>
<b>b) Adaptabilidad de la apariencia de la interfaz</b>	<b>A</b> - Número de elementos de la interfaz del sistema cuya apariencia puede ser adaptada por el usuario. <b>B</b> - Número de elementos de la interfaz del sistema cuya apariencia querría adaptar el usuario.	

## Anexo 4 - Plantilla # 2

Plantilla de preevaluación y medición de la calidad con las métricas.

Calidad externa

Proyecto No. \_\_\_\_\_ Siglas: \_\_\_\_\_

Denominación: \_\_\_\_\_

CARACTERISTICA	SUBCARACTERISTICA	PESOS	METRICAS	NIVEL REQUE- RIDO	RESU L- TADO REAL	
<b>Funcionalidad</b>	Idoneidad		1.1.a) $X = 1 - A/B$	1 (0 <= X <= 1)		
			1.1.b) $X = 1 - A/B$	1 (0 <= X <= 1)		
			1.1.c) $X = 1 - A/B$	1 (0 <= X <= 1)		
		Exactitud		1.2.a) $X = A/T$	0 (0 <= X)	
	Interoperabilidad		1.3.a) $X = A / B$	1 (0 <= X <= 1)		
			1.3.b.1) $X = 1 - A/B$	1 (0 <= X <= 1)		
			1.3.b.2) $Y = A / T$	0 (0 <= Y)		
<b>Confiabilidad</b>	Madurez		2.1.a) $X = (ABS (A1 - A2)) / B$	(0 <= X)		
			2.1.b) $X = A1 / A2$	(0 <= X)		
			2.1.c) $X = A1 / A2$	1 (0 <= X <= 1)		
			2.1.d) $X = A / B$	(0 <= X)		
			2.1.e.1) $X = A1 / A2$	1 (0 <= X <= 1)		
			2.1.e.2)	0		

			$Y = A1 / A3$	$(0 \leq Y)$	
			2.1.f.1) $X = T1 / A$	$0 < X, Y$	
			2.1.f.2) 2) $Y = T2 / A$	$0$ $(0 \leq Y)$	
			2.1.g) $X = A / B$	$1$ $(0 \leq X \leq 1)$	
			2.1.h) $X = A / B$	$1$ $(0 \leq X \leq 1)$	
	Tolerancia ante fallos		2.2.a) $X = 1 - A / B$	$1$ $(0 \leq X \leq 1)$	
			2.2.b) $X = A / B$	$1$ $(0 \leq X \leq 1)$	
	Recuperabilidad		2.3.a) $X = (T_o / (T_o + T_r))$	$1$ $(0 \leq X \leq 1)$	
			2.3.b) $X = T / N$	$0$ $(0 < X)$	
			2.3.c) $X = \text{SUM}(T_n) / N$	$0$ $(0 < X)$	
			2.3.d) $X = A / B$	$1$ $(0 \leq X \leq 1)$	
			2.3.e) $X = A / B$	$1$ $(0 \leq X \leq 1)$	
			2.3.f) $X = A / B$	$1$ $(0 \leq X \leq 1)$	
<b>Usabilidad</b>	Comprensibilidad		3.1.a) $X = A / B$	$1$ $(0 \leq X \leq 1)$	
			3.1.b) $X = A / B$	$1$ $(0 < X \leq 1)$	
			3.1.c) $X = A / B$	$1$ $(0 \leq X \leq 1)$	
	Atracción		3.2.b) $X = A /$	$1$ $(0 \leq X \leq 1)$	

Escala para cuando el nivel requerido es 1

Rango	Evaluación	Puntuación
0 - 0,2	Mal	0
0,3 – 0,5	Regular	1
0,6 – 0,7	Bien	2
0,8- 1	Muy Bien	3

Escala para cuando el nivel requerido es 0

Rango	Evaluación	Puntuación
0 - 0,2	Muy Bien	3
0,3 – 0,5	Bien	2
0,6 – 0,7	Regular	1
0,8- 1	Mal	0

### Anexo 5 - Plantilla # 3

#### CUESTIONARIO INDIVIDUAL DE EVALUACION DE LA CONFORMIDAD

##### SECCION A VALORACION DE LOS RESULTADOS DE LAS PRUEBAS Y EL COMPORTAMIENTO DE LAS CARACTERISTICAS DE CALIDAD

Proyecto No. \_\_\_\_\_ Siglas: \_\_\_\_\_  
 Denominación: \_\_\_\_\_  
 Nivel de Clasificación: \_\_\_\_\_ Software crítico: No crítico:  
 Control No.: \_\_\_\_\_ Etapa: \_\_\_\_\_

CARACTERISTICAS Y SUBCARACTERISTICAS DE CALIDAD	PUNTUACION			
	3	2	1	0
<b>1. FUNCIONALIDAD</b>				
<b>1.1 Idoneidad.</b> Capacidad del software para mantener un conjunto apropiado de funciones para las tareas y los objetivos del usuario especificados.				
a) Funcionamiento correcto (ausencia de fallos totales - ciclos infinitos, interrupción de la ejecución o salidas abruptas-, errores críticos, errores de ejecución, resultados incorrectos, correspondencia de las descripciones con los objetos, por ejemplo en el nivel de ayuda solicitado o los mensajes de error según el fallo detectado )				
b) Correspondencia de las funciones con los requisitos funcionales de la Descripción del software (especificaciones)				
<b>1.2 Precisión.</b> Capacidad del software para proporcionar efectos o resultados correctos o convenidos con el grado de exactitud necesario.				
a) Exactitud de los cálculos				
<b>1.3 Interoperabilidad.</b> Capacidad del producto de software para interactuar recíprocamente con uno o más sistemas especificados.				
a) Funcionamiento correcto al interactuar (ausencia de fallos totales – ciclos infinitos, interrupción de la ejecución o salidas abruptas-, errores críticos, errores de ejecución, resultados incorrectos,)				

## CUESTIONARIO INDIVIDUAL DE EVALUACION DE LA CONFORMIDAD

### SECCION B VALORACION DE LOS RESULTADOS DE LAS PRUEBAS Y EL COMPORTAMIENTO DE LAS CARACTERISTICAS DE CALIDAD

Proyecto No. \_\_\_\_\_ Siglas: \_\_\_\_\_  
 Denominación: \_\_\_\_\_  
 Nivel de Clasificación: \_\_\_\_\_ Software crítico:      No crítico:  
 Control No.: \_\_\_\_\_ Etapa: \_\_\_\_\_

CARACTERISTICAS Y SUBCARACTERISTICAS DE CALIDAD	PUNTUACION			
	3	2	1	0
<b>2 CONFIABILIDAD</b>				
<b>2.1 Madurez.</b> Capacidad del producto de software de evitar un fallo total como resultado de haberse producido un fallo del software.				
a) Mantenimiento del funcionamiento correcto ante la aparición de un fallo (ausencia de fallos totales - ciclos infinitos, interrupción de la ejecución o salidas abruptas-, errores críticos, errores de ejecución, resultados incorrectos,)				
b) Uniformidad del retorno al procesamiento ante la aparición de un fallo (restablecimiento de pantallas, menús y ayudas)				
<b>2.2 Tolerancia ante fallos.</b> Capacidad del producto de software de mantener un nivel de desempeño o ejecución específico en caso de fallos del software o de infracción de sus interfaces especificadas				
a) Verificación de la memoria interna y externa (para la instalación del producto, la reconfiguración del producto o para la salva de la información)				
b) Validación previa de condiciones potenciales de errores (particiones de trabajo inexistentes, disponibilidad e integridad de los ficheros, disponibilidad de los periféricos)				
c) Tratamiento de los errores (procedimientos para la detección y corrección de errores internos del software)				
d) Procesamiento degradado (procedimiento para el funcionamiento degradado en caso de los fallos no recuperables como la ausencia de ficheros o deficiencias del hardware)				
<b>2.3 Recuperabilidad.</b> Capacidad del producto de software de restablecer un nivel de ejecución especificado y recuperar los datos directamente afectados en caso de fallo total				
a) Opciones de recuperabilidad (pérdida o deterioro de datos y elementos componentes del software, errores del operador)				

## CUESTIONARIO INDIVIDUAL DE EVALUACION DE LA CONFORMIDAD

### SECCION C VALORACION DE LOS RESULTADOS DE LAS PRUEBAS Y EL COMPORTAMIENTO DE LAS CARACTERISTICAS DE CALIDAD

Proyecto No. \_\_\_\_\_ Siglas: \_\_\_\_\_  
 Denominación: \_\_\_\_\_  
 Nivel de Clasificación: \_\_\_\_\_ Software crítico:      No crítico:  
 Control No.: \_\_\_\_\_ Etapa: \_\_\_\_\_

<b>CARACTERISTICAS Y SUBCARACTERISTICAS DE CALIDAD</b>	3	2	1	0
<b>3 USABILIDAD</b>				
<b>3.1 Comprensibilidad.</b> Capacidad del producto de software para permitirle al usuario entender si el software es idóneo, y cómo puede usarse para las tareas y condiciones de uso particulares				
a) Terminología (acorde con la actividad específica de la aplicación o actividad del usuario)				
b) Ayuda en línea (presencia de diferentes niveles de ayudas, sistema, pantalla, campo)				
c) Interfaz de usuario adecuada (representación de los objetos con análogos en el ambiente del usuario, utilización de iconos)				
<b>3.2 Cognoscibilidad.</b> Capacidad del producto del software para permitirle al usuario aprender su aplicación				
a) Existencia de un DEMO				
b) Existencia de un Tutorial				
<b>3.3 Operabilidad.</b> Capacidad del producto del software para permitirle al usuario operarlo y controlarlo				
a) Utilidad de las ayudas (utilidad de la información que brinda la ayuda)				
b) Operabilidad de las ayudas (facilidad de desplazamiento en la ventana de ayuda)				
c) Carga automática (generación durante la instalación del fichero de comandos para la ejecución del producto de software)				
d) Minimización del trabajo extra del usuario (durante la operación y en caso de fallos)				
e) Facilidades de operación opcional (selección de opciones mediante el cursor de barras, video invertido, brillantez de las opciones, mouse)				
<b>3.4 Atracción.</b> Capacidad del producto del software de ser atractivo o amigable para el usuario				
a) Uniformidad de la estructura, del contenido y del formato de los elementos componentes				
b) Uniformidad del vocabulario, de la simbología y de otras convenciones utilizadas				

**CUESTIONARIO INDIVIDUAL DE EVALUACION  
DE LA CONFORMIDAD**

**SECCION A**

**RESUMEN DEL CUESTIONARIO INDIVIDUAL**

<b>PUNTUACION</b>	
<b>Funcionalidad</b>	
<b>VALOR</b>	

<b>GRADO DE CONFORMIDAD</b>	
<b>3</b>	<b>Conforme</b>
<b>2</b>	<b>Suficientemente conforme</b>
<b>1</b>	<b>Medianamente conforme</b>
<b>0</b>	<b>No conforme</b>

<b>CRITERIO DE EVALUACION</b>
<input type="checkbox"/> Sin modificaciones
<input type="checkbox"/> Pequeñas modificaciones
<input type="checkbox"/> Grandes modificaciones
<input type="checkbox"/> Nueva elaboración

(área de firma)

Evaluador:

Fecha:

\_\_\_\_\_  
Cargo:

**CUESTIONARIO INDIVIDUAL DE EVALUACION  
DE LA CONFORMIDAD**

**SECCION B**

**RESUMEN DEL CUESTIONARIO INDIVIDUAL**

<b>PUNTUACION</b>	
<b>Confiabilidad</b>	
<b>VALOR</b>	

<b>GRADO DE CONFORMIDAD</b>	
<b>3</b>	<b>Conforme</b>
<b>2</b>	<b>Suficientemente conforme</b>
<b>1</b>	<b>Medianamente conforme</b>
<b>0</b>	<b>No conforme</b>

<b>CRITERIO DE EVALUACION</b>
<input type="checkbox"/> Sin modificaciones
<input type="checkbox"/> Pequeñas modificaciones
<input type="checkbox"/> Grandes modificaciones
<input type="checkbox"/> Nueva elaboración

(área de firma)

Evaluador:

Fecha:

Cargo: \_\_\_\_\_

**CUESTIONARIO INDIVIDUAL DE EVALUACION  
DE LA CONFORMIDAD**

**SECCION C**

**RESUMEN DEL CUESTIONARIO INDIVIDUAL**

<b>PUNTUACION</b>	
<b>Usabilidad</b>	
<b>VALOR</b>	

<b>GRADO DE CONFORMIDAD</b>	
<b>3</b>	<b>Conforme</b>
<b>2</b>	<b>Suficientemente conforme</b>
<b>1</b>	<b>Medianamente conforme</b>
<b>0</b>	<b>No conforme</b>

<b>CRITERIO DE EVALUACION</b>
<input type="checkbox"/> Sin modificaciones
<input type="checkbox"/> Pequeñas modificaciones
<input type="checkbox"/> Grandes modificaciones
<input type="checkbox"/> Nueva elaboración

(área de firma)

Evaluador:

Fecha:

Cargo: \_\_\_\_\_

## Anexo 6 - Plantilla # 4

### CUESTIONARIO COLECTIVO DE EVALUACION DE LA CONFORMIDAD

#### RESUMEN GENERAL

Proyecto No. \_\_\_\_\_ Siglas: \_\_\_\_\_  
 Denominación: \_\_\_\_\_  
 Nivel de Clasificación: \_\_\_\_\_ Software crítico:      No crítico:  
 Control No.: \_\_\_\_\_ Etapa: \_\_\_\_\_  
 \_\_\_\_\_

#### PUNTUACION PROMEDIADA A PARTIR DE LA INDIVIDUAL

<b>Funcionalidad</b>	
<b>Confiabilidad</b>	
<b>Usabilidad</b>	
<b>VALOR*</b>	

**\*Nota:** Este valor no se obtiene matemáticamente, sino por medio del razonamiento lógico en conformidad con el los objetivos de la evaluación, la aplicación y criticidad del software y otros factores. De requerirse votación (por no existir consenso) el líder evaluador cuenta con 2 votos.

GRADOS DE CONFORMIDAD POR SECCIONES (poner una X)		
<b>Sección A</b>		<input type="checkbox"/> Conforme <input type="checkbox"/> Suficientemente conforme <input type="checkbox"/> Medianamente conforme <input type="checkbox"/> No conforme
<b>Sección B</b>		<input type="checkbox"/> Conforme <input type="checkbox"/> Suficientemente conforme <input type="checkbox"/> Medianamente conforme <input type="checkbox"/> No conforme
<b>Sección C</b>		<input type="checkbox"/> Conforme <input type="checkbox"/> Suficientemente conforme <input type="checkbox"/> Medianamente conforme <input type="checkbox"/> No conforme

<b>VEREDICTO</b>	
	<b>ACEPTADO</b>
	<b>DIFERIDO</b>
	<b>NO ACEPTADO</b>

(área de firma)

Evaluador lider:  
Cargo:

Fecha: \_\_\_\_\_

Evaluador:  
Cargo:  
Evaluador:  
Cargo:  
Evaluador:  
Cargo:

## **Anexo 7 - Modelo para la recogida de información referente al peso de los criterios**

**Guía para informar el peso de los criterios.**

**Fecha de recepción...00/00/00.....**

**Fecha de entrega....00/00/00.....**

**Nombre y Apellidos del evaluador.....**

**Le otorgará un peso a cada criterio de acuerdo a su opinión y el peso total de cada grupo debe sumar:**

- Grupo No.1..... 40
- Grupo No.2..... 20
- Grupo No.3.....20
- Grupo No.4.....20

Para que el peso total asignado sea 100.

Grupo No 1: Criterios de mérito científico.

1. Valor científico de la propuesta.

Peso.....

2. Calidad de la investigación.

Peso.....

3. Contribución científica.

Peso.....

4. Responsabilidad científica y profesionalidad de los investigadores.

Peso.....

Grupo No 2: Criterios de implantación.

5. Necesidad de empleo de la propuesta.

Peso.....

6. Posibilidades de aplicación.

Peso.....

Grupo No 3: Criterios de flexibilidad.

7. Adaptabilidad a entidades dedicadas a evaluar la calidad de los productos de software.

Peso.....

8. Capacidad del proceso de evaluación para la admisión de cambios que impliquen mejoras.

Peso.....

Grupo No 4. Criterios de impacto.

9. Impacto en el área para la cual está destinada la guía.

Peso.....

10. Organización en el proceso de desarrollo.

Peso.....



8. Capacidad del proceso de evaluación para la admisión de cambios que impliquen mejoras.

Evaluación.....

Grupo No 4. Criterios de impacto.

9. Impacto en el área para la cual está destinada la guía.

Evaluación.....

10. Organización en el proceso de desarrollo.

Evaluación.....

### **Categoría final del proyecto**

\_\_\_ Excelente: Alta novedad científica, con aplicabilidad y resultados relevantes.

\_\_\_ Bueno: Novedad científica, resultados destacados.

\_\_\_ Aceptable: Suficientemente bueno con reservas.

\_\_\_ Cuestionable: No tiene relevancia científica y los resultados son malos.

\_\_\_ Malo: No aplicable.

Valoración final

Sugerencias del experto para mejorar la calidad del proyecto

Elementos críticos que deben mejorarse.