

**Universidad de las Ciencias Informáticas**

**Facultad 6**



**Título: Procedimiento para la validación de los requisitos no funcionales en los Sistemas de Gestión de Información.**

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

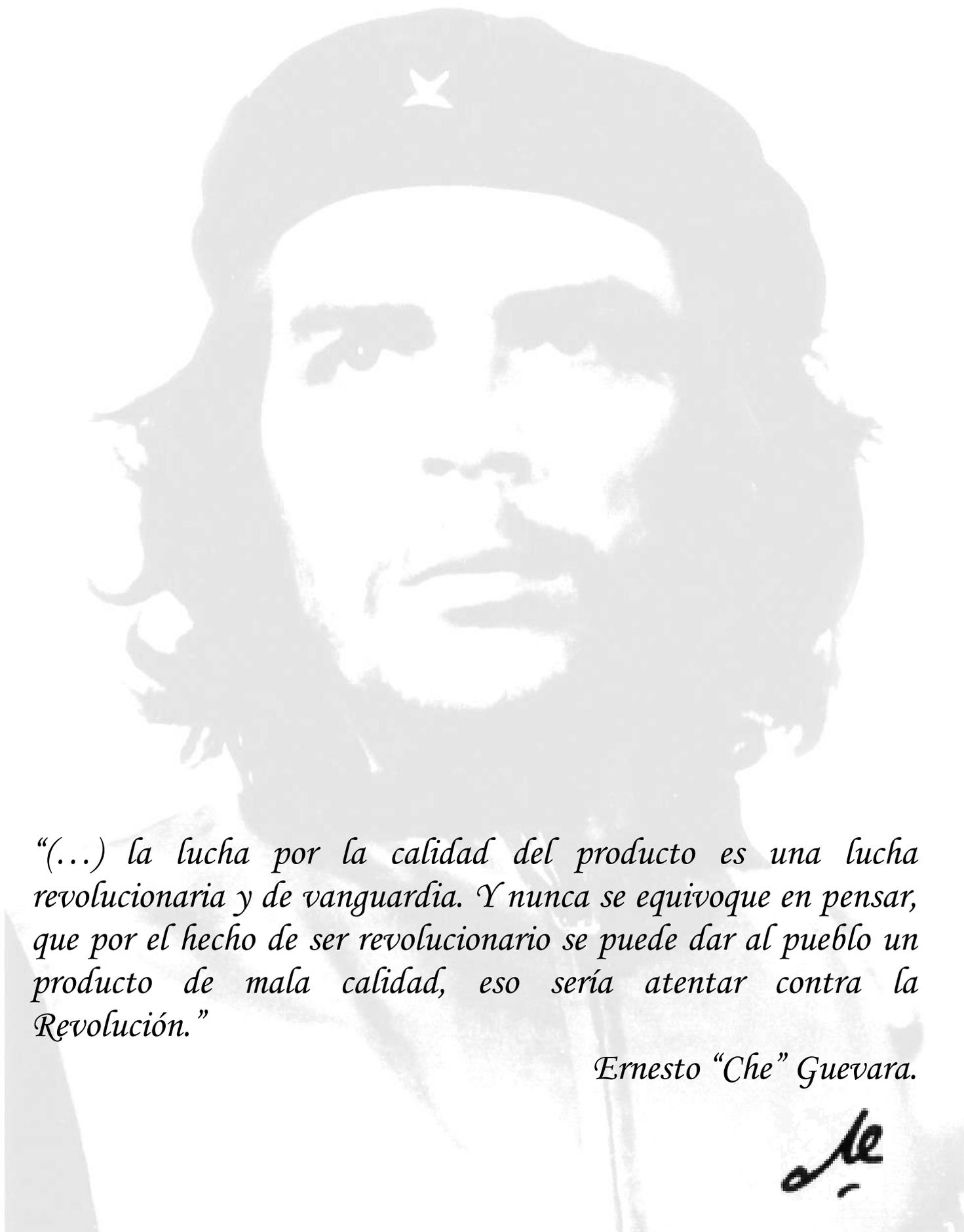
**Autor:**

Ibet Roblejo Rondón

**Tutor:** Ing. Niurka Martínez Durán

La Habana, junio del 2012

“Año 54 de la Revolución”



*“(...) la lucha por la calidad del producto es una lucha revolucionaria y de vanguardia. Y nunca se equivoque en pensar, que por el hecho de ser revolucionario se puede dar al pueblo un producto de mala calidad, eso sería atentar contra la Revolución.”*

*Ernesto “Che” Guevara.*

## **DECLARACIÓN DE AUDITORÍA**

Declaro ser autor de la presente tesis y reconozco 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 \_\_\_\_\_ del año \_\_\_\_\_.

---

Firma del Autor

**Ibet Roblejo Rondón**

---

Firma del Tutor

**Ing. Niurka Martínez Durán**

## **DATOS DE CONTACTO**

### **Tutor:**

Ing. Niurka Martínez Durán

Especialidad de graduación: Ingeniería en Ciencias Informáticas

Categoría docente: Instructor

Cargo: Asesor Técnico Docente

Área: Centro de Tecnologías de Gestión de Datos (DATEC)

Años de graduado: 4

Correo Electrónico: [nduran@uci.cu](mailto:nduran@uci.cu)

## AGRADECIMIENTOS

*A las dos personas más importantes en mi vida:*

*A mi madre por darme la vida cuando nadie me quería, por salvarme la vida cuando una cardiopatía quería arrebatármela y por hacerme la mujer que soy. Mi carrera, mi educación, mi independencia, mis sueños, mi manera de proyectarme en la vida y hasta mi carácter, te lo debo a ti mamá. Gracias por ser mi amiga, mi luz, mi fuerza, mi ejemplo... gracias por ser mi madre. Esta tesis y este título son tuyos.*

*A mi hermano, que para mí ha sido un padre, un amigo y mi apoyo incondicional. Siempre voy a seguir tus pasos en la vida, tú eres mi guía profesional, la luz que seguiré. Mi hermanito "querido", este título también es tuyo.*

*A mi papá por rectificar sus errores cometidos. Papi mi sueño era que estuvieses aquí y te lo pedí tantas veces y siempre me ponías "peros". Cuando te decidiste a venir un accidente se convirtió en ese "pero" sin remedios, gracias Dios estas vivo y eso es lo importante.*

*A mi abuela Mirta, abuelita cambiaría todo esto ahora, sin pensarlo dos veces, por un poquito de tu cariño. Siempre te dije que era una mujer de estudios y nunca reconociste mis logros, ni me apoyaste, ahora ya soy Ingeniera.*

*A mi hermanito chiquito, tía China, tío Nené, primo Emir, tía Margarita, tío Migue, tía Iraida, primo Robert, mi madrastra Anita, Oneida y tía Merci, por su amor y formar parte de mi vida.*

*A los amigos que con un "tu puedes", "dime como saliste" o un simple "hola" a través de un correo, estuvieron estos 5 años y me dieron fuerzas para levantarme y seguir adelante, gracias: Causa, Dami, "pima" Nancy, Gretel, Deglis, Nena, Zenia, Katy e Iri.*

*A amigos como Dunia, Yadiris, Carlí, Pupo, Alfredo, Layda, Nilo, Yosmany, Gema, Yeneisis, Yadirra, Lincon, Dorisley, Lili, Rafael (mello), Daimel, Héctor, Reynaldo, Ramón, Enrique, César, Lorenzo, Tahimí, Raúl (chuchi), Yasnelis, Ricardito, Ismel, Dalied, Manuel Labrada y Rajiv.*

*A los profes Garnache, Ponce, Alexei, Richar, Glennis, José Santiesteban, Cecilia, Dayris, Loenardo y Anthony.*

*A mi tutora Niurka por su paciencia, su tiempo y sus locuras que me hicieron reír tanto.*

*A Betty y Tahimé por su amistad en estos últimos momentos de la universidad. Ustedes dos son esa parte positiva que siempre se saca cuando se pasa por malos momentos. Su amistad sincera y transparente me ha permitido cambiar lágrimas por risas, pesimismo por fuerzas, ahogo por oportunidades, cansancio por ojos abiertos, nerviosismos por "tu puedes", caminos cerrados por decisiones, aburrimiento por música y canto, soledad por AMISTAD.*

*A "pima" Nancy, Aliet y Orisel por su ayuda incondicional.*

*A mi abuelo Enrique, que aunque no lo conocí sé que espiritualmente siempre me acompaña y alumbra mi mente.*

*A Dios.*

## DEDICATORIA

*A mi mamita Irene y mi hermanito "querido" Chuchi.*



## RESUMEN

El Centro de Tecnologías de Gestión de Datos (DATEC) requiere una validación de requisitos no funcionales para el correcto desarrollo de los Sistemas de Gestión de Información. Esta actividad es realizada por el Centro de Calidad para Soluciones Tecnológicas (CALISOFT), pero no desde las fases iniciales, sino una vez implementado el sistema. Provocando que los errores cometidos en la especificación de requisitos sean arrastrados hasta el final, con repercusión en cada etapa de desarrollo del software. La presente investigación tiene como objetivo diseñar un procedimiento para la validación de los requisitos no funcionales de los Sistemas de Gestión de Información desarrollados en el centro DATEC. Para ello se realizó un análisis de la Ingeniería de Requisitos, específicamente la etapa de validación. Se identificaron las tres fases que componen el diseño del procedimiento, cada una de las cuales contienen: actividades, responsables, participantes, artefactos de entrada y salida y técnica a utilizar. Se propuso un conjunto métricas para evaluar la calidad de los requisitos no funcionales, teniendo como guía el estándar ISO/IEC 9126. Finalmente se obtiene como resultado un procedimiento para la validación de requisitos no funcionales, que permite la detección y corrección de errores en los mismos desde la etapa inicial. La propuesta es aplicada a la Solución Integral para la Gestión de Proyectos y Acciones Centralizadas (SIGEPAC). Para su validación se empleó el método Delphi, a través del cual se utilizaron como elementos de certificación los criterios aportados por un panel de expertos en el tema.

### Palabras claves:

Artefactos, Ingeniería de Requisitos, métricas, requisitos no funcionales, responsables, técnicas de validación y validación.

**Índice**

<b>DATOS DE CONTACTO</b> .....	<b>I</b>
<b>AGRADECIMIENTOS</b> .....	<b>I</b>
<b>DEDICATORIA</b> .....	<b>II</b>
<b>RESUMEN</b> .....	<b>III</b>
<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>Capítulo 1 Fundamento Teórico</b> .....	<b>4</b>
Introducción .....	4
1.1 Requisito o requerimiento .....	4
1.1.2 Clasificación de los requisitos de software .....	4
1.2 Requisitos no funcionales (RNF).....	5
1.2.1 Importancia de los requisitos no funcionales .....	5
1.2.2 Clasificación de los requisitos no funcionales.....	5
1.3 Atributos de calidad asociados a los requisitos no funcionales.....	9
1.4 Ingeniería de Requisitos (IR).....	10
1.5 Validación de requisitos. Particularidades .....	11
1.5.1 Diferencia entre verificación y validación de requisitos de software.....	11
1.5.2 Métodos de validación de requisitos de software.....	12
1.5.3 Técnicas de validación .....	13
1.6 Concepto de métrica .....	14
1.6.1 Métricas de Software.....	14
1.6.3 Métricas de Calidad.....	15
1.7 Modelos de calidad de software .....	15
1.8 Sistemas de Gestión de Información.....	16
1.8.1 Validación de los Sistemas de Gestión de Información.....	17
1.9 Definición de procedimiento .....	18
1.9.1 Diferencias entre procedimiento y proceso.....	18
1.10 Herramientas de modelado .....	19
1.10.1 Rational Rose.....	19
1.10.2 Visual Paradigm .....	19
1.11 Métodos de prospectiva.....	20
1.11.1 Características del Método Delphi .....	20
1.11.2 Ventajas y desventajas del método de expertos .....	21
Conclusiones del capítulo .....	22

<b>Capítulo 2 Descripción de la propuesta</b> .....	<b>23</b>
Introducción .....	23
2.1 Características principales del procedimiento.....	23
2.1.1 Objetivo .....	23
2.1.2 Alcance.....	23
2.1.3 Estructura .....	23
2.1.4 Responsables y participantes .....	25
2.1.5 Artefactos de entrada .....	26
2.1.6 Artefactos de Salida .....	27
2.2 Fase I. Validación del Listado de Requisitos No Funcionales .....	29
2.2.1 Actividad 1: Especificar por escrito en el Listado de Requisitos No Funcionales todos los RNF de la aplicación .....	30
2.2.2. Actividad 2: Verificar el enunciado de los requisitos no funcionales .....	30
2.3 Fase II. Validación de las especificaciones de los RNF .....	31
2.3.1 Actividad 1: Verificar que la especificación de los RNF sea correcta .....	32
2.3.2 Actividad 2: Verificar que la especificación de los RNF sea no ambigua.....	33
2.3.3 Actividad 3: Verificar que la especificación de los RNF sea completa .....	33
2.3.4 Actividad 4: Verificar que la especificación de los RNF sea probable .....	34
2.3.5 Actividad 5: Verificar que la especificación de los RNF sea consistente .....	35
2.3.6 Actividad 6: Verificar que la especificación de los RNF sea modificable .....	35
2.4 Fase III. Validación por Métricas de Calidad.....	36
2.4.1 Actividad 1: Validación de los requisitos no funcionales asociados a la funcionalidad .....	37
2.4.2 Actividad 2: Validación de requisitos no funcionales relacionados con la confiabilidad .....	40
2.4.3 Actividad 3: Validación de los requisitos no funcionales relacionados con la usabilidad .....	43
2.4.4 Actividad 4: Validación de los requisitos no funcionales asociados a la eficiencia .....	45
2.4.5 Actividad 5: Validación de los requisitos no funcionales asociados a la mantenibilidad .....	47
2.4.6 Actividad 6: Validación de los requisitos no funcionales asociados a la portabilidad .....	48
2.5 Representación de los resultados de los requisitos no funcionales .....	50
Conclusiones del capítulo .....	51
<b>Capítulo 3 Validación del procedimiento.....</b>	<b>52</b>
Introducción .....	52
3.1 Aplicación del procedimiento .....	52
3.1.1 Aplicación de la Fase I.....	52
3.1.2 Aplicación de la Fase II.....	53
3.1.3 Aplicación de la Fase III.....	55

3.2 Validación de la propuesta mediante el método Delphi.....	57
Conclusiones del capítulo .....	63
<b>CONCLUSIONES GENERALES .....</b>	<b>64</b>
<b>RECOMENDACIONES.....</b>	<b>65</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>66</b>
<b>BIBLIOGRAFÍA.....</b>	<b>69</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>72</b>

## INTRODUCCIÓN

El desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) ha alcanzado un auge mundial sin precedentes, trayendo a su vez un avance prolífico en la industria del software. Uno de los problemas significativos en la esfera de la informática es la Gestión de la Calidad, por las innovaciones tecnológicas que han llevado a que el tamaño y la complejidad de los sistemas informáticos aumenten vertiginosamente. Desde su surgimiento, este tema ha sido motivo de preocupación para especialistas, ingenieros, investigadores y comercializadores de esta rama, los cuales han realizado estudios encaminados alrededor de dos objetivos fundamentales: primero obtener un software con calidad y segundo evaluar la calidad del software.

Según el ingeniero de software, Roger Pressman, “La calidad es definida como una característica o atributo de algo. Como atributo de un elemento, la calidad se refiere a las características mensurables, es decir: cosas que se pueden comparar para conocer estándares”. (1) La existencia de esta en los productos se ha convertido en una necesidad conveniente para permanecer en el mercado, es por ello que la industria del software realiza grandes esfuerzos en este tema.

En medio de este panorama competitivo surgen los Sistemas de Gestión de Información, aplicaciones informáticas diseñadas especialmente para la gestión y mejora continua de las políticas, los procedimientos y procesos de la organización. El Centro de Tecnologías de Gestión de Datos (DATEC), enmarcado en la creación de bienes y servicios informáticos, produce dichas aplicaciones siguiendo esta línea. Uno de los aspectos primordiales para el funcionamiento correcto de esta variante de desarrollo de software, es determinar y validar los requisitos no funcionales.

“Los requisitos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este, como son la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento”. (2) Surgen de las necesidades del usuario, debido a las restricciones en el presupuesto, las políticas de la organización, la necesidad de interoperabilidad con otros sistemas de software o hardware, o por factores externos como regulaciones de seguridad o legislaciones sobre privacidad.

En este escenario una validación correcta aumenta sistemáticamente las expectativas de los usuarios finales. Su ejecución en general es difícil, debido a que suelen aparecer defectos típicos como: contradicciones en la especificación, diferencias pequeñas entre los requerimientos funcionales y no funcionales, especificaciones poco comprensibles, redundantes, que entran en conflicto y que no indican todos los recursos necesarios del hardware. El costo de validación es muy alto y los clientes que pagan el sistema piensan a veces que estos no son justificados.

En la universidad la validación de los productos de software se realiza por el Centro de Calidad para Soluciones Tecnológicas (CALISOFT) y los grupos de calidad pertenecientes a cada centro de

desarrollo. La estrategia seguida es realizar esta actividad al final de la implementación del sistema, mediante la comprobación de que los requisitos no funcionales iniciales estén implementados correctamente, según lo descrito en el documento de Especificación de Requisitos de Software (ERS). Muchos de los errores en estos requisitos están dados por una especificación inadecuada que solo se puede evidenciar cuando ya el producto se encuentra en su fase final, provocando retrasos, aumento de los costos e insatisfacción del cliente. La validación al final permite detectar las deficiencias con la ejecución de la implementación del sistema, pero no realizarla de forma inicial provoca arrastrar errores y omisiones de los requisitos no funcionales, que aumentan su repercusión con el paulatino desarrollo del software.

En un Sistema de Gestión de Información que en su desarrollo no se lleve a cabo una adecuada validación de los requisitos no funcionales, desde fases iniciales, suele tener problemas en la eficiencia, portabilidad, despliegue, facilidad de uso, robustez, reutilización y compatibilidad con otros sistemas. Provoca que en la organización que lo utiliza, se haga engorroso renovar constantemente sus objetivos, estrategias, operaciones y niveles de servicio.

Por todo lo anteriormente mencionado se identifica como **problema de la investigación**: Las deficiencias en la validación de los requisitos no funcionales dificultan el correcto funcionamiento de los Sistemas de Gestión de Información; teniendo como **objeto de estudio**: Proceso de validación de software; y enmarcando la investigación en el **campo de acción**: Proceso de validación en la Ingeniería de Requisitos para los Sistemas de Gestión de Información.

Dada la problemática planteada se propone como **objetivo general**: Diseñar un procedimiento para la validación de los requisitos no funcionales de los Sistemas de Gestión de Información desarrollados en el Centro de Tecnologías de Gestión de Datos (DATEC).

Para dar cumplimiento al objetivo general se trazaron los siguientes **objetivos específicos**:

1. Analizar la bibliografía de las normas y estándares de calidad vinculados a la validación de requisitos para los Sistemas de Gestión de Información.
2. Diseñar un procedimiento para la validación de los requisitos no funcionales de los Sistemas de Gestión de Información.
3. Validar el procedimiento diseñado.

Para dar cumplimiento a los objetivos se desglosan las siguientes **tareas de la investigación**:

1. Análisis bibliográficos relacionados a los Sistemas de Gestión de Información.
2. Análisis bibliográficos relacionados a las normas y estándares de calidad vinculados a la validación de requisitos.
3. Selección de los requisitos no funcionales que se validan con el procedimiento.
4. Identificación de las fases y actividades del procedimiento.
5. Descripción de las fases y actividades del procedimiento.

6. Aplicación del método de expertos (método Delphi) para la validación de la propuesta.
7. Aplicación del procedimiento a un Sistema de Gestión de Información.

El presente trabajo de diploma se organizó en tres capítulos:

**Capítulo 1: Fundamento Teórico.** Se encuentra todo el contenido relacionado con los conceptos fundamentales, haciendo mayor énfasis en el estudio de los requisitos no funcionales. Se caracteriza los Sistemas de Gestión de Información, las métricas de software, la Ingeniería de Requisitos, destacándose la validación de requisitos de software como principal objetivo de esta investigación, y algunos aspectos importantes relacionados con las diferentes técnicas de validación de requisitos utilizadas en el ámbito mundial.

**Capítulo 2: Descripción de la propuesta.** Se propone el diseño del procedimiento a ser utilizado, explicando detalladamente cada una de las fases, actividades, técnicas y los artefactos de entrada y salida generados en cada actividad. Destacándose el propósito, alcance y características principales.

**Capítulo 3: Validación del procedimiento.** Se realiza la validación técnica del procedimiento a través del método de expertos (método Delphi). Es aplicada la propuesta a la Solución Integral para la Gestión de Proyectos y Acciones Centralizadas (SIGEPAC).

## Capítulo 1 Fundamento Teórico

### Introducción

En el presente capítulo se realiza el marco teórico de la investigación, incluyendo además el marco conceptual sobre temas como requisitos no funcionales, la Ingeniería de Requisitos, calidad de software, procedimiento, métricas de software y Sistemas de Gestión de Información; centrándose en sus características e importancia. Se enfatiza en las etapas de la Ingeniería de Requisitos, haciendo mayor hincapié en la validación de requisitos de software y los estándares que se utilizan para el desarrollo de este trabajo de diploma. Se estudian los diferentes métodos de validación existentes y las técnicas que permiten su ejecución.

### 1.1 Requisito o requerimiento

En el glosario de la IEEE<sup>1</sup> se ha definido diferentes conceptos sobre el término “Requisito”:

- i) “Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo”.
- ii) “Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal”.
- iii) “Una representación documental de una condición o capacidad como en i o en ii.” (3)

Interpretando las definiciones anteriores se puede especificar que, un requerimiento es una descripción de una condición o capacidad que debe cumplir un sistema. Puede ser derivado de una necesidad de usuario identificada, o bien, estipulada en un contrato, estándar, especificación u otro documento formalmente impuesto al inicio del proceso.

Cuando se especifican los requisitos frecuentemente se encuentran un conjunto de dificultades que unidas a las características que deben tener hacen más difícil su extracción, documentación, implementación y por ende, su validación. Nunca significan lo mismo, algunos son más riesgosos, importantes o estables que otros. Se relacionan unos con otros, y a su vez, se relacionan con otros artefactos del proceso de desarrollo. Cada uno tiene propiedades únicas, abarcan áreas funcionales específicas y pueden cambiar a lo largo del ciclo de vida del software.

### 1.1.2 Clasificación de los requisitos de software

Existen dos tipos de requisitos, los requisitos de usuario: son descripciones ya sea en lenguaje natural o diagramas de lo que se espera que el sistema posea, así como las restricciones que podrá presentar el mismo; y los requisitos del sistema: establecen con más detalle las funciones, servicios y restricciones operativas que el sistema tendrá. Estos últimos se dividen en dos partes: requisitos funcionales y requisitos no funcionales. Dentro del sistema se encuentran también los requisitos del

---

<sup>1</sup> IEEE: Institute of Electrical and Electronics Engineers

dominio, los cuales surgen del dominio de la aplicación del sistema y no de las necesidades del usuario; se suelen tratar como otros requisitos funcionales o no funcionales, pues reflejan características y restricciones del dominio de la aplicación. (4)

Los requisitos funcionales (RF) describen lo que el sistema debe hacer. (5) Esta funcionalidad es la capacidad proporcionada y utilizable por uno o más componentes de un sistema, en ocasiones se les llaman conductuales u operacionales, debido a que estos especifican las entradas (estímulos) al sistema. Los requisitos no funcionales (RNF) son propiedades o cualidades que el producto debe tener. (5) Estos últimos serán objeto de análisis en este trabajo de diploma, lo cual se abordará en el epígrafe que aparece a continuación.

## **1.2 Requisitos no funcionales (RNF)**

Son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este, como son la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, define las restricciones del sistema como la capacidad de los dispositivos de entrada/salida (E/S) y las representaciones de datos que se utilizan en las interfaces del sistema. (2)

Forman una parte importante en el logro de la calidad de los productos de software, debido a que las propiedades no funcionales como cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

### **1.2.1 Importancia de los requisitos no funcionales**

La implementación de los requisitos no funcionales contribuye a un correcto desarrollo del software, sin importar la metodología que se utiliza para obtener el producto. Juegan un papel elemental, pues así como los requisitos funcionales tributan a funciones futuras que tendrá la aplicación informática, estos enmarcan el entorno donde se desenvolverá, porque proveen las condiciones que deberá tener la misma para su uso efectivo. Permiten una constante interacción con la propuesta de la arquitectura del sistema, pues determinan sus restricciones y usos. En el ciclo de vida del software un correcto desarrollo de los RNF contribuye a la obtención de una adecuada calidad del mismo, pues como se abordará en el siguiente epígrafe, los atributos de calidad enmarcan propiedades que el sistema debe cumplir.

### **1.2.2 Clasificación de los requisitos no funcionales**

Se hace complicado determinar una clasificación estándar para los RNF, pues existen diversas categorías determinadas, en ocasiones, por el equipo de trabajo que se enfrenta a resolver los mismos, incluso suele encontrarse el término “tipo de requisito no funcional”. Entre estas clasificaciones la correspondiente a los atributos de calidad de software es comúnmente utilizada en

las ciencias informáticas. Esta establece dos grupos: el grupo de los atributos importantes para los usuarios y el grupo de los atributos que son importantes para los desarrolladores.

Karl E. Wiegers<sup>2</sup> presenta una completa descripción de estos atributos, donde destaca que los importantes para los usuarios son: (6)

**Disponibilidad** (*Availability*): Es una medida de tiempo planeado durante el uso del software, en el cual el sistema es disponible y completamente operacional, así como la manera en que podrá recuperarse ante fallos encontrados. Disponibilidad abarca también confiabilidad, mantenibilidad e integridad.

**Eficiencia** (*Efficiency*): Es la medida en que el sistema aprovecha correctamente los recursos que presenta, como la capacidad del procesador, espacio en disco y memoria, ancho de banda, etc. Si el sistema no distribuye de manera eficiente estos recursos el funcionamiento no será el esperado por el cliente. Está relacionada con el funcionamiento (*performance*).

**Flexibilidad** (*Flexibility*): También conocida como extensibilidad y expansibilidad, es la medida de cómo se puede hacer variaciones en el producto final, o sea, de cómo el sistema se adaptará a los nuevos cambios funcionales, cambios que se pueden contemplar en el diseño del software. Este atributo se manifiesta principalmente en productos de muchas iteraciones e incrementos en las funcionalidades.

**Integridad** (*Integrity*): Relacionado con la seguridad del producto, se encarga de bloquear el acceso al sistema del personal no-autorizado, tanto a personas como virus; además de mantener la integridad de la información manejada por los usuarios del sistema. Es de vital importancia en el software que se encuentra soportado sobre una aplicación web.

**Interoperabilidad** (*Interoperability*): Indica cómo el sistema puede intercambiar información o servicios con otros sistemas, datos relacionados y semejantes. Es la capacidad que tendrá el software de migrar a otro sistema.

**Confiabilidad** (*Reliability*): Es la probabilidad de que el software se ejecute por un período de tiempo específico, sin fallas. Incluye el por ciento de operaciones que se realizan correctamente antes de que el sistema pueda fallar. La robustez es considerada en ocasiones como un aspecto de la confiabilidad.

**Robustez** (*Robustness*): Es el grado en que el sistema continúa funcionando ante fallas inesperadas, ya sean de componentes de hardware o software, o ante entradas inválidas de datos. Es la posibilidad de recuperación de datos que presentará el sistema ante los errores de usuarios o fallas de otras índoles.

---

<sup>2</sup> Consultor Principal del Proceso de Impacto en Portland, Oregón. EEUU. Ha llevado a las actividades de mejora de la calidad en pequeños grupos de desarrollo de aplicaciones. Forma parte del Consejo Editorial de IEEE Software, y como editor colaborador de la revista de Desarrollo de Software.

**Usabilidad** (*Usability*): El software se dice usable o útil, cuando es fácil de manejar por los usuarios sin necesidad de una completa preparación para enfrentarse al sistema. Es la ayuda que le brindará el sistema a los usuarios durante su uso, dirigida hacia el esfuerzo que hará el usuario a la hora de entrar datos y manejar las salidas.

Entre los más importantes para los desarrolladores se encuentran: (6)

**Mantenibilidad** (*Maintainability*): Indica qué tan claro puede ser modificar o corregir algún defecto en el software, representa cómo el software puede ser entendido, cambiado o probado de manera fácil. Este atributo está muy relacionado con la flexibilidad y comprobabilidad (*testability*).

**Portabilidad** (*Portability*): Es el esfuerzo requerido a la hora de migrar una parte del software de un ambiente de trabajo a otro, o la habilidad de vincular o localizar un producto con las características específicas.

**Reusabilidad** (*Reusability*): Reusabilidad o reutilidad, indica la conversión de un componente de software con funcionalidades determinadas a otros usos, es decir, a otras funciones.

**Comprobabilidad** (*Testability*): También conocido como verificabilidad, representa en qué medida los componentes del software o el producto como tal, pueden ser probados con el objetivo de buscarle defectos al funcionamiento del mismo.

El grupo de desarrolladores colombianos “Informática Siglo 21<sup>3</sup>” define las siguientes clasificaciones: (7)

**Rendimiento:** Está relacionado con tiempos de respuesta estimados, requeridos y esperados para la ejecución en línea de procesos del sistema, teniendo como base la plataforma tecnológica y escenarios específicos a los que en teoría el sistema estará expuesto y frente a los que deberá responder.

**Fiabilidad:** Está interrelacionada con la capacidad del usuario para confiar en las respuestas del sistema en un sentido técnico, es decir, que la funcionalidad del sistema no se vea afectada por factores ajenos al mismo.

**Disponibilidad:** Se relaciona con la capacidad del sistema para estar disponible para los usuarios, esto se refleja en el tiempo estimado, esperado y requerido por el usuario para que el sistema esté disponible.

---

<sup>3</sup> Compañía sólida y confiable en suministro de servicios de sistemas de información con 19 años de experiencia continua y exitosa en ejecución de proyectos de Outsourcing en tecnología, desarrollo de software a la medida y consultoría de sistemas de información en destacadas empresas. En la actualidad, cuenta con equipo de trabajo está conformado por expertos en diversas herramientas y sectores de la economía nacional, con amplia trayectoria en el desarrollo de sistemas de información, gerencia y control de proyectos.

# Capítulo 1: Fundamento Teórico

---

**Seguridad:** Requisitos relacionados con la confidencialidad de los datos en la transmisión y en el almacenamiento, junto con las necesidades del sistema para evitar intrusiones no-autorizadas al mismo y la capacidad para seguir eventos que comprometan esta seguridad a través del tiempo.

**Portabilidad:** Describe la capacidad del sistema para migrar de una plataforma hardware a otra sin que esto represente mayores traumatismos para el cliente del mismo.

**Mantenibilidad:** Relacionados con la capacidad para realizar revisiones y cambios sobre la funcionalidad del sistema de manera que no represente una exagerada inversión en recursos para el desarrollo del cambio mencionado. Estos requisitos están orientados a consideraciones de diseño, de codificación y al uso de modelos de desarrollo, para lograr que el mantenimiento de sistema sea lo más natural posible.

**Escalabilidad:** Hace referencia a la capacidad del sistema para acoplar módulos, componentes y extensiones, sugiere el empleo de tendencias en tecnologías que puedan permitir al sistema tener vigencia en su diseño por lo menos en un período de tiempo considerable, de tal manera que los componentes de extensión que se propongan puedan ser desarrollados basados en tecnologías conocidas.

**Reusabilidad:** Considera la capacidad de los componentes del sistema de prestar servicios a otros sistemas, de tal manera que el componente como valor adicional al cumplimiento de sus funciones pueda prestar sus servicios a otros sistemas sin que esto implique modificaciones o redefiniciones en dicho componente.

**Interfaz:** Contempla las características del sistema respecto a intercomunicación con otros sistemas, ya sea a través de servicios o de salidas en archivo, define los formatos y tecnologías para la exposición o la captura de información desde fuentes externas.

**Amigabilidad** (usabilidad): Requisitos que determinan las características generales de la capa de presentación del sistema, en cuanto a las características de diseño gráfico de la misma, además de las facilidades para su uso por parte del usuario final.

**Capacidad** (consumo de recursos): Considera los requisitos de tecnología que permite al sistema ejecutar sus funciones de manera que responda a las expectativas del usuario en términos de eficiencia en la respuesta de las operaciones. Se tienen en cuenta también las características, que hacen que sea posible que el sistema funcione de manera estable durante el tiempo planeado para el crecimiento del mismo.

En la presente investigación son utilizadas, de las clasificaciones anteriores, las que se relacionan con los atributos de calidad establecidos por el estándar ISO/IEC 9126, es decir, con las seis características (incluyendo las subcaracterísticas que las componen) que están vinculadas con los requisitos no funcionales. Este tema es tratado en el siguiente epígrafe.

## 1.3 Atributos de calidad asociados a los requisitos no funcionales

Según estándar ISO/IEC 9126 (ver epígrafe 1.7), la calidad del producto de software se debe desglosar jerárquicamente en un modelo compuesto de características y subcaracterísticas. Dicho modelo relaciona requisitos funcionales y no funcionales por medio de los seis atributos de calidad que propone (funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad) y por las subcaracterísticas que los categoriza, como se muestra en la siguiente figura:

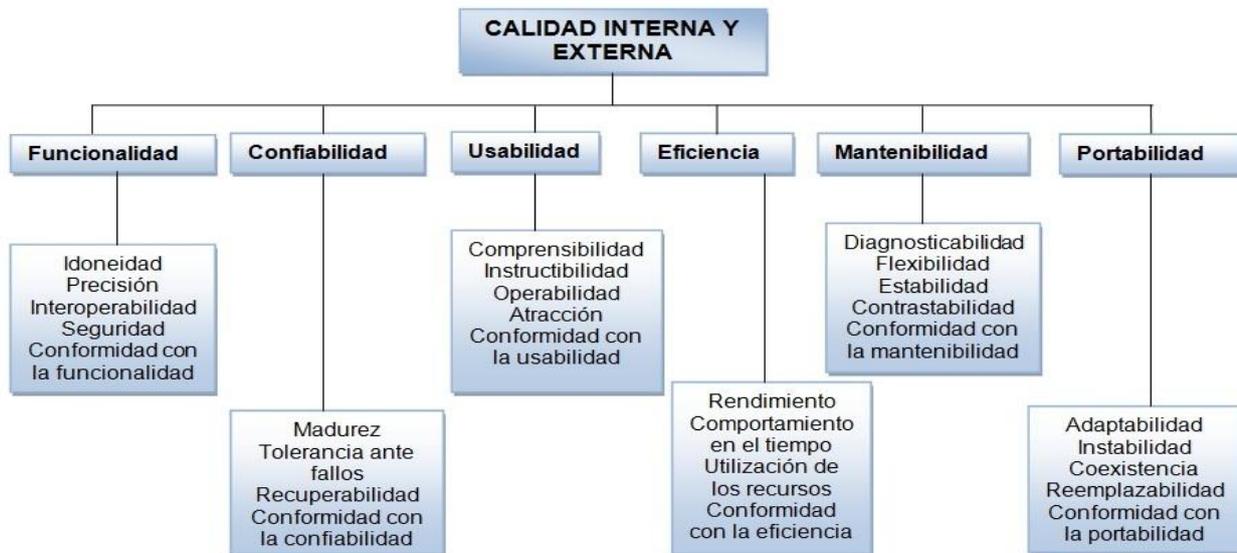


Figura 1 Características y subcaracterísticas del modelo de calidad interna y externa

En este trabajo de diploma serán utilizadas aquellas características y subcaracterísticas que coincidan con las clasificaciones de los RNF expuestas en el epígrafe anterior, que a continuación se relacionan:

### Funcionalidad

- **Seguridad:** Capacidad del producto de software para proteger los sistemas o datos de modo que las personas o los sistemas no-autorizados no puedan leerlos o modificarlos, y las personas o sistemas autorizados tengan el acceso a ellos.
- **Interoperabilidad:** Capacidad del producto de software para interactuar recíprocamente con uno o más sistemas especificados.

### Confiabilidad

- **Madurez:** Capacidad del producto de software de evitar un fallo total como resultado de haberse producido un fallo del software.
- **Recuperabilidad:** Capacidad del producto de software de restablecer un nivel de ejecución especificado y recuperar los datos directamente afectados en caso de fallo total.

### Usabilidad

- **Comprensibilidad:** Capacidad del producto de software para permitirle al usuario entender si el software es idóneo y cómo puede usarse para las tareas y condiciones de uso particulares.

- **Conformidad con la usabilidad:** Capacidad del producto de software para adherirse a las normas, convenciones, guías de estilo o regulaciones relativas a la usabilidad.

## **Eficiencia**

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

## **Mantenibilidad**

- **Diagnosticabilidad:** Capacidad del producto del software de ser objeto de un diagnóstico para detectar deficiencias o causas de los fallos totales en el mismo o para identificar las partes que van a ser modificadas.

## **Portabilidad**

- **Instabilidad:** Capacidad del producto de software de ser instalado en un ambiente especificado.
- **Conformidad con la portabilidad:** Capacidad del producto de software de adherirse a las normas o convenciones relativas a la portabilidad. (8)

## **1.4 Ingeniería de Requisitos (IR)**

La validación constituye una de las etapas que se realizan en la Ingeniería de Requisitos, por su importancia para este trabajo de diploma será analizada esta terminología en aras de una mejor comprensión en cuanto a los objetivos que se quieren lograr.

La Ingeniería de Requisitos es primordial en el proceso de producción de software, ya que se enfoca en un área fundamental: la definición de lo que se desea producir. Según Roger Pressman: “La Ingeniería de Requisitos es el uso sistemático de procedimientos, técnicas, lenguajes y herramientas para obtener con un coste reducido: análisis, documentación, evolución continua de las necesidades del usuario y especificación del comportamiento externo que satisfagan las necesidades del usuario”. (9)

La IR permite gestionar las necesidades del proyecto en forma estructurada: cada actividad consiste de una serie de pasos organizados y bien definidos. Mejora la capacidad de predecir el cronograma del proyecto, así como sus resultados: proporciona un punto de partida para controles subsecuentes y actividades de mantenimiento. Contribuye a la calidad del software con el cumplimiento de un conjunto de requerimientos (los requerimientos mencionados en el epígrafe anterior). Acrecienta la comunicación del equipo, ya que la especificación de requisitos representa una forma de consenso entre clientes y desarrolladores; si este no ocurre, el proyecto no será exitoso. Evita rechazos de

usuarios finales, ya que obliga al cliente a considerar sus requisitos cuidadosamente y revisarlos dentro del marco del problema. (10)

El proceso de Ingeniería de Requisitos está constituido por un conjunto de etapas, las cuales proporcionan una mayor organización durante todo el flujo de trabajo:

- Elicitación o captura de requisitos.
- Análisis y negociación de requisitos.
- Especificación de requisitos.
- Validación de requisitos.
- Administración o gestión de requisitos

## 1.5 Validación de requisitos

La validación es una de las etapas de la Ingeniería de Requisitos (IR). Su objetivo es verificar que todos los requisitos que aparecen en el documento especificado representan una descripción, por lo menos aceptable, del sistema que se debe implementar. Esto implica verificar que los requisitos sean consistentes y que estén completos. Permite además demostrar que los requerimientos definidos en el sistema son los que realmente quiere el cliente, revisa que no se haya omitido ninguno, que no sean ambiguos, inconsistentes o redundantes. Tiene como misión demostrar que la definición de requisitos especifica el sistema que quiere el usuario. Descubre problemas en el documento de requisitos antes de comprometer recursos en la implementación. Entre las actividades que realiza está la evaluación de los requerimientos, hacia la cual estará enfocado este trabajo de diploma.

La validación de un sistema no sólo se realiza al principio, como parte de la Ingeniería de Requisitos, sino también al final del desarrollo para determinar si se satisface los requisitos iniciales. Se realiza por medio de varios casos de prueba por cada requisito o caso de uso especificado. Según el estándar IEEE 1012-2004 Verificación y Validación, la validación es un proceso que proporciona evidencias de si el software, sus productos asociados y procesos:

- Satisfacen los requerimientos del sistema asignados al software al final de cada actividad del ciclo de vida.
- Resuelven el problema correctamente (por ejemplo: utilizan el modelo adecuado y ponen en práctica las reglas de negocio correctamente).
- Satisfacen el uso y las necesidades de los usuarios. (11)

### 1.5.1 Diferencia entre verificación y validación de requisitos de software

Para hacer más comprensible el contexto de la presente investigación, se hace necesario destacar, mediante la siguiente tabla, las diferencias entre los conceptos de validación y verificación de requisitos de software con el objetivo de erradicar confusiones e interpretaciones erróneas que puedan surgir.

**Tabla 1 Validación y Verificación**

Verificación	Validación
Confirmación mediante examen y evidencias objetivas que un elemento cumple con los requisitos específicos. (12)	Confirmación mediante examen y evidencias objetivas que un software cumple con los requisitos particulares para un uso específico. (12)
El proceso de verificación debe asegurar que para cada actividad existen las entradas y especificaciones adecuadas y que las salidas de esa actividad son correctas y consistentes con dichas entradas y especificaciones.	El proceso de validación asegura que en el producto final, las funciones líneas base y los rendimientos requeridos están correctos y completamente implementados.
En el diseño y desarrollo, la verificación involucra el proceso de examinar el resultado de una actividad dada para determinar la conformidad con el requisito declarado para esa actividad. (12)	Normalmente se realiza la validación en el producto final bajo las condiciones de operación definidas. Puede ser necesario en las fases más tempranas. (12)

## 1.5.2 Métodos de validación de requisitos de software

Los métodos para validación pueden clasificarse en:

- **Métodos estáticos:** Se centran en el análisis y comprobación de la representación del sistema, incluyendo los documentos, diagramas y el código.
- **Métodos dinámicos:** Implican ejecutar algún tipo de implementación del sistema. Podría parecer que sólo con los métodos estáticos es suficiente, pero esto no tiene sentido, puesto que los métodos estáticos se orientan más bien a la verificación, no pueden demostrar que el sistema satisface las expectativas del usuario, lo cual es demostrado mediante la validación. (13)

Este trabajo de diploma tiene interés en utilizar ambos métodos con el fin de poder detectar y corregir errores en los requisitos no funcionales. A continuación se detallan las características de las técnicas de validación de requisitos de software.

## 1.5.3 Técnicas de validación

Las técnicas de validación de los requisitos se realizan con el fin de examinarlos, para asegurarse de que se define el sistema adecuado. Permite detectar los errores de forma temprana, con el fin de no conducir a resultados inesperados, evitar gastos excesivos y grandes pérdidas de tiempo. Entre las técnicas de validación se encuentran:

### Revisión de requisitos

La revisión de requisitos es una de las mejores técnicas de validación. Generalmente permiten descubrir una gran cantidad de defectos en los requisitos, reducir los costos de desarrollo entre un 25% y un 30%, y el tiempo de pruebas entre un 50% y un 90%. Consisten en una o varias reuniones planificadas, donde se intenta confirmar que los requisitos poseen los atributos de calidad deseados. Estas reuniones son llevadas a cabo por el analista principal del proyecto y la representación del cliente. (14)

### Auditorías

Una auditoría según la norma internacional ISO 9000:2000, se puede definir como un proceso sistemático, independiente y documentado para obtener evidencias de la auditoría y evaluarlas de manera objetiva con el fin de determinar la extensión en que se cumplen los criterios de auditoría. (15) Además la IEEE, define a la auditoría como “un examen independiente de un producto de software, proceso del software, o sistema de procesos del software para determinar la conformidad con las especificaciones, los estándares, los acuerdos contractuales, u otros criterios”. (16)

La revisión de la documentación con esta técnica consiste en una verificación de los resultados contra una lista de chequeo predefinida o definida por el equipo de calidad del proyecto. (17) Una lista de chequeo es un mecanismo para controlar los riesgos. Existen muchos formatos de estas para evaluar y controlar diversas actividades o equipos, sin que esto quiera decir que se puede tomar una y aplicarla en un equipo o actividad similar, ya que se considera un error. Las listas de chequeo pueden transmitir la idea de que usted ha analizado cuidadosamente la situación, la secuencia que se debe seguir o la organización del trabajo. (18)

### Prototipo de interfaz de usuario

Esta técnica de validación se les muestra a los usuarios para que ellos obtengan una idea más precisa del producto final, es decir, es un diseño de menor porción que verificará que las especificaciones han sido realizadas de acuerdo con los requisitos del sistema. En un primer momento estos prototipos no tienen que obligatoriamente ser los finales, pues pueden surgir nuevas propuestas, que llevan a escoger según el más usable, debido a que este muestra el comportamiento externo del sistema. (13)

## Pruebas de validación

Las pruebas de validación en la ingeniería de software son el proceso previsto para mostrar que el producto cumple sus requerimientos, es decir, con su especificación y contenido. Normalmente es una parte del proceso de pruebas de software de un proyecto, que revela la presencia de errores, no su ausencia. Constituye la técnica que más se adhiere para la validación de requerimientos no funcionales, ya que el software tiene que ser ejecutado para ver su comportamiento, condición primordial para comprobar la correcta implementación de los mismos.

Se basa en ejercitar la implementación, por lo que solo puede ser aplicada si existe una versión operativa o ejecutable del producto. Consiste en evaluar el sistema o parte de este durante o al final del desarrollo para determinar si satisface los requisitos iniciales mediante la realización de uno o varios casos de prueba <sup>4</sup> por cada requisito no funcional. Es la aproximación formalizada de la revisión del documento, pensada explícitamente para la detección de defectos, no su corrección. Los defectos pueden ser errores lógicos, anomalías en el código o una no conformidad con los estándares. Existen tres tipos: pruebas de aceptación (realizadas por el cliente), pruebas de alfa (realizadas por el desarrollador) y pruebas de beta (realizadas por el cliente en su entorno de trabajo). (19)

Este trabajo de diploma utilizará las técnicas: revisiones, auditorías y pruebas de validación, por su uso mundial y efectividad en cuanto a dar soporte a los procesos de validación.

## 1.6 Concepto de métrica

En la mayoría de los desafíos técnicos, las métricas ayudan a entender tanto el proceso que se utiliza para desarrollar un producto, como el propio producto, así como también proporciona una manera de conocer el estado de su calidad, por lo que es utilizado en este trabajo de diploma, con el fin de medir la implementación correcta de los RNF.

Según el Ingeniero de Software Roger Pressman, es una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. (20) En general se entiende por métrica una forma de medir y una escala, definidas para realizar mediciones de uno o varios atributos, donde la escala es un conjunto de valores con propiedades definidas y el atributo es una propiedad mensurable, física o abstracta de una entidad, donde la entidad va a ser un objeto. (21)

### 1.6.1 Métricas de software

Las métricas de software se definen como: “La aplicación continua de mediciones basadas en técnicas para el proceso de desarrollo del software y sus productos, para suministrar información relevante a

---

<sup>4</sup> Artefactos bien definidos en el contexto de la prueba del software, es decir, es la descripción detallada de una acción que se debe realizar con el mismo. Acción específica donde están perfectamente descritos tanto los datos de entrada como las tareas a realizar y los resultados esperados.

tiempo, así el administrador junto con el empleo de estas técnicas mejorará el proceso y sus productos". (22)

Las métricas de software se caracterizan por tener: unidades y escalas de medición, ser simples y fácil de calcular, empíricas e intuitivamente persuasivas, sin ambigüedades y objetivas, consistentes en el empleo de unidades y tamaños, independientes del lenguaje de programación y eficaces para aumentar la calidad del software. (23)

Se debe destacar que no existen clasificaciones exactas sobre las métricas, los autores o especialistas las clasifican o nombran en función de las necesidades específicas del producto. En el caso particular de la universidad la situación no es diferente debido a que no se ha establecido tampoco una clasificación estándar, a pesar de conocer la importancia de la aplicación de estas en la calidad del producto. Algunas categorías son: métricas directas, métricas indirectas, métricas técnicas, métricas de proyecto, métricas de producto, métricas de complejidad, métricas internas, métricas externas y métricas de calidad. Este trabajo de diploma en aras de utilizar las que más de adecuen al objetivo que se quiere cumplir, estudiará la última clasificación a la que se hace referencia a continuación.

### 1.6.3 Métricas de Calidad

Las métricas de calidad se asocian a determinados atributos medibles, no siempre evidentes, para reconocer la presencia o ausencia de calidad. (24) Proporcionan una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente, es decir, cómo se va a medir para que el sistema se adapte a los requisitos que desea el cliente. Son todas las métricas de software que definen de una u otra forma la calidad del producto. (24)

### 1.7 Modelos de calidad de software

Diferentes literaturas coinciden en que un modelo de calidad es el conjunto de características y las relaciones entre las mismas, que proveen la base o regla para especificar requisitos de calidad, evaluar la calidad o comparar algún aspecto del software. (25) Promueven el buen uso de métodos y herramientas, y permiten la comunicación entre los desarrolladores. (25)

A continuación se mencionan los modelos de calidad del software que son utilizados por esta investigación para que sirvan de referencia y guía en el logro de los objetivos propuestos.

- ISO/IEC 9126 -1991. Define seis características de calidad deseables en cualquier software.
- IEEE 830-1998: Práctica recomendada para la especificación de requisitos de software.

### ISO/IEC 9126

La norma ISO/IEC 9126 permite que puedan ser utilizadas para especificar tanto los requisitos funcionales como no funcionales, de los clientes y usuarios. Especifica la calidad del producto, para evaluarlo desde perspectivas diferentes: adquisición, regulación, desarrollo, uso, evaluación, soporte,

mantenimiento, aseguramiento de la calidad y auditoría del software. Se compone de las partes que se relacionan a continuación:

- Parte 1. Modelo de Calidad ISO/IEC 9126-1.
- Parte 2. Métricas Externas ISO/IEC 9126-2.
- Parte 3. Métricas Internas ISO/IEC 9126-3.
- Parte 4. Métricas de Calidad en el uso.

La calidad del software puede evaluarse midiendo los atributos internos (medidas estáticas o productos intermedios) o atributos externos (comportamiento del código cuando se ejecuta). (8) El modelo se ha desarrollado en un intento de identificar los atributos más importantes para la calidad interna y externa en un producto software, por lo que su contexto identifica seis características claves de calidad.

## **IEEE 830-1998**

El estándar IEEE 830-1998 provee un enfoque recomendado de cómo especificar los requisitos del software, con el fin de conseguir un documento completo y sin ambigüedades que ayude: a los clientes o compradores, a describir con precisión lo que quieren obtener; a los suministradores, a comprender exactamente lo que el cliente quiere; a los individuos a cargo, a desarrollar una especificación de requisitos del software (ERS) normalizada para su organización. Define el formato y contenido de esas especificaciones, o bien de comprobar su calidad. (26)

La IEEE 830 no solo proporciona los objetivos de una buena especificación de requisitos, sino también los beneficios de estas adecuadas prácticas, que conllevan a una serie de ventajas entre las que destacan el contrato entre cliente y desarrolladores, la reducción del esfuerzo en el desarrollo, una buena base para la estimación de costes y planificación, un punto de referencia para procesos de verificación y validación, y una base para la identificación de posibles mejoras en los procesos analizados. (26)

## **1.8 Sistemas de Gestión de Información**

Aplicación que contiene un conjunto integrado de procesos, principalmente formales, que la organización conoce y sabe cómo utilizar (los informales no están excluidos), y son registrados en datos, a través de una base de datos. Desarrollada en un entorno usuario-ordenador y operando, sobre un conjunto de datos estructurados (base de datos), utilizando hardware y software computacional, redes de telecomunicaciones, técnicas de administración u otras formas de tecnología de información. (27)

Se caracteriza por la disponibilidad de la información cuando es necesaria y con los medios adecuados; por el suministro de la información de manera selectiva (cantidad contra calidad), variedad en la forma de presentación de la información (gráfica, numérica, etc.); grado de "inteligencia" del sistema (relaciones preestablecidas); tiempo de respuesta del sistema: desde una petición hasta su

realización; exactitud: conformidad entre los datos suministrados y los reales; generalidad: disponibilidad para atender diferentes necesidades; flexibilidad: capacidad de adaptación a nuevas necesidades; fiabilidad: probabilidad de operatividad correcta durante un período determinado de uso; seguridad: protección contra pérdida y/o uso no-autorizado de recursos; reserva: nivel de repetición de la información para proteger las pérdidas; y amigabilidad: necesidad de aprendizaje para su manejo.

## 1.8.1 Validación de los Sistemas de Gestión de Información

La validación de requisitos no funcionales en los Sistemas de Gestión de Información es importante realizarla tanto en la Ingeniería de Requisitos como terminada la implementación. Actualmente existen empresas dedicadas al aseguramiento de la calidad que realizan esta actividad en dependencia del recurso que es gestionado y de los servicios que son solicitados. Ejemplo de estas tenemos:

- **Green SQA<sup>5</sup>**: Ofrece servicios de pruebas de software, aseguramiento de calidad y consultoría a la industria del software. Brindan acompañamiento en el desarrollo y/o mantenimiento de aplicaciones, así como en la implementación de sistemas de gestión de calidad tales como la norma ISO9001 y el modelo CMMI.
- **Software Quality Assurance, SQA S.A<sup>6</sup>**: Dedicada a prestar servicios de consultoría y aseguramiento de calidad de software, cuyo enfoque es entender las necesidades de los clientes para brindar soluciones dirigidas en contribuir a la disminución de costos, generando mayor confianza de los productos de software en producción.
- **Indudata Ltda<sup>7</sup>**: Lleva 20 años en el mercado colombiano prestando servicios profesionales en Ingeniería de requerimientos, prueba , gestión de la configuración, ISO 27000, gestión de proyectos.
- **TSOFT<sup>8</sup>**: Empresa de Ingeniería de Software especializada en brindar soluciones de aseguramiento de calidad, optimización de las inversiones en tecnología informática, control gerencial de las operaciones informáticas y control de la disponibilidad de las transacciones de negocio.

Estas empresas tienen un alto reconocimiento a nivel internacional por la calidad de los servicios prestados. Toda la información referente a las metodologías practicadas o los procedimientos ejecutados, así como los resultados obtenidos, solo pueden ser accedidos por el personal implicado en el contrato, lo que se considera una limitante para satisfacer las necesidades del Centro de Tecnologías de Gestión de Datos.

---

<sup>5</sup> Green SQA .Web- <http://www.greensqa.com>

<sup>6</sup> Software Quality Assurance, SQA S.A. Web - <http://www.sgasa.com/>

<sup>7</sup> Indudata Ltda. Web- <http://www.indudata.com>

<sup>8</sup> TSOFT. Web- <http://www.tsoft.com.ar>

En la universidad la validación de los RNF de estos sistemas es realizada por CALISOFT y los grupos de calidad asociados a cada centro de desarrollo. Se ejecuta una vez terminada la implementación mediante un Plan de Pruebas de Liberación. Este define los tipos de pruebas que serán realizados, los cuales se establecen en función de las seis características de calidad definidas en la norma ISO/IEC 9126.

Por cada característica a evaluar es emitido una estrategia de prueba que precisa la planificación, el diseño y la ejecución de las pruebas. Esta validación realizada al final, para la liberación del sistema, trae consigo que los errores cometidos, muchas veces no evidentes, en la especificación de los RNF sean arrastrados hasta las últimas fases del ciclo de vida del software.

Luego de un análisis realizado a las alternativas existentes para la validación de los requisitos no funcionales, se concluyó que las mismas no son factibles para la presente investigación. La principal desventaja en el ámbito internacional es la imposibilidad de obtener la información respecto a cómo realizar la validación a los Sistemas de Gestión de Información. Particularmente en la universidad se realiza la validación solo en la etapa final de desarrollo del software provocando implementaciones fallidas, atrasos en tiempo de entrega, costos imprevistos e insatisfacción del cliente.

## 1.9 Definición de procedimiento

Según el reconocido ingeniero en software, Roger S. Pressman, un procedimiento se define como "los pasos que caracterizan el uso específico de cada elemento del sistema o el contexto del procedimiento en que reside el sistema". (20) La ISO/IEC 14598 define que los procedimientos están conformados por actividades, que no son más que un conjunto de tareas cooperativas e interacciones entre el solicitante y el evaluador (28).

Para este trabajo de diploma se define como procedimiento, una especificación de la serie de acciones, los actos o las operaciones que tienen que ser ejecutados de manera semejante para obtener siempre el mismo resultado en las mismas circunstancias. Indica una secuencia de actividades, tareas, pasos, decisiones y cálculos; que cuando están emprendidos en la secuencia colocada producen el producto o el resultado descrito, induciendo generalmente a un cambio; cumpliendo así, un objetivo específico.

### 1.9.1 Diferencias entre procedimiento y proceso

Es importante tener presente las diferencias entre procesos y procedimientos, para un mejor entendimiento de los objetivos que se quieren alcanzar con este trabajo de diploma.

Los procedimientos documentados sirven para establecer documentalmente el modo de realizar una actividad o un conjunto de actividades, centrándose en la forma en la que se debe trabajar. Mientras que, un proceso, transforma entradas en salidas, lo que resalta el propósito de las tareas que lo componen. La ISO 9000-2000 plantea que el proceso es un "Conjunto de actividades mutuamente

relacionadas o que interactúan, las cuales transforman entradas en salidas” y un procedimiento es una “Forma especificada para llevar a cabo una actividad o un proceso”. (29)

Por tanto, la principal diferencia está en que, un procedimiento permite que se realice una actividad o un conjunto de actividades, y puede ser documentado, mientras que un proceso permite que se consiga un resultado. Las labores que estructuran un proceso se pueden explicar a través de un procedimiento documentado.

## 1.10 Herramientas de modelado

La herramienta de modelado es una técnica que ayuda al ingeniero de software a "visualizar" lo que quiere construir y por ende, a verificar la corrección del modelo. Algunas de las herramientas CASE<sup>9</sup> conocidas son el ArgoUML, Rational Rose, Visual Paradigm, Easy CASE, Xcase, CASE Studio 2, CASEWise, entre otras. Dentro de las más utilizadas se encuentran el Rational Rose y el Visual Paradigm.

### 1.10.1 Rational Rose

El Rational Rose es una herramienta CASE potente, que se basa principalmente en una integración con el resto de las herramientas de desarrollo de IBM Racional<sup>10</sup>. Posibilita que halla varias personas trabajando a la vez en el proceso iterativo controlado, para ello permite que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios. Integración con cualquier sistema de control de versiones compatibles, como IBM Rational ClearCase<sup>11</sup>. Proporciona una facilidad de uso para modificar y crear nuevos diagramas.

### 1.10.2 Visual Paradigm

Visual Paradigm para UML es una herramienta CASE de modelado profesional que soporta el ciclo de vida completo del desarrollo de software. Brinda un ambiente amigable a los usuarios para analizar, diseñar y mantener diagramas. Es una herramienta colaborativa, es decir permite a múltiples usuarios trabajar sobre el mismo proyecto. (30) Admite la importación y exportación de XML e imágenes y cuenta con un soporte que facilita el trabajo simultáneo sobre un mismo diagrama entre dos desarrolladores en un tiempo real. Permite dibujar 13 tipos de diagramas diferentes a través de un intuitivo modelado visual. Proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos. Tiene licencia libre y comercial.

---

<sup>9</sup> Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora

<sup>10</sup> **International Business Machines** o **IBM** es una empresa transaccional que fabrica y comercializa herramientas, programas y servicios relacionados con la informática. IBM tiene su sede en Armonk (Nueva York, Estados Unidos).

<sup>11</sup> El Rational ClearCase se compone de varios programas y herramientas para apoyar la gestión de configuración de software (SCM) de código fuente y otros activos de desarrollo de software. Es desarrollado por la división Rational Software de IBM.

Teniendo en cuenta las características antes mencionadas del Visual Paradigm y Rational Rose, se seleccionó el Visual Paradigm, como herramienta para el diseño del procedimiento para validar requisitos no funcionales en los Sistemas de Gestión de Información, por la ventaja de que presenta una interfaz de usuario de fácil uso y que permite realizar los diagramas necesarios para el desarrollo del procedimiento, además generación de documentación en formatos HTML y PDF, sin necesidad de utilizar herramientas externas; y disponibilidad en múltiples plataformas: Microsoft Windows (98, 2000, XP, o Vista), Linux, Mac OS X, Solaris o Java.

## 1.11 Métodos de prospectiva

Los métodos de prospectiva estudian el futuro, en lo que se refiere a la evolución de los factores del entorno tecno-socio-económico y sus interacciones. Dentro de los métodos generales de prospectiva se destacan:

- **Métodos de expertos (método Delphi):** Se basan en la consulta a personas que tienen grandes conocimientos sobre el entorno en el que la organización desarrolla su labor. Estas personas exponen sus ideas y finalmente se redacta un informe en el que se indican cuáles son, en su opinión, las posibles alternativas que se tienen en el futuro.
- **Métodos extrapolativos:** En este método se proyectan hacia el futuro los datos de evolución que se tienen del pasado. Para ello se recopila la información histórica disponible y se buscan posibles tendencias o ciclos evolutivos. Estos proporcionan los posibles entornos futuros.
- **Métodos de correlación:** En éstos se intenta ver qué factores están implicados en un desarrollo y en qué grado influyen. Teniendo esto presente se determina cuál es la posible línea evolutiva que van a seguir todos estos factores.

Por la existencia en la universidad de personal con conocimientos en el tema pertenecientes al Centro de Calidad de Soluciones Tecnológicas (CALISOFT), la importancia de una valoración de ingenieros que realizan pruebas de software para aumentar la calidad de los productos y las características anteriormente mencionadas, se utilizará el método Delphi para la validación de la propuesta de este trabajo de diploma. A continuación se relacionan otros aspectos de este método.

### 1.11.1 Características del método Delphi

El método Delphi<sup>12</sup> fue ideado originalmente a comienzos de los años 50 en el seno del Centro de Investigación estadounidense RAND<sup>13</sup> Corporation por Olaf Helmer y Theodore J. Gordon, como un

---

<sup>12</sup> **Delphi:** El nombre de este método proviene del oráculo de Delphos, que se encontraba en la antigua Grecia, al que se acudía para hacer preguntas al dios a través de una sacerdotisa. El oráculo de Delphos poseía gran reputación por la certeza de sus predicciones.

<sup>13</sup> **Proyecto de Investigación y Desarrollo (RAND)** por sus siglas en inglés Research and Development, con más de 1000 investigaciones en la actualidad.

instrumento para realizar predicciones sobre un caso de catástrofe nuclear. Desde entonces, ha sido utilizado frecuentemente como sistema para obtener información sobre el futuro.

Es un método de estructuración de un proceso de comunicación grupal, que es efectivo a la hora de permitir a un grupo de individuos, como un todo, tratar un problema complejo. (31) Consiste en la selección de un grupo de expertos a los que se les pregunta su opinión sobre cuestiones referidas de acontecimientos del futuro, por lo que su capacidad de predicción se basa en la utilización sistemática de un juicio intuitivo emitido por estos. (32) Presenta las siguientes características fundamentales:

- **Anonimato:** Durante su ejecución, ningún experto conoce la identidad de los otros que componen el grupo de debate.
- **Iteración:** Se consigue al presentar varias veces el mismo cuestionario donde además se van presentando los resultados obtenidos con los cuestionarios anteriores. Esta característica puede no estar presente cuando no es necesario llegar a consenso entre los expertos.
- **Respuesta estadística:** La información que se presenta a los expertos no es sólo el punto de vista de la mayoría, sino que se presentan todas las opiniones indicando el grado de acuerdo que se ha obtenido.
- **Heterogeneidad:** Pueden participar expertos de determinadas ramas del conocimiento sobre las mismas bases.

## 1.11.2 Ventajas y desventajas del método de expertos

Los métodos de expertos tienen las siguientes ventajas:

- El conocimiento del grupo es siempre mayor que la del experto más versado en el tema.
- Permite obtener criterios desde diferentes puntos de vista. Cada experto puede aportar a la discusión general la idea que tiene sobre el tema debatido desde su área de conocimiento.
- Permite la participación de un considerable número de personas.

Sin embargo este método presupone también ciertas desventajas:

- La desinformación que presenta el grupo, como mínimo, es tan grande como la que presenta cada individuo aislado. Se debe suponer que la falta de información de unos participantes es solventada con la que aportan otros, y esto no se puede asegurar con certeza.
- La presión social que el grupo ejerce sobre sus participantes puede provocar acuerdos con la mayoría.
- En estos grupos a veces el argumento que triunfa es el más citado, en lugar del más válido.
- Estos grupos son vulnerables a la posición y personalidad de algunos de los individuos.

## **Conclusiones del capítulo**

En este capítulo se han identificado las características propias de la Ingeniería de Requisitos y sus etapas, haciendo hincapié en la validación de los requisitos, como proceso a desarrollar en este trabajo de diploma. Se seleccionaron los dos métodos: estático y dinámico, y las técnicas: revisiones, auditoría y pruebas de validación para el diseño del procedimiento. La herramienta de modelado Visual Paradigm permitirá ilustrar las fases. Se caracterizaron las diferentes clasificaciones de requisitos no funcionales, tras resultar seleccionado los requisitos de funcionalidad, eficiencia, portabilidad, confiabilidad, usabilidad y mantenibilidad. Se indicaron los estándares ISO e IEEE que guiarán su elaboración y el método Delphi para la validación de la propuesta. Se conceptualizó los términos: procedimientos, Métricas de Calidad y Sistemas de Gestión de Información.

## Capítulo 2 Descripción de la propuesta

### Introducción

En este capítulo se presenta la propuesta de un procedimiento para la validación de requisitos no funcionales en los Sistemas de Gestión de Información. Destacándose su propósito, alcance y características principales. Se precisa la descripción gráfica y textual de las actividades y los elementos que las componen, tales como: responsables, objetivos, artefactos de entrada y de salida del flujo de trabajo y técnicas a utilizar. Se muestra una descripción general y detallada de las fases por las que está compuesto el procedimiento, además de las tareas propuestas para darle cumplimiento a los objetivos establecidos.

### 2.1 Características principales del procedimiento

El procedimiento está conformado por un conjunto de actividades organizadas de forma lógica y secuencial de acuerdo al elemento que se esté analizando. La ejecución del procedimiento es un proceso iterativo e incremental por su propia naturaleza, es decir, se realizan varias iteraciones mientras los requisitos a validar no cumplan con las características establecidas; lo que permite a los desarrolladores ejecutar varias secuencias de forma escalonada, o sea, que mientras progresa el tiempo en el calendario cada una de estas producen un incremento de calidad en el software.

Se ajusta a cualquier metodología de desarrollo de software. En el mismo los artefactos y roles propuestos son declarados por la autora según sus investigaciones. Es proactivo, por el carácter preventivo de sus actividades, pues indica cómo desarrollar bien, de manera coherente y organizada, cada actividad y tarea. Tiene un amplio campo de aplicabilidad, dado por la factibilidad de su utilización para diversos tipos de proyectos durante la actividad de validación como parte de la Ingeniería de Requisitos. Mantiene un enfoque hacia la retroalimentación entre sus participantes, propiciada por la aplicación de sus actividades.

#### 2.1.1 Objetivo

Validar los requisitos no funcionales para la detección y corrección de errores durante el ciclo de vida del desarrollo de los Sistemas de Gestión de Información.

#### 2.1.2 Alcance

Sistemas de Gestión de Información desarrollados en el Centro de Tecnologías de Gestión de Datos (DATEC).

#### 2.1.3 Estructura

El procedimiento está estructurado por tres fases para facilitar la organización de las actividades que se realizan y un alcance satisfactorio del objetivo propuesto. Las fases se nombran: Fase I. Validación del Listado de Requisitos No Funcionales, Fase II. Validación de las especificaciones de los RNF y

Fase III. Validación por Métricas de Calidad (ver Figura 3), las mismas son explicadas con mayor detalle en el desarrollo del presente capítulo.

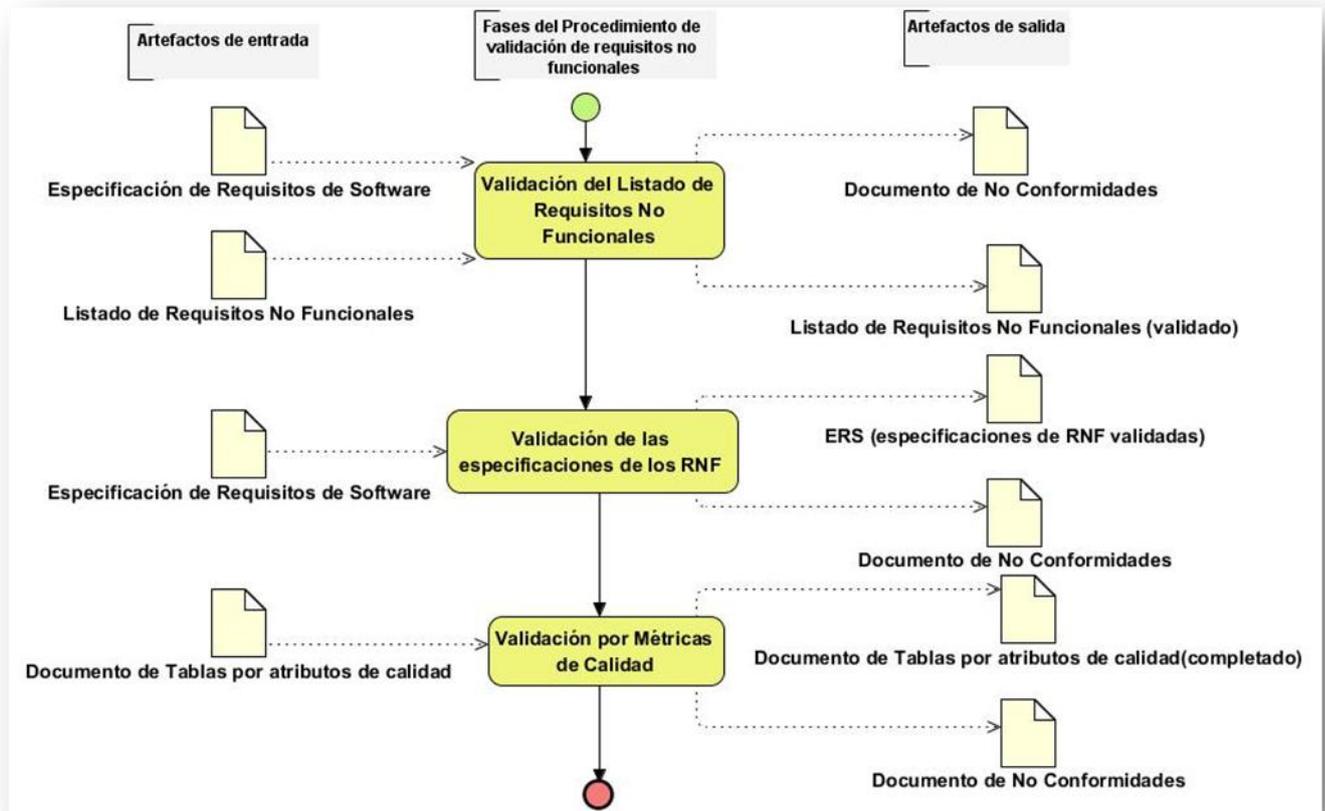


Figura 2. Fases del procedimiento

Las actividades de las tres fases del procedimiento están constituidas por los siguientes elementos:

- **Descripción:** Consiste en explicar detalladamente lo que deben realizar los involucrados en cada actividad y cómo se utilizan las técnicas definidas para darle cumplimiento al objetivo trazado. Describe el tratamiento que se les dará a las entradas de las actividades para producir las salidas.
- **Objetivo:** Constituye la meta fundamental de la actividad, define cuál es el propósito de la misma, en base al cuál trabajan las personas involucradas en su realización.
- **Responsable:** Juega un papel protagonista en el desarrollo de las actividades. Máximo encargado de los artefactos de entrada y salida, y del flujo de trabajo que se realiza.
- **Participantes:** Son las personas encargadas de materializar las actividades de corrección de los errores detectados. Su principal tarea es trabajar en conjunto con el responsable del procedimiento.
- **Actividades:** Conjunto de acciones que están en función de la fase en que se encuentra el procedimiento, que tienen como finalidad alcanzar el objetivo de la misma. Cada actividad está conformada por una secuencia de tareas o pasos ordenados lógicamente.

- **Artefactos de entrada:** Constituidos por toda la información y documentación necesaria para la ejecución de las actividades. Los participantes hacen uso de las mismas, procesándolas para obtener las salidas correspondientes a cada actividad en particular.
- **Artefactos de salida:** Documentos, modelos, tablas e información en general que se obtienen como resultado de la ejecución de las actividades. Muchos pueden constituir las entradas de otras actividades que a su vez se utilizan para generar otras salidas.
- **Técnicas:** Se emplean para permitir a los participantes en cada una de las actividades, la recopilación y obtención de la información necesaria para su realización. Proporciona un intercambio entre las partes implicadas favoreciendo su buen entendimiento y retroalimentación.

### 2.1.4 Responsables y participantes

A continuación se relaciona el responsable y los participantes propuestos para el desarrollo de las actividades que conforman el procedimiento. De cada uno se define el nombre, una breve descripción y responsabilidades que debe cumplir.

#### **Responsable: Administrador de la calidad**

##### **Descripción:**

El administrador de la calidad es la persona con experiencia en proyectos de desarrollo de software y con conocimientos suficientes sobre técnicas que aseguren la calidad del producto. Entre sus facultades está el de negociar con la calidad del mismo y, ocasionalmente, modificar el criterio de los desarrolladores. Además, dirige y controla todas las acciones de los participantes para que cumplan adecuadamente con su trabajo y se logre el objetivo final.

##### **Responsabilidades:**

Dirigir el desarrollo del procedimiento.

Generar los artefactos del procedimiento.

Informar las no conformidades encontradas al encargado de tratarlas.

Revisar toda la documentación que enmarca el procedimiento.

#### **Participante: Jefe del proyecto**

##### **Descripción:**

Es la persona que participa en todo el proceso de validación de los requisitos no funcionales, controlando y administrando todos los recursos que sean necesarios para su desarrollo, con el propósito de que se cumplan correctamente las actividades. Posee un conocimiento profundo de todas las actividades del negocio.

##### **Responsabilidades:**

Responsable de atender las necesidades del administrador de la calidad.

#### **Participante: Analista**

##### **Descripción:**

Es una persona con conocimientos en técnicas de validación, capaz de supervisar junto al jefe del proyecto todo el trabajo que enmarca el procedimiento. Debe tener preparación necesaria para la corrección de errores obtenidos durante la validación de requisitos no funcionales, y capacidad de comunicación, debido al contacto estrecho que establece con el cliente y demás participantes.

**Responsabilidades:**

Responsable de corregir los errores detectados.

**Participante: Cliente**

Persona encargada de velar que los requisitos no funcionales representen los intereses de su empresa. Sus conocimientos abarcan todas las proyecciones, las tendencias del negocio e información que se quieren informatizar. Participa junto al analista en las reuniones que se realizan para la corrección de los errores detectados.

**Responsabilidades:**

Participar activamente en las reuniones planificadas por el administrador de la calidad o el analista.

**Participante: Arquitecto**

El rol de arquitecto engloba varias tareas o dominios sobre las tecnologías del software, enfocado en los objetivos de la organización y las decisiones técnicas. Explora las alternativas de soluciones con un conocimiento pleno de las estrategias de negocios, estándares de calidad y políticas de la organización.

**Responsabilidades:**

Responsable de corregir los errores detectados.

### 2.1.5 Artefactos de entrada

**Listado de Requisitos No Funcionales**

Documento que constituye el modo habitual de guardar y comunicar requisitos no funcionales. Forma parte de los artefactos generados por el procedimiento. Primeramente se encuentra vacío y después de ejecutar la Fase I va a contener un listado de RNF, los cuales fueron seleccionados del Documento de Especificación de Requisitos de Software y están separados de sus especificaciones, es decir, solo contiene la clasificación (la que es atribuida en el documento de donde se extrajeron), número y enunciado. A continuación se muestra su estructura:



Figura 3. Listado de Requisitos No Funcionales

### Documento de Especificación de Requisitos de Software (ERS)

Documento que contiene los requisitos funcionales y requisitos no funcionales del sistema especificados por escrito. Supone una especie de contrato entre el cliente y los desarrolladores. Entre sus principales objetivos se encuentran: ayudar al cliente a describir claramente lo que se desea obtener mediante un determinado software, ayudar a los desarrolladores a entender qué quiere exactamente el cliente y servir de base para desarrollos de estándares de ERS particulares para cada organización. (33) Forma parte de la documentación asociada al software que se está implementando. Para su estructura la universidad se rige por lo establecido en el estándar IEEE 830-1998. La información que contiene referente a los RNF es utilizado por el procedimiento para la ejecución de la Fase I y Fase II, no se utiliza la de los RF porque no es de interés para esta investigación.

#### 2.1.6 Artefactos de Salida

##### Documento de No Conformidades (DNC)

Documento emitido por el administrador de la calidad durante la aplicación del procedimiento. En este quedan plasmados los errores detectados, a través de la tabla correspondiente a la fase que se está desarrollando. En estas tablas se registra toda la información pertinente a las no conformidades encontradas, con el objetivo de hacer más organizado el trabajo en el momento de corregir las mismas (ver Anexo 2). A continuación se muestra su estructura:

**DATTEC**  
CENTRO DE TECNOLOGÍAS DE GESTIÓN DE DAT

Documento de No Conformidades

<Nombre del Proyecto>  
<Nombre del producto>  
<Versión>

**Control del Documento**

	Nombre	Cargo	Fecha	Firma
Elaborado por:				
Aprobado por:				

Reglas de Confidencialidad  
Introducción  
Definiciones, Acrónimos y Abreviaturas  
Referencia  
Forma de uso  
Registro de No Conformidades

Figura 4. Documento de No Conformidades

### Documento de Tablas por atributos de calidad

Documento producido por el administrador de la calidad que recoge todas las tablas de la Fase III por atributos de calidad. Se plasman todas las métricas utilizadas para evaluar estos atributos, así como el resultado cualitativo y cuantitativo obtenido, que permite conocer el estado y las deficiencias de los mismos (ver Anexo 3). A continuación se muestra su estructura:

**DATTEC**  
CENTRO DE TECNOLOGÍAS DE GESTIÓN DE DAT

Documento de Tablas por atributos de calidad

<Nombre del Proyecto>  
<Nombre del producto>  
<Versión>

**Control de versiones**

Fecha	Versión	Descripción	Autor

Reglas de Confidencialidad  
Introducción  
Definiciones, Acrónimos y Abreviaturas  
Referencia  
Tablas por atributos de calidad  
Análisis de resultados

Figura 5. Documento de Tablas por atributos de calidad

### 2.2 Fase I. Validación del Listado de Requisitos No Funcionales

La Fase I se encuentra enfocada en analizar cada requisito no funcional, contenido en el Listado de Requisitos No Funcionales. Este artefacto es generado por el administrador de la calidad, el cual hace una solicitud al analista de la ERS, y plasma en el listado los enunciados, separado de sus especificaciones y respetando las clasificaciones ya definidas, de todos los RNF existentes, de la siguiente manera:

<Clasificación>

RNF<Número>:<Enunciado>

Se verifica el enunciado del RNF en cuanto a ambigüedad, consistencia, que sea posible de probar y que describa propiedades del sistema. Permitiendo obtener las no conformidades a partir de los errores encontrados, las cuales son reflejarlas en el DNC para que sean analizadas y corregidas por el analista.

Se realizan tantas iteraciones sean necesarias hasta que se cumpla, en un rango de 95% a 100%, los objetivos de cada tarea. El porcentaje es definido con estos límites para permitir que puedan existir de uno a tres errores en relación con la cantidad de RNF, debido a que su corrección depende de la opinión del analista, encargado de realizar esta actividad. El por ciento de cumplimiento se especifica en la Tabla 1: Deficiencias de los RNF, encontrada en el documento de Listados de Requisitos No Funcionales (ver Anexo 1). Se muestra además, aquellos conceptos que pueden resultar de difícil interpretación en la aplicación de estas actividades.

**Tabla 1 Fase I. Validación del Listado de Requisitos No Funcionales**

<b>Fase I. Validación del Listado de Requisitos No Funcionales</b>	
Objetivo	Validar el enunciado de los requisitos no funcionales.
Responsable	Administrador de la calidad.
Participantes	Analista, Jefe del proyecto, Cliente.
Artefactos de entrada	Listado de Requisitos No Funcionales. Documento de Especificación de Requisitos de Software
Artefactos de salida	Listado de Requisitos No Funcionales (validado). Documento de No Conformidades.
Actividades	Especificar por escrito en el Listado de Requisitos No Funcionales todos los enunciados de los RNF de la aplicación. Verificar que los enunciados de los RNF sean: <ul style="list-style-type: none"><li>➤ no ambiguos e inequívocos.</li><li>➤ posibles de probar.</li><li>➤ descritos como una restricción o propiedades del sistema.</li></ul>

	<ul style="list-style-type: none"> <li>➤ abstracto y conciso.</li> </ul> Especificar los errores encontrados en el DNC.
Técnica	Revisiones.

### 2.2.1 Actividad 1: Especificar por escrito en el Listado de Requisitos No Funcionales todos los RNF de la aplicación

La actividad consiste en detallar, en el Listado de Requisitos No Funcionales, todos los enunciados de los RNF que se encuentran en la ERS. Se verifica que han sido plasmados en su totalidad mediante la aplicación de reuniones con el analista, hasta que el documento esté completado.

**Tabla 2 Fase I. Actividad 1**

Objetivo	Especificar todos los enunciados de los RNF en el Listado de Requisitos No Funcionales.
Responsable	Administrador de la calidad.
Participantes	Analista, Jefe del proyecto, Cliente.
Artefactos de entrada	Listado de Requisitos No Funcionales. Documento de Especificación de Requisitos de Software.
Artefactos de salida	Listado de Requisitos No Funcionales (completado).
Tareas	Plasmar el enunciado de todos los RNF en el documento. Verificar que todos los requisitos no funcionales se encuentren contenidos en el documento recibido por el analista. Reunión con el analista para el análisis de las dificultades. Refinar el Listado de Requisitos No Funcionales.
Técnica	Revisiones.

### 2.2.2. Actividad 2: Verificar el enunciado de los requisitos no funcionales

Una vez plasmados todos los enunciados de los RNF en el Listado de Requisitos No Funcionales se verifica que cumplan con las siguientes características:

- **No sean ambiguos e inequívocos:** Los requisitos no funcionales deben estar escrito de forma tal que el texto sea claro, preciso y con una única interpretación posible lo mismo para el cliente que para el desarrollador del sistema. La escritura del requisito debe utilizar un vocabulario controlado, si es necesario utilizar una tabla de términos equivalentes (sinónimos); respetar normas ortográficas, sintácticas y gramaticales.
- **Posibles de probar:** Consiste en poder enunciar una prueba específica que establezca que se satisface. Un requisito es verificable cuando puede ser cuantificado o utiliza condiciones concretas y no es ambiguo.

## Capítulo 2: Descripción de la propuesta

- **Descritos como una restricción o propiedad del sistema:** Su contexto describa qué propiedades debe tener el sistema para realizar sus funciones y no cómo debe de hacerlo, que cada requisito no funcional represente las necesidades objetivas que debe cumplir el producto de software.
- **Abstractos y concisos:** Los RNF deben permitir tanto a los analistas como a los desarrolladores abstraerse lo más detalladamente posible del futuro sistema con la menor cantidad de información posible.

Se emplea la técnica de revisión con una participación activa del analista (principal protagonista en el levantamiento de requisitos) y procediendo a través de reunión con el mismo para ejecutar la tarea.

**Tabla 3 Fase I. Actividad 2**

Objetivo	Verificar el enunciado de todos los requisitos no funcionales que se encuentran en el Listado de Requisitos No Funcionales.
Responsable	Administrador de la calidad.
Participantes	Analista, Jefe del proyecto, Cliente.
Artefactos de entrada	Listado de Requisitos No Funcionales.
Artefactos de salida	Listado de Requisitos No Funcionales (validado). Documento de No Conformidades.
Tareas	Verificar que los enunciados de los RNF sean: <ul style="list-style-type: none"><li>➤ no ambiguos e inequívocos.</li><li>➤ posibles de probar.</li><li>➤ descritos como una restricción o propiedades del sistema.</li><li>➤ abstracto y conciso.</li></ul> Solicitar reuniones con el analista. Especificar los errores encontrados en el DNC.
Técnica	Revisiones.

### 2.3 Fase II. Validación de las especificaciones de los RNF

La Fase II está dirigida en revisar las especificaciones de los requisitos no funcionales de software, comprobando ante todo, que cumplan con las características establecidas por el estándar IEEE 830. Las actividades se rigen por lo que indica este estándar, según las características deseables para una correcta especificación de requisitos de software.

Corregidos los errores del enunciado del RNF (Fase I), se hace necesario verificar su especificación que se encuentra en el ERS de la siguiente manera:

<Clasificación>

RNF<Número>:<Enunciado>

## Capítulo 2: Descripción de la propuesta

### <Especificación>

Se realizan varias iteraciones de esta fase tantas veces sea necesario, es decir, hasta que el objetivo de cada actividad que la conforma este cumplido dentro del rango de un 95% a un 100%. El porcentaje es definido con estos límites debido a que pueden existir de uno a tres errores, en relación con la cantidad de RNF existentes, que no puedan erradicarse. Este problema se evidencia porque la corrección de las no conformidades depende de la opinión del analista, encargado de realizar esta actividad.

La técnica a utilizar es auditoría, que será aplicada mediante una lista de chequeo (ver Anexo 4). Esta no solo evalúa el artefacto ERS en general, también establece una serie de preguntas por cada actividad para velar por su desarrollo correcto y calcular su porcentaje de cumplimiento.

**Tabla 4 Fase II. Validación de las especificaciones de los RNF**

Fase II. Validación de las especificaciones de los RNF	
Objetivo	Validar la especificación de los requisitos no funcionales.
Responsable	Administrador de la calidad.
Participantes	Analista, Jefe del proyecto.
Artefactos de entrada	Documento de Especificación de Requisitos de Software.
Artefactos de salida	Documento de Especificación de Requisitos de Software (con las especificaciones de los RNF validadas). Documento de No Conformidades.
Actividades	Verificar que la especificación de los RNF sea correcta. Verificar que la especificación de los RNF sea no ambigua. Verificar que la especificación de los RNF sea completa. Verificar que la especificación de los RNF sea posible de probar. Verificar que la especificación de los RNF sea consistente. Verificar que la especificación de los RNF sea modificable.
Técnica	Auditoría.

### 2.3.1 Actividad 1: Verificar que la especificación de los RNF sea correcta

Una especificación se dice correcta si representan algo que es requerido para la construcción del sistema y no hay errores que afecten al diseño. (34) Un aspecto fundamental a tener en cuenta acerca de la corrección es que depende del cliente final del producto. Un problema habitual en este ámbito es la omisión (cliente) y adición (analista) de información.

**Tabla 5 Fase II. Actividad 1**

Objetivo	Verificar que la especificación de los RNF represente realmente las necesidades del cliente.
----------	--

## Capítulo 2: Descripción de la propuesta

Responsable	Administrador de la calidad.
Participantes	Analista, Jefe del proyecto.
Artefactos de entrada	Documento de Especificación de Requisitos de Software.
Artefactos de salida	Documento de Especificación de Requisitos de Software (especificación de los RNF validada en cuanto a que sea correcta). Documento de No Conformidades.
Tareas	Responder a las preguntas contenidas en las listas de chequeo referentes a esta actividad. Cuantificar de aquellas especificaciones que no son correctas. Registrar los errores encontrados en el DNC.
Técnica	Auditoría.

### 2.3.2 Actividad 2: Verificar que la especificación de los RNF sea no ambigua

El estándar IEEE 830 establece que un requisito es no ambiguo “si, y solo si, puede estar sujeto a una única interpretación”. (33) Asegurar que una especificación está libre de ambigüedades es difícil, pues las personas que intervienen en este proceso pueden provenir de diferentes culturas o tienen sus propios enfoques de percibir los conceptos. Para elaborar una ERS no ambigua es necesario mantener y documentar un acuerdo entre el equipo de desarrollo y los interesados respecto a los requisitos. Otra recomendación para reducir el problema, es que se deben añadir definiciones al glosario de términos.

**Tabla 6 Fase II. Actividad 2**

Objetivo	Verificar que la especificación de los RNF posea una única interpretación.
Responsable	Administrador de la calidad.
Participantes	Analista, Jefe del proyecto.
Artefactos de entrada	Documento de Especificación de Requisitos de Software.
Artefactos de salida	Documento de Especificación de Requisitos de Software (especificación de los RNF validada en cuanto a que no sea ambigua). Documento de No Conformidades.
Tareas	Responder las preguntas de la lista de chequeo correspondiente a esta actividad. Cuantificar de aquellas especificaciones que son ambiguas. Registrar los errores de ambigüedad en el DNC.
Técnica	Auditoría.

### 2.3.3 Actividad 3: Verificar que la especificación de los RNF sea completa

El estándar establece que una especificación es completa “si, y solo si”, existe una definición de

## Capítulo 2: Descripción de la propuesta

respuestas a todas las posibles entradas o salidas, tanto válidas como inválidas, en todas las posibles situaciones. (33)

**Tabla 7 Fase II. Actividad 3**

Objetivo	Verificar que las especificaciones de los requisitos no funcionales definan todas las posibles propiedades del sistema.
Responsable	Administrador de la calidad.
Participantes	Analista, Jefe del proyecto.
Artefactos de entrada	Documento de Especificación de Requisitos de Software.
Artefactos de salida	Documento de Especificación de Requisitos de Software (especificación de los RNF validada en cuanto a que sea completa). Documento de No Conformidades.
Tareas	Responder todas las preguntas contenidas en las listas de chequeo definidas para esta actividad. Cuantificar de aquellas especificaciones que no están completas. Registrar los errores encontrados en el DNC.
Técnica	Auditoría.

### 2.3.4 Actividad 4: Verificar que la especificación de los RNF sea probable

Se considera que una especificación es posible de probar, “si lo son cada uno de los requisitos constituyentes”. A su vez, se considera que un requisito individual es verificable “si existe un proceso acotado (en plazo y presupuesto) que permita determinar que el sistema construido satisface lo descrito en el propio requisito”. (33) Los requisitos son verificables dependiendo de la forma en que estén escritos. Un requisito no puede ser verificable si:

- Es ambiguo.
- Usa medidas no cuantificables.

**Tabla 8 Fase II. Actividad 4**

Objetivo	Verificar que la especificación de RNF pueda ser probada en una serie finita de pasos.
Responsable	Administrador de la calidad.
Participantes	Analista, Jefe del proyecto.
Artefactos de entrada	Documento de Especificación de Requisitos de Software.
Artefactos de salida	Documento de Especificación de Requisitos de Software (especificación de los RNF validada en cuanto a que sea posible de probar). Documento de No Conformidades.

## Capítulo 2: Descripción de la propuesta

Tareas	Responder a las preguntas formuladas en las listas de chequeo correspondientes a esta actividad. Cuantificar aquellas especificaciones que no son posibles de probar. Registrar los errores encontrados en el DNC.
Técnica	Auditoría.

### 2.3.5 Actividad 5: Verificar que la especificación de los RNF sea consistente

Según el estándar IEEE 830 se considera que una especificación es consistente “si, y solo si, no hay ningún subconjunto de requisitos descrito dentro de ella que esté en conflicto con cualquier otro”. (33) El mayor problema en este ámbito es la repetición de información en los requisitos (requisitos repetitivos y redundantes). A veces se usa la misma palabra para designar conceptos diferentes, lo que es nombrado como incoherencias.

Tabla 9 Fase II. Actividad 5

Objetivo	Verificar que la especificación de los RNF sea consistente.
Responsable	Administrador de la calidad.
Participantes	Analista, Jefe del proyecto.
Artefactos de entrada	Documento de Especificación de Requisitos de Software.
Artefactos de salida	Documento de Especificación de Requisitos de Software (especificación de los RNF validada en cuanto a que sea consistente). Documento de No Conformidades.
Tareas	Verificar que las especificaciones no sean incoherentes respondiendo a las preguntas formuladas en las listas de chequeo correspondiente a esta actividad. Cuantificar de aquellas especificaciones que no son consistentes. Registrar los errores encontrados en el DNC.
Técnica	Auditoría.

### 2.3.6 Actividad 6: Verificar que la especificación de los RNF sea modificable

Se considera que una especificación es modificable “si su estructura, es tal, que permite realizar cambios sobre los requisitos que contiene de forma sencilla, completa y consistente, manteniendo la estructura inicial del conjunto”. (33)

Para que sea modificable se requiere que contenga:

- Una facilidad de uso en la organización de los volúmenes de información, un índice y las referencias cruzadas explícitas.

## Capítulo 2: Descripción de la propuesta

- No sea redundante, es decir, el mismo requisito no debe aparecer en más de un lugar en el ERS.

**Tabla 10 Fase II. Actividad 6**

Objetivo	Demostrar que la estructura de la ERS permite hacer cambios sencillos en los requisitos sin que sean afectados ningún otro requisito relacionado.
Responsable	Administrador de la calidad.
Participantes	Analista, Jefe del proyecto.
Artefactos de entrada	Documento de Especificación de Requisitos de Software.
Artefactos de salida	Documento de Especificación de Requisitos de Software (especificación de los RNF validada). Documento de No Conformidades.
Tareas	Responder las preguntas de la lista de chequeo correspondiente a esta actividad. Cuantificar de aquellas especificaciones que no son modificables. Registrar los errores encontrados en el DNC.
Técnica	Auditoría.

### 2.4 Fase III. Validación por Métricas de Calidad

La Fase III está enmarcada en evaluar los seis atributos de calidad plasmados en la ISO/IEC 9126 que coinciden con los requisitos no funcionales seleccionados. Las métricas establecidas, utilizando como guía el mismo estándar, permiten calibrar las características y subcaracterísticas asociadas a los requisitos no funcionales.

Las validaciones tienen en cuenta las métricas que mejor se ajustan a las necesidades actuales de los RNF. Se efectúa un análisis de los resultados obtenidos luego de su aplicación para convertirlos de cuantitativos a cualitativos, que posteriormente posibilita definir el por ciento del cumplimiento de la característica medida. Todos estos cálculos y conversiones son recogidos en el Documento de Tablas por atributos de calidad, definido por la autora de este trabajo de diploma (ver Anexo 3). Se realiza varias iteraciones hasta lograr el cumplimiento de cada atributo de calidad entre un rango de 90% al 100%, debido a que pueden existir RNF que no son evaluados con las métricas planteadas.

#### Estructura de las métricas

Para un mayor entendimiento de las métricas se ha definido una estructura común para su planteamiento:

- Nombre de la métrica: Denominación de la métrica.
- La métrica se propone medir: Interrogante a ser respondida con la aplicación de la métrica.
- Método de aplicación: Provee una secuencia de pasos para la aplicación de la métrica.

## Capítulo 2: Descripción de la propuesta

- Medición (fórmula): Provee la fórmula de medición y el significado de los datos empleados.
- Interpretación del valor obtenido: Provee el rango para la limitación del valor obtenido y la conversión del mismo en un valor cualitativo.
- Unidad de medida: La estandarización de la medición que se realiza.

**Tabla 11 Validación por Métricas de Calidad**

<b>Fase III. Validación por Métricas de Calidad</b>	
Objetivo	Validar la calidad de los RNF a través de las métricas definidas.
Responsable	Administrador de la calidad.
Participantes	Arquitecto, Jefe del proyecto.
Artefactos de entrada	Documento de Tablas por atributos de calidad.
Artefactos de salida	Documento de Tablas por atributos de calidad (completado) Documento de No Conformidades.
Tareas	Validar los RNF asociados a la funcionalidad. Validar los RNF asociados a la confiabilidad. Validar los RNF asociados a la usabilidad. Validar los RNF asociados a la eficiencia. Validar los RNF asociados a la mantenibilidad. Validar los RNF asociados a la portabilidad.
Técnica	Pruebas de validación.

### 2.4.1 Actividad 1: Validación de los requisitos no funcionales asociados a la funcionalidad

Los RNF relacionados con la funcionalidad, no se refiere explícitamente a los que definen las funciones que deben desarrollar el software, sino a la capacidad del software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando se usa bajo las condiciones especificadas. (35) Se relacionan con lo que hace el software para satisfacer las necesidades, al tiempo que las otras características principalmente están vinculadas con el cuándo y cómo lo hacen.

**Tabla 12 Fase III. Actividad 1**

Objetivo	Medir los requisitos de funcionalidad en cuanto a las métricas planteadas por cada subcaracterística de calidad seleccionada.
Responsable	Administrador de la calidad.
Participantes	Arquitecto, Jefe del proyecto.
Artefactos de entrada	Documento de Tablas por atributos de calidad.
Artefactos de salida	Documento de Tablas por atributos de calidad (tablas del atributo de funcionalidad completadas).

## Capítulo 2: Descripción de la propuesta

	Documento de No Conformidades.
Tareas	<p>Evaluar la subcaracterística de seguridad a través de las métricas:</p> <ul style="list-style-type: none"> <li>➤ Porcentaje de ocurrencia de violaciones al control de acceso.</li> <li>➤ Porcentaje de cuentas de usuarios compartidas.</li> </ul> <p>Evaluar la subcaracterística de interoperabilidad a través de las métricas:</p> <ul style="list-style-type: none"> <li>➤ Probabilidad de intercambio de datos, en base a su formato.</li> <li>➤ Porcentaje de intercambio de datos, en base al éxito del intento.</li> </ul>
Técnica	Pruebas de validación.

Tabla 13 Fase III. Métrica 1

Nombre de la métrica	Porcentaje de ocurrencia de violaciones al control de acceso
La métrica se propone medir	¿Cuán ocurrente son las violaciones al control de acceso del sistema?
Método de aplicación	Deben ejecutarse los casos de pruebas dirigidas al acceso del sistema y contar las violaciones efectuadas al control de acceso. Contar el número de casos de pruebas ejecutados para acceder al sistema.
Medición (fórmula)	$C_{\text{Acceso}} = V_{\text{Efectuadas}} / N_{\text{CP}} * 100$ <p><math>V_{\text{Efectuadas}}</math> = Número de casos encontrados donde ha ocurrido acceso no-autorizado a la funcionalidad del producto.</p> <p><math>N_{\text{CP}}</math> = Número de casos de pruebas ejecutados para acceder al sistema.</p> <p><math>C_{\text{Acceso}}</math> = Porcentaje de ocurrencia de violaciones al control de acceso durante la ejecución de las pruebas.</p>
Interpretación del valor obtenido	<p><b>Rango:</b> <math>0 \% &lt;= C_{\text{Acceso}} &lt;= 100 \%</math></p> <p><b>Caso adecuado:</b> <math>0\% &lt; C_{\text{Acceso}} &lt;= 50\%</math> (con cercanía a cero). Ocurrieron pocas violaciones al sistema.</p> <p><b>Peor caso:</b> <math>50\% &lt; C_{\text{Acceso}} &lt; 100\%</math>. A mayor cercanía a 100% es que hubo una mayor ocurrencia de violaciones al control de acceso del sistema.</p> <p><b>Peor caso:</b> <math>C_{\text{Acceso}} = 100\%</math>. Todos los intentos de acceso no-autorizado al sistema han sido efectivos.</p> <p><b>Caso establecido:</b> <math>C_{\text{Acceso}} = 0\%</math>. No ocurrieron violaciones al control de acceso del sistema.</p>
Unidad de medida	Por ciento.

Tabla 14 Fase III. Métrica 2

Nombre de la métrica	Porcentaje de cuentas de usuarios compartidas
----------------------	---

## Capítulo 2: Descripción de la propuesta

La métrica se propone medir	¿Qué por ciento de usuarios que pueden acceder al sistema mediante cuentas compartidas?
Método de aplicación	Contar el número de usuarios con cuentas compartidas y el número de total de usuarios existentes.
Medición (fórmula)	$A_{CC} = U_{CC} / T_{Usuarios} * 100$ <p><math>U_{CC}</math> = Número de usuarios con cuentas compartidas.  <math>T_{Usuarios}</math> = Total de usuarios existentes.  <math>A_{CC}</math> = Porcentaje de usuarios con cuentas compartidas.</p>
Interpretación del valor obtenido	<p><b>Rango:</b> <math>0\% &lt; A_{CC} \leq 100\%</math></p> <p><b>Caso adecuado:</b> <math>0\% &lt; C_{Acceso} &lt; 50\%</math> (con cercanía a cero). Existen pocos usuarios con cuentas compartidas.</p> <p><b>Peor caso:</b> <math>50\% &lt; C_{Acceso} &lt; 100\%</math>. A mayor cercanía a 100% es que existe un número considerable de usuarios con cuentas compartidas.</p> <p><b>Peor caso:</b> <math>C_{Acceso} = 100\%</math>. Todos los usuarios tienen cuenta compartida.</p> <p><b>Caso establecido:</b> <math>C_{Acceso} = 0\%</math>. No existen usuarios con cuentas compartidas que comprometan la seguridad del sistema.</p>
Unidad de medida	Por ciento.

**Tabla 15 Fase III. Métrica 3**

Nombre de la métrica	Probabilidad de intercambio de datos, en base a su formato.
La métrica se propone medir	¿Cuán correctamente ha sido implementado el intercambio de funciones de interfaz para una transferencia de datos específica?
Método de aplicación	Ejecutar las pruebas a cada registro de salida de las funciones de interfaz de acuerdo con la especificación de los campos de datos. Contar el número de formatos de datos que deben ser intercambiados con otro software o sistemas durante las pruebas, en comparación con el número total.
Medición (fórmula)	$I_{Formato} = F_{Intercambiados} / T_{Formatos}$ <p><math>F_{Intercambiados}</math> = Número de formatos de datos intercambiados exitosamente con otro software o sistemas durante las pruebas.  <math>T_{Formatos}</math> = Número total de formatos de datos a intercambiar.  <math>I_{Formato}</math> = Intercambiabilidad de formatos de datos.</p>
Interpretación del valor obtenido	<b>Rango:</b> $0 \leq I_{Formato} \leq 1$

## Capítulo 2: Descripción de la propuesta

	<p><b>Caso adecuado:</b> <math>0 &lt; I_{\text{Formato}} &lt; 1</math>. A mayor cercanía al 1 resultará mayor intercambiabilidad.</p> <p><b>Peor caso:</b> <math>I_{\text{Formato}} = 0</math>. No hubo intercambio de datos.</p> <p><b>Caso establecido:</b> <math>I_{\text{Formato}} = 1</math>. Resultará un éxito, pues ha sido implementado correctamente las funciones de interfaz para la transferencia de datos.</p>
Unidad de medida	Datos intercambiados.

Tabla 16 Fase III. Métrica 4

Nombre de la métrica	Porcentaje de intercambio de datos, en base al éxito del intento.
La métrica se propone medir	¿Cuán frecuentemente falló el intento de intercambio de datos entre el software objeto de la prueba y otro software o sistemas?
Método de aplicación	Ejecutar las pruebas. Contar el número de casos en que las funciones de interfaz fueron usadas y fallaron en su ejecución.
Medición (fórmula)	<p><math>I_{\text{Exitosa}} = C_{\text{Exitosa}} * 100 / C_{\text{Totales}}</math></p> <p><math>C_{\text{Exitosa}}</math> =Número de casos en que fue exitoso al proceder a un intercambio de datos con otro software o sistemas.</p> <p><math>C_{\text{Totales}}</math> =Número de casos en que se intentó proceder a un intercambio de datos.</p> <p><math>I_{\text{Exitosa}}</math> =Por ciento de intercambiabilidad de datos exitosa.</p>
Interpretación del valor obtenido	<p><b>Rango :</b> <math>0\% \leq I_{\text{Exitosa}} \leq 100\%</math></p> <p><b>Caso adecuado:</b> <math>50\% \leq I_{\text{Exitosa}} &lt; 100\%</math>. A mayor cercanía al 100% es que hubo pocos fallos al intentar el intercambio.</p> <p><b>Peor caso:</b> <math>0\% &lt; I_{\text{Exitosa}} &lt; 50\%</math>. Existieron fallos en casi todos los intentos de transferencia.</p> <p><b>Peor caso:</b> <math>I_{\text{Exitosa}} = 0\%</math>. Existieron fallos en todos los intentos de transferencia.</p> <p><b>Caso establecido:</b> <math>I_{\text{Exitosa}} = 100\%</math>. No existieron fallos en los intentos de transferencia.</p>
Unidad de Medida	Por ciento.

### 2.4.2 Actividad 2: Validación de requisitos no funcionales relacionados con la confiabilidad

Los requisitos de confiabilidad permiten mantener un nivel de ejecución especificado cuando se usa

## Capítulo 2: Descripción de la propuesta

bajo las condiciones especificadas. (35)

**Tabla 17 Fase III. Actividad 2**

Objetivo	Medir los requisitos de confiabilidad a través de las métricas definidas por cada subcaracterística seleccionada.
Responsable	Administrador de la calidad.
Participantes	Arquitecto, Jefe del proyecto.
Artefactos de entrada	Documento de Tablas por atributos de calidad.
Artefactos de salida	Documento de Tablas por atributos de calidad (tablas del atributo de confiabilidad completadas). Documento de No Conformidades.
Tareas	Evaluar la subcaracterística madurez a través de las métricas: <ul style="list-style-type: none"> <li>➤ Porcentaje de erradicación de fallos reales.</li> <li>➤ Tiempo de medio entre fallos totales.</li> </ul> Evaluar la subcaracterística de recuperabilidad por la siguiente métrica: <ul style="list-style-type: none"> <li>➤ Tiempo medio de recuperación.</li> </ul>
Técnica	Pruebas de validación.

**Tabla 18 Fase III. Métrica 5**

Nombre de la métrica	Porcentaje de erradicación de fallos reales
La métrica se propone medir	¿Cuántos fallos reales (ocurridos durante la aplicación de las pruebas) han sido erradicados (solucionado las deficiencias de la implementación de los RNF)?
Método de aplicación	Contar el número total de fallos detectados y el número de fallos resueltos durante el período de pruebas.
Medición (fórmula)	$E_{\text{FallosReales}} = F_{\text{solucionados}} / F_{\text{detectados}} * 100$ $F_{\text{solucionados}} = \text{Número de fallos solucionados.}$ $F_{\text{detectados}} = \text{Número total de fallos reales detectados.}$ $E_{\text{FallosReales}} = \text{Por ciento de fallos solucionados respecto a los fallos reales detectados.}$
Interpretación del valor obtenido	<p><b>Rango:</b> <math>0\% \leq E_{\text{FallosReales}} \leq 100\%</math></p> <p><b>Caso adecuado:</b> <math>50\% &lt; E_{\text{FallosReales}} &lt; 100\%</math>. A mayor cercanía a 100% es que hubo un mayor número de fallos reales fueron corregidos.</p> <p><b>Peor caso:</b> <math>E_{\text{FallosReales}} = 0\%</math>. Los fallos reales no han sido corregidos.</p> <p><b>Peor caso:</b> <math>0\% &lt; E_{\text{FallosReales}} &lt; = 50\%</math> (con cercanía a cero). Un gran número de fallos no han sido corregidos.</p>

## Capítulo 2: Descripción de la propuesta

	<b>Caso establecido:</b> $E_{\text{FallosReales}} = 100\%$ . Todos los fallos reales han sido corregidos.
Unidad de medida	Por ciento.

**Tabla 19 Fase III. Métrica 6**

Nombre de la métrica	Tiempo medio entre fallos totales
La métrica se propone medir	¿Cuán frecuentemente el software fracasa en su operación?
Método de aplicación	Contar el número de fallos totales ocurridos durante el período de operación definido y computar el intervalo promedio entre fallos totales.
Medición (fórmula)	<p><math>P_{\text{FallosTotales}} = \sum(T_1+T_2+T_3...T_o) / N</math></p> <p><math>T_o = T_f - T_i</math></p> <p><math>T_o</math> = Tiempo de recuperación de la inactividad en cada n oportunidad.</p> <p><math>T_f</math> = Tiempo de fin de la recuperación.</p> <p><math>T_i</math> = Tiempo de inicio de la recuperación.</p> <p><math>N</math> = Número total de fallos realmente detectados (fallos totales ocurridos durante el tiempo de operación observado).</p> <p><math>P_{\text{FallosTotales}}</math> = Promedio de fallos ocurridos durante el tiempo de operación observado.</p>
Interpretación del valor obtenido	<p><b>Rango:</b> <math>0 &lt; P_{\text{FallosTotales}} \leq T_M</math></p> <p><math>T_M</math> = Mayor de los tiempos medidos.</p> <p><math>T_{M/2}</math> = Mitad del mayor de los tiempos medidos.</p> <p><b>Caso adecuado:</b> <math>0 &lt; P_{\text{FallosTotales}} \leq T_{M/2}</math>. A mayor cercanía a <math>T_{M/2}</math> mejor.</p> <p><b>Peor caso:</b> <math>T_{M/2} &lt; P_{\text{FallosTotales}} \leq T_M</math> Con cercanía a <math>T_M</math>.</p> <p><b>Caso establecido:</b> <math>P_{\text{FallosTotales}} = 0</math>. No hubo ocurrencia de fallos totales.</p>
Unidad de medida	$P_{\text{FallosTotales}}$ = horas/número de fallos. $T_o$ = horas.

**Tabla 20 Fase III. Métrica 7**

Nombre de la métrica	Tiempo medio de recuperación.
La métrica se propone medir	¿Cuál es el tiempo promedio que toma el sistema para completar la recuperación, una vez iniciada la misma?
Método de aplicación	Medir el tiempo total de recuperación cada vez que haya pasado a la inactividad (caído el sistema), contabilizar el número de veces que el sistema se recupera y calcular el tiempo promedio invertido en la

## Capítulo 2: Descripción de la propuesta

	recuperación.
Medición (fórmula)	$P_{\text{Recuperación}} = \sum(T_1+T_2+T_3\dots T_n) / N$ $T_n = T_f - T_i$ <p> <math>T_n</math> = Tiempo de recuperación de la inactividad en cada n oportunidad.  <math>T_f</math> = Tiempo de fin de la recuperación.  <math>T_i</math> = Tiempo de inicio de la recuperación.  <math>N</math> = Número de oportunidades en que el sistema entró en recuperación.  <math>P_{\text{Recuperación}}</math> = Promedio del tiempo de recuperación.         </p>
Interpretación del valor obtenido	<p><b>Rango:</b> <math>0 &lt; P_{\text{Recuperación}} \leq T_M</math></p> <p><math>T_M</math> = Mayor de los tiempos medidos.</p> <p><math>T_{M/2}</math> = Mitad del mayor de los tiempos medidos.</p> <p><b>Caso adecuado:</b> <math>0 &lt; P_{\text{Recuperación}} \leq T_{M/2}</math>. A mayor cercanía a <math>T_{M/2}</math> mejor.</p> <p><b>Peor caso:</b> <math>T_{M/2} &lt; P_{\text{Recuperación}} \leq T_M</math>. Con cercanía a <math>T_M</math>.</p> <p><b>Caso establecido:</b> <math>P_{\text{Recuperación}} = 0</math>. No hubo ocurrencia de recuperación del sistema.</p>
Unidad de Medida	$P_{\text{Recuperación}}$ = horas/número oportunidades. $T_n$ = horas.

### 2.4.3 Actividad 3: Validación de los requisitos no funcionales relacionados con la usabilidad

Los requisitos no funcionales de usabilidad se enmarcan en la capacidad del producto de software de ser comprendido, aprendido, utilizado y atractivo para el usuario, cuando se utilice bajo las condiciones especificadas. (35) La usabilidad debe abordar todos los ambientes del usuario que el software puede afectar, lo cual puede incluir la preparación para el uso y la evaluación de los resultados.

**Tabla 21 Fase III. Actividad 3**

Objetivo	Medir los RNF asociados al atributo de usabilidad a través de las métricas establecida por cada subcaracterística seleccionada.
Responsable	Administrador de la calidad.
Participantes	Arquitecto, Jefe del proyecto.
Artefactos de entrada	Documento de Tablas por atributos de calidad.
Artefactos de salida	Documento de Tablas por atributos de calidad (tablas del atributo de usabilidad completadas). Documento de No Conformidades.
Tareas	<p>Evaluar la subcaracterística de comprensibilidad a través de la siguiente métrica:</p> <ul style="list-style-type: none"> <li>➤ Accesibilidad a tutoriales.</li> </ul> <p>Evaluar la subcaracterística de conformidad con la usabilidad por la</p>

## Capítulo 2: Descripción de la propuesta

	siguiente métrica: ➤ Conformidad con la usabilidad.
Técnica	Pruebas de validación.

Tabla 22 Fase III. Métrica 8

Nombre de la métrica	Accesibilidad a tutoriales.
La métrica se propone medir	¿A qué proporción de tutoriales se puede acceder?
Método de aplicación	Buscar la especificación en la ERS que indica el número de tutoriales que debe tener el sistema. Acceder a los tutoriales. Contar el número de tutoriales al que se pudo acceder correctamente.
Medición (fórmula)	$A_{\text{Tutoriales}} = A_{\text{Exitoso}} / T_{\text{Tutoriales}}$ <p><math>A_{\text{Exitoso}}</math> =Número de tutoriales a los que pueden acceder los usuarios exitosamente.</p> <p><math>T_{\text{Tutoriales}}</math> =Número total de tutoriales a los que se pueden acceder (indicado en la ERS).</p> <p><math>A_{\text{Tutoriales}}</math> = Probabilidad de acceso a tutoriales.</p>
Interpretación del valor obtenido	<p><b>Rango:</b> <math>0 \leq A_{\text{Tutoriales}} \leq 1</math></p> <p><math>A_M = A_{\text{Tutoriales}} / 2</math> (Redondeado a unidades).</p> <p><b>Caso adecuado:</b> <math>A_M \leq A_{\text{Tutoriales}} &lt; 1</math>. A mayor cercanía a 1 es que hubo un mayor número de tutoriales accedidos.</p> <p><b>Peor caso:</b> <math>0 = A_{\text{Tutoriales}}</math>. Es que no se pudo acceder al número de tutoriales existentes.</p> <p><b>Peor caso:</b> <math>0 &lt; A_{\text{Tutoriales}} &lt; A_M</math> (con cercanía a cero). Es que hubo poco acceso a los tutoriales existentes.</p> <p><b>Caso establecido:</b> <math>A_{\text{Tutoriales}} = 1</math>. Existió un total acceso a todos los tutoriales existentes.</p>
Unidad de medida	Tutoriales accedidos.

Tabla 23 Fase III. Métrica 9

Nombre de la métrica	Conformidad con la usabilidad
La métrica se propone medir	¿Cuánto se adhiere el producto de software a las regulaciones aplicables, normas, convenciones, estilos y guías relacionadas con la usabilidad que son indicadas por el usuario en la ERS?
Método de aplicación	Contar los elementos que requieren estar en conformidad especificados en

## Capítulo 2: Descripción de la propuesta

	la ERS y los requisitos no funcionales de usabilidad que han sido implementados.
Medición (fórmula)	$C_{Usabilidad} = \frac{E_{Implementados}}{T_{Elementos}} * 100$ <p><b>E<sub>Implementados</sub></b> = Número de elementos especificados que requiriendo estar en conformidad con la usabilidad que han sido implementados.</p> <p><b>T<sub>Elementos</sub></b> = Número total de elementos que requieren estar en conformidad con la usabilidad especificados en la ERS.</p> <p><b>C<sub>Usabilidad</sub></b> = Por ciento de elementos de conformidad con la usabilidad implementados.</p>
Interpretación del valor obtenido	<p>Rango: <math>0\% \leq C_{Usabilidad} \leq 100\%</math></p> <p><b>Caso adecuado:</b> <math>50\% \leq C_{Usabilidad} &lt; 100\%</math>. A mayor cercanía a 100% es que hubo un mayor número elementos especificados implementados correctamente.</p> <p><b>Peor caso:</b> <math>C_{Usabilidad} = 0\%</math>. No existen elementos especificados requiriendo estar en conformidad con la usabilidad.</p> <p><b>Peor caso:</b> <math>0\% &lt; C_{Usabilidad} &lt; 50\%</math> (con cercanía a cero por ciento). Es que existen pocos elementos especificados requiriendo estar en conformidad con la usabilidad.</p> <p><b>Caso establecido:</b> <math>C_{Usabilidad} = 100\%</math>. Todos los elementos de conformidad con la usabilidad han sido implementados.</p>
Unidad de medida	Por ciento.

### 2.4.4 Actividad 4: Validación de los requisitos no funcionales asociados a la eficiencia

La eficiencia es la capacidad del producto de software para proporcionar una ejecución o desempeño apropiado, en relación con la cantidad de recursos utilizados, bajo condiciones establecidas. (35) Los recursos pueden incluir otros productos, otra configuración y el hardware del sistema.

Tabla 24 Fase III. Actividad 4

Objetivo	Medir los requisitos de eficiencia en cuanto a las métricas planteadas por cada subcaracterística de calidad.
Responsable	Administrador de la calidad.
Participantes	Jefe del Proyecto, Arquitecto.
Artefactos de entrada	Documento de Tablas por atributos de calidad.
Artefactos de salida	Documento de Tablas por atributos de calidad (tablas del atributo eficiencia completadas). Documento de No Conformidades.

## Capítulo 2: Descripción de la propuesta

Tareas	<p>Evaluar la subcaracterística de rendimiento a través de la métrica:</p> <ul style="list-style-type: none"> <li>➤ Tiempo de respuesta.</li> </ul> <p>Evaluar la subcaracterística utilización de recursos a través de la métrica:</p> <ul style="list-style-type: none"> <li>➤ Tiempo de espera del usuario ante la utilización de los equipos de entrada/salida (E/S).</li> </ul>
Técnica	Pruebas de validación.

Tabla 25 Fase III. Métrica 10

Nombre de la métrica	Tiempo de Respuesta
La métrica se propone medir	¿Cuánto tiempo toma completar una tarea en particular? ¿Cuánto toma antes de que el sistema responda a determinada operación?
Método de aplicación	Comenzar una tarea específica. Medir el tiempo que esta toma en completar una muestra hasta que concluya la operación emprendida. Guardar los registros de cada intento.
Medición (fórmula)	$T = S_F - S_I$ $S_F$ = Minutos marcados al concluir la operación. $S_I$ = Minutos marcados al iniciarla operación. $T$ = Tiempo empleado en obtener el resultado.
Interpretación del valor obtenido	<p><b>Rango:</b> <math>0 &lt; T \leq X</math>    <math>X</math> = Valor establecido por el cliente en la ERS.</p> <p><b>Caso adecuado:</b> <math>0 &lt; T &lt; X</math></p> <p><b>Peor caso:</b> <math>T &gt; X</math> <math>T=0</math> (no satisfactorio)</p> <p><b>Caso establecido:</b> <math>T = X</math>.</p>
Unidad de medida	Minutos.

Tabla 26 Fase III. Métrica 11

Nombre de la métrica	Tiempo de espera del usuario ante la utilización de los equipos de entrada/salida (E/S)
La métrica se propone medir	¿Cuál es el impacto de la utilización de los equipos de E/S en el tiempo de espera de los usuarios?
Método de aplicación	Ejecutar un número de tareas concurrentes, donde se utilicen equipos de E/S establecidos en la especificación y medir el tiempo de espera del usuario ante la utilización de dichos equipos.
Medición (fórmula)	$T_E = S_F - S_I$ $S_F$ = Minutos marcados al concluir la operación. $S_I$ = Minutos marcados al iniciarla operación. $T_E$ = Tiempo empleado en obtener el resultado con un equipo de E/S.

## Capítulo 2: Descripción de la propuesta

Interpretación del valor obtenido	<p><b>Rango:</b> <math>0 &lt; T_E \leq X</math> <math>X =</math> Valor establecido por el cliente (igual que el tipo de equipo de E/S a utilizar).</p> <p><b>Caso adecuado:</b> <math>0 &lt; T_E &lt; X</math></p> <p><b>Peor caso:</b> <math>T_E &gt; X</math> <math>T_E = 0</math> (no satisfactorio).</p> <p><b>Caso establecido:</b> <math>T_E = X</math>.</p>
Unidad de medida	Minutos.

### 2.4.5 Actividad 5: Validación de los requisitos no funcionales asociados a la mantenibilidad

La mantenibilidad se refiere a la capacidad del producto de software de ser modificado. Las modificaciones pueden incluir las correcciones, mejoras o adaptaciones del software a cambios en el ambiente.

**Tabla 27 Fase III. Actividad 5**

Objetivo	Medir la característica de mantenibilidad a través de la métrica establecida por la subcaracterística seleccionada.
Responsable	Administrador de la calidad.
Participantes	Jefe del Proyecto, Arquitecto.
Artefactos de entrada	Documento de Tablas por atributos de calidad.
Artefactos de salida	Documento de Tablas por atributos de calidad (tablas del atributo de mantenibilidad completadas). Documento de No Conformidades.
Tareas	Evaluar la subcaracterística de diagnosticabilidad a través de la siguiente métrica: <ul style="list-style-type: none"> <li>➤ Grado de implementación de las funciones de diagnóstico.</li> </ul>
Técnica	Pruebas de validación.

**Tabla 28 Fase III. Métrica 12**

Nombre de la métrica	Grado de implementación de las funciones de diagnóstico
La métrica se propone medir	¿Cuán capaces son las funciones de diagnóstico implementadas para efectuar análisis de fallos? ¿Puede el usuario identificar una operación que causa un fallo (y evitarla, recurriendo a una operación alternativa)?
Método de aplicación	Observar el comportamiento del administrador de la calidad que trata de resolver los fallos utilizando funciones de diagnóstico. Contar el número de fallos que pueden ser diagnosticados y el número de fallos registrados.
Medición (fórmula)	$G_{\text{Implementación}} = F_{\text{Diagnosticado}} / T_{\text{FR}}$ $F_{\text{Diagnosticado}} = \text{Número de fallos que pueden ser diagnosticadas (utilizando$

## Capítulo 2: Descripción de la propuesta

	<p>las funciones de diagnóstico) para evitar un fallo.</p> <p><math>T_{FR}</math> = Número total de fallos registrados.</p> <p><math>G_{Implementación}</math> = Grado de implementación de fallos diagnosticados.</p>
Interpretación del valor obtenido	<p><b>Rango:</b> <math>0 \leq G_{Implementación} \leq 1</math></p> <p><math>G_M = G_{Implementación} / 2</math> (Redondeado a las unidades)</p> <p><b>Caso adecuado:</b> <math>G_M \leq G_{Implementación} &lt; 1</math>. A mayor cercanía a 1 es que hubo un mayor número de fallos diagnosticados.</p> <p><b>Peor caso:</b> <math>G_{Implementación} = 0</math>. No se diagnosticaron fallos, revisar si son capaces las funciones de diagnóstico implementadas y si son las más correctas.</p> <p><b>Peor caso:</b> <math>0 &lt; G_{Implementación} &lt; G_M</math> (con cercanía a cero). Es que existen pocos fallos diagnosticados, revisar si son capaces las funciones de diagnóstico implementadas y si son las más correctas.</p> <p><b>Caso establecido:</b> <math>G_{Implementación} = 1</math>. Todos los fallos fueron diagnosticados.</p>
Unidad de Medida	Fallos.

### 2.4.6 Actividad 6: Validación de los requisitos no funcionales asociados a la portabilidad

La característica de portabilidad se refieren a la capacidad de producto de software de ser transferido de un ambiente a otro, este ambiente puede incluir el ambiente del software, del hardware u organizacional. (35)

Tabla 29 Fase III. Actividad 6

Objetivo	Medir la característica de portabilidad a través de las métricas establecida por cada subcaracterística seleccionada.
Responsable	Administrador de la calidad.
Participantes	Jefe del Proyecto, Arquitecto.
Artefactos de entrada	Documento de Tablas por atributos de calidad.
Artefactos de salida	Documento de Tablas por atributos de calidad (tablas del atributo de portabilidad completadas). Documento de No Conformidades.
Tareas	<p>Evaluar la subcaracterística instalabilidad por la métrica siguiente:</p> <ul style="list-style-type: none"> <li>➤ Facilidad de instalación.</li> </ul> <p>Evaluar la subcaracterística conformidad con la portabilidad por la métrica siguiente:</p> <ul style="list-style-type: none"> <li>➤ Conformidad con la portabilidad.</li> </ul>

## Capítulo 2: Descripción de la propuesta

Técnica	Pruebas de validación.
---------	------------------------

**Tabla 30 Fase III. Métrica 13**

Nombre de la métrica	Facilidad de instalación
La métrica se propone medir	¿Qué por ciento de facilidad de instalación tiene el producto de software en su ambiente de operación para los usuarios?
Método de aplicación	Observar el comportamiento del cliente cuando tratan de instalar el producto de software en su ambiente de operación.
Medición (fórmula)	$F_I = I_E / T_I * 100\%$ <p><math>I_E</math> =Número de casos en los cuales el usuario tiene éxito en las operaciones para una instalación adecuada a su conveniencia.</p> <p><math>T_I</math> = Número total de casos en los cuales el usuario intenta adecuar la instalación a su conveniencia.</p> <p><math>F_I</math> =Por ciento de facilidad de instalación.</p>
Interpretación del valor obtenido	<p><b>Rango:</b> <math>0\% \leq F_I \leq 100\%</math></p> <p><b>Caso adecuado:</b> <math>50\% &lt; F_I &lt; 100\%</math>. A mayor cercanía a 100% es que hubo un mayor número de casos en los cuales el cliente tiene éxito en las operaciones de instalación.</p> <p><b>Peor caso:</b> <math>F_I = 0\%</math>. No hubo éxito en las operaciones de instalación, existirá problemas al instalar el producto de software en su ambiente de operación.</p> <p><b>Peor caso:</b> <math>0\% &lt; F_I \leq 50\%</math> (con cercanía a cero). Es que existen pocas operaciones de éxito en la instalación.</p> <p><b>Caso establecido:</b> <math>F_I = 100\%</math>. El cliente no va tener problemas al instalar el producto de software en su entorno de trabajo.</p>
Unidad de Medida	Por ciento.

**Tabla 31 Fase III. Métrica 14**

Nombre de la métrica	Conformidad de la portabilidad
La métrica se propone medir	¿Cuán conforme es la portabilidad del producto de software con las regulaciones, normas y convenciones que les son aplicables?
Método de aplicación	Contar el número de elementos que se encontraron en conformidad y compararlo con el número de elementos que requieren estar en conformidad con la portabilidad, según la especificación en la ERS.
Medición (fórmula)	$C_P = (T_{EC} - E_C) * 100$

## Capítulo 2: Descripción de la propuesta

	<p><b>E<sub>c</sub></b> = Número de elementos especificados que requiriendo estar en conformidad con la portabilidad no han sido implementados.</p> <p><b>T<sub>EC</sub></b> = Número total de elementos especificados que requieren estar en conformidad con la portabilidad.</p> <p><b>C<sub>p</sub></b> = Por ciento de conformidad con la portabilidad.</p>
Interpretación del valor obtenido	<p><b>Rango:</b> <math>0\% \leq C_p \leq 100\%</math></p> <p><b>Caso adecuado:</b> <math>50\% &lt; C_p &lt; 100\%</math>. A mayor cercanía a 100% es que hubo un mayor número de elementos especificados que requiriendo estar en conformidad con la portabilidad.</p> <p><b>Peor caso:</b> <math>C_p = 0\%</math>. No han sido implementados los elementos especificados que requieren estar en conformidad con la portabilidad.</p> <p><b>Peor caso:</b> <math>0\% &lt; C_p \leq 50\%</math> (con cercanía a cero). Se debe poner empeño en el arreglo de la portabilidad del producto software.</p> <p><b>Caso establecido:</b> <math>C_p = 100\%</math>. Los elementos especificados que requieren estar en conformidad con la portabilidad han sido implementados.</p>
Unidad de medida	Por ciento.

### 2.5 Representación de los resultados de los requisitos no funcionales

Con aplicación de las métricas a los RNF, según los resultados obtenidos, se establecen tres casos: Caso adecuado, Peor caso y Caso establecido, con el porcentaje de cumplimiento que significa. Este por ciento es especificado para cada métrica en la Tabla: Interpretación del valor obtenido (ver Anexo 3). La sumatoria del por ciento de los casos definido representa el estado de la subcaracterística del atributo que se está evaluando.

Si alguna de las métricas obtienen un resultado que la ubica en Caso adecuado: significa que aunque el RNF está implementado, existen pocos elementos deficientes que no le permiten su funcionamiento correcto; o Peor caso: significa que existen RNF con una implementación deficiente o sin realizar; se registra una no conformidad. Permitiendo ir directamente hacia el área afectada que engloba el requisito, para que de esta manera no generalizar el estado del mismo, evitando caer en ambigüedades.

El Documento de Tablas por atributos de calidad (ver Anexo 3) contiene al final una tabla resumen que especifica el porcentaje del cumplimiento de los atributos (RNF) que son medidos por las métricas, así como la cantidad final de no conformidades encontradas. Esta permitirá decidir si se realiza una nueva iteración de la fase, es decir, si el por ciento del cumplimiento de cada atributo de calidad no se encuentra entre un 90 % y un 100%, se ejecutará otra iteración.

### **Conclusiones del capítulo**

En el presente capítulo se describió el procedimiento propuesto para la validación de los requisitos no funcionales en los Sistemas de Gestión de Información. Se definieron las tres fases que lo rigen, conformadas por 14 actividades en su totalidad. Fueron generados en el transcurso de las fases, tres artefactos de entrada y cuatro artefactos de salida. Se utilizaron 14 métricas para evaluar los atributos de calidad establecidos por el estándar ISO/IEC 9126.

## Capítulo 3 Validación del procedimiento

### Introducción

En este capítulo se realiza la aplicación del procedimiento propuesto a la Solución Integral para la Gestión de Proyectos y Acciones Centralizadas (SIGEPAC) del Ministerio del Poder Popular para la Comunicación y la Información (MPPCI) perteneciente a DATEC. Se aborda lo referente al análisis de los resultados obtenidos en la ejecución de las tres fases con una visualización gráfica. Se realiza una validación cuantitativa mediante método Delphi, basado en consultas a expertos, que permite determinar el posible nivel de aceptación y éxito de la propuesta definida.

### 3.1 Aplicación del procedimiento

El procedimiento es aplicado SIGEPAC, una herramienta informática para registrar, seguir y evaluar la ejecución física y financiera de los proyectos, así como el impacto de sus resultados. La solución de software está formada esencialmente por dos subsistemas: el de gestión de proyectos, al cual se le aplicará el procedimiento, y el de almacén de datos. A continuación se reflejan los resultados obtenidos en la aplicación de la propuesta diseñada.

#### 3.1.1 Aplicación de la Fase I

Generado por el administrador de la calidad, el Listado de Requisitos No Funcionales es el primer artefacto al cual según el procedimiento propuesto, se le aplica la validación. Luego de plasmar el enunciado de todos los requisitos no funcionales, separados de su especificación y respetando la clasificación propuesta por la ERS, se realiza un análisis detallado por parte del administrador de la calidad para verificar el cumplimiento de los objetivos de esta fase.

Se detectó en una primera iteración que solo el 85,41 % de los RNF no eran ambiguos, ya en la segunda iteración el 96 % contaban con esa característica. Ejemplo de requisitos refinados donde se ha eliminado la ambigüedad:

Requisito inicial: RNF 4: Permitir uso del teclado para realizar operaciones sobre el sistema (Permitir acceso rápido al sistema usando el teclado)

Requisito corregido: RNF 4: Permitir el uso del teclado para realizar operaciones de acceso al sistema.

Otros de los errores detectados durante la ejecución de esta fase es que en la primera iteración solo el 93,75 % de los requisitos eran concisos y abstractos. En la segunda iteración ya el 98 % cumplían con esta característica, es decir, permitían abstraerse lo más posible sobre lo que podría ser el futuro del sistema. Como ejemplo de un requisito abstracto y conciso se tiene:

RNF 37: Definir interfaz de comunicación: Requisitos no funcional que permitió hacer una abstracción precisa de cómo sería el futuro del sistema en cuanto a la interfaz de comunicación.

Continuando con la aplicación del procedimiento se detectó que el 96 % de los RNF eran descritos

como una restricción o propiedades del sistema, logrando en la segunda iteración que el 100 % de los requisitos cumplan con esta característica (por el primer porcentaje arrojado no era necesario realizar otra iteración porque se encontraba en el rango establecido, del 95 % al 100%, pero los demás tareas no lo cumplían).

Los RNF posibles de probar en la primera iteración se encontraban en un 93,75 %, ya en la segunda iteración cumplían con esta característica en un 100%.

Según el procedimiento propuesto en este trabajo, todos los errores detectados se encuentran en el Documento de No Conformidades, el cual es enviado a los analistas, estos a su vez lo revisan y aplicando nuevamente la técnica de revisiones se reúnen con los clientes y jefe de proyecto para corregir los mismos.

Fue necesario realizar un total de dos iteraciones de esta fase para la validación del Listado de Requisitos No funcionales. Dichos resultados se muestran a continuación en la siguiente figura:

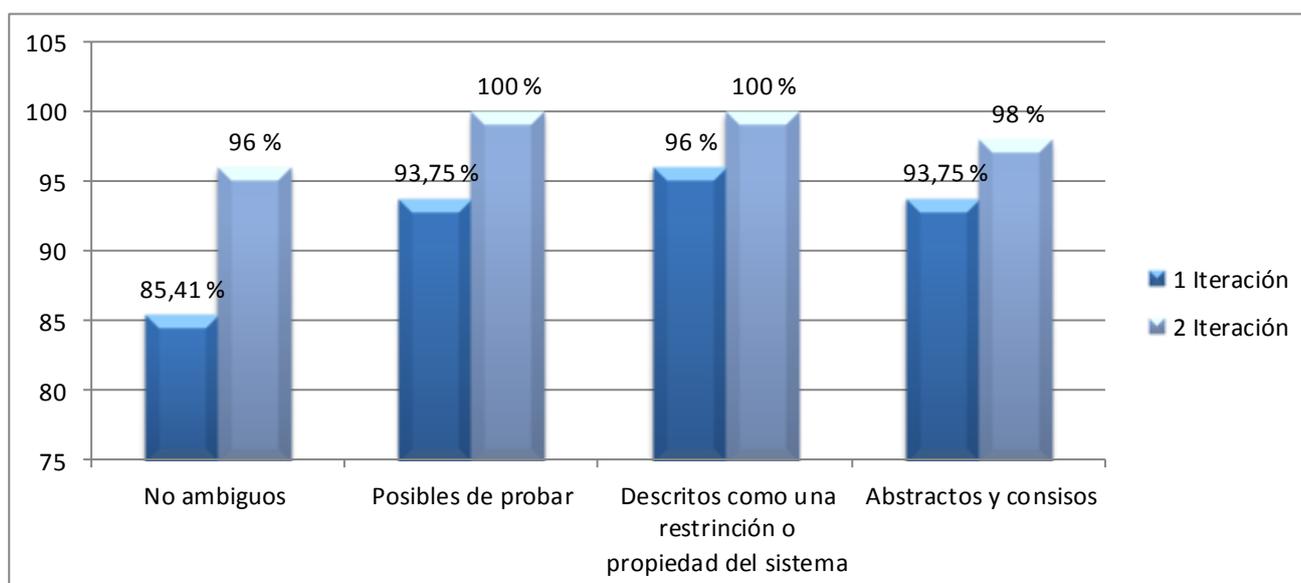


Figura 6. Resultados de la fase I

### 3.1.2 Aplicación de la Fase II

Luego de ser analizado los RNF separados de sus especificaciones y plasmadas en el DNC las no conformidades encontradas, las cuales son solucionadas por los analistas, que posteriormente realizan el refinamiento del documento de ERS; se procede a contestar las preguntas contenidas en la lista de chequeo perteneciente a esta fase.

Ejemplo de la aplicación de las listas de chequeo al documento de ERS:

## *Capítulo 3: Validación del procedimiento*

**Tabla 32 Ejemplo de aplicación de la lista de chequeo**

Elementos definidos por actividades de la Fase II.					
Actividad 1: Verificar que la ERS sea correcta.					
Peso	Indicadores a evaluar	Eval	(NP)	Cantidad de elementos afectado	# No conformidad
	1. ¿Están presentes todos los requisitos no funcionales requeridos por el cliente?	0			
crítico	2. ¿Todos los RNF especificados contribuyen a satisfacer una necesidad real del software?	0			
	3. ¿Existen RNF con omisión de información por parte del cliente?	1		2	18, 19
	4. ¿Existen RNF con adición de información por parte del analista?	1		3	20,21,22
	5. ¿La fuente de los RNF (por ejemplo: una persona, una regulación o un reglamento) está identificada?	0			

En el documento de ERS se detectaron que el 10,41 % de un total de 48 requisitos no funcionales, no son correctos. El 2,08% tienen problemas de ambigüedad, no son completos y no son modificables, y el 4,16 % no son consistentes, pero no existieron preguntas de peso críticos con evaluación de mal, es decir, definido con 1.

Para lograr que el documento de ERS cumpliera con todas las características requeridas por el procedimiento existió la necesidad de iterar en dos ocasiones, porque aunque la mayoría de los requisitos no funcionales cumplían con todas las características establecidas por esta fase, la característica de corrección no se encontraba en el rango definido (95% a 100%). Se obtuvieron como resultado los datos siguientes:

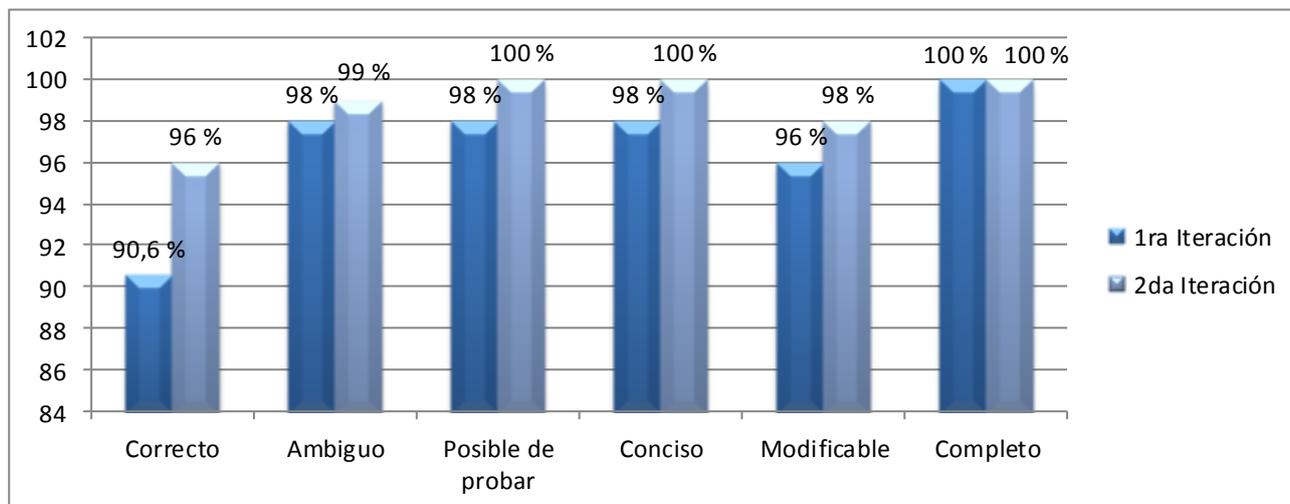


Figura 7. Resultados de la fase II

### 3.1.3 Aplicación de la Fase III

Para la aplicación de la Fase III se hizo necesario ejecutar las métricas recogidas en el Documento de Tablas por atributos de calidad, las cuales permiten una evaluación porcentual de las seis características de calidad deseables para el software establecidas por el estándar ISO/IEC 9126.

#### Funcionalidad

En la funcionalidad fueron utilizadas las métricas: porcentaje de ocurrencia de violaciones al control de acceso y porcentaje de cuentas de usuario compartidas, para medir la subcaracterística de seguridad. Todos los casos de pruebas fueron ejecutados al módulo de seguridad del SIGEPAC, de los cuales se obtuvieron resultados satisfactorios al no existir ocurrencia de violaciones al acceso del sistema (ver Anexo 6). En la métrica de porcentaje de cuentas de usuario compartidas los resultados fueron los mismos, porque solo existe una cuenta compartida: administrador, responsable de gestionar las cuentas de usuario. Aclarar que no se ha probado el RNF 21: Cantidad de usuarios conectados de forma simultánea, el cual establece que pueden existir al menos 1000 usuarios conectados de esa forma.

Otras métricas que evalúan la funcionalidad, en cuanto a la subcaracterísticas de interoperabilidad, son: la probabilidad de intercambio de datos, en base a su formato, y el porcentaje intercambio de datos, en base al éxito del intento. Los casos de pruebas fueron ejecutados a los cuatros componentes del Pentaho BI Suite Community Edition en su versión 3.5, los cuales son especificados en el RNF 30: Capa de inteligencia de negocios. Aclarar que aunque fueron pocos los intercambios de datos deficientes, si existieron problemas con el componente de Pentaho Desing Studio 3.5.

Después de sumar el porcentaje de las dos subcaracterísticas se llegó a determinar que la funcionalidad está en un 95 % de cumplimiento.

### **Confiabilidad**

Las medidas efectuadas a la confiabilidad la ubican en un 90 %. Aunque si se cumplió el porcentaje en cuanto a la erradicación de fallos, existió una evaluación cualitativa de Caso adecuado para las métricas: tiempo medio de recuperación y tiempo medio entre fallos totales. Esta evaluación es causada por la existencia de fallos totales en el intercambio de datos con el componente Pentaho Desing Studio 3.5 que no cumplían con el RNF 14 que establece un rango de recuperación que fluctúa entre 10 minutos y 72 horas para la solución del problema, validación y prueba.

### **Usabilidad**

La usabilidad se encuentra en un 95 %. Aunque no son especificados en la ERS la cantidad de tutoriales que debe tener la aplicación, aspecto necesario para aplicar la métrica de accesibilidad a tutoriales, sí se detalla que debe implementar una ayuda. Esta última está dividida en dos partes: una enfocada hacia los usuarios y otra a los desarrolladores (Guía del Usuario y Guía del desarrollador), que en su totalidad son 18 tutoriales a los que se pueden acceder fácilmente. De los 12 elementos (RNF) requiriendo estar en conformidad con la usabilidad, uno no está implementado en su totalidad (RNF 12: Cambiar de componente sin necesidad de autenticarse nuevamente).

### **Eficiencia**

La característica de eficiencia, que es evaluada por las métricas: tiempo de respuesta y tiempo de espera del usuario ante la utilización de los equipos de E/S, está en un 100 % según los RNF medidos. Para la primera métrica empleada se utiliza el RNF18: Tiempo de respuesta del sistema, el cual establece que 5 minutos como tiempo máximo para las respuestas, y para la segunda el RNF 4: Permitir el uso del teclado para realizar operaciones de acceso rápido al sistema.

### **Mantenibilidad**

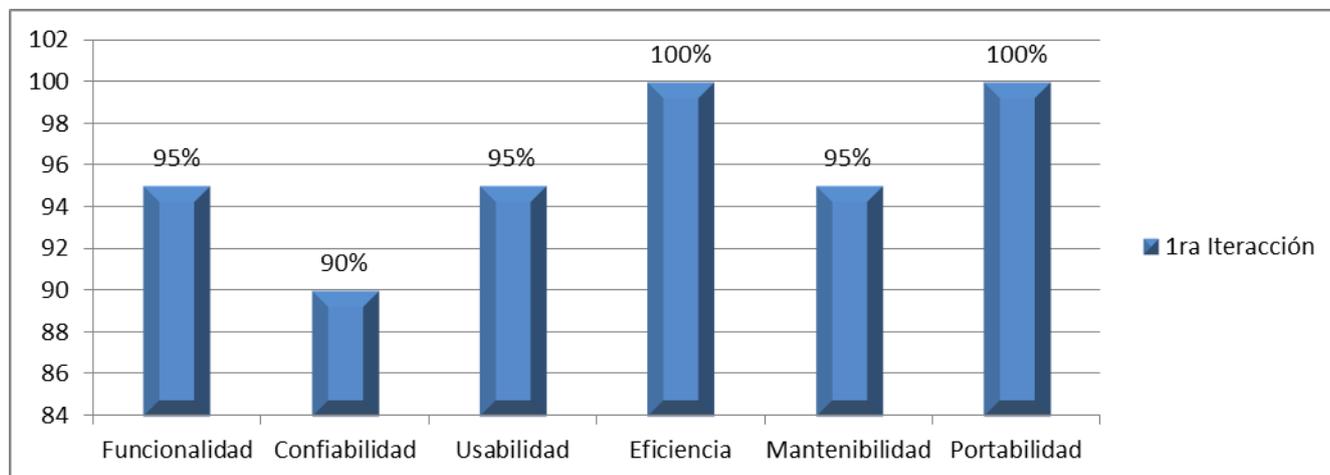
La mantenibilidad, que es medida por la métrica grado de implementación de las funciones de diagnósticos, se encuentra en un 95 %, porque no todos los fallos registrados fueron diagnosticados con dichas funciones.

### **Portabilidad**

Es necesario aclarar que esta característica en el módulo de gestión de proyectos, donde es aplicado el procedimiento, se encuentra en un 100% de implementación, debido a que es fácil de instalar por el usuario porque está soportado sobre una aplicación web y además todos los RNF referentes a la portabilidad han sido desarrollados de forma satisfactoria (el sistema ya ha sido probado en otro ambiente fuera del ámbito de producción). Si esta característica fuese medida en toda la solución tecnológica en general del MPPCI, se obtuviese un porcentaje inferior al 50 %, no por la

implementación de los elementos de portabilidad, sino porque resulta engorroso instalar esta de manera fácil por usuario común.

A continuación se representa gráficamente los resultados de esta fase:



**Figura 8. Resultados de la Fase III**

### 3.2 Validación de la propuesta mediante el método Delphi

De manera resumida para ejecutar el método se llevan a cabo los siguientes pasos:

1. Formulación del problema.
2. Elección de expertos.
3. Elaboración y lanzamiento de los cuestionarios (en paralelo con el paso 2).
4. Desarrollo práctico y explotación de resultados.

A continuación se desglosan los pasos planteados teniendo en cuenta dos cuestiones importantes: la selección de expertos y la elaboración del cuestionario, lo cual se efectúa sincrónicamente. Los criterios de evaluación son definidos por la autora de este trabajo de diploma en conciliación con los expertos seleccionados y se organizan por grupos en las siguientes categorías:

**Tabla 33 Criterios por grupo**

Grupo No 1: Criterios de mérito científico.	
1	Valor científico de la propuesta.
2	Calidad de la investigación.
3	Profesionalidad del investigador.
4	Contribución científica.
Grupo No 2: Criterios de implantación.	
5	Necesidad de utilización de la propuesta.
6	Posibilidades de aplicación.

## Capítulo 3: Validación del procedimiento

7	Satisfacción del problema.
8	Aporte práctico.
9	Obtención de productos finales con calidad.
<b>Grupo No 3: Criterios de flexibilidad.</b>	
10	Facilidad de entendimiento.
11	Facilidad de aplicación.
12	Interpretación de los resultados después de la implantación.
13	Capacidad de admitir cambios en el uso de métricas necesarias para la correcta validación.
<b>Grupo No 4: Criterios de impacto.</b>	
14	Repercusión en la Ingeniería de Requisitos de los proyectos productivos.
15	Posibilidad de aceptación.
16	Aceptación de la propuesta.

Se determina el peso relativo de cada grupo, el cual es asignado en conciliación con los expertos y teniendo en cuenta la importancia y cantidad de los criterios, además de los intereses a evaluar.

**Tabla 34 Peso por grupo**

Grupo	No 1	No 2	No 3	No 4
Peso	20	50	20	10

### Proceso de selección de los especialistas

La selección de especialistas se hace a un grupo de personas con conocimientos y capaces de ofrecer un criterio, aportar ideas o valoraciones concluyentes, sobre el tema a tratar. Se entiende por experto tanto al individuo en sí, como a un grupo de personas u organizaciones capaces de ofrecer valoraciones conclusivas de un problema en cuestión y hacer las recomendaciones que considere válidas para su enriquecimiento. Considerando la disponibilidad de las personas, las áreas a las que pertenecían y su vinculación con los ambientes de impacto de la propuesta, se proponen 15 especialistas de departamento de Pruebas de Ingeniería de Software y 20 del departamento de Desarrollo, del Centro de Calidad para Soluciones Tecnológicas (CALISOFT) de la universidad. Previa explicación del objetivo perseguido, se tiene en cuenta las siguientes características para la selección:

- Competencia.
- Creatividad.
- Disposición a participar en la encuesta.
- Capacidad de análisis y pensamiento.
- Espíritu colectivista.
- Espíritu autocrítico.

## Capítulo 3: Validación del procedimiento

La selección de expertos, atendiendo a estos criterios, proporciona la obtención de resultados con calidad, junto a otras cualidades propias, que pueden ser: la honestidad, la sinceridad y responsabilidad, hacen que las opiniones brindadas sean confiables y válidas para el objetivo propuesto. De esta forma culmina este proceso, logrando la participación de los seis expertos que cumplieran con las condiciones planteadas.

### Aplicación del cuestionario

Conformado el listado se le entrega el cuestionario donde es explicado el trabajo en general, la propuesta a evaluar y las instrucciones necesarias para contestar las preguntas, es decir, todos los detalles para que emitan en este la valoración del peso de cada criterio (ver Anexo 5).

Luego de haber aplicado el cuestionario a los expertos se obtuvieron los siguientes datos:

**G:** Grupos, **P:** Peso y **C:** Criterios.

Tabla 35 Resultado del cuestionario

G	P	C	Expertos						Calificación (Importancia)					
			E1	E2	E3	E4	E5	E6	E1	E2	E3	E4	E5	E6
1	20	1	5	5	6	3	5	3	5	5	5	4	4	5
		2	5	5	6	5	6	4	5	5	5	4	4	5
		3	5	5	8	5	5	10	4	5	5	5	4	5
		4	5	5	0	7	4	3	5	5	3	5	3	5
2	50	5	10	10	10	10	5	6	4	5	5	5	3	4
		6	10	10	10	10	10	15	5	5	5	5	5	4
		7	15	10	10	10	10	7	5	5	5	5	3	4
		8	10	10	10	10	15	6	5	5	5	5	4	3
		9	5	10	10	10	10	16	5	5	5	4	4	5
3	20	10	5	7	4	4	5	4	5	5	4	4	4	5
		11	6	3	6	6	5	7	5	5	5	5	4	4
		12	5	5	7	4	5	5	5	5	5	4	4	4
		13	4	5	3	6	5	4	4	5	4	5	3	5
4	10	14	3	5	5	4	4	2	5	5	5	4	4	3
		15	2	2	2	3	3	3	4	5	5	4	4	4
		16	5	3	3	3	3	5	5	5	5	4	4	4
Total		100	100	100	100	100	100							

### Concordancia de Kendall y estadígrafo Chi cuadrado

## Capítulo 3: Validación del procedimiento

Se verifica la consistencia en el trabajo de los expertos, para lo que se utiliza el coeficiente de concordancia de Kendall y el estadígrafo Chi cuadrado ( $X^2$ ). Se realizan los siguientes pasos:

Siendo **C** el número de criterios que se evaluó y **E** el número de expertos que realizaron la evaluación.

$$C=16 \text{ y } E=6$$

Se determina:

- **ΣE**: Sumatoria del peso dado por todos los expertos de cada criterio (por ejemplo sumatoria dada por todos los expertos del criterio 1).
- **Ep**: Puntuación promedio del peso dado por cada experto en concordancia con el número de criterios:  $Ep = \Sigma E / C$ .
- **MΣE**: media de los **ΣE**:  $M\Sigma E = \Sigma (\Sigma E / C)$ .
- **ΔC**: Diferencia (desviación) entre **ΣE** y **MΣE**:  $\Delta C = \Sigma E - \Sigma (\Sigma E / C)$  o  $\Delta C = \Sigma E - M\Sigma E$ .
- La desviación de la media (**ΔC**), que posteriormente se eleva al cuadrado para obtener la dispersión (**S**):  $S = \Sigma (\Sigma E - \Sigma \Sigma E / C)^2$  o  $S = \Sigma (\Delta C)^2$ .

Tabla 36 Datos para el cálculo de la dispersión (S)

C	E1	E2	E3	E4	E5	E6	ΣE	ΣE/C	MΣE	ΔC	(ΔC) <sup>2</sup>
1	5	5	6	3	5	3	27	1,69	37.50	-10.5	110.25
2	5	5	6	5	6	4	31	1,94	37.50	-6.5	42.25
3	5	5	8	5	5	10	38	2,38	37.50	0.5	0.25
4	5	5	0	7	4	3	24	1,50	37.50	-13.5	182.25
5	10	10	10	10	5	6	51	3,19	37.50	13.5	182.25
6	10	10	10	10	10	15	65	4,06	37.50	27.5	756.25
7	15	10	10	10	10	7	62-	3,88	37.50	24.5	600.25
8	10	10	10	10	15	6	61	3,81	37.50	23.5	552.25
9	5	10	10	10	10	16	61	3,81	37.50	23.5	552.25
10	5	7	4	4	5	4	29	1,81	37.50	-8.5	72.25
11	6	3	6	6	5	7	33	2,06	37.50	-4.5	20.25
12	5	5	7	4	5	5	31	1,94	37.50	-6.5	42.25
13	4	5	3	6	5	4	27	1,69	37.50	-10.5	110.25
14	3	5	5	4	4	2	23	1,44	37.50	-14.5	210.25
15	2	2	2	3	3	3	15	0,94	37.50	-22.5	506.25
16	5	3	3	3	3	5	22	1,38	37.50	-15.5	240.25
<b>S</b>											4180

Conociendo la dispersión se calculó el nivel de concordancia de **Kendall (W)**<sup>14</sup>

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

$$W = 4180 / 12 * (6^2) * (16^3 - 16)$$

$$W = 0.03$$

Con el coeficiente de concordancia de Kendall se calculó el **Chi cuadrado (X<sup>2</sup>)** para utilizar las tablas de distribución relacionadas con este estadígrafo:

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

$$X^2 = (16-1) * 6 * 0.03$$

$$X^2 = 2.7$$

El Chi cuadrado obtenido se comparó con los valores críticos de las tablas de distribución con 15 grados de libertad (C-1) indicando el valor **30.578** en la tabla que corresponde a **0.01** de probabilidad. Esto significa que en el valor calculado, la probabilidad es aún menor que 0.01, arrojando la conclusión que como se cumple que:

**X<sup>2</sup> real < X<sup>2</sup> (α, C-1)** Existe concordancia entre las opiniones de los expertos.

$$2.7 < 30.578$$

### Índice de aceptación (IA)

Para el cálculo del índice de aceptación se establecen primero los rangos de probabilidad de éxito:

- 0.8 < IA Existe **alta** probabilidad de éxito.
- 0.5 < IA ≤ 0.8 Existe probabilidad **media** de éxito.
- 0.3 < IA ≤ 0.5 Probabilidad de éxito **baja**.
- IA ≤ 0.3 **Fracaso seguro**.

Después de comprobar la consistencia del trabajo de los expertos y con el resultado del cuestionario aplicado se pudo definir el peso relativo de cada criterio: **P = EP / 100**

Dónde:

**EP:** Promedio de la evaluación dada por cada experto en concordancia con la cantidad de expertos.

$$EP = \sum E_i / E$$

**Tabla 37 Cálculo del peso relativo**

<b>C</b>	<b>Expertos</b>						<b>EP</b>	<b>P</b>
	<b>E1</b>	<b>E2</b>	<b>E3</b>	<b>E4</b>	<b>E5</b>	<b>E6</b>		
1	5	5	6	3	5	3	4.500	0.0450
2	5	5	6	5	6	4	5.167	0.0517
3	5	5	8	5	5	10	6.333	0.0633

<sup>14</sup> El coeficiente de concordancia de Kendall (W) fue obtenido independientemente por Kendall y Babington- Smith, 1939, y por Wallis, 1939.

## Capítulo 3: Validación del procedimiento

4	5	5	0	7	4	3	4.000	0.0400
5	10	10	10	10	5	6	8.500	0.0850
6	10	10	10	10	10	15	10.833	0.1083
7	15	10	10	10	10	7	10.333	0.1033
8	10	10	10	10	15	6	10.167	0.1017
9	5	10	10	10	10	16	10.167	0.1017
10	5	7	4	4	5	4	4.833	0.0483
11	6	3	6	6	5	7	5.500	0.0550
12	5	5	7	4	5	5	5.167	0.0517
13	4	5	3	6	5	4	4.500	0.0450
14	3	5	5	4	4	2	3.833	0.0383
15	2	2	2	3	3	3	2.500	0.0250
16	5	3	3	3	3	5	3.667	0.0367

Conociendo el peso de cada criterio y la calificación dada por los expertos en una escala de 1 al 5 se pudo construir la siguiente tabla para conocer la calificación promedio,  $c = \sum E1+E2...En / E$  y calcular el valor de  $P * c$  de cada criterio.

Se procede a calcular el índice de aceptación (IA), con la siguiente fórmula:  $IA = \sum (P * c) / 5$

Tabla 38 Cálculo del índice de aceptación

C	Calificación (Importancia)						Calificación Promedio (c)	P	P*c
	E1	E2	E3	E4	E5	E6			
1	5	5	5	4	4	5	5	0.0450	0.2100
2	5	5	5	4	4	5	5	0.0517	0.2411
3	4	5	5	5	4	5	5	0.0633	0.2956
4	5	5	3	5	3	5	4	0.0400	0.1733
5	4	5	5	5	3	4	4	0.0850	0.3683
6	5	5	5	5	5	4	5	0.1083	0.5236
7	5	5	5	5	3	4	5	0.1033	0.4650
8	5	5	5	5	4	3	5	0.1017	0.4575
9	5	5	5	4	4	5	5	0.1017	0.4744
10	5	5	4	4	4	5	5	0.0483	0.2175
11	5	5	5	5	4	4	5	0.0550	0.2567

## Capítulo 3: Validación del procedimiento

12	5	5	5	4	4	4	5	0.0517	0.2325
13	4	5	4	5	3	5	4	0.0450	0.1950
14	5	5	5	4	4	3	4	0.0383	0.1661
15	4	5	5	4	4	4	4	0.0450	0.1083
16	5	5	5	4	4	4	5	0.0517	0.1650
								$\Sigma(P * c)$	<b>4.5500</b>
								<b>IA</b>	<b>0.91</b>

Por último se determinó la probabilidad de éxito de la propuesta según los rangos predefinidos de índice de aceptación. Concluyendo que como el valor de IA (0.91) es mayor que 0.8 entonces la probabilidad de éxito es alta, lo que indica la factibilidad de la propuesta.

### Conclusiones del capítulo

En este capítulo se aplica el procedimiento para validar requisitos no funcionales a la Solución Integral para la Gestión de Proyectos y Acciones Centralizadas (SIGEPAC) del MPPCI. Se analizaron los resultados haciéndose necesario realizar dos iteraciones de la Fase I y Fase II. Se detalla además, todo el proceso de validación de la propuesta mediante el método Delphi, el cual no sólo aportó valores numéricos sobre la misma, sino que permitió también analizar sus resultados, con la obtención cifras positivas tanto en el nivel de concordancia de los expertos como en el índice de aceptación. Los expertos permitieron confirmar que el procedimiento es necesario, fácil de entender y aplicar, y con un elevado aporte práctico, por las métricas que propone.

## CONCLUSIONES GENERALES

Con la culminación de esta investigación se diseñó un procedimiento para la validación de los requisitos no funcionales de los Sistemas de Gestión de Información, arribándose a las siguientes conclusiones:

- Se analizó las normas y estándares de calidad, como parte de la realización del marco teórico de la investigación, incluyendo además el marco conceptual, analizándose dentro del mismo temáticas como el estudio de la Ingeniería de Requisitos, el cual arrojó como resultados que la validación de requisitos y las métricas de software contribuye al control, seguimiento y mejora de la calidad del proceso de desarrollo de software.
- Se diseñó un procedimiento capaz de mejorar la calidad de los requisitos no funcionales de los Sistemas de Gestión de Información, siendo preciso en la propuesta definir un conjunto de actividades, responsable, participantes, artefactos de entrada y salida, y técnicas que potenciaron su mejor entendimiento.
- Se validó la propuesta mediante el método de expertos, logrando una alta probabilidad de éxito, determinando la factibilidad del procedimiento y se aplicó a la Solución Integral para la Gestión de Proyectos y Acciones Centralizadas (SIGEPAC).

### RECOMENDACIONES

Para extender la investigación presentada en este trabajo de diploma se recomienda:

- Identificar otras métricas de calidad que se puedan emplear en aras de evaluar la calidad de los requisitos no funcionales, basándose en otras clasificaciones existentes.
- Efectuar un análisis de la posible vinculación del procedimiento propuesto con estrategias docentes e investigativas dentro del marco de la Ingeniería de Software.

## REFERENCIAS BIBLIOGRÁFICAS

1. **Pressman, Roger.** *Ingeniería del Software. Un enfoque práctico.* [ed.] McGraw-Hill. s.l. : Interamericana de España, 2002. págs. 184-186. Vol. 2.
2. **Sommerville, Ian.** *Ingeniería de Software.* [ed.] IV. Séptima Edición. Addison-Wesley : s.n., 2005.
3. **IEEE Standard 1990. Terminology, IEEE Standard Glossary of Software Engineering.** s.l. : Institute of Electronics Engineers, 1990, Vol..2.
4. **Requisitos de Software. Barros, Pablo Straud.** Santiago de Chile: Universidad Diego Portales : s.n., 2005.
5. **Young, Ralph Rowland.** *The Requirements Engineering Handbook.* London: Artech House : s.n., 2004.
6. **Wieggers, Karl Eugene.** *More About Software Requirements: Thory Inssues and Practical Advice.* s.l. : Microsoft Press, 2006.
7. *Documento de Especificación Requerimientos No Funcionales del Proyecto Mejoramiento de procesos, Análisis y Diseño del Sistema de Información para la Vigilancia de los eventos en salud pública en la Fase 1.* **Siglo21, Informática.** Colombia : s.n., 2006.
8. *Mastering the Requirements Process.* **Robertson, Suzanne Robertson y James.** Inglaterra: Pearson : s.n., 1999.
9. **I. Brito, A. Moreira and J. Araújo.** A requeriments model for quality attributes. [En línea] Universidad Twente, 22 de Abril de 2002. [Citado el: 25 de Noviembre de 2011.] <http://www.utwente.nl/AOSD-EarlyAspectsWS/Papers/Brito.pdf>.
10. **Ruth Malan, Dana Brendmeyer.** resources for Software Architects. Defining Non-Functional Requirements. [En línea] 2001. [Citado el: 10 al 25 de Nobiembre de 2011.] [http://www.bredemeyer.com/pdf\\_files/NonFunctReq.PDF](http://www.bredemeyer.com/pdf_files/NonFunctReq.PDF).
11. *Non-Functional Requirements in Software Engineering.* **otros, Lawrence Chung y.** ISBN: 0792386663, London: Kluwer Academic Plublishers : s.n., 1999.
12. **Boehm, B. M ,** *Characteristics of Software Quality .* Nueva York: North Holland : s.n., 1979.
13. **Arias, M.** [En línea] 2006. [Citado el: 24 de Noviembre de 2011.] [http://www.intersedes.ucr.ac.cr/pdfs\\_10/10-art\\_11.pdf](http://www.intersedes.ucr.ac.cr/pdfs_10/10-art_11.pdf).
14. **Dávila, D.** *Ingeniería de Requerimientos.* 2001.
15. **Pressman, Roger.** *Ingeniería de software: Un enfoque práctico.* 5ta. La Habana : Editorial Félix Valera, 2005.
16. [En línea] [Citado el: 27 de Noviembre de 2011.] <http://pesona.mmu.edu.my/~wruslan/SE2/Readings/detail/Reading-7.pdf>.

17. *Ingeniería de Software. Calidad del Producto (ISO 9126)*. ISO, 2001. 2001, Norma Cubana, Vols. Parte 1: Modelo de Calidad(ISO/IEC 9126-1:2001).
18. **M.J. Escalona, N. Koch.** *Ingeniería de requisitos en aplicaciones para la web. Un estudio comparativo.*
19. **L. C. Briand, J. W. Daly y J. K. Wust.** *A Unified framework for Coupling Measurement in Object-Oriented Systems.* 1999. págs. 91-121. Vol. IEEE transactions on software engineering .
20. **D.C.C.Muñoz.** *MÉTRICAS DEL SOFTWARE: Conceptos básicos, definición y formalización.* 2006.Vol. 13.
21. **Pressman, Roger.** *Ingeniería del Software. Un enfoque Práctico.* 5ta. 1998.
22. **Olsina, D.L.** *Métricas e Indicadores: Dos conceptos Claves para la Medición y Evaluación.* 2003. pág. 52.
23. **Gonzalez, C.** Conceptos Generales de calidad total. [En línea] 2005. [Citado el: 4 de Diciembre de 2011.] <http://www.monografias.com/trabajos11/conge/conge.shtml>.
24. **García, M.B.** *Estimación de costes de un proyecto informático.* pág. 60.
25. **Robles, M.C..E.A.** *Métricas para la Gestión de Proyectos de Software.* pág. 65.
26. **Giraldo, O.P.** *Métricas, Estimación y Planificación de Software.* pág. 10.
27. *TENOLÓGÍA DE LA INFORMACIÓN. CARACTERÍSTICAS DE LA CALIDAD Y MÉTRICAS DEL SOFTWARE. 9126-1, Norma Cubana NC ISO/IEC.* pág. 38.
28. **Crosby, Philip.** *La Calidad del Software .* [En línea] [Citado el: 5 de Diciembre de 2011.] <http://www.softqanetwork.com/philip-crosby-y-la-calidad-del-software>.
29. **Vázquez, Roberto Hugo.** *Taller de Calidad de Sostware: Introducción a la Calidad de Software.* [En línea] [Citado el: 15 de Diciembre de 2011.] <http://gridtics.frm.utn.edu.ar/docs/Introduccion%20a%20la%20Calidad%20de%20Software%20Vazquez.pdf>
30. **Calidad de Software.** [En línea] Enero de 2011. [Citado el: 17 de Diciembre de 2011.] [http://www.ecured.cu/index.php/Calidad\\_de\\_Software](http://www.ecured.cu/index.php/Calidad_de_Software).
31. **Estándares de Calidad para el Desarrollo y Mantenimiento de Software.** [En línea] Escuela de Computación e Informática. Universidad de Costa Rica. San Pedro, Costa Rica. [Citado el: 13 de Diciembre de 2011] [http://www.eici.ucm.cl/Academicos/R\\_Villarroel/descargas/calidad.produccion/Tutorial.calidad.M.Jenkins.pdf](http://www.eici.ucm.cl/Academicos/R_Villarroel/descargas/calidad.produccion/Tutorial.calidad.M.Jenkins.pdf).
32. *Especificación de Requisitos de Software según el estándar de IEEE 830 IE78.* **Agut, Ing. Raúl Monferrer.** Departamento de Informática. Universidad de Jaume : s.n., Ingeniería de Software. 5to Curso de Ingeniería Informática(2000-2001).
33. *NORMAS DE CALIDAD DEL PRODUCTO DE SOFTWARE.* **Angeleri, M.P.M.** Vol. Volumen.

34. SlideShare Inc. All rights reserved. [En línea] 2009. [Citado el: 5 de Enero de 2012.] <http://www.slideshare.net/ivonnetrabajosocial/gestion-de-la-informacion-sigeuis-3477938>.
35. **Gloria Ponjuán Dante**. Gestión de Información: Dimensiones e Implementación para el éxito organizacional. *NUEVO PARHADIGMA EDICIONES*. [En línea] 2004. [Citado el: 14 de Enero de 2012.] [http://ftp.ceces.upr.edu.cu/centro/repositorio/Textuales/Articulos/IT\\_Gestion%20de%20informacion.pdf](http://ftp.ceces.upr.edu.cu/centro/repositorio/Textuales/Articulos/IT_Gestion%20de%20informacion.pdf).
36. **Pechúan, Ingacio Gil**. Sistemas y Tecnologías de la Información para la Gestión. [En línea] 2006. [Citado el: 15 de Enero de 2012.]
37. Visual Paradigm International. [En línea] 2005. [Citado el: 20 de Enero de 2012.] <http://www.visualparadigm.com/product/vpuml/provides/>, consultado el 28 de febrero de 2012.
38. **Engineers, Institute of Electrical and Electronics**. IEEE 830 Recommended practice for software requirements specification. [En línea] 1998. [Citado el: 12 de Febrero de 2012.] <http://www.computer.org/>.

## BIBLIOGRAFÍA

**Agut, Ing. Raúl Monferrer.** *Especificación de Requisitos de Software según el estándar de IEEE 830 /E78*. Departamento de Informática. Universidad de Jaume : s.n., Ingeniería de Software. 5to Curso de Ingeniería Informática(2000-2001).

**Angeleri, M.P.M.** *NORMAS DE CALIDAD DEL PRODUCTO DE SOFTWARE*. Vol. Volumen.

**Arias, M.** [En línea] 2006. [Citado el: 24 de Noviembre de 2011.]  
[http://www.intersedes.ucr.ac.cr/pdfs\\_10/10-art\\_11.pdf](http://www.intersedes.ucr.ac.cr/pdfs_10/10-art_11.pdf). [En línea] [Citado el: 27 de Noviembre de 2011.]

**Barros, Pablo Straud.** *Requisitos de Software*. Santiago de Chile: Universidad Diego Portales : s.n., 2005.

**Boehm, B. M.** *Characteristics of Software Quality* . Nueva York: North Holland : s.n., 1979.

**Campistrous, Luis.** *Indicadores e Investigaciones Educativa*. Instituto Central de Ciencias Pedagógicas de Cuba : s.n.

**Crosby, Philip.** *La Calidad del Software* . [En línea] [Citado el: 5 de Diciembre de 2011.]  
<http://www.softqanetwork.com/philip-crosby-y-la-calidad-del-software>.

**D.C.C.Muñoz.** *MÉTRICAS DEL SOFTWARE: Conceptos básicos, definición y formalización*. 2006. Vol. 13.

**Dávila, D.** *Ingeniería de Requerimientos*. 2001.

*Estándares de Calidad para el Desarrollo y Mantenimiento de Software*. [En línea] Escuela de Computación e Informática. Universidad de Costa Rica. San Pedro, Costa Rica. [Citado el: 13 de Diciembre de 2011]

[http://www.eici.ucm.cl/Academicos/R\\_Villarroel/descargas/calidad.produccion/Tutorial.calidad.M.Jenkins.pdf](http://www.eici.ucm.cl/Academicos/R_Villarroel/descargas/calidad.produccion/Tutorial.calidad.M.Jenkins.pdf).

**García, M.B.** *Estimación de costes de un proyecto informático*. pág. 60.

[http://www.eici.ucm.cl/Academicos/R\\_Villarroel/descargas/calidad.produccion/Tutorial.calidad.M.Jenkins.pdf](http://www.eici.ucm.cl/Academicos/R_Villarroel/descargas/calidad.produccion/Tutorial.calidad.M.Jenkins.pdf).

**Garzón, Darwin Jiménez.** *Ingeniería de Software II*.

**Giraldo, O.P.** *Métricas, Estimación y Planificación de Software*. pág. 10.

**Gloria Ponjuán Dante.** *Gestión de Información: Dimensiones e Implementación para el éxito organizacional*. NUEVO PARHADIGMA EDICIONES. [En línea] 2004. [Citado el: 14 de Enero de 2012.]  
[http://ftp.ceces.upr.edu.cu/centro/repositorio/Textuales/Articulos/IT\\_Gestion%20de%20informacion.pdf](http://ftp.ceces.upr.edu.cu/centro/repositorio/Textuales/Articulos/IT_Gestion%20de%20informacion.pdf).

**Gonzalez, C.** *Conceptos Generales de calidad total*. [En línea] 2005. [Citado el: 4 de Diciembre de 2011.] <http://www.monografias.com/trabajos11/conge/conge.shtml>.

*Guía para una Gestión basada en Procesos*. 84-923464-7-7. Andalucía, España : Instituto Andaluz de Tecnología, 2006.

**H. Listone, M. Turoff.** *The Delphi Method. Techniques and Applications*. Addison-Wesley : s.n., 1975,

pág. 3.

**I. Brito, A. Moreira and J. Araújo.** A requirements model for quality attributes. [En línea] Universidad Twente, 22 de Abril de 2002. [Citado el: 25 de Noviembre de 2011.] <http://www.utwente.nl/AOSD-EarlyAspectsWS/Papers/Brito.pdf>.

**IEEE Standard 1990.** Terminology, IEEE Standard Glossary of Software Engineering. s.l. : Institute of Electronics Engineers, 1990, Vol. 610.2.

**IEEE Standard for Software Reviews-IEEE Std.1028-1999,** [En línea] [Citado el: 20 de Marzo de 2012.] <http://pesona.mmu.edu.my/~wruslan/SE2/Readings/detail/Reading-6.pdf>.

**IEEE. 610-1990,** Calidad de Software. [En línea] Enero de 2011. [Citado el: 17 de Diciembre de 2011.] [http://www.ecured.cu/index.php/Calidad\\_de\\_Software](http://www.ecured.cu/index.php/Calidad_de_Software).

**Informática Siglo21.** Documento de Especificación Requerimientos No Funcionales del Proyecto Mejoramiento de procesos, Análisis y Diseño del Sistema de Información para la Vigilancia de los eventos en salud pública en la Fase 1. Colombia : s.n., 2006.

**Engineers, Institute of Electrical and Electronics. IEEE 830** Recommended practice for software requirements specification. [En línea] 1998. [Citado el: 12 de Febrero de 2012.] <http://www.computer.org/>.

**ISO 9000** (Sistemas de gestión de la Calidad).

**ISO, 2001.** Ingeniería de Software. Calidad del Producto (ISO 9126). 2001, Norma Cubana, Vols. Parte 1: Modelo de Calidad(ISO/IEC 9126-1:2001).

**Jon Landeta, J.M.D.A, V. Ruiz Herrano, Villarreal Larringa.** Alimentación de modelos Cuantitativos con Información subjetiva: Aplicación Delphi en la elaboración de un modelo de imputación del gasto turístico individual en Catalunya. 2002, págs. 175-196.

**L. C. Briand, J. W. Daly y J. K. Wust.** A Unified framework for Coupling Measurement in Object-Oriented Systems. 1999. págs. 91-121. Vol. IEEE transactions on software engineering .

**Landeta, Jon.** El método Delphi. Una técnica de previsión para la incertidumbre. 1999.

**Lawrence Chung y otros.** Non-Functional Requirements in Software Engineering. ISBN: 0792386663, London: Kluwer Academic Publishers : s.n., 1999.

**Leishman, T.** Requirements risks can draw software projects. 2002.

**M.J. Escalona, N. Koch.** Ingeniería de requisitos en aplicaciones para la web. Un estudio comparativo. Master en Ingeniería de Requisitos- A distancia. Asignatura: Validación de requisitos. Modulo II: Requisitos de Sistemas de Software.

**Norma Cubana ISO/IEC, 9126-1,** Tecnología de la Información. Características de calidad y métricas del software. 2005, Vol. Parte 1: Las características y subcaracterísticas de calidad.

**Oliva, MSC. Patricio Mella Fabián.** Contrucción de lista de chequeo. Chile : Gobierno de Chile, 2009.

**Olsina, D.L.** Métricas e Indicadores: Dos conceptos Claves para la Medición y Evaluación. 2003. pág.

52.

**Pechúan, Ingacio Gil.** Sistemas y Tecnologías de la Información para la Gestión. [En línea] 2006. [Citado el: 15 de Enero de 2012.]

**Pressman, Roger.** *Ingeniería del Software. Un enfoque Práctico.* 5ta. 1998.

**Pressman, Roger.** *Ingeniería del Software. Un enfoque práctico.* [ed.] McGraw-Hill. s.l. : Interamericana de España, 2002. págs. 184-186. Vol. 2.

**Pressman, Roger.** *Ingeniería de software: Un enfoque práctico.* 5ta. La Habana : Editorial Félix Valera, 2005.

**Robertson, Suzanne Robertson y James.** *Mastering the Requirements Process.* Inglaterra: Pearson : s.n., 1999.

**Robles, M.C..E.A.** *Métricas para la Gestión de Proyectos de Software.* pág. 65.

**Ruth Malan, Dana Brendmeyer.** resources for Software Architects. Defining Non-Functional Requirements. [En línea] 2001. [Citado el: 10 al 25 de Noviembre de 2011.] [http://www.bredemeyer.com/pdf\\_files/NonFunctReq.PDF](http://www.bredemeyer.com/pdf_files/NonFunctReq.PDF).

SlideShare Inc. All rights reserved. [En línea] 2009. [Citado el: 5 de Enero de 2012.] <http://www.slideshare.net/ivonnetrabajosocial/gestion-de-la-informacion-sigeuis-3477938>.

**Sommerville, Ian.** *Ingeniería de Software.* [ed.] IV. Séptima Edición. Addison-Wesley : s.n., 2005.

**Validación y Verificación.** slideshare. [En línea] 28 de Septiembre de 2008. [Citado el: 2012 de Marzo de 20.] <http://www.slideshare.net/yosamunar/validacion-y-verificacion-presentation>.

**Vázquez, Roberto Hugo.** Taller de Calidad de Software: Introducción a la Calidad de Software. [En línea] [Citado el: 15 de Diciembre de 2011.] <http://gridtics.frm.utn.edu.ar/docs/Introduccion%20a%20la%20Calidad%20de%20Software%20Vazquez.pdf>.

Visual Paradigm International. [En línea] 2005. [Citado el: 20 de Enero de 2012.] <http://www.visualparadigm.com/product/vpuml/provides/>.

**Wieggers, Karl Eugene.** *More About Software Requirements: Thory Inssues and Practical Advice.* s.l. : Microsoft Press, 2006.

**Young, Ralph Rowland.** *The Requirements Engineering Handbook.* London: Artech House : s.n., 2004.

## GLOSARIO DE TÉRMINOS

**CALISOFT:** Centro de Calidad para Soluciones Tecnológicas.

**CMMI:** Modelo de Madurez y Capacidad Integrada.

**DATEC:** Centro de Tecnologías de Gestión de Datos

**DNC:** Documento de No conformidades.

**ERS:** Especificación de Requisitos de Software.

**Funciones de diagnóstico:** Funciones implementadas en el software para la detección y manejo de fallos. Permite monitorear y en algunos casos controlar la funcionalidad del hardware, como: computadoras, servidores y periféricos, según el tipo y sus funciones. Posibilitan al sistema operar de forma aceptable incluso después de la aparición de un fallo, debido a que son capaces de parar el proceso antes de que se origine daños irreparables.

**IA:** Índice de aceptación.

**IEEE:** Instituto de Ingenieros Eléctricos y Electrónicos.

**IR:** Ingeniería de Requisitos.

**ISO:** International Organization for Standardization (Organización Internacional para Estandarización).

**NC:** No conformidad.

**RF:** Requisitos funcionales.

**RNF:** Requisitos no funcionales.

**TIC:** Tecnologías de la Información y la Comunicación.

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

**UML:** Lenguaje para modelamiento unificado.