



# **Trabajo de Diploma para optar por el título de Ingeniero en ciencias Informáticas.**

## **Facultad 3.**

**Título:** Procedimiento para la validación y verificación de los requerimientos no funcionales en aplicaciones web de gestión.

### **Autor:**

Ana Inés Novella Machado.

### **Tutores:**

Ing. Giselle Almeida González

Ing. René Rodrigo Bauta Camejo.

Ciudad de la Habana, Junio 2012.

“Año 54 de la Revolución”.

## **DECLARACIÓN DE AUTORÍA**

Declaro ser autora 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 firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2012.

-----  
**AUTOR**

Ana Inés Novella Machado.

-----  
**TUTOR**

ING. Giselle Almeida González.

-----  
**TUTOR**

ING. René R. Bauta Camejo.



*"Nunca consideres el estudio como una obligación sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber."*

*Albert Einstein.*

## **DEDICATORIA**

*La realización de este gran sueño se lo dedico a dos personas muy importantes en mi **VIDA**, que aunque no estén presente en este momento, me han dado las fuerzas de seguir adelante y conducirme por el camino correcto.*

*Gracias por tu amor y cariño **PAPITO**.*

*Y a ti **ABUELO** gracias por quererme y malcriarme tanto.*

**LOS QUIERO MUCHO A LOS DOS.**

## **AGRADECIMIENTOS**

*A mi mamita linda por su cariño incondicional de madre, por ser mi amiga y compañera en todos los momentos de mi vida, por tu apoyo y comprensión, por aconsejarme y brindarme confianza, por ser un ejemplo como persona para mí... Aquí está tu niña convertida en ingeniera.*

*A mi papá y a mi abuelo Carlos sé que van a estar a mi lado para ayudarme en lo que sea necesario, y que siempre me han sabido guiar por el buen camino. Los quiero mucho a los dos.*

*A mi hermanito Rafe, gracias por quererme y soportarme tanto, estudia para que pueda alcanzar y ser lo desea, tú eres mi niño lindo.*

*A mi abuela Neguito, hay abue gracias por ser mi abuelita malcriada, por tu apoyo incondicional en todo momento, gracias por cuidarme, a ti te debo gran parte de lo que soy.*

*También quiero agradecer a mis tíos Ana Luisa y Enrique y a mis primos Isabel, Rita, Eugenio y Jose por el apoyo he recibido de su parte.*

*A Eulalia, a Sijo e Iliana por su apoyo incondicional a mi madre y a mi familia.*

*A mis tutores René y Giselle y a mi oponente Yunet, que no por ser los últimos son menos importante, al contrario me han apoyado y ayudado en todos los problemas, le agradezco todo el tiempo para que esta tesis pudiera tener éxito, sin ustedes no lo hubiera logrado, Gracias.*

*Quisiera agradecer a mi familia en general por su ayuda y apoyo para que este sueño se hiciera realidad.*

*A mis compañeros y amigos que de una forma u otra me han ayudado y apoyado.*

*A Fidel, Raúl y a la Revolución porque sin ellos no habría sido posible que este sueño se hiciera realidad. Muchas gracias.*

## **Resumen**

El presente trabajo de diploma hace un estudio de cómo se puede asegurar y evaluar la calidad de los requerimientos no funcionales en los sistemas web de gestión, debido a la necesidad que tiene actualmente el CEIGE en sus proyectos productivos, en los cuales no existe un procedimiento para evaluar estas características no funcionales. Se describen brevemente aspectos vinculados con este tema como la calidad del software, los requerimientos, tanto funcionales como no funcionales, herramientas de modelado, normas, estándares internacionales y se hace un resumen de cada herramienta de prueba teniendo en cuenta sus características y entorno de prueba.

Además se define un procedimiento estructurado y argumentado que provee los pasos y las acciones necesarias para conseguir un buen diseño de las pruebas de requerimientos no funcionales del software. También en él se identifican cuáles son sus entradas y salidas, roles y artefactos a generar y se describen de forma detallada las actividades a realizar en las diferentes fases de desarrollo, garantizando una mejora en la organización así como la calidad del software.

Por último se realiza la validación del procedimiento a través de la aplicación del método Delphi con el juicio intuitivo de un grupo de expertos demostrando que los objetivos propuestos se cumplieron.

Palabras clave: Calidad, Procedimiento, Requerimientos no funcionales, Validación.

# Índice de contenido

<b>INTRODUCCIÓN .....</b>	<b>- 1 -</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....</b>	<b>- 5 -</b>
INTRODUCCIÓN .....	- 5 -
1.1 CONCEPTOS GENERALES .....	- 5 -
1.2 REQUERIMIENTOS NO FUNCIONALES.....	- 8 -
1.2.1 <i>Clasificaciones de los requerimientos no funcionales.</i> .....	- 9 -
1.3 NORMAS Y ESTÁNDARES PARA LA V&V DE RNF. ....	- 12 -
1.4 TÉCNICAS PARA LA VALIDACIÓN DE REQUERIMIENTOS. ....	- 17 -
1.5 HERRAMIENTAS PARA VALIDACIÓN Y VERIFICACIÓN. ....	- 18 -
1.5.2 <i>Tipos de herramientas para pruebas de software.</i> .....	- 18 -
1.6 LENGUAJE Y NOTACIÓN DE MODELADO. ....	- 20 -
1.6.1 <i>Lenguaje UML (Larman, 2002).</i> .....	- 20 -
1.6.2 <i>BPM, nuevo paradigma (IBERMATICA, 2008).</i> .....	- 21 -
1.7 HERRAMIENTAS CASE.....	- 21 -
1.7.1 <i>Visual paradigm</i> .....	- 22 -
1.7 CONCLUSIONES.....	- 22 -
<b>CAPÍTULO 2: PROPUESTA DEL PROCEDIMIENTO .....</b>	<b>- 23 -</b>
2.1 INTRODUCCIÓN.....	- 23 -
2.2.1 <i>Objetivos</i> .....	- 23 -
2.2.2 <i>Alcance.</i> .....	- 24 -
2.3 MODELADO DE LAS FASES DEL PROCEDIMIENTO.....	- 24 -
2.3.1 <i>Selección de los roles.</i> .....	- 25 -
2.3.2 <i>Modelado y descripción por fases.</i> .....	- 26 -
<i>Fase inicio.</i> .....	- 27 -
<i>Fase de ejecución.</i> .....	- 31 -
<i>Fase análisis.</i> .....	- 39 -
2.3 CONCLUSIONES.....	- 42 -
<b>CAPÍTULO 3: VALIDACIÓN DEL PROCEDIMIENTO .....</b>	<b>- 43 -</b>
3.1 INTRODUCCIÓN.....	- 43 -
3.2 MÉTODOS DE EXPERTOS.....	- 43 -
3.3 MÉTODO DELPHI. ....	- 44 -
3.4 APLICACIÓN DEL MÉTODO DELPHI.....	- 45 -
3.5 FORMULACIÓN DEL PROBLEMA. ....	- 45 -
3.6 ELECCIÓN DE EXPERTOS.....	- 46 -
3.7 ELABORACIÓN Y LANZAMIENTO DEL CUESTIONARIO. ....	- 49 -
3.7.1 <i>Establecimiento de la concordancia entre los expertos.</i> .....	- 49 -

3.8 DESARROLLO PRÁCTICO Y ANÁLISIS DE LOS RESULTADOS. ....	- 52 -
3.9 CONCLUSIONES.....	- 56 -
<b>CONCLUSIONES GENERALES .....</b>	<b>- 57 -</b>
<b>RECOMENDACIONES.....</b>	<b>- 58 -</b>
<b>REFERENCIA BIBLIOGRÁFICA.....</b>	<b>- 59 -</b>
<b>BIBLIOGRAFÍA.....</b>	<b>- 61 -</b>

## Índice de figuras

FIGURA 1: CLASIFICACIONES DE REQUERIMIENTOS NO FUNCIONALES (SOMMERVILLE, 2005). .....	- 9 -
FIGURA 2: MODELO PARA LA CALIDAD INTERNA Y EXTERNA (ISO_9126, 2005). .....	- 13 -
FIGURA 3: FASES DEL PROCEDIMIENTO.....	- 24 -
FIGURA 4: PLANTILLA DE MODELADO DE LA FASE INICIO. ....	- 27 -
FIGURA 5: PLANTILLA DE MODELADO DE LA FASE EJECUCIÓN.....	- 31 -
FIGURA 6: MODELADO DE LA ACTIVIDAD SELECCIONAR DISEÑO DE CASO DE PRUEBA. ....	- 34 -
FIGURA 7: MODELADO DE LA FASE ANÁLISIS. ....	- 39 -
FIGURA 8: ANALISIS DE LAS NC CONTRA LAS ITERACIONES.....	- 41 -
FIGURA 9: IMPACTO DE LAS NC EN LOS TIPOS DE PRUEBAS. ....	- 42 -
FIGURA 10: COEFICIENTE DE COMPETENCIA DE LOS EXPERTOS.....	- 49 -
FIGURA 11: PUNTOS DE CORTES.....	- 55 -

## Índice de tablas

TABLA 1: COMPARACIÓN DE LAS HERRAMIENTAS. ....	- 19 -
TABLA 2: ROLES Y ACTIVIDADES. ....	- 25 -
TABLA 3: DESCRIPCIÓN DE LAS ACTIVIDADES DE LA FASE INICIO. ....	- 28 -
TABLA 4: PRINCIPALES ACTIVIDADES DE LA FASE INICIO. ....	- 30 -
TABLA 5: DESCRIPCIÓN DE LAS ACTIVIDADES DE LA FASE EJECUCIÓN. ....	- 32 -
TABLA 6: PRINCIPALES ACTIVIDADES DE LA FASE EJECUCIÓN.....	- 33 -
TABLA 7: DESCRIPCIÓN DE LAS PRUEBAS CON LA HERRAMIENTAS JMMETER.....	- 36 -
TABLA 8: DESCRIPCIÓN DE LAS ACTIVIDADES DE LA FASE ANÁLISIS. ....	- 39 -
TABLA 9: PRINCIPALES ACTIVIDADES DE LA FASE ANÁLISIS.....	- 41 -
TABLA 10: GRADO DE CONOCIMIENTO DEL EXPERTO. ....	- 47 -
TABLA 11: FUENTES PARA ARGUMENTAR EL CONOCIMIENTO DE LOS EXPERTOS. ....	- 47 -
TABLA 12: COEFICIENTE DE EXPERTOS.....	- 48 -
TABLA 13: DATOS OBTENIDOS DE LOS EXPERTOS. ....	- 50 -
TABLA 14: FRECUENCIA ABSOLUTA DE LAS PREGUNTAS DE LAS ENCUESTA.....	- 52 -
TABLA 15: FRECUENCIA ABSOLUTA ACUMULADA. ....	- 53 -
TABLA 16: FRECUENCIA RELATIVA ACUMULADA.....	- 53 -
TABLA 17: PUNTO DE CORTES. ....	- 54 -
TABLA 18: GRADOS DE ADECUACIÓN. ....	- 55 -

## **Introducción**

Con el avance de las Tecnologías de la Información y las Comunicaciones (TIC), las empresas pasaron a utilizar sofisticados sistemas de información para soportar sus procesos operativos y de gestión. En cada empresa han sido desarrollados sistemas para satisfacer las necesidades particulares de las diferentes áreas del negocio, sin embargo a veces estas soluciones se encuentran con procesos no definidos y requerimientos cambiantes (Reyes, 2010).

Los proyectos de desarrollo de software han padecido tradicionalmente problemas de calidad, tanto en el propio proceso de desarrollo como en los productos que entregan. Esta problemática tiene su origen en las habituales desviaciones de plazos y esfuerzo sobre los valores previstos y en la frecuente aparición de fallos durante la implantación y operación de los productos resultantes. Uno de los principales problemas radica en que las empresas no dedican recursos ni la atención necesaria para que la actividad tenga éxito y también los esfuerzos que se hacen en su mayoría son retroactivos, es decir, se detectaron en la recta final del producto. Para muchas organizaciones, la información y la tecnología que la soporta representan los activos más valiosos de la empresa; es más, en el competitivo, rápido y cambiante ambiente actual, la gerencia ha incrementado sus expectativas relacionadas con la entrega del servicio y la calidad en cada uno de los sistemas de información (Lebrún, 2008).

El desarrollo del software es uno de los mayores adelantos científicos a nivel mundial. Para Cuba la industria del software es una de las prioridades en el sector de las nuevas tecnologías, puesto que no se necesitan grandes plantas industriales ni grandes importaciones para desarrollar software; su principal ventaja está dada porque es una industria del conocimiento y para lo cual Cuba ya dispone de un capital humano muy bien formado, graduados de varias universidades entre ellas la Universidad de las Ciencias Informáticas (UCI).

Esta universidad vincula la producción con el desarrollo de software, el cual ha incrementado su competitividad a través de la difusión, la mejora continua y el conocimiento de las tecnologías de información. Hoy se orientan los esfuerzos hacia el entrenamiento, diagnóstico, consultoría y formación según la demanda de los proyectos productivos de la universidad. Se espera elevar la calidad del software que se produce, garantizando y certificando la calidad de los productos, e implantar mecanismos que permitan mejorar la calidad de los procesos de desarrollo de software. En su constante desarrollo se

promueve la investigación y la búsqueda de soluciones de los principales problemas en el área de ingeniería y la calidad de software.

En la UCI se encuentra el Centro de Informatización de la Gestión de Entidades (CEIGE), el cual cuenta con un grupo de gestión de calidad que garantiza que el software cumpla con los requerimientos definido por el cliente. Una de las funciones de este departamento es la realización de pruebas de software, que son efectuadas por especialistas y probadores. Estos realizan diferentes tipos pruebas en las cuales se le identifican no conformidades a los productos. A pesar de la realización de estas pruebas todavía existen grandes dificultades en la validación y verificación de los requerimientos no funcionales, debido a la mala planificación en las primeras fases de desarrollo del proyecto al no contar con un estándar de trabajo para realizar las mismas.

Durante la realización de las pruebas se detectan grandes problemas de rendimiento, carga, fiabilidad y seguridad, lo que en ocasiones provocan fallas potenciales, las cuales son manifestaciones catastróficas o de incapacidad del sistema y de alguno de sus componentes. Todo esto trae consigo un atraso en la culminación de los proyectos, debido a que estos problemas que se podían detectar desde un principio ahora se vuelven más complejos para la mayoría de los módulos, provocando una insatisfacción de los clientes por las demoras y los gastos de recursos.

Los requerimientos no funcionales (RNF) forman una parte importante en el logro de la calidad de los productos de software, principalmente para que los usuarios puedan valorar las características no funcionales del producto. Cumpliendo con todas las funcionalidades definidas, entonces las propiedades no funcionales, como cuál usable, seguro, conveniente y agradable puede ser un software, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

Otro problema frecuente es la captura de requerimientos, que no siempre se realiza por los mismos especialistas ni estos se rigen por una guía, realizando su registro en documentos de cientos de páginas (Rumbaugh, 2000). Los requerimientos no son siempre obvios y vienen dados por distintas fuentes, no son fáciles de expresar textualmente, además hay muchos tipos de requerimientos con distintos niveles de detalle, su número puede ser inmanejable si no se controla adecuadamente, no todos son igualmente importantes, ni tienen la misma dificultad, cambian constantemente y pueden ser dependientes del tiempo (Chaves, 2006).

En el proceso de elaboración del software los requerimientos no funcionales (RNF) no son evaluados con anterioridad a la entrega final del producto, es decir que todos aquellos errores que se han cometido en la etapa de requerimiento tienen luego una inmensa repercusión y son altamente perjudiciales; ante un error, omisión o mala interpretación en los requerimientos, por lo que los desarrolladores se han visto obligados a rehacer todo el trabajo sobre la base de los requerimientos incorrectos, provocando un arduo y costoso trabajo de reprogramación para corregirlos. Se hace necesario entonces evaluar los RNF, pues al transcurrir el tiempo las sucesivas evaluaciones suministran una valiosa información permitiendo desarrollar un producto más acertado, mejorar costos y satisfacer en mayor medida las necesidades del cliente, asegurando el cumplimiento en los plazos establecidos (Giraldo, 2007).

Otra de las dificultades es que los desarrolladores no le prestan igual atención a los RNF que a los requerimientos funcionales (RF) y además un estudio realizado en la universidad arrojó como resultado que no existe un procedimiento que valide y verifique las características no funcionales para elevar la calidad de los productos

Es por ello que de la problemática antes descrita se identificó el siguiente **problema resolver**: ¿Cómo garantizar la validación y verificación de los RNF en el CEIGE para disminuir la ocurrencia de fallas potenciales del software?

Obteniendo como **objeto de estudio**: Proceso de validación y verificación de los RNF.

Enmarcado en el **campo de acción**: Proceso de validación y verificación de los RNF para los productos del CEIGE.

Para resolver el problema anterior se plantea como **objetivo general**: Proponer un procedimiento para la validación y verificación de los RNF para disminuir la ocurrencia de fallas potenciales del software. El cual se desglosó en los siguientes **objetivos específicos**:

- Analizar las tendencias actuales en la validación y verificación de los RNF.
- Analizar normas, estándares, procedimientos y herramientas para la validación y verificación de los RNF.
- Definir el procedimiento para la validación y verificación de los RNF.
- Validar la propuesta de procedimiento para la validación y verificación de los RNF.

La investigación parte de la siguiente **Idea a Defender**: Si se define un procedimiento para la validación y verificación de RNF en el CEIGE se podrá reducir la existencia de fallas potenciales en el software.

A lo largo de esta investigación se utilizarán un conjunto de métodos científicos los cuales se describen a continuación:

**A nivel teórico:**

Métodos de **análisis-síntesis**: Para el estudio de los conceptos empleados dentro de la aplicación de pruebas no funcionales en los proyectos, analizando todos los documentos para la extracción de los elementos más importantes sobre el tema en cuestión.

Análisis **histórico-lógico**: Para conocer, con mayor profundidad los antecedentes y las tendencias actuales referidas a la aplicación de pruebas no funcionales en los proyectos de software sabiendo así la trayectoria histórica que tiene a través de su origen.

Para una mejor comprensión del trabajo, el mismo se estructuró de la siguiente manera:

Capítulo 1: Se realiza la fundamentación teórica del trabajo, donde se describen los conceptos fundamentales, las clasificaciones de los RNF, normas, estándares, herramientas de validación usadas en la actualidad, el lenguaje y herramientas para el modelado del procedimiento.

Capítulo 2: Se describe la solución propuesta basada en normas y estándares descritos en el capítulo 1, y además se obtienen los roles, artefactos, las técnicas y las herramientas seleccionadas para automatizar las pruebas, así como la definición del procedimiento.

Capítulo 3: Se enfocará a la evaluación de los resultados obtenidos a través de la valoración técnica del método multicriterio **Delphi** con la utilización de criterios de expertos.

## Capítulo 1: Fundamentación Teórica

### Introducción.

En la actualidad el desarrollo de software se ha convertido por sus múltiples aplicaciones en uno de los elementos fundamentales de la informática y de las comunicaciones. La creciente necesidad de construir sistemas complejos en cortos períodos, con menores esfuerzos tanto humanos como económicos y sobre todo, satisfacer las expectativas del cliente, es un proceso complicado. Debido que con el compromiso de entregar el producto final en tiempo, se pasa por alto un aspecto importante, la calidad, la cual está estrechamente vinculada con todo el ciclo de vida del software.

En el transcurso de este capítulo se definen varios conceptos útiles para la investigación, como son calidad, pruebas de software, validación y verificación. Además se hace referencia a las normas, estándares, técnicas y herramientas para la validación y verificación de RNF.

### 1.1 Conceptos generales.

#### Calidad:

- “Concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente” (Pressman, 2010).
- “El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas” (Lovellette, 1999).
- “Grado con el cual el cliente o usuario percibe que el software satisface sus expectativas” (Cahuich, 2009).

Después de haber analizado estos conceptos se puede decir que la calidad del software es un conjunto de cualidades que caracterizan y determinan la utilidad y existencia del software, además varía de un sistema a otro.

#### Prueba:

- Es una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos específicos, los resultados son observados y registrados. Las mismas no pueden asegurar

la ausencia de defectos; solo pueden demostrar que existen defectos en el software. Para lograr un mejor entendimiento se proponen los siguientes conceptos de prueba tomados de diferentes fuentes (ISO-9000, 2005).

- La prueba de un sistema se define como el proceso de ejercitar o evaluar el sistema por medios manuales o automáticos, para verificar que satisface los requerimientos o para identificar diferencias entre los resultados esperados y los que producen el sistema (IEEE, 1990).
- “Las pruebas del software son el proceso de ejecución de un programa con la intención de descubrir un error” (Espinosa, 2008).

En la práctica las pruebas de software son actividades necesarias que se realizan sobre un programa informático para verificar su correcto funcionamiento, así como la existencia de errores de diversas índoles, por lo que sin dudas les corresponden uno de los flujos más importantes en el ciclo de desarrollo de un software.

## **Requerimientos:**

- “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” (Chaves, 2006).
- “Un requerimiento es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste” (Sommerville, 2005).

Los requerimientos de software pueden clasificarse en 2 categorías: RF y RNF.

Los RF son los que definen las funciones que el sistema será capaz de realizar.

Los RNF tienen que ver con características que de una u otra forma puedan limitar el sistema (Chaves, 2006).

Analizando las definiciones anteriores, se establece en la investigación que un requerimiento es una descripción de una condición o capacidad que debe cumplir un sistema. Se caracteriza por ser:

- Especificado por escrito: Como todo contrato o acuerdo entre dos partes.
- Posible de probar o verificar .Si un requerimiento no se puede comprobar, entonces ¿Cómo se sabe si se cumplió con él o no?

- Conciso: Un requerimiento es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.
- Completo: Un requerimiento está completo si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información para su comprensión.
- Consistente: Un requerimiento es consistente si no es contradictorio con otro requerimiento.
- No ambiguo: Un requerimiento no es ambiguo cuando tiene una sola interpretación. El lenguaje usado en su definición, no debe causar confusiones al lector (Chaves, 2006).

## **Fallas potenciales:**

Una falla es la manifestación percibida por el cliente de que algo no funciona correctamente e impacta su percepción de la calidad (SG, 2008).

Siempre que se habla de fiabilidad del software, surge la pregunta fundamental: ¿Qué se entiende por el término de falla? En el contexto de cualquier discusión sobre calidad y fiabilidad del software, la falla es cualquier falta de concordancia con los requerimientos del software. Estas pueden ser simplemente desconcertantes o catastróficas. Puede que una falla sea corregida en segundos, mientras que otra lleve semanas o incluso meses. Para complicar más las cosas, la corrección de ellas pueden llevar a la introducción de otros errores que, finalmente lleven a mil fallas (SG, 2008).

Una falla potencial es una manifestación catastrófica o la incapacidad del sistema y de alguno de sus componentes para realizar las funciones requeridas que afecten directamente software.

Se pueden encontrar distintos tipo de fallas potenciales en las aplicaciones web como son: La sobrecarga por la transferencia de datos entre cliente servidor, esto se debe por las solicitudes de muchos pedidos a la vez y por la entrada de datos al servidor, ya que a veces no existe la suficiente disponibilidad de espacio en disco para realizar estas operaciones. También se encuentran los mensajes de error, que ocurren cuando se inhabilita cierta parte de la aplicación.

En ocasiones se encuentran dificultades con las informaciones ocultas que necesitan de una acción para que sea visualizada, o que el sitio no se actualiza periódicamente ni indica cuando.

Además se encuentras grandes problemas de seguridad, enlaces rotos o que no lleven a ningún sitio y el software no puede ser modificado para los ambientes especificados sin aplicar acciones o medios.

## **Validación y Verificación (V&V).**

La V&V de requerimientos es una etapa primordial en el desarrollo del producto, estas funcionan en la aplicación de métodos estáticos y dinámicos que han convertido a estos procesos en focos de atención para las empresas que buscan un lugar cimero en el competitivo entorno de la industria de software.

Mientras que la *validación* puede aplicarse a todos los aspectos del producto en cualquiera de sus entornos previstos, tales como los servicios de operación, de formación, de fabricación, de mantenimiento y de soporte; su propósito es demostrar que un producto o componente del producto se ajusta a su uso previsto cuando se sitúa en su entorno (Chrissis, 2009).

La *verificación* es inherentemente un proceso incremental, debido a que ocurre durante todo el desarrollo del software, comenzando con la verificación de los requerimientos y progresando a través de la verificación de los productos de trabajo según van evolucionando y culminando en la verificación del software finalizado. Su propósito es asegurar que los productos de trabajo seleccionados cumplen sus requerimientos especificados (Chrissis, 2009).

De forma general la validación demuestra que el producto tal como se proporciona, compensará su uso previsto; mientras que la verificación comprueba si el producto de trabajo refleja apropiadamente los requerimientos especificados; aunque la V&V usan enfoques similares como son las pruebas, el análisis, la inspección y la demostración o simulación, estas áreas de procesos tratan aspectos diferentes por ejemplo: La validación demuestra que el producto, tal como se proporciona (o tal como se proporcionará), se ajustará a su uso previsto, mientras que la verificación trata sobre si el producto de trabajo refleja apropiadamente los requerimientos especificados.

## **1.2 Requerimientos no funcionales.**

Los requerimientos no funcionales (RNF), como se referenciaba anteriormente, son aquellas propiedades emergentes del sistema, algunas de ellas 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 y las representaciones de datos que se utilizan en las interfaces del sistema (Sommerville, 2005).

## 1.2.1 Clasificaciones de los requerimientos no funcionales.

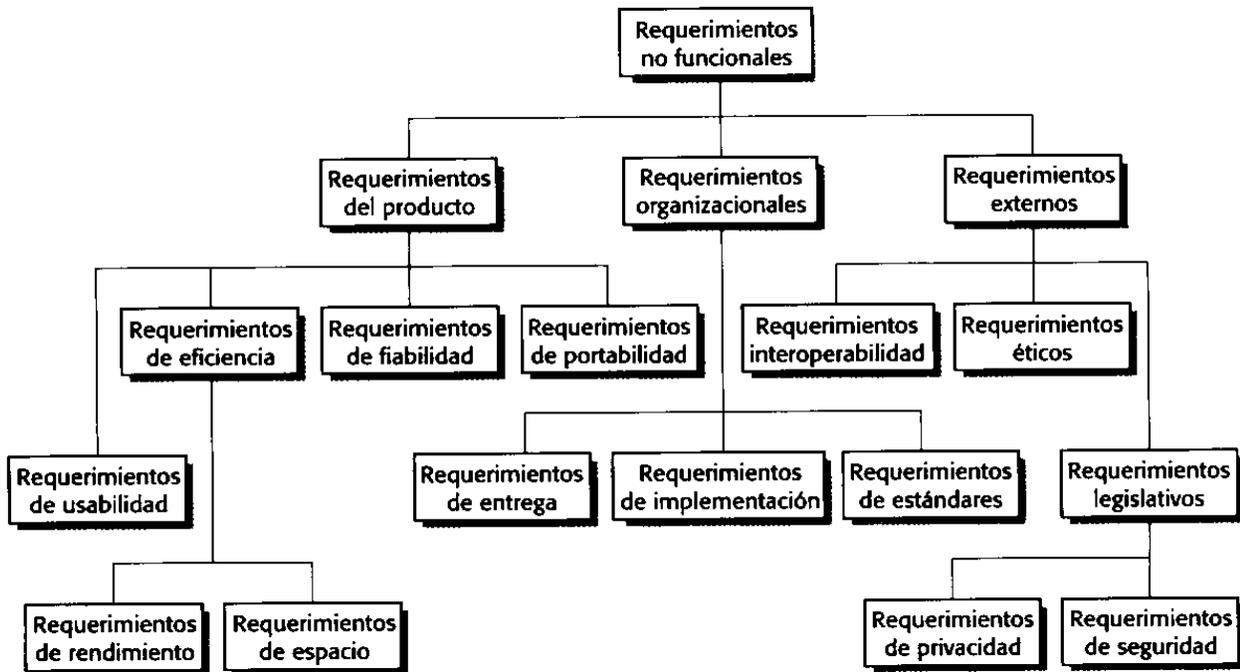


Figura 1: Clasificaciones de requerimientos no funcionales (Sommerville, 2005).

Se observan tres tipos de RNF:

- **Requerimientos del producto.** Especifican el comportamiento del producto; como los requerimientos de desempeño en la rapidez de ejecución del sistema y cuánta memoria se requiere.
- **Requerimientos organizacionales.** Se derivan de las políticas y procedimientos existentes en la organización del cliente y en la del desarrollador
- **Requerimientos externos.** Se derivan de los factores externos al sistema y de su proceso de desarrollo (Sommerville, 2005).

# Capítulo 1: *Fundamentación Teórica*

---

## Requerimientos del Producto.

- **Eficiencia:** La habilidad del software para poner la cantidad mínima de demanda sobre los recursos de hardware como sea posible. El sistema debe ser rápido y el tiempo de respuesta debe ser el mínimo posible.
- **Fiabilidad:** Es el tiempo medio entre fallos. Probabilidad de no disponibilidad. Tasa de ocurrencia de fallos. Disponibilidad (Sommerville, 2005).
- **Portabilidad:** Es el porcentaje de declaraciones dependientes del objetivo. Número de sistema objetivo (Sommerville, 2005).
- **Usabilidad:** El sistema debe ser intuitivo y fácil de aprender por los usuarios, su manejo debe ser eficiente.
- **Rendimiento o desempeño:** Establece una métrica que puede ser la cantidad de trabajos que debe realizar por unidad de tiempo, o los plazos que se deben cumplir en una aplicación.
- **Espacio o almacenamiento:** Estos requerimientos responden a la pregunta de qué información debe almacenar y administrar el sistema.

## Requerimientos organizacionales.

- **Entrega:** Los requerimientos de entrega garantizan la ausencia de errores y malentendidos para solicitar un presupuesto, reducir el plazo de entrega de resultados y es muy conveniente que a la hora de requerir un ensayo, se especifique muy claramente cuál es la causa o la razón por la que se solicita.
- **Implementación:** Tiene en cuenta varios aspectos como son el tipo de paradigma de programación a utilizar en el sistema, el tipo de entorno sobre el que va a trabajar, el lenguaje de programación utilizado. En caso de que se use base de datos (BD), se especificará. Además se tendrá en cuenta las herramientas con las que se va a modelar el sistema, así como la herramienta del entorno de desarrollo integrado (IDE) en la que se va a desarrollar el sistema.
- **Estándares:** Definen y describen aspectos de la *world wide web*. En término que han sido frecuentemente asociado con la tendencia de aprobar un conjunto de mejores prácticas estandarizadas para desarrollar softwares.

## Requerimientos externos.

- **Interoperabilidad:** Establecen criterios de normalización y equivalencia entre los datos de manera que estos queden identificados y puedan ser tratados de forma automática por los diferentes sistemas. Además asegura que aplicaciones creadas en diferentes orígenes puedan integrarse con facilidad para trabajar juntas.
- **Legislativos:** Describe la serie de reglamentos con los que debe cumplir el sistema. La parte legal y reglamentaria abarca: Reglas de negocio, reglamentación legal.
- **Seguridad:** La seguridad de un sistema no solo tiene en cuenta la seguridad del sistema sino, el ambiente en el que se usará el mismo. Por lo que se tiene que contemplar la seguridad física del lugar donde se usa la aplicación, los controles administrativos que se establecen para acceder al sistema y las regulaciones legales que afecta o determina su uso y que serán tenidas en cuenta si se incumplen. La seguridad puede ser tratada en tres aspectos diferentes:
  - ✓ **Confidencialidad:** La información está protegida de acceso no autorizado y divulgación. El sistema debe ser capaz de mantener la calidad de los datos de manera que garantice su integridad durante su aplicación. Si el sistema se conecta a una base de datos (BD), cada vez que se detecta un cambio en la BD se debe actualizar de manera automática los cambios para asegurar que los equipos de respaldo se mantengan al día. Si el sistema tiene interfaz con otros sistemas externos, debe permitir restablecer comunicación con otros tras la ocurrencia de una falla.
  - ✓ **Integridad:** La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, de la misma forma será considerada igual a la fuente o autoridad de los datos. Pueden incluir también mecanismos de chequeo de integridad y realización de auditorías.
  - ✓ **Disponibilidad:** Los usuarios autorizados tienen acceso a la información, y los dispositivos o mecanismos utilizados para lograr la seguridad no ocultan ni retrasan a los usuarios para obtener los datos deseados en un momento dado.

Los requerimientos no funcionales son todos muy importantes ya que de una forma u otra enmarcan el entorno donde se desenvolverá el software. Esta investigación se enfocará hacia los RNF del producto, ya

que estos proveen las condiciones de fiabilidad, eficiencia, usabilidad, rendimiento, portabilidad y espacio que deberá cumplir el software para que sea efectivo.

### 1.3 Normas y estándares para la V&V de RNF.

En la industria del software se emplean varios estándares de calidad que a continuación se nombran:

- CMMI describe las buenas prácticas de las áreas de proceso de V&V (Chrissis, 2009).

Prácticas específicas para esta área de proceso de validación:

- ✓ *La práctica específica de seleccionar los productos para la validación permite la identificación del producto o componente de producto para ser validado y los métodos que serán usados para llevar a cabo la validación.*
- ✓ *La práctica específica de establecer el entorno de validación permite la determinación del ambiente que será usado para llevar a cabo la validación.*
- ✓ *La práctica específica de establecer los procedimientos y criterios de validación permite el desarrollo de los procedimientos y criterios de validación que están alineados con las características de los productos seleccionados, con las restricciones del cliente en la validación, con los métodos y el entorno de validación.*
- ✓ *La práctica específica de ejecutar la validación permite la ejecución de la validación de acuerdo con los métodos, procedimientos y criterios.*

Prácticas específicas para esta área de proceso de verificación:

- ✓ *La selección del producto de trabajo para la verificación, permite la identificación del producto de trabajo para ser verificado, los métodos que se usan para realizar la verificación y los requerimientos que deben ser satisfechos (o que deben cumplir) por cada producto de trabajo seleccionado.*
- ✓ *El establecimiento del entorno de verificación permite la determinación del ambiente que se usará para llevar a cabo la verificación.*

# Capítulo 1: Fundamentación Teórica

✓ El establecimiento de los procedimientos y criterios de verificación permite el desarrollo de los procedimientos y criterios de verificación en línea con los productos de trabajo seleccionados, requerimientos, métodos, y las características del entorno de verificación.

✓ La ejecución de la verificación conduce la verificación acorde a los métodos permitidos, los procedimientos y criterios.

- NC-ISO/IEC 9126-1: 2005 (Ingeniería de Software-Calidad del Producto: Modelo de la Calidad) (ISO\_9126, 2005): Este apartado define el modelo de calidad interna y externa. Categoriza los atributos de calidad del software, da las definiciones para cada característica de la calidad y las sub-características del software, además muestra las métricas de la calidad, el uso que miden y hasta qué punto un producto satisface las necesidades de los usuarios específicos.

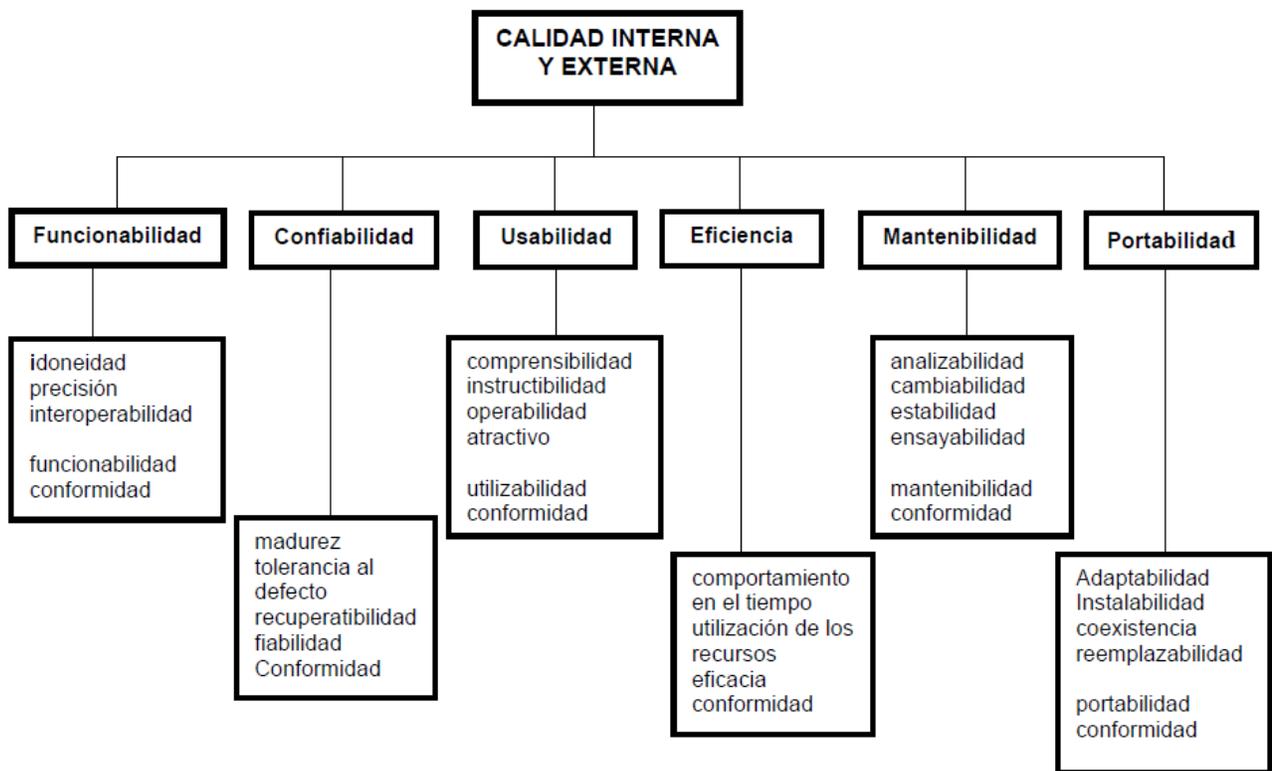


Figura 2: Modelo para la calidad interna y externa (ISO\_9126, 2005).

## **Funcionalidad**

- ✓ La idoneidad.

La capacidad del software para mantener un juego apropiado de funciones para las tareas especificadas y los objetivos del usuario.

- ✓ La precisión.

La capacidad del software para proporcionar efectos o resultados correctos o convenientes.

- ✓ La interoperabilidad.

La capacidad del software para actuar recíprocamente con uno o más sistemas especificados.

- ✓ La seguridad.

La capacidad del software para proteger la información y los datos, para que personas o sistemas desautorizados no puedan leer o modificar los mismos y a las personas o sistemas autorizados no les sea denegado el acceso a ellos.

- ✓ La conformidad.

La capacidad del software para adherirse a las normas que se le apliquen, convenciones, regulaciones, leyes y las prescripciones similares.

## **Confiabilidad**

- ✓ La madurez.

Capacidad del software de evitar una avería como resultado de haberse producido un fallo del software.

- ✓ La tolerancia ante fallos.

Capacidad del software de mantener un nivel de ejecución específico en caso de fallos del software o de infracción de sus interfaces especificadas.

- ✓ La restaurabilidad o posibilidad de recuperación.

Capacidad del software de restablecer su nivel de ejecución y recobrar los datos directamente afectados en caso de avería.

# Capítulo 1: Fundamentación Teórica

---

- ✓ La conformidad.

La capacidad del software para adherirse a las normas que se le apliquen, convenciones, regulaciones, leyes y las prescripciones similares.

## **Usabilidad**

- ✓ La comprensibilidad.

La capacidad del software para permitirle al usuario entender si el software es conveniente y cómo puede usarse para las tareas particulares.

- ✓ La cognoscitibilidad.

La capacidad del software para permitirle al usuario aprender su aplicación.

- ✓ La operabilidad.

La capacidad del producto del software para permitirle al usuario operarlo y controlarlo.

- ✓ La atracción.

La capacidad del software de ser amigable para el usuario.

- ✓ La conformidad.

La capacidad del software para adherirse a las normas que se le apliquen, convenciones, regulaciones, leyes y las prescripciones similares.

## **Eficiencia**

- ✓ El crono-comportamiento.

La capacidad del software para proporcionar una respuesta apropiada, los tiempos de procesamiento y tasas de "throughput" al realizar su función bajo condiciones declaradas.

- ✓ La utilización de recursos.

La capacidad del software para usar los recursos apropiados en un plazo de tiempo adecuado cuando el software realiza su función bajo las condiciones declaradas.

- ✓ La conformidad.

# Capítulo 1: *Fundamentación Teórica*

---

La capacidad del software para adherirse a las normas que se le apliquen o convenciones que se relacionan con la eficiencia.

## **Mantenibilidad**

- ✓ La diagnóstibilidad.

La capacidad del software de ser diagnosticado para detectar deficiencias, causas de fracasos y averías.

- ✓ La mutabilidad.

La capacidad del software para permitir llevar a cabo una modificación especificada.

- ✓ La estabilidad.

La capacidad del software para minimizar los efectos inesperados de las modificaciones del software.

- ✓ La contrastabilidad.

La capacidad del software para permitir validar el software modificado.

- ✓ La conformidad.

La capacidad del software para adherirse a las normas que se le apliquen o convenciones que se relacionan con la mantenibilidad.

## **Portabilidad**

- ✓ La adaptabilidad.

La capacidad del software de ser modificado para los ambientes especificados sin aplicar acciones o medios.

- ✓ La instalabilidad.

La capacidad del software ser instalado en un ambiente especificado.

- ✓ La coexistencia.

La capacidad del software para coexistir con otro software independiente en un ambiente común, que comparte recursos comunes.

- ✓ La remplazabilidad.

La capacidad del software de ser usado en lugar de otro software especificado en el ambiente de ese software.

- ✓ La conformidad.

La capacidad del software para adherirse a las normas que se le apliquen o convenciones que se relacionan con la portabilidad.

- *IEEE Standard for Software Verification and Validation*. IEEE Std 1012-1998 (IEEE, 1998): El propósito de esta norma es establecer un marco común para los procesos de V&V, las actividades y tareas en apoyo de todo ciclo de vida del software. Además de:

1. Facilitar la detección temprana y corrección de errores de software.
2. Mejorar la gestión de conocimiento en el proceso y el riesgo del producto.
3. Apoyar los procesos del ciclo de vida del software para asegurar el cumplimiento con el desempeño del programa, horario y presupuesto.

Haciendo un estudio de estas normas se han tomado varios aspectos que sirven de ayuda a la realización del procedimiento, por ejemplo de la NC-ISO/IEC 9126-1: 2005 se toman los comportamientos de los atributos de calidad con las respectivas definiciones de característica y sub-características que plantea el mismo. Mientras que de CMMI se analizan las buenas prácticas de las áreas de proceso de V&V para ayudar a definir las actividades del procedimiento y así contribuir a la eficiencia del software.

## **1.4 Técnicas para la validación de requerimientos.**

Las técnicas de validación de los requerimientos se realizan con el fin de examinarlos para asegurar que definen el sistema adecuado. Dichas técnicas tienen gran importancia, ya que permiten detectar los errores de forma temprana, con el fin de no conducir a resultados inesperados evitando provocar gastos excesivos y grandes pérdidas de tiempo.

### **1.4.1 Listas de Chequeo.**

Definen un listado de preguntas para los atributos de calidad en forma de cuestionario. Estas se aplican para verificar el grado de cumplimiento de determinadas reglas establecidas con prioridad y en un fin

determinado. Además son fáciles de aplicar, resumir y comparar. Su análisis es rápido, pues consiste en verificar si existe o no un control que sea aplicable al objeto desarrollado.

Se decide aplicar esta técnica ya que permite validar y verificar algunas características no funcionales además de comprobar cuándo se está en presencia de requerimientos no funcionales del producto.

## **1.5 Herramientas para validación y verificación.**

En la actualidad el uso de las herramientas se ha hecho tan imprescindible para automatizar muchos procesos entre los que se encuentran las realizaciones de pruebas no funcionales, las cuales se encargan de detectar los problemas y las vulnerabilidades que existen en el software, así como de documentar dichos errores para ir analizándolos y comparándolos entre sí.

Se hizo indispensable para la investigación el uso de herramientas para mejorar la calidad y eficiencia del software ya que ayudan a los equipos de desarrollo a investigar los errores de software, verificar la funcionalidad de los sistemas para asegurarse de que el producto que se realiza es seguro y confiable.

### **1.5.2 Tipos de herramientas para pruebas de software.**

- Webking de Parasoft: Es una herramienta de pruebas web automatizada, que proporciona pruebas exhaustivas y análisis de los sitios y aplicaciones web para asegurar que cumplen con la fiabilidad, seguridad y metas de desarrollo necesarios para sostener su negocio con eficiencia. WebKing mejora la calidad de la aplicación web a través del ciclo de vida del software y mejora la productividad en todo el equipo (Solutions, 2012).
- Open load: Es la primera solución rápida de optimización de rendimiento basada en navegador, fácil de usar, para las pruebas de carga y stress de aplicaciones y sitios web dinámicos. Puede capturar y traducir cualquier acción del usuario de cualquier sitio web o aplicación web. Genera hasta 1000 usuarios simultáneos con el hardware mínimo. Necesita de la URL del sistema para acceder al mismo y al ser una herramienta que debe ser configurada como localhost; por lo que se comporta como proxy para dar libre acceso (Opendemand, 2012).
- JMeter: Es una herramienta Java desarrollada dentro del proyecto Jakarta que permite realizar pruebas de rendimiento y pruebas funcionales sobre aplicaciones web. JMeter permite realizar pruebas

web clásicas, también permite realizar test de *FTP*, *JDBC*, *JNDI*, *LDAP*, *SOAP/XML-RPC*, y *WebServices* (*en Beta*). Permite la ejecución de pruebas distribuidas entre distintos ordenadores para realizar pruebas de rendimiento. También activa o desactiva una parte del test, lo que es muy útil cuando se está desarrollando un test largo y se desea deshabilitar ciertas partes iniciales que sean muy pesadas o largas. Además genera casos de pruebas a través de la navegación del usuario. JMeter es una herramienta de prueba que cuenta con varios componentes que facilitan la elaboración de los escenarios de prueba, con la ventaja de simular para cada escenarios miles de usuarios (Almenares, 2008).

- Wapt Pro: Es una herramienta para cargar y estresar una aplicación web, de fácil uso, consistente, que te permite analizar el rendimiento y encontrar cuellos de botellas según distintas configuraciones. Además ofrece simulaciones precisas de la navegación realizada por un usuario, admite diferentes usuarios en un único test, lo cual es válido para las aplicaciones dinámicas y de contenidos *HTTP/SSL*, por lo que devuelve detallados informes y datos sobre los tests realizados (SGuia, 2012).

A continuación se muestra una tabla comparativa de las herramientas descritas anteriormente. En ella se muestra, la selección de la herramienta que cumple con todas las características en la que es evaluada.

Tabla 1: Comparación de las herramientas.

Tipo de herramienta	RNF que mide		Multiplataforma	Licencia libre
	Eficiencia	Fiabilidad		
Webking de Parasoft	---	X	X	X
Open load	---	X	X	X
JMeter	X	X	X	X
Wapt Pro	X	---	---	---

Con el estudio de estas herramientas de pruebas automatizadas, se llega a la conclusión que actualmente existen múltiples herramientas para realizar pruebas de fiabilidad y eficiencia sobre aplicaciones web, pero basándose en las condiciones concretas de Cuba la herramienta que se destaca por su versatilidad, estabilidad, por ser multiplataforma y tener licencia libre es Jmeter, ya que mediante ella se examinan las características no funcionales de una aplicación bajo condiciones específicas, permitiendo así analizar cómo responde el sistema, bajo la acción de una cantidad de usuarios conectados concurrentemente a

una aplicación. Además el mismo es adaptable al CEIGE, por las características que presentan los sistemas de desarrollo en el centro.

## 1.6 Lenguaje y notación de modelado.

En la actualidad existen innumerables lenguajes y notaciones de modelado que han sido diseñados para la modelar y analizar los procesos de negocio.

El lenguaje de modelado es un conjunto estandarizado de símbolos y de modos para modelar un software, mientras que la notación es el estándar gráfico para representar los procesos de negocio. En la mayoría de los casos son utilizados en combinación con una metodología de desarrollo de software para realizar la especificación del mismo y de este modo hacerlo extensivo a todo el equipo de desarrollo.

### 1.6.1 Lenguaje UML (Larman, 2002).

El lenguaje unificado de modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, además de describir la semántica esencial de lo que estos diagramas y símbolos significan (LYCOS, 2005).

Con el uso de UML se pretende alcanzar los siguientes objetivos:

- Proporcionar a los usuarios un lenguaje de modelado visual expresivo y utilizable para el desarrollo e intercambio de modelos significativos.
- Proporcionar mecanismos de extensión y especialización.
- Ser independiente del proceso de desarrollo y de los lenguajes de programación.
- Proporcionar una base formal para entender el lenguaje de modelado.
- Soportar conceptos de desarrollo de alto nivel como pueden ser colaboraciones, *frameworks*, *patterns*, y componentes.

## **1.6.2. BPM, nuevo paradigma (IBERMATICA, 2008).**

El BPM (Business Process Management), o administración de procesos de negocio, es uno de los segmentos de mercado que crece con mayor velocidad en la industria del software. Por BPM se entiende la aplicación de técnicas y herramientas de software para modelar, gestionar y optimizar los procesos de negocio de la organización.

BPMN implementa una notación de modelos para procesos. Concretamente el conjunto original de especificaciones propuestas por BPMI (Business Process Management Initiative) parte ahora del OMG (Object Management Group). Se trata de una notación gráfica de los pasos y actividades de un proceso de negocio. Además modela tanto la secuencia de actividades como los datos o mensajes intercambiados entre los distintos participantes. BPMN no está pensado para modelar aplicaciones, sino procesos que correrán dentro de dichas aplicaciones. Por ello, la salida de BPMN necesita ser expresada en algo que no sea un lenguaje de programación (BPMI).

## **1.7 Herramientas CASE.**

Las herramientas CASE<sup>1</sup> fueron desarrolladas para automatizar los procesos y facilitar las tareas de coordinación de los eventos que necesitan ser mejoradas en el ciclo de desarrollo de software.

CASE se define también como (Herramienta CASE 2006):

- Conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases.
- La sigla genérica para una serie de programas y una filosofía de desarrollo de software que ayuda a automatizar el ciclo de vida de desarrollo de los sistemas.
- Una innovación en la organización, un concepto avanzado en la evolución de tecnología con un potencial efecto profundo en la organización. Se puede ver al CASE como la unión de las herramientas automáticas de software y las metodologías de desarrollo de software formales.

---

<sup>1</sup> Computer-Aided Software Engineering es un sistema de ayuda al analista, o administrador de bases de datos. Se inicia a principios de los 80 con la introducción de la documentación asistida por computadoras y de herramientas de diagramación. Estas fueron creadas para desarrollar diagramas estructurados basados en metodología de análisis y diseño estructurado.

Actualmente existen Herramientas CASE tales como:

- Visual Paradigm para UML.
- Rational Rose.
- Enterprise Architect.
- Umbrello.
- ArgoUML.

## **1.7.1 Visual paradigm**

Visual Paradigm es una herramienta CASE, profesional que soporta la última versión de UML 2.1, es considerada como muy completa y fácil de usar, es multiplataforma y proporciona excelentes facilidades de interoperabilidad con otras aplicaciones, la misma soporta el ciclo de vida completo del desarrollo de software, como es el análisis y diseño orientados a objetos, ayuda a una rápida y mejor construcción de aplicaciones de calidad a un menor coste. Además permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas, generar documentación (Herramienta CASE 2006).

## **1.7 Conclusiones.**

La necesidad de buscar calidad absoluta en el mundo del desarrollo de software se hace cada día más importante, debido a que demuestra que la buena calidad de un producto, proporciona la reducción de los costos y facilita su mejora continua.

Es por ello que el estudio de los principales conceptos relacionados con la investigación permitió sentar las bases para el desarrollo de la misma, también la investigación sobre normas, técnicas y herramientas para la V&V de RNF concedió cuáles de estas podían aplicarse al procedimiento y el análisis de las diferentes clasificaciones de los RNF, permitió centrar la investigación en los RNF del producto.

## **Capítulo 2: Propuesta del procedimiento**

### **2.1 Introducción.**

Generalmente cuando se desarrolla un software se hace mayor énfasis en las funcionalidades: que es lo que el cliente necesita, lo que ve y lo que desea hacer. Desde el punto de vista del equipo de trabajo, es fundamental cumplir con esta funcionalidad para que el cliente se encuentre satisfecho.

Pero suele suceder que en el transcurso de dichas tareas se dejen de lado los RNF, siendo quizás estos, de igual o mayor importancia que los solicitados puntualmente por las áreas analizadas. Está demostrado que un RNF incorrecto o no relevado conduce directamente al fracaso del proyecto en alguna de sus tres aristas: como son tiempo, costos y/o funcionalidad.

La realización de las pruebas no funcionales cumple un papel muy importante en el aseguramiento de la calidad de un software, puesto que constituyen elementos de vital importancia para lograr el éxito de un producto. Es por ello que este procedimiento abordará las pruebas que aseguran la fiabilidad, usabilidad, portabilidad, rendimiento y soporte de un producto, además tiene como objetivo trazar las pautas a seguir en el proceso de pruebas, basándose en la NC-ISO/IEC 9126-1: 2005 y en las buenas prácticas que plantea CMMI en las áreas de trabajo de V&V, logrando así un buen diseño de este proceso y a su vez una buena V&V en el desarrollo de las pruebas.

### **2.2 Nombre del procedimiento.**

Procedimiento para la validación y verificación de RNF en el CEIGE.

#### **2.2.1 Objetivos.**

- Especificar las bases para la ejecución de las pruebas enfocadas a los RNF de los proyectos del CEIGE.
- Establecer una guía a seguir para la ejecución de las pruebas enfocadas a los RNF.
- Definir un procedimiento estándar para la V&V de las pruebas enfocadas a los RNF en los proyectos del CEIGE.

### 2.2.2 Alcance.

El procedimiento para validar y verificar las pruebas no funcionales de forma estandarizada está enmarcado en los proyectos de la CEIGE.

### 2.2.3. Definiciones, acrónimos y abreviaturas.

SW – Software.

HW – Hardware.

GB – Gigabyte.

MB – Megabyte.

RAM – Random Access Memory.

DBMS – Database Management System.

### 2.3 Modelado de las fases del procedimiento.

El modelado de las fases es la base para comprender mejor la operación de una documentación y publicación de los procesos buscando una estandarización en la organización. Además es una representación que permite desarrollar una descripción lo más exacta posible del proceso.



Figura 3: Fases del procedimiento.

## Capítulo 2: Propuesta del Procedimiento

- La **fase de inicio** le da apertura al procedimiento y comienza con el envío de la solicitud de pruebas por parte del jefe de proyecto a los especialistas del laboratorio de calidad para comprobar los errores y fallas que pueda contener el software. Esta solicitud es evaluada y en función del resultado se planifica la reunión de inicio para elaborar el plan de pruebas
- En la **fase de ejecución** interviene los especialistas de calidad, los probadores y otro rol invitado para realizar la ejecución de las pruebas planificadas en la fase de inicio. Como resultado se obtiene el registro de las no conformidades (NC) detectadas, el cual es entregado al jefe de proyecto para que realice la depuración de los errores y luego poder planificar las próximas iteraciones de pruebas, hasta que la aplicación quede lista para ser liberada.
- En la **fase de análisis** los especialistas realizan un análisis estadístico a los resultados obtenidos en las pruebas realizadas, para obtener experiencia de cómo prevenir errores en las próximas realizaciones del software.

### 2.3.1 Selección de los roles.

Para la aplicación de este procedimiento se realizó una selección de roles asociados a los documentos generales de los procesos que están descritos en el programa de mejora nivel 2 de CMMI. Del mismo se escogieron las responsabilidades que se efectúan hoy en el centro y se definieron otras necesarias para el mismo. También se tuvo en cuenta invitar en el proceso de ejecución y análisis de las prueba a un nuevo rol para que adquiera experiencia de cómo encontrar problemas o errores.

Tabla 2: Roles y actividades.

ROLES	RESPONSABILIDADES
<b>Jefe de proyecto</b>	<ul style="list-style-type: none"><li>• Es la persona encargada de planear, dirigir y destinar los recursos necesarios para que se ejecuten las pruebas al producto, además de ser el responsable de asegurar que el equipo de desarrollo depure los errores detectados.</li></ul>

## Capítulo 2: Propuesta del Procedimiento

<b>Diseñador de pruebas</b>	<ul style="list-style-type: none"><li>• Identificar, priorizar, seleccionar y describir los casos de pruebas y los procedimientos correspondientes.</li><li>• Definir la configuración del entorno para realizar las pruebas.</li></ul>
<b>Especialista de calidad</b>	<ul style="list-style-type: none"><li>• Elaborar lista de chequeo.</li><li>• Elaborar el plan de pruebas.</li><li>• Supervisar el trabajo de pruebas, recogiendo las NC que serán plasmadas en el registro de NC.</li><li>• Controlar, monitorear y ejecutar el plan de pruebas.</li><li>• Evaluación del proceso de pruebas.</li></ul>
<b>Probador</b>	<ul style="list-style-type: none"><li>• Se encarga de la ejecución de los Casos de pruebas que le son asignados por los especialistas.</li><li>• Registrar y documentar los defectos detectados durante la ejecución de las mismas.</li></ul>
<b>Desarrollador (rol invitado)</b>	<ul style="list-style-type: none"><li>• Presenciar el proceso de ejecución y análisis de las pruebas para identificar junto con los especialistas y probadores los errores del software.</li></ul>

### 2.3.2 Modelado y descripción por fases.

Con ayuda del libro de proceso para la planeación del proyecto perteneciente al programa de mejora de CMMI se realiza una representación gráfica de forma detallada y ordenada, de cómo se desglosan las fases del procedimiento, apreciándose con facilidad las relaciones existentes entre distintas actividades: como el análisis de cada actividad, los flujos de trabajo y roles de las mismas, así como la identificación de los artefactos generados en cada una de ellas.

# Capítulo 2: Propuesta del Procedimiento

## Fase inicio.

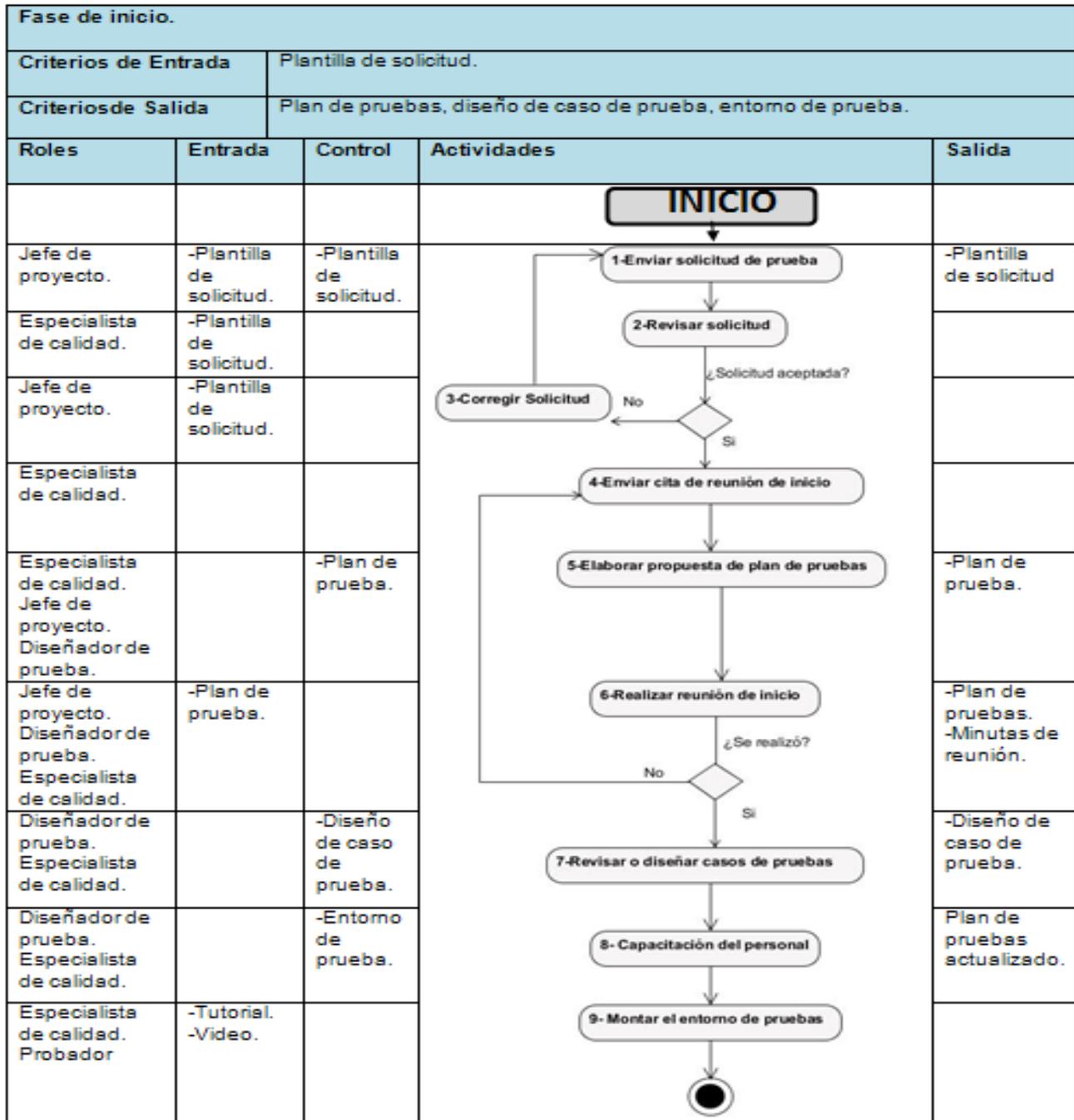


Figura 4: Plantilla de modelado de la fase inicio.

## Capítulo 2: Propuesta del Procedimiento

Tabla 3: Descripción de las actividades de la fase inicio.

Fase de inicio.		
Criterios de Entrada		Plantilla de solicitud.
Criterios de Salida		Plan de pruebas, diseño de caso de prueba.
Actividades	Descripción	Salida
1	Se realiza y se envía una plantilla de solicitud para la realización de pruebas al producto, la cual tiene que tener bien definidos todos sus parámetros como son los datos del solicitante, sus artefactos ,entorno de pruebas, entre otros (Anexo 1).	Plantilla de solicitud.
2	Se efectúa la revisión de la plantilla de solicitud y si cumple con todos los aspectos a tener en cuenta, se procede a la actividad 4. De no cumplir con los aspectos definidos se le enviaría al jefe de proyecto para que este realice la corrección de la misma (actividad 3).	
3	Esta actividad da inicio partiendo de la actividad anterior, donde la plantilla no cumple con todos los aspectos para la aceptación de la realización de las pruebas, por lo que se le es remitida al jefe de proyecto para que la corrija y luego vuelva ejecutar la actividad 1.	Plantilla de solicitud.
4	Una vez aprobada la plantilla se procede a enviar la cita de reunión de inicio.	
5	Se elabora un plan de prueba, el cual constituye el plan maestro que va a dirigir los esfuerzos de las pruebas de un producto. Este se realiza con el objetivo de esquematizar y comunicar la intención de las prueba de una planificación determinada, además de definir el alcance de las pruebas, quienes van a realizar estos ensayos, los recursos a utilizar,	Plan de pruebas.

## Capítulo 2: Propuesta del Procedimiento

	los tipos de pruebas a realizar, entre otros (Anexo 2).	
<b>6</b>	Esta actividad es muy importante debido a que se reúnen para definir o corregir el plan de pruebas, se identifican los posibles riesgos, se realiza el plan de mitigación, además de tener en cuenta las minutas que se pautaron en la reunión. En el caso de que no se pueda realizar esta reunión por algún motivo se volvería a la actividad 4.	Plan de pruebas (Anexo 2). Minutas de reunión. Plan de mitigación.
<b>7</b>	Se incorporan todas las decisiones referentes a los elementos estructurales de diseño y sus colaboraciones, necesarios para habilitar las pruebas que ya se han identificado. En esta actividad se diseñarán los Casos de pruebas correspondientes y se utilizara para la realización de los mismos la plantilla que se encuentra en el anexo 9.	Diseños de casos de pruebas.
<b>8</b>	Se revisa el entorno de pruebas que está definido en la solicitud y se purifica para ajustarlo al del laboratorio de pruebas, para así poder actualizar el plan de pruebas para los productos.	Actualización del plan de pruebas.
<b>9</b>	Se capacita al personal que va a realizar la ejecución de las pruebas, a través del tutorial de Jmeter y un video que describe un ejemplo de las realizaciones de estas pruebas.	

## Capítulo 2: Propuesta del Procedimiento

En la siguiente tabla se muestran de manera más específica las principales actividades de la fase inicio, con sus respectivos roles y artefactos.

Tabla 4: Principales actividades de la fase inicio.

Roles	Actividades	Artefactos
Jefe de proyecto	Solicitud de pruebas	Plantilla de solicitud
Especialista de calidad	Revisar solicitud	
Jefe de proyecto Diseñador de pruebas	Corregir solicitud	
Especialista de calidad	Enviar cita de reunión de inicio	
Jefe de proyecto Especialista de calidad Diseñador de pruebas	Reunión de inicio	Plan de pruebas Minutas de reunión
Especialista de calidad Diseñador de pruebas	Diseñar o revisar casos de pruebas	Diseños de casos de pruebas
Especialista de calidad Probadores	Capacitar al personal	
Especialista de calidad Diseñador de pruebas	Montar entorno de pruebas	

## Capítulo 2: Propuesta del Procedimiento

### Fase de ejecución.

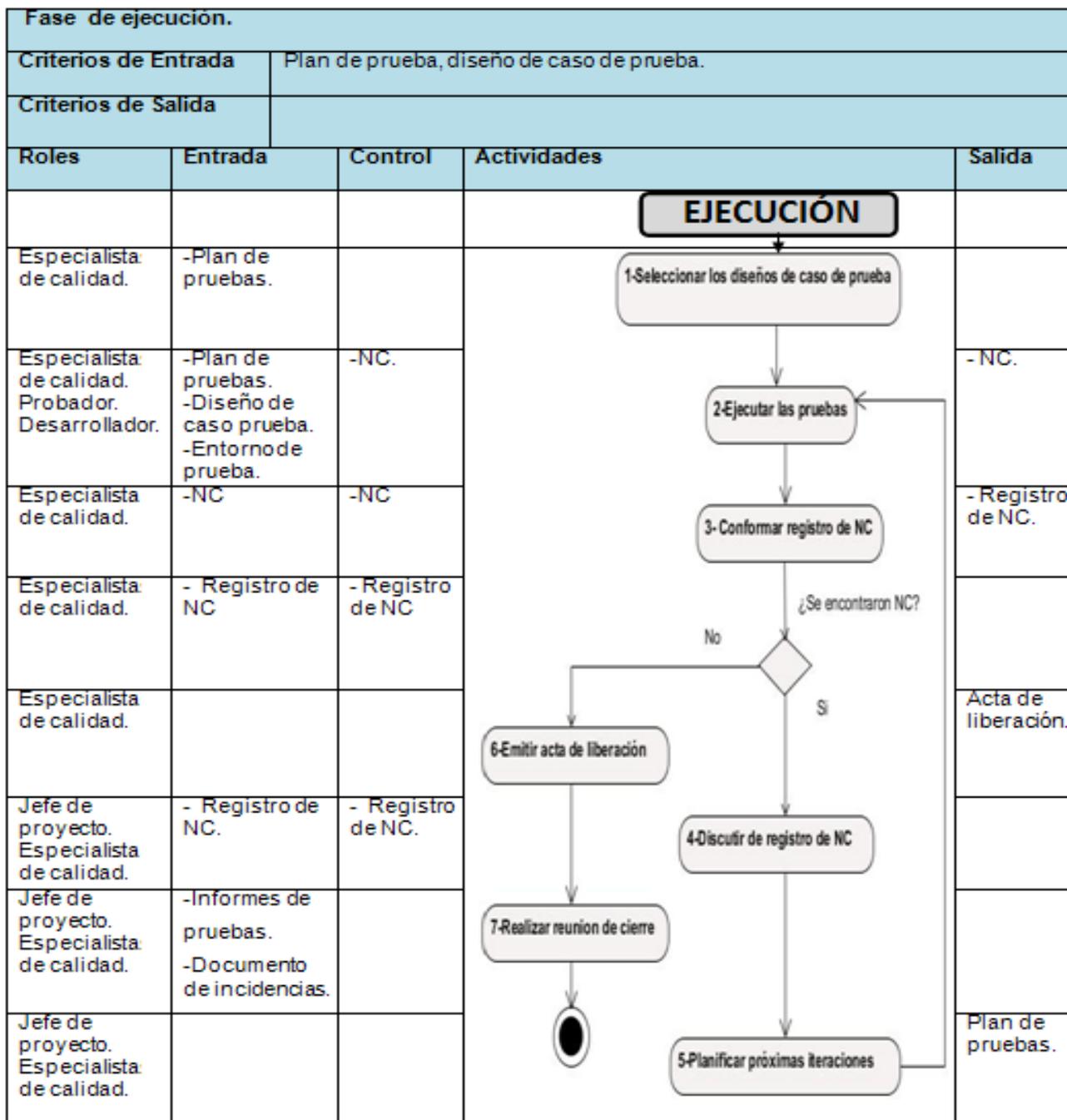


Figura 5: Plantilla de modelado de la fase ejecución.

## Capítulo 2: Propuesta del Procedimiento

Tabla 5: Descripción de las actividades de la fase ejecución.

Fase de ejecución.		
Criterios de Entrada		Plan de prueba, diseño de caso de prueba.
Criterios de Salida		
Actividades	Descripción	Salida
1	Esta fase da inicio con la actividad seleccionar los diseños de caso de pruebas correspondientes a los RNF que se van a evaluar (figura 6).	
2	Una vez seleccionado el diseño de caso de prueba se procede a la ejecución de las pruebas planificadas, a través de las técnicas de lista de chequeo o con la herramienta automatizada escogida. Esto está dado por la aplicación que tengan las características no funcionales (figura 6), para así comprobar la funcionalidad e interactividad de las mismas.	NC
3	Con la ejecución de las pruebas los especialistas recopilan todas las NC encontradas en todos los ensayos de las pruebas y la registran en el registro de NC (Anexo 7) se recolectan todas las NC encontradas para así confeccionar el registro de NC.	Platilla de NC
4	Después de tener confeccionado el registro de NC, se reúnen los especialistas con el jefe de proyecto para realizar el despacho de las NC encontradas y así el jefe de proyecto junto con su equipo de desarrollo realice las depuraciones estas.	
5	Esta actividad da inicio después de haber concluido la actividad anterior; en ella se reúnen los especialistas con el jefe de proyecto para realizar la planificación de las próximas	Plan de pruebas

## Capítulo 2: Propuesta del Procedimiento

	iteraciones de las pruebas.	
6	Cuando se le hayan realizado las pruebas al producto y no se le encuentren NC, es que se da inicio esta actividad que es donde se procede a enviar el acta de liberación del producto.	Acta de liberación (Anexo 3).
7	En esta actividad se efectúa la reunión de cierre donde se discute el informe de las pruebas y el documento de incidencias de las mismas.	

En la siguiente tabla se muestran de manera más específica las principales actividades de la fase ejecución, con sus respectivos roles y artefactos.

Tabla 6: Principales actividades de la fase ejecución.

Roles	Actividades	Artefactos
Especialistas de calidad	Seleccionar los Diseños de casos de pruebas	Plantilla de solicitud
Especialistas de calidad Desarrollador Probador	Ejecutar las pruebas	NC
Especialistas de calidad	Realizar el registro de NC	Registro de NC
Jefe de proyecto Especialistas de calidad	Discutir el registro de NC	
Jefe de proyecto Especialistas de calidad	Planificar proximas iteraciones	Plan de pruebas
Especialistas de calidad	Emitir acta de liberación	Acta de liberación
Especialistas de calidad Diseñador de pruebas	Realizar reunión de cierre	Informes de pruebas Documentos de incidencia

### 2.2.3.1 Descripción detallada de la actividad seleccionar de los diseño de caso de prueba.

En esta actividad se realiza la selección del caso de prueba que utilizará un tipo de prueba específico (figura 6) debido a la complejidad que tenga el RNF descrito en el mismo.

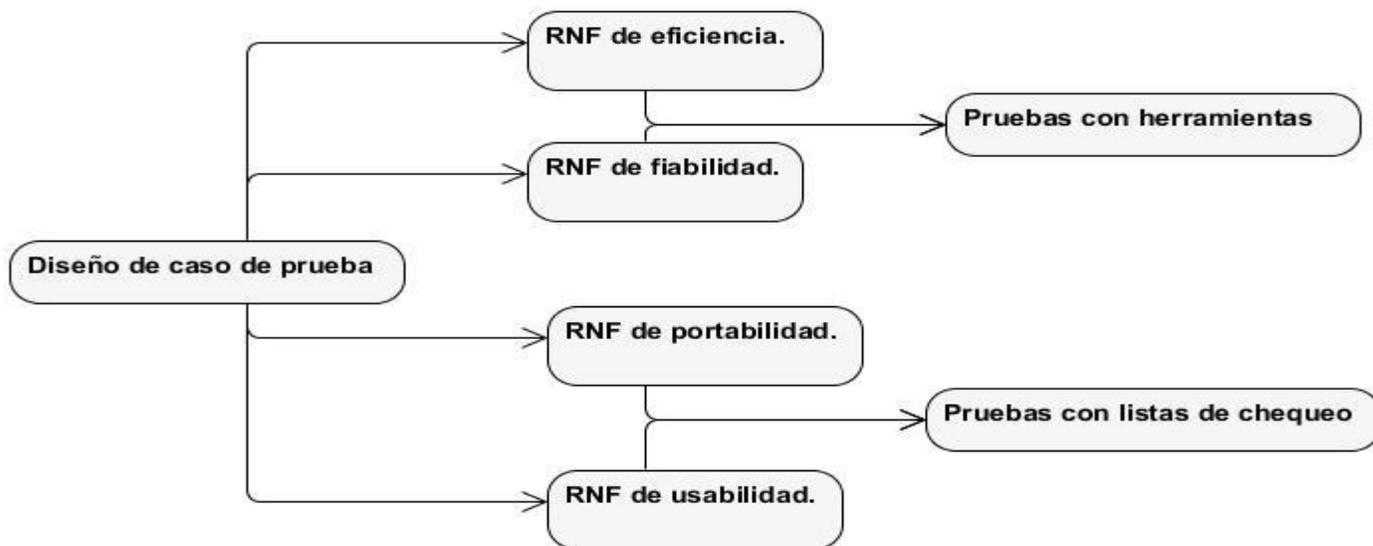


Figura 6: Modelado de la actividad seleccionar diseño de caso de prueba.

Las herramientas automatizadas brindan grandes beneficios, ya que simplifican el trabajo y el esfuerzo humano necesario para realizar cualquier tipo de tarea, además aporta velocidad, eficiencia y precisión. Pero a pesar de tener estos increíbles beneficios, se le anteponen ciertas desventajas, ejemplo, una gran cantidad de aplicaciones contienen elementos que no son compatibles con las herramientas que se utilizan para ponerlas a prueba.

Esta actividad tiene un nivel de complejidad, ya que se debe realizar un análisis de los RNF para lograr la correcta selección del tipo de prueba que se le va aplicar. Esto está dado por la aplicación de cada una de estas características no funcionales, ya que todas no tienen la misma aplicación y presentan diferentes subcaracterísticas que de una manera u otra permiten la correcta funcionalidad del software.

Los casos de pruebas que contengan descritos los RNF de fiabilidad y eficiencia son validados con la herramienta automatizada, porque sería complejo evaluar cómo el sistema responde bajo condiciones anormales como son las extremas sobrecargas, insuficiente memoria, servicios y/o hardware no

disponible, recursos compartidos no disponible, además se valoran la robustez, es decir la resistencia a fallos de la aplicación.

Mientras que los casos de pruebas referentes a los RNF de usabilidad y portabilidad se validan a través de las listas de chequeo, ya que están encaminados a los factores humanos, estéticos, consistencia en la interfaz de usuario, ayuda sensitiva al contexto o en línea, documentación de usuarios y materiales de entrenamiento. Además de asegurar que la instalación sea en diferentes configuraciones de hardware y software bajo algunas condiciones como el insuficiente espacio en disco.

### 2.2.3.2 Descripción detallada de la ejecución de pruebas con listas de chequeo.

En la ejecución de las pruebas con listas de chequeo se analizan los criterios de usabilidad y de portabilidad que están descritos en los diseños de casos de pruebas. Además a través de la ISO 9126 se elaboraron la listas de chequeo que se van a utilizar para la evaluación del producto. Luego en la realización de las pruebas se reflejarán en el registro de NC (Anexo 7) todas las anomalías y los errores encontrados.

Las listas de chequeo para la V&V de los RNF de usabilidad y portabilidad se encuentran en los anexos (Anexo 4) y (Anexo 5).

Pero a pesar de tener la lista de chequeo para validar la portabilidad, se hace necesario evaluar la soportabilidad del sistema teniendo en cuenta la configuración e instalación de las aplicaciones web. Debido a que se necesita comprobar aquellos elementos configurados para diferentes hardwares o configuraciones de software y además demostrar que el elemento a probar se instala como se indica en diferentes configuraciones de hardware y software bajos algunas condiciones como el insuficiente espacio en disco.

Las pruebas de configuración se realizan evaluando el manual de configuración frente al software, de esta forma se enfocan las diferentes variaciones de la aplicación integrada, contra sus requerimientos. Sus objetivos principales son reportar las fallas que han cometidos los desarrolladores, determinar el efecto de añadir o modificar los recursos de hardware y la configuración óptima del sistema.

Mientras que las pruebas de instalación se realizan con el objetivo de verificar que el software sea instalado correctamente bajo algunas condiciones como: el insuficiente espacio en disco, la instalación en una nueva máquina con versión anterior o superior y en diferentes plataformas.

El uso de las listas de chequeo es de gran importancia para la verificación del grado de cumplimiento de determinadas reglas establecidas a priori con un fin determinado. Además sus preguntas ayudan a identificar problemas por áreas y sirven para motivar posibles soluciones o la detección de oportunidades para la mejora.

### 2.2.3.3 Descripción detallada de la ejecución de pruebas con herramienta.

La ejecución de las pruebas con herramientas automatizadas facilita la gestión y realización de actividades de pruebas con el objetivo de verificar los requerimientos que debe cumplir el software para monitorear la ejecución de los ensayos, comparar resultados, elaborar condiciones. Además de:

- Aumentar la calidad del proceso de pruebas.
- Mayor comprensión del artefacto a probar.
- Reducción en la introducción de errores.
- Simplificar el trabajo y el esfuerzo humano.
- Facilitar la ejecución de las pruebas.
- Facilitar la interpretación de los resultados.

Tabla 7: Descripción de las pruebas con la herramientas JMeter.

<b>Requerimientos:</b> Fiabilidad y Eficiencia.	<b>Herramienta:</b> JMeter.
<b>Tipo de pruebas:</b> Carga, Stress, seguridad.	
<b>Objetivo de la prueba:</b> Verificar el comportamiento del rendimiento de determinados componentes del software bajo condiciones de trabajo diferentes, además comprueba que el software funcione apropiadamente y sin error bajo condiciones de esfuerzo, como son: <ul style="list-style-type: none"><li>• Poca memoria o sin disponibilidad de memoria en el servidor.</li><li>• Cantidad máxima de clientes conectados.</li><li>• Múltiples usuarios realizando la misma operación sobre los mismos datos.</li><li>• Peor caso de volumen de operaciones.</li></ul>	
<b>Descripción:</b> Esta prueba está enfocada a evaluar cómo el sistema responde bajo condiciones anormales. Extrema sobrecarga, insuficiente memoria, servicios y hardware no disponible, recursos	

## Capítulo 2: Propuesta del Procedimiento

compartidos no disponible. También se centra en determinar y asegurar que el sistema funciona apropiadamente en circunstancias de máxima carga de trabajo esperada. Además evalúa las características del rendimiento, como tiempos de respuesta y tiempos de transacciones.	
<b>Características técnicas para Openload.</b>	<p><u>Sistemas Operativos:</u> Multiplataforma.</p> <p><u>Licencia:</u> Libre.</p> <p><u>Código:</u> Abierto.</p> <p><u>Memoria RAM:</u> 1GB.</p> <p><u>Espacio en disco:</u> 512MB.</p> <p><u>Tipo de PC:</u> Pentium IV (1 GHz).</p> <p><u>Plataformas:</u> Máquina virtual de Java 1.3 o superior.</p> <p><u>SGBD:</u> PostgreSQL, MySQL, Oracle, SQL Server.</p> <p><u>Navegador:</u> Internet Explorer [6.x, 7.x], Mozilla Firefox [2.x, 3.x].</p> <p><u>Servidor Web:</u> Apache, Tomcat.</p>
<b>Precondiciones.</b>	<ol style="list-style-type: none"> <li>1. Se instala la máquina virtual de java.</li> <li>2. Se busca la aplicación a la cual se le van a realizar las pruebas.</li> <li>3. Se configura el JMeter como servidor proxy.</li> <li>4. Se configura el navegador como: Localhost y se arranca el servidor proxy.</li> <li>5. Se inicia la navegación por el sitio deseado.</li> <li>6. Se registra todo el tráfico desde el servidor hacia el cliente.</li> <li>7. Se guarda los resultados de la navegación.</li> </ol>
<b>Entrada.</b>	<ol style="list-style-type: none"> <li>1. Se seleccionan la cantidad de usuarios a simular.</li> <li>2. Se especifica el tiempo que va a demorar la conexión.</li> </ol>
<b>Pos condiciones.</b>	Los listeners (elementos en escucha) registran el mismo tráfico previamente realizado con la cantidad de usuarios definidos.
<b>Salida.</b>	<ul style="list-style-type: none"> <li>• En los listeners las tablas y las gráficas se muestran el tiempo de respuesta del sistema bajo las condiciones configuradas.</li> <li>• Se muestran los componentes enviados desde el servidor así como los errores que se produjeron.</li> <li>• Informa del porcentaje de rendimiento.</li> </ul>

Además para la realización del plan de pruebas que define los scripts de prueba se definen un conjunto de elementos los cuales serán utilizados en dependencia a la acción que se quiera realizar. Los elementos pueden ser:

- Elementos jerárquicos:
  - ✓ Listeners (elementos en escucha).
  - ✓ Elementos de configuración.
  - ✓ Post-procesadores.
  - ✓ Pre-procesadores.
  - ✓ Aserciones (afirmaciones).
  - ✓ Temporizador (cronómetros).
- Elementos ordenados:
  - ✓ Controladores.
  - ✓ Samplers (agentes de pruebas).

Las pruebas con esta herramienta se van realizando sobre la base de una lista ordenada de peticiones (*http*) utilizando *samplers* que representa los pasos a ejecutar en el plan de pruebas con JMeter. Normalmente, las peticiones se organizan dentro de controladores lógicos. Algunos tipos de controladores afectan el orden de ejecución de los elementos que controlan lo cual contrasta con el realismo que se quiere alcanzar en la simulación.

Apoyándose en los samplers se pueden realizar peticiones a determinado servidor, también pueden ser configurados a través de los “*elementos de configuración*”. Los controladores lógicos funcionan de manera diferente. Estos permiten controlar el comportamiento de la prueba, en ellos se modelan las decisiones en función a las necesidades del plan de pruebas. Por ejemplo, un controlador *If Controller* permite decidir si realizar o no una petición *http* en función de una condición. Cada controlador puede tener uno o más elementos por defecto.

## Capítulo 2: Propuesta del Procedimiento

### Fase análisis.

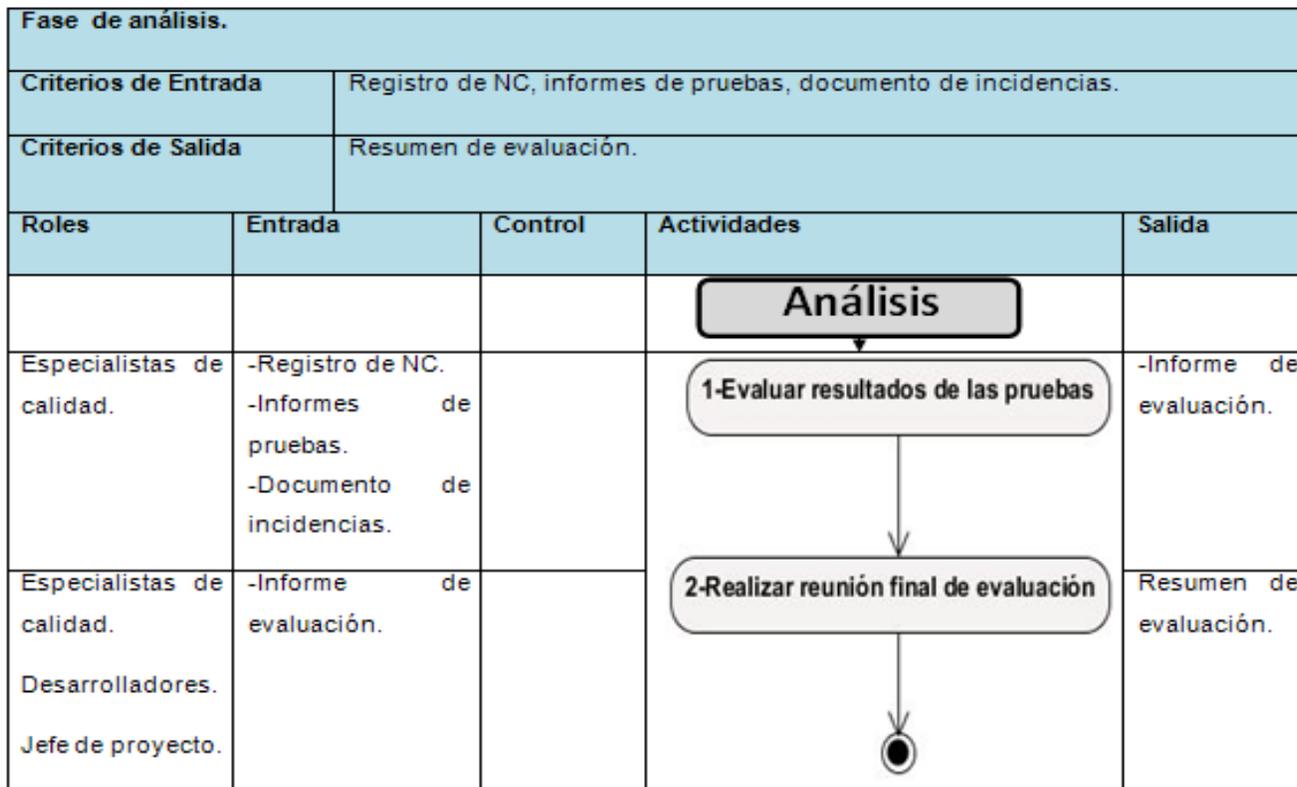


Figura 7: Modelado de la fase análisis.

Tabla 8: Descripción de las actividades de la fase análisis.

Fase de análisis.		
Criterios de Entrada		Registro de NC, informes de pruebas, documento de incidencias.
Criterios de Salida		Resumen de evaluación.
Actividades	Descripción	Salida
1	Esta actividad se realiza con el objetivo de resumir los resultados de las de prueba en un análisis estadístico, que no va ser más, que la recolección, descripción, visualización y resumen de las pruebas	Informe de evaluación.

	<p>realizadas. Estos datos son resumidos gráficamente con el fin de realizar predicciones para la fiabilidad del software, detectar causas habituales de un error y por tanto mejorar el proceso de desarrollo del software.</p> <p>También se evalúa los criterios de las pruebas descritos en el correspondiente plan de pruebas. De manera general esta actividad sirve para realizar predicciones de la fiabilidad del software y por tanto mejorar los procesos de desarrollo.</p> <p>Es por ello que si se encuentra un error, se recogen fundamentalmente toda esta información:</p> <ul style="list-style-type: none"> <li>• ¿Cuándo se cometió?</li> <li>• ¿Quién lo hizo?</li> <li>• ¿Qué se hizo mal?</li> <li>• ¿Cómo se podría haber prevenido?</li> <li>• ¿Por qué no se detectó antes?</li> <li>• ¿Cómo se podría haber detectado antes?</li> <li>• ¿Cómo se encontró el error?</li> </ul> <p>Esta información no es usada para evaluar al personal, sino para la formación del mismo sobre cómo prevenir los errores y se utiliza para predecir futuros fallos del software.</p> <p>Además todas estas informaciones son recopiladas para que queden plasmadas en un informe de evaluación.</p>	
2	<p>En esta actividad se efectúa la reunión de cierre donde se discute el informe de evaluación para verificar que todos estén de acuerdo y para que los desarrolladores cojan experiencia de cómo y dónde se cometieron errores para próximas realizaciones de software.</p>	Resumen de evaluación.

## Capítulo 2: Propuesta del Procedimiento

En la siguiente tabla se muestran de manera más específica las principales actividades de la fase análisis, con sus respectivos roles y artefactos.

Tabla 9: Principales actividades de la fase análisis.

Roles	Actividades	Artefactos
Especialistas de calidad	Evaluar resultados de las pruebas	Informe de evaluación
Especialistas de calidad Desarrollador Jefe de proyecto	Realizar reunión final de evaluación	Resumen de evaluación

Ejemplos de las gráficas del informe de evaluación.

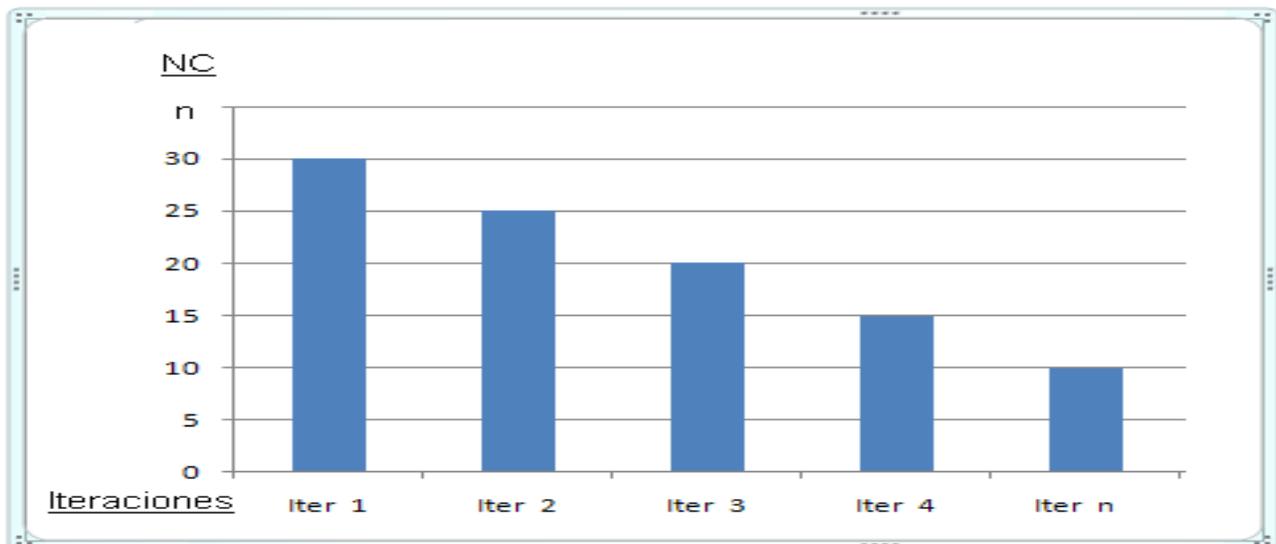


Figura 8: Analisis de las NC contra las iteraciones.

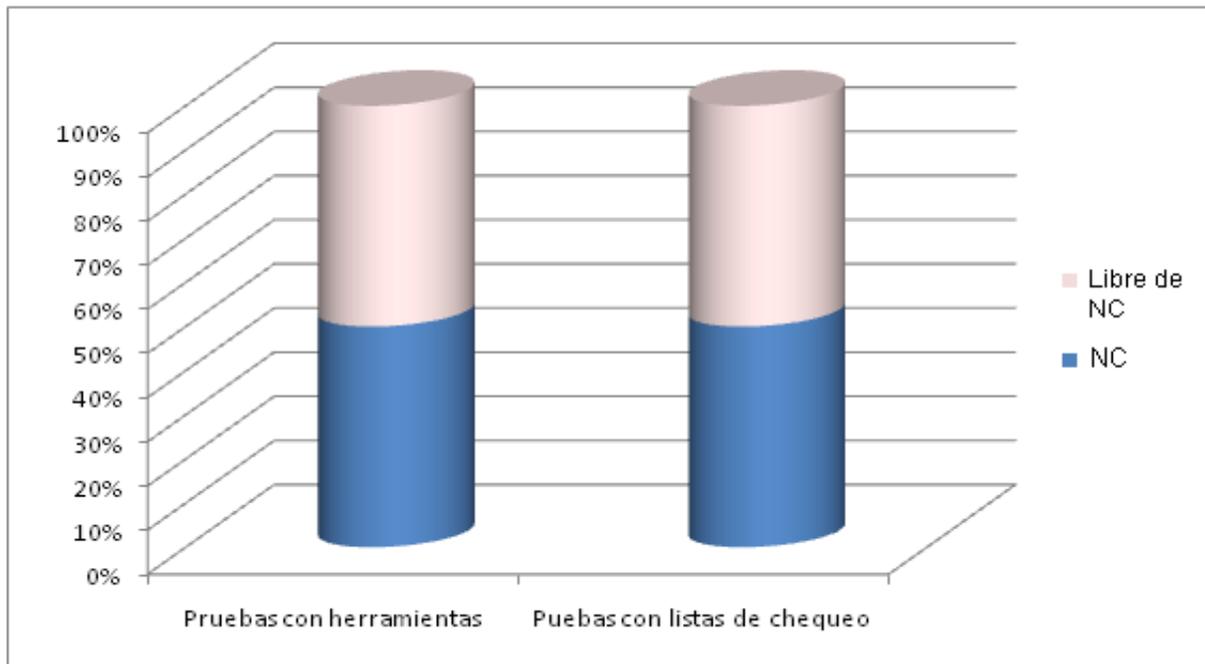


Figura 9: Impacto de las NC en los tipos de pruebas.

### 2.3 Conclusiones.

En este capítulo se definió el procedimiento estructurado que provee los pasos y las acciones necesarias para conseguir un buen diseño de las pruebas a los RNF del software para el CEIGE, donde se identifican las entradas y salidas, roles y artefactos a generar además de describir de forma detallada las actividades a realizar en las diferentes fases del proceso de desarrollo, garantizando una mejora en la organización así como en la calidad del software.

También se crean listas de chequeo y se utiliza JMeter como herramienta automatizada para la V&V de las características no funcionales, así como la inclusión de un nuevo rol en el procedimiento para que obtenga experiencia de cómo evitar errores en el desarrollo de los próximos productos.

### **Capítulo 3: Validación del procedimiento**

#### **3.1 Introducción.**

A veces se hace necesario conocer el criterio de otras personas con conocimiento para saber si lo investigado es realmente cercano a la realidad. En el presente capítulo se realiza la validación del procedimiento para la V&V de RNF en el CEIGE. Para ello se utilizó el método de consulta a especialistas (Delphi), el cual consiste en la selección de varios expertos, que ofrecen su criterio sobre la utilidad de la propuesta presentada.

#### **3.2 Métodos de expertos.**

Desde el momento en el que se procede al desarrollo de una investigación, el principal problema que surge es la posibilidad de comprobar y demostrar la fidelidad de la propuesta resultante. Para erradicar este problema es que se crean los métodos expertos, estos se basan en la consulta a un grupo de expertos para la validación de la propuesta. Los métodos expertos tienen las siguientes ventajas (Hurtado, 2012).

Se basa en la suposición de que varios expertos pueden llegar a un mejor pronóstico que una sola persona. El experto se siente involucrado plenamente en la solución del problema y facilita su implantación. De ello es importante el principio de voluntariedad del experto en participar en la investigación. Como pronóstico visionario es una profecía que usa ideas y juicios personales, vinculados entre sí.

De igual manera este método también posee desventajas y estas son (Routio, 2007):

- No siempre el argumento más válido es el que triunfa, en ocasiones es el más citado.
- Estos grupos son vulnerables a la posición y personalidad de algunos de los individuos.
- La presión social que el grupo ejerce sobre sus participantes puede provocar acuerdos con la mayoría, aunque la opinión de ésta sea errónea. Así, un experto puede renunciar a la defensa de su opinión ante la persistencia del grupo en rechazarla.

### 3.3 Método Delphi.

El método Delphi consiste en la utilización sistemática del juicio intuitivo de un grupo de expertos para obtener un consenso de opiniones informadas. Representa uno de los métodos de pronosticación más confiables siempre que se aplique siguiendo las indicaciones correspondientes, constituye un procedimiento para confeccionar un cuadro de la evolución de situaciones complejas, a través de la elaboración de las opiniones de los expertos en el tema tratado.

Este método pretende extraer las ventajas y minimizar las desventajas a través de los conocimientos de las personas que conforman el panel, evitando así la presión que puede ejercer un individuo en el grupo debido a su posición o personalidad en el grupo. Posee tres características fundamentales, explicadas a continuación:

- Anonimato: Impide la posibilidad de que un miembro del grupo sea influenciado por la reputación de otro o por el peso que supone oponerse a la mayoría. La única influencia posible es la de la congruencia de los argumentos.
  1. Permite que un miembro pueda cambiar sus opiniones sin que eso suponga una pérdida de imagen.
  2. El experto puede defender sus argumentos con la tranquilidad que da saber que en caso de que sean erróneos, su equivocación no va a ser conocida por los otros expertos.
- Iteración y retroalimentación controlada: La iteración se consigue al presentar varias veces el mismo cuestionario. Como, además, se van presentando los resultados obtenidos con los cuestionarios anteriores, se consigue que los expertos vayan conociendo los distintos puntos de vista y puedan ir modificando su opinión si los argumentos presentados les parecen más apropiados que los suyos.
- Respuesta del grupo en forma 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 (Routio, 2007).

### **3.4 Aplicación del método Delphi.**

En el método Delphi las estimaciones de los expertos se realizan en sucesivas rondas, anónimas, al objeto de tratar de conseguir consenso, pero con la máxima autonomía por parte de los participantes. Procede por medio de la interrogación a expertos con la ayuda de cuestionarios sucesivos, a fin de poner de manifiesto convergencias de opiniones y deducir eventuales consensos (Ramirez, 1999).

La encuesta se lleva a cabo de una manera anónima (actualmente es habitual realizarla haciendo uso del correo electrónico o mediante cuestionarios web establecidos al efecto) para evitar los efectos de "líderes". La calidad de los resultados depende, sobre todo, del cuidado que se ponga en la elaboración del cuestionario y en la elección de los expertos consultados. Por lo tanto, en su conjunto el método Delphi permitirá prever las transformaciones más importantes que puedan producirse en el fenómeno analizado en el transcurso de los próximos años (Ramirez, 1999).

Los pasos o fases que se llevan a cabo para garantizar la calidad de los resultados son los siguientes:

- Fase 1: Formulación del problema.
- Fase 2: Elección de expertos.
- Fase 3: Elaboración y lanzamiento de los cuestionarios.
- Fase 4: Desarrollo práctico y análisis de los resultados.

### **3.5 Formulación del problema.**

En un método de expertos, la importancia de definir con precisión el campo de investigación es muy grande por lo que es preciso estar muy seguros de que los expertos reclutados y consultados poseen todos, la misma noción en este campo.

La elaboración del cuestionario debe ser llevada a cabo según ciertas reglas: Las preguntas deben ser precisas, cuantificables (versan por ejemplo sobre probabilidades de realización de hipótesis y/o acontecimientos, la mayoría de las veces sobre datos de realización de acontecimientos) e independientes (la supuesta realización de una de las cuestiones en una fecha determinada no influye sobre la realización de alguna otra cuestión) (Ramirez, 1999).

### 3.6 Elección de expertos.

Se considerará un experto aquel individuo, grupo de personas u organizaciones que sean capaces de brindar valoraciones conclusivas sobre la propuesta, además de hacer recomendaciones con un determinado coeficiente de competencia. Los mismos no sólo deben ser conocedores del tema sino que deben presentar gran diversidad en sus planteamientos. Los criterios que se tuvieron en cuenta para la selección de los posibles expertos son reflejados a continuación:

- Graduado de nivel superior.
- Más de 3 años de experiencia.
- Conocimientos sobre proceso de pruebas.
- Que esté vinculado al desarrollo del software.
- Conocimientos de las aplicaciones Web.

Una vez conocidos los requisitos a cumplir por los expertos, se realiza una encuesta de autoevaluación para medir la fuente y el grado de conocimiento de cada uno, con el objetivo de seleccionar los que conozcan más sobre el tema que se está investigando.

#### Compañero (a):

En la presente tesis, se desea someter a la valoración de un grupo de expertos una propuesta de procedimiento para la V&V de los RNF en aplicaciones web, para garantizar una mejor calidad del software que se produce en el CEIGE. Para ello se necesita conocer el grado de dominio que usted posee sobre el tema de las de la V&V de los RNF en sistemas web de gestión y con ese fin se desea que responda lo que se le pide a continuación.

Nombre y Apellidos: \_\_\_\_\_

Rol que desempeña: \_\_\_\_\_

Años de experiencia: \_\_\_\_\_

Especialidad: \_\_\_\_\_

Categoría docente: \_\_\_\_\_

Categoría científica: \_\_\_\_\_

## Capítulo 3: Validación del Procedimiento

1.- Marque con una cruz (x) el grado de conocimiento que usted tiene sobre el tema que se lleva a cabo en esta investigación.

Tabla 10: Grado de conocimiento del experto.

1	2	3	4	5	6	7	8	9	10

2.- Marque con una cruz (x) las fuentes que le han servido para argumentar el conocimiento que tiene usted del tema que se lleva a cabo en esta investigación.

Tabla 11: Fuentes para argumentar el conocimiento de los expertos.

Grado de influencia			Fuentes de argumentación
Alto	Medio	Bajo	
			Análisis teóricos realizados por usted.
			Experiencia.
			Trabajos de autores nacionales.
			Trabajos de autores extranjeros.
			Su propio conocimiento del tema.
			Su intuición.

Para seleccionar los expertos se tiene en cuenta la valoración de sus competencias, es por esta razón que se hace necesario calcular el coeficiente de competencia (**K**) que es basado en los resultados obtenidos en las encuestas aplicadas anteriormente. Su objetivo es demostrar si el nivel de conocimientos que poseen los expertos así como las fuentes de argumentación son las adecuadas, de manera tal que garantice la confiabilidad en los resultados.

El coeficiente de competencia se calcula mediante la fórmula:  $K = 0.5 (Kc + Ka)$ , donde **Kc** es el coeficiente de conocimientos y **Ka** el coeficiente de argumentación. El **Kc** es la información que posee la persona acerca del problema (sobre la base de su autovaloración, es el resultado de la primera pregunta de la encuesta); sus valores están en una escala de 0 a 10, que para ajustarla a la teoría de las

## Capítulo 3: Validación del Procedimiento

probabilidades se multiplica por 0.1, el cero indica que la persona no posee absolutamente ningún conocimiento de la problemática en estudio, mientras que el 10 expresa pleno conocimiento.

El coeficiente de argumentación o fundamentación de los criterios de la persona se obtiene de la siguiente forma  $K_a = \Sigma$  de los valores que se obtienen de sustituir las cruces por los valores correspondientes en su posición de la tabla patrón para el cálculo del coeficiente de argumentación (Anexo 10).

Una vez calculado el  $K_c$  y el  $K_a$ , se puede calcular  $K$  mediante la fórmula descrita anteriormente. El coeficiente  $K$ , teóricamente, se encuentra siempre entre 0.25 y 1. Mientras más cercano esté el valor de  $K$  a uno, mayor es el grado de competencia de la persona. Este resultado se debe interpretar a través de la siguiente escala brindada por el método Delphi (Ramirez, 1999):

- Si  $0.8 < K < 1.0$ , el coeficiente de competencia es alto.
- Si  $0.5 < K < 0.8$ , el coeficiente de competencia es medio.
- Si  $K < 0.5$ , el coeficiente de competencia es bajo.

Para obtener mejores resultados en la aplicación del Delphi es conveniente utilizar los expertos cuyo coeficiente de competencia sea alto o medio.

Luego de realizada la encuesta de autoevaluación a los posibles expertos, solo 7 fueron seleccionados para formar parte del grupo de validación de la propuesta, pues fueron aquellos cuyos resultados arrojaron un coeficiente de competencia alto y medio. A continuación se muestra tabla 10 los resultados del coeficiente de competencia de los expertos.

Tabla 12: Coeficiente de expertos.

Experto	Kc	Ka	K	Competencia
1	0,9	0,9	0,9	Alto
2	0,7	0,9	0,8	Medio
3	0,9	1	0,95	Alto
4	0,6	0,7	0,65	Medio
5	0,7	1	0,85	Alto
6	0,8	0,8	0,8	Medio

7	0,8	0,9	0,85	Alto
---	-----	-----	------	------

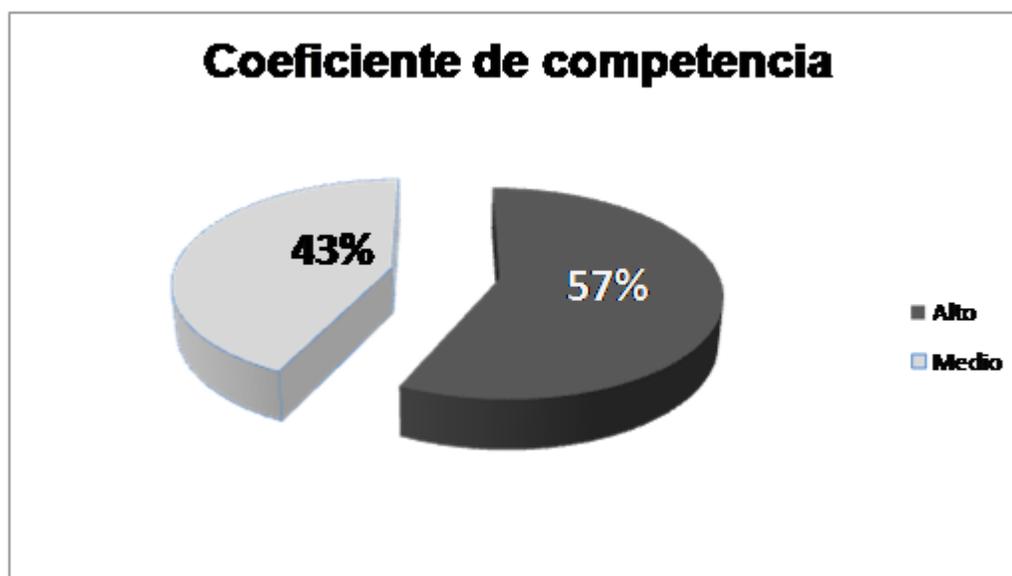


Figura 10: Coeficiente de competencia de los expertos.

### 3.7 Elaboración y lanzamiento del cuestionario.

Una vez seleccionados los expertos se elabora el cuestionario de validación de la propuesta, que tiene como objetivo principal validar los elementos fundamentales que conforman la propuesta, el mismo cuenta con diez preguntas orientadas a evaluar aspectos críticos de la solución, el cual fue realizado por los siete expertos seleccionados pidiéndosele que evaluaran en las categorías ordinales de Muy Adecuado (MA), Bastante Adecuado (BA), Adecuado (A), Poco Adecuado (PA) y No Adecuado (NA). Con las respuestas dadas por los expertos se podrá obtener la concordancia entre ellos y la aceptación de la validez de la propuesta. El cuestionario de validación utilizado en la investigación se muestra en el Anexo 8.

#### 3.7.1 Establecimiento de la concordancia entre los expertos.

Cuando se tienen datos de tipo ordinal se utiliza el coeficiente de concordancia de Kendall que es un instrumento estadístico muy adecuado para indicar el grado de asociación de las evaluaciones ordinales hechas por evaluadores múltiples cuando se evalúa la misma muestra. Los valores del coeficiente deben

## Capítulo 3: Validación del Procedimiento

oscilar entre 0 y 1. Entre mayor sea el valor Kendall, más fuerte será la asociación (Alvarado, 2008). El coeficiente Kendall (**W**) significativo implica que los evaluadores están aplicando esencialmente el mismo estándar cuando evalúan la muestra.

Valores para calcular el coeficiente de concordancia de Kendall.

K Es el número de expertos que intervienen en el proceso de validación, por tanto va ser 7.

N Cantidad de preguntas a validar. En este caso N = 10.

R<sub>j</sub> Es la suma de los rangos asignados a cada pregunta por parte de los expertos

Es la media de los rangos y se determina mediante la fórmula:

$\bar{R}_j$

Ya con todos estos datos se puede decir que **W** se calcula mediante la siguiente fórmula:

$$W = \frac{12 * S}{k^2(N^3 - N)}$$

Donde **S** es la suma de los cuadrados de las desviaciones observadas de la media de **R<sub>j</sub>** (rangos) y se calcula

$$S = \sum_{j=1}^n (R_j - \bar{R}_j)^2$$

y  $\bar{R}_j = \frac{\sum_{j=1}^n R_j}{N}$  es la suma de los rangos dividido entre la cantidad de preguntas.

A continuación se observan los resultados obtenidos de las preguntas realizadas por los expertos y se muestran los cálculos realizados para determinar la concordancia de ellos:

Tabla 13: Datos obtenidos de los expertos.

	Exp1	Exp2	Exp3	Exp4	Exp5	Exp6	Exp7	Rj
<b>P1</b>	4	3	2	4	3	3	4	23
<b>P2</b>	5	4	4	4	3	4	4	28
<b>P3</b>	5	4	4	4	4	4	4	29
<b>P4</b>	4	5	3	4	5	3	4	28
<b>P5</b>	3	4	3	4	5	4	4	27

## Capítulo 3: Validación del Procedimiento

<b>P6</b>	4	4	2	3	5	3	3	24
<b>P7</b>	5	4	5	4	4	4	5	31
<b>P8</b>	5	5	4	3	4	3	5	29
<b>P9</b>	4	4	3	3	5	4	3	26
<b>P10</b>	4	5	2	4	5	3	4	27

Sustituyendo  $\overline{R_j} = \frac{\sum_{j=i}^n R_j}{N}$  se obtiene el valor  $272/10 = 27,2$ .

Luego se determina la desviación media  $S = ((23-27,2)^2 + (28-27,2)^2 + (29-27,2)^2 + (28-27,2)^2 + (27-27,2)^2 + (24-27,2)^2 + (31-27,2)^2 + (29-27,2)^2 + (26-27,2)^2 + (27-27,2)^2) = 51,6$

Sustituyendo los valores en la ecuación Kendall se obtiene el valor de  $W = 12 * 51,6 / 7^2 * (10^3 - 10) = 619,2 / 49(990) = 0,012764$

El coeficiente de Kendall obtenido permite calcular el Chi cuadrado real, el cual tiene el objetivo de medir si existe o no concordancia entre los expertos y se obtiene a través de la fórmula:

$$X^2 = K (N - 1) W = 7(10-1) 0,012764 = 0,8041$$

Después de calcular el Chi Cuadrado se procede a comparar el valor con el de la tabla estadística (Anexo 6). Si se cumple que  $X_{real}^2 < X^2_{(\alpha, N-1)}$  entonces quiere decir que existe concordancia entre los expertos.

Teniendo en cuenta la probabilidad de error de un 0,05 se realizar los cálculos:

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

$0,8041 < X^2(0.05, 9) = 0,8041 < 16,9190$  lo cual afirma el cumplimiento de la comparación y por tanto la concordancia entre los expertos.

### 3.8 Desarrollo práctico y análisis de los resultados.

Los expertos seleccionados recibieron un resumen de la propuesta como documentación para contestar los criterios encuestados, de igual manera recibieron la encuesta con un total de 10 preguntas, estas documentaciones fueron entregadas personalmente o enviadas por vía e-mail, además se realizó una sola ronda de encuesta y luego se procedió al análisis de los resultados.

**Paso 1:** Se realiza la construcción de la siguiente tabla de frecuencias absoluta en la que se encuentran los resultados obtenidos de las encuestas evaluadas en las categorías de muy adecuado (MA), bastante adecuado (BA), adecuado (A), poco adecuado (PA), y no Adecuado (NA).

Tabla 14: Frecuencia absoluta de las preguntas de las encuestas.

Tabla de frecuencias absolutas						
No.	MA	BA	A	PA	NA	Total
1	0	3	3	1	0	7
2	1	5	1	0	0	7
3	1	6	0	0	0	7
4	2	3	2	0	0	7
5	1	4	2	0	0	7
6	1	2	3	1	0	7
7	3	4	0	0	0	7
8	3	2	2	0	0	7
9	1	3	3	0	0	7
10	2	3	1	1	0	7
Total	15	35	17	3	0	

**Paso 1:** Construir una tabla de frecuencias absoluta acumulada. Esto se realiza la sumándole a cada número, el número anterior de la fila, exceptuando al primer número de la fila.

## Capítulo 3: Validación del Procedimiento

Tabla 15: Frecuencia absoluta acumulada.

Tabla de frecuencias absolutas acumulada					
No.	MA	BA	A	PA	NA
1	0	3	6	7	7
2	1	6	7	7	7
3	1	7	7	7	7
4	2	5	7	7	7
5	1	5	7	7	7
6	1	3	6	7	7
7	3	7	7	7	7
8	3	5	7	7	7
9	1	4	7	7	7
10	2	5	6	7	7

**Paso 2:** Partiendo de tabla anterior se construye una nueva pero de **frecuencias relativas acumuladas**. En esta, los datos se obtienen mediante la división de los valores numéricos de la tabla de **frecuencia absoluta acumulada** entre el número total de expertos, en este caso serian siete.

Tabla 16: Frecuencia relativa acumulada.

Tabla de frecuencias relativa acumulada					
No.	MA	BA	A	PA	NA
1	0	0,43	0,86	1	1
2	0,14	0,86	1	1	1
3	0,14	1	1	1	1
4	0,28	0,71	1	1	1
5	0,14	0,71	1	1	1
6	0,14	0,43	0,86	1	1
7	0,43	1	1	1	1
8	0,43	0,71	1	1	1
9	0,14	0,57	1	1	1

## Capítulo 3: Validación del Procedimiento

10	0,28	0,71	0,86	1	1
----	------	------	------	---	---

**Paso 3:** Finalmente, se buscan las imágenes de los elementos de la tabla anterior por medio de la función. Estas imágenes se representan en la misma tabla anterior, sólo que se le adicionan 3 columnas y una fila para representar los siguientes resultados:

**Suma de las columnas:** Esta nueva fila recoge en cada una de sus celdas la suma de todos los valores de la columna correspondiente.

**Suma de las filas:** Esta nueva fila recoge en cada una de sus celdas la suma de todos los valores de la fila correspondiente.

**Para hallar el promedio General (N):** Se divide la suma de las sumas de las filas (la cual tiene que ser igual a la suma de las sumas de las columnas) entre el resultado de multiplicar el número de aspectos que se están evaluando por el número de preguntas, en este caso se divide por 4 porque quedan sólo cuatro categorías, ya que la última ha sido eliminada.  $N = 28,83 / 10 \cdot 4 = 0,72$

Tabla 17: Punto de cortes.

Puntos de corte						N=0,72	
No.	MA	BA	A	PA	Suma	P	N-P
1	0	0,43	0,86	1	2,29	0,57	0,15
2	0,14	0,86	1	1	3	0,75	-0,03
3	0,14	1	1	1	3,14	0,78	-0,06
4	0,28	0,71	1	1	2,99	0,75	-0,03
5	0,14	0,71	1	1	2,85	0,71	0,01
6	0,14	0,43	0,86	1	2,43	0,61	0,11
7	0,43	1	1	1	3,43	0,86	-0,14
8	0,43	0,71	1	1	3,14	0,78	-0,06
9	0,14	0,57	1	1	2,71	0,67	0,05
10	0,28	0,71	0,86	1	2,85	0,71	0,01

## Capítulo 3: Validación del Procedimiento

Suma	2,12	7,13	9,58	10	28,83		
Puntos de corte	0,21	0,71	0,96	1			

La suma obtenida de las 4 primeras columnas da los puntos de corte. Estos puntos de corte se utilizan para determinar el grado de adecuación o categoría de cada aspecto encuestado según los expertos. El grado de adecuación se muestra en la tabla siguiente:

Tabla 18: Grados de adecuación.

Muy adecuado	Bien adecuado	Adecuado	Poco adecuado	No adecuado
0,21	0,71	0,96	1	

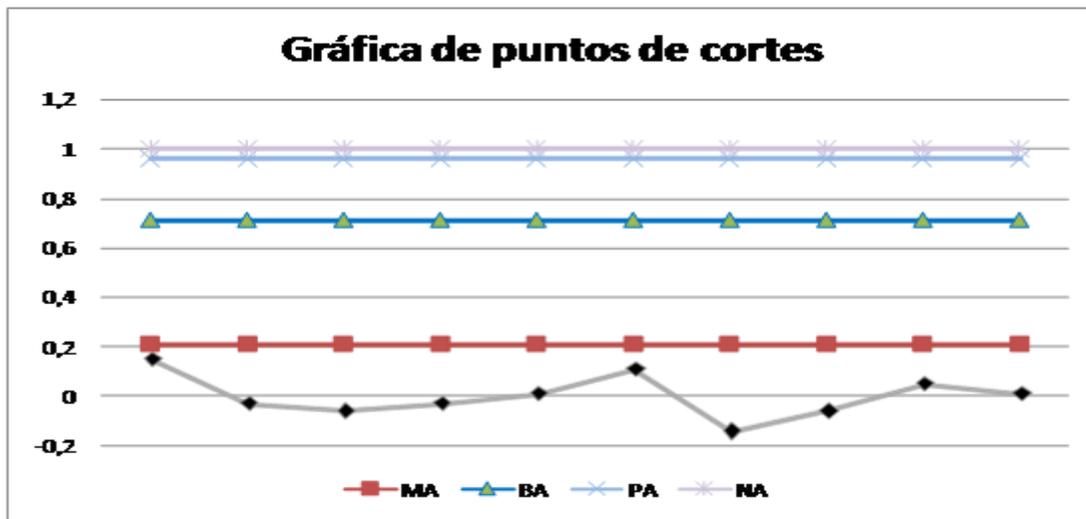


Figura 11: Puntos de cortes.

Si el valor promedio de adecuación del elemento a evaluar es:

- Menor o igual que 0,21 el nivel de adecuación es **muy adecuado**.
- Mayor que 0,21 menor o igual que 0,71 el nivel de adecuación es **bastante adecuado**.
- Mayor que 0,71 y menor o igual que 0,96 el nivel de adecuación es **adecuado**.

- Entre 0.96 y 1 el nivel de adecuación es **poco adecuado**.
- Mayor que 1 el nivel de adecuación es **no adecuado**.

Una vez aplicado el método Delphi queda demostrado que las encuestas realizadas a los 7 expertos arrojaron resultados satisfactorios:

- Todas las preguntas fueron valoradas por los expertos de muy adecuada.
- El 100% de los expertos coinciden en la utilidad que tiene la puesta en práctica de la propuesta en los proyectos del CEIGE.

### **3.9 Conclusiones.**

Las características de los expertos seleccionados para la validación del procedimiento, garantiza la calidad de las opiniones emitidas por los mismos, demostrado que los objetivos propuestos se cumplieron, como quedó demostrado en el análisis estadístico de los resultados obtenidos.

Además los encuestados coinciden en la importancia de la puesta en práctica del procedimiento y que las actividades que se definieron en él aseguran la validación y verificación de los requerimientos no funcionales.

## **Conclusiones generales**

Luego del estudio realizado para confeccionar el procedimiento de las pruebas no funcionales en los proyectos del CEIGE, se arribó a las siguientes conclusiones:

- El análisis de las tendencias actuales en la validación y verificación de los RNF, evidencia la necesidad que tiene la puesta en práctica de un procedimiento de pruebas no funcionales que contribuya a la calidad de los productos software que se desarrollan en el CEIGE.
- Con el estudio las normas, estándares, procedimientos y herramientas para la validación y verificación de los RNF se concedió cuáles de estas podían aplicarse al procedimiento.
- Se demostró la importancia de contar con un procedimiento para asegurar la máxima organización en la planificación, diseño, ejecución y evaluación de los resultados de las pruebas no funcionales, evitando que el software sea rechazado al final del ciclo de desarrollo.
- Con la validación de la propuesta a través del método Delphi se demostró la aceptación de la misma con un respaldado de 100% de coincidencia en los criterios de los expertos.

## **Recomendaciones**

Se recomienda que:

- Se aplique el procedimiento para la validación y verificación de pruebas no funcionales en los proyectos del CEIGE.
- Extender el procedimiento a todos los grupos de calidad a nivel de facultad, así como en el centro de calidad para las soluciones informáticas (Calisoft).

## Referencia bibliográfica

1. **Almenares, Liudmila Sánchez. 2008.** *Cómo realizar pruebas de carga y stres en JMeter.* Ciudad Habana : s.n., 2008.
2. **Alvarado, Federico Picado. 2008.** Análisis de concordancia de atributos. [En línea] diciembre de 2008. [http://www.tec.ac.cr/sitios/Vicerrectoria/vie/editorial\\_tecnologica/Revista\\_Tecnologia\\_Marcha/pdf/tecnologia\\_marcha\\_21-4/cap%203.pdf](http://www.tec.ac.cr/sitios/Vicerrectoria/vie/editorial_tecnologica/Revista_Tecnologia_Marcha/pdf/tecnologia_marcha_21-4/cap%203.pdf).
3. **Cahuich, Lourdes. 2009.** Calidad del Software. *SlideShare.* [En línea] 2009. [Citado el: 16 de 2 de 2010.] <http://www.slideshare.net/lcahuich/calidad-del-software-presentation>.
4. **Chaves, M.A. 2006.** *La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software.* s.l. : Volume 13, 2006.
5. **Chrissis, Mary Beth. 2009.** CMMI. *CMMI.* s.l. : Pearson Educación, 2009.
6. —. 2009. *CMMI.* 2009. Segunda .
7. **Espinosa, Susana Gonzalez. 2008.** *Estrategia para la aplicacion de pruebas de caja blanca y caja negra al proyecto Registro y Notarías.* Ciudad Habana : UCI, 2008.
8. **Giraldo, O.P. 2007.** *Ingeniería de Requisitos.* s.l. : Volume 13, 2007.
9. **Hurtado, Sandra de Mendoza Fernández. 2012.** Histodidáctica. [En línea] 2012. <http://www.ub.es/histodidactica/Epistemolog%EDa/Delphy.htm>.
10. **IEEE. 1990.** *IEEE.* 1990.
11. —. 1998. *IEEE Standard for Software.* 1998. 1012.
12. **ISO\_9126. 2005.** *INGENIERÍA DE SOFTWARE—CALIDAD DEL PRODUCTO.* 2005. 1.
13. **ISO-9000. 2005.** *ISO-9000. Fundamentos y Vocabulario.* 2005.
14. **Lebrún, C. V. 2008.** *Mejores prácticas para el establecimiento y aseguramiento de la calidad del software.* 2008.
15. **Herramienta CASE. 2006.** En línea] 2012. [Citado el: 7 de 4 de 2012.] Disponible en: <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>
16. **Lovelle, Juan Manuel Cueva. 1999.** *Calidad del Software.* España : s.n., 1999.
17. **LYCOS. 2005.** UML. [En línea] 2005. <http://usuarios.lycos.es/oopere/uml.htm>.
18. **Opendemand. 2012.** Opendemand. . [En línea] 2012. [Citado el: 7 de 4 de 2012.] <http://www.opendemand.com>.  
**Pressman, Roger S., Adaptado Dariel INCE. 2010.** *Ingeniería del software, Un Enfoque Práctico.* s.l. : Quinta Edición, 2010.
19. **Ramirez, Luis Arturo Urizarri. 1999.** Algunas consideraciones acerca del método de evaluación. [En línea] 6 de Julio de 1999. Conferencia dictada en Santafé de Bogotá. <http://www.revistaciencias.com/publicaciones/EEkZyIEFEVDEhxqKXi.php%fi>.
20. **Reyes, Ing. Julio César Álvarez. 2010.** ERP Programas integrados de gestión corporativa. [En línea] 12 de 11 de 2010. <http://www.slideshare.net/juliozet/erp-2634865>.

21. **Routio, Pentti. 2007.** Prospectiva . [En línea] 2007. <http://www.uiah.fi/projekti/metodi/290.htm>.
22. **Rumbaugh, Ivar Jacobson Graddy Booch james. 2000.** El Proceso Unificado de Desarrollo de Software. s.l. : Addison Wesley, 2000, págs. 125-163, 187-202.
23. **SG. 2008.** SG-Ingeniería de la Confiabilidad de Software. [En línea] 2008. [Citado el: 12 de mayo de 2012.] <http://www.sg.com.mx/content/view/271>.
24. **SGuia. 2012.** Wapt. [En línea] 2012. [Citado el: 7 de 4 de 2012.] <http://www.sg.com.mx/guia/node/793..>
25. —. **2012.** SGuia. [En línea] 2012. [Citado el: ] <http://www.sg.com.mx/guia/node/781..>
26. **Solutions, Applications LifeCycle. 2012.** Applications LifeCycle Solutions. [En línea] 2012. [Citado el: 7 de 4 de 2012.] <http://www.als-es.com/home.php?location=herramientas/entorno-pruebas/webking..>

## **Bibliografía**

1. **IBERMATICA. 2008.** BPM, Disponible en: <http://www.ibermatica.com/ibermatica/bpm>.
2. **Larman, C.2002.** UML y Patrones: Una introducción al análisis y diseño orientado a objetos y al Proceso Unificado.
3. **Sommerville, Ian. 2005.** Ingeniería de Software. 7ma edición. Madrid : Addison–Wesley, 2005.
4. **IPP-3510\_2009** Libro de Proceso para la Administración de Requisitos.
5. **Programa de mejora. 0515\_Roles** asociados a las Áreas de procesos del nivel 2 de CMMI.
6. **Programa de mejora. 0516\_Roles** y responsabilidades de las Áreas de procesos del nivel 2 de CMMI.
7. **CMMI.** Guía para la integración de procesos y la mejora de productos (Segunda edición).
8. **(ISO/IEC 9126)** Ingeniería de software—Calidad del producto—Parte 1: Modelado de la calidad

## **Anexo 4.**

**Introducción:** Esta lista de chequeo cuenta con varios atributos, los cuales serán evaluados teniendo en cuenta su pertinencia, además de tener un punto donde se referencia las observaciones en cuanto al aspecto a evaluar.

**Objetivo general:** Evaluar las distintas características de los atributos de calidad.

**Evaluación (Eval.):** Es la forma de evaluar el indicador en cuestión. El mismo se evalúa de valor mínimo 0 y valor máximo 5.

**Comentario:** Especifica los señalamientos o sugerencias que quiera incluir la persona que aplica la lista de chequeo.

**N.P. (No Procede):** Se usa para especificar que el indicador a evaluar no se puede aplicar en ese caso.

**Atributos de calidad:** Usabilidad

<b>Visibilidad del estado del sistema.</b>					
<b>Peso</b>	<b>Indicadores a Evaluar</b>	<b>Eval</b>	<b>(NP)</b>	<b>Cantidad de elementos afectados</b>	<b>Comentarios</b>
	El sitio tiene una URL correcta, clara y fácil de recordar.				
	Cada pantalla empieza con un título que describe su contenido.				
	Cuando se selecciona un icono se diferencia de los no seleccionados.				
	La ruta de la página en la estructura del sitio es visible.				
	No existe información oculta que necesite una acción para que se visualicen.				
	La terminología del menú es constante en todo el sitio.				

	Los enlaces del menú cambian de color cuando se seleccionan.				
	El sitio web se actualiza periódicamente e indica cuando.				
	Los enlaces cambian de color cuando se han visitado.				
<b>Adecuación entre el sistema y el mundo real.</b>					
<b>Peso</b>	<b>Indicadores a Evaluar</b>	<b>Eval</b>	<b>(NP)</b>	<b>Cantidad de elementos afectados</b>	<b>Comentarios</b>
	Los iconos que aparecen se identifican claramente con lo que representan.				
	El lenguaje es simple, con un tono adecuado.				
	La información que se presenta en la aplicación es fácil de entender y memorizar.				
	La información está organizada con categorías lógicas, fácilmente memorizables por el usuario.				
	El orden de los menús es por frecuencia de utilización.				
	La página refleja la identidad de la empresa (logos).				
	El usuario puede ponerse en contacto con el encargado del Sitio Web para hacer sugerencias o comentarios.				
	Los mensajes de error están en texto plano, entendible.				
	La información es de rápida lectura, y con una disposición asequible.				

	Existen faltas de ortografía.				
	La estructura y representación no requiere información adicional para su comprensión.				
	Existe una manera obvia de acceder a páginas relacionadas o a otras secciones.				
	El nombre de los botones de un formulario es adecuado, aplicado a la acción, no general (Ej.: Utilizar "Enviar" en vez de "OK").				
<b>Control y libertad del usuario.</b>					
<b>Peso</b>	<b>Indicadores a Evaluar</b>	<b>Eval</b>	<b>(NP)</b>	<b>Cantidad de elementos afectados</b>	<b>Comentarios</b>
	Tras una acción relevante hay una opción de vuelta atrás				
	Si una acción tiene consecuencias, el sistema proporciona información y pide confirmación antes de continuar.				
	Hay un acceso a la página de inicio en una zona visible y reconocible.				
	Si la respuesta se retrasa, aparece un mensaje como que el sistema está procesando la acción.				
	El Sitio cuenta con un mapa o buscador que facilite el acceso directo a los contenidos.				
	Los enlaces se abren en una nueva ventana.				
	Es posible aumentar y disminuir el				

	tamaño de letra.				
	En caso que un proceso requiera varios pasos es posible volver a los pasos anteriores para modificar los datos.				
	Existe alguna página mayor de una pantalla y media.				
	Se puede guardar una página del web.				
<b>Consistencia.</b>					
<b>Peso</b>	<b>Indicadores a Evaluar</b>	<b>Eval</b>	<b>(NP)</b>	<b>Cantidad de elementos afectados</b>	<b>Comentarios</b>
	Para tareas similares, los diálogos, formularios son similares.				
	No hay enlaces rotos o que no lleven a ningún sitio.				
	Se mantiene una navegación consistente y coherente en todas las pantallas.				
	Se utilizan colores estándar para los enlaces.				
	En los formularios el texto está alineado a la derecha y los campos a la izquierda.				
	Se usa la misma fuente para todos los navegadores.				
	Trabaja el sitio correctamente en exploradores modernos y antiguos.				
<b>Prevención de errores.</b>					
<b>Peso</b>	<b>Indicadores a Evaluar</b>	<b>Eval</b>	<b>(NP)</b>	<b>Cantidad de elementos afectados</b>	<b>Comentarios</b>
	Se dan indicaciones para				

	completar campos problemáticos.				
	El buscador (si existe) permite errores tipográficos y ortográficos (tildes).				
	En situaciones donde se pueden producir errores de escritura existe la posibilidad de seleccionar la información de una lista.				
	Se ofrecerán valores por defecto en los campos en caso de que tengan que estar completados y el usuario desconozca como completarlo.				
<b>Estética.</b>					
<b>Peso</b>	<b>Indicadores a Evaluar</b>	<b>Eval</b>	<b>(NP)</b>	<b>Cantidad de elementos afectados</b>	<b>Comentarios</b>
	Los títulos son descriptivos y distintivos.				
	Los tipos y tamaños de letra son legibles y distinguibles.				
	La información está jerarquizada.				
	El menú de navegación aparece en un lugar destacado.				
	Se usan anclas visuales (negrita, cursiva).				
	La información importante aparece destacada.				
	El uso de los colores es moderado.				
	No hay contenido en movimiento.				
<b>Eficiencia.</b>					
<b>Peso</b>	<b>Indicadores a Evaluar</b>	<b>Eval</b>	<b>(NP)</b>	<b>Cantidad de elementos afectados</b>	<b>Comentarios</b>

	¿El sistema permite descargar rápidamente los archivos existentes?				
	¿Se realizan de forma rápida las búsquedas en el sistema?				
	¿Se encuentra la información solicitada por el usuario?				
	¿El sistema permite el desplazamiento rápido entre las páginas?				
<b>Otros.</b>					
<b>Peso</b>	<b>Indicadores a Evaluar</b>	<b>Eval</b>	<b>(NP)</b>	<b>Cantidad de elementos afectados</b>	<b>Comentarios</b>
	¿Es simple el vocabulario utilizado?				
	¿Se proporciona tiempo suficiente para realizar las entradas por teclado?				
	¿Hay algún tipo de asistencia para los usuarios que hacen uso del sistema por primera vez?				
	¿Se entienden la interfaz y su contenido?				
	¿Está diseñada la interfaz para facilitar la realización eficiente de las tareas de la mejor forma posible?				
	¿Es fácil de utilizar el sistema en la realización de tareas?				
	¿Facilita la interfaz de usuario un uso eficiente (para interfaces basadas en pantallas)?				

¿Es apropiada la retroalimentación presentada por el sistema?				
¿Permite al usuario interrumpir su tarea y continuar más tarde?				
¿Se utiliza mensajes y textos descriptivos?				
¿Facilita la interfaz de usuario un uso eficiente (para interfaces basadas en pantallas)?				
¿Es apropiada la retroalimentación presentada por el sistema?				
¿Se permite la utilización del ratón o el teclado?				
¿Permite deshacer las acciones, e informar el estado?				
¿Permite una cómoda navegación dentro del producto y una fácil salida de este?				
¿Se proporciona información visual de dónde está el usuario, qué está haciendo y qué puede hacer a continuación?				
¿Proporciona funciones deshacer, rehacer?				
¿Se presenta al usuario la información que sólo necesita?				
¿Permite deshacer las acciones, e informar el estado?				
¿Existen diferentes niveles de ayuda?				

## Anexo 5.

### Atributos de calidad: Portabilidad.

<b>Manual de Instalación.</b>					
<b>Peso</b>	<b>Indicadores a Evaluar</b>	<b>Eval</b>	<b>(NP)</b>	<b>Cantidad de elementos afectados</b>	<b>Comentarios</b>
	¿El resumen describe el objetivo que se persigue con el manual de instalación?				
	¿La introducción describe todas las herramientas necesarias y las condiciones necesarias para iniciar la instalación?				
	¿Con los pasos descritos en la instalación es posible instalar el sistema?				
	¿Se especifica con imágenes cada pasos de la instalación para un mejor entendimiento?				
	¿Existe un instalador para la aplicación en caso que sea muy complicada (si es necesario crear variables en los registros del sistema, copiar dll en la carpeta window, etc.)?				
<b>Otros</b>					
<b>Peso</b>	<b>Indicadores a Evaluar</b>	<b>Eval</b>	<b>(NP)</b>	<b>Cantidad de elementos afectados</b>	<b>Comentarios</b>
	¿El producto puede adaptarse eficaz y eficientemente a diferente hardware especificado, software o funcionamiento a otro entorno?				

¿Existe facilidad para producto pueda ser instalado satisfactoriamente y/o desinstalado en un entorno específico?				
¿Existe independencia del ambiente de hardware? (características particulares del hardware).				
¿Existe independencia del ambiente de software? (sistema operativo, lenguaje de programación).				
¿El software puede ser modificado para los ambientes especificados sin aplicar acciones o medios?				
¿El software comparte los recursos comunes para coexistir con otro software independiente en un ambiente común?				

## **Anexo 8.**

### **Encuesta para validar el procedimiento para la validación y verificación de los requerimientos no funcionales en aplicaciones web de gestión en el CEIGE.**

Usted ha sido seleccionado para validar el trabajo de diploma “Procedimiento para la validación y verificación de los Requerimientos No Funcionales en aplicaciones web de gestión”, por sus conocimientos y experiencia en el tema de investigación. Usted debe evaluar las siguientes preguntas en las categorías ordinales de Muy Adecuado (MA), Bastante Adecuado (BA), Adecuado (A), Poco Adecuado (PA), y No Adecuado (NA). Para ello marque con una x según considere.

No.	Preguntas	MA	BA	A	PA	NA
1	¿Considera usted que la propuesta tiene un valor científico apropiado?					
2	¿Considera usted que tiene lógica la secuencia de las etapas definidas?					
3	¿Considera usted que tiene lógica la sucesión de las actividades definidas?					
4	¿Considera usted que la solución propuesta contribuye a mejorar la calidad producto del CEIGE?					
5	¿Considera usted que la solución propuesta contribuye a mejorar el control de las actividades durante la validación y verificación de los RNF?					
6	¿Considera usted que la solución propuesta cuenta con las actividades necesarias para realizar correctamente el proceso de validación y verificación de los RNF?					
7	¿Considera usted que los roles definidos son suficientes para llevar a cabo de manera efectiva la solución propuesta?					
8	¿Considera usted que las responsabilidades definidas para los roles son suficientes para llevar a cabo su trabajo de manera efectiva en el procedimiento?					
9	¿Considera usted que los artefactos propuestos son suficientes para realizar correctamente el proceso de pruebas de en proyectos desarrollados en el CEIGE?					
10	¿Considera usted que la solución propuesta contribuye a evaluar la fiabilidad, usabilidad, portabilidad y eficiencia de los proyectos desarrollados en el CEIGE?					