

**Universidad de las Ciencias Informáticas**  
**Facultad 8**



**Medición de la calidad de Software durante el  
Proceso de Pruebas en el Proyecto  
Modernización del CICPC**

**Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas**

**Autor (es):** Maria de los Angeles Rubalcaba Betancourt

Yayneris Zambrana Hernández

**Tutor (a):** Msc. Ing. Haydée Maria Cruz Torres

Ciudad de la Habana, Junio 2008

“Año 50 de la Revolución Cubana”

## DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los       días del mes de Junio del año 2008.

Maria de los Angeles Rubalcaba Betancourt       Yayneris Zambrana Hernández

---

Firma de la Autora

---

Firma de la Autora

Msc Ing. Haydée Maria Cruz Torres

---

Firma de la Tutora

*“La calidad de un producto es una función de los muchos  
cambios del mundo por mejorar”  
Tom De Marco.*

## **AGRADECIMIENTOS**

Les agradecemos a nuestros familiares por habernos apoyado en todo momento, porque dieron mucho de ellos para que saliéramos adelante con nuestra carrera universitaria, con el fin de que seamos profesionales.

A nuestra tutora por habernos guiado en la investigación del trabajo, por el tiempo que dedicó a la tesis y por ayudarnos cuando más la necesitábamos.

A Jorge Amado Soria Ramírez, amigo nuestro y que nos ayudó muchísimo a pesar de su poco tiempo en el proyecto.

A Ramón E. González Peralta que nos ha ayudado en todo el desarrollo de nuestro trabajo.

A nuestras amistades que han sabido estar presente en las buenas y las malas; y que siempre han comprendido nuestro estado de estrés.

A Fidel y a nuestra Revolución que nos han dado la posibilidad de hacernos profesionales en esta Universidad de Ciencias Informáticas, una universidad de excelencia y de la cual estamos muy orgullosas por habernos graduado en ella.

## DEDICATORIA

***Autora: Maria de los Angeles Rubalcaba Betancourt***

*Un especial agradecimiento a mi abuela, por ser como una madre para mí.*

*A mi madre, que tanto se ha sacrificado por mí, por brindarme amor y cariño.*

*A mi tío Alfonso que ha sido como un padre, a mis tíos, a mis hermanos, a mi sobrina, a mis primos que siempre me han apoyado y me han dado todo lo que he necesitado en la vida: ejemplo, cariño incondicional, consejos, fuerzas para seguir adelante, y no decaer.*

*A mi amiga Yadira, por estar ahí y ser una hermana más para mí. A mi compañera de tesis, Yayneris, gracias por brindarme tu amistad, y haber compartido estos momentos. A mis compañeros que hemos compartido cinco años de estudio y esfuerzo, Daisel, Sulima, Liudmila, Yudita, Yasirys, Ramón.*

*A la memoria de los que no están, pero sé que desde el cielo están orgullosos de mí.*

*A todos aquellos que sienten orgullo de haber aportado un granito de arena en mi formación como persona.*

*A la Revolución, por darme la oportunidad de estudiar en esta Universidad de Excelencia.*

*A todos, muchas gracias.*

***Autora: Yayneris Zambrana Hernández***

*Dedico este trabajo de diploma a mis padres que me quieren y quiero tanto y que siempre han estado conmigo y me han guiado por el buen camino para hacer de mí una mejor persona y una profesional.*

*A mi hermana y su esposo que me han apoyado siempre en todo momento y a mi sobrino que quiero con la vida.*

*A toda mi familia que de una manera u otra me han apoyado a ser cada día mejor y han confiado siempre en mí.*

*A mis amistades, las buenas amistades que me ha dado esta Universidad y que nunca olvidaré, Liván, Yaimé, Jorge Amado, Liu, Maria, Ramón, Liana y los demás que también han sido muy buenos compañeros. Que los extrañaré cuando cada cual tome su camino cuando nos gradúemos, y estemos separados.*

*A Jorge que lo quiero tanto y que siempre me ha dado mucha fuerza para seguir adelante, con su cariño y comprensión.*

*Dedico este trabajo a todas las personas que hicieron posible que llegara a ser lo que soy.*

*Los quiero muchísimo a todos.*

## **RESUMEN**

El trabajo presenta un mecanismo de medición de la calidad de un producto software, para el Proyecto Modernización del CICPC (Cuerpo de Investigaciones Científicas Penales y Criminalísticas). Se brinda una información acerca de Calidad de Software, Control y Aseguramiento de la Calidad, del proceso de pruebas y el proceso de medición; se describe además los diferentes subprocesos que se desarrollan dentro del proceso de evaluación, como son: El proceso de medición, el proceso de recopilación de datos y el proceso de pre-evaluación, y junto a estos se describen además los diferentes artefactos que se generan en los mismos. Se mencionan cuáles son los roles y responsabilidades que se desempeñarán durante todo el proceso de pruebas y de evaluación. Se explica detalladamente el método de evaluación que se propone y se aplica además en el Proyecto Modernización del CICPC, obteniendo resultados que se ajustan a las características y condiciones actuales del proyecto. Guiándose por este trabajo se tiene la posibilidad de conocer la calidad del producto software que se pretende construir. El método de evaluación propuesto puede ser aplicado en otros proyectos productivos de la Universidad de las Ciencias Informáticas con el objetivo de evaluar la calidad del producto durante el proceso de pruebas y la correspondencia con los requisitos pactados con el cliente.

**Palabras Clave:** Calidad, Métricas, Medición, Evaluación, Producto, Prueba.

# ÍNDICE GENERAL

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>FUNDAMENTACIÓN TEÓRICA</b> .....	<b>4</b>
1.1 CALIDAD DE SOFTWARE .....	4
1.2 GARANTÍA Y CONTROL DE LA CALIDAD .....	5
1.3 ESTÁNDAR INTERNACIONAL ISO/IEC 9126.....	5
1.4 MODELO DE EVALUACIÓN ISO/IEC 14598 .....	10
1.5 PROCESO DE EVALUACIÓN DE CMMI-SW (SCAMPI) .....	14
1.5.1 Indicadores de Implementación de Proceso (PIIs) .....	14
1.5.2 Roles y Responsabilidades que define el Modelo CMMI-SW (SCAMPI) .....	15
1.5.3 Ventajas y Desventajas del Modelo de Evaluación que propone CMMI-SW.....	16
1.6 PROCESO DE PRUEBAS.....	17
1.7 PROCESO DE MEDICIÓN DE SOFTWARE .....	19
1.8 PROYECTO MODERNIZACIÓN DEL CICPC. ....	22
1.8.1 Sistema de Investigación e Información Policial.....	23
1.9 CONCLUSIONES DEL CAPÍTULO.....	23
<b>PROCESO DE EVALUACIÓN DE LA CALIDAD</b> .....	<b>25</b>
2.1 DESCRIPCIÓN DEL PROCESO DE EVALUACIÓN DEL SOFTWARE.....	25
2.1.1 Descripción del Proceso de Pre-Evaluación del software.....	26
2.1.2 Descripción del Proceso de Medición.....	27
2.1.2.1 Descripción del Proceso de Recopilación de datos.....	28
2.1.2.2 Propuesta de las métricas a utilizar .....	29
2.1.2.2.1 Métricas de Funcionalidad.....	30
2.1.2.2.2 Métricas de Confiabilidad .....	32
2.1.2.2.3 Métricas de Usabilidad .....	33
2.1.2.2.4 Métricas de Eficiencia .....	34
2.1.3 Descripción del Proceso de Análisis de los resultados de la medición.....	35
2.2 ARTEFACTOS GENERADOS DURANTE EL PROCESO DE EVALUACIÓN .....	40
2.2.1 Registro de las NC generadas durante las pruebas al software .....	40
2.2.2 Informe de los resultados de las pruebas al software .....	42
2.2.3 Informe de Pre-Evaluación del producto software .....	42
2.2.4 Informe de Evaluación del producto software .....	43
2.3 ROLES Y RESPONSABILIDADES EN EL PROCESO DE EVALUACIÓN .....	43
2.4 CONCLUSIONES DEL CAPÍTULO.....	43
<b>EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE SIIPOL</b> .....	<b>44</b>
3.1 RESULTADOS DE LA ENCUESTA PARA LA PRE-EVALUACIÓN DEL PRODUCTO SOFTWARE SIIPOL .....	44
3.2 CÁLCULO DE LAS MÉTRICAS PROPUESTAS PARA MEDIR LA CALIDAD DEL SIIPOL .....	52
3.3 PRINCIPALES PROBLEMAS QUE FUERON DETECTADOS DURANTE EL PROCESO DE EVALUACIÓN .....	57
EN EL SOFTWARE EVALUADO SE ENCONTRARON VARIAS DEFICIENCIAS PRINCIPALMENTE RELACIONADAS CON LA FUNCIONALIDAD, LA CONFIABILIDAD Y LA USABILIDAD. ....	57
3.4 PROPUESTA DE ACCIONES CORRECTIVAS .....	58
3.5 CONCLUSIONES DEL CAPÍTULO.....	58
<b>CONCLUSIONES</b> .....	<b>59</b>
<b>RECOMENDACIONES</b> .....	<b>60</b>
<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	<b>61</b>
<b>BIBLIOGRAFÍA</b> .....	<b>63</b>

<b>GLOSARIO DE TÉRMINOS Y DEFINICIONES .....</b>	<b>65</b>
<b>ANEXOS.....</b>	<b>67</b>

## **ÍNDICE DE FIGURAS**

FIGURA 1: FLUJO DE INFORMACIÓN ENTRE LAS ACTIVIDADES DEL PROCESO DE EVALUACIÓN.....	12
FIGURA 2: PASOS PARA REALIZAR EL PROCESO DE EVALUACIÓN.....	13
FIGURA 3: PROCESO DE MEDICIÓN.....	21
FIGURA 4: PROCESO DE EVALUACIÓN DEL PRODUCTO SOFTWARE.....	26
FIGURA 5: PROCESO DE PRE-EVALUACIÓN DEL PRODUCTO SOFTWARE .....	27
FIGURA 6: PROCESO DE MEDICIÓN DEL PRODUCTO SOFTWARE .....	28
FIGURA 7: PROCESO DE RECOPIACIÓN DE DATOS .....	29
FIGURA 8: PROCESO DE ANÁLISIS DE LOS RESULTADOS DE LA MEDICIÓN .....	35

## **ÍNDICE DE TABLAS**

TABLA 1: TABLA DE MÉTRICAS DE FUNCIONALIDAD.....	30
TABLA 2: TABLA DE MÉTRICAS DE CONFIABILIDAD .....	32
TABLA 3: TABLA DE MÉTRICAS DE USABILIDAD.....	33
TABLA 4: TABLA DE MÉTRICAS DE EFICIENCIA.....	34
TABLA 5: PLANTILLA PARA LOS RESULTADOS DE LAS MÉTRICAS APLICADAS EN LA EVALUACIÓN .....	36
TABLA 6: ESCALA PARA CUANDO SE ACERCA A 1.....	36
TABLA 7: ESCALA PARA CUANDO SE ACERCA A 0.....	37
TABLA 8: PLANTILLA PARA LA PUNTUACIÓN ASIGNADA A LAS CARACTERÍSTICAS Y SUBCARACTERÍSTICAS DE CALIDAD .....	37
TABLA 9: PLANTILLA PARA LA PUNTUACIÓN GENERAL DE CADA UNA DE LAS CARACTERÍSTICAS .....	39
TABLA 10: GRADO DE CONFORMIDAD GENERAL .....	39
TABLA 11: CRITERIO DE EVALUACIÓN.....	40
TABLA 12: RESULTADO DE LA ENCUESTA DE PRE-EVALUACIÓN .....	44
TABLA 13: RESULTADOS DE LAS MÉTRICAS DE FUNCIONALIDAD.....	52
TABLA 14: RESULTADOS DE LAS MÉTRICAS DE CONFIABILIDAD .....	53
TABLA 15: RESULTADOS DE LAS MÉTRICAS DE USABILIDAD.....	53
TABLA 16: RESULTADOS DE LAS MÉTRICAS DE EFICIENCIA.....	53
TABLA 17: RESULTADOS DE LAS MÉTRICAS APLICADAS EN LA EVALUACIÓN .....	54
TABLA 18: PUNTUACIÓN ASIGNADA A LAS SUBCARACTERÍSTICAS DE CALIDAD .....	55
TABLA 19: PUNTUACIÓN ASIGNADA A LAS CARACTERÍSTICAS DE CALIDAD.....	56
TABLA 20: GRADO DE CONFORMIDAD DEL SIIPOL .....	57
TABLA 21: CRITERIO DE EVALUACIÓN DEL SIIPOL .....	57



## INTRODUCCIÓN

La necesidad de construir un producto software con la calidad máxima requerida y que además satisfaga al cliente en la mayor medida posible es un reto que tienen hoy en día las empresas productoras de software. Queda claro que un software con calidad es uno de los principios fundamentales que hace sentir al cliente confianza en su producto. Pero ¿cómo saber que el producto cuenta con la calidad que se requiere?

Actualmente, en la Universidad de las Ciencias Informáticas (UCI) se le realizan pruebas al producto software cuando se finaliza el mismo, lo que provoca que se detecten problemas que debieron haberse evitado en etapas tempranas de su elaboración; lo que conlleva a un aumento del costo y el tiempo planificado para su construcción, provocando un atraso en el cronograma de trabajo e incumplimiento con lo pactado con el cliente.

Otro de los puntos esenciales que no se tiene en cuenta es la evaluación de la conformidad del producto, que tiene como objetivo fundamental obtener una medida cuantitativa de la calidad con la que cuenta el mismo y si cumple o no con los requerimientos y necesidades o expectativas del cliente. Esta evaluación de la conformidad resulta necesario realizarla en cada uno de los procesos de prueba para tener siempre en foco las restricciones impuestas por el cliente en la Pre-evaluación.

Para realizar un buen proceso de pruebas se necesita establecer una Estrategia de Pruebas bien diseñada y un mecanismo que permita evaluar la calidad durante el desarrollo de este proceso. De esta manera se puede tener una idea de cómo se llevó a cabo el proceso de pruebas y también se puede detectar cuáles fueron las principales deficiencias que afectaron la calidad del software para proceder a su mitigación tomando las medidas necesarias y acciones correctivas para contrarrestarlos.

El proceso de evaluación le brinda al equipo evaluador criterios para liberar o aceptar el producto software, garantizando el desarrollo de un producto que satisfaga, en la mayor medida posible, los requerimientos del cliente, indicando la medida de calidad buscada en la producción.

Dada la situación problemática anteriormente planteada el **problema científico** es: ¿Cómo evaluar la Calidad de un producto software durante el proceso de pruebas?

En la presente investigación se tiene como **Objeto de estudio** el Proceso de pruebas y la medición de la calidad del producto software, siendo el **Campo de Acción** la

Evaluación de la Calidad de un producto software durante el proceso de pruebas en el Proyecto Modernización del CICPC.

Para el desarrollo de la investigación se trazó como **Objetivo General** definir un plan de evaluación de software en el Proyecto Modernización del CICPC para la medición de la calidad del producto aplicando métricas que permitan detectar las deficiencias y recomendar soluciones a las mismas. Del mismo, se pueden derivar los siguientes

**Objetivos Específicos:**

- ❖ Elaborar un Plan de Evaluación del Sistema de Seguridad del CICPC validado por expertos del tema.
- ❖ Definir las métricas que se utilizarán durante el proceso de pruebas en el Proyecto Modernización del CICPC.
- ❖ Controlar y Evaluar la calidad del software durante el proceso de pruebas.

Se parte de la siguiente **Idea a Defender:**

Se logrará obtener un mayor control de la calidad del producto si se define un plan de evaluación para medir la calidad del producto durante el Proceso de Pruebas del Proyecto Modernización del CICPC.

Para dar cumplimiento a los objetivos de la investigación se definieron una serie de

**Tareas a desarrollar**, tales como:

1. Investigar sobre cómo se lleva a cabo el Proceso de Pruebas.
2. Investigar sobre métricas de Calidad.
3. Definir un método que indique cómo deben ser registrados y analizados los datos recopilados.
4. Diseñar plantillas donde se registrarán los datos.
5. Proponer métricas que se puedan aplicar durante el Proceso de Pruebas de un Proyecto Productivo.
6. Evaluar los resultados después de aplicar las métricas durante el Proceso de Pruebas.
7. Enumerar las posibles deficiencias que pueden aparecer en este proceso.
8. Recomendar soluciones a estas deficiencias.

Para el desarrollo de esta investigación se usaron métodos empíricos y teóricos. A continuación se definirá cada uno de estos:

Entre los **Métodos empíricos** se encuentran:

- ❖ **Observación:** al determinar el método de recolección de datos para la evaluación del software y escoger el modelo de calidad que guiará el desarrollo del proceso medición del software.
- ❖ **Entrevista:** al entrevistar a especialistas y expertos en el tema que se quiere investigar y al responsable o encargado de este proceso en el proyecto.
- ❖ **Medición:** al obtener una información numérica de los distintos indicadores de calidad durante todo el proceso de pruebas.
- ❖ **Encuesta:** usada para conformar las estadísticas de las características de la calidad que el software a evaluar debe cumplir.

Entre los **Métodos teóricos** se encuentran:

- ❖ **Histórico-Lógico:** al realizar estudios de trabajos anteriores asociados a la propuesta que se pretende construir, y para utilizar estos como punto de referencia.

El presente trabajo de diploma consta de 3 capítulos bien estructurados y de forma organizada para un buen entendimiento del lector.

El primer capítulo presenta el estado del arte abordando de las diferentes Normas y Modelos de Calidad que hicieron posible el desarrollo del trabajo, tales como: ISO/IEC 9126, ISO/IEC 14598, y el Modelo CMMI-SW (Método SCAMPI). Trata además algunos conceptos de calidad de software, así como el control de la calidad. Inserta una breve descripción del proceso de pruebas en un proyecto productivo y el proceso de medición del producto software.

El segundo capítulo describe de forma organizada y detallada como se le dará solución a la situación problemática presentada. Se presenta la propuesta de las métricas que medirán la calidad del producto software Sistema de Información e Investigación Policial (SIIPOL) en el Proyecto Productivo Modernización del CICPC.

En el tercer capítulo se presentan los resultados reales después de haber aplicado la propuesta de solución en el SIIPOL, y se presenta además una serie de deficiencias encontradas con sus posibles recomendaciones para que se corrijan de forma temprana y puedan ser realizadas en caso de que se presenten situaciones similares en etapas posteriores de pruebas.

# 1

## CAPÍTULO FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se aborda acerca de conceptos como calidad de software, garantía y control de la calidad. Se describen además, algunas Normas y Modelos de Calidad de Software utilizados como guía para el desarrollo del proceso de evaluación del producto software. En el capítulo se describirá además acerca del Proceso de Pruebas a un producto software, y el proceso de medición de la calidad. Y por último se brindará una breve información del CICPC y la solución de software SIIPOL.

### 1.1 Calidad de software

Muchos Estándares Internacionales definen la calidad de un producto como:

*“Grado con el cual el cliente o usuario percibe que el software satisface sus expectativas”. [1]*

Según Roger Pressman la calidad de un producto software no es más que 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”. [2]*

En el mundo actual, donde la comercialización de productos software es cada vez mayor, la calidad que se espera del mismo es importante que sea satisfactoria y que se esté conforme con la misma; ya que representa el grado de satisfacción del cliente, y si cumple o no con las expectativas que se espera del mismo.

## 1.2 Garantía y Control de la Calidad

Según la Norma ISO 8402 **Garantía de Calidad** es un:

*“Conjunto de acciones planificadas y sistemáticas necesarias para proporcionar la confianza adecuada de que un producto o servicio satisfará los requerimientos dados sobre calidad”. [3]*

La misma norma define el **Control de calidad** como:

*“Conjunto de técnicas y actividades de carácter operativo, utilizadas para verificar los requerimientos relativos a la calidad del producto o servicio”. [3]*

Durante el proceso de desarrollo de un producto software se deben planificar, organizar, dirigir y controlar una serie de actividades sobre las cuales se debe tener cierto control de la calidad con que se realizan las mismas, para de esta forma tratar de garantizar que se obtenga un producto con la calidad requerida y que se cumplan los objetivos previstos desde el inicio. Se debe tener en cuenta que según sea el control de la calidad durante el desarrollo del producto, mayor será la garantía que se tenga de la calidad del software al finalizar el mismo.

## 1.3 Estándar Internacional ISO/IEC 9126

Según la Organización Internacional para la Estandarización (ISO), un estándar es “un conjunto de acuerdos documentados que contienen especificaciones técnicas u otros criterios precisos para ser usados constantemente, como reglas, lineamientos o definiciones de características. Todo esto con la finalidad de asegurar que los materiales, productos, procesos y servicios son óptimos para su propósito”. [4]

La norma **ISO/IEC 9126** es un estándar internacional para la evaluación del software y está enfocada a la calidad del producto. Está supervisado por el proyecto SQuaRE (Software Quality Requirements and Evaluation), ISO 25000:2005, el cual sigue los mismos conceptos.

La especificación y la evaluación de la calidad de producto software se puede conseguir definiendo características de calidad apropiadas, tomando en cuenta el objetivo de uso del producto software.

El estándar está dividido en cuatro partes las cuales dirigen, respectivamente, lo siguiente:

Parte 1: Modelo de Calidad

Parte 2: Métricas Externas

Parte 3: Métricas Internas

Parte 4: Métricas de Calidad en el Uso

El modelo de calidad establecido en la primera parte del estándar, ISO/IEC 9126-1, clasifica la calidad del software en un conjunto estructurado de **Características y Subcaracterísticas** de la siguiente manera:

❖ **Funcionalidad:**

Es la capacidad del software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas cuando el software se usa bajo las condiciones especificadas.

- **Idoneidad:**

Capacidad del software para mantener un conjunto apropiado de funciones para las tareas y los objetivos del usuario especificados.

- **Exactitud:**

Capacidad del software para proporcionar efectos o resultados correctos o convenidos con el grado de exactitud necesario.

- **Interoperabilidad:**

Capacidad del producto software para interactuar recíprocamente con uno o más sistemas especificados.

- **Seguridad:**

Capacidad del producto software para proteger información y los datos, para que personas o sistemas desautorizados no puedan leer o modificar los mismos, y las personas o sistemas autorizados tenga el acceso a ellos.

- **Conformidad con la Funcionalidad:**

Capacidad del software para adherirse a las normas que se le apliquen, convenciones, regulaciones, leyes y las prescripciones similares relativos a la funcionalidad.

❖ **Confiabilidad:**

La capacidad del producto software para mantener un nivel de ejecución especificado cuando se usa bajo las condiciones especificadas.

- **Madurez:**  
Capacidad del producto software de evitar un fallo total como resultado de haberse producido un fallo del software.
- **Recuperabilidad:**  
Capacidad del producto software de restablecer un nivel de ejecución especificado y recuperar los datos directamente afectados en caso de fallo total.
- **Tolerancia a fallos:**
- Capacidad del producto software de mantener un nivel de ejecución o desempeño especificado en caso de fallos del software o de infracción de su interface especificada.
- **Conformidad con la Confiabilidad:**  
Capacidad del producto software para adherirse a las normas que se le apliquen, convenciones, regulaciones, leyes y las prescripciones similares relativos a la confiabilidad.

❖ **Usabilidad:**

Capacidad del producto software de ser comprendido, aprendido, utilizado y de ser atractivo para el usuario, cuando se utilice bajo las condiciones especificadas.

- **Comprensibilidad:**  
Capacidad del producto software para permitirle al usuario entender si el software es idóneo, y cómo puede usarse para las tareas y condiciones de uso particulares.
- **Cognoscibilidad:**  
Capacidad del producto software para permitirle al usuario aprender su aplicación.
- **Operabilidad:**  
Capacidad del producto software para permitirle al usuario operarlo y controlarlo.
- **Atracción:**  
Capacidad del producto software de ser atractivo o amigable para el usuario.
- **Conformidad con la Usabilidad:**  
Capacidad del producto software para adherirse a las normas, convenciones, guías de estilo o regulaciones relativas a la usabilidad.

#### ❖ **Eficiencia:**

Capacidad del producto software para proporcionar una ejecución o desempeño apropiado, en relación con la cantidad de recursos utilizados bajo condiciones establecidas.

- **Rendimiento:**

Capacidad del producto software para proporcionar apropiados tiempos de respuesta y procesamiento, así como tasas de producción de resultados, al realizar su función bajo condiciones establecidas.

- **Utilización de recursos:**

Capacidad del producto software para utilizar la cantidad y el tipo apropiado de recursos cuando el software realiza su función bajo las condiciones establecidas.

- **Conformidad con la Eficiencia:**

Capacidad del producto software de adherirse a las normas o convenciones que se relacionan con la eficiencia.

#### ❖ **Mantenibilidad:**

Capacidad del producto software de ser modificado. Las modificaciones pueden incluir las correcciones, mejoras o adaptaciones del software a cambios en el ambiente, así como en los requisitos y las especificaciones funcionales.

- **Diagnosticabilidad:**

Capacidad del producto 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.

- **Flexibilidad:**

Capacidad del producto software para permitir la aplicación de una modificación especificada.

- **Estabilidad:**

Capacidad del producto software para minimizar los efectos inesperados de las modificaciones realizadas al software.

- **Contrastabilidad:**

Capacidad del producto software para permitir la validación de un software modificado.

- **Conformidad con la Mantenibilidad:**



Capacidad del producto software para adherirse a las normas o convenciones que se relacionan con la mantenibilidad.

❖ **Portabilidad:**

Capacidad de producto software de ser transferido de un ambiente a otro.

- **Adaptabilidad:**

Capacidad del producto 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.

- **Instalabilidad:**

Capacidad del producto software de ser instalado en un ambiente especificado.

- **Coexistencia:**

Capacidad del producto software de coexistir con otro software independiente en un ambiente común y compartir los recursos comunes.

- **Remplazabilidad:**

Capacidad del producto software de ser usado en lugar de otro producto software especificado para los mismos fines y en el mismo ambiente.

- **Conformidad con la Portabilidad:**

Capacidad del producto software de adherirse a las normas o convenciones relativas a la portabilidad.

La norma provee un entorno para que las organizaciones definan un modelo de calidad para el producto software. Esto podría ser hecho, por ejemplo, especificando los objetivos para las métricas de calidad que evalúan el grado de presencia de los atributos de calidad.

La NC ISO/IEC 9126 define los tipos de métricas existentes para evaluar la calidad del producto software, tales como:

Las **Métricas Internas** permiten atender los aspectos de la calidad desde las etapas más tempranas antes de que el producto software devenga ejecutable. [5]

Las **Métricas Externas** permiten evaluar la calidad del producto software durante el ensayo ó la operación. [5]

Las **Métricas de Calidad en el Uso** miden hasta qué punto un producto satisface las necesidades de usuarios específicos para lograr las metas especificadas con la eficacia, productividad, seguridad y satisfacción en un contexto determinado de uso.[5]

Las métricas internas son utilizadas internamente para que el equipo de desarrollo evalúe la calidad de su producto mientras que las métricas externas se utilizan por un evaluador externo que certifica la calidad del producto antes de ser entregado al cliente. Idealmente, la calidad interna determina la calidad externa y esta a su vez la calidad en el uso.[5]

#### **1.4 Modelo de Evaluación ISO/IEC 14598**

La serie de normas ISO/IEC 14598 indica los requisitos a tener en cuenta para la aplicación de los métodos de medición y para el proceso de evaluación, proporcionando un entorno de trabajo para la evaluación de la calidad de diferentes tipos de productos software.

La ISO/IEC 14598 consta de seis partes que especifican el proceso a seguir para evaluar software:[6]

ISO/IEC 14598-1 Visión general.

ISO/IEC 14598-2 Planificación y Gestión.

ISO/IEC 14598-3 Procedimiento para desarrolladores.

ISO/IEC 14598-4 Procedimiento para compradores.

ISO/IEC 14598-5 Procedimiento para evaluadores.

ISO/IEC 14598-6 Documentación de los módulos de evaluación.

La norma ISO/IEC 14598 se pretende usar por desarrolladores, compradores y evaluadores independientes, particularmente aquellos responsables de la evaluación del producto software. [7]

La quinta parte de la Norma ISO/IEC 14598 (ISO/IEC 14598-5) se ajusta a las necesidades actuales del equipo ya que especifica y explica el procedimiento por el cual los evaluadores del software se pueden guiar para desarrollar este proceso de evaluación. [7]

Esta parte de la Norma proporciona requisitos y recomendaciones para la implantación práctica de la evaluación del producto software. Puede ser usada para aplicar los conceptos descritos en la Norma ISO/IEC 9126. [8]

Uno de los principales **Objetivos del proceso de evaluación** descrito en esta parte de la Norma ISO/IEC 14598 es promover las siguientes características deseables en el proceso de evaluación:

- *Repetibilidad*: La evaluación repetida de mismo producto con respecto a la misma especificación de evaluación por el mismo evaluador debería producir resultados que pueden ser aceptados por ser idénticos.
- *Reproducibilidad*: La evaluación repetida del mismo producto con respecto a la misma especificación de evaluación por un evaluador diferente debería producir resultados que pueden ser aceptados por ser idénticos.
- *Imparcialidad*: La evaluación no debería estar orientada hacia un resultado particular.
- *Objetividad*: Los resultados de la evaluación deberían ser ciertos, por ejemplo, no influidos por los sentimientos u opiniones del evaluador. [6]

El principal **Propósito de la evaluación del software** es apoyar directamente tanto el desarrollo como la adquisición de un software que satisfaga las necesidades del usuario y del cliente teniendo como objetivo final asegurar que el producto aporte la calidad requerida y satisfaga las necesidades declaradas e implícitas de los usuarios. [6]

Esta norma plantea cinco **Actividades del proceso de evaluación** las cuales se listan a continuación:

1. Establecimiento de los requisitos de evaluación.
2. Especificación de la evaluación basada en los requisitos de evaluación y en la descripción del producto suministrado por el solicitante.
3. Diseño de la evaluación, que produce un plan de evaluación sobre la base de la especificación de evaluación; esta actividad tiene en cuenta los componentes del producto software a ser evaluados y los métodos de evaluación propuestos para el evaluador.
4. Ejecución del plan de evaluación, que consiste en inspeccionar, modelar, medir, y probar los productos y sus componentes de acuerdo al proceso de evaluación.

5. Conclusión de la evaluación, que consiste en la entrega del informe de evaluación. [6]

Los resultados de la evaluación que se obtienen a partir de la aplicación de ISO/IEC 14598 los pueden utilizar los administradores y desarrolladores/sostenedores para medir el cumplimiento de los requisitos y para realizar mejoras cuando sea necesario. Los resultados de la evaluación también los pueden utilizar los analistas para establecer las relaciones entre las métricas internas y externas. El personal de mejora de los procesos puede utilizar los resultados de la evaluación para definir cómo se pueden mejorar los procesos mediante el estudio y examen de la información de calidad del producto del proyecto. [6]

La siguiente figura da una visión general donde se identifica el flujo de información entre actividades. (Ver figura 1)

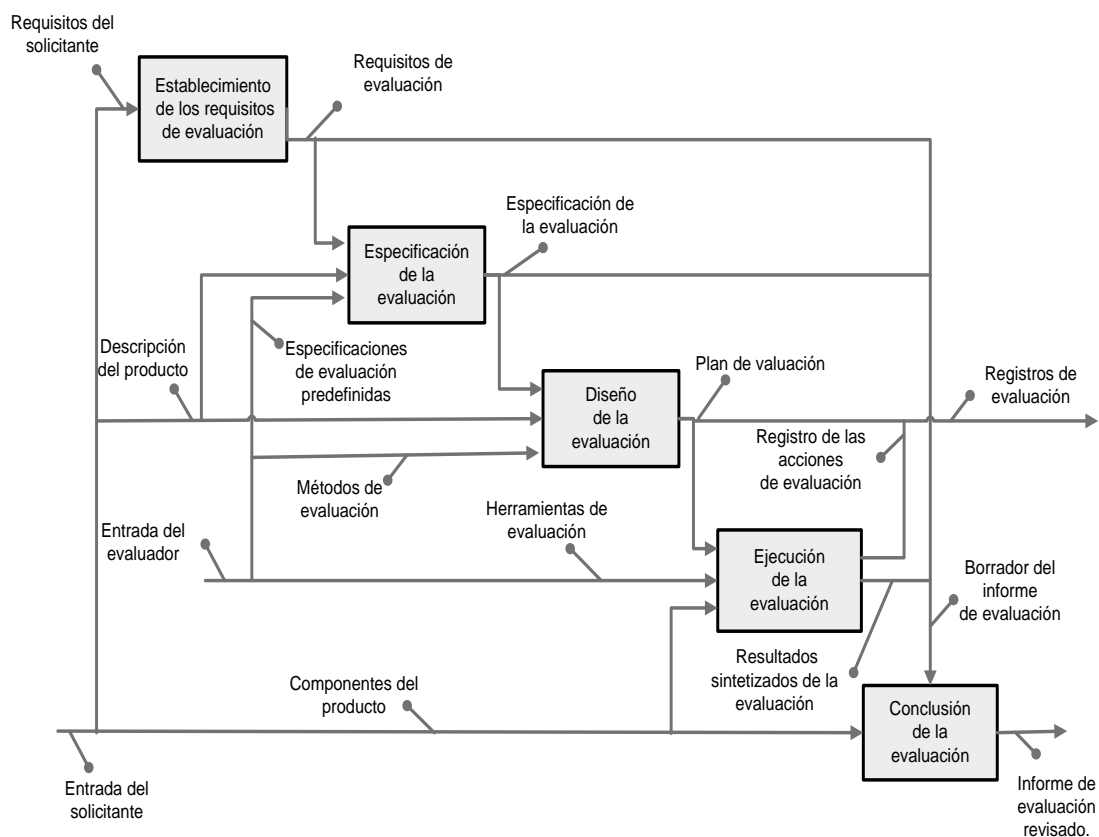


Figura 1: Flujo de información entre las actividades del proceso de evaluación

Como se puede observar en la figura anterior, el primer paso del flujo de actividades del proceso de evaluación es establecer los requisitos, teniendo en cuenta previamente los requisitos del solicitante. Posteriormente se especifica la evaluación, para desarrollar su diseño teniendo en cuenta el método que se usará. Una vez culminadas estas actividades, entonces se ejecuta la evaluación y finalmente se da a conocer la conclusión de la misma, mostrándose los resultados sintetizados en un Informe de Evaluación.

Actualmente el Comité Técnico Conjunto **ISO/IEC JTC1** Tecnologías de la Información / **CS7** Ingeniería de Software y Sistemas está revisando ambas series de normas, y está elaborando una nueva serie, la **ISO/IEC 25000**, bajo el título general de Requerimientos de calidad y evaluación de software conocido como **SQuaRE**, que abarcará a las series ISO/IEC 14598 e ISO/IEC 9126. [7]

En la siguiente figura (Ver figura 2) se muestra una serie de actividades a tener en cuenta para realizar el proceso de evaluación, indicando las normas por las cuales regirse para llevar a cabo este proceso. Para evaluar la calidad del software primero se establecen los requisitos de evaluación, después se especifican, diseñan y se ejecuta la evaluación.

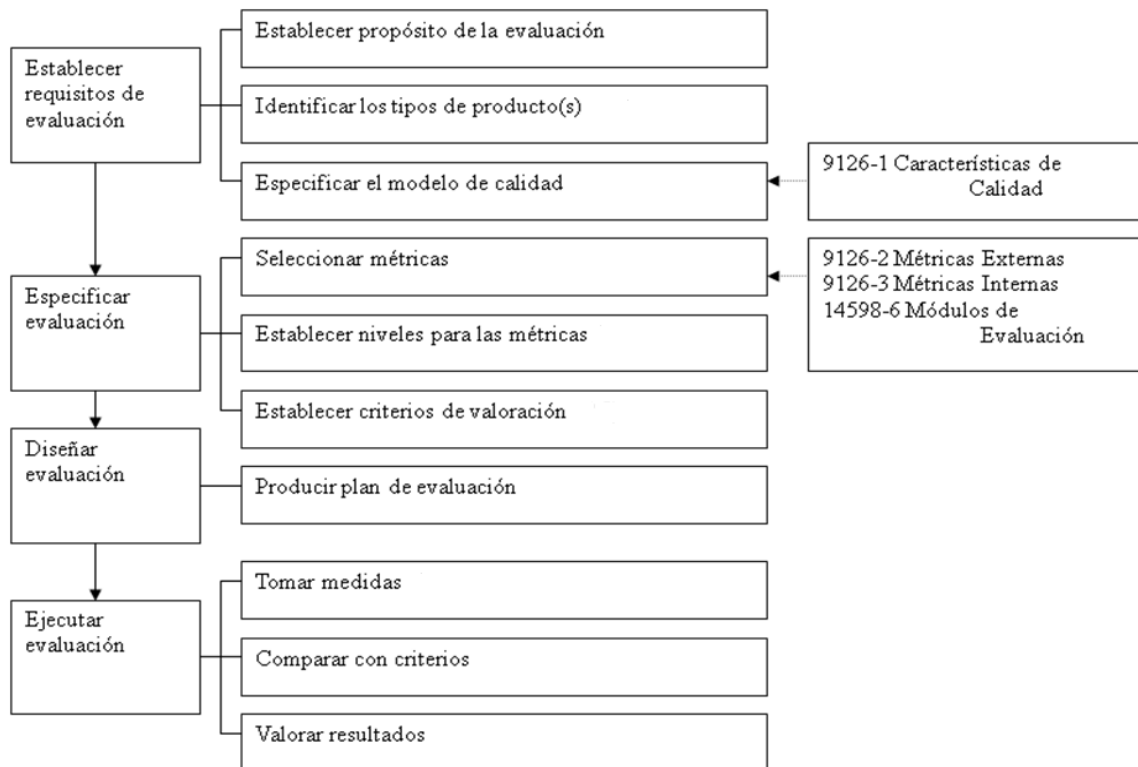


Figura 2: Pasos para realizar el proceso de evaluación

## **1.5 Proceso de Evaluación de CMMI-SW (SCAMPI)**

El Modelo de Madurez de las Capacidades (CMMI-SW) es un conjunto de modelos elaborados por el Instituto de Ingeniería de Software de la Universidad Carnegie Mellon, California (SEI) que permiten obtener un diagnóstico preciso de la madurez de los procesos relacionados con las tecnologías de la información de una organización, y describen las tareas que se tienen que llevar a cabo para mejorar esos procesos. [9]

El Método de Evaluación para Mejora de Procesos (SCAMPI) está diseñado para proporcionar medidas de evaluación de la calidad. Dicho de otro modo, SCAMPI permite determinar el nivel de madurez de una organización en relación al modelo CMMI-SW. Es aplicable a un rango amplio de modos de uso de la evaluación, incluyendo la mejora del proceso interno y la determinación de la capacidad externa. Generalmente el grupo evaluador se conforman por integrantes de la misma empresa que es evaluada. [10]

Según González [11] los objetivos de SCAMPI son:

- ❖ Identificar los puntos fuertes y las debilidades de los procesos.
- ❖ Relacionar estos puntos fuertes y debilidades con el modelo CMMI-SW.
- ❖ Centrarse en las mejoras (corregir las debilidades que generan riesgos) que son más beneficiosas a la organización, dado su nivel general de madurez organizativa o capacidades de proceso.
- ❖ Obtener una valoración del nivel de madurez.
- ❖ Identificar los riesgos de desarrollo/adquisición relativos a las determinaciones de capacidad/madurez.
- ❖ Proporcionar un método de evaluación común e integrado, capaz de soportar evaluaciones en el contexto de la mejora de los procesos internos, selección de proveedores y supervisión de procesos.
- ❖ Proporcionar un método de evaluación eficiente, capaz de ser implementado con restricciones de ejecución razonables.

### **1.5.1 Indicadores de Implementación de Proceso (PIIs)**

En SCAMPI, los PIIs son la consecuencia necesaria de realizar las prácticas del Modelo CMMI-SW. SCAMPI caracteriza los PIIs de acuerdo con los tipos de indicadores descritos a continuación: [12]

**Artefactos Directos:** Las salidas tangibles resultan directamente de la realización de prácticas específicas o genéricas. Puede ser indicado o implicado explícitamente por la declaración de la práctica o el material informativo asociado.

Por ejemplo:

- ❖ Documentos.
- ❖ Productos entregables.
- ❖ Materiales de entrenamiento.

**Artefactos Indirectos:** Son elementos que surgen como consecuencia de la implementación de una práctica del Modelo CMMI-SW, pero que no constituyen en sí el propósito de la práctica. Este tipo de indicador es especialmente útil cuando puede haber dudas sobre si el propósito de la práctica ha sido alcanzado.

Por ejemplo:

- ❖ Minutas de reunión.
- ❖ Resultados de revisiones.
- ❖ Informes de estado.
- ❖ Realización de medidas.

**Observaciones:** Declaraciones orales o escritas confirmando o dando soporte a la realización de una práctica genérica o específica. Estas son proporcionadas normalmente por los que realizan la práctica y/o clientes internos o externos, pero puede también incluir a cualquier persona interesada (directores, proveedores, etc).

Por ejemplo:

- ❖ Respuestas a cuestionarios.
- ❖ Entrevistas.
- ❖ Presentaciones.

### **1.5.2 Roles y Responsabilidades que define el Modelo CMMI-SW (SCAMPI)**

Los roles que se definen en este modelo son:

Evaluados: [13]

- ❖ Patrocinador: directivo que solicita la evaluación, respalda el proceso, define el alcance y recibe los resultados de la misma.
- ❖ Gerentes: responsables de áreas de la organización que revisan los resultados preliminares y participan en discusiones grupales.

- ❖ Líderes de Proyectos, Analistas, Programadores y Técnicos: participan en entrevistas personales y revisan los resultados preliminares.

#### Evaluadores: [13]

- ❖ Líder (Lead Appraisal o Team Leader): responsable de la evaluación, capacitado y autorizado por el SEI.
- ❖ Integrante del Equipo (Team Member): persona con experiencia en ingeniería de software y sólidos conocimientos del Modelo CMMI-SW.
- ❖ Equipo Evaluador (Team): entre 4 y 8 personas recolectan evidencia, realizan las entrevistas, evalúan el cumplimiento de las prácticas, etc
- ❖ Mini equipo (Mini Team): dos o más integrantes responsables evaluar Áreas de Procesos Particulares.

### **1.5.3 Ventajas y Desventajas del Modelo de Evaluación que propone CMMI-SW**

Las ventajas y desventajas que proporciona el método de evaluación SCAMPI se exponen a continuación:

#### **Ventajas**

- ❖ Guía paso a paso la mejora, a través de niveles de madurez y capacidad. [15]
- ❖ La organización sabe el estado actual en que se encuentra la empresa.
- ❖ Reducción consistente de errores (reduciendo el número de defectos y detección en las fases más tempranas del ciclo de vida). [16]
- ❖ Reduce costes de desarrollo y mantenimiento. [17]

#### **Desventajas**

- ❖ Indican a las organizaciones qué actividades han de realizar, pero nada sobre cómo hacerlo. [18]
- ❖ La carga de papeleo que impone el modelo, viéndose más como un mecanismo de control por la dirección que como una herramienta que ayude en el trabajo.[19]
- ❖ La aplicación del modelo requiere una inversión importante y a largo plazo, lo cual supone un obstáculo importante para la pequeña empresa. [15]



## 1.6 Proceso de Pruebas

Un proceso de pruebas en un Proyecto Productivo se debe caracterizar siempre por la buena planificación y control de todas las actividades en general, para garantizar que se cumpla el trabajo como está establecido y que se obtenga un producto con la mejor calidad posible.

*“La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación”. [2]*

Otro criterio define la prueba como:

*“... una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente”. [20]*

Los resultados arrojados en las pruebas aplicadas al producto software se documentan para posteriormente hacer una evaluación de los mismos, analizando cuáles fueron las principales deficiencias y tomando ciertas medidas para evitar que esto vuelva a ocurrir. En una etapa de pruebas al software se pueden realizar máximo tres iteraciones de pruebas, para evitar que se extienda el proceso y que sea engorroso el mismo y de esta forma se tratará de encontrar la mayor cantidad de defectos posibles. En caso que se necesite seguir probando el software se comenzará entonces una segunda etapa de pruebas.

Glen Myers afirma que: *“La prueba no puede asegurar la ausencia de defectos; solo puede demostrar que existen defectos en el software”. [21]*

Existen varios niveles de prueba definidos y estos niveles a su vez especifican qué tipos y métodos de pruebas se deben utilizar en cada uno de ellos y cuáles son sus objetivos. A continuación se hace referencia a los niveles de pruebas:

- ❖ Unitarias:

- Enfocadas al código fuente de los componentes.

- Para verificar todos los flujos de control.

- Primero pasa por la revisión del programador.

- ❖ De integración:

- Prueba los componentes combinados para ejecutar un CU.

Para verificar, descubrir errores o incompletitud en las especificaciones de las interfaces de las clases.

❖ De sistema:

Prueba el software funcionando como un todo.

Aceptable para cuando el software se encuentra en la Fase de Construcción.

Trata de probar que los objetivos para los que fue construida la aplicación no se cumplen en su totalidad y que por tanto hay que cambiar cosas en la aplicación.

Se usa como base los objetivos originales.

No existe un método en específico, sino que se dan lineamientos, a la hora de preparar los casos de prueba.

Se finaliza cuando se cumplieron los meses o las semanas programadas y se han hallado N errores.

❖ De liberación:

Prueba el software funcionando como un todo.

Trata de probar que los objetivos para los que fue construida la aplicación no se cumplen en su totalidad y que por tanto el software no está en condiciones de ser presentado al cliente y hay que hacer modificaciones a la aplicación.

Se realiza cuando el software funciona como un todo y está prácticamente listo para ser presentado al cliente.

❖ De aceptación:

Prueba el software funcionando como un todo, se realiza dándole un uso real a la aplicación y es realizada por parte de los clientes, los errores que se encuentren en la misma son reportados como defectos.

Las pruebas de sistema están divididas en dos grandes grupos:

❖ **Funcionales:**

Enfocadas a los requisitos funcionales del software, a su interacción con el cliente de la forma que ha sido pactada.

❖ **No funcionales:**

Enfocadas a los requisitos no funcionales del proyecto. Desglosadas en un número de pruebas definidas como:

- ✓ Prueba de Seguridad

- ✓ Prueba de Disponibilidad y Red
- ✓ Prueba de Rendimiento o Carga
- ✓ Prueba de Compatibilidad
- ✓ Prueba de Resistencia o Stress
- ✓ Prueba de Usabilidad
- ✓ Prueba de Fiabilidad

Existen principios planteados por Davis, Edward y Myers que guían las pruebas de software tales como: [2]

1. La principal dificultad del proceso de prueba es decidir cuándo parar.
2. Evitar casos de pruebas no planificados, no reusables y triviales a menos que el programa sea verdaderamente sencillo.
3. Una parte necesaria de un caso de prueba es la definición del resultado esperado.
4. Los casos de pruebas tienen que ser escritos no solo para condiciones de entradas válidas y esperadas, sino también para condiciones no válidas e inesperadas.
5. Los casos de pruebas tienen que ser escritos para generar las condiciones de salida deseadas.
6. El número de errores sin descubrir es directamente proporcional al número de errores descubiertos.
7. Las pruebas deberían empezar por "lo pequeño" y progresar hacia "lo grande".
8. Con la excepción de las pruebas de unidad e integración, un programa deberá ser probado por la persona u organización que lo desarrolló.
9. Asigna el programador más creativo a la prueba.

### **1.7 Proceso de Medición de Software**

El proceso de medición es conducido por las necesidades de información del proyecto en general. El mismo se desarrolla dentro del proceso de pruebas, y juega un papel fundamental para llevar a cabo la evaluación del producto software, debido a que durante las pruebas se realiza la recopilación de los datos, que servirán de medida para la aplicación posterior de las métricas. La medición del software brinda una visión cuantitativa de cómo se ha ido desarrollando el proceso de desarrollo y si hay deficiencias en el mismo, dando la posibilidad de detectarlas y corregirlas a tiempo.

*“La **medición** nos permite tener una visión más profunda proporcionando un mecanismo para la evaluación objetiva”. [2]*

“Una **medida** proporciona una indicación cuantitativa de la extensión, cantidad, dimensiones, capacidad o tamaño de algunos atributos de un proceso o producto”. [2]

El IEEE Glosario estándar de términos en la Ingeniería de Software define **métrica** como “una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado”. [22]

Un **atributo** es una entidad la cual puede ser verificada o medida en el producto software. [23]

Se recopilan medidas y desarrollan métricas para obtener indicadores. Un **indicador** es una métrica o una combinación de métricas que proporcionan una visión profunda del proceso del software, del proyecto software o del producto en sí. [24]

Los principios fundamentales que deben seguir las métricas según el Proceso Unificado de Rational [25] son:

- Las métricas deben ser simples, objetivas, fáciles de coleccionar, fáciles de interpretar y difíciles de malinterpretar.
- La colección de las métricas debe ser automática y no intrusiva, o sea, no interferir en las actividades de los desarrolladores.
- Las métricas deben contribuir a la evaluación de la calidad temprana en el ciclo de vida, cuando los esfuerzos por mejorar la calidad del software son efectivos.
- Los valores absolutos y las tendencias de las métricas, deben ser usados activamente por el personal administrativo y el personal ingenieril, para comunicar progreso y calidad en un formato coherente.
- La selección de un mínimo o más extensivo conjunto de métricas, dependerá de las características y contexto del proyecto: Si es muy grande o si tiene restricciones de seguridad o de confiabilidad de los requerimientos; y si el equipo de desarrollo y de evaluación es conocedor de las métricas, lo cual hará muy útil coleccionar y analizar las métricas técnicas.
- Indicar la calidad del producto es una de las razones para medir un producto software, además de evaluar la productividad del personal involucrado en el desarrollo del mismo, evaluar los beneficios en cuanto a la productividad y la calidad y poder establecer una línea base para la estimación.

Para el proceso de medición del producto software se definen plantillas en las cuales se recopilan los datos necesarios para llevar a cabo el mismo, obteniendo variables independientes como son las No Conformidades (NC) encontradas como resultado de la ejecución de las pruebas, el tiempo que se tardó en realizarla, entre otros. (Ver figura 3)

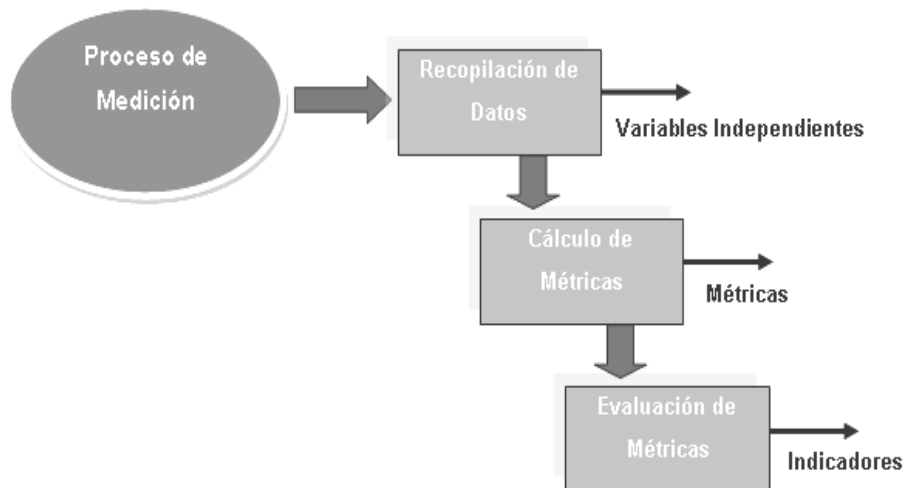


Figura 3: Proceso de medición

Una vez recopilados estos datos se procede al cálculo de las métricas obteniendo valores cuantitativos, que servirán para evaluar la calidad proporcionando una visión profunda del producto.

La medición puede ser aplicada a un proyecto software para ayudar al control y evaluación de la calidad, y de esta manera obtener resultados que facilitarían la toma de decisiones en el mismo a medida que avanza y se desarrolla.

Siempre que se pueda medir la calidad del producto, es recomendable que esta actividad se realice para así tener siempre el valor cuantitativo de la calidad del mismo.

Lord Kelvin dijo en una frase lo siguiente:

*“Cuando pueda medir lo que está diciendo y expresarlo con números, ya conoce algo sobre ello; cuando no pueda medir, cuando no pueda expresar lo que dice con números, su conocimiento es precario y deficiente: puede ser el comienzo del conocimiento, pero en sus pensamientos, apenas está avanzando hacia el escenario de la ciencia”.* [26]

De forma general las métricas ayudan a tener una visión más profunda del producto software, teniendo un seguimiento del mismo e identificando y eliminando lo antes posible las causas de los defectos principales.

### **1.8 Proyecto Modernización del CICPC.**

El CICPC cuenta con una aplicación informática, desarrollada sobre ADABAS y Natural. Cuenta además con algunas aplicaciones que sirven de apoyo a sus actividades de investigación, tales aplicaciones son: el Sistema Integrado de Identificación Balística (IBIS), el Sistema Automático de Identificación de Huellas Dactilares (AFIS), el Sistema Integrado de Información Policial (SIIPOL). También desarrolla un sitio Web de carácter informativo orientado a la población venezolana e internacional.

Inicialmente el SIIPOL era solo para manejar la información policial, actualmente ya incluye: Adiestramiento, Auditoría, Control de visitas, Evaluación de desempeño, Recursos humanos, Investigaciones Internas, entre otros. Además se utilizan datos de otras instituciones como las Bases de Datos (BD) del Sistema Autónomo de Identificación, Migración y Extranjería (SAIME), y la del Instituto Nacional de Tráfico y Transporte Terrestre (INTTT), esta información se trae en soportes digitales para actualizar la BD del SIIPOL.

Esta solución de software debe presentar algunas características que son indispensables en la Aplicación como son el acceso al sistema a través de un navegador Web; un servidor de BD y un servidor de Aplicación que trabajen en conjunto; una red de comunicación constante entre todas las PC clientes y el servidor de Aplicaciones y una comunicación directa entre el servidor de Aplicaciones y el servidor de BD.

El Proyecto Modernización del CICPC consta de seis Módulos ó Subsistemas de Desarrollo, y cada uno de estos Módulos contiene Sub-Módulos que están compuestos por una serie de CU. Además se cuenta con un equipo de Calidad, de Arquitectura, de Base de Datos, de Analistas, y un equipo encargado del Portal Web. Cada unos de estos equipos se subordinan a un líder y un sublíder del proyecto. De manera general, el proyecto está estructurado organizadamente y cada equipo consiente de las tareas que deben realizar.

### **1.8.1 Sistema de Investigación e Información Policial**

El SIIPOL es una aplicación que utiliza el CICPC para realizar actividades de investigación policial, que ayuda a controlar y organizar el trabajo en las dependencias del CICPC.

Los objetivos fundamentales del SIIPOL son tener centralizada toda la información necesaria de modo que el proceso de investigación sea lo más ágil posible. Aumentar el nivel de comunicación entre las diferentes áreas. Disminuir el trabajo manual que se realiza para resolver casos. Proporcionar información actualizada de las personas, vehículos, objetos y armas relacionadas con algún delito. Brindar información a entidades del gobierno y la población en general.

Con el SIIPOL habrá una mayor comunicación entre las diferentes dependencias, la información siempre estará actualizada y organizada teniendo en cuenta los niveles de confidencialidad requeridos. Tendrá un mayor control de las actividades de trabajo del CICPC posibilitando el seguimiento, la evaluación y el control de los procesos.

### **1.9 Conclusiones del capítulo**

Después de haber concluido el análisis bibliográfico de la investigación se ha arribado a la conclusión de que la calidad es un factor muy importante en el producto software respondiendo a los requisitos especificados y a las necesidades y expectativas del cliente o usuario final.

Se abordó sobre los diferentes modelos y normas de calidad, como por ejemplo el Modelo CMMI-SW que propone el método de evaluación SCAMPI que aporta cuáles deberán ser los roles y responsabilidades en un equipo de evaluación. También se describió la ISO/IEC 9126 que se selecciona ya que propone las características y sub características de calidad a medir en un producto software, brindando la posibilidad de emplear y aplicar las métricas internas que la misma propone. Otra de las normas que se describió fue la ISO/IEC 14598, la cual se selecciona gracias a que es una guía para los evaluadores de software, describe cómo debe llevarse a cabo el proceso de evaluación además de los artefactos que deben generarse del mismo.

Uno de los epígrafes abordó acerca del proceso de pruebas, definiéndose los niveles y tipos de pruebas que existen, y se mencionaron además los principios de las pruebas. También se abarcó el proceso de medición, indicando qué es una métrica, una medida y demás conceptos relacionados con la medición de la calidad. Se describió en qué

consiste el proceso, explicando cada una de las actividades que componen el mismo, así como la recopilación de datos, el cálculo de las métricas seleccionadas y la evaluación de las mismas. Se presentó una breve descripción del Proyecto Modernización del CICPC, centrándose en las características fundamentales del SIIPOL, sus objetivos específicos y los beneficios que traerá el uso de esta aplicación en la República Bolivariana de Venezuela.



## **CAPÍTULO**

### **PROCESO DE EVALUACIÓN DE LA CALIDAD**

En el presente capítulo se abordará acerca de la descripción del proceso de evaluación de la calidad de un producto. Presentará los diferentes subprocesos que se encuentran dentro de la evaluación, como son el proceso de pre-evaluación, de medición, de recopilación de datos y el proceso de análisis de los resultados de la medición. En cada uno de estos subprocesos se modelan las diferentes actividades y los principales artefactos que se generan de cada una de ellas. Se mencionarán además los principales roles y responsabilidades que juegan un papel importante dentro del proceso de evaluación, y finalmente se presentará la propuesta para evaluar la calidad del producto software.

#### **2.1 Descripción del Proceso de Evaluación del software**

El Proceso de Evaluación consta de tres subprocesos dentro de él, primero el Proceso de Pre-evaluación, donde se definen las características y subcaracterísticas que se medirán en el software. Segundo el Proceso de Medición que dentro del mismo se encuentra el proceso de recopilación de datos, para la aplicación de las métricas. Y por último el Proceso de Análisis de los resultados de la medición, en este proceso se analiza los resultados y se detectan las deficiencias en el software y además se recomiendan soluciones. En la siguiente figura se muestran los diferentes subprocesos que recoge el Proceso de Evaluación del producto software. (Ver figura 4)

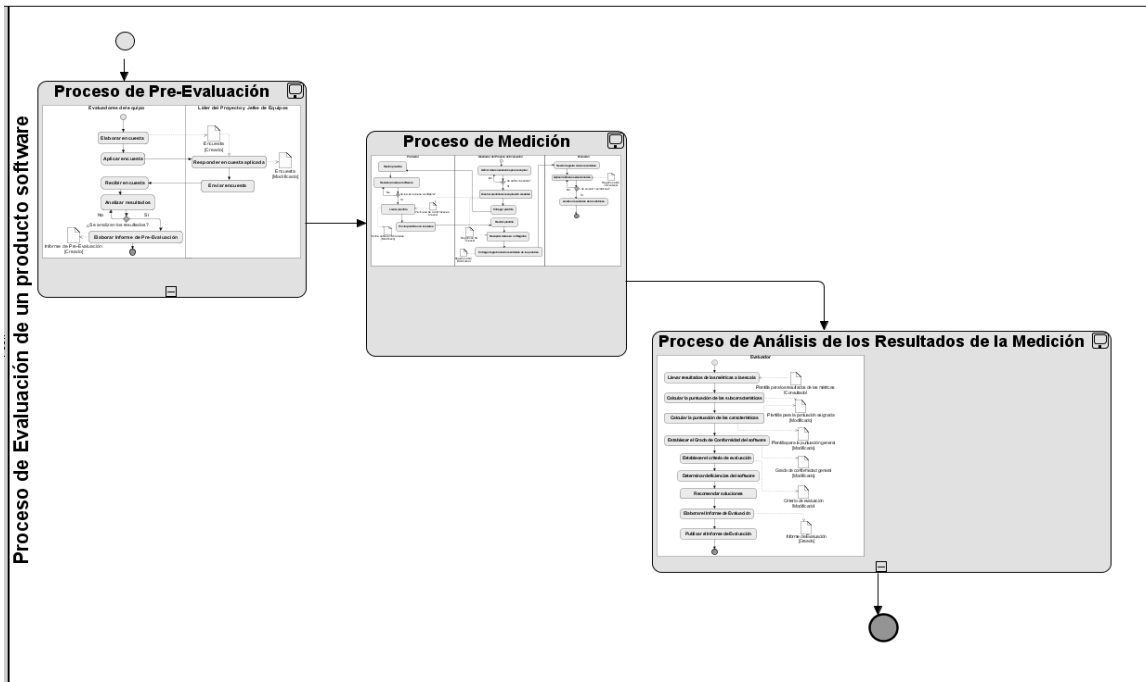


Figura 4: Proceso de Evaluación del producto software

### 2.1.1 Descripción del Proceso de Pre-Evaluación del software

Para dar comienzo al Proceso de Pre-Evaluación del producto software en un proyecto, se le debe aplicar una encuesta (Ver Anexo 1) a los directivos del mismo, así como a los Jefes de Roles, demás expertos involucrados con el proyecto y de ser posible al cliente, con el objetivo de que estos den su criterio acerca de cuáles deben ser las características de calidad de software según la Norma ISO/IEC 9126-1 que se deben medir en el producto.

Una vez que se aplique la encuesta se analizan los resultados obtenidos, definiendo cuáles son las características que presentan mayor peso, y en base a esto enfocar todos los esfuerzos para monitorear el cumplimiento de las mismas. Para tomar esta decisión el evaluador se puede apoyar en el estudio de sistemas similares al que se pretende evaluar, se debe consultar el documento de Especificaciones Suplementarias y las especificaciones de los CU del sistema. Posteriormente se realiza un Informe de Pre-Evaluación y se procede a la selección de las métricas que se aplicarán para la evaluación de la calidad del producto software. A modo de resumen en la siguiente figura se representa el proceso de Pre-Evaluación del producto software. (Ver figura 5)

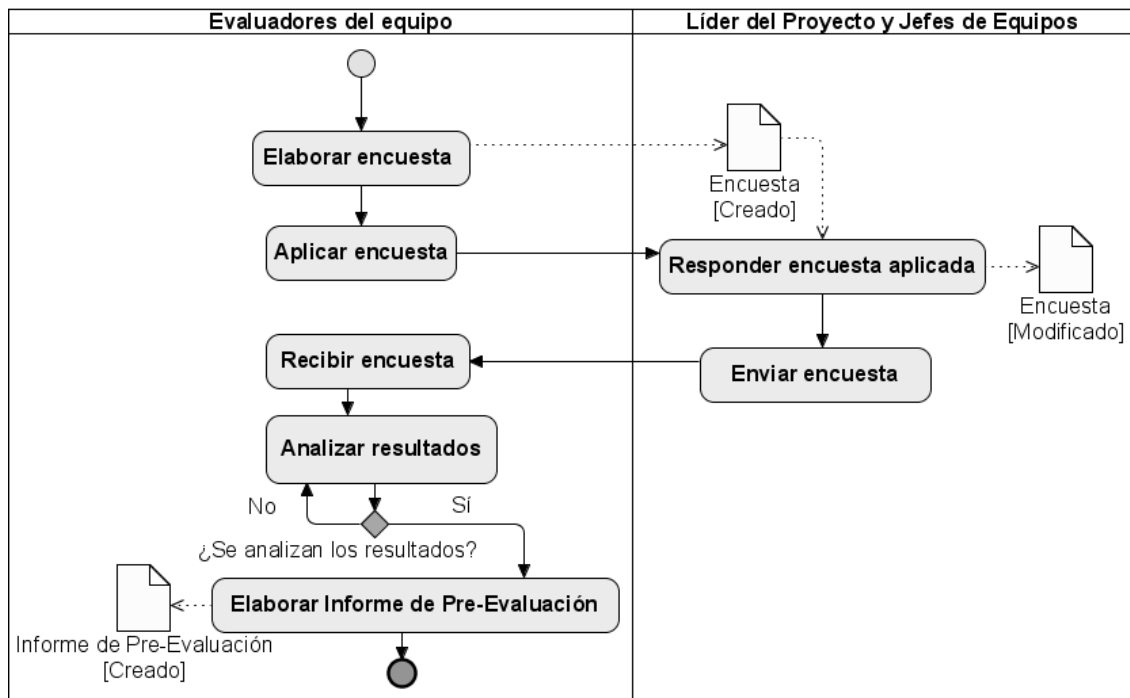


Figura 5: Proceso de Pre-Evaluación del producto software

### 2.1.2 Descripción del Proceso de Medición

En el capítulo anterior se abordó brevemente acerca del proceso de medición, que es un proceso para la evaluación de la calidad del producto software y que ayuda además a obtener una visión tanto cualitativa como cuantitativa de la calidad del mismo, permitiendo tomar decisiones para la mejora de su proceso de desarrollo.

En este proceso de medición (Ver figura 6) se deben involucrar de forma activa las personas del equipo de calidad designadas a encargarse de la evaluación de la calidad del producto software. El proceso consta de tres partes fundamentales, la recopilación de datos una vez que se realicen las pruebas al software, luego se procede al cálculo de las métricas seleccionadas para una posterior evaluación de la calidad del producto software.

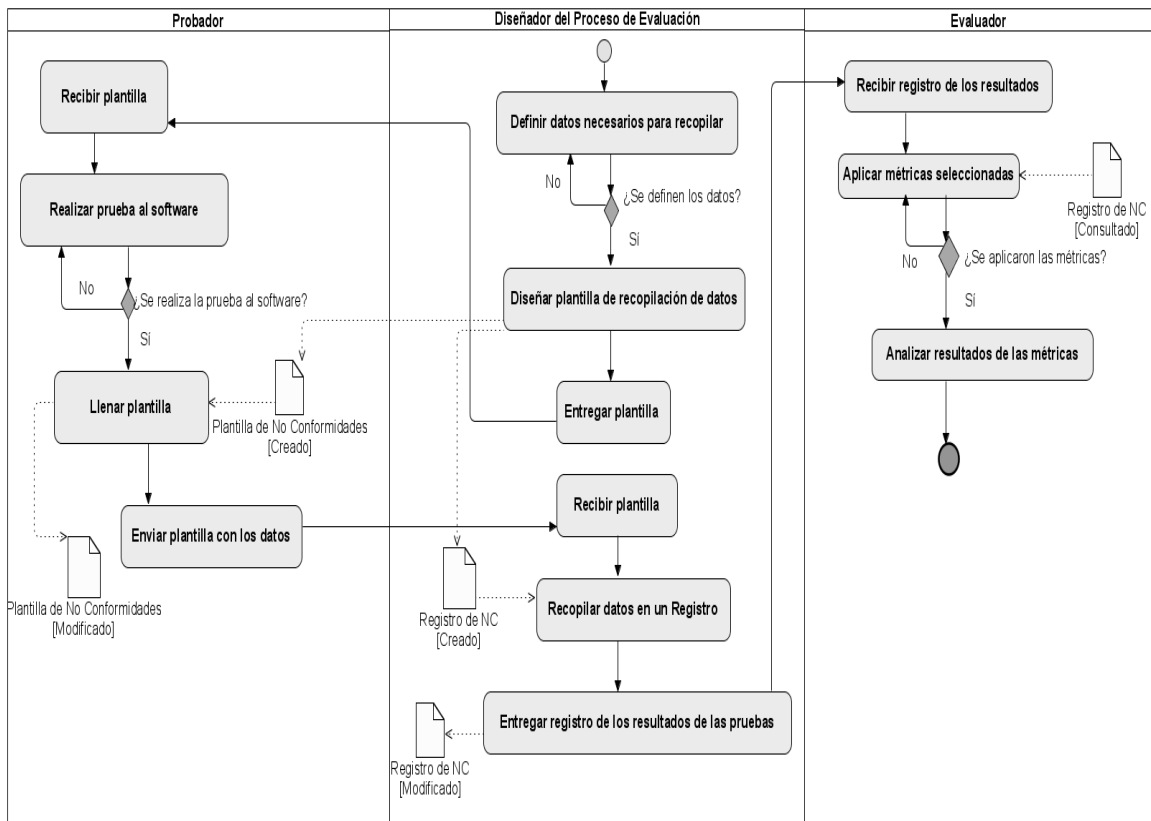


Figura 6: Proceso de Medición del producto software

Durante todo este proceso se generan diferentes artefactos los cuales son necesarios para el desarrollo del proceso de evaluación. Artefactos donde se almacena toda la información real generada durante el proceso de pruebas, de estos artefactos se abordará en un posterior epígrafe, explicando en qué consiste cada uno de ellos y su propósito en el proceso de evaluación.

### 2.1.2.1 Descripción del Proceso de Recopilación de datos

Para el desarrollo del proceso de medición es necesario realizar la recopilación de datos durante el desarrollo del proceso de pruebas. (Ver figura 7) Estos datos se deberán recopilar a medida que se vayan realizando las pruebas al software. El documento en el cual estarán estos datos llevará por nombre *Registro de las No Conformidades (NC) generadas durante las pruebas al software*, del cual se abordará en un posterior acápite.

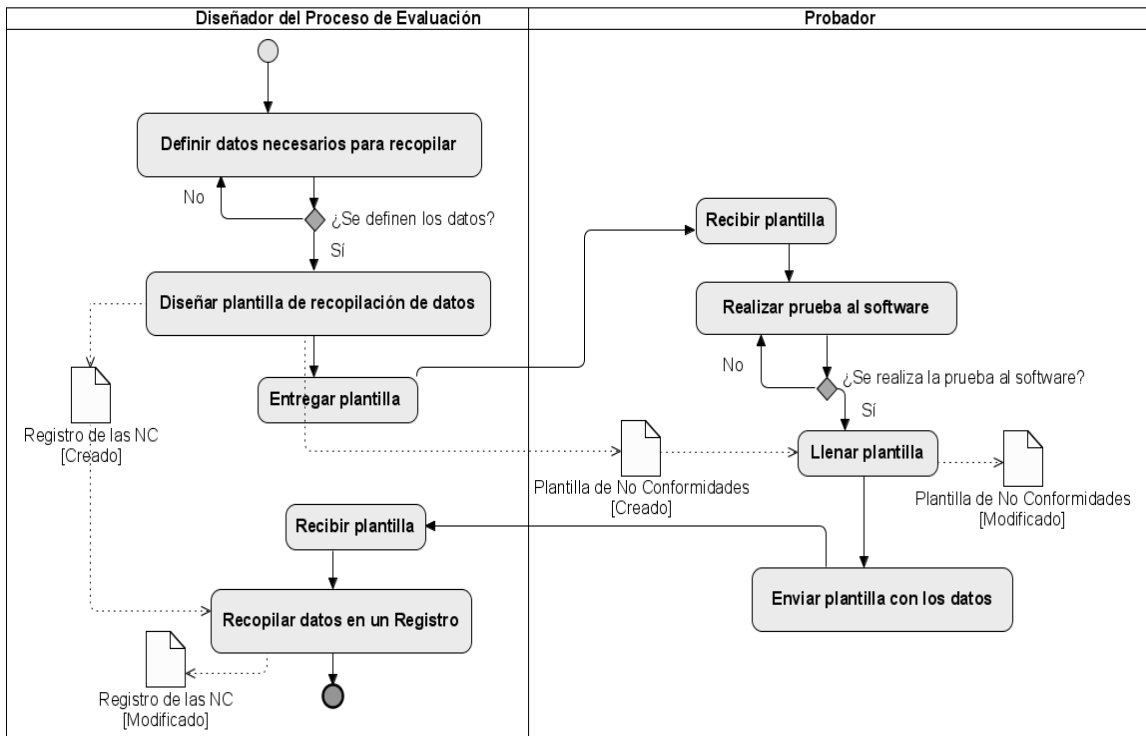


Figura 7: Proceso de Recopilación de Datos

Como se puede ver en la figura, para llevar a cabo el proceso de recopilación de datos, primero se deben definir qué datos son los necesarios para medir, se elabora un plantilla que recoja todos estos datos y se le entrega esta plantilla al probador el cual anota estos datos a medida que vaya realizando la prueba al software. Y cuando se tengan todos los datos recopilados se almacenan en el registro anteriormente mencionado.

### 2.1.2.2 Propuesta de las métricas a utilizar

En el presente sub-epígrafe se presentará una propuesta de las métricas a aplicar para evaluar la calidad del producto software. La Norma ISO/IEC 9126-3 de Calidad Interna propone métricas para evaluar la calidad de un producto software.

Las métricas que se mostrarán a continuación han sido traducidas y adaptadas a la medición durante el proceso de pruebas internas y podrán ser utilizadas en procesos similares en otros proyectos productivos.[27]

### 2.1.2.2.1 Métricas de Funcionalidad

Tabla 1: Tabla de Métricas de Funcionalidad

<b>MÉTRICAS DE FUNCIONALIDAD</b>				
<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Subcaracterística: Idoneidad</b>				
<b>Adecuación funcional</b>	¿Cuán adecuadas son las funciones verificadas?	Contar la cantidad de funciones implementadas que son fáciles de entender para completar la tarea especificada, y luego se divide entre la cantidad total de funciones implementadas.  Pueden ser medidas en:  Todas o parte de las especificaciones de diseño.  Módulos completados / componentes del producto de SW.	<b><math>X = I-A/B</math></b>  A: Número de funciones en las cuales se detectaron problemas en la evaluación.  B: Número de funciones evaluadas.	<b><math>0 \leq X \leq 1</math></b>  A mayor cercanía al 1 resultará más adecuada.
<b>Subcaracterística: Exactitud</b>				
<b>Exactitud</b>	¿Cuán completa estuvo la implementación de los niveles de precisión específicos para los elementos de datos?	Contar la cantidad de elementos de datos que tienen niveles específicos de precisión y compararlo con el total de	<b><math>X = A/B</math></b>  A: Número de datos implementados con los niveles de precisión	<b><math>0 \leq X \leq 1</math></b>  A mayor cercanía al 1 resultará más adecuada.

		<i>elementos de datos que lo requerían según la especificación.</i>	<i>especificados, confirmado en la evaluación.</i>  <i>B: Número de datos que requirieron los niveles de precisión.</i>	
<b><i>Subcaracterística: Interoperabilidad</i></b>				
Esta Subcaracterística no se medirá porque aún no se ha probado el intercambio de datos del SIIPOL con otros sistemas.				
<b><i>Subcaracterística: Seguridad</i></b>				
<b><i>Controlabilidad de acceso</i></b>	<i>¿Cuán controlable es el acceso al sistema?</i>	<i>Contar la cantidad de requerimientos de control de acceso que fueron implementados correctamente y compararlo con la cantidad de requerimientos de control de acceso descritos en la especificación.</i>	<b><i>X = 1 - A/B</i></b>  <i>A: Número de CU que presentan problemas con el control de acceso.</i>  <i>B: Número total de CU probados.</i>	<b><i>0 &lt;= X &lt;= 1</i></b>  <i>A mayor cercanía al 1 resultará más controlable.</i>

### 2.1.2.2 Métricas de Confiabilidad

Tabla 2: Tabla de Métricas de Confiabilidad

<b>MÉTRICAS DE CONFIABILIDAD</b>				
<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Subcaracterística: Madurez</b>				
<b>Adecuación de pruebas</b>	¿Cuántos casos de pruebas requeridos están cubiertos por el plan de pruebas?	Contar el número de casos de pruebas planificados y comparar con el número de casos de pruebas requeridos para obtener una adecuada cobertura de pruebas.	$X = A/B$ A: Número de casos de prueba diseñados en el plan de pruebas y confirmados en la revisión B: Número de casos de prueba requeridos.	$0 \leq X$ Donde X sea más grande será más adecuado.
<p>Para calcular el número de Casos de pruebas requeridos se realiza una estimación de los escenarios que debe tener un CU, aplicando la siguiente métrica:</p> <p><b>B= Cantidad de flujos alternos +Cantidad de Secciones +1</b></p>				
<b>Subcaracterística: Tolerancia ante fallos</b>				
<b>Evitación de fallos</b>	¿Cuántos patrones de fallos fueron traídos bajo control para evitar fallas críticas y serias?	Contar el número de patrones de fallos evitados y comparar con el número de patrones de fallos a ser considerados.	$X = A/B$ A: Número de fallos evitados. B: Número de fallos totales.	$0 \leq X$ Donde X sea más grande es mayor la evitación de fallos.
<b>Subcaracterística: Recuperabilidad</b>				
<b>Recuperabilidad</b>	¿Cuán capaz es el producto en	Contar el número de	$X = A/B$	$0 \leq X \leq 1$



	restaurarse después de eventos o peticiones anormales?	requisitos de restauración implementados y comparados con el número de requisitos de restauración descritos en la especificación.	A: Número de requisitos de recuperación implementados confirmados en la revisión.  B: Número de requisitos de recuperación de la especificación.	Mientras mayor sea X mejor será la recuperabilidad.
--	--	---	--	---

### 2.1.2.2.3 Métricas de Usabilidad

Tabla 3: Tabla de Métricas de Usabilidad

<b>MÉTRICAS DE USABILIDAD</b>				
<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Subcaracterística: Comprensibilidad</b>				
<b>Entendimiento de funciones</b>	¿Qué proporción de las funciones del producto será el usuario capaz de comprender correctamente?	Contar el número de interfaces de usuario donde su propósito es el entendimiento del usuario y compararlo con el número de funciones de interfaces de usuarios.	$X = A/B$  A: Número de funciones de interfaz de usuario cuyo propósito es entendido por el usuario.  B: Número total de funciones de interfaz de usuario.	$0 \leq X \leq 1$  Mientras más cercano a 1, mejor.
<b>Subcaracterística: Operabilidad</b>				
<b>Chequeo de la Validez de la</b>	¿Qué proporción de	Contar el número de	$X = A/B$	$0 \leq X \leq 1$

<i>entrada</i>	<i>número de elementos de entrada chequean validez de los datos?</i>	<i>elementos de entrada, que chequean la validez de los datos y compara con el número de elementos de entrada, que podría verificar para datos válidos.</i>	<i>A: Número de elementos de entrada que chequean validez de los datos. B: Número de elementos de entrada que pueden chequear la validez de los datos.</i>	<i>Mientras más cercano a 1, mejor.</i>
----------------	--	---	--	---

#### 2.1.2.2.4 Métricas de Eficiencia

Tabla 4: Tabla de Métricas de Eficiencia

<b>MÉTRICAS DE EFICIENCIA</b>				
<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Subcaracterística: Rendimiento</b>				
<b>Rendimiento</b>	<i>¿Cuál es el número estimado de tareas que pueden ser realizadas sobre una unidad de tiempo?</i>	<i>Evaluar la eficiencia de manejar recursos en el sistema. Haga un factor basado sobre las llamadas del uso al sistema en la manipulación de los recursos.</i>	<i>X = Número de tareas por unidad de tiempo</i>	<i>Mientras mayor mejor.</i>
<b>Subcaracterística: Utilización de recursos</b>				
<b>Uso de memoria</b>	<i>¿Cuál es el tamaño de memoria estimado que el</i>	<i>Estimar los requerimientos de memoria.</i>	<i>X = Tamaño en bytes (calculado</i>	<i>Mientras menor sea, mejor.</i>

	<i>producto ocupará para completar una tarea específica?</i>		<i>o simulado).</i>	
--	--	--	---------------------	--

### 2.1.3 Descripción del Proceso de Análisis de los resultados de la medición

A continuación se muestra cómo debe realizarse el Proceso de Análisis de los resultados de la medición, explicando posteriormente cada una de las actividades que conforman este proceso, y con cada una de las plantillas que se generan del mismo. (Ver figura 8)

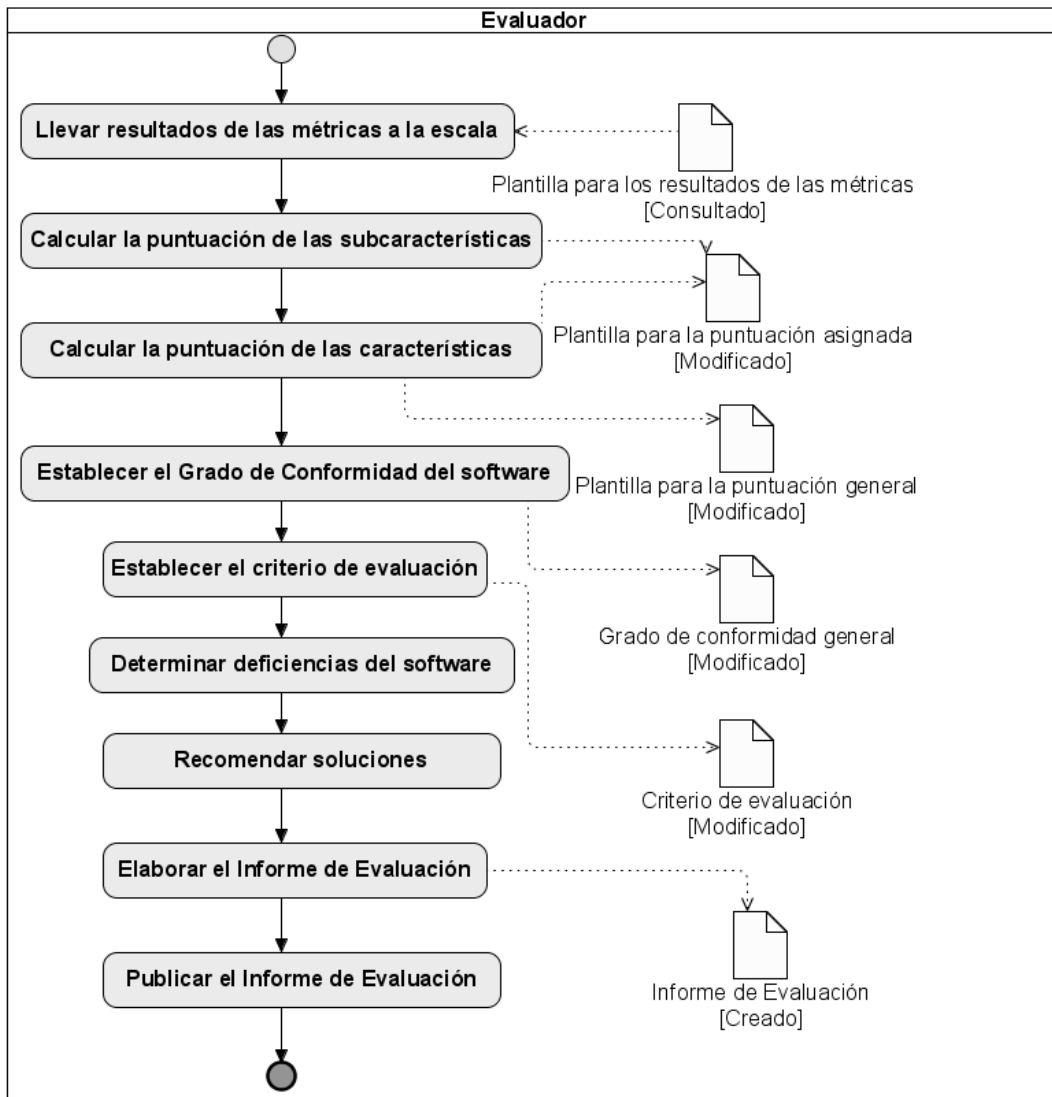


Figura 8: Proceso de Análisis de los resultados de la medición

Una vez que se apliquen las métricas y se obtengan los resultados de cada una de ellas, se recopila en la siguiente tabla una breve descripción numérica de cada una de ellas. (Ver tabla 5)

Tabla 5: Plantilla para los resultados de las métricas aplicadas en la evaluación

Característica	Subcaracterística	Peso	Métrica	Nivel Requerido	Resultado de la métrica	Resultado Real
<b>Funcionalidad</b>	Idoneidad					
	Exactitud					
	Seguridad					
<b>Confiabilidad</b>	Madurez					
	Tolerancia ante fallos					
	Recuperabilidad					
<b>Usabilidad</b>	Comprensibilidad					
	Operabilidad					
<b>Eficiencia</b>	Rendimiento					
	Utilización de recursos					

Los resultados de cada una de las métricas usadas se llevan a una escala con valores entre 0 y 1 (Ver tablas 6 y 7), y en dependencia del resultado real que se recoge en la tabla anterior se evalúan de la siguiente forma:

#### Escala para cuando se acerca a 1:

Tabla 6: Escala para cuando se acerca a 1

<b>0 - 0,4</b>	<b>Mal</b>	<b>0</b>
<b>0,4 - 0,7</b>	<b>Regular</b>	<b>1</b>
<b>0,7 - 0,9</b>	<b>Bien</b>	<b>2</b>
<b>0,9 - 1</b>	<b>Muy Bien</b>	<b>3</b>

### Escala para cuando se acerca a 0:

Tabla 7: Escala para cuando se acerca a 0

<b>0 - 0,1</b>	<b>Muy bien</b>	<b>3</b>
<b>0,1 – 0,3</b>	<b>Bien</b>	<b>2</b>
<b>0,3 – 0,6</b>	<b>Regular</b>	<b>1</b>
<b>0,6 – 1</b>	<b>Mal</b>	<b>0</b>

Como se puede observar las escalas están concebidas en dependencia del resultado real de la métrica, si el resultado es mejor cuando se acerca a 1 entonces se usa la primera escala y si el resultado es mejor cuando se acerca a 0, entonces se escoge la segunda escala. Estos resultados se recopilan en la siguiente tabla: (Ver tabla 8)

Tabla 8: Plantilla para la puntuación asignada a las características y subcaracterísticas de calidad

<b>Características y Subcaracterísticas de Calidad</b>	<b>Puntuación</b>			
	3	2	1	0
<b>1. Funcionalidad</b>				
<b>1.1 Idoneidad</b>				
<b>1.2 Exactitud</b>				
<b>1.3 Seguridad</b>				
<b>2. Confiabilidad</b>				
<b>2.1 Madurez</b>				
<b>2.2 Tolerancia ante fallos</b>				
<b>2.3 Recuperabilidad</b>				
<b>3. Usabilidad</b>				
<b>3.1 Comprensibilidad</b>				
<b>3.2 Operabilidad</b>				
<b>4. Eficiencia</b>				
<b>4.1 Rendimiento</b>				
<b>4.2 Utilización de recursos</b>				

La tabla anterior se completa aplicando las escalas especificadas en las tablas 7 y 8 por cada una de las subcaracterísticas. Estos valores de las subcaracterísticas se sumarán y se promediarán en caso que el peso sea el mismo para todas las subcaracterísticas de una determinada característica dándole la puntuación final a la misma.

En el caso en que exista una característica en la cual sus subcaracterísticas presenten peso distinto, se seguirá el siguiente procedimiento:

- Primero se debe tener en cuenta la cantidad de subcaracterísticas que se van a medir y el peso de cada una de estas.
- Se le asignará un valor entre 0 y 1 a cada uno de los pesos Alto, Medio o Bajo. El peso Alto es el que mayor valor debe tomar, cumpliéndose así con el peso Medio con respecto al Bajo.
- El valor que se le asigna a cada uno de los pesos Alto, Medio o Bajo, se debe multiplicar por la cantidad de subcaracterísticas que tengan igual peso respectivamente. La suma de cada uno de los resultados después de haber multiplicado, debe ser 1.
- Se multiplica el valor asignado al peso por la puntuación de la subcaracterística que se definió después de aplicada la métrica y llevada a una escala.
- Y finalmente se suman los resultados de cada una de las subcaracterísticas y este será la puntuación final de la característica.

Para comprender mejor el procedimiento anteriormente presentado se muestra el siguiente ejemplo:

Se tiene la característica **Eficiencia**, y de ella se medirán las subcaracterísticas **Rendimiento** con un peso Alto y **Utilización de recursos** con un peso Medio.

Como son dos pesos diferentes se le asigna un valor a cada uno de manera que la suma de los pesos sea 1:

Alto: 0.6

Medio: 0.4.

Estos valores de cada uno de los pesos se multiplican por la puntuación de la subcaracterística.

Rendimiento: 2

Utilización de recursos: 1

Después se multiplican estos valores respectivamente:

$$0.6 * 2 = 1.2$$

$$0.4 * 1 = 0.4.$$

Entonces estos valores que se obtienen se suman  $1.2 + 0.4 = 1.6$ , y el resultado se aproxima a 2 para asignarle la puntuación a la característica **Eficiencia**.

Los resultados de cada una de las características se recogen en la siguiente tabla (Ver tabla 9) y se suman las puntuaciones, obteniendo un valor final al cual se le aplica la escala que representa el grado de conformidad en el software. (Ver tabla 10)

Tabla 9: Plantilla para la puntuación general de cada una de las características

<b>Puntuación</b>	
<b>Funcionalidad</b>	
<b>Confiabilidad</b>	
<b>Usabilidad</b>	
<b>Eficiencia</b>	
<b>Valor Final</b>	

Se debe tener en cuenta el peso que se le asignó a cada una de las características. En el caso en que todas tengan el mismo peso, después de sumados los valores se promedia y se determina el grado de conformidad y junto a él se dará el criterio de evaluación del software.

En el caso en que los pesos de las Características difieran entonces se tomará en cuenta el mismo procedimiento que se mencionó anteriormente. Esto se hace con el objetivo de que el peso de las características esté balanceado y se corresponda para realizar una justa evaluación del software. Después de aplicado este mecanismo, se obtiene el Grado de Conformidad.

Tabla 10: Grado de conformidad general

<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>

Cuando se tenga definido el grado de conformidad y se hayan detectado las deficiencias en el software se dará un Criterio de Evaluación para el software, según lo muestra la siguiente tabla. (Ver Tabla 11)

Tabla 11: Criterio de evaluación

<b>Criterio de Evaluación</b>
( ) Sin modificaciones
( ) Pequeñas modificaciones
( ) Grandes modificaciones
( ) Nueva elaboración

## **2.2 Artefactos generados durante el proceso de evaluación**

Durante el proceso de evaluación se generan una serie de artefactos los cuales brindan información tanto cualitativa como cuantitativa del proceso de pruebas. Los mismos se mencionan a continuación:

- ❖ Registro de las NC generadas durante las pruebas al software.
- ❖ Informe de los resultados de las pruebas al software.
- ❖ Informe de Pre-Evaluación del producto software.
- ❖ Informe de Evaluación del producto software.

Cada uno de estos artefactos tiene como objetivo brindar a todas las personas del proyecto la información de cómo se ha estado trabajando, cuáles fueron las principales deficiencias detectadas en el software, los equipos de desarrollo que presentaron más problemas en la implementación del mismo, así como los equipos en los que hubo mejor organización, entre otras informaciones de interés.

A continuación se dará a conocer en qué consiste cada uno de los artefactos que se deberán generar durante todo el proceso de pruebas. Es válido aclarar que estos artefactos deben estar publicados en un repositorio de información del proyecto con el propósito de que cualquier persona interesada que tenga permiso y acceso al mismo pueda consultar la información que desee.

### **2.2.1 Registro de las NC generadas durante las pruebas al software**

Este registro presentará toda la información de manera organizada y comprensible que se genera del proceso de pruebas. La información deberá estar organizada por



Subsistemas o Módulos, y cada uno de estos contendrá los nombres de los CU que a ellos pertenecen.

A continuación se mencionan los principales datos que se deben almacenar en el registro:

- ❖ Nombre del Subsistema o Módulo
- ❖ Nombre del CU
- ❖ Fecha en que se realizó la prueba al CU
- ❖ Nombre de la persona que realizó la prueba
- ❖ Cantidad de NC que son significativas
- ❖ Cantidad de NC que no son significativas
- ❖ Cantidad de recomendaciones del probador
- ❖ Clasificación de la NC

Las NC se pueden clasificar en uno de los siguientes tipos:

- ❖ NC de ortografía
- ❖ NC de validación
- ❖ NC de funcionalidad sin implementar
- ❖ NC de rendimiento
- ❖ NC de funcionalidad
- ❖ NC de seguridad
- ❖ NC de ausencia de campos
- ❖ NC de cualquier otro tipo

Una vez que se recopilan estos datos se debe pasar a la parte estadística de este proceso, calculando por Subsistemas o Módulos según la cantidad de NC detectadas durante las pruebas. De aquí se pueden obtener valores cuantitativos en cuanto a:

- ❖ Cantidad de NC por Subsistemas o Módulos
- ❖ Cantidad de NC que no proceden por Subsistemas o Módulos
- ❖ Cantidad de NC significativas por Subsistemas o Módulos
- ❖ Cantidad de NC no significativas por Subsistemas o Módulos
- ❖ Cantidad de recomendaciones por Subsistemas o Módulos
- ❖ Cantidad de CU que no presentaron NC
- ❖ Cantidad de NC detectadas según su tipo de clasificación por Subsistemas o Módulos

De forma general esta información permite valorar cómo se ha estado trabajando en el proyecto y si existe una mejoría entre las diferentes etapas e iteraciones de las pruebas. Este registro de NC es la base para la elaboración del *Informe de los resultados de las pruebas al software*, que presentará toda la información generada durante el proceso de pruebas al producto software en sus diferentes etapas.

### **2.2.2 Informe de los resultados de las pruebas al software**

Este informe presentará toda la información del proceso de pruebas al software en cada una de sus etapas e iteraciones. Para reflejar con mayor nivel de detalle la información que se muestra, los resultados serán mostrados a través de gráficas y tablas que ilustrarán las tipologías de errores, los totales de NC detectadas por Subsistemas o Módulo, la cantidad de CU a los que no se les encontró NC, la cantidad de NC que No Proceden de cada CU y por Subsistemas o Módulo, además de la cantidad de NC según su complejidad por cada CU y por Subsistemas o Módulo.

Con este informe se pretende generar información a los miembros del equipo de desarrollo para la toma de decisiones inmediatas, así como a cualquier interesado en conocer sobre este tema, además de algunas de las acciones correctivas que deberán ser ejecutadas en el proyecto para mitigar las principales causas de error.

### **2.2.3 Informe de Pre-Evaluación del producto software**

El Informe de Pre-Evaluación como su nombre lo indica presentará una pre-evaluación cualitativa del software. En el mismo se presentará los requisitos que el software deberá cumplir en dependencia de las características de calidad que establece la NC ISO/IEC 9126-1.

Para la elaboración de este informe es necesario encuestar a los dirigentes del proyecto para que estos asignen un peso estimado a cada una de las características y subcaracterísticas de calidad que crean que deben cumplirse en el software a evaluar. Estos pesos pueden ser Alto, Medio o Bajo.

Además del criterio de cada una de las personas encuestadas se debe tener en cuenta el estudio de sistemas similares, de las especificaciones de los CU y del documento de Especificaciones Suplementarias.

## 2.2.4 Informe de Evaluación del producto software

Este Informe muestra el resultado de la evaluación del producto software durante el proceso de pruebas, este resultado contiene el grado de conformidad y el criterio de evaluación que se obtuvo. En él se encontrarán enumeradas las principales deficiencias que se encontraron en el software y su recomendación para solucionar las mismas, evitando que vuelvan a cometerse los mismos errores en etapas posteriores del ciclo de vida del software y durante el proceso de pruebas.

## 2.3 Roles y responsabilidades en el proceso de evaluación

Los roles propuestos para realizar el proceso de evaluación que se pretende realizar en el proyecto Modernización del CICPC, pensando que es la forma más adecuada de organizar el equipo de evaluadores, son los siguientes:

- ❖ *Asesor del equipo de Calidad:* Responsable del proceso de evaluación al producto software y la persona más capacitada en el tema de evaluación de la calidad de software.
- ❖ *Diseñador del proceso de evaluación:* Responsable de diseñar las plantillas para la recopilación de información, datos, entre otros. Responsable además de elaborar los diferentes informes que se generan durante todo el proceso de evaluación.
- ❖ *Evaluador:* Persona con sólidos conocimientos de evaluación de software y métricas de software. Estos son los encargados de ejecutar las actividades de los procesos de evaluación y medición.
- ❖ *Probador:* Persona que realizará pruebas al producto software.

## 2.4 Conclusiones del capítulo

En este capítulo se describieron cada uno de los subprocesos que se ejecutan dentro del Proceso de evaluación en general y se propusieron las métricas para realizar la medición de la calidad del producto software. Abordó de los principales roles y responsabilidades dentro del proceso de evaluación, además de los artefactos que se deben generar dentro de este proceso. Y finalmente se presentó el procedimiento para realizar la evaluación de la conformidad del producto software teniendo en cuenta el informe de pre-evaluación y los resultados de las pruebas.

## CAPÍTULO

**EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE SIIPOL**

En el presente capítulo se pondrá en práctica la propuesta de evaluación del producto software. Se mostrarán los resultados arrojados del Proceso de Pre-evaluación y del Proceso de Medición, así como el análisis posterior de los mismos, el cual indicará el Grado de Conformidad y el Criterio de Evaluación del producto software SIIPOL. Se expondrán además las principales deficiencias detectadas y las posibles acciones correctivas a tener en cuenta para etapas posteriores.

### 3.1 Resultados de la Encuesta para la Pre-Evaluación del producto software SIIPOL.

Después de aplicada la encuesta a los Jefes de Subsistemas o Módulos en el proyecto Modernización del CICPC, se determinó que los resultados finales de la encuesta son los siguientes. (Ver tabla 12)

Tabla 12: Resultado de la encuesta de pre-evaluación

<b>Característica</b>	<b>Subcaracterística</b>	<b>Pesos (Alto / Medio / Bajo )</b>
<b>Funcionalidad</b>	Idoneidad	Alto
	Exactitud	Alto
	Seguridad	Alto
<b>Confiabilidad</b>	Madurez	Alto
	Tolerancia ante fallos	Alto
	Recuperabilidad	Alto
<b>Usabilidad</b>	Comprensibilidad	Medio
	Operabilidad	Medio
<b>Eficiencia</b>	Rendimiento	Alto
	Utilización de recursos	Medio

Como se puede percibir en el resultado de la encuesta, la mayoría de las subcaracterísticas de calidad tienen peso Alto y pocas peso Medio, esto demuestra la importancia de que el software cumpla con las características y subcaracterísticas de calidad y la necesidad de medir la calidad del producto.

Para fundamentar la Pre-evaluación del SIIPOL se realizó un estudio del Documento de Especificaciones Suplementarias del Proyecto y las Especificaciones de los CU del sistema. A continuación se presenta la justificación del grado de presencia en el SIIPOL de las características y subcaracterísticas propuestas por la Norma ISO/IEC 9126-1.

## **Funcionalidad**

### **❖ Idoneidad**

El peso de esta Subcaracterística es **Alto** ya que el sistema:

- Soportará un alto nivel de parametrización para que siempre que sea posible se adapte al tipo de información que se está recabando o intentando obtener.
- Permitirá generar reportes estándares y parametrizables (con uso de asistentes en caso que lo requiera) uso de formatos en: tablas, gráficos, estadísticas, análisis comparativos, etc.
- Permitirá la importación/exportación de información. El formato para la exportación debe ser .pdf o algún otro que ayude a mantener la integridad de la información.
- Soportará la manipulación del correo electrónico para intercambio de información.

### **❖ Exactitud**

El peso de esta Subcaracterística es **Alto** ya que:

- Todos los errores del sistema serán registrados. Los errores fatales del sistema siempre resultarán en un ordenado apagado de la PC.
- El mensaje de error del sistema debe incluir una descripción textual del error.
- Todos los errores del sistema deberán ser registrados en la base de datos de registro de errores.
- Gestión de errores.
- Registrar todos los errores en almacenamiento persistente.

## ❖ Seguridad

El peso de esta Subcaracterística es **Alto** ya que:

- Todo uso requiere la autenticación de los usuarios.
- El sistema manejará la seguridad de acceso y administración de usuarios: otorgamiento de privilegios y roles, asignación de perfiles.
- El sistema permitirá el acceso a la aplicación solo a terminales que pertenezcan al CICPC, previa validación de un administrador.
- El sistema concederá acceso al sistema a partir de una credencial y una contraseña e incluso por identificación de huellas dactilares para algunas funciones claves.
- El sistema concederá acceso solo a funciones que sean permitidas de acuerdo al perfil del usuario que fue logeado. (Menús generados a partir del perfil).
- El sistema solicitará los permisos de acceso cada determinado tiempo que se detecte inactividad por parte del usuario.
- El sistema manejará mecanismos de encriptación de algunas informaciones que sea transmitida por la red.
- El sistema implementará el uso de campos obligatorios y validaciones para garantizar la integridad de la información que se introduce.
- El sistema implementará mecanismos de recuperación de validaciones, cuando en el momento de la inclusión no exista comunicación en línea con los entes que validan (ej. Validación de la cédula en el registro de identidad, o serial de vehículos en INTTT).
- El sistema permitirá la clasificación de la información en cuanto a importancia y nivel de confidencialidad.
- El sistema implementará un mecanismo de auditoría para el registro de todos los accesos al efectuado por los usuarios; proporcionando un registro de actividades (Log) de cada usuario en el sistema.
- El sistema soportará el uso de firmas digitales para la transferencia de información cuya certificación sea imprescindible para validar el uso de la misma.
- El sistema implementará un control de cambios a determinados campos de forma tal que sea posible determinar cuáles han sido las actualizaciones que se le han realizado.

- Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la BD, independientemente de que para el sistema, este elemento ya no exista.
- El sistema aplicará un mecanismo de respaldo y recuperación de base de datos: Realizar copias de seguridad de la base de datos hacia otro dispositivo de almacenamiento externo. Permitir la recuperación de la base de datos a partir de los respaldos realizados.
  - Requerimientos de administración
- El sistema permitirá la creación de perfiles de usuario y asignación de funcionarios a un determinado perfil de usuario.
- El sistema permitirá la creación y actualización de tablas de códigos que soportarán las funciones para los procesos de negocio. (empresas, tipo delito, marcas, series, dependencias, cargos, etc.).
- El sistema permitirá el monitoreo en línea: funcionalidad ubicada en la sede central que permitirá el monitoreo de las funciones que se están haciendo en ese momento sobre el sistema, y el usuario que las está realizando.

## **Confiabilidad**

### ❖ **Madurez**

El peso de esta Subcaracterística es **Alto** ya que:

- El sistema debe permitir un elevado volumen de almacenamiento, permitiendo la escalabilidad de dicho volumen de manera más fácil, económica y en menos tiempo. Para ello, los datos serán almacenados en una cabina HP de última generación EVA 8000.
- El sistema usará igualmente la potencialidad del EVA 8000 para realizar balanceo de carga entre los servidores de aplicación permitiendo acortar los tiempos de respuesta.
- Tiempo medio entre fallos (Mean Time Between Failures MTBF) esto usualmente es especificado en horas, pero también puede ser especificado en términos de días, meses o años.

### ❖ **Tolerancia ante fallos**

El peso de esta Subcaracterística es **Alto** ya que:

- El sistema debe brindar una manera óptima de recuperación ante fallos en el centro de datos en tiempo de ejecución de manera transparente al usuario sin necesidad de parar la aplicación. Para ello se potenciará el uso

de la configuración de los discos en espejo (RAID 1) para la protección de los datos, la cual es una de las mejores soluciones disponibles actualmente en el mercado.

#### ❖ **Recuperabilidad**

El peso de esta Subcaracterística es **Alto** ya que:

- El sistema tendrá un respaldo de la información del centro de datos mediante un sistema de backups, permitiendo la recuperación ante la pérdida parcial o total de la información. Para ello se potenciará el uso de las mejores tecnologías disponibles en el mercado actualmente para tal propósito como es la funcionalidad de RMAN de Oracle y el HP Data Protector.
- El sistema debe ser capaz de hacer copias de respaldo de la base de datos a intervalos de tiempo fijos o a voluntad del usuario administrador.

#### **Usabilidad**

##### ❖ **Comprensibilidad**

El peso de esta Subcaracterística es **Medio** ya que en la Interfaz de usuario:

- El sistema brindará una interfaz amigable para usuarios finales. Su funcionamiento deberá ser intuitivo, y requerir de información mínima.
- El sistema proporcionará claridad y buena organización de la información, permitiendo la interpretación correcta e inequívoca de esta.
- El sistema implementará la ejecución de acciones de una manera rápida, minimizando los pasos a dar en cada proceso.
- El sistema priorizará el uso de elementos visuales para la selección de información siempre que sea posible para minimizar los riesgos de equivocaciones.
- Todos los textos y mensajes en pantalla aparecerán en idioma español. Los errores serán visibles al usuario e incluirán sugerencias de las posibles soluciones.
  - Ubicación, guías y ayudas al usuario:
- Se presentará en todas las páginas un Mapa del Sitio y este debe tener hipervínculos a las diferentes secciones que muestra.
- Se accederá desde todas las vistas a la ayuda general.
- Se mostrarán las rutas de acceso según la navegación que tenga el usuario.



- Las rutas de acceso tendrán vínculos a las secciones que muestran.
- El sistema mostrará el nombre del usuario que está trabajando en el sistema.
- El sistema mostrará el menú general desde cualquiera de sus páginas.
- Se deberá definir un color único para los enlaces visitados en una misma sección de trabajo.
- Subrayar los enlaces textuales y dado el caso colocarlos en azul.
- Se deberán usar etiquetas alternativas para todas las imágenes descriptivas (fotos, documentos escaneados, etc.).
- Ubicar en cada página el Pie de Página.
- El Pie de Página mostrará elementos como los derechos de autor, datos de ubicación de la institución, nombre de la institución, Año en curso y un menú abreviado de navegación.

#### ❖ **Operabilidad**

El peso de esta Subcaracterística es **Medio** ya que:

##### ➤ Acceso:

- El sistema mostrará un texto de bienvenida al acceder a su página de entrada.
- El sistema mostrará en cada página la opción de insertar el usuario y la contraseña.
- El sistema mostrará de una forma distintiva mensajes para cada eventualidad que ocurra con la entrada al sistema.
- El sistema mostrará textos personalizados para los mensajes de error.
- El sistema informará quien es el usuario correspondiente con la sección que esté activa.

##### ➤ Navegadores:

- El sistema funcionará de manera óptima en los navegadores Web como Mozilla, Firefox, Netscape, Lynx, Internet Explorer, Maxton.
- El sistema funcionará correctamente en todos los tamaños de ventanas de los navegadores Web.

#### **Eficiencia**

#### ❖ **Rendimiento**

El peso de esta Subcaracterística es **Alto** ya que:

- El sistema estará disponible 24 horas los 7 días de la semana los 365 días del año.
- El sistema procesará una transacción en un tiempo promedio de 1 segundo.
- El sistema minimizará el volumen de datos en las peticiones y además optimizará el uso de recursos críticos como la memoria, para ello se potenciará como regla guardar en la cache de datos y recursos de alta demanda.
- El sistema seguirá buenas prácticas de programación para incrementar el rendimiento en operaciones costosas para la máquina virtual como la creación de objetos, se deberá usar siempre que sea posible el patrón Singleton, destruir referencias que ya no estén siendo usadas, optimizar el trabajo con cadenas, entre otras buenas prácticas que ayudan a mejorar el rendimiento.
- El sistema identificará por cada CU aquellas operaciones que impliquen un elevado nivel de procesamiento en la base de datos y usará procedimientos almacenados para manejarlas.
  - Interfaz de software
- El sistema tendrá un alto desempeño con el objetivo de que el tiempo de respuesta tiene que ser lo más bajo posible para los clientes que acceden a través de un navegador Web.
- Los clientes tendrán acceso a los módulos y aplicaciones a través de cualquier navegador web, recomendados Mozilla 1.5, Internet Explorer 5.0 o superior.
- El sistema se diseñará de manera que pueda brindar servicio a:
  - 6 Coordinaciones Nacionales, y dentro de cada coordinación hay aproximadamente 20 dependencias.
  - 24 Delegaciones.
  - 88 Subdelegaciones.
  - 24 Medicaturas Forenses.
  - Supervisión de Delegaciones Estatales.
  - Inspectoría General.
  - Asesoría Jurídica.
  - Policía Internacional.
- El sistema en su pico más crítico soportará un volumen de más de 7800 usuarios conectados concurrentemente.

- El sistema soportará un volumen ordinario de más de 590 usuarios conectados concurrentemente.

#### ❖ **Utilización de recursos**

El peso de esta Subcaracterística es **Medio** ya que:

- El sistema debe permitir un elevado volumen de almacenamiento, permitiendo la escalabilidad de dicho volumen de manera más fácil, económica y en menos tiempo. Para ello los datos serán almacenados en una cabina HP de última generación EVA 8000.
- El sistema usará igualmente la potencialidad del EVA 8000 para realizar balanceo de carga entre los servidores de aplicación permitiendo acortar los tiempos de respuesta.
  - Interfaz de hardware
- El sistema necesitará interactuar con impresoras para imprimir los diferentes documentos que genera la aplicación como respuesta a las funcionalidades del sistema, estas deberán tener una asignación de papel y tonel que les permita ejecutar todas las peticiones que la aplicación les haga.
- El sistema necesitará interactuar con dispositivos capta huellas para diferentes funcionalidades claves de la aplicación donde se necesite reforzar la seguridad como la autenticación y la autorización para gestionar recursos críticos. Es necesario que los mismos estén ubicados en aquellas terminales desde donde se acceda a recursos críticos de la aplicación.
- El sistema necesitará de un servicio de red VPN, que funcione 24x7, los 7 días de la semana los 365 días del año sin interrupción.
- Interfaz de comunicación
- El sistema soportará el protocolo NIST para la comunicación con el sistema externo AFIS Criminal.
- El sistema usará el formato estándar WSDL para la descripción de los servicios web.
- El sistema implementará mecanismos de encriptación de datos para el intercambio de información con sistemas externos.
- El sistema utilizará mecanismos de compactación de los datos que se intercambiarán con sistemas externos con el objetivo de minimizar el tráfico en la red y economizar el ancho de banda.

La característica de **Mantenibilidad** no se medirá ya que las métricas que propone la ISO/IEC 9126-3 no se ajustan a las características propias del proyecto Modernización del CICPC, además según la NC ISO/IEC 12119 "Paquete de software. Requerimientos de Calidad y Pruebas" las tres primeras características de calidad son obligatorias para realizar la medición (Funcionalidad, Fiabilidad y Usabilidad). Y las tres últimas son opcionales (Eficiencia, Portabilidad y Mantenibilidad). Sucede así con la característica de **Portabilidad**, la cual tampoco se medirá.

La subcaracterística **Conformidad** correspondiente a cada una de las características de calidad a evaluar no se medirán ya que actualmente no existen en el Ministerio de Informática y Comunicaciones (MIC) las regulaciones, normas u otras convenciones relacionadas, que estipulen cómo deben regirse legalmente las mismas.

### 3.2 Cálculo de las métricas propuestas para medir la calidad del SIIPOL

Tabla 13: Resultados de las métricas de Funcionalidad

<i>Característica: Funcionalidad</i>				
<b>Subcaracterística</b>	<b>Nombre de la métrica</b>	<b>Métrica</b>	<b>Datos de la métrica</b>	<b>Resultado de la métrica</b>
<b>Idoneidad</b>	Adecuación funcional	$X = 1 - A/B$	A = 49 B = 77	X = 0.37
<b>Exactitud</b>	Exactitud	$X = A/B$	A = 57 B = 77	X = 0.74
<b>Seguridad</b>	Controlabilidad de acceso	$X = 1 - A/B$	A = 3 B = 77	X = 0.97

Tabla 14: Resultados de las métricas de Confiabilidad

<b>Característica: Confiabilidad</b>				
<b>Subcaracterística</b>	<b>Nombre de la métrica</b>	<b>Métrica</b>	<b>Datos de la métrica</b>	<b>Resultado de la métrica</b>
<b>Madurez</b>	Adecuación de las pruebas	$X = A/B$	A = 1291 B = 1683	X = 0.76
<b>Tolerancia ante fallos</b>	Evitación de fallos	$X = A/B$	A = 237 B = 84	X = 2.82
<b>Recuperabilidad</b>	Recuperabilidad	$X = A/B$	A = 2 B = 3	X = 0.67

Tabla 15: Resultados de las métricas de Usabilidad

<b>Característica: Usabilidad</b>				
<b>Subcaracterística</b>	<b>Nombre de la métrica</b>	<b>Métrica</b>	<b>Datos de la métrica</b>	<b>Resultado de la métrica</b>
<b>Comprensibilidad</b>	Entendimiento de funciones	$X = A/B$	A = 461 B = 625	X = 0.73
<b>Operabilidad</b>	Chequeo de validez de entrada	$X = A/B$	A = 1820 B = 2334	X = 0.78

Tabla 16: Resultados de las métricas de Eficiencia

<b>Característica: Eficiencia</b>				
<b>Subcaracterística</b>	<b>Nombre de la métrica</b>	<b>Métrica</b>	<b>Datos de la métrica</b>	<b>Resultado de la métrica</b>
<b>Rendimiento</b>	Rendimiento	X = número de tareas por unidad de tiempo.	X = 16 peticiones /seg	X = 16 peticiones /seg
<b>Utilización de recursos</b>	Uso de memoria	X = tamaño en bytes.	X = 650 Mb/seg	X = 650 Mb/seg

Una vez resueltas cada una de las métricas se procede a recopilar todos los resultados en la siguiente plantilla. (Ver tabla 17)

Tabla 17: Resultados de las métricas aplicadas en la evaluación

Característica	Subcaracterística	Peso	Métrica	Nivel Requerido	Resultado de la métrica	Resultado Real
Funcionalidad	Idoneidad	Alto	$X = 1 - A/B$	$0 \leq X \leq 1$	$X = 0.37$	Cerca de 1
	Exactitud	Alto	$X = A/B$	$0 \leq X \leq 1$	$X = 0.74$	Cerca de 1
	Seguridad	Alto	$X = 1 - A/B$	$0 \leq X \leq 1$	$X = 0.97$	Cerca de 1
Confiabilidad	Madurez	Alto	$X = A/B$	$0 \leq X$	$X = 0.76$	Mientras X sea mayor.
	Tolerancia ante fallos	Alto	$X = A/B$	$0 \leq X$	$X = 2.82$	Mientras X sea mayor.
	Recuperabilidad	Alto	$X = A/B$	$0 \leq X \leq 1$	$X = 0.67$	Mientras X sea mayor.
Usabilidad	Comprensibilidad	Medio	$X = A/B$	$0 \leq X \leq 1$	$X = 0.73$	Cerca de 1
	Operabilidad	Medio	$X = A/B$	$0 \leq X \leq 1$	$X = 0.78$	Cerca de 1
Eficiencia	Rendimiento	Alto	X = número de tareas por unidad de tiempo.	Mientras mayor mejor.	X = 16 peticiones /seg	Mientras mayor mejor.
	Utilización de recursos	Medio	X = tamaño en bytes.	Mientras menor sea, mejor.	X = 650 Mb/seg	Mientras menor sea, mejor.

Cuando se tienen los resultados recopilados de las métricas aplicadas en la anterior plantilla, cada uno de estos valores se ajusta a la escala correspondiente y se les asigna la puntuación. (Ver tabla 18)

Tabla 18: Puntuación asignada a las subcaracterísticas de calidad

Características y Subcaracterísticas de Calidad	Puntuación			
	3	2	1	0
<b>1. Funcionalidad</b>		X		
<b>1.1 Idoneidad</b>				X
<b>1.2 Exactitud</b>		X		
<b>1.3 Seguridad</b>	X			
<b>2. Confiabilidad</b>			X	
<b>2.1 Madurez</b>			X	
<b>2.2 Tolerancia ante fallos</b>		X		
<b>2.3 Recuperabilidad</b>			X	
<b>3. Usabilidad</b>		X		
<b>3.1 Comprensibilidad</b>		X		
<b>3.2 Operabilidad</b>		X		
<b>4. Eficiencia</b>	X			
<b>4.1 Rendimiento</b>	X			
<b>4.2 Utilización de recursos</b>		X		

Como se puede observar el caso de la característica Eficiencia, los pesos de las subcaracterísticas que le corresponden que son Rendimiento y Utilización de recursos, son Alto y Medio respectivamente, por lo que se realiza el siguiente procedimiento para definir la puntuación de la característica Eficiencia.

Se le asigna un valor a cada uno que la suma de ambos sea 1:

Alto: 0.6

Medio: 0.4.

La puntuación de cada una de las subcaracterísticas es la siguiente:

Rendimiento: 3

Utilización de recursos: 2

Entonces se multiplican los valores de cada uno de los pesos por la puntuación de la subcaracterística:

$$0.6 * 3 = 1.8$$

$$0.4 * 2 = 0.8$$

Obteniéndose de la suma de ambos  $1.8 + 0.8 = 2.6$ .

Este resultado se aproxima a 3 que finalmente es la puntuación de la característica de **Eficiencia**.

En la siguiente tabla (Ver tabla 19) se ubican cada una de las puntuaciones de las características.

Tabla 19: Puntuación asignada a las características de calidad

Puntuación	
Funcionalidad	2
Confiabilidad	1
Usabilidad	2
Eficiencia	3
Valor Final	2

En este caso, solo la característica de Usabilidad difiere en el peso (Medio) con respecto a las demás características. Por lo que se le asigna un valor al peso Alto y Medio, teniendo en cuenta la cantidad de subcaracterísticas, que en este caso son cuatro.

Se le asigna un valor a cada uno de los pesos:

Alto: 0.28

Medio: 0.16

La puntuación de cada una de las características es la siguiente:

Funcionalidad: 2

Confiabilidad: 1

Usabilidad: 2

Eficiencia: 3

Entonces se multiplican los valores de cada uno de los pesos por la puntuación de la subcaracterística:

$$0.28 * 2 = 0.56$$

$$0.28 * 1 = 0.28$$

$$0.16 * 2 = 0.32$$



$$0.28 * 3 = 0.84$$

Estos valores se suman  $0.56 + 0.28 + 0.32 + 0.84 = 2.00$ , y se obtiene el valor final.

Lo que representa el Grado de Conformidad del producto software siendo este **Suficientemente conforme** (Ver tabla 20), que en correspondencia con el Criterio de Evaluación se sugiere **Pequeñas modificaciones** para el software (Ver tabla 21).

Tabla 20: Grado de Conformidad del SIIPOL

Grado de Conformidad		
3		Conforme
2	X	Suficientemente conforme
1		Medianamente conforme
0		No conforme

Tabla 21: Criterio de Evaluación del SIIPOL

Criterio de Evaluación
( ) Sin modificaciones
( X ) Pequeñas modificaciones
( ) Grandes modificaciones
( ) Nueva elaboración

### 3.3 Principales problemas que fueron detectados durante el proceso de evaluación

En el software evaluado se encontraron varias deficiencias principalmente relacionadas con la funcionalidad, la confiabilidad y la usabilidad.

En cuanto a la Funcionalidad se encontraron problemas con las funcionalidades de los CU del sistema. La mayoría de los CU no cumplía con sus especificaciones y muchos no fueron implementados con los niveles de precisión que ellos requerían. Hubo problemas con la asignación de perfiles y permisos de usuarios ya que no se tenía definido desde un inicio la asignación de estos a los diferentes roles del sistema.

La confiabilidad es la característica más crítica que presenta el software debido a que los CP diseñados no abarcaban todas las funcionalidades que establecía la especificación del CU. Tampoco se previó la evitación de los fallos que podían ocurrir en el software por lo que muchas de las NC que se detectaban en una iteración se

repetían en la posterior iteración de pruebas. Otro de los problemas que se presentaron fue la inconsistencia de la Base de Datos debido a las pruebas constantes de los programadores.

La usabilidad del software resultó con algunos problemas ya que para probar un CU muchas de las interfaces de usuario no daban a entender con claridad los pasos para realizar una función determinada. Tampoco se cumplieron en su totalidad las pautas de arquitectura de la información establecida en el proyecto, lo que provocó un desajuste entre las diferentes interfaces de usuario de los diferentes Módulos.

### **3.4 Propuesta de acciones correctivas**

- Revisión de los procesos (políticas y procedimientos).
- Capacitación a los Equipos de Desarrollo (ED), y en general a todos los integrantes del proyecto.
- Crear, revisar, mejorar estándares y especificaciones.
- Reorganizar, reubicar a las personas y recursos dentro del proyecto.
- Mejorar, reforzar las actividades de auditoría.
- Reforzar el tema del control a la asignación de responsabilidades dentro del ED para garantizar que no quede nada por hacer y que cada cual tenga su responsabilidad claramente definida.
- Realizar proceso de revisión y prueba más rigurosa y eficiente, con el objetivo de detectar lo antes posible, los posibles defectos antes de que pasen a etapas posteriores.
- Tomar medidas con los integrantes del equipo que no cumplieran con lo establecido por sus superiores, para evitar que estos tomaran decisiones por su cuenta.

### **3.5 Conclusiones del capítulo**

En el capítulo se aplicó el proceso propuesto para realizar la evaluación de la calidad del SIIPOL. Se presentan los resultados que se obtuvieron, se describen cuáles fueron las principales deficiencias detectadas en el software recomendando como podrían evitarse las mismas, en etapas posteriores.

## **CONCLUSIONES**

La evaluación de la calidad de un producto software es un tema tan importante y polémico, ya que un software sin la calidad requerida hace perder prestigio a la organización que lo desarrolla. En el Proyecto Modernización del CICPC no se tenía establecido cómo llevar a cabo este proceso de evaluación por lo que se propuso realizar un plan de evaluación para medir la calidad de un producto software durante el proceso de pruebas.

Para darle cumplimiento a los objetivos propuestos se realizó un profundo estudio acerca de normas y modelos calidad, seleccionándose las adecuadas para realizar un plan de evaluación para la medición de la calidad del producto software en el Proyecto Modernización del CICPC. Se modelaron los diferentes subprocesos que se ejecutan, se propusieron las métricas para realizar la medición de la calidad durante el proceso de pruebas, y se abordaron los principales roles y responsabilidades dentro del mismo. Como resultado del proceso de evaluación se obtuvo el Grado de conformidad: Suficientemente conforme, y como Criterio de Evaluación: Pequeñas modificaciones. Las principales deficiencias detectadas en el software se relacionan con la funcionalidad, la confiabilidad y la usabilidad. Siendo propuestas las acciones correctivas para mitigarlos: reforzamiento del control a la asignación de responsabilidades, realizar proceso de revisión y prueba más rigurosa y eficiente, mejorar las actividades de auditoría, entre otras.

Un aporte novedoso del presente Trabajo de Diploma es que la propuesta de este método de evaluación es aplicada por primera vez en Cuba.

## **RECOMENDACIONES**

El proceso de evaluación de la calidad del producto software se efectuó satisfactoriamente, sin embargo, con el propósito de alcanzar mejores resultados se recomienda:

- ❖ Realizar el proceso de evaluación propuesto desde el inicio del proyecto, en todas las etapas e iteraciones de pruebas, para medir y evaluar la calidad del producto software durante su construcción.
- ❖ Crear conciencia a los Equipos de Desarrollo del proyecto y en especial al Equipo de Calidad, sobre la importancia de llevar a cabo un buen proceso de pruebas.
- ❖ Registrar todos los datos que arrojen las pruebas para así poder medir la calidad y establecer una comparación entre cada uno de las etapas de pruebas, y ver el grado de avance que se tuvo en el software.
- ❖ Elaborar un informe de pruebas al finalizar cada una de las iteraciones en este proceso, para dar a conocer el estado en el cual se encuentra el proyecto.
- ❖ Utilizar las medidas directas que proporcionan las herramientas automáticas de pruebas para evaluar determinadas características de calidad del software en revisión.

## REFERENCIAS BIBLIOGRÁFICAS

1. IEEE 729-83. 1983.
2. Pressman, R.S., Ingeniería del software. Un enfoque práctico. . Vol. 1. 1998: McGrawHill.
3. Norma ISO 8402
4. International Organization for Standardization (ISO).
5. NC ISO/IEC 9126-1.
6. ISO/IEC 14598-1:1999 Information technology - Software product evaluation - Part 1: General overview. 1999.
7. Angeleri, M.P.M., Programa Calidad de software del proyecto BID-FOMIN ATN-ME-8532-RG, I.A.D.N.Y. CERTIFICACIÓN, Editor.
8. ISO/IEC 14598-5, International Standard, Information technology - Software product evaluation - Part 5: Process for evaluators. 1998.
9. Pérez, C.L. Modelo de Madurez de la Capacidad del Software. 2005 [cited 2007 13 de febrero]; Available from: [http://www.cii-murcia.es/informas/ene05/articulos/Modelo\\_de\\_Madurez\\_de\\_la\\_Capacidad\\_del\\_Software.html](http://www.cii-murcia.es/informas/ene05/articulos/Modelo_de_Madurez_de_la_Capacidad_del_Software.html).
10. SEI, Standard CMMISM Appraisal Method for Process Improvement (SCAMPI), Version 1.1: Method Definition Document. 2001, Carnegie Mellon Software Engineering Institute.
11. González., R.A.H.L.y.S.C., El paradigma cuantitativo de la investigación científica. 2002, Ciudad de la Habana Editorial Universitaria.
12. Cukier, J.J. Evaluaciones CMMI (Standard CMMI Appraisal Method for Process Improvement). 2006 [cited 2007 3 de mayo ]; Available from: <http://www.ieee.org.ar/downloads/2006-cukier-cmmi-scampi.pdf>.
13. González, P.H., et al. Marco de Evaluación CMMI-SW (por etapas). 2005 [cited 2007 7 de febrero]; Available from: <http://alarcos.inf-cr.uclm.es/doc/calidad/Trabajos/CMMI%20por%20etapas%202.pdf>.
14. Qualitatis. Métodos de evaluación SCAMPI B y C. 2006 [cited 2007 3 de mayo ]; Available from: [http://www.qualitatis.org/files/1/scampi\\_b\\_c.pdf](http://www.qualitatis.org/files/1/scampi_b_c.pdf).
15. Villa, M.d.I., M. Ruiz, and I. Ramos. Modelos de Evaluación y Mejora de Procesos: Análisis Comparativo 2004 [cited 2007 5 de febrero]; Available from: <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-120/paper4.pdf>.
16. Sagrario, A. Mejora de procesos: La evolución natural. 2002 [cited 2007 3 de mayo]; Available from: <http://www.als-es.com/recursos/articulos/mejora-procesos-cmmi.pdf>.

- 17.** Calvo-Manzano, S. Feliu, and Serrano. CMMI: Mitos y Realidades. 2006 [cited 2007 3 de mayo]; Available from:  
[http://www.aemes.org/conferencias/VII\\_ConfAEMES/pdf/04\\_CMMI\\_%20Mitos\\_Realidades.pdf](http://www.aemes.org/conferencias/VII_ConfAEMES/pdf/04_CMMI_%20Mitos_Realidades.pdf).
- 18.** Taboada, G. Establecer estrategias de implementación de un proceso de mejora continua utilizando modelos de calidad. 2007 [cited 2007 3 de mayo]; Available from:  
[https://www.palermo.edu/Intranet/ingenieria/contenidos\\_minimos\\_informatica2007.html#](https://www.palermo.edu/Intranet/ingenieria/contenidos_minimos_informatica2007.html#).
- 19.** Villena, A.M., et al. Modelos de Calidad del Software Gestión de Proyectos 05/06. 2005 [cited 2007 13 de febrero]; Available from:  
[http://www.lcc.uma.es/~guzman/gp/docs/22\\_MAY\\_MAN\\_2.pdf](http://www.lcc.uma.es/~guzman/gp/docs/22_MAY_MAN_2.pdf).
- 20.** IEEE, 1991. 1991.
- 21.** Myers, G., Fase de Elaboración. FT Prueba (Procedimientos genéricos y aplicación de algunos tipos de pruebas simples). 2005.
- 22.** IEEE93 Standard Glossary of Software Engineering Terms 1993.
- 23.** ISO/IEC Standard 9126, Software Product Evaluation – Quality Characteristics and Guidelines for their use. 1992.
- 24.** Ragland, B., Measure, Metric or Indicator: What’s the Difference? 1995.
- 25.** Rational Unified Process. 2001 [cited].
- 26.** Kelvin, L.
- 27.** ISO/IEC 9126-3:2003 Software engineering -- Product quality - Part 3: Internal metrics. 2003.

## **BIBLIOGRAFÍA**

1. ISO/IEC Standard 9126, Software Product Evaluation – Quality Characteristics and.
2. *Requerimientos Suplementarios*. 2007.
3. **Jimenez, Sasha**. *Documento Especificaciones Suplementarias*. 2006.
4. **Pressman, Roger**. *Ingeniería del software. Un enfoque práctico*. . s.l. : McGraw Hill, 1998.
5. ISO/IEC 14598-1: 1999 Information technology - Software product evaluation -Part 1: General overview. . 1999.
6. **Angeleri, M.P.M**. Programa Calidad de software del proyecto BID-FOMIN ATN-ME-8532-RG.
7. ISO/IEC 14598-5 International Standard. *Information technology - Software product evaluation - Part 5: Process for evaluators*. 1998.
8. **González., R.A.H.L.y.S.C**. El paradigma cuantitativo de la investigación científica. Ciudad de la Habana : Editorial Universitaria, 2002.
9. ISO/IEC Standard 9126. *Software Product Evaluation – Quality Characteristics and*. 1992.
10. **Losavio, F., Ortega, D., Pérez**. MODELO CONCEPTUAL PARA EAI.
11. **Lovelle, Juan Manuel Cueva**. Calidad del Software. *www.uniovi.es*. [En línea] Grupo GIDIS, Universidad Nacional de la Pampa, 1999.
12. **Z., Dr. Marcello Visconti**. Ingeniería de Software Avanzada.
13. **Guillermo Javier Lafuente, Luis Olsina**. Catalogando Métricas Web. *gidis.ing.unlpam.edu.ar*. [En línea] <http://gidis.ing.unlpam.edu.ar>.
14. **Muñoz, Dra. Coral Calero**. MODELOS DE CALIDAD.Calidad de Sistemas de Información. Universidad de Castilla-La Mancha : s.n., 2005.
15. **Carlos, Universidad Rey Juan**. Calidad del software.
16. **Abraham Dávila (edavila@pucp.edu.pe), Karin Melendez (melendez.ka@pucp.edu.pe) y Luis Flores (flores.la@pucp.edu.pe)**. Determinación de los Requerimientos de Calidad del Producto Software Basados en Normas Internacionales. *Sección Ingeniería Informática*. Lima, Perú : s.n.
17. **Society, Software Engineering Standards Committee of the IEEE Computer**. IEEE Std 1044-1993 Standard Classification for Software Anomalies. United States of America : s.n., 1993. ISBN 1-55937-383-0.
18. **Fernández-Pello, J**. Software Quality Assurance Network. [En línea] 27 de Junio de 2007. [Citado el: 03 de Agosto de 2007.] <http://www.softqanetwork.com/>.
19. **Francisco Ruiz, M. P**. *Mantenimiento del Software*. Ciudad Real, España : s.n., 2000/2001.

20. **González, C. D.** Consultoría en Usabilidad y Estándares web. [En línea] [Citado el: 31 de Julio de 2007.] <http://www.usabilidadweb.com.ar>.
21. **Hetzel, W.** Making Software Measurement Work. [En línea] 1993.
22. **Iris Sofía, L.** [www.angelfire.com](http://www.angelfire.com). [En línea] [Citado el: 1 de Agosto de 2007.] [http://www.angelfire.com/theforce2/irissofia\\_leonel/metricas.htm](http://www.angelfire.com/theforce2/irissofia_leonel/metricas.htm).
23. **Mamdouh El Cuera, R. C.** [En línea] [Citado el: 31 de Julio de 2007.] <http://squac.iti.upv.es/JTS/JTS2005/contenido.html>.
24. **Munson, J.** Software Engineering Measurement. 2003.
25. **Rossi, L.** METRICAS DE SOFTWARE.
26. **Santos, Profesor Adjunto Jose Manuel.** Mediciones. Enfoques de las métricas.
27. **17000:2005, NC ISO/IEC.** Evaluación de la conformidad- Vocabulario y principios generales. Suiza : s.n., 2005.
28. **2005, NC ISO/IEC 17011:.** EVALUACIÓN DE LA CONFORMIDAD – REQUISITOS GENERALES PARA LOS ORGANISMOS DE ACREDITACIÓN QUE REALIZAN LA ACREDITACIÓN DE ORGANISMOS DE EVALUACIÓN DE LA CONFORMIDAD . Suiza : s.n., 2005.
29. **9126-3, ISO/IEC TR.** ISO/IEC TR 9126-3:2003 Software engineering -- Product quality - Part 3: Internal metrics. 2003. Switzerland : s.n., 2003.
30. **2005, NC ISO/IEC 9126-1:.** INGENIERÍA DE SOFTWARE—CALIDAD DEL PRODUCTO—PARTE 1: MODELO DE LA CALIDAD. Suiza : s.n., 2005.
31. Software Quality Management. [En línea] 2006. <http://softqm.blogspot.com/2006/10/auditora-de-la-calidad-del-software.html>.



## **GLOSARIO DE TÉRMINOS Y DEFINICIONES**

**CICPC:** Cuerpo de Investigaciones Científicas Penales y Criminalísticas.

**SIIPOL:** Sistema de Investigación e Información Policial.

**NC:** No conformidad: incumplimiento a un requisito.

**CU:** Caso de Uso.

**NP:** No Procede.

**IU:** Interfaz de Usuario.

**ED:** Equipo de Desarrollo.

**BD:** Base de Datos.

**PC:** Personal Computer (Computadora Personal).

**INTTT:** Instituto Nacional de Tráfico y Transporte Terrestre.

**Evaluación de la conformidad:** Demostración de que se cumplen los requisitos especificados relativos a un producto, proceso, sistema, persona u organismo.

**Producto:** Resultado de un proceso.

**Estándares:** Son acuerdos documentados que contienen especificaciones técnicas u otros criterios específicos para ser usados como referentes, guías o definiciones de características, para asegurar que materiales, productos, procesos y servicios son obtenidos o han sido realizados de acuerdo a sus propósitos.

**Informe de Evaluación:** Es el informe que presenta los resultados de la evaluación y cualquier otra información relevante para la evaluación.

**Evaluador:** La persona que realiza la evaluación.

**Evaluación de un producto software:** Operación técnica que consiste en la generación de una o más características de un producto software de acuerdo a un procedimiento especificado.

**Acción correctiva:** Acción tomada para eliminar la causa de una no conformidad detectada u otra situación indeseable.

**Registro de un producto software:** Evidencias objetivas documentadas de todas las actividades ejecutadas y todos los resultados obtenidos en el proceso de evaluación.

**Desarrollador del producto software:** La persona u organización que fabrica un producto software.

**IBIS:** Sistema Integrado de Identificación Balística.

**AFIS:** Automated Fingerprint Identification System (Sistema Automático de Identificación por Huellas Digitales).

**SAIME:** Sistema Autónomo de Identificación, Migración y Extranjería.

**EVA 8000:** Este nuevo dispositivo es un receptor multimedia que se conecta a cualquier sistema de home cinema doméstico para acceder a los contenidos multimedia almacenados en los ordenadores, unidades de almacenamiento en red y reproductores USB. El EVA 8000 pone al alcance del usuario, de una forma fácil y cómoda, un universo digital a través del televisor.

**HP:** Es proveedor de soluciones tecnológicas para particulares, empresas e instituciones en todo el mundo. La oferta de la compañía abarca infraestructura de tecnologías de la información, informática personal y dispositivos de acceso, servicios globales, imagen e impresión.

**Protocolo NIST:** Protocolo propuesto por Instituto Nacional Estándares y Tecnologías de USA (NIST).

**Servicio de red VPN:** Servicios de redes privadas virtuales.

**WSDL:** Es el lenguaje de la interfaz pública para los servicios Web. Es una descripción basada en XML de de los requisitos funcionales necesarios para establecer una comunicación con los servicios Web y cómo debe establecerse la misma.

**RAID 1:** (Originalmente del inglés Redundant Array of Inexpensive Disks, 'conjunto redundante de discos baratos', en la actualidad también de Redundant Array of Independent Disks, 'conjunto redundante de discos independientes') hace referencia a un sistema de memoria que usa múltiples discos duros entre los que distribuye o replica los datos.

## **ANEXOS**

Anexo1: Encuesta para la Pre-Evaluación del software

**ENCUESTA PARA LA PREEVALUACIÓN DEL PRODUCTO DE SOFTWARE  
PROYECTO CICPC**

Evalúe las siguientes subcaracterísticas según su criterio sobre cómo **debe** comportarse el software que se está produciendo de acuerdo a lo pactado con el cliente. Asigne pesos: Alto, Medio o Bajo. Esta pre-evaluación se tendrá en cuenta para evaluar el producto una vez terminadas las pruebas internas del software.

<i>Características</i>	<i>Sub-Características</i>	<i>Significado</i>	<i>Pesos (alto/medio/bajo)</i>
<b>Funcionalidad</b>	<i>Idoneidad</i>	La capacidad del producto software para proporcionar un apropiado conjunto de funciones para las tareas especificadas y los objetivos de los usuarios.	
	<i>Exactitud</i>	La capacidad del producto software para proporcionar resultados correctos o efectos convenidos con el grado necesario de precisión. (Esto incluye el grado necesario de precisión de los valores calculados).	
	<i>Seguridad</i>	La capacidad del producto 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.	
<b>Confiabilidad</b>	<i>Madurez</i>	La capacidad del producto software para evitar fallas como resultado de fallas en el software.	
	<i>Tolerancia ante fallos</i>	La capacidad del producto software para mantener un nivel de desempeño especificado en casos de fallas (en el software) o de que se infrinjan sus enlaces especificados.	
	<i>Recuperabilidad</i>	La capacidad del producto software para restablecer su nivel de desempeño y recuperar los datos directamente afectados en caso de falla (así como con el tiempo y el esfuerzo necesario para ello).	
<b>Usabilidad</b>	<i>Comprensibilidad</i>	La capacidad del producto software para permitirle al usuario entender si el software es conveniente, cómo puede ser utilizada para las tareas determinadas y las condiciones de uso.	
	<i>Operabilidad</i>	La capacidad del producto software para permitirle al usuario operarlo y controlarlo.	
<b>Eficiencia</b>	<i>Rendimiento</i>	La capacidad del producto software para proporcionar adecuados tiempos de respuesta, de procesamiento y de tasas de eficiencia en el desempeño de su función, bajo condiciones establecidas.	
	<i>Utilización de recursos</i>	La capacidad del producto software para utilizar una apropiada cantidad y tipos de recursos cuando el software desempeña su función bajo condiciones establecidas. Los recursos humanos están incluidos como parte de la productividad.	

Anexo 2: Métricas Internas de la Norma ISO/IEC 9126-3

<b>MÉTRICAS DE FUNCIONALIDAD</b>				
<b>Subcaracterística: Idoneidad</b>				
<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Adecuación funcional</b>	¿Cuán adecuadas son las funciones verificadas?	Contar la cantidad de funciones implementadas que son fáciles de entender para completar la tarea especificada, y luego se divide entre la cantidad total de funciones implementadas.  Pueden ser medidas en: Todas o parte de las especificaciones de diseño. Módulos completados / componentes del producto de SW.	<b><math>X = 1 - A/B</math></b> A: Número de funciones en las cuales se detectaron problemas en la evaluación. B: Número de funciones evaluadas.	<b><math>0 \leq X \leq 1</math></b>  A mayor cercanía al 1 resultará más adecuada.
<b>Plenitud de la implementación funcional</b>	¿Cuán completa es la implementación funcional?	Contar la cantidad de funciones que no se encontraron durante la evaluación y compararlo con la cantidad de funciones descritas en la especificación de requisitos.	<b><math>X = 1 - A/B</math></b> A: Número de funciones perdidas detectadas en la evaluación. B: Número de funciones descritas en especificación de requisitos.	<b><math>0 \leq X \leq 1</math></b>  A mayor cercanía al 1 resultará más adecuada.
<b>Cobertura de la implementación funcional</b>	¿Cuán correcta es la implementación funcional?	Contar la cantidad de funciones incorrectamente implementadas o no encontradas y compararlo con la cantidad de funciones descritas en la especificación de requisitos.	<b><math>X = 1 - A/B</math></b> A: Número de funciones incorrectamente implementadas o funciones perdidas detectadas. B: Número de funciones descritas en la especificación de requisitos.	<b><math>0 \leq X \leq 1</math></b>  A mayor cercanía al 1 resultará más adecuada.

<b>Estabilidad de la especificación funcional</b>	¿Cuán estable es la especificación funcional durante el ciclo de desarrollo de software?	Contar la cantidad de funciones cambiadas (adicionadas, modificadas o borradas) durante las fases de desarrollo y luego compararla con la cantidad de funciones descritas en la especificación de requisitos.	$X = 1 - A/B$ A: Número de funciones cambiadas durante el desarrollo del ciclo de vida. B: Número de funciones descritas en la especificación de requerimientos.	$0 \leq X \leq 1$ A mayor cercanía al 1 resultará más adecuada.
---	--	---	--	--

**Subcaracterística: Exactitud**

<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Exactitud computacional</b>	¿Cuán completos han sido implementados los requerimientos de precisión?	Contar la cantidad de funciones que tuvieron que implementar requerimientos de precisión y compararlo con la cantidad de funciones con requerimientos específicos de precisión.	$X = A/B$ A: Número de funciones en las cuales ciertos requerimientos específicos han sido implementados, como se ha confirmado en la evaluación. B: Número de funciones para las cuales ciertos requerimientos específicos necesitan ser implementados.	$0 \leq X \leq 1$ A mayor cercanía al 1 resultará más adecuada.
<b>Exactitud</b>	¿Cuán completa	Contar la cantidad de elementos de datos que	$X = A/B$	$0 \leq X \leq 1$

	<i>estuvo la implementación de los niveles de precisión específicos para los elementos de datos?</i>	<i>tienen niveles específicos de precisión y compararlo con el total de elementos de datos que lo requerían según la especificación.</i>	<i>A: Número de datos implementados con los niveles de precisión especificados, confirmado en la evaluación.</i>  <i>B: Número de datos que requirieron los niveles de precisión.</i>	<i>A mayor cercanía al 1 resultará más adecuada.</i>
--	--	--	---	--

**Subcaracterística: Interoperabilidad**

<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Intercambiabilidad de datos (en base a su formato)</b>	<i>¿Cuán correctos han sido implementados los formatos de interfaces de datos?</i>	<i>Contar la cantidad de interfaces de datos que fueron implementados correctamente tal como se describió en la especificación y compararlo con la cantidad de datos a intercambiar según la especificación.</i>	$X = A/B$  <i>A: Número de formatos de datos que han sido correctamente implementados según la especificación.</i>  <i>B: Número de formato de datos a intercambiar.</i>	$0 \leq X \leq 1$  <i>A mayor cercanía al 1 resultará más correcto.</i>
<b>Consistencia de la Interfaz (Protocolo)</b>	<i>¿Cuán correctos fueron implementados los protocolos de interfaz?</i>	<i>Contar la cantidad de protocolos de interfaz que fueron implementados correctamente tal como se describió en la especificación y compararlo con la cantidad de protocolos de interfaz a ser implementados según la especificación.</i>	$X = A/B$  <i>A: Número de protocolos de interfaz que ponen en práctica un formato consecuente como en la</i>	$0 \leq X \leq 1$  <i>A mayor cercanía al 1 resultará más consistente.</i>

			<p>especificación confirmada en la revisión.</p> <p>B: Número de protocolos de interfaz a ser implementados como en la especificación.</p>	
<b>Subcaracterística: Seguridad</b>				
<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Auditabilidad de acceso</b>	¿Cuán auditable es la autenticación de acceso?	Contar la cantidad de tipos de acceso que fueron autenticados correctamente según la especificación y compararlo con la cantidad de tipos de acceso que se definen en la especificación.	<p><math>X = A/B</math></p> <p>A: Número de tipos de acceso que se autenticaron correctamente.</p> <p>B: Número de tipos de acceso que se deben autenticar según las especificaciones.</p>	<p><math>0 \leq X \leq 1</math></p> <p>A mayor cercanía al 1 resultará más auditable.</p>
<b>Controlabilidad de acceso</b>	¿Cuán controlable es el acceso al sistema?	Contar la cantidad de requerimientos de control de acceso que fueron implementados correctamente y compararlo con la cantidad de requerimientos de control de acceso descritos en la especificación.	<p><math>X = A/B</math></p> <p>A: Número de accesos con requerimientos de control implementados correctamente como dicen las especificaciones.</p>	<p><math>0 \leq X \leq 1</math></p> <p>A mayor cercanía al 1 resultará más controlable.</p>



			<i>B: Número de accesos con requerimientos de control en las especificaciones.</i>	
<b>Prevención de datos</b>	<i>¿Cuán completa es la implementación de la prevención de corrupción de datos?</i>	<i>Contar la cantidad de instancias de prevención de corrupción de datos implementadas y compararlo con la cantidad de instancias de operación/acceso especificado en los requerimientos como datos que pueden destruirse o corromperse con facilidad.</i>	<p><b><math>X = A/B</math></b></p> <p><i>A: Número de instancias de prevención de corrupción de datos implementadas según especificaciones.</i></p> <p><i>B: Número de instancias de operaciones o accesos identificadas en los requerimientos como capaces de destruir o corromper datos</i></p> <p><i>Nota: Considerar los niveles de seguridad al usar esta métrica.</i></p>	<p><b><math>0 \leq X \leq 1</math></b></p> <p><i>A mayor cercanía al 1 resultará más completo.</i></p>
<b>Encriptamiento de datos</b>	<i>¿Cuán completa es la implementación de encriptación de datos?</i>	<i>Contar la cantidad de instancias implementadas de elementos de datos encriptables especificados y compararlo con la cantidad de instancias de datos que requieren facilidad de encriptación/desencriptación según la</i>	<p><b><math>X = A/B</math></b></p> <p><i>A: Número de instancias implementadas de datos encriptables según las</i></p>	<p><b><math>0 \leq X \leq 1</math></b></p> <p><i>A mayor cercanía al 1 resultará más completo.</i></p>

		especificación.	especificaciones.  B: Número de datos que requieren encriptamiento según las especificaciones.	
--	--	-----------------	--	--

**Subcaracterística: Conformidad con la Funcionalidad**

<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Conformidad uniforme en el sistema</b>	¿Cuán conforme es la funcionalidad del producto a regulaciones aplicables, estándares y convenciones?	Contar el número de puntos que requieren la conformidad que han sido encontrados y compare con el número de puntos que requieren conformidad según la especificación.	$X = A/B$  A: Número de interfaces correctamente implementadas confirmado en la revisión.  B: Número total de interfaces en conformidad.	$0 \leq X \leq 1$  A mayor cercanía al 1 resultará más adecuada.
<b>Conformidad funcional</b>			$X = A/B$  A: Número de ítems correctamente implementados en relación con la funcionalidad confirmada en la evaluación.  B: Total de ítems en conformidad.	$0 \leq X \leq 1$  A mayor cercanía al 1 resultará más adecuada.



**MÉTRICAS DE CONFIABILIDAD**

**Subcaracterística: Madurez**

<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Detección de fallos</b>	<i>¿Cuántos fallos fueron detectados en la revisión del producto?</i>	<i>Contar el número de fallos detectados en la revisión y compararlo con el número de fallos estimados para ser detectados en esta fase.</i>	$X = A/B$  <i>A: Número absoluto de fallos detectados en la revisión</i>  <i>B: Número de fallos estimados que se debían detectar en la revisión (usando la historia pasada o un modelo de referencia).</i>	$0 \leq X$  <i>Un mayor valor para X implica una mayor calidad del producto, mientras <math>A=0</math> no necesariamente implica un fallo en la iteración.</i>
<b>Corrección de fallos</b>	<i>¿Cuál es la proporción de fallos corregidos?</i>	<i>Contar el número de fallos corregidos durante el diseño/codificación y comparar con el número de fallos detectados en la revisión durante el diseño/codificación.</i>	$X = A$  $Y = A/B$  <i>A: Número de fallos corregidos en el diseño o el código.</i>  <i>B: Número de fallos detectados en la revisión (Es necesario convertir el valor de X al intervalo 0-1 si estamos</i>	$0 \leq X$  <i>Un mayor valor de X implica menos fallos prevalecen.</i>  $0 \leq Y \leq 1$  <i>Mientras más cercano a 1 mejor. (Más fallos se eliminan)</i>

			haciendo resumen de características).	
<b>Adecuación de pruebas</b>	¿Cuántos casos de pruebas requeridos están cubiertos por el plan de pruebas?	Contar el número de casos de pruebas planificados y comparar con el número de casos de pruebas requeridos para obtener una adecuada cobertura de pruebas.	$X = A/B$ A: Número de casos de prueba diseñados en el plan de pruebas y confirmados en la revisión  B: Número de casos de prueba requeridos.	$0 \leq X$ Donde X sea más grande será más adecuado.

**Subcaracterística: Tolerancia ante fallos**

<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Evitación de fallos</b>	¿Cuántos patrones de fallos fueron traídos bajo control para evitar fallas críticas y serias?	Contar el número de patrones de fallos evitados y comparar con el número de patrones de fallos a ser considerados.	$X = A/B$ A: Número de fallos.  B: Número de fallos totales.	$0 \leq X$ Donde X sea más grande es mayor la evitación de fallos.
<b>Evitación de operaciones incorrectas</b>	¿Cuántas funciones están implementadas con capacidad de evitación de operaciones incorrectas?	Cuenta el número de funciones implementadas para evitar fallas críticas y serias causadas por operaciones incorrectas y compárelas con el número de patrones de operaciones incorrectas a ser consideradas.  Notas: Ejemplos de patrones de operaciones	$X = A/B$ A: Número de funciones implementadas para evitar operaciones incorrectas.	$0 \leq X$ Donde X sea más grande es mayor la evitación operaciones incorrectas.

		<i>incorrectas: Tipos de datos como parámetros incorrectos, secuencia de entrada de datos incorrecta, incorrecta secuencia de operación.</i>	<i>B: Número de operaciones incorrectas para ser consideradas.</i>	
<b>Subcaracterística: Recuperabilidad</b>				
<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Recuperabilidad</b>	<i>¿Cuán capaz es el producto en restaurarse después de eventos o peticiones anormales?</i>	<i>Contar el número de requisitos de restauración implementados y comparar con el número de requisitos de restauración descritos en la especificación.</i>	$X = A/B$  <i>A: Número de requisitos de recuperación implementados confirmados en la revisión.</i>  <i>B: Número de requisitos de recuperación de la especificación.</i>	$0 \leq X \leq 1$  <i>Mientras mayor sea X mejor será la recuperabilidad.</i>
<b>Efectividad de la recuperación</b>	<i>¿Cuán efectiva es la capacidad de restauración?</i>	<i>Contar el número de requerimientos de recuperación implementados que cumplen con el plazo de tiempo previsto y comparar con el número de requerimientos de restauración requeridas en el plazo de tiempo especificado.</i>	$X = A/B$  <i>A: Número de requerimientos de recuperación implementados que cumplen con el tiempo objetivo de restauración.</i>  <i>B: Número de requerimientos de restauración con tiempos objetivos</i>	$0 \leq X \leq 1$  <i>Mientras mayor sea X mejor será la efectividad.</i>

			especificados.	
<b>Subcaracterística: Conformidad con la Confiabilidad</b>				
<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Conformidad con la confiabilidad</b>	¿Cuánto se adhiere el producto de software a las regulaciones aplicables, las normas, y convenciones, relacionadas con la confiabilidad?	Contar el número de elementos que se encontraron en conformidad, y comparar con el número de elementos que requieren estar en conformidad según la especificación.	$X = A/B$ A: Número de ítems correctamente implementados en relación con la confiabilidad confirmada en la evaluación. B: Total de ítems en conformidad.	$0 \leq X \leq 1$ A mayor cercanía al 1 resultará más adecuada.

**MÉTRICAS DE USABILIDAD**

**Subcaracterística: Comprensibilidad**

<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Compleitud de la descripción</b>	<i>¿Qué proporción de funciones (o tipos de funciones) son descritas en la descripción del producto?</i>	<i>Contar el número de funciones las cuales han sido descritas adecuadamente y comparar con el número de funciones en el producto.</i>	<p><b><math>X = A/B</math></b></p> <p><i>A: Número de funciones o tipos de funciones descritos en la descripción del producto.</i></p> <p><i>B: Número total de funciones o tipos de funciones. (Esto indica si los usuarios potenciales entenderán la capacidad del producto después de leer la descripción del mismo).</i></p>	<p><b><math>0 \leq X \leq 1</math></b></p> <p><i>Mientras más cercano a 1, más completo.</i></p>
<b>Capacidad de demostración</b>	<i>¿Qué proporción de funciones requieren capacidad de demostración?</i>	<i>Contar el número de funciones de adecuada demostración y comparar con el número de funciones que requieren una adecuada demostración.</i>	<p><b><math>X = A/B</math></b></p> <p><i>A: Número de funciones demostradas y confirmadas en la revisión</i></p>	<p><b><math>0 \leq X \leq 1</math></b></p> <p><i>Mientras más cercano a 1, más capaz.</i></p>



			<p><i>B: Número total de funciones que requieren capacidad de demostración.</i></p> <p><i>(Las demostraciones pasan por el proceso mostrando cómo se usa el software).</i></p>	
<b>Funciones evidentes</b>	<i>¿Qué proporción de funciones del producto son evidentes para el usuario?</i>	<i>Contar el número de funciones que son evidentes para el usuario y comparar con el número total de funciones.</i>	<p><b><math>X = A/B</math></b></p> <p><i>A: Número de funciones o tipos de funciones evidentes para el usuario.</i></p> <p><i>B: Número total de funciones o tipos de funciones.</i></p>	<p><b><math>0 \leq X \leq 1</math></b></p> <p><i>Mientras más cercano a 1, mejor.</i></p>
<b>Entendimiento de funciones</b>	<i>¿Qué proporción de las funciones del producto será el usuario capaz de comprender correctamente?</i>	<i>Contar el número de interfaces de usuario donde su propósito es el entendimiento del usuario y comparar con el número de funciones de interfaces de usuarios.</i>	<p><b><math>X = A/B</math></b></p> <p><i>A: Número de funciones de interfaz de usuario cuyo propósito es entendido por el usuario.</i></p> <p><i>B: Número total de funciones de interfaz de usuario.</i></p>	<p><b><math>0 \leq X \leq 1</math></b></p> <p><i>Mientras más cercano a 1, mejor.</i></p>
<b>Subcaracterística: Operabilidad</b>				

<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Chequeo de la Validez de la entrada</b>	¿Qué proporción de número de elementos de entrada chequean validez de los datos?	Contar el número de elementos de entrada, que chequean la validez de los datos y compara con el número de elementos de entrada, que podría verificar para datos válidos.	$X = A/B$  A: Número de elementos de entrada que chequean validez de los datos.  B: Número de elementos de entrada que pueden chequear la validez de los datos.	$0 \leq X \leq 1$  Mientras más cercano a 1, mejor.
<b>Cancelación de operaciones de usuarios</b>	¿Qué proporción de funciones pueden ser canceladas antes de la terminación?	Contar el número de funciones implementadas, que pueden ser canceladas por el usuario antes de la terminación y comparar con el número de funciones que requieren la capacidad de pre-cancelación.	$X = A/B$  A: Número de funciones implementadas que pueden ser canceladas por el usuario.  B: Número de funciones que requieren la capacidad de pre-cancelación.	$0 \leq X \leq 1$  Mientras más cercano a 1, mejor cancelación.
<b>Deshacer operaciones de usuario</b>	¿Qué proporción de funciones pueden ser deshechas?	Contar el número de funciones implementadas, que pueden ser deshechas por el usuario después de completadas y comparar con el número de funciones.	$X = A/B$  A: Número de funciones implementadas que pueden ser deshechas por el usuario.	$0 \leq X \leq 1$  Mientras más cercano a 1, mejor.

			<i>B: Número de funciones.</i>	
<b>Customización</b>	<i>¿Qué proporción de funciones pueden ser personalizadas durante la operación?</i>	<i>Contar el número de funciones implementadas, que pueden ser personalizadas por el usuario durante la operación y comparar con el número de funciones que requieren capacidad de personalización.</i>	$X = A/B$  <i>A: Número de funciones que pueden ser personalizadas durante su operación. B: Número de funciones que requieren capacidad de personalización.</i>	$0 \leq X \leq 1$  <i>Mientras más cercano a 1, mejor customización.</i>
<b>Accesibilidad física</b>	<i>¿Qué proporción de la función puede ser customizadas para el acceso por usuarios con desventajas físicas?</i>	<i>Contar el número de funciones implementadas, que pueden ser customizadas y comparar con el número de funciones.</i>	$X = A/B$  <i>A: Número de funciones que se pueden customizar. B: Número de funciones. (Ejemplos de accesibilidad física son la imposibilidad de usar un mouse, o la ceguera).</i>	$0 \leq X \leq 1$  <i>Mientras más cercano a 1, mejor accesibilidad física.</i>
<b>Capacidad de monitoreo de estado de operaciones</b>	<i>¿Qué proporción de funciones tienen capacidad de monitoreo de estado?</i>	<i>Contar el número de funciones implementadas, que presentan capacidad de monitoreo y comparar con el número de funciones que requieren capacidad de monitoreo.</i>	$X = A/B$  <i>A: Número de funciones que tienen capacidad de monitoreo de estado.  B: Número de funciones que deben tener capacidad de monitoreo de</i>	$X \leq 0 \leq 1$  <i>Mientras más cercano a 1, mejor capacidad de monitoreo.</i>

			<i>estado (el estado incluye monitoreo de progreso).</i>	
<b>Consistencia operacional</b>	<i>¿Qué proporción de operaciones se comporta la misma manera a operaciones semejantes en otras partes del sistema?</i>	<i>Contar el número de instancias de operaciones de comportamiento inconsistente y comparar con el número total de operaciones.</i>	$X = I - A/B$ <i>A: Número de instancias de operaciones con comportamiento inconsistente.</i>  <i>B: Número total de operaciones.</i>	$X \leq 0 \leq I$  <i>Mientras más cercano a 1, más consistente.</i>
<b>Claridad de mensaje</b>	<i>¿Qué proporción de mensajes es auto explicativa?</i>	<i>Contar el número de mensajes con una clara explicación y comparar con el número de mensajes implementados.</i>	$X = A/B$ <i>A: Número de mensajes implementados con explicaciones claras.</i>  <i>B: Número de mensajes implementados.</i>	$X \leq 0 \leq I$  <i>Mientras más cercano a 1, mejor.</i>
<b>Claridad de elementos de interfaz</b>	<i>¿Qué proporción de elementos de interfaz es auto explicativa?</i>	<i>Contar el número de elementos de interfaz que son auto explicativas y comparar con el número total de elementos de interfaz.</i>	$X = A/B$ <i>A: Número de elementos de interfaz que se explican por sí solos.</i>  <i>B: Número total de elementos de interfaz.</i>	$X \leq 0 \leq I$  <i>Mientras más cercano a 1, mejor.</i>
<b>Recuperación operacional de</b>	<i>¿Qué proporción de funciones</i>	<i>Contar el número de funciones implementadas</i>	$X = A/B$	$X \leq 0 \leq I$

<b>errores</b>	<i>pueden tolerar errores de usuarios?</i>	<i>con tolerancia de errores de usuarios y comparar con el número de funciones que requieren capacidad de tolerancia.</i>	<i>A: Número de funciones implementadas con tolerancia a errores de usuario. B: Número total de funciones que requieren capacidad de tolerancia.</i>	<i>Mientras más cercano a 1, mejor.</i>
----------------	--	---	--	---

**Subcaracterística: Atracción**

<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Interacción atractiva</b>	<i>¿Cuán atractiva es la Interfaz de usuario?</i>	<i>Cuestionario para usuarios.</i>	<i>Cuestionario para medir la atracción de la interfaces para los usuarios, tomando en cuenta atributos tales como color y diseño gráfico.  (Algunos temas que influyen potencialmente a esta atracción incluyen alineación de los elementos, agrupación, uso de colores, gráficos de tamaño</i>	<i>Clasificación de evaluación</i>

			<p><i>apropiado, uso de separadores, bordes y espacios en blanco, animaciones, tipografía e interfaz 3D).</i></p> <p><i>(Esto puede estar basado en prototipos de interfaz o en mocks).</i></p>	
<p><b>Capacidad de customización de la apariencia de la interfaz de usuario</b></p>	<p><i>¿Qué proporción de elementos de Interfaz pueden ser customizadas?</i></p>	<p><i>Inspección (por un experto)</i></p>	<p><b><math>X = A/B</math></b></p> <p><i>A: Número de tipos de elementos de interfaz que pueden ser customizados.</i></p> <p><i>B: Número total de tipos de elementos de interfaz.</i></p>	<p><b><math>0 \leq X \leq 1</math></b></p> <p><i>Mientras más cercano a 1 mejor</i></p>
<p><b>Subcaracterística: Conformidad con la usabilidad</b></p>				
<p><b>Nombre de la métrica</b></p>	<p><b>Propósito de la Métrica</b></p>	<p><b>Método de Aplicación</b></p>	<p><b>Fórmula</b></p>	<p><b>Interpretación del valor obtenido</b></p>
<p><b>Conformidad con la usabilidad</b></p>	<p><i>¿Cuán conforme es el producto a regulaciones aplicables, los estándares y las convenciones para el valor práctico?</i></p>	<p><i>Contar el número de elementos que requieren la conformidad que ha sido encontrado y comparar con el número de artículos que requieren la conformidad como en la especificación.</i></p>	<p><b><math>X = A/B</math></b></p> <p><i>A: Número de funciones implementadas relativa a la usabilidad confirmada en la evaluación.</i></p>	<p><b><math>0 \leq X \leq 1</math></b></p> <p><i>Mientras más cercano a 1 mejor</i></p>

			<i>B: Número de iteraciones en conformidad.</i>	
--	--	--	---	--

<b>MÉTRICAS DE EFICIENCIA</b>				
<b>Subcaracterística: Rendimiento</b>				
<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Tiempo de respuesta</b>	¿Cuánto tiempo toma completar una tarea en particular? ¿Cuánto toma antes de que el sistema responda a determinada operación?	Evaluar la eficacia del sistema operativo y de las llamadas del sistema a la aplicación. Estime el tiempo de reacción basado en esto.  //Comenzar una tarea específica. Mida el tiempo que esta toma en completar una muestra hasta que concluya la operación emprendida. Guardar los registros de cada intento.	$X = \text{Tiempo}$ (calculado o simulado)	Mientras más pequeño mejor.
<b>Rendimiento</b>	¿Cuál es el número estimado de tareas que pueden ser realizadas sobre una unidad de tiempo?	Evaluar la eficiencia de manejar recursos en el sistema. Haga un factor basado sobre las llamadas del uso al sistema en la manipulación de los recursos.	$X = \text{Número de tareas por unidad de tiempo}$	Mientras mayor mejor.

<b>Tiempo de ejecución de un grupo de tareas relacionadas</b>	¿Cuál es el tiempo estimado para completar un grupo de tareas relacionadas como porción de trabajo?	Evaluar la eficiencia de las llamadas al sistema operativo y a la aplicación. Basado en esto, estime el tiempo de reacción para completar un grupo de tareas relacionadas.	$X = \text{Tiempo}$ (calculado o simulado)	Mientras más pequeño mejor.
<b>Subcaracterística: Utilización de Recursos</b>				
<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Uso de entrada/salida</b>	¿Cuál es el tiempo estimado en la utilización de Entrada/Salida para completar tareas específicas?	Estime los requisitos de utilización de Entrada/Salida para la aplicación.	$X = \text{Número de buffers}$ (calculados o simulados).	Mientras más pequeño mejor.
<b>Densidad de mensajes de entrada/salida</b>	¿Cuál es la densidad de mensajes relacionados a la utilización de Entrada/Salida en líneas de código responsable en llamadas al sistema?	Cuente el número de errores pertenecientes a la Entrada/Salida y compárelo con el número de líneas de código relacionadas directamente con llamadas al sistema.	$X = A/B$ A: Número de mensajes de error relacionadas con la entrada/salida. B: Número de líneas de código relacionadas directamente con llamadas al sistema.	Mientras mayor, mejor.
<b>Uso de memoria</b>	¿Cuál es el tamaño de memoria estimado que el producto ocupará para completar una tarea específica?	Estimar los requerimientos de memoria.	$X = \text{Tamaño en bytes}$ (calculado o simulado).	Mientras menor sea, mejor.



<b>Densidad de mensajes de uso de memoria</b>	¿Cuál es la densidad de mensajes relacionados a la utilización de memoria en líneas de código responsable en llamadas al sistema?	Cuenta el número de mensajes de error relacionados con memoria y compárelo con el número de líneas de código directamente relacionado con llamadas al sistema.	$X = A/B$  A: Número de mensajes de error relacionados con memoria. B: Número de líneas de código directamente relacionado con llamadas al sistema.	Mientras mayor sea, mejor.
<b>Uso de la transmisión</b>	¿Cuál es la cantidad estimada de utilización de los recursos de la transmisión?	Estime los requisitos de la utilización de los recursos de la transmisión estimando los volúmenes de la transmisión.	$X = \text{bits/time}$ (calculado o simulado)	Mientras menor sea, mejor.

**Subcaracterística: Conformidad con la Eficiencia**

<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Conformidad con la Eficiencia</b>	¿Cuán conforme es la eficiencia del producto de software con las regulaciones, las normas y convenciones que les son aplicables?	Cuenta el número de elementos que se encontraron en conformidad y compárelo con el número de elementos que requieren estar en conformidad según la especificación.	$X=A/B$  A: Número de elementos implementados correctamente relacionados con la eficiencia confirmada en la evaluación.  B: Número total de elementos en conformidad.	$0 \leq X \leq 1$  Mientras más cercano a 1 mejor

**MÉTRICAS DE MANTENIBILIDAD**

**Subcaracterística: Diagnosticabilidad**

<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Registro de actividades</b>	¿Cómo se registran las actividades de estado del sistema?	Contar el número de elementos de datos que deben registrarse definidos en las especificaciones y comparar con el número de elementos que requieren ser registrados.	$X = A/B$ A: Número de elementos de registro de datos implementados según la especificación confirmada en la revisión. B: Número de elementos de datos que deben registrarse definidos en las especificaciones.	$0 \leq X \leq 1$ Mientras más cercano a 1, más datos proporciona para registrar el estado del sistema. Nota: Es necesario hacer un intervalo $<0,1>$ si se hace una suma de las características.
<b>Disponibilidad de función de diagnóstico</b>	¿Cómo es la provisión de las funciones diagnóstico?	¿Contar el número de funciones diagnósticos implementados según la especificación y comparar con el número de funciones diagnóstico requeridas en la especificación?	$X = A/B$ A: Número de funciones diagnóstico implementadas según la especificación y confirmadas en la revisión. B: Número de funciones de diagnóstico requeridas.	$0 \leq X$ Mientras más cercano a 1, mejor funciones diagnóstico implementadas.

### **Subcaracterística: Cambiabilidad**

<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<i>Capacidad de registro de cambios</i>	<i>¿Cambian las especificaciones y los registros de los módulos de programas adecuadamente con líneas de código comentadas?</i>	<i>Registrar información de los cambios entre los módulos.</i>	$X = A/B$  <i>A: Número de cambios en funciones o módulos que tienen comentario de cambios confirmados en la revisión.</i>  <i>B: Número total de funciones o módulos que han cambiado desde el código original.</i>	$0 \leq X \leq 1$  <i>Mientras más cercano a 1, mejor.</i>

### **Subcaracterística: Estabilidad**

<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Impacto del cambio</b>	<i>¿Qué es la frecuencia de impactos después de la modificación?</i>	<i>Contar el número de impactos adversos discernidos después de la modificación y comparar con el número de modificaciones realizadas.</i>	$X = 1 - A/B$  <i>A: Número de impactos adversos detectados después de las modificaciones.</i>  <i>B: Número de modificaciones hechas.</i>	$0 \leq X \leq 1$  <i>Mientras más cercano a 1, mejor.</i>
<b>Localización del impacto de modificación</b>	<i>Cuán grande es el impacto de la modificación en el producto</i>	<i>Contar el número de variables afectadas en la modificación y comparar con el número de variables en el producto.</i>	$X = A/B$  <i>A: Número de datos variables afectados por la modificación,</i>	$0 \leq X \leq 1$  <i>Mientras más cercano a 1, menor el impacto de</i>

<b>ón</b>	<p>software.</p> <p>a) Todas las variables en la instrucción que fue cambiada.</p> <p>b) Variable que está en la misma instrucción con la variable definida por a)</p>		<p>confirmados en la revisión.</p> <p>B: Número total de variables.</p>	<p>modificación.</p>
-----------	--	--	---	----------------------

**Subcaracterística: Contrastabilidad**

<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<b>Completitud de funciones nativas de prueba</b>	¿Cuán completa es la capacidad incorporada de la prueba?	Contar el número de funciones incorporadas de la prueba implementadas según la especificación y comparar con el número de funciones incorporadas de la prueba en los requerimientos.	$X = A/B$  A: Número de funciones nativas de pruebas implementadas según la especificación y confirmadas en la revisión.  B: Número de función de pruebas nativas requeridas.	$0 \leq X \leq 1$  Mientras más cercano a 1, más completo.
<b>Autonomía de prueba</b>	¿Cuán independiente puede el software ser probado?	Contar el número de dependencias con otros sistemas para probar que han sido simulados con stubs y contar con el número de dependencias de prueba con otros sistemas.	$X = A/B$  A: Número de dependencias con otros sistemas para probar que han sido simulado con stubs.  B: Número total de dependencias de	$0 \leq X \leq 1$  Mientras más cercano a 1, más completo.

			<i>prueba con otros sistemas.</i>	
<b>Obseabilidad del progreso de las pruebas</b>	<i>¿Cuán completo es el progreso en la prueba mostrado durante la misma?</i>	<i>Contar el número de puntos de chequeo implementados según la especificación y comparar con el número de puntos de chequeo requeridos por el diseño.</i>	<b><math>X = A/B</math></b> <i>A: Número de puntos de chequeo implementados según la especificación y confirmados en la revisión.</i> <i>B: Número de puntos de chequeo designados</i>	<b><math>0 \leq X \leq 1</math></b> <i>Mientras más cercano a 1, más completo.</i>
<b>Subcaracterística: Conformidad con la Mantenibilidad</b>				
<b>Nombre de la métrica</b>	<b>Propósito de la Métrica</b>	<b>Método de Aplicación</b>	<b>Fórmula</b>	<b>Interpretación del valor obtenido</b>
<i>Conformidad con la mantenibilidad</i>	<i>¿Cuán conforme es la mantenibilidad del producto a regulaciones aplicables, estándares y convenciones?</i>	<i>Contar el número de puntos que requieren la conformidad que han sido encontrados y compare con el número de puntos que requieren conformidad según la especificación.</i>	<b><math>X=A/B</math></b> <i>A: Número de ítems correctamente implementados en relación con la Mantenibilidad confirmada en la evaluación.</i> <i>B: Total de ítems en conformidad.</i>	<b><math>0 \leq X \leq 1</math></b> <i>Mientras más cercano a 1 mejor</i>

