

MANUAL DE PROCESOS NO FUNCIONALES PARA UNA EMPRESA FABRICANTE DE SOFTWARE

MANUAL NOT FUNCTIONAL PROCESS FOR A SOFTWARE ENTERPRISE

Sandra Carolina Gualteros Pachón
Ingeniera de Sistemas
Universidad Militar Nueva Granada, Bogotá, Colombia,
u6700518@unimilitar.edu.co

RESUMEN

Un manual de procesos no funcionales es un documento que pone en contexto los pasos que se deben seguir al momento de ejecutar pruebas no funcionales en el software antes de salir a producción. Estas pruebas son de gran importancia y por lo tanto deben ser plasmados en manuales prácticos que sirvan como mecanismo de consulta permanente, por parte de todos los analistas de pruebas, permitiéndoles un mayor control y ejecución de estas pruebas. Una de las características que se tienen en los procesos de las pruebas no funcionales, es la de encontrar defectos en la realización de una aplicación para luego poder corregirlos en las etapas del ciclo de vida antes de salir al mercado, logrando un software de buena calidad y reducir costos en estos. Las pruebas no funcionales facilitan la detección temprana de no conformidades y ayuda a la verificación del cumplimiento del atributo de calidad del software. La finalidad de este artículo es describir la metodología que se utilizará en la identificación y diagramación de todos los procesos que se realizan al momento de ejecutar pruebas no funcionales, explicando de una manera utilizada en las pruebas no funcionales, la realización de este artículo es brindar una orientación a las organizaciones desarrolladoras de software. El presente trabajo muestra la elaboración de un manual de procesos para realizar pruebas a requisitos no funcionales de software e implementar una herramienta para automatizar la gestión de estas pruebas y que sirva de apoyo para la aplicación de este procedimiento. Como resultado de este trabajo se obtuvo un procedimiento que define detalladamente cómo realizar las pruebas no funcionales de portabilidad, mantenimiento, fiabilidad, usabilidad y eficiencia a un software.

Palabras clave: Metodología, Proceso, funcional, pruebas, software.

ABSTRACT

A nonfunctional manual processes is a document that puts into context the steps to follow when executing nonfunctional testing in software before going into production. These tests are very important and therefore should be reflected in practical manuals that serve as permanent consultation mechanism, by all evidence analysts, allowing greater control and execution of these tests. One of the features that are in the process of non- functional testing is to find defects in the performance of an application to correct them later stages of the life cycle before hitting the market, achieving a good quality software and reduce these costs. Non-functional tests facilitate early detection of non -compliance and helps verify compliance of software quality attribute. The purpose of this article is to describe the methodology to be used in the identification and layout of all processes performed when executing nonfunctional testing,

explaining in a manner used in non- functional testing, the implementation of this article is to provide guidance to software development organizations. This article traces the general objective to develop a procedure for testing non-functional requirements and implement a software tool to automate management of these tests and that will support the implementation of this procedure. As a result of this work a procedure that defines in detail how to perform non-functional testing of portability, maintainability, reliability, usability and efficiency was obtained software.

Keywords: methodology, process, functional, testing software.

1. INTRODUCCIÓN

Los modelos de procesos para la empresa fabricante de software es una recopilación de información que ayuda a entender de manera sencilla las definiciones que se encuentran en el sistema con el fin de ayudar al personal de pruebas.

En los últimos años, el incremento constante del software desarrollado en Colombia, y la necesidad de mejorar su calidad, ha conllevado la necesidad de utilizar una serie de estándares y procesos para mejorar la calidad de este. Sin embargo, muchas empresas no han tenido en cuenta estos estándares o procesos de mejora hacia la calidad, haciendo que sus productos sean deficientes y muy costosos; es por ello que existe la necesidad de establecer unos procesos de pruebas no funcionales en el software haciendo de ello un producto confiable, usable y de alta calidad [1].

Una de las características que se tienen en los procesos de las pruebas no funcionales, es la de encontrar errores en la realización de una aplicación para luego poder corregirlos en las etapas del ciclo de vida antes de salir al mercado, logrando un acercamiento a un software de buena calidad y reducir costos en estos.

Las empresa dedicadas a las tecnologías de información (TI) en algún punto de su ciclo de vida, necesitan considerar los requerimientos no funcionales y las pruebas asociadas a los mismos para evitar inconvenientes cuando el software se encuentre en producción evitando re trabajo y esfuerzos innecesarios[2].

La empresa de fabricación y comercialización de software, objeto de estudio, debe ofrecer un producto de buena calidad, para esto necesita documentar los procedimientos de las pruebas no funcionales que le permita optimizar el funcionamiento del software durante las pruebas y entrega al cliente, con esto, garantizando la buena calidad del producto y prever futuras fallas.

El interés de la realización de este artículo es brindar una orientación, a las organizaciones desarrolladoras de software, en la realización de pruebas no funcionales, debido a que los diferentes problemas que surgen por no realizar estas antes de salir a producción, pueden ser muy costoso. En el mundo del software, la importancia que se le debe dar las pruebas no funcionales en el desarrollo es vital, porque con estas se genera confiabilidad, seguridad y lo mejor de todo: hasta se pueden salvar vidas [3].

En este artículo se describe la estrategia recomendada para realizar pruebas no funcionales de acuerdo al procedimiento que se establece en este documento. Este manual de procesos no funcionales está dirigido a las personas responsables de realizar pruebas no funcionales e ingenieros de pruebas, con el fin de proporcionar una representación visual de como se ve el

proceso de pruebas no funcionales de principio a fin, incluyendo los pasos, la secuencia de pasos y otras interacciones. El mapa de procesos no funcionales que presenta este artículo ofrece una base para identificar acciones de mejora y realizar un proceso más eficiente y eficaz al momento de realizar pruebas no funcionales [4].

1.1 CONTEXTO DE LA EMPRESA

La empresa objeto de estudio, fabricante de software en Colombia es un sector nuevo, donde la dedicación de desarrollo y distribución de software buscan una mejora continua, pero en casos se comete equivocaciones al querer ofrecer un buen producto, donde se enfatizan más en los tiempos establecidos de entrega de software y los precios del producto, olvidando así el desarrollo de software de calidad, omitiendo de esta manera la aplicación de pruebas no funcionales a lo largo del proceso. Estas pruebas permiten establecer la medición y evaluación del software que si antes de ser aplicadas se puede evitar inconvenientes al futuro reduciendo costos cuando el producto se encuentra producción, ya que si una aplicación no es funcional simplemente será inútil.

En la actualidad el enfoque de algunas fábricas desarrolladoras de software, es hacia el mejoramiento continuo en el proceso, para llevar esto a cabo se aplican las pruebas funcionales y no funcionales en el software, aunque en las mayorías de empresas no es muy conocido este tema. El proceso de ejecución de Pruebas de sistema debe ser considerado durante todo el ciclo de vida de un Proyecto.

La propuesta será aplicada a una compañía de desarrollo de software. Esta empresa objeto de estudio es una compañía 100% colombiana dedicada hace más de 22 años a ofrecer soluciones empresariales de la más alta calidad en el sector de las tecnologías de la información. Se dedica en a prestar servicios de administración de base de datos, desarrollo de software. Sus servicios están destinados a empresas cuyos procesos de TI son críticos para su buen funcionamiento, permitiéndoles no solo enfocarse en lo principal de su negocio sino optimizar recursos, disminuir costos, aumentar sus ventajas competitivas y generar un mayor valor por medio de la automatización de sus procesos.

En febrero de 2013 fue evaluado bajo las prácticas del modelo CMMI obteniendo la valoración en nivel 5 de madurez, lo cual ratifico la madurez de los procesos de la Organización y su orientación hacia la innovación y resolución de problemas a través de la gestión cuantitativa. En el año 2003 certificó su Sistema de Gestión de Calidad bajo la norma NTC-ISO 9001 versión 2000. Este Certificado garantiza el compromiso con la calidad y asegura la implementación de procesos eficaces con altos niveles de productividad y rentabilidad, que van a ser transmitidos a los clientes a través de la prestación de nuestros servicios [6].

1.2. DEFINICIONES

Pruebas no funcionales: Éstas se realizan para verificar que el software desarrollado cumple con los requerimientos no funcionales establecidos por el cliente. Existen varios tipos de pruebas no funcionales, entre las más comunes están las pruebas de seguridad, pruebas de rendimiento, pruebas de usabilidad, pruebas de portabilidad, entre otras.[7]

CMMI (Capability Maturity Model Integration): Modelo de madurez de mejora de los procesos para el desarrollo de productos y de servicios. Consiste en las mejores prácticas que tratan las actividades de desarrollo y de mantenimiento que cubren el ciclo de vida del producto, desde la concepción a la entrega y el mantenimiento [8].

Procesos: El proceso ayuda a los miembros de una organización a alcanzar los objetivos estratégicos ayudando a trabajar más inteligentemente, no más duro, y de un modo más consistente. Los procesos eficaces también proporcionan un medio para introducir y utilizar nuevas tecnologías de forma que permitan responder mejor a los objetivos estratégicos de la organización. Un proceso de software proporciona la estructura desde la que se puede establecer un detallado plan para el desarrollo del software [9]

Calidad del software: Es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario

Requerimientos No Funcionales: Los requerimientos no funcionales son los que especifican criterios para evaluar la operación de un servicio de tecnología de información, en contraste con los requerimientos funcionales que especifican los comportamientos específicos. Por lo general, el Plan para implementar los requerimientos no funcionales se detalla en la Arquitectura del Sistema, mientras que el de los requerimientos funcionales se especifica en el Diseño [10].

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; los de fiabilidad que fijan la tasa de fallas para que el sistema sea aceptable; los de portabilidad y los de usabilidad [11].

Requerimientos organizacionales. Se derivan de las políticas y procedimientos existentes en la organización del cliente y en la del desarrollador: estándares en los procesos que deben utilizarse; requerimientos de implementación como los lenguajes de programación o el método de diseño a utilizar, y los requerimientos de entrega que especifican cuándo se entregará el producto y su documentación.

Requerimientos externos. En principio, los requerimientos funcionales y no funcionales se diferencian en el documento de requerimientos. En la práctica, esto es difícil. Si un requerimiento no funcional se declara de forma separada a los funcionales, algunas veces es difícil ver la relación entre ellos.

Sitio WEB: es un documento o información electrónica capaz de contener texto, sonido, vídeo, programas, enlaces, imágenes, y muchas otras cosas, adaptada para la llamada World Wide Web, y que puede ser accedida mediante un navegador. Esta información se encuentra generalmente en formato HTML o XHTML, y puede proporcionar navegación (acceso) a otras páginas web mediante enlaces de hipertexto [14].

WebServices (WSV): es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet.

Procesos de Bases de datos (BDLINK): es un tipo de objeto que permite realizar una conexión desde una base de datos a otra. Su principal objetivo es ocultar el detalle de los parámetros de conexión necesarios, facilitándonos un sencillo acceso a los recursos disponibles en otras bases de datos, independientemente de que estas se encuentren instaladas en el mismo servidor o no [15].

Framework: es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software [16].

2. MATERIALES Y MÉTODOS

2.1 MATERIALES

A continuación se lista la normatividad asociada a este trabajo:

- **CMMI** (Capability Maturity Model Integration). Modelo de madurez de mejora de los procesos para el desarrollo de productos y de servicios. Consiste en las mejores prácticas que tratan las actividades de desarrollo y de mantenimiento que cubren el ciclo de vida del producto, desde la concepción a la entrega y el mantenimiento [17].
- **ISO 9001:2008.** es una norma internacional que se aplica a los sistemas de gestión de calidad (SGC) y que se centra en todos los elementos de administración de calidad con los que una empresa debe contar para tener un sistema efectivo que le permita administrar y mejorar la calidad de sus productos o servicios.
- **IEEE 829:1998.** Documentación de pruebas de software. El propósito de esta norma es describir una serie de documentos de prueba de software. Un documento de prueba estándar puede facilitar la comunicación, proporcionando un marco común de referencial.
- **ISO/IEC 25051:2006.** Requisitos de calidad de productos de software y las instrucciones para las pruebas: Esta norma determina como probar los productos de software con los requerimientos de calidad establecidos.
- **ISO/IEC 29119:2012.** Estándar de pruebas software: sirve de base a las pruebas en el mundo de desarrollo del software eliminando inconsistencia entre algunas normas actuales, las cuales no cuentan con pruebas establecidas

2.2 MÉTODO

El método adoptado fue descriptivo documental cuya fuente de formación fueron las personas vinculadas a los procesos, las leyes, decretos, resoluciones y personal vinculado a los procesos.

Investigación Descriptiva: trata de obtener información acerca del fenómeno o proceso, para describir sus implicaciones”. Este tipo de investigación, no se ocupa de la verificación de la hipótesis, sino de la descripción de hechos a partir de un criterio o modelo teórico definido

previamente. En la investigación se realiza un estudio descriptivo que permite poner de manifiesto los conocimientos teóricos y metodológicos del autor para darle solución al problema a través de información obtenida.

Investigación documental: Es la que se realiza, como su nombre lo indica, apoyándose en fuentes de carácter documental, esto es, en documentos de cualquier especie tales como, las obtenidas a través de fuentes bibliográficas, hemerográficas o archivísticas; la primera se basa en la consulta de libros, la segunda en artículos o ensayos de revistas y periódicos, y la tercera en documentos que se encuentran en archivos como cartas oficios, circulares, expedientes [18].

2.2.1 Estado del arte

El Mapa de Procesos de pruebas no funcionales de la empresa objeto de estudio, fue diseñado de acuerdo a los lineamientos de CMMI, en donde se definen unos parámetros precisos para la ejecución de pruebas no funcionales.

2.2.2 Caracterización de los procedimientos no funcionales

Mediante la caracterización de procesos se identifican solo aquellos deben aparecer en la estructura de procesos del sistema. Al caracterizar el proceso, se define el objetivo, responsable, proveedores, clientes y el personal que participa de la realización de las actividades para que todos adquieran una visión integral, fortaleciendo el trabajo en equipo y la comunicación.

2.2.3 Mapa de procesos no funcionales

La empresa fabricante de software necesita un mapa de procesos de pruebas no funcionales con el fin de dar enfoque más adecuado para la correcta interpretación y ejecución de estas pruebas, llegando a todos los colaboradores de la compañía con mayor claridad y un mejor entendimiento del mapa de procesos.

2.2.4 Validación mapa de procesos no funcionales

Se realiza validación del mapa de procesos no funcionales propuesto, para certificar que cumple con las expectativas de la empresa de acuerdo a lo planificado en este artículo.

3. RESULTADOS Y ANÁLISIS

Como consecuencia del diseño del mapa de procesos no funcionales para la empresa fabricante de software se obtuvieron los resultados que se mencionan a continuación.

3.1 ESTADO DEL ARTE

En los últimos años, el incremento constante del software desarrollado en Colombia, y la necesidad de mejorar su calidad, ha conllevado la necesidad de utilizar una serie de estándares y procesos para mejorar la calidad de este. Sin embargo, muchas empresas no han tenido en cuenta estos estándares o procesos de mejora hacia la calidad, haciendo que sus productos sean deficientes y muy costosos; es por ello que existe la necesidad de establecer unos

procesos de pruebas no funcionales en el software haciendo de ello un producto confiable, usable y de excelente calidad.

Una de las características que se tienen en los procesos de las pruebas no funcionales, es la de encontrar errores en la realización de una aplicación para luego poder corregirlos en las etapas del ciclo de vida antes de salir al mercado, logrando un acercamiento a un software de buena calidad y reducir costos en estos.

Para la realización de este material monográfico se analizaron varios estándares de niveles pruebas no funcionales, aplicados a las metodologías ágiles y robustas durante las etapas de desarrollo. También se observaron las diferentes herramientas a utilizar durante cada requerimiento no funcional que se tiene [20].

La industria del software en Colombia es un sector joven, si se encuadra en las organizaciones que se dedican a la investigación, desarrollo y comercialización de software; por ser tan joven es inmaduro pero en constante evolución y en búsqueda de mejorar continuamente.

Dichas formas de mejoramiento, son entre tanto no solo los modelos de mejora de proceso disponibles, sino además que se encuentren al alcance financiero de la empresa, o se halla también a través de adopciones de buenas prácticas de desarrollo, sin la supervisión de alguna entidad autorizada para certificarles los avances de mejora realizados, o a través de los apoyos que proporciona el gobierno por medio del Ministerio de Tecnologías de la Información y las comunicaciones [21].

Debido a la gran desarticulación entre las federaciones y el estado, no se tienen cifras exactas acerca de la cantidad de casas desarrolladoras en Colombia, sin embargo, según el DANE¹ en el país existen más de 650 empresas de desarrollo de software, de las cuales en la región cafetera se cuenta con 48 empresas legalmente establecidas en las cámaras de comercio respectivas, 20 en la ciudad de Manizales, 16 en Pereira y 12 en Armenia, sin tener en cuenta las que ingresaron al gremio el año inmediatamente anterior.

Es el constante surgimiento de nuevas empresas, el que exige a las organizaciones que ya se encuentran en dicho segmento, que se preocupen por mejorar la calidad de su producto. Adicionalmente, deben enfocar sus esfuerzos en garantizar cumplimiento en la entrega del producto, respetando los tiempos establecidos y precios, y es el factor del tiempo de entrega al mercado, por el cual las organizaciones priorizan el desarrollo del software sobre la calidad, ya que además no tienen en cuenta la aplicación de pruebas a lo largo del proceso de desarrollo del producto software, por ende, olvidan también las pruebas no funcionales [22].

Las fábricas de software del país, no sólo se enfrentan contra su propios parámetros establecidos para sus proyectos de desarrollo, como lo es tiempo, el precio final, entre otros, sino que además, deben enfrentar al mercado respectivo, el cual exige ser competitivo debido a la globalización.

Es por lo anterior que las actuales empresas del gremio, deben contar con un alto grado de especialización en sus funciones, lo cual han logrado en el país 130 empresas mediante la certificación ISO 9000, una empresa con certificación CMMI categoría V y otras cinco están en proceso de obtener esta certificación datos correspondientes al 2006.

En la actualidad la realización de pruebas no funcionales, como lo son usabilidad, estabilidad, escalabilidad, eficiencia, seguridad, permiten la evaluación y medición de la calidad del

software. Si se tiene en cuenta que por muy alto que sea el nivel funcional del software desarrollado, si la aplicación no es funcional, simplemente será inútil, sin embargo siendo tan importantes, no son aplicadas ya que se prioriza el desarrollo del producto, buscando cumplir con las entregas y minimizar costos por retrasos, pero esto luego se revierte con el tiempo, ya que el error hallado en producción es más costoso que el hallado en un ambiente controlado [23].

3.2 CARACTERIZACIÓN DE LOS PROCEDIMIENTOS NO FUNCIONALES

Se identificaron 8 etapas de los procesos no funcionales para la aplicación de pruebas, a continuación se describen cada una de ellas.

3.2.1 Metodología de desarrollo

La metodología de desarrollo consta de 3 fases funcionales para el desarrollo del software las cuales son importantes para llevar a cabo la ejecución de pruebas no funcionales, a continuación se describirán cada uno de estos procesos. En la Figura 1 se muestra las etapas de la metodología que consta el desarrollo de software el cual se refiere a un framework que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información.

El marco de trabajo para metodología de desarrollo de software consiste en:

- Una filosofía de desarrollo de programas de computación con el enfoque del proceso de desarrollo de software
- Herramientas, modelos y métodos para asistir al proceso de desarrollo de software

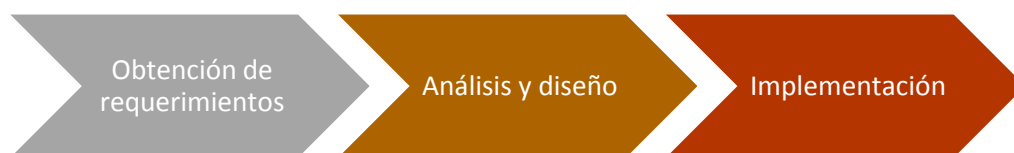


Figura 1. Etapas de la metodología de desarrollo.

3.2.2 Obtención de requerimiento no funcional:

Obtención de requerimientos: búsqueda y obtención de los requerimientos desde los grupos de interés considerando las técnicas existentes para la recolección de la información, en este punto el arquitecto realiza la búsqueda del requerimiento de calidad que desea probar en el sistema. (Confiabilidad, usabilidad, portabilidad, etc.) Se debe diligenciar el formato de la Tabla 1.

Para definir el tipo de requerimientos no funcionales, el equipo de desarrolladores y el arquitecto se basan en el estándar ISO/IEC 9126 el cual define un modelo independiente de la tecnología para caracterizar la calidad de software y toma en consideración las siguientes características:

- **Funcionalidad:** describe la presencia de funciones (funcionalidades) para alcanzar propiedades definidas. Ejemplos: interoperabilidad, seguridad
- **Confiabilidad:** describe la disponibilidad que tiene el producto para mantener sus niveles de rendimiento bajo condiciones específicas y en un tiempo dado. Ejemplos: madurez, tolerante a fallas, recuperabilidad.
- **Usabilidad:** describe el esfuerzo requerido por el usuario en la utilización de la aplicación. Ejemplo: velocidad de aprendizaje, eficacia, operabilidad.
- **Eficiencia:** describe la tasa entre el nivel de rendimiento de la aplicación y los recursos que ésta utiliza bajo condiciones específicas.
- **Portabilidad:** describe la conveniencia de que una aplicación pueda ser llevada de un ambiente a otro. Ejemplo: adaptabilidad, capacidad de instalación, capacidad de replicar.
- Otras según las condiciones y restricciones del dominio

Proyecto		Responsable		Fecha:							
ID del Requerimiento:	<i>Colocar el ID del requerimiento no funcional.</i>										
Nombre del requerimiento:	<i>Colocar el nombre del requerimiento no funcional.</i>										
Identificación del requerimiento:	<i>Identificación del requerimiento no funcional (debe ser consistente con su identificación y utilización en el glosario).</i>										
Características:	Canal	Atributo de calidad									
		Atributos de calidad ISO/IEC 9126-1							Otros		
	usabilidad	Performance	Seguridad	Confiabilidad	Mantenibilidad	portabilidad	Adaptabilidad	Reutilización	Fiabilidad	Disponibilidad	Escalabilidad
	WEB										
	WSV										
DBL											
Descripción del requerimiento:	<i>Describir el requerimiento no funcional con información suficiente para ser utilizada más adelante en el proceso de especificación y diseño. Pueden utilizar representaciones gráficas si lo desean, y estas deben anexarse al documento.</i>										
Requerimiento funcional impactado:	<i>Especificar el (los) requerimiento(s) funcional(es) que es (son) impactado(s) por este requerimiento no funcional. Explique en detalle.</i>										
Prioridad del requerimiento:											
<input type="radio"/> Alta	<input type="radio"/> Media Alta	<input type="radio"/> Media	<input type="radio"/> Media Baja	<input type="radio"/> Baja							

Tabla1. Formato de definición de requerimientos no funcionales

3.2.3 Análisis y diseño

- **Análisis**

Para el análisis de los requerimientos no funcionales se debe tener en cuenta lo siguiente:

- Cada requerimiento debe ser especificado de forma clara, consistente, verificable, cuantificado en cifras verídicas, medibles y reales. De tal forma que no queden abiertos a un amplio rango de interpretaciones subjetivas.
- **Específico:** sin ambigüedad, usando terminología consistente para evitar diferentes interpretaciones.
- **Medible:** Que permitan su seguimiento y evaluación.
- **Accesible:** que sea factible
- **Realizable:** es decir, alcanzables, con sentido, desafiantes
- **Rastreable (ubicable):** se lo puede vincular desde su concepción (a través de su especificación), a su diseño siguiente, implementación y test.

Clasificarlos por categorías, según correspondan a los siguientes atributos de calidad software, hasta el día de hoy solo se trabaja con los siguientes requerimientos no funcionales: Fiabilidad, Portabilidad, Eficiencia y Usabilidad.

Priorizar los requerimientos no funcionales según la prioridad complejidad del requerimiento, Las prioridades de los planes y los casos de pruebas se deben clasificar como de importancia alta, media o baja. Cuando las pruebas deben llevarse a cabo en poco tiempo, esto ayudará a decidir qué casos deben probarse y cuáles pueden omitirse sin problemas.

El equipo de pruebas tendrá que decidir el grado de pruebas necesario en cada una de estas áreas. Cada área puede clasificarse con el siguiente baremo de importancia:

- Alta: un área muy importante, que debe probarse detenidamente
- Media: requiere una prueba estándar
- Baja: se debe probar si el tiempo lo permite

- **Diseño**

Partiendo de la documentación de los requerimientos no funcionales, se procede a la fase a la modelación realizado en casos de uso extendido para los requerimientos no funcionales. Ver Tabla 2, en esta área se definen todos los elementos que se utilizaran y estrategias que se realizaran al momento de realizar pruebas no funcionales.

1. Información General	Nombre del Plan de Pruebas	<i>Asigne un nombre al plan de pruebas detallado, que tenga que ver con la característica, módulo, funcionalidad, componente, caso de uso, o requerimiento que se va a probar</i>
	Responsable(s)	<i>Nombre los responsables de la ejecución del plan de pruebas detallado</i>
	Fecha Elaboración	<i>Fecha en que se elabora el plan de pruebas detallado</i>
	Documentación Fuente	<i>Nombre de los documentos de referencia al módulo a probar</i>
2. Características a ser probadas	Característica	<i>Nombre de la característica, módulo, funcionalidad, componente, caso de uso, o requerimiento que se encuentra dentro del alcance de la prueba</i>
	Prioridad	<i>Prioridad asociada a la característica a probar (alta, media, baja)</i>
	Pruebas	<i>Ingresar Pruebas no funcionales</i>
	Características Excluidas	<i>Nombre las características, módulos, funcionalidades, componentes, casos de uso, o requerimientos que se excluyen de las pruebas. Indique también los aspectos particulares que no serán tenidos en cuenta en las pruebas (por ejemplo: aspectos de navegabilidad, funciones específicas de una forma, etc.)</i>
Datos Requeridos para las Pruebas	<i>Determine explícitamente las características que deben tener los datos que se requieren para las pruebas, definiendo la siguiente información: * ¿Cuáles datos se requieren para ejecutar las pruebas? * ¿De dónde se van a obtener los datos? * ¿Cuál es el volumen de datos en la base de datos de producción? * ¿Cuál es el volumen de datos a instanciar en la base de datos de pruebas?</i>	
3. Configuración de ambiente	Servidor de Aplicaciones	<i>Indique la ruta, instancia o características del servidor de aplicaciones contra el cual será ejecutada la prueba</i>
	Base de Datos	<i>Indique el nombre o instancia de la base de datos sobre la cual se ejecutarán las pruebas</i>
	Interfaces	<i>Mencione las interfaces con otros sistemas que deben estar configuradas para efectuar las pruebas</i>
	Otros Requerimientos de Ambiente	<i>Enumere cualquier otro requerimiento de ambiente que considere necesario para la ejecución de las pruebas</i>
	Browser y Sistema Operativo	<i>Indique el navegador (incluyendo versión) y sistema operativo del entorno cliente sobre el cual se ejecutarán las pruebas</i>
4. Herramientas requeridas	Nombre Herramienta	<i>Nombre de la herramienta que se empleará para soportar la ejecución de este plan de pruebas</i>
	Descripción	<i>Indique cual tipo de prueba se va a ejecutar a través de la herramienta</i>
5. Estrategia de Pruebas	Estrategia de Pruebas	<i>Explique la estrategia de pruebas que se va a seguir para probar las características, módulos, funcionalidades, componentes, casos de uso, o requerimientos que se va a probar. En esta explicación se deben incluir aspectos como la secuencia que se va a usar para ejecutar las pruebas, el enfoque de las pruebas, la intensidad de las pruebas, si se van a realizar simulaciones de algún tipo, si se van a generar archivos que permitan verificar los cálculos, y cualquier otro aspecto que permita entender la manera como se tiene planeado ejecutar las pruebas.</i>
5. Observaciones	Observaciones	<i>Complemente la información del plan de pruebas con sus observaciones, procedimiento de ejecución de las pruebas, énfasis o cualquier otra información relevante</i>

Tabla 1 Plan de pruebas detallado - sección diseño

- **IMPLEMENTACION**

En esta fase como se cuenta con un producto o módulo, se realizan pruebas cortas que fueron diseñadas en la etapa anterior, es decir, se ejecutan teniendo en cuenta que en su diseño se categorizó dependiendo del atributo de calidad a evaluar, para verificar el nivel de cumplimiento de los requerimientos no funcionales.

3.2.4 Planeación de las pruebas no funcionales

La planeación de pruebas logra enfocar al grupo que realiza estas cual es la situación aplicativo y así prepararlos a los diferentes tipos de pruebas que se van a emplear. Los planes de pruebas están constituido por un conjunto pruebas, en donde se determina que atributos de pruebas se desean llevar acabo como la fiabilidad, usabilidad, entre otros.

El primer paso es definir y documentar los objetivos de las pruebas en un plan de pruebas detallado, se ilustran en las Tablas 2 y 3. Este plan documenta también las siguientes cuestiones:

- Nombre del plan de pruebas
- Responsable
- Fecha de elaboración
- Documentación fuente
- Objetivo de la prueba
- Tipo de prueba
- Escenarios de la prueba
- Observaciones

1. Información General	Nombre del Plan de Pruebas	<i>Asigne un nombre al plan de pruebas detallado, que tenga que ver con la característica, módulo, funcionalidad, componente, caso de uso, o requerimiento que se va a probar</i>
	Responsable(s)	<i>Nombre los responsables de la ejecución del plan de pruebas detallado</i>
	Fecha Elaboración	<i>Fecha en que se elabora el plan de pruebas detallado</i>
	Documentación Fuente	<i>Nombre de los documentos de referencia al módulo a probar</i>
2. Procedimiento de pruebas	Objetivo de la prueba	<i>Indique cual es el objetivo de la prueba</i>
	Tipo de prueba	<i>Indique el requerimiento probar</i>
	Escenarios de la prueba	<i>Nombre los posibles escenarios de las pruebas a ejecutar, interacciones, numero de concurrencia y volumen de datos por cada escenario, pasos de prueba y programación de la prueba.</i>
4. Observaciones	Observaciones	<i>Complemente la información del plan de pruebas con sus observaciones, procedimiento de ejecución de las pruebas, énfasis o cualquier otra información relevante</i>

Tabla 2 Plan de pruebas detallado - sección planificación de escenarios

En un plan de pruebas detallado, se asigna a cada situación una prioridad basada en la probabilidad con la que se producirá y su impacto en el negocio. Los planes de pruebas detallados se suelen dividir en función de la organización del grupo de pruebas.

Para probar la arquitectura para los proyectos de software, se han identificado los siguientes requisitos no funcionales:

- fiabilidad
- portabilidad
- eficiencia
- Usabilidad

A continuación se definirán la medición de cada requerimiento no funcional para la selección de escenario y cumplimiento del requerimiento seleccionado, la metodología que se usara será el modelo de McCall el cual organiza los factores en tres ejes o puntos de vista desde los cuales el usuario puede contemplar la calidad de un producto. Ver tabla 4.

Requerimiento	Características evaluables	Cumplimiento
Fiabilidad	Recuperabilidad	<ul style="list-style-type: none"> ➤ Madurez Capacidad del producto software para evitar fallar como resultado de fallos en el software. ➤ Tolerancia a fallos Capacidad del software para mantener un nivel especificado de prestaciones en caso de fallos software o de infringir sus interfaces especificados. ➤ Capacidad de recuperación Capacidad del producto software para reestablecer un nivel de prestaciones especificado y de recuperar los datos directamente afectados en caso de fallo. ➤ Cumplimiento de la fiabilidad Capacidad del producto software para adherirse a normas, convenciones o regulaciones relacionadas con la fiabilidad.
	Disponibilidad	
	Frecuencia de fallas	
	Severidad de fallas	
Portabilidad	Adaptabilidad	<ul style="list-style-type: none"> ➤ Adaptabilidad Capacidad del producto software para ser adaptado a diferentes entornos especificados, sin aplicar acciones o mecanismos distintos de aquellos proporcionados para este propósito por el propio software considerado. ➤ Instalabilidad Capacidad del producto software para ser instalado en un entorno especificado. ➤ Coexistencia Capacidad del producto software para coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes. ➤ Capacidad para reemplazar Capacidad del producto software para ser usado en lugar de otro producto software, para el mismo propósito, en el mismo entorno. ➤ Cumplimiento de la portabilidad Capacidad del producto software para adherirse a normas o convenciones relacionadas con la portabilidad.
	Instalabilidad	
	Facilidad de adaptación al cambio	
	Co - existencia	
Eficiencia	Tiempo de respuesta	<ul style="list-style-type: none"> ➤ Comportamiento temporal Capacidad del producto software para proporcionar tiempos de respuesta, tiempos de proceso y potencia apropiados, bajo condiciones determinadas. ➤ Utilización de recursos Capacidad del producto software para usar las cantidades y tipos de recursos adecuados cuando el software lleva a cabo su función bajo condiciones determinadas. ➤ Cumplimiento de la eficiencia Capacidad del producto software para adherirse a normas o convenciones relacionadas con la eficiencia.
	Desempeño	
	Tráfico generado en las comunicaciones	
	Tiempo de recuperación a fallas	
	Utilización de recursos (Memoria)	
Usabilidad	Operabilidad	<ul style="list-style-type: none"> ➤ Capacidad para ser entendido: Capacidad del producto software que permite al usuario entender si el software es adecuado y cómo puede ser usado para unas tareas o condiciones de uso particulares. ➤ Capacidad para ser aprendido: Capacidad del producto software que permite al usuario aprender sobre su aplicación. ➤ Capacidad para ser operado: Capacidad del producto software que permite al usuario operarlo y controlarlo ➤ Capacidad de atracción: Capacidad del producto software para ser atractivo al usuario. ➤ Cumplimiento de la usabilidad: Capacidad del producto software para adherirse a normas, convenciones, guías de estilo o regulaciones relacionadas con la Usabilidad.
	Aprendizaje	
	Atractividad	
	Comprensibilidad	

Tabla 3 Atributos de calidad del software que son evaluables al aplicar pruebas no funcionales

3.2.5. Diseñar los casos de pruebas detallados

Partiendo de la documentación de los requerimientos no funcionales, se procede en esta etapa a la modelación de éstos en casos de uso extendido para los requerimientos no funcionales.

Cada uno de los escenarios mostrados en los planes de pruebas detallados se traduce en uno o varios casos de pruebas detallados con el fin de crear documentos de casos de pruebas detallados. Los casos de pruebas muestran los pasos detallados, los requisitos de datos para realizar la prueba, los resultados esperados y los métodos para registrar los resultados.

3.2.5.1. Crear casos de pruebas detallados

El contenido típico que debe encontrarse en un documento de caso de prueba detallado incluye el número de referencia único en su escenario en el documento del plan de pruebas detallado, su prioridad, la descripción de la condición que se va a probar, detalles de la ejecución, los datos necesarios para realizar la prueba, el resultado esperado, el resultado real y el estado de la prueba. Los detalles de ejecución de los casos de prueba pueden simplificarse. Ver Tabla4.

3.2.5.2. Ejecución de las pruebas no funcionales

Una vez completado el diseño y con la aplicación o infraestructura lista para comenzar las pruebas, el ingeniero de sistemas empieza el proceso la fase de prueba. El equipo de pruebas lleva a cabo una prueba de aceptación de versión, o prueba de fuego, con el fin de comprobar que la versión es lo suficientemente estable como para continuar con las pruebas. Con frecuencia, estas dos pruebas se combinan en una sola.

Se ejecutan en cada versión principal y los resultados se registran y se comprueban. Se documenta cualquier desviación de los resultados esperados y se realiza un seguimiento con una herramienta de seguimiento de errores. Al finalizar la ejecución de las pruebas planeadas y diseñadas, se organizan en un documento, de tal forma que se le presenta el análisis de los resultados de las pruebas, como también los casos de pruebas usados y las métricas utilizadas para definir si pasaba o no las pruebas. Ver tabla 5.

3.2.5.3. Análisis de las pruebas no funcionales

En esta etapa se deben analizar los resultados de las evaluaciones propuestas, según la categoría y se compara con los resultados esperados, para determinar un nivel de cumplimiento

Cuando el equipo de pruebas completa los casos de prueba, se prepara un informe de las pruebas donde se indican los errores abiertos y se resume el rendimiento del sistema, las determinaciones del equipo de dirección del sistema y las recomendaciones. Los errores se muestran según la prioridad. En este informe también se resalta el impacto que supondría para el negocio si no se corrigieran los errores; además, el equipo de pruebas recomienda si se debe continuar con la producción [24].

En este informe se describen las áreas que han sido probadas y se resumen los resultados. Además, se detallan los errores o problemas propuestos para versiones futuras. El informe resalta también el impacto que supondría para el negocio si los errores restantes no se corrigieran, junto con una recomendación acerca de si se debe seguir con la producción.

El informe completo se presenta al equipo de dirección del proyecto para ayudarles a evaluar si se pueden autorizar la fase de producción del producto o si es necesario corregirlo.

3.2.6. Corrección de errores de diseño

El equipo de desarrollo corrige los errores de diseño registrados por el equipo de pruebas. Después de que el equipo de desarrollo solucione estos errores, se les vuelve a asignar al miembro del equipo de pruebas que lo abrió en un principio. El equipo de pruebas vuelve a probar la implementación y cierra cada error una vez corregido. Cuando se han corregido todos los errores, se crea una nueva versión para volver a realizar las pruebas

3.2.7. Validar criterios de aceptación

Se analizan los resultados uno por uno, determinando que va y que no va. Después se analizan todos los resultados para determinar si se puede aceptar o no, los requerimientos no funcionales en el software, éstos deben ser aceptados por el cliente final, quien evalúa los entregables por cada iteración de pruebas [25].

Todo debe quedar registrado en un reporte donde se tenga constancia de cuáles fueron los errores habidos y su posible falla, así dando a conocer al desarrollador cuales fueron las inconsistencias presentadas durante la etapa de pruebas

3.2.8. Certificación

Actividades de cierre: gestión de las pruebas.

Los ciclos de pruebas deben estar coordinados con los ciclos de desarrollo, porque esto puede causar un retraso en el producto del software y por ende va en detrimento de la percepción del cliente.

Los atributos no funcionales se conocen como características de calidad, las cuales se deben tener en cuenta tanto al diseñar como al implementar el software.

Por ende en el presente documento, se propone un modelo que sirva de orientación hacia la evaluación de éstos, que a su vez afectan la calidad del software desarrollado.

Se da por certificado el software cuando cumpla con el requerimiento del atributo de calidad propuesto en la Etapa 5.

Caso de uso	Numero de referencia único en su escenario en el documento del plan de pruebas detallado											Prioridad:												
Descripción	Se refiere al objetivo a cumplirse a través del caso de uso y la fuente de requerimiento.																							
Actores	Se define una lista de actores que participan en el caso de uso																							
Detalle	detalles de la ejecución, los datos necesarios para realizar la prueba																							
	Escenario	Precondiciones					Pasos				Salidas esperadas													
	1.																							
	2.																							
	3.																							
Resultados	Fecha de ejecución	Funcionalidad	Ambiente	Versión	Escenario	canal	hilos	Periodo de subida	muestra	media	Min	Max	%error	rendimiento										
Análisis	--Zona de grafico					<p>En esta etapa se deben analizar los resultados de las evaluaciones propuestas, según la categoría y se compara con los resultados esperados, para determinar un nivel de cumplimiento</p>																		
	<table border="1"> <caption>Data for Performance Graph</caption> <thead> <tr> <th>Escenario</th> <th>media</th> <th>mediana</th> <th>Rendimiento</th> </tr> </thead> <tbody> <tr> <td>escenario1</td> <td>4.5</td> <td>2.5</td> <td>2.0</td> </tr> <tr> <td>escenario2</td> <td>2.5</td> <td>4.5</td> <td>2.0</td> </tr> <tr> <td>escenario3</td> <td>3.5</td> <td>1.8</td> <td>3.0</td> </tr> <tr> <td>escenario4</td> <td>4.5</td> <td>2.8</td> <td>5.0</td> </tr> </tbody> </table>														Escenario	media	mediana	Rendimiento	escenario1	4.5	2.5	2.0	escenario2	2.5
Escenario	media	mediana	Rendimiento																					
escenario1	4.5	2.5	2.0																					
escenario2	2.5	4.5	2.0																					
escenario3	3.5	1.8	3.0																					
escenario4	4.5	2.8	5.0																					
Observaciones	Descripción de errores, observaciones adicionales, comentarios.																							

Tabla 4 Resultado de pruebas no funcionales

3.2.9. Mapa de procesos no funcionales

Basado en la estructura de procesos con la que cuenta la empresa, a través de este trabajo se presenta el siguiente Mapa de Procesos para que aquellos personas del área de pruebas de una forma más clara y sencilla interpretación, Ver figura 2.

El mapa de procesos para pruebas no funcionales describen las pruebas necesarias para medir las características de los sistemas y software que se puede cuantificar en una escala variable, tales como tiempos de respuesta para las pruebas de rendimiento

El mapa de procesos de las pruebas no funcionales consta de varias etapas, descritas anteriormente.

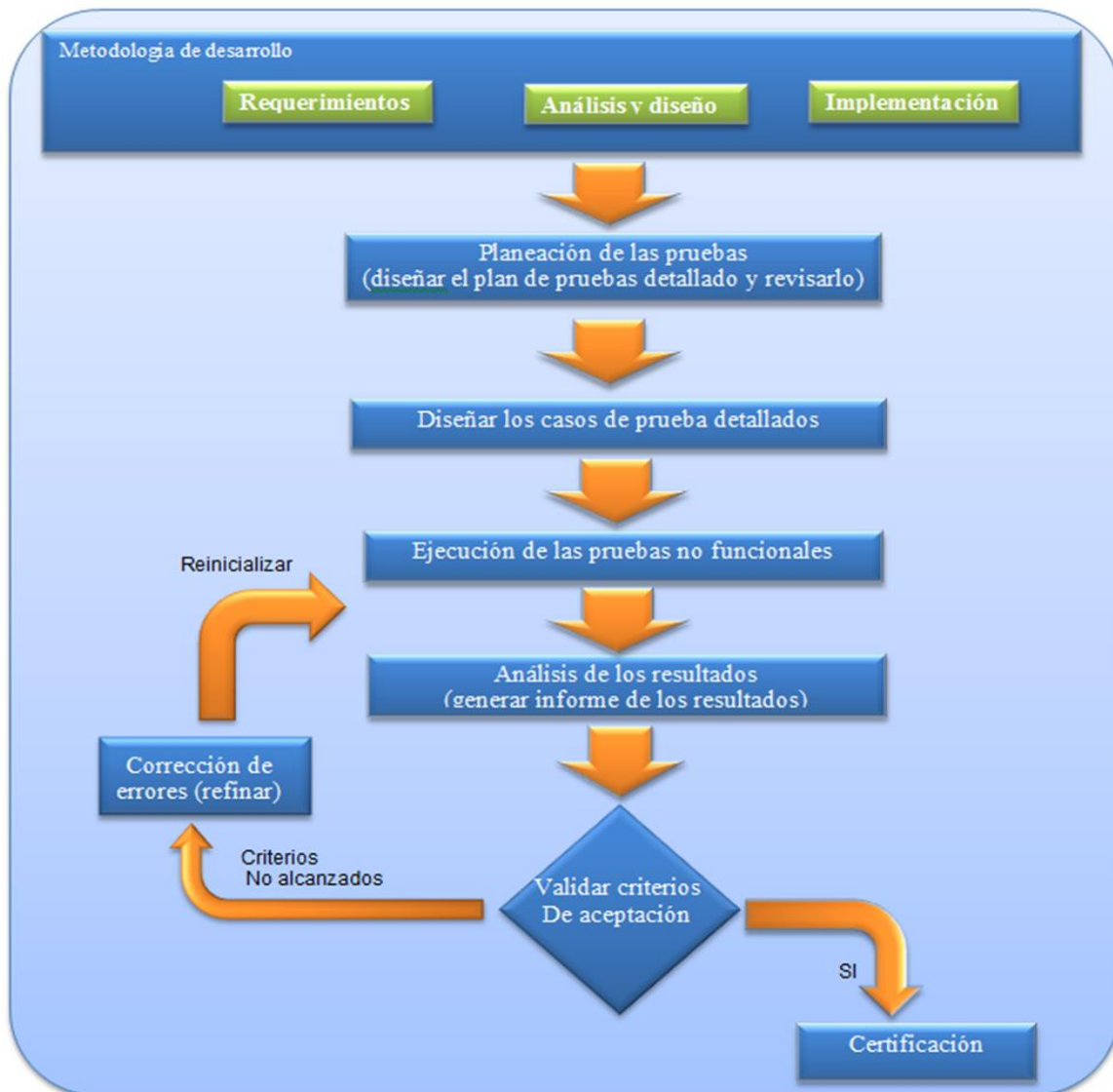


Figura 2 Mapa de procesos no funcionales

3.2.10. Validación mapa de procesos no funcionales

El anterior mapa de procesos no funcionales se validó durante un mes llevando los formatos que se establecieron en este documento, del cual se obtuvieron resultados óptimos que sirvieron para la mejora continua de las pruebas no funcionales.

Actualmente el mapa de procesos se encuentra establecido en la empresa y en funcionamiento, en la validación de los procesos se sugirió por parte del área establecer más pruebas no funcionales y agregarlas a los formatos establecidos.

4. CONCLUSIONES

- Las empresas desarrolladoras de software se deben concientizar que las pruebas no funcionales son parte fundamental en el impacto del producto final. Los errores detectados a tiempo pueden evitar el incremento de costos, los cuales afectan directamente a la organización.
- El proceso de pruebas genera costos, por tanto se debe seleccionar de manera cuidadosa tanto las pruebas a realizar, como los componentes a probar. Esto es posible determinarlo, con base en la importancia de las piezas de software en el producto final.
- El mapa de procesos Incrementan la productividad del equipo de desarrollo, ayudando a la facilitación de la detección temprana de no conformidades.
- Las pruebas no funcionales son un mecanismo objetivo para asegurar que los requerimientos de los productos se cumplen.

REFERENCIAS

- [1]. SOMMERVILLE, Ian. Ingeniería de software. Séptima Edición. Madrid, España: Pearson Educación S.A, 2005. 677 p.
- [2]. PRESSMAN, Roger S. Ingeniería del software. Un enfoque práctico. Sexta Edición. Estados Unidos: McGraw Hill, 2005. 900 p.
- [3]. REVISTA ESPAÑOLA DE ELECTRÓNICA. (Número 550: 2000: Pág. 90). Ingeniería del software y pruebas. María del Mar Aguilar Fernández.
- [4]. PLAN DE PRUEBAS. Internet (www ldc.usb.ve
<<http://ldc.usb.ve/~teruel/ci4713/clases2001/planPruebas.html>>)
- [5]. Plantilla para caso de prueba. Internet (www.scielo.unal.edu.co
http://www.scielo.unal.edu.co/scielo.php?pid=S1692-33242009000300004&script=sci_arttext)
- [6]. Desarrollo de software. Internet <www.asesoftware.com/interno>
- [7]. CALIDAD DE COMPONENTES SOFTWARE. Internet (www.eduardoleyton.com
<http://www.eduardoleyton.com/apuntes/ISO_9126_12207_UST.pdf>)
- [8]. REVISTA EIA, ISSN 1794-1237. (Número 13: Julio 2013: Pág.131). Escuela de Ingeniería de Antioquia, Medellín -Colombia.
- [9]. Metodologías ágiles para el desarrollo de software. Internet
<www.willydev.net/descargas/masyxp.pdf>
- [10]. Metodología Rational Unified Process .Internet
<<http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulosRUP%20vs.%20XP.pdf>>
- [11]. Un estudio inicial de requerimientos no funcionales a través del caso de uso. Internet <www.jidis.frc.utn.edu.ar/papers/141242db4850df4caa8b371faf07.pdf>
- [12]. Desafíos y oportunidades de la industria del software en América Latina. Internet <www.eclac.org/publicaciones/xml/5/35655/Capitulo5.pdf>
- [13]. Departamento Nacional de Planeación, agenda interna para la competitividad y productividad sector software. Internet www.dnp.gov.co<<http://www.dnp.gov.co>> .
- [14]. ESI, European Software Institute, Numero de empresas certificadas en CMMI. Internet (www.sei.cmu.edu. <<http://www.sei.cmu.edu>>).
- [15] MG informática. Internet <http://www.mginformatica.com.ar/modelo-de-calidad.htm>>

- [16] Metodologías de requerimientos no funcionales. Internet
<https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales#_Toc324099576>
- [17] Requerimientos no funcionales, internet
<<http://www.pmoinformatica.com/2013/01/requerimientos-no-funcionales-porque.html>>
- [18] HP. Internet < <http://www8.hp.com/co/es/software-solutions/software.html?compURI=1172026#.Uyo4Nfl5Mfk>>
- [19] Pruebas no funcionales. Internet. <<http://blog.fdv-solutions.com/2011/03/pruebas-no-funcionales/> > [Enero 2014]
- [20] procesos no funcionales. Internet.
<<http://repositorio.utp.edu.co/dspace/bitstream/11059/2470/1/0057565015i.pdf>>
- [21] Metodologías pruebas. Internet
<<http://www.oocities.org/es/annadugarte/seminario/Metodologia.htm>>
- [22] Clasificación de métodos. Internet
<<http://www.encyclopedia.humanet.com.co/dic/clasifimtodo.htm>>
- [23] Consultado en: FI en línea disponible en
<<http://www.fi.upm.es/?id=tablon&acciongt=consulta1&idet=226>>
- [24] Pruebas no funcionales. Internet. <<http://qualitydave.blogspot.com/p/me-preguntan-sobre-pruebas-no.html>>
- [25] Procesos para el desarrollo de software. Internet.
<http://es.wikipedia.org/wiki/Proceso_para_el_desarrollo_de_software>
- [26] Calidad de software. Internet.
<http://calidadyssoftware.blogspot.com.es/2012/04/testing-ii-planificacion-de-las-pruebas.html>[Enero 2014]