

**Universidad de las Ciencias Informáticas**

**Facultad Siete**



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

**Título:**

*Procedimientos para normalizar las fases de Depuración y Cambios en el desarrollo de Aplicaciones Web*

**Autoras:**

Irina Carralero Gómez

Marileydis Sosa Damas

**Tutoras:**

Ing. Anet Conde Almeida

Ing. Yanitza Ramírez Stambor

**Ciudad de La Habana, Mayo 2009**

**“Año del 50 Aniversario del Triunfo de la Revolución Cubana”**

## **Declaración de Autoría**

Declaramos que somos las únicas autoras de este trabajo y autorizamos a la Facultad Siete de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los 8 días del mes de Mayo del año 2009.

---

**Autora: Marileydis Sosa Damas    Autora: Irina Carralero Gómez**

---

**Tutora: Ing. Yanitza Ramírez Stambor    Tutora: Ing. Anet Conde Almeida**

## **Datos de Contacto**

### **Ing. Anet Conde Almeida**

Graduada en la Universidad de las Ciencias Informáticas en el curso 2007-2008. Actualmente es profesor instructor. Pertenece al Proyecto de Calidad de la Facultad Siete como Asesora de un equipo de calidad.

Empresa: UCI Dirección: Carretera a San Antonio Km. 2 1/2 Reparto Torrens, Facultad 7, Ciudad Habana.

e-mail: [aconde@uci.cu](mailto:aconde@uci.cu).

### **Ing. Yanitza Ramírez Stambor**

Graduada en la Universidad de las Ciencias Informáticas en el curso 2007-2008. Actualmente es profesor instructor. Pertenece al Proyecto de Calidad de la Facultad Siete como Asesora de un equipo de calidad.

Tiene dos publicaciones: “Aplicación de las TIC en el proceso de informatización del Sistema Nacional de Salud. Experiencias en la UCI”, “PROPUESTA DE ARQUITECTURA DE UN SERVIDOR DE INTEGRACIÓN PARA LA COMUNICACIÓN ENTRE SISTEMAS DE INFORMACIÓN EN UN HOSPITAL”.

Empresa: UCI Dirección: Carretera a San Antonio Km. 2 1/2 Reparto Torrens, Facultad 7, Ciudad Habana.

e-mail: [yramirez@uci.cu](mailto:yramirez@uci.cu).

*“La responsabilidad nuestra es luchar porque la calidad del producto que aquí se haga sea de las mejores y la mejor posible...”*

*Ernesto “Che” Guevara*



*Agradecimientos*

*A la Revolución por darnos la oportunidad de forjarnos como buenos profesionales.*

*A Fidel por ser el máximo líder e inspirador de esta universidad de excelencia.*

*A nuestras tutoras Yanitza y Anet por todo el apoyo y ayuda brindado durante la  
realización del trabajo.*

*A la universidad en general y a todos los profesores que nos ayudaron a fundar el camino  
para llegar hasta el final.*

*De Marisleydis:*

*A mis padres Rosa María y Nelson, por el apoyo incondicional que me han dado durante toda mi vida y por impulsarme a escoger el mejor camino en la vida.*

*A mis abuelos Adelina y Pedro Luis por ser los mejores abuelos del mundo a pesar de no estar presente en este día tan importante.*

*A mi novio Roger por guiarme siempre y apoyarme en todo lo que necesito.*

*A mis hermanos y mi madrastra por ayudarme mucho en toda mi carrera.*

*A la Universidad de las Ciencias Informáticas por haber forjado en mí una universitaria con principios y valores.*

*A mis queridas tutoras Yanitza y Anet por el apoyo tan grande y la amistad que me brindaron. Ustedes han hecho posible que hoy termine mi carrera en la Universidad de las Ciencias Informáticas, gracias por confiar en mí.*

*A todos los profesores que me han brindado todo su apoyo durante mi formación en estos cinco años.*

*A todos mis compañeros que de una forma u otra han contribuido a la realización de esta investigación y que desde primer año me han apoyado en todo.*

*A mis compañeros Yislenys, Osvaldo por ayudarme en todas mis dudas en esta investigación y en especial a José Yadiel Montalván por estar siempre apoyándome en toda mi carrera.*

*A mi especial compañera de tesis Irina por soportarme y ser la mejor en la realización de este trabajo.*

*De Irina:*

*A mis padres Milagros y Walfrido por darme su apoyo, por estar siempre al tanto de mis problemas y ayudarme sin condición alguna, por guiarme en todo momento y haber hecho de mi una persona incondicional. Gracias.*

*A toda mi familia porque todos de una forma u otra me han ayudado muchísimo en estos cinco años (a mis abuelos, tíos y primos).*

*A Yariel por estar siempre a mi lado ayudándome y guiándome en todo momento.*

*A esta escuela porque ha sido mi casa todo este tiempo y porque ha hecho de mi una persona con principios y valores.*

*A todas mis amistades y compañeros que desde primer año que han sido fundamental en el desempeño de mi vida durante todo este tiempo en la Universidad.*

*A los profesores que siempre me dieron su mano para poder cumplir el objetivo de estos cinco años.*

*Agradecerles en especial a Yislenys, Osvaldo y a Yenisel por toda su ayuda y colaboración durante el transcurso de esta investigación.*

*A ti Mary por ser tan comprensible, por ayudarme tanto, por estar siempre ahí y haber llegado a ser mi amiga.*

*A todos los que hoy están aquí ayudándome a cumplir con mi sueño.*

*A todos mil gracias.*

*Dedicatoria de Marisleydis:*

*Desde niña creía que llegar a este momento era un sueño imposible pero gracias a muchas personas he podido llegar hasta aquí, a las cuales quiero dedicar este trabajo:*

*Ante todo quiero agradecer a Dios porque me ha ayudado siempre que se lo he pedido y me ha dado las fuerzas necesarias para seguir adelante en los momentos difíciles.*

*A mis padres por ser los ojitos que me han guiado siempre, porque ustedes son mi inspiración y sin su apoyo esto hoy no hubiera sido posible. Los amo mucho.*

*A la memoria de mis abuelitos Pedro Luis y Adelina que estarán siempre en mi corazón.*

*A mis amigos con quienes he compartido momentos increíbles que siempre llevaré en mi corazón, quienes han enriquecido mi vida con su cariño y alegría. Gracias por recordarme que hay personas valiosas en el mundo y gracias por estar en el mío.*

*A toda mi familia en general que ha contribuido a que pudiera llegar a este día tan especial.*

*Dedicatoria de Irina:*

*A mis padres por ser los principales responsables de que haya cumplido este gran sueño, por confiar en mí y por estar a mi lado justo en el momento y lugar donde los necesito, porque toda una vida no me alcanzaría para recompensarlos por todo lo que han hecho en todos los momentos de mi vida.*

*A Yariel por ayudarme en todo momento y brindarme su cariño.*

*A mis abuelitos que quiero tanto y por esperar tanto para ver que este día se hiciera realidad.*

*A mi familia que tanto me han ayudado para que yo pudiera cumplir con mi sueño.*

*A ustedes, mis amigos porque han sido mi familia todo este tiempo, ayudándome y compartiendo en todo momento, a ustedes que siempre los recordaré y los llevaré en mi corazón.*

*A todas las personas que hoy están aquí compartiendo uno de los días más felices de mi vida.*

**Resumen**

Esta investigación propone tres procedimientos para normalizar las fases de Depuración y Cambios (Ejecución de las Pruebas de Integración, Sistema, Liberación y/o Aceptación) en el desarrollo de Aplicaciones Web en la Facultad Siete de la Universidad de las Ciencias Informáticas (UCI).

En la realización del presente trabajo se realizaron entrevistas y auditorías a los diferentes grupos de calidad de cada área temática de la facultad involucrada, lo que permitió obtener un diagnóstico del estado de la realización de las pruebas en cada una de las fases del software. Además, se valoraron los diferentes métodos de pruebas que existen y los modelos para la elaboración de procedimientos que permitan desarrollar pruebas durante todo el ciclo de vida de una aplicación Web.

Como resultado de esta investigación se hace la propuesta de tres procedimientos que garantizan una mejor revisión durante las tres últimas fases del ciclo de vida de las Aplicaciones Web. Estos permitirán normalizar las pruebas en el desarrollo de las Aplicaciones Web que son implementadas en la Facultad Siete y se proponen mejoras para que este trabajo en un futuro sea realizado con mayor facilidad.

## Índice

<b>Introducción</b> .....	1
<b>Fundamentación Teórica</b> .....	5
<b>1.1 Introducción</b> .....	5
<b>1.2 Conceptos Básicos</b> .....	5
<b>1.3 Proceso de Pruebas</b> .....	6
1.3.1 Objetivos de un Proceso de Prueba. ....	7
1.3.2 Estrategia de Prueba.....	7
1.3.3 Aspectos Estratégicos para las Pruebas.....	8
<b>1.4 Clasificación de Pruebas</b> .....	9
1.4.1 Prueba de Caja Negra: .....	9
1.4.2 Prueba de Caja Blanca:.....	10
1.4.3 Técnicas o Métodos de Prueba de Caja Negra y Caja Blanca.....	11
1.4.4 Tipos de Pruebas durante las Fases de Depuración y Cambios.....	13
1.4.4.1 Pruebas de Integración. ....	13
1.4.4.2 Pruebas de Sistema.....	14
1.4.4.3 Pruebas de Aceptación y/o Liberación. ....	15
<b>1.5 Herramientas para el entorno de Pruebas</b> .....	16
<b>1.6 Modelos Usados para la aplicación de Pruebas</b> .....	18
1.6.1 Modelo en Cascada .....	18
1.6.2 Modelo en V .....	21
1.6.3 Modelo en W .....	22

<b>1.7 Procedimiento de Prueba</b> .....	23
<b>2.1 Introducción</b> .....	29
<b>2.2 Caracterización de la producción de software en la Facultad Siete</b> .....	29
<b>2.4 Diagnóstico de la producción de software en la Facultad Siete</b> .....	36
2.4.1 Análisis del resultado de las Entrevistas realizadas en cada proyecto a los responsables del grupo interno de calidad y a un miembro del mismo grupo.....	36
2.4.2 Relación de preguntas y respuestas de los entrevistados. ....	38
<b>2.5 Análisis del resultado de la revisión de los documentos de las áreas temáticas</b> .....	42
<b>3.1 Introducción</b> .....	45
<b>3.2 Fase Depuración y Cambios; Ejecución de las Pruebas de Integración</b> .....	48
3.2.1 Propósito: .....	49
3.2.2 Alcance:.....	50
3.2.3 Descripción General del Procedimiento .....	50
3.2.4 Roles del Procedimiento .....	50
3.2.5 Artefactos de entrada .....	51
3.2.6 Artefactos de salida .....	51
3.2.7 Fases del Procedimiento: .....	52
3.2.7.1 Fase #1: Análisis de los requerimientos de integración .....	53
3.2.7.2 Fase #2: Diseño de Casos de Prueba.....	54
3.2.7.3 Fase #3: Validar y ejecutar el diseño de los Casos de Prueba .....	55
<b>3.3 Fase Depuración y Cambios; Ejecución de las Pruebas de Sistema</b> .....	57
3.3.1 Propósito: .....	57
3.3.2 Alcance:.....	57
3.3.3 Descripción General de Procedimiento.....	58

3.3.4 Roles del Procedimiento.....	58
3.3.5 Artefactos de Entrada.....	59
3.3.6 Artefactos de Salida.....	59
3.3.7 Fases del Procedimiento:.....	59
3.3.7.1 Fase #1: Inicio de las Pruebas de Sistema.....	59
3.3.7.2 Fase #2: Ejecución de las Pruebas de Sistema.....	60
3.3.7.3 Fase #3: Conclusión y Validación de las Pruebas de Sistema.....	64
<b>3.4 Fase Depuración y Cambios; Ejecución de las Pruebas de Aceptación y/o Liberación.....</b>	<b>65</b>
3.4.1 Propósito:.....	66
3.4.2 Alcance:.....	67
3.4.3 Descripción General de Procedimiento.....	67
3.4.4 Roles del Procedimiento.....	67
3.4.5 Artefactos de Entrada.....	68
3.4.6 Artefactos de Salida.....	68
3.4.7 Fases del Procedimiento:.....	69
3.4.7.1 Fase #1: Inicio del Proceso de Pruebas de Aceptación.....	69
3.4.7.2 Fase #2: Revisión de Documentos y Diseño de Casos de Prueba.....	70
3.4.7.3 Fase #3: Aplicación de los Casos de Prueba y Listas de Chequeos.....	71
3.4.7.4 Fase #4: Liberación del Software.....	72
<b>Conclusiones.....</b>	<b>76</b>
<b>Recomendaciones.....</b>	<b>77</b>
<b>Referencias Bibliográfica.....</b>	<b>78</b>
<b>Bibliografía.....</b>	<b>80</b>

<b>Glosario de Términos .....</b>	<b>82</b>
<b>Anexos.....</b>	<b>84</b>

## **Introducción**

El uso de las Nuevas Tecnologías de la Información a nivel mundial es una realidad y uno de los motores impulsores de la sociedad actual. En los últimos años la sociedad ha vivido una verdadera revolución tecnológica y social derivada de los avances obtenidos en el campo de la Informática.

Este avance forma parte de una revolución de nivel superior, que otorga de forma general al desarrollo en el campo de las comunicaciones un futuro en el que la información abandonará los formatos y lugares en los que actualmente se almacena, para estar distribuida globalmente y a disposición del que desee acceder a ella.

En el mundo, los sistemas informáticos, alcanzan un nivel de desarrollo que está a la altura de los problemas que resuelven. Debido a la evolución de la tecnología y las comunicaciones, el software de gestión de datos necesita estar al mismo nivel de otras aplicaciones que revolucionan el mundo actual, por esto es importante proveer un ambiente para el aumento de las capacidades de un sistema.

La necesidad de que cada sistema informatizado tenga la calidad requerida para su uso, se ha vuelto en la actualidad uno de los principales objetivos estratégicos de las organizaciones debido a que; su supervivencia depende cada vez más de los sistemas informáticos para su buen funcionamiento.

La calidad en cada uno de los sistemas que surgen en el mundo tiene un costo asociado y las pruebas como proceso para controlar la calidad consumen tiempo, personal y otros recursos. Como en todo proyecto de cualquier índole, siempre se debe tratar que las fallas sean mínimas y al menor costo posible. La fase de prueba añade valor al producto que se maneja: todos los programas tienen errores y esta fase los descubre; además de tener como objetivo principal encontrar la mayor cantidad de errores y defectos existentes. Este es uno de los principios que debe seguirse para la mejora de la calidad de productos y procesos de software.

En Cuba, se pretende llevar el desarrollo de la informática hacia todos los sectores, logrando la informatización de la sociedad con sistemas donde la calidad es un objetivo primordial para obtener un

mejor desarrollo de estos. Actualmente, la Universidad de las Ciencias Informáticas (UCI) lleva a cabo un gran movimiento de desarrolladores web, los cuales implementan aplicaciones que se usan tanto en el país como fuera de este, las que necesitan un amplio proceso de pruebas en todo su ciclo de vida. Para mejorar este proceso existen propuestas encaminadas a indicar las actividades a realizar por los equipos de calidad de cada facultad, quienes se encargan de obtener un producto con la calidad requerida.

La Facultad Siete centra el desarrollo de software en dos perfiles fundamentales; sistemas informáticos para la salud y el procesamiento de imágenes. Para garantizar la gestión de la calidad de dichos sistemas existe un equipo de trabajo encargado de llevar a cabo acciones para lograr este objetivo. Muchas de estas aplicaciones son desarrolladas en ambiente web, las cuales son más sencillas, económicas y a menudo más poderosas que las de escritorio. Además pueden ser usadas por múltiples usuarios al mismo tiempo y presentan facilidad de pruebas.

Muchos de estos productos han sido revisados por el área de calidad detectándose algunas dificultades como:

- ✓ La inexistencia de un diseño coherente para las pruebas.
- ✓ No se realizan Pruebas de Integración.
- ✓ No se realizan Pruebas de Aceptación y/o Liberación.

Las dificultades mencionadas anteriormente se mantienen vigentes en los proyectos productivos de la Facultad Siete debido a que no son utilizados los procedimientos existentes para realizarles pruebas al software antes de ser entregados al proyecto de Calidad. Además de esto surge una problemática, que es la difícil revisión al sistema ya que los procedimientos son muy generales lo que se hace necesario tener procedimientos específicos para hacerles revisiones a cada fase durante la vida del sistema.

La idea de realizar procedimientos específicos a las fases de Depuración y Cambios surge de la tesis propuesta en el año anterior con el nombre " Propuesta de procedimiento general para normalizar las pruebas en el desarrollo de Aplicaciones Web en la Facultad Siete", ya que en dicho trabajo se realiza un procedimiento en general para la realización de pruebas durante las fases propuestas por el Modelo en W pero no se enfatiza en sus tres últimas fases las cuales son muy importantes ya que en ellas es donde se

prueba al software como un conjunto después de haber integrado todos los componentes del mismo, además de que se realizan todas las pruebas pertinentes al sistema y se determina la aceptación del producto por parte del usuario final.

Partiendo de esta situación, se plantea como **problema a resolver**:

¿Cómo normalizar las fases de Depuración y Cambios (Ejecución de las Pruebas de Integración, Sistema, Liberación y/o Aceptación) en el desarrollo de Aplicaciones Web en la Facultad Siete?

Conociendo el problema, se define como **objeto de estudio** el Procedimiento para normalizar las pruebas, y **el campo de acción** se enmarca en las Fases de Depuración y Cambios (Ejecución de las Pruebas de Integración, Sistema, Liberación y/o Aceptación).

Teniendo en cuenta lo antes expuesto el **objetivo general** de la investigación es: Proponer procedimientos para normalizar las fases de Depuración y Cambios (Ejecución de las Pruebas de Integración, Sistema, Liberación y/o Aceptación) en el desarrollo de Aplicaciones Web en la Facultad Siete.

Para lograr cumplir con el objetivo de esta investigación y dar solución a la problemática antes expuesta, se necesitan realizar las siguientes **tareas**:

1. Determinar los fundamentos teóricos metodológicos para la realización de procedimientos que normalicen las fases de Depuración y Cambios (Ejecución de las Pruebas de Integración, Sistema, Liberación y/o Aceptación) en el desarrollo de Aplicaciones Web en la Facultad Siete.
2. Analizar el "Procedimiento general para normalizar las pruebas en el desarrollo de Aplicaciones Web" en las fases de Depuración y Cambios (Ejecución de las Pruebas de Integración, Sistema, Liberación y/o Aceptación) si se utiliza algún procedimiento en los proyectos productivos de la Facultad Siete.
3. Entrevistar a profesionales que tengan experiencia en la aplicación de métricas, Listas de Chequeos y otras herramientas de calidad en las etapas objetos de este trabajo.
4. Analizar las herramientas, Listas de Chequeos u otros métodos aplicables a las fases indicadas que sirvan como base para la propuesta de procedimientos y mejoras de dichas fases.

5. Evaluar toda la información recopilada sobre herramientas, Listas de Chequeos u otros métodos aplicables a las fases de Depuración y Cambios (Ejecución de las Pruebas de Integración, Sistema, Liberación y/o Aceptación).
6. Confeccionar la Propuesta de procedimientos a las fases de Depuración y Cambios (Ejecución de las Pruebas de Integración, Sistema, Liberación y/o Aceptación).

## Fundamentación Teórica

### 1.1 Introducción

En el presente capítulo se abordan los principales conceptos y aspectos significativos relacionados con las principales temáticas de las diferentes fuentes bibliográficas consultadas, referentes al estado del arte que presenta el tema que se investiga. También se hace mención de los dos grandes tipos de pruebas “Prueba de Caja Blanca y Prueba de Caja Negra” de la última se abordan sus técnicas usadas que pueden ser aplicadas en productos Web, teniendo en cuenta el uso de herramientas existentes a nivel mundial.

Además se presenta un estudio de los diferentes tipos de pruebas a realizar durante las fases de “Depuración y Cambios”, así como los modelos que permiten la elaboración de un procedimiento de prueba que pueda ser aplicado durante las fases de “Depuración y Cambios” de una Aplicación Web.

### 1.2 Conceptos Básicos

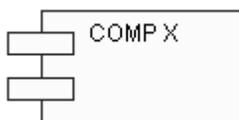
#### Calidad de Software

“Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente” (1)

#### Componente

Un componente (**Ver Figura 1.0**) según la definición de Sterling Software es:

- ✓ Una parte reutilizable que encapsula elementos del modelo (por ejemplo una DLL, documentos, tablas, biblioteca, etc.).
- ✓ Un paquete de software el cual ofrece servicios a través de sus interfaces.
- ✓ Un paquete de Software que puede ser usado para construir aplicaciones o componentes más grandes. (Software 2000) (2)



**Figura 1.0 Representación de Componente**

## Procedimiento

Un Procedimiento es el modo de ejecutar determinadas acciones que suelen realizarse de la misma forma, con una serie común de pasos claramente definidos, que permiten el cumplimiento de un objetivo específico.

## Pruebas de Software

Las **Pruebas de software** son los procesos que permiten verificar y revelar la calidad de un producto software.

Las pruebas son actividades en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos específicos, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente. (3)

Ejecución de un programa con la intención de descubrir un error. Técnica experimental para la búsqueda de errores en los programas. (4)

Una “Prueba” se define como un proceso de evaluación de un producto desde un punto de vista crítico, donde el probador somete el producto a una serie de acciones inquisitivas, y el producto responde con su comportamiento como reacción. Por supuesto, nunca se debe probar el software en un entorno de producción. Es necesario probar los nuevos programas en un entorno de pruebas separado físicamente del de producción.

Se define además como “Prueba” al proceso que se enfoca sobre la lógica interna del software y las funciones externas. Es un proceso de ejecución de un programa con la intención de descubrir un error. Un buen caso de prueba es aquel que tiene alta probabilidad de mostrar un error no descubierto hasta entonces. Una prueba tiene éxito si descubre un error no detectado hasta entonces. (5)

### 1.3 Proceso de Pruebas

Es el proceso de ejecutar un programa con la intención de encontrar fallos.

El proceso de pruebas de software es una tarea técnica, precisa del dominio del lenguaje de programación en el que el artefacto a probar fue generado, también del conocimiento necesario para comprender la arquitectura del sistema implementado y de las implicaciones de tipo lógico que su diseño pueda suponer.

Adicionalmente, el probador deberá conocer los lenguajes y herramientas que ha de usar para llevar a cabo este proceso de pruebas. (6)

### 1.3.1 Objetivos de un Proceso de Prueba.

- ✓ Encontrar defectos en el software que no hayan sido detectados anteriormente.
- ✓ Verificar que el software cumpla con las descripciones hechas en los documentos que describen su funcionalidad.
- ✓ Verificar la interacción de componentes.
- ✓ Verificar que se realice correctamente la Integración de Componentes.
- ✓ Verificar la buena implantación de todos los requisitos que debe cumplir el software.
- ✓ Asegurar que antes de la liberación del sistema todos los defectos se hayan corregido.
- ✓ Verificar que un gran conjunto de partes de software funcionan juntos. (7)

### 1.3.2 Estrategia de Prueba.

Se puede ver el proceso de la Ingeniería de Software como una espiral (**Ver Figura 1.1**). En el inicio la ingeniería define los papeles del software que conduce al análisis de los requerimientos de este, se establece el dominio de información así como la función, comportamiento, rendimiento, restricciones y los criterios de validación del software. Al moverse hacia el interior de la espiral se llega al diseño y por último a la codificación, para desarrollar un software se da vueltas en la espiral a través de los flujos, transitando por diferentes niveles de pruebas Unidad, Integración, Sistema, Implantación y Aceptación.

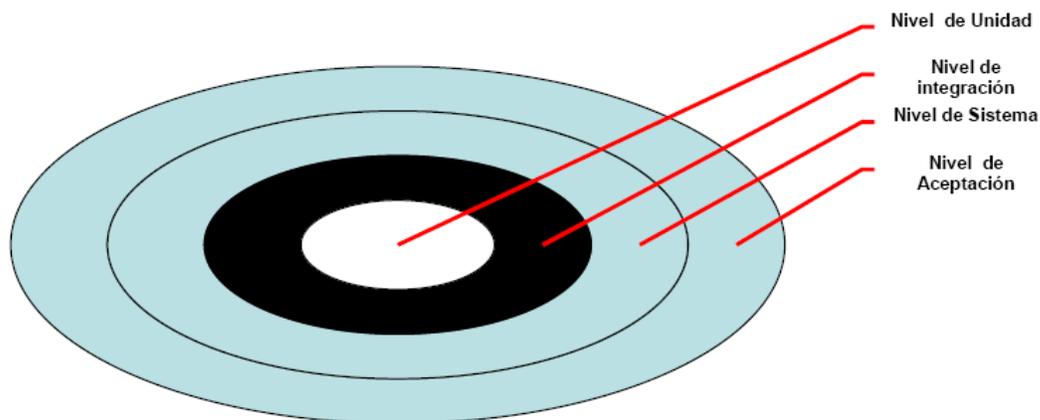
El nivel de Unidad comienza en el vértice de la espiral y se centra en cada unidad del software según como esté implementada en el código fuente. En este nivel se prueba el comportamiento de cada uno de los componentes de forma independiente, estas pruebas se ejecutan una vez realizada la implementación en el componente. Se prueba la funcionalidad de una clase o conjunto de clases correlacionadas, es decir aquí se ejecutan métodos de Casos de Prueba de Caja Blanca. La prueba avanza moviéndose hacia fuera de la espiral y llega al nivel de Integración.

Este nivel de prueba se centra en la ejecución de métodos de Caja Negra aunque se pueden llevar a cabo métodos de pruebas de Caja Blanca para asegurar que se cumplen los principios de caminos de control.

Además de la construcción de la arquitectura del software, su objetivo es probar la integración de los módulos del sistema, una vez que a estos se les hayan aplicado Casos de Prueba de unidad, se deberá verificar que estos interactúan correctamente a través de las funcionalidades expuestas en sus interfaces, se verifican las interfaces entre las partes de una arquitectura.

Este tipo de prueba deberá realizarse durante la fase de construcción, una vez sea implementado el componente. El proceso continúa dando otra vuelta hacia fuera y se encuentra el nivel de Sistema el cual se centra en elementos de sistema, en comprobar la funcionalidad del sistema como un todo, así como verificar fundamentalmente los requisitos no funcionales definidos para el producto, por ejemplo, que el sistema puede gestionar los volúmenes de información requeridos, se ajusta a los tiempos de respuesta deseados y que los procedimientos de respaldo, seguridad e interfaces con otros sistemas funcionan correctamente.

Se debe verificar también el comportamiento del sistema bajo las condiciones más extremas. Al avanzar una vuelta más hacia fuera se encuentra el nivel de Aceptación, el cual se centra en verificar formalmente con el cliente que el sistema satisface todas sus necesidades.



**Figura 1.1 Niveles de Prueba**

### 1.3.3 Aspectos Estratégicos para las Pruebas

Pressman plantea que se deben abordar los siguientes puntos si se desea implementar con éxito una estrategia de prueba del software: (8)

- ✓ Especificar los requisitos del producto de manera confiable mucho antes de que comiencen las pruebas. Aunque el objeto principal de una buena estrategia también evalúa otras características de la calidad tales como la portabilidad, facilidad de mantenimiento y facilidad de uso.
- ✓ Establecer los objetivos de la prueba de manera explícita: Se deberían establecer en términos medibles los objetivos específicos de la prueba. Por ejemplo, la efectividad de la prueba, la cobertura de la prueba, tiempo medio de fallo, el costo para encontrar y arreglar errores, densidad de fallos remanente o frecuencia de ocurrencia, y horas de trabajo por prueba de regresión que deberían establecerse dentro de la planificación de la prueba.
- ✓ Construir un software robusto. El software deberá diseñarse de manera que use técnicas de depuración es decir, el software debería ser capaz de diagnosticar ciertas clases de errores. Además, el diseño debería incluir pruebas automatizadas y pruebas de regresión.

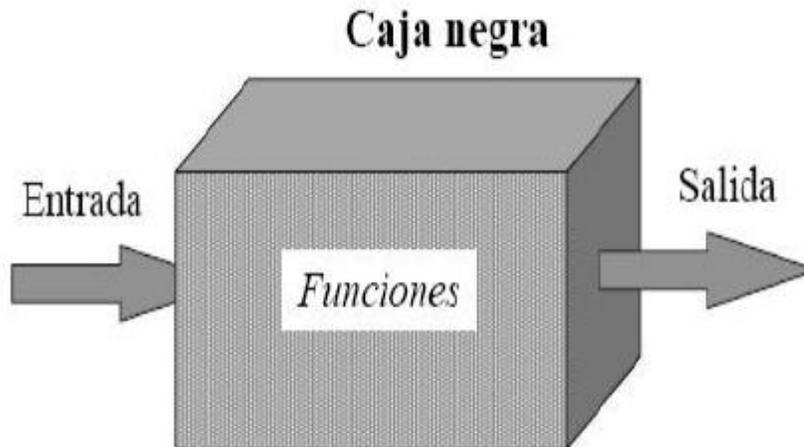
Desarrollar un enfoque de mejora continua al proceso de prueba. Debería medirse la estrategia de prueba. Las métricas agrupadas durante la prueba deberían usarse como parte de un enfoque estadístico de control del proceso para la prueba del software.

#### **1.4 Clasificación de Pruebas.**

Para la realización de un software es necesario cumplir con todos los requisitos funcionales pero es muy importante que se realice con la mejor calidad posible, para esto existen dos tipos de pruebas que logran que se detecten la mayor cantidad de errores en la aplicación entre las que se encuentran:

##### **1.4.1 Prueba de Caja Negra:**

La Prueba de Caja Negra (como se ilustra en la **figura1.2**): se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los Casos de Prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.



**Figura 1.2 Método de Prueba de Caja Negra**

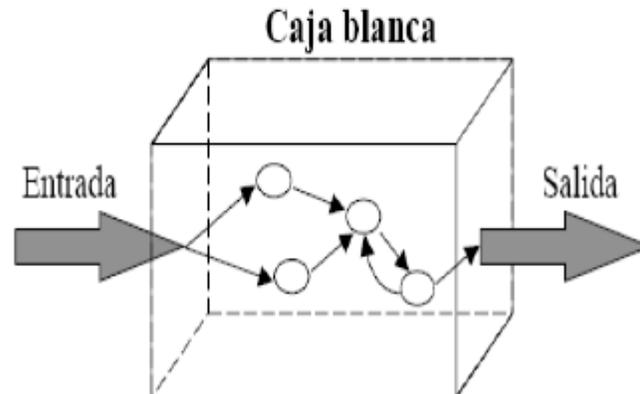
#### **1.4.2 Prueba de Caja Blanca:**

La prueba de Caja Blanca (como se ilustra en la **Figura 1.3**) del software consiste en comprobar los caminos lógicos del software proponiendo Casos de Prueba que se ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coinciden con el esperado o mencionado.

Requieren del conocimiento de la estructura interna del programa y son derivadas a partir de las especificaciones internas de diseño o el código.

Mediante los métodos de prueba de la Caja Blanca, el ingeniero de software puede obtener Casos de Prueba que garanticen que:

- ✓ Se ejerciten por lo menos una vez todos los caminos independientes para cada módulo.
- ✓ Se ejerciten todas las decisiones lógicas en sus vertientes verdaderas y falsa.
- ✓ Ejecuten todos los bucles en sus límites y con sus límites operacionales.
- ✓ Se ejerciten las estructuras internas de datos para asegurar su validez.



**Figura 1.3 Método de Prueba de Caja Blanca**

Dado las alternativas de las pruebas de Caja Negra y las pruebas de Caja Blanca y de la necesidad de que las pruebas abarquen los requerimientos, las funciones y la lógica interna del programa entonces:

- ✓ Las pruebas para los requerimientos deben emplear la estrategia de la Caja Negra.
- ✓ Las pruebas para las funciones deben emplear la estrategia de la Caja Negra.
- ✓ Las pruebas para la lógica interna tiene que necesariamente emplear la estrategia de la Caja Blanca.

#### **1.4.3 Técnicas o Métodos de Prueba de Caja Negra y Caja Blanca.**

##### **Métodos de pruebas basados en Caja Blanca:**

##### **La prueba del camino básico:**

Esta prueba permite al diseñador de Casos de Prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución.

Los Casos de Prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa.

**Componentes del Grafo de Flujo:**

**Arista:** Representa flujo de control.

**Nodo:** representa un conjunto de sentencias.

**Región:** Área limitada por aristas y nodos.

Se considera como un área + lo que esta fuera del grafo.

**Tipo de Prueba adecuada al Nivel de Integración:****Prueba basada en hilos:**

Integrar las clases necesarias para cumplir eventos del sistema.

**Prueba basada en uso:**

Integrar las clases independientes primero, y luego las dependientes.

**Métodos de pruebas basados en Caja Negra:****Técnica de la Partición de Equivalencia:**

Esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software. (9)

**Técnica del Análisis de Valores Límites:**

La técnica de Análisis de Valores Límites selecciona Casos de Prueba que ejerciten los valores límites.

La misma complementa la prueba de Partición Equivalente, pero en lugar de realizar la prueba con cualquier elemento de la partición equivalente, se escogen los valores en los bordes de la clase. Más que concentrarse únicamente en el dominio de entrada (condiciones de entrada), los Casos de Prueba se generan considerando también el espacio de salida.

**Reglas:**

- 1) Si una condición de entrada especifica un rango de valores, se deben generar casos para los extremos del rango y casos no válidos para situaciones justo más allá de los extremos.
- 2) Si la condición de entrada especifica un número de valores, hay que escribir casos para los números máximo, mínimo, uno más del máximo y uno menos del mínimo de valores.
- 3) Usar la regla 1 para la condición de salida.

4) Usar la regla 2 para cada condición de salida.

En esta regla 3 y 4, debe recordarse que:

- ✓ Los valores límite de entrada no generan necesariamente los valores límite de salida.
- ✓ No siempre se pueden generar resultados fuera del rango de salida (pero es interesante considerarlo).

5) Si la entrada o la salida de un programa es un conjunto ordenado (por ejemplo, una tabla, un archivo secuencial), los casos deben concentrarse en el primero y en el último elemento.

#### **1.4.4 Tipos de Pruebas durante las Fases de Depuración y Cambios.**

Durante las fases de Depuración y Cambios es necesario para garantizar la buena calidad del producto que se lleven a cabo una serie de pruebas las cuales se denominan:

##### **1.4.4.1 Pruebas de Integración.**

Las Pruebas de Integración son realizadas sobre agrupaciones de componentes o funciones.

Se examinan las interfaces para asegurar que estos componentes individuales son llamados cuando es necesario y que los datos que se transmiten entre dichos componentes son los requeridos.

Estas pruebas tienen como propósito asegurar que no hay errores de interfaces y encontrar defectos en el sistema.

#### **Desarrollo Incremental**

El tipo de prueba incremental consiste en agregar cada módulo o componente individual al conjunto de componentes existentes y el conjunto resultante se prueba.

Es importante determinar la secuencia en que se van a producir e integrar los componentes para esto existen varias estrategias de integración tales como:

##### **Estrategia de arriba a abajo (top-down).**

- ✓ El primer componente o módulo que se desarrolla y prueba es el que está arriba en la jerarquía.

- ✓ Los módulos de nivel más bajo se sustituyen por módulos auxiliares que simulan el resto de la funcionalidad.

Una ventaja de esta estrategia es que las interfaces entre los módulos son probadas en una fase temprana y con frecuencia.

### **Estrategia de abajo a arriba (bottom-up)**

- ✓ En este caso se crean primero los componentes de más bajo nivel y se crean módulos conductores para simular las invocaciones.
- ✓ Los módulos auxiliares son necesarios en raras ocasiones.

Este tipo de enfoque de las pruebas permite un desarrollo más paralelizado que el enfoque el Top-Down, pero presenta mayores dificultades a la hora de planificar y de gestionar las pruebas.

Cuando las pruebas unitarias y de Integración se realizan separadamente es difícil identificar los componentes individuales o módulos que causan resultados incorrectos.

Lo más probable es que los problemas que surjan al incorporar un nuevo componente, surjan de este último o de las interfaces entre él y los otros componentes. (10)

#### **1.4.4.2 Pruebas de Sistema.**

La fase de Pruebas de Sistema tiene como objetivo verificar el sistema software para comprobar si este cumple sus requisitos. Dentro de esta fase pueden desarrollarse varios tipos de pruebas en función de los objetivos de las mismas. Algunos tipos son: Pruebas Funcionales, de Usabilidad, Rendimiento y Pruebas de Seguridad. Este trabajo se centra en Pruebas Funcionales de aplicaciones con interfaces gráficas. Estas pruebas verifican que el sistema software ofrece a los actores humanos la funcionalidad recogida en su especificación.

En fin con las **Pruebas de Sistema** se verifica que cada elemento encaje de forma adecuada y que se alcance la funcionalidad y el rendimiento del sistema total. La Prueba de Sistema está constituida por una

serie de pruebas diferentes cuyo propósito primordial es ejercitar profundamente el sistema basado en computadora. Algunas de estas pruebas son: Pruebas de Carga, Volumen, Estabilidad y Seguridad.

#### **Pruebas de Desempeño (carga):**

Las pruebas de desempeño son una actividad importante del desarrollo de sistemas pues evitan que las aplicaciones de software alcancen su máximo potencial en cuanto a desempeño. Finalmente, cabe recalcar que una prueba de desempeño es un proceso iterativo. Se prueba, recolectan datos, se verifican, analizan y se optimiza. Este proceso ocurre repetidamente hasta que se alcancen los objetivos de todos los escenarios de prueba. (11)

#### **Pruebas de Volumen:**

Consisten en examinar el funcionamiento del sistema cuando está trabajando con grandes volúmenes de datos, simulando la carga de trabajo esperada. (12)

#### **Pruebas de Estabilidad:**

Consisten en dejar corriendo la aplicación durante periodos de ocho horas y registrar si ocurrieron fallas y hacer una estadística de tiempo contra número de errores. La razón de correr ciclos de ocho horas y no de más tiempo se debe a las limitaciones del tiempo de desarrollo del sistema. (13)

#### **Pruebas de Seguridad:**

Verifica que los mecanismos de seguridad del software son válidos y confiables luego de esto le brinda al software garantía y robustez.

Consisten en verificar los mecanismos de control de acceso al sistema para evitar alteraciones indebidas en los datos.

#### **1.4.4.3 Pruebas de Aceptación y/o Liberación.**

El objetivo de las Pruebas de Aceptación es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento.

Las Pruebas de Aceptación son definidas por el usuario del sistema y preparadas por el equipo de desarrollo, aunque la ejecución y aprobación final corresponden al usuario.

La validación del sistema se consigue mediante la realización de pruebas de Caja Negra que demuestran la conformidad con los requisitos y que se recogen en el plan de pruebas, el cual define las verificaciones a realizar y los Casos de Prueba asociados. Dicho plan está diseñado para asegurar que se satisfacen todos los requisitos funcionales especificados por el usuario teniendo en cuenta también los requisitos no funcionales relacionados con el rendimiento, seguridad de acceso al sistema, a los datos y procesos, así como a los distintos recursos del sistema.

Cuando se construye software a medida para un cliente, se lleva a cabo una serie de Pruebas de Aceptación para permitir que el cliente valide todos los requisitos. La mayoría de los desarrolladores de productos de software llevan a cabo un proceso denominado **Pruebas Alfa y Beta** para descubrir errores que parezca que sólo el usuario final puede descubrir.

- **Pruebas Alfa:**

Se lleva a cabo, por un cliente, en el lugar de desarrollo. Se usa el software de forma natural con el desarrollador como observador del usuario y registrando los errores y problemas de uso. Las Pruebas Alfa se llevan a cabo en un entorno controlado.

- **Pruebas Beta:**

Se llevan a cabo por los usuarios finales del software en los lugares de trabajo de los clientes. A diferencia de las Pruebas Alfa, el desarrollador no está presente normalmente. Así, la Prueba Beta es una aplicación en vivo del software en un entorno que no puede ser controlado por el desarrollador. El cliente registra todos los problemas que encuentra durante la prueba beta e informa a intervalos regulares al desarrollador.

### **1.5 Herramientas para el entorno de Pruebas.**

Dan soporte a las **Pruebas de Integración, Sistema, diagnóstico o afinado de las aplicaciones** en los entornos de preproducción o certificación y los de producción, que generalmente son difíciles de realizar de una forma integrada.

A continuación se relacionan las herramientas que más se utilizan en la realización de pruebas:

### **CheckKing de ALS**

CheckKing es la herramienta de monitorización del proceso de desarrollo software y sus resultados, que cubre las necesidades de organizaciones que desean controlar la calidad del software antes de su puesta en producción. (14)

### **QACenter Enterprise Edition de Compuware**

QACenter Enterprise Edition analiza cómo se distribuye el tiempo y permite a los equipos de calidad tomar decisiones estudiadas acerca del cambio de prioridades, cuánto tiempo adicional es necesario o cuántos recursos adicionales se requieren para cumplir los plazos establecidos. (15)

### **JKing QA de ALS**

Jking QA es una herramienta de análisis estático, pensada para facilitar y automatizar el proceso de adopción de los estándares de calidad para un departamento de calidad. (16)

Además de existir estas herramientas para la aplicación de pruebas, las que se van a usar para la realización de los procedimientos de este trabajo son:

### **JMeter**

Es una herramienta de Apache que es buena a la hora de hacer Pruebas de Estrés de Aplicaciones Web. Con esta herramienta se puede atacar aquellos puntos que se quieran probar con respecto a concurrencia de usuarios y velocidad de carga en un sitio.

JMeter no se limita a pruebas de carga para Aplicaciones Web. Incluso puede servir para realizar pruebas de estrés sobre drivers para base de datos (realmente el hecho de hacer benchtests para bases de datos no arrojaría resultados fiables para comparaciones entre motores, puesto que dependería de los drivers), por lo que las pruebas para JDBC deben enfocarse para medir el nivel de fiabilidad del driver con respecto a la base de datos. (17)

## **JUnit**

JUnit es un conjunto de clases (framework) que permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera. Es decir, en función de algún valor de entrada se evalúa el valor de retorno esperado; si la clase cumple con la especificación, entonces JUnit devolverá que el método de la clase pasó exitosamente la prueba; en caso de que el valor esperado sea diferente al que regresó el método durante la ejecución, JUnit devolverá un fallo en el método correspondiente.

JUnit es también un medio de controlar las pruebas de regresión, necesarias cuando una parte del código ha sido modificado y se desea ver que el nuevo código cumple con los requerimientos anteriores y que no se ha alterado su funcionalidad después de la nueva modificación.

### **1.6 Modelos Usados para la aplicación de Pruebas.**

Para llegar al cumplimiento de una determinada tarea es necesario realizar una secuencia de pasos con un orden lógico, para que la actividad se cumpla con logros satisfactorios, al igual debe existir a la hora de llevar a cabo el desarrollo de un software, el cual se puede desarrollar por modelos de procesos, estos son importantes para la guía del mismo.

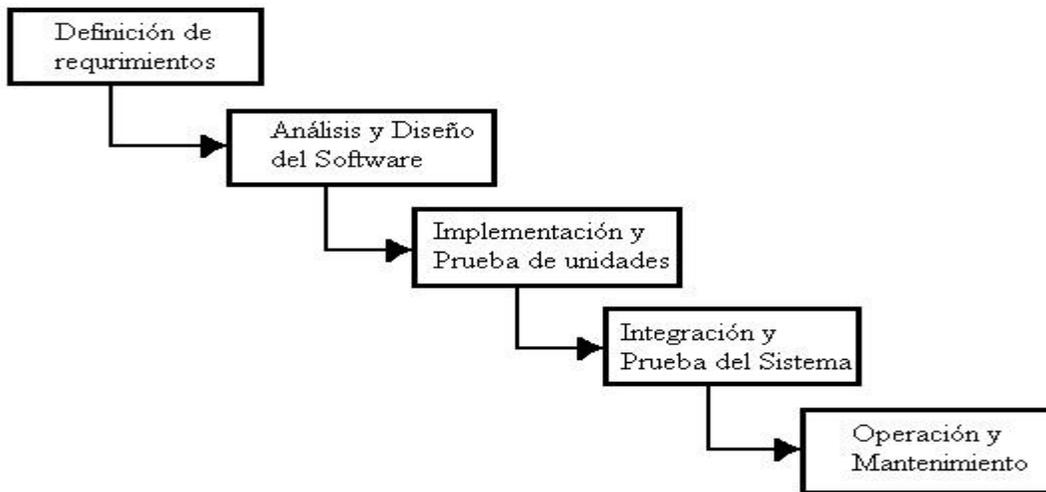
Según **Somerville** se define el modelo de proceso de software como “Una representación simplificada de un proceso de software, representada desde una perspectiva específica. Por su naturaleza los modelos son simplificados, por lo tanto un modelo de procesos del software es una abstracción de un proceso real.”

A continuación se mencionan una serie de modelos que constituyen la base para la elaboración de cualquier procedimiento de pruebas de software, los que pueden adaptarse a cualquier metodología de desarrollo como es el caso de RUP.

#### **1.6.1 Modelo en Cascada**

El **desarrollo en cascada** (Ver figura 1.4), también llamado **modelo en cascada**, es el enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de forma tal que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior.

Este modelo consta de seis fases las cuales son:



**Figura 1.4 Modelo en Cascada**

## **Análisis de requisitos**

El proceso de recopilación de los requisitos se centra e intensifica especialmente en el software. El ingeniero de software (Analistas) debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas. (18)

## **Diseño del Sistema**

El diseño del software se enfoca en cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación. (19)

## **Codificación**

El diseño debe traducirse en una forma legible para la máquina. El paso de codificación realiza esta tarea. Si el diseño se realiza de una manera detallada la codificación puede realizarse mecánicamente. (20)

## **Pruebas**

Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente antes de ser puesto en explotación. (21)

## **Implantación**

El software sufrirá cambios después de que se entrega al cliente. Los cambios ocurrirán debidos a que se hayan encontrado errores, a que el software deba adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos), o debido a que el cliente requiera ampliaciones funcionales o del rendimiento. (22)

## **Ventajas del Modelo en Cascada**

- ✓ Se tiene todo bien organizado y no se mezclan las fases.
- ✓ Es perfecto para proyectos que son rígidos, y además donde se especifiquen muy bien los requerimientos y se conozca muy bien la herramienta a utilizar.

### Desventajas del Modelo en Cascada

- ✓ Los proyectos reales raramente siguen el flujo secuencial que propone el modelo, siempre hay iteraciones y se crean problemas en la aplicación del paradigma.
- ✓ Normalmente, es difícil para el cliente establecer explícitamente al principio todos los requisitos. El ciclo de vida clásico lo requiere y tiene dificultades en acomodar posibles incertidumbres que pueden existir al comienzo de muchos productos.
- ✓ El cliente debe tener paciencia. Hasta llegar a las etapas finales del proyecto, no estará disponible una versión operativa del programa. Un error importante no detectado hasta que el programa esté funcionando puede ser desastroso. (23)

### 1.6.2 Modelo en V

El Modelo V (Ver Figura 1.5) tiende a ser muy relacionado con el Modelo de Cascada puesto que es una evolución del mismo.

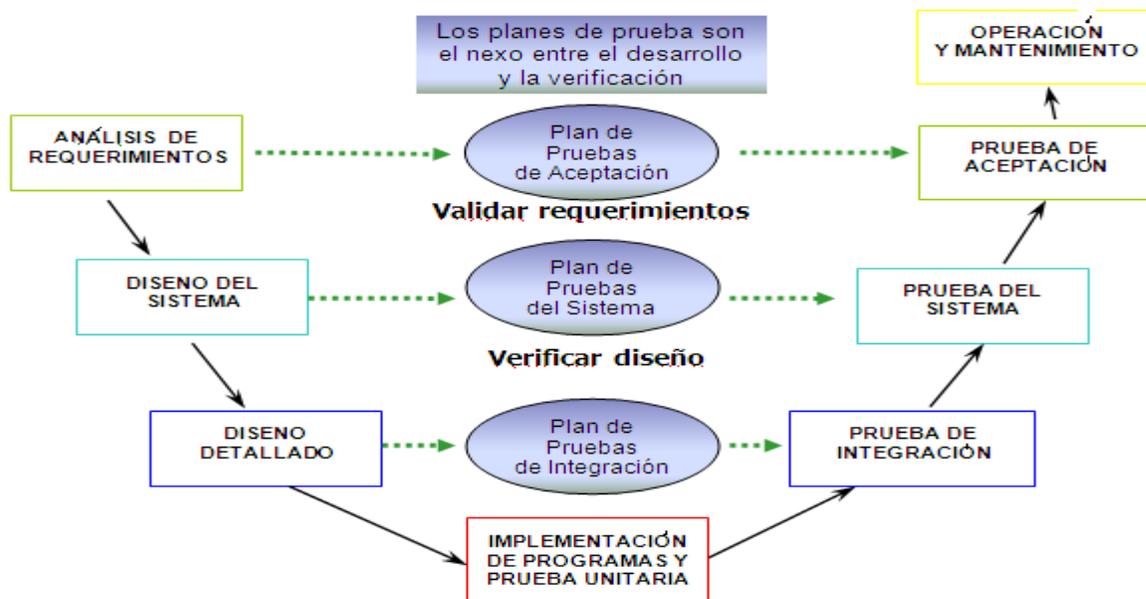


Figura 1.5 Modelo en V

Puede notarse que su primera mitad es similar al Modelo en Cascada, y la otra mitad tiene como finalidad hacer pruebas e integración asociado a cada una de las etapas de la mitad anterior.

Se puede identificar una ventaja principal con respecto al Modelo Cascada más simple, y se refiere a que este modelo involucra chequeos de cada una de las etapas del modelo de cascada.

### Desventajas del Modelo en V

- ✓ El riesgo es mayor que el de otros modelos, pues en lugar de hacer Pruebas de Aceptación al final de cada etapa, las pruebas comienzan a efectuarse luego de haber terminado la implementación, lo que puede traer como consecuencia un “roll-back” de todo un proceso que costó tiempo y dinero.
- ✓ El modelo no contempla la posibilidad de retornar a etapas inmediatamente anteriores, cosa que en la realidad puede ocurrir.
- ✓ Se toma toda la complejidad del problema de una vez y no en iteraciones o ciclos de desarrollo, lo que disminuye el riesgo.

A pesar de todo lo antes mencionado, se trata de un modelo más robusto y completo que el Modelo de Cascada, y puede producir software de mayor calidad que con el modelo de cascada. (24)

### 1.6.3 Modelo en W

El modelo en W surge como refinamiento del modelo en V. Refleja mejor la interdependencia que existe entre equipo de desarrollo y el equipo de pruebas a lo largo de todo el proceso de desarrollo del sistema destacando los siguientes dos puntos:

- ✓ En las primeras etapas se consideran labores de pruebas que no aparecen reflejadas en el modelo original como son la revisión de los requisitos, la revisión de la especificación del sistema, la revisión de la arquitectura y del diseño de detalle y las revisiones de código.
- ✓ En las etapas finales también se distingue del modelo original en V porque se desglosan las tareas de pruebas propiamente dichas y las labores de depuración y corrección de los errores detectados, que serán llevadas a cabo por desarrolladores.

Según el gráfico del modelo en W (**Ver Figura 1.6**), las labores de pruebas aparecen en los bloques verdes. Los bloques azules representan el resto de actividades del ciclo de vida del software antes de su puesta en producción. En las primeras etapas se concentran labores de verificación de productos no

software y de preparación para la verificación y validación del software. En las etapas finales se desarrolla todo el esfuerzo de verificación y validación del software producido. (25)

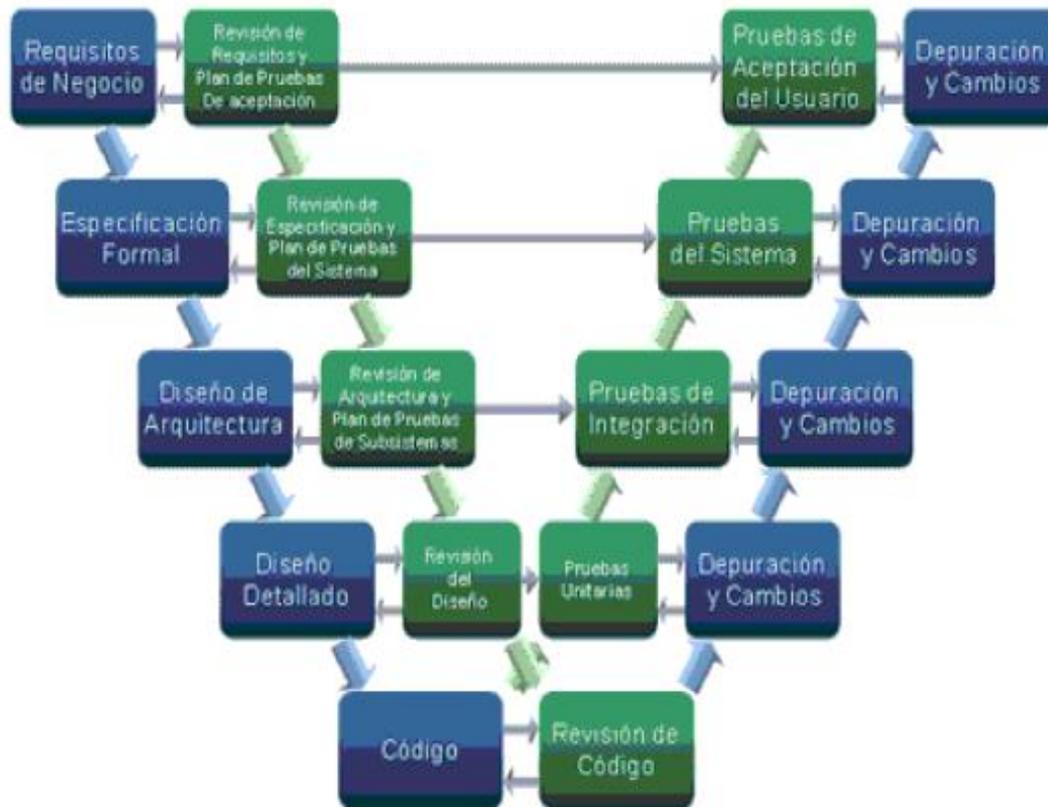


Figura 1.6 Modelo en W

### 1.7 Procedimiento de Prueba

En el mundo las pruebas que se realizan al software para su buen funcionamiento son muy importantes, pues determinan su calidad y la aceptación que se desea obtener de los clientes.

Para optimizar todas las funcionalidades de un programa se realizan un determinado número de pruebas que se distribuyen escalonadamente para permitir un desarrollo constante y a largo plazo del producto.

La esfera de la computación es tan cambiante en la actualidad, y sobre todo de gran evolución tecnológica, se ha hecho necesario desarrollar procedimientos para contribuir a la calidad del software y obtener un mejoramiento continuo de todos los procesos relacionados con el avance de los mismos.

En Cuba existen procedimientos para llevar a cabo las pruebas que se realizan a los sistemas de forma general a todas las fases por las que pasa el mismo, desarrollados en trabajos de Diplomas por estudiantes de la UCI, pero la inexistencia de la aplicación de estos procedimientos propuestos es una realidad en los proyectos productivos de la Facultad Siete, es por esto que se hace necesario realizar una serie de pasos adecuados más específicos para la realización de pruebas en cada fase. Dentro de estas se hará énfasis en las fases de Depuración y Cambios, que son muy importantes en la vida del software, es donde se libera el producto salvo de errores y es aceptado por el cliente.

Al terminar el presente capítulo se arribaron a las siguientes conclusiones:

- ✓ La aplicación de pruebas durante las fases de Depuración y Cambios proporciona una fiabilidad al cliente ya que las mismas contribuyen al desarrollo de la solución de sus necesidades.
- ✓ Aún con la existencia de diversos métodos, que persiguen los mismos objetivos, la selección del más adecuado según las características del software, garantizaría la calidad del mismo.
- ✓ El Modelo de pruebas por el cual se guía el procedimiento de pruebas a definir es el Modelo de Pruebas W.

## **Caracterización y Diagnóstico de la Producción de software en la Facultad Siete.**

### **2.1 Introducción**

En los proyectos de la Facultad Siete se ha seguido mejorando cada día acerca de la estructura que cada uno persigue, así como el resultado de su trabajo, sin embargo aún siguen existiendo problemas, los cuales se lucha para que desaparezcan poco a poco, por esta razón se hace necesario que cada persona que integre el equipo interno de calidad sepa su función y responsabilidad, para que el producto alcance la calidad requerida.

En este capítulo se hace una caracterización de la producción de la Facultad Siete con el objetivo de diagnosticar el estado actual de la misma, además de que se muestran los resultados que se obtienen luego de hacer un estudio de los problemas existentes de los proyectos de la facultad. Se realiza un análisis detallado de esta situación, mediante la aplicación de técnicas, como la entrevista, que permiten recoger información con respecto a la calidad que se quiere obtener en los software antes de que los mismos lleguen al proyecto de Calidad de la facultad.

### **2.2 Caracterización de la producción de software en la Facultad Siete.**

La producción de la Facultad Siete se basa fundamentalmente en lograr la informatización del sector de la salud en el país. También ayuda a países hermanos poniendo en práctica productos de gran importancia en este sector.

En esta facultad existen varias áreas temáticas las cuales están constituidas por proyectos, facilitando así la organización de la producción de software de la misma. En cada área temática existe un grupo interno encargado de llevar a cabo un aseguramiento de la calidad pretendiendo librar de errores simples al producto antes de pasar a ser revisado por el proyecto de Calidad de la facultad.

En la Facultad Siete existen actualmente 23 proyectos vinculados al polo de Informática para la Salud y 7 al Procesamiento de Imágenes y Señales, además del proyecto encargado de asegurar la calidad del software. De forma general la producción se caracteriza por presentar en total 30 proyectos (**Ver Figura1.7**) de los cuales 9 de ellos ya han terminado, 5 están en Investigación y 2 en Conceptualización y

a todos estos proyectos están vinculados un total de 505 estudiantes y 66 profesores más 49 estudiantes que están en los proyectos centrales del MININT y MINFAR para sumar un total de 554 estudiantes.

<b>Polo</b>	<b>Nombre del Proyecto</b>
Informática para la Salud.	<i>ALAS SAFE: Balance y Planificación de Materiales Médico</i>
Informática para la Salud.	<i>ALAS Sistema de Información Estadístico Complementario de Salud</i>
Informática para la Salud.	<i>ALAS Sistema de Gestión de Información en el Proceso de Formación de Recursos Humanos en Salud</i>
Informática para la Salud.	<i>ALAS Sistema de Colaboración Médica</i>
Informática para la Salud.	<i>ALAS Nefrored: Red Nacional de Nefrología</i>
Informática para la Salud.	<i>Control Sanitario Internacional</i>
Informática para la Salud.	<i>Tele Consulta</i>
Informática para la Salud.	<i>ALAS SENDN: Sistema de Evaluación del Neurodesarrollo en Niños</i>
Informática para la Salud.	<i>ALAS APS: Registro de Fallecidos</i>
Informática para la Salud.	<i>ALAS APS: Registro de Partos y Nacimientos</i>
Informática para la Salud.	<i>ALAS HIS Bloque Quirúrgico Oftalmológico</i>
Informática para la Salud.	<i>ALAS HIS</i>
Informática para la Salud	<i>ALAS APS Estadística Descriptiva Registro de Áreas de Salud</i>
Informática para la Salud	<i>ALAS APS Registro de Vacunación</i>
Informática para la Salud	<i>Sistema para la Gestión de la Información</i>

	<i>Estomatológica</i>
Imágenes	<i>SASA Sistema para el registro de hierros y señales. Vers2.0</i>
Imágenes	<i>Batería Neuropsicológica para el Diagnóstico de Dislexia</i>
<b>Terminaron</b>	
Informática para la Salud.	<i>Sistema de Rehabilitación Integral</i>
Informática para la Salud.	<i>ALAS APS: Estadística Descriptiva Registro de Enfermedades de Declaración Obligatoria</i>
Informática para la Salud.	<i>ALAS APS: Registro de Actividades Diarias</i>
Informática para la Salud.	<i>ALAS APS: Registro de Indicadores y Conductas de la Atención Primaria</i>
Imágenes	<i>ALAS PACS</i>
Imágenes	<i>ALAS RIS</i>
Informática para la Salud.	<i>ALAS APS: Registro de Vacunación</i>
Informática para la Salud.	<i>ALAS APS: Registro de Áreas de Salud</i>
Informática para la Salud.	<i>ALAS APS: Registro de Población</i>
<b>Proyectos Investigativos</b>	
Imágenes	<i>Cugnot Suite para la identificación de los vehículos</i>
Imágenes	<i>DETEXT: Detección y Extracción de Texto en Videos</i>
Imágenes	<i>Sistema Integral de Vigilancia</i>
Imágenes	<i>Sistema de Identificación por Perfiles de ADN</i>

Imágenes	<i>Algoritmos de Identificación por Huellas Dactilares</i>
<b>En Conceptualización</b>	
Imágenes	<i>Sistema de Identificación Balística</i>
Informática para la Salud.	<i>ALAS Tools SAAA</i>

**Figura 1.7 Distribución de Proyectos por Polos Productivos.**

### 2.3 Técnicas aplicadas para el diagnóstico.

Existen varias técnicas de recopilación de información dentro de las que se encuentran:

- ✓ Entrevistas
- ✓ Observación
- ✓ Encuestas
- ✓ Cuestionarios
- ✓ Revisión de Documentos
- ✓ Verificación
- ✓ Confirmación
- ✓ Investigación
- ✓ Análisis
- ✓ Evaluación

Las principales técnicas empleadas para poder realizar una valoración y a partir de ella elaborar el procedimiento adecuado para normalizar las fases de Depuración y Cambios fueron:

#### **La Revisión de Documentos:**

Se utiliza para recoger la información que se encuentra registrada en un documento establecido. Se hizo uso de los expedientes de proyectos en las distintas áreas temáticas revisados en auditoría en la Facultad Siete.

#### **La entrevista:**

Entrevista es la acción y efecto de entrevistar o entrevistarse. Se trata de una conversación entre una o varias personas para un fin determinado, generalmente las preguntas son de forma oral y se trata de que sean lo más claras posibles para que las respuestas sean las más precisas.

Hay dos tipos básicos de entrevista: **la declaración** y **la entrevista** propiamente dicha.

La declaración busca obtener la opinión o el punto de vista del entrevistado acerca de hechos diversos. A veces, las declaraciones forman parte de otros textos periodísticos como, por ejemplo, los reportajes.

La entrevista busca profundizar en la dimensión humana del reportaje y constituye un documento extenso sobre la vida, la obra o la forma de pensar del entrevistado. (26)

La entrevista (**Ver Anexo#1**) se realizó a todas las áreas temáticas de la facultad dentro de las cuales se encuentran distintos proyectos productivos (**Ver Figura1.8**) tales como:

#### Área Temática Sistemas Especializados(SE)

ALAS Nefrored Red Nacional de Nefrología.

Sistema para la Gestión de la Información Estomatológica.

Sistema de Rehabilitación Integral.

Control Sanitario Internacional.

Tele Consulta.

ALAS SENDN Sistema de Evaluación del Neurodesarrollo en Niños.

#### Área Temática Sistemas de Atención para la Salud(SAS)

ALAS tools SAAA.

ALAS Sistema de Información Estadística Complementario de Salud.

ALAS Sistema Colaboración Médica.

ALAS SAFE Balance y Planificación de Materiales Médico.

ALAS Sistema de Gestión de Información en el Proceso de Formación de Recursos Humanos en Salud.

### **Área Temática Atención Primaria para la Salud(APS)**

ALAS APS Registro de Partos y Nacimientos.

ALAS APS Registro de Vacunación.

ALAS Registro de Fallecidos.

ALAS APS Registro de Indicadores y Conductas de la Atención Primaria.

ALAS APS Registro de Actividades Diarias.

ALAS APS Registro de Población.

ALAS APS Estadística Descriptiva del Registro de Áreas de Salud.

ALAS APS Estadística Descriptiva Registro de Enfermedades de Declaración Obligatoria

### Área Temática Gestión Hospitalaria

ALAS HIS Bloque Quirúrgico Oftalmológico.

ALAS HIS.

### Área Temática Software Médico Imagenológico

ALAS RIS.

Alas PACS.

### Área Temática Proceso Digital de Imágenes

Cugnot Suite para la identificación de los vehículos.

Sistema Integral de Vigilancia.

SASA Sistema para el registro de hierros y señales. Vers2.0

Sistema de Identificación Balística.
Detext Detección y Extracción de Texto en Videos.
Batería Neuropsicológica para el Diagnóstico de Dislexia.
Algoritmos de Identificación por Huellas Dactilares.
Sistema de Identificación por Perfiles de ADN.

**Figura 1.8 Distribución de Proyectos por Áreas Temáticas.**

## **2.4 Diagnóstico de la producción de software en la Facultad Siete.**

### **2.4.1 Análisis del resultado de las Entrevistas realizadas en cada proyecto a los responsables del grupo interno de calidad y a un miembro del mismo grupo.**

Las pruebas son una parte muy significativa del proyecto de adecuación de las aplicaciones, no sólo por su importancia en el logro de resultados correctos si no por el tiempo y recursos requeridos, por ser esto de gran importancia se decide realizar un análisis de la situación que presenta la producción de la Facultad Siete.

Con el objetivo de realizar un profundo análisis de la situación que existe en las diferentes áreas temáticas que presenta la facultad involucrada con respecto al uso de los procedimientos propuestos para realizarles pruebas al software, además de saber si son capaces de seguir los consejos y cursos que el proyecto de Calidad de la facultad les imparte, se hace uso de las técnicas que anteriormente se relacionaron, estas permitieron obtener una estadística general de las seis áreas temáticas que fueron revisadas. Se tuvo en cuenta el criterio de la persona encargada de dirigir al grupo interno de calidad además de un integrante de dicho grupo.

Se confeccionó y aplicó una entrevista de doce preguntas con el fin de abarcar los principales problemas que afectan a estas personas que son encargadas de llevar el control de la calidad de los productos que son realizados.

Examinando los resultados se llega a la conclusión de que en cada área temática existe el grupo de calidad, el cual está organizado de forma centralizada, respondiendo a las necesidades de todos los proyectos que integran cada área. La mayoría de las personas que se entrevistaron están de acuerdo en que la calidad con que se realizan las aplicaciones en cada proyecto es buena. También cuentan con el conocimiento básico de las fases que propone el Proceso Unificado de Desarrollo de Software (RUP), pero no conocen las demás fases que han sido propuestas por compañeros que han realizado sus tesis años anteriores y que son de suma importancia para una mejor revisión del producto. Estas fases también son importantes para mantener el control de los errores; porque se enmarcan en menos espacio dentro de la vida del software, pues se tiene más especificado por cada fase en las que transcurre el mismo, permitiendo tener un mejor control.

Todos estos grupos cuentan con el conocimiento sobre los tipos de pruebas que se le realizan al producto e incluyendo toda la documentación que a este se le incorporan: pruebas de Caja Negra y en algunos casos se están iniciando el trabajo con respecto a la realización de pruebas de Caja Blanca, pero no tienen un método específico para esto, pues le realizan pruebas en una sola fase o en dos, son muy pocos los que realizan pruebas en todo el ciclo de vida del software, se tiene el concepto que, como existe una fase en la vida del mismo denominada pruebas, es en este momento donde se deben realizar.

Se presenta un desconocimiento casi total de las herramientas que existen para la realización de pruebas al producto, pero si conocen y aplican técnicas básicas para facilitar el trabajo y obtener un éxito total en las pruebas: como son las listas de chequeo.

Lo relacionado anteriormente demuestra que los procedimientos propuestos no han sido estudiados, trayendo como consecuencia que no han podido ser útiles a estas personas que son tan importantes para poder entregar el software al proyecto de Calidad de la facultad con menos errores. No se cumple con lo que está reglamentado, que es realizarles pruebas al software en todas las fases para obtener un mejor producto y no tener que llegar al final del trabajo con errores que pudieron detectarse al inicio del proceso

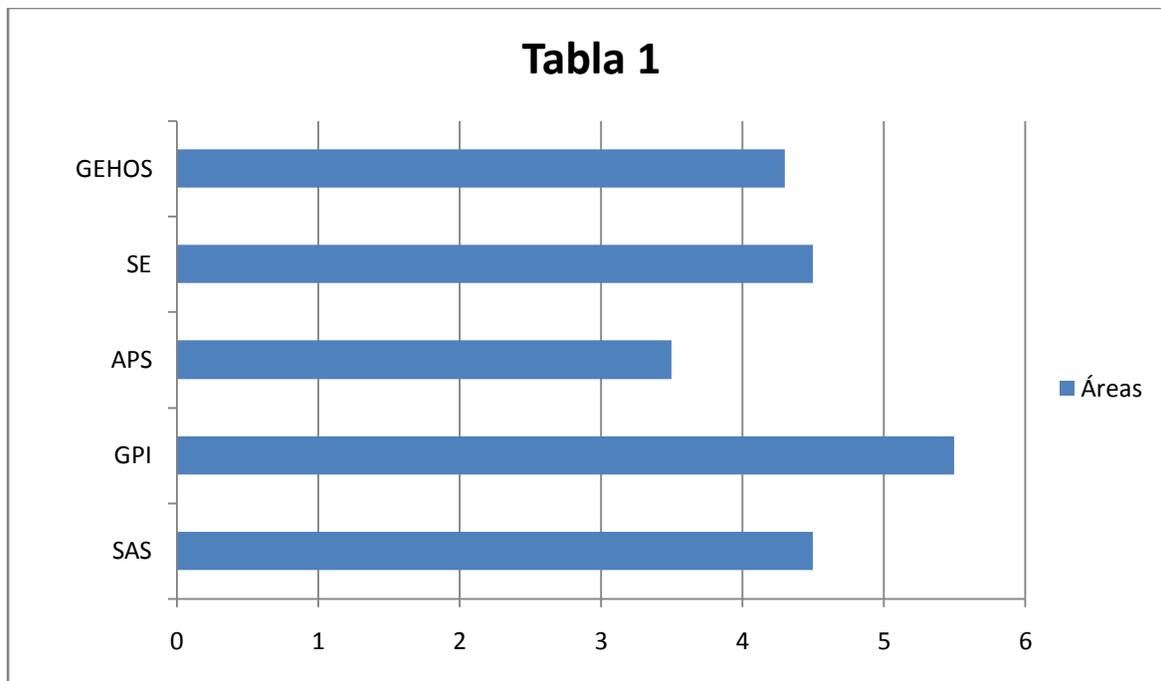
de desarrollo del sistema. También se demuestra que el proyecto de Calidad de la facultad continúa recibiendo aplicaciones con errores que debieron ser corregidos antes de llegar al mismo.

Se demuestra además que no trabajan correctamente porque en algunos casos no se guardan las no conformidades que se obtienen cuando realizan revisiones en el documento indicado que existe para esto, es decir en el registro de las no conformidades, sino que se guardan en otros documento como en un Excel, esto trae como consecuencias, que no exista un plan real del trabajo y que no exista una buena comunicación entre grupo de calidad y grupo de desarrolladores.

### 2.4.2 Relación de preguntas y respuestas de los entrevistados.

A continuación se muestran algunas de las preguntas que formaron parte de la entrevista realizada a las áreas temáticas así como las respuestas dadas por los entrevistados que se muestran en un gráfico determinado.

1. ¿Cómo considera la calidad de las aplicaciones que se realizan en su proyecto?

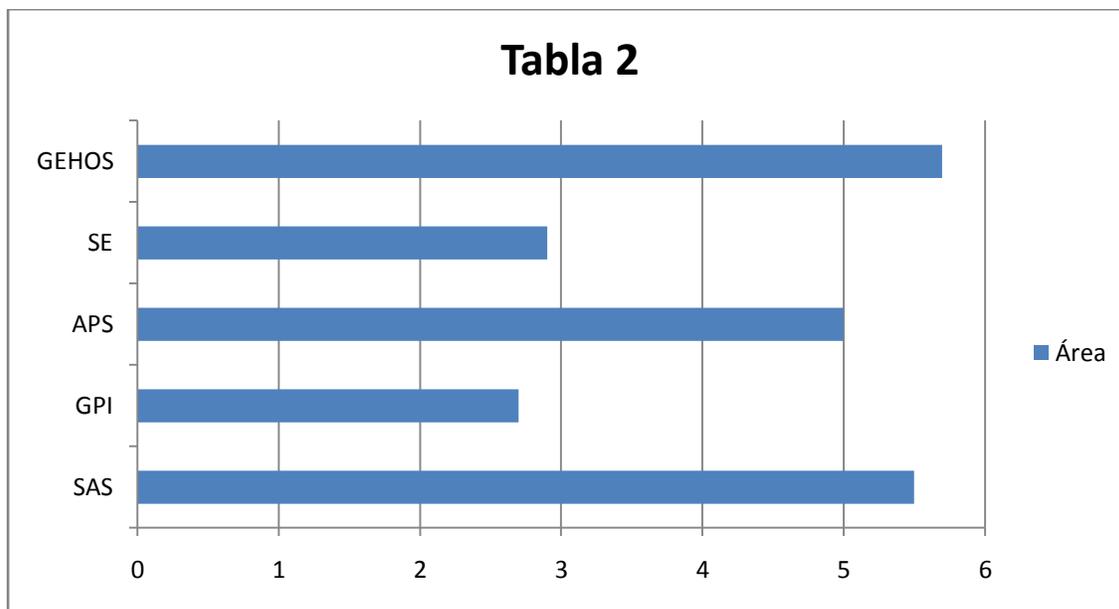


**Leyenda:**

Si el entrevistado respondió que los productos que se hacen en su respectiva área temática tienen una calidad de mal, regular, bien o excelente coincide con una probabilidad en un rango de 0 a 6. Además se tuvo en cuenta la precisión con que el entrevistado respondió.

- Si se respondió una evaluación de mal (M) coincide en un intervalo de 0 a 2,9.
- Si se respondió una evaluación de regular (R) coincide en un intervalo de 3 a 3,9.
- Si se respondió una evaluación de bien (B) coincide en un intervalo de 4 a 4,9.
- Si se respondió una evaluación de excelente (E) coincide en un intervalo de 5 a 5,9.

2. ¿En qué fases se le realizan pruebas al software?

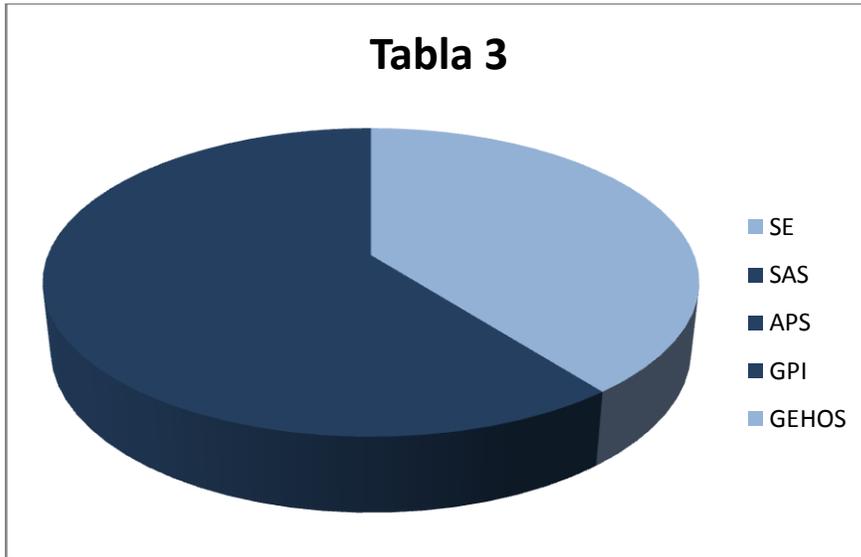


**Leyenda:**

Si el entrevistado respondió alguna fase en específico se le otorga una probabilidad de 0 a 2,9 (es un rango bajo porque las pruebas se deben realizar en todas las fases).

Si el entrevistado respondió que en todas las fases se le otorga una probabilidad de 5 a 6.

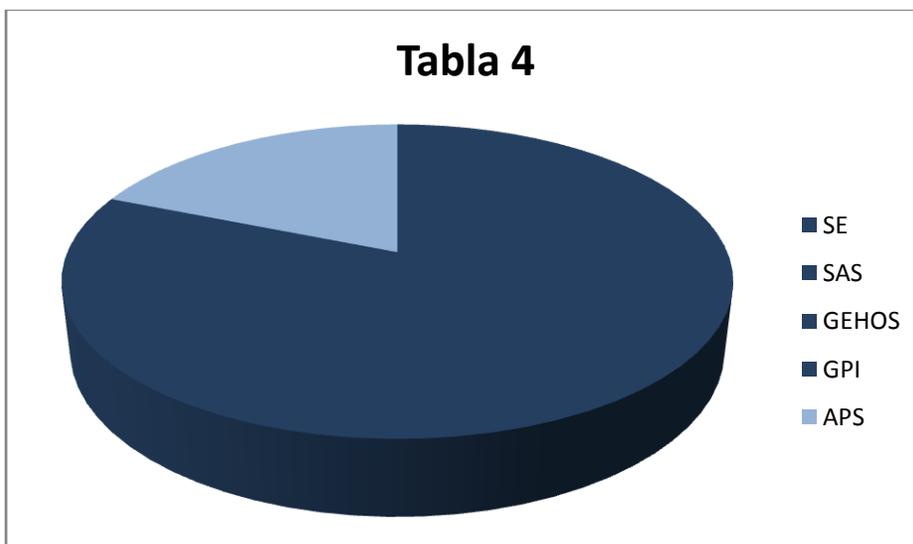
3. ¿En su proyecto se aplica algún tipo de Listas de chequeo?



**Leyenda:**

Si el entrevistado respondió de forma positiva se relaciona con el color azul fuerte.  
 Si el entrevistado respondió de forma negativa se relaciona con el color azul claro.

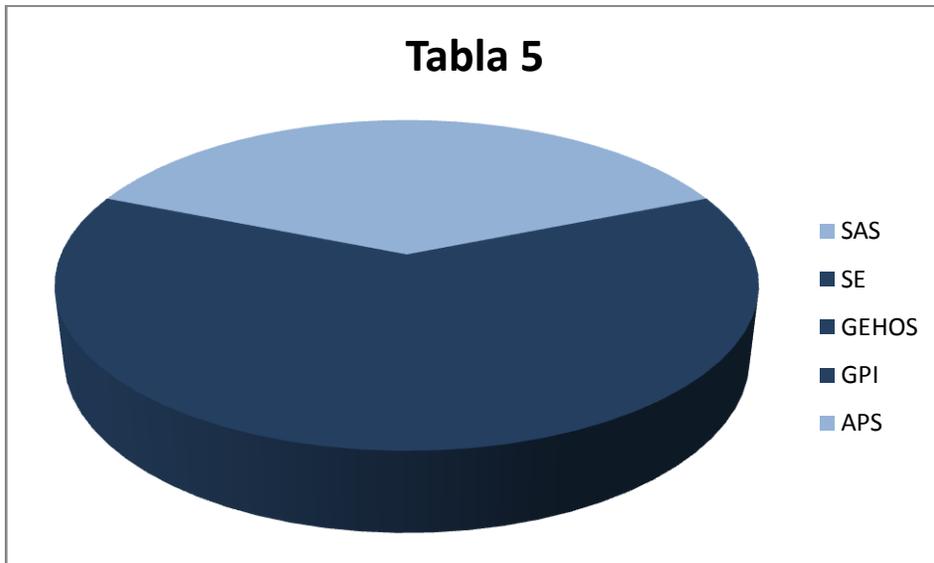
4. ¿Conoce alguna herramienta para la realización de pruebas?



**Leyenda**

Si el entrevistado respondió de forma negativa se relaciona con el color azul fuerte.  
 Si el entrevistado respondió de forma positiva se relaciona con el color azul claro.

5. ¿Sabe usted qué es un plan de pruebas?

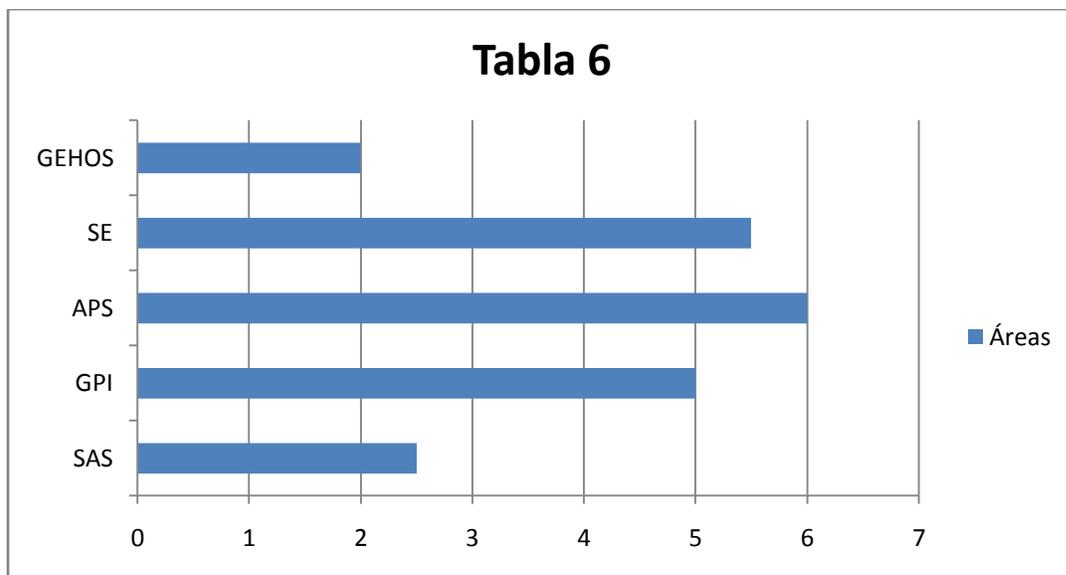


**Leyenda**

Si el entrevistado respondió de forma negativa se relaciona con el color azul fuerte.

Si el entrevistado respondió de forma positiva se relaciona con el color azul claro.

6. ¿Según sus conocimientos, las Pruebas de Aceptación las realiza: el desarrollador, grupo interno de calidad, usuario, proyecto de Calidad de la facultad o cliente final?

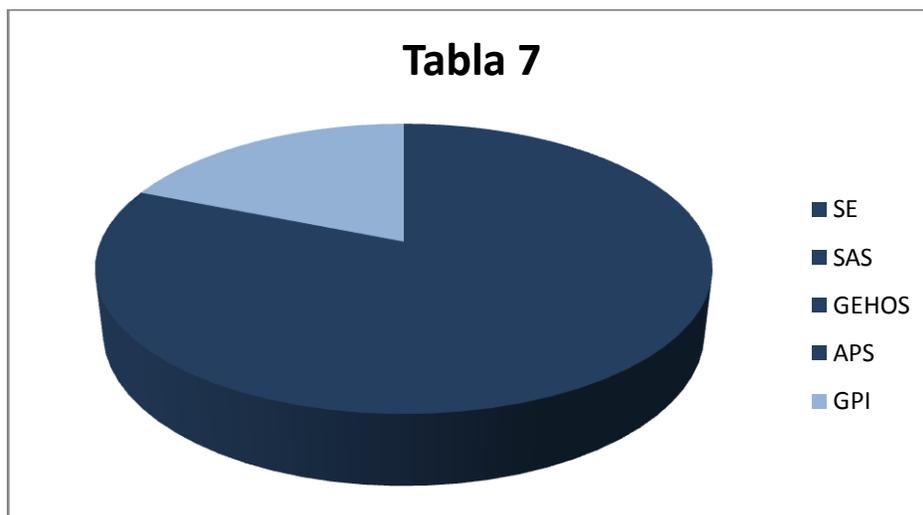


**Leyenda**

Si el entrevistado presentó dudas a la hora de responder se le relaciona una probabilidad de 0 a 2,9.

Si el entrevistado respondió de forma correcta se le relaciona una probabilidad de 5 a 6.

7. ¿En su proyecto se genera algún documento de No conformidades?



**Leyenda**

Si el entrevistado respondió de forma negativa se relaciona con el color azul fuerte.

Si el entrevistado respondió de forma positiva se relaciona con el color azul claro.

**2.5 Análisis del resultado de la revisión de los documentos de las áreas temáticas.**

Con el propósito de hacer una revisión completa de los proyectos por área temática para obtener una información más amplia acerca de la producción de la Facultad Siete, se revisó la documentación de todos los proyectos haciéndose un resumen que a continuación se relacionan, enfatizándose en los principales problemas que más golpean en general a las áreas temáticas.

**Área Temática Atención Primaria a la Salud (APS):**

Principales problemas:

- ✓ Existen algunos proyectos que en el plan de Desarrollo de Software no se especifica el Polo Productivo al que pertenecen, no se referencia la planilla de Roles y responsabilidades y otros se

encuentran de manera parcial pues tiene muchos puntos sin llenar aún, además de que no se cuenta con el Plan de Control del cronograma ni con el Plan de Reportes.

- ✓ Algunos no tienen confeccionado el Plan de Mediciones así como el Plan de Gestión de Requisitos.
- ✓ En otro caso se tiene con el problema de que cuentan con el documento Visión pero parcial, pues este tiene algunos puntos sin llenar aún.
- ✓ No se tiene confeccionada la planilla de Roles y Responsabilidades.
- ✓ No se tienen los resultados de las estimaciones de tiempo y esfuerzos del proyecto.
- ✓ No se cuenta con el Plan de Evaluación.

### **Área Temática Hospitales:**

Principales problemas:

- ✓ No se encuentra información referente al Polo Productivo al que pertenece el proyecto.
- ✓ No se encuentra en el cronograma la fecha de los entregables definidos en el Plan de Desarrollo de Software.
- ✓ No se encontró evidencia de un Plan de Evaluación en el documento Plan de Desarrollo de Software.
- ✓ Faltan documentos como Plan de Capacitación, Plantilla Ambiente de Desarrollo y Plan de Gestión de Configuración.

### **Área Temática Sistema de Apoyo a la Salud (SAS):**

Principales Problemas:

- ✓ No se identifica el Polo Productivo al que pertenecen algunos proyectos.
- ✓ Existen algunos proyectos que no se define la estructura organizativa.
- ✓ En otros casos no se evidencia un Plan de control del cronograma.
- ✓ No existe registro de las estimaciones de esfuerzo y tiempo de desarrollo.
- ✓ En algunos proyectos no se evidencia un Plan de Mediciones en el Plan de Desarrollo de Software.

### **Área Temática Sistemas Especializados:**

**Principales Problemas:**

- ✓ Algunos proyectos no tienen documentado los estudiantes del proyecto por año.
- ✓ Al igual, no se evidencia un Plan de Evaluación.
- ✓ No se identifica el Polo Productivo al que pertenece el proyecto.
- ✓ En la documentación revisada de los proyectos de esta área en algunos no se evidencia la metodología de desarrollo, sólo mencionan algunos elementos de RUP.
- ✓ No se evidencia un Plan de control del cronograma.

**Área Temática de GPI:**

- ✓ En algunos proyectos de esta área temática el cronograma no está totalmente actualizado y en otros no existe.
- ✓ El expediente de proyecto no todos los proyectos lo tienen completo.

Al concluir este capítulo se arribaron a las siguientes conclusiones:

- ✓ La Facultad Siete ha obtenido logros con respecto a la producción como es la implantación del grupo de calidad en cada área temática.
- ✓ Los grupos internos de calidad de las áreas temáticas no siguen minuciosamente los pasos con respecto a la realización de un excelente trabajo.
- ✓ No se conoce por parte de los grupos internos de calidad los procedimientos que ya han sido propuesto para obtener un excelente producto.
- ✓ No se trabaja con herramientas ya existentes, pues no todos tienen el conocimiento de que existen las mismas.

## Propuesta de Procedimientos.

### 3.1 Introducción

En el proceso de elaboración del software más que atención, precisión y comunicación se requiere que el mismo se obtenga con la mayor calidad posible, lo más semejante a sus requisitos y libre de defectos y errores para así satisfacer al cliente más allá de sus expectativas.

El principal objetivo de la realización de pruebas en las tres últimas fases (Depuración y Cambios (Ejecución de las Pruebas de Integración, Sistema, Liberación y/o Aceptación)) del ciclo de vida del software es prevenir posibles fallos que puedan ser detenidos antes de que el producto llegue a manos del cliente y pueden ser aplicadas al producto una vez ya terminado el mismo.

Debido al estudio del procedimiento propuesto por el Ing. Edel Ávila Cruz donde se recomendó la realización de procedimientos específicos en cada fase que propone el “Modelo en W” se hace necesario confeccionar la propuesta que refine el procedimiento antes mencionado haciendo énfasis en las últimas tres fases del ciclo de vida de las Aplicaciones Web que se desarrollan en la producción de la Facultad Siete.

Este modelo se puede adaptar a cualquier metodología de desarrollo de software por lo que es adaptable al Proceso Unificado de Desarrollo de Software (RUP) que es una de las metodologías más usadas dentro de las Aplicaciones Web que son implementadas en la Facultad Siete.

Todas las pruebas que se le realicen al producto deben ser planificadas con antelación para garantizar el desarrollo exitoso de la aplicación, mediante un plan de pruebas que se hace más importante en estas últimas fases del software.

Normalmente un sistema es sometido a Pruebas de Unidad, Integración, Sistema, Aceptación entre otras. Las Pruebas de Unidad se realizan a clases o interfaces individuales o a un grupo de estas, y son típicamente ejecutadas por el programador, este tipo de prueba generalmente cuenta con herramientas para automatizarlas.

Las Pruebas de Integración integran componentes y clases según el orden especificado en el plan de integración. Las Pruebas de Sistema ven al sistema como una "Caja Negra" y validan que el sistema tenga la funcionalidad que el usuario final espera. Las Pruebas de Aceptación conducidas por el cliente verifican que el sistema satisface los requerimientos y son similares a las Pruebas de Sistema, pues se basan fundamentalmente en pruebas de funcionalidad.

Siguiendo como espina dorsal el modelo antes mencionado se propone realizar procedimientos para normalizar las pruebas durante las fases de Depuración y Cambios del ciclo de vida del software. Para la realización de estos procedimientos se ha propuesto realizar un Plan de Pruebas para cada una de las fases con el propósito de:

- ✓ Garantizar el cumplimiento de los requerimientos planteados en el marco de las tres últimas fases del software.
- ✓ Asegurar que se tengan en cuenta todos los Casos de Prueba posibles para validar la solución informática a un requerimiento o solicitud de cambio.
- ✓ Definir la estrategia de pruebas a seguir en el proyecto, para las fases de Depuración y Cambios de desarrollo del mismo.
- ✓ Establecer las técnicas y herramientas de prueba a utilizar en el desarrollo de cada Plan de Pruebas.
- ✓ Garantizar el aseguramiento de la calidad, con el objetivo de identificar y evaluar los defectos que puedan afectar al software, para cumplir con esto se recomienda como guía el trabajo de diploma que se está realizando en el proyecto de Calidad, este se titula "Estrategia para el aseguramiento y control del aseguramiento de la calidad en la Facultad Siete".

**Nombre del procedimiento:**

Procedimiento para la realización de pruebas durante las fases de Depuración y Cambios (Ejecución de las Pruebas de Integración, Sistema, Liberación y/o Aceptación) en Aplicaciones Web.

**Objetivo:**

- ✓ Normalizar las fases de Depuración y Cambios (Ejecución de las Pruebas de Integración, Sistema,

Liberación y/o Aceptación) en el desarrollo de Aplicaciones Web en la Facultad Siete.

- ✓ Garantizar el aseguramiento de la calidad en los proyectos productivos en la Facultad Siete, con el objetivo de identificar los errores de forma temprana en el proceso de software, para esto utilizar como guía el trabajo de diploma que se está realizando en el proyecto Calidad, este se titula “Estrategia para el aseguramiento y control del aseguramiento de la calidad en la Facultad Siete”

**Responsable:**

Jefes de Áreas Temáticas de la Facultad Siete, Líderes de proyecto y personal designado en los grupos de desarrollo, grupo interno de calidad y proyecto Calidad de la Facultad Siete.

**Responsable de quien lo ejecuta:**

Vicedecano de producción de la Facultad Siete y Asesor(a) de calidad de esta facultad.

**Revisa y actualiza este procedimiento:**

Asesor(a) de calidad y jefes de los polos productivos de la Facultad Siete y personal designado para tal actividad.

**Fiscaliza su cumplimiento:**

Asesor(a) de calidad y Vicedecano de producción de conjunto con los líderes de proyecto y jefes de área temática y jefes de los polos productivos, con especialistas en temas de calidad del software.

**Definiciones importantes**

**Pruebas de Aceptación:**

- ✓ Una Prueba de Aceptación tiene como propósito demostrar al cliente el cumplimiento de un requisito del software.
- ✓ Puede estar asociada a requisitos funcionales o no funcionales. (27)
- ✓ El objetivo de las Pruebas de Aceptación es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento. (28)

**Pruebas de Liberación:**

Conjunto de pruebas y revisiones desarrolladas por el grupo de calidad de la facultad y posteriormente por el equipo de calidad central de la universidad para garantizar la satisfacción del cliente.

**Plan de Pruebas:**

El propósito del Plan de Pruebas es explicitar el alcance, enfoque, recursos requeridos, calendario, responsables y manejo de riesgos de un proceso de pruebas (**Ver Anexo#2**).

**Usuario Final:**

El usuario final de un producto informático (bien sea hardware o software), es la persona a la que va destinada dicho producto una vez ha superado las fases de desarrollo correspondientes. (29)

**Disposiciones Generales.**

- ✓ El Procedimiento debe ser analizado en la dirección de producción de la Facultad Siete para así ponerlo en práctica en las distintas áreas temáticas que existen.
- ✓ Se recomienda automatizar este proceso de pruebas para normalizar las fases de Depuración y Cambios.
- ✓ Este documento contiene información propietaria de la Universidad de las Ciencias Informáticas y fue elaborado confidencialmente para un propósito específico.
- ✓ El que recibe el documento asume la custodia y control, comprometiéndose a no reproducir, divulgar, difundir o de cualquier manera hacer de conocimientos público su contenido, excepto para cumplir el propósito para el cual se ha generado.
- ✓ Estas reglas son aplicables para todas las páginas de este documento.

**3.2 Fase Depuración y Cambios; Ejecución de las Pruebas de Integración.**

Las Pruebas de Integración se llevan a cabo durante la construcción del sistema, involucran a un número creciente de módulos y terminan probando el sistema como conjunto.

Estas pruebas se pueden plantear desde un punto de vista estructural o funcional. Las pruebas estructurales de integración son similares a las pruebas de Caja Blanca; pero trabajan a un nivel conceptual superior. En lugar de hacer referencia a sentencias del lenguaje, serán llamadas entre módulos. Se trata pues de identificar todos los posibles esquemas de llamadas y ejercitarlos para lograr una buena cobertura de segmentos o de ramas.

Las Pruebas Funcionales de Integración son similares a las pruebas de Caja Negra. Aquí se tratará de encontrar fallos en la respuesta de un módulo cuando su operación depende de los servicios prestados por otro(s) módulo(s). Según se va acercando al sistema total, estas pruebas se estarán basando más en la especificación de requisitos del usuario.

Las pruebas finales de integración cubren todo el sistema y pretenden cubrir plenamente la especificación de requisitos del usuario. Además, a estas alturas ya suele estar disponible el Manual de Usuario, que también se utiliza para realizar pruebas hasta lograr una cobertura aceptable.

En todas estas Pruebas Funcionales se siguen utilizando las técnicas de partición en clases de equivalencia y análisis de casos límite (fronteras). (30)

### **3.2.1 Propósito:**

- ✓ Garantizar el cumplimiento de los requerimientos planteados en el marco del proyecto en cuanto a la integración de aplicaciones.
- ✓ Asegurar que se tengan en cuenta todos los Casos de Prueba posibles para validar la solución informática a un requerimiento o solicitud de cambio, garantizando que en el momento de entregar el producto a pruebas de usuario éste cuente con un nivel de calidad apropiado.
- ✓ Definir todas las actividades relacionadas con la ejecución de las Pruebas de Integración, las responsabilidades individuales para cada tarea, los recursos y los prerrequisitos que deben ser considerados en el esfuerzo de pruebas.
- ✓ Garantizar el funcionamiento adecuado de las reglas de negocio que involucran la integración de aplicaciones.
- ✓ Garantizar el buen funcionamiento de la infraestructura de integración utilizada.

### 3.2.2 Alcance:

- ✓ Se aplica para el diseño y la elaboración de los planes de Pruebas de Integración que son ejecutadas por los desarrolladores antes de realizar la entrega de una versión estable del producto para que le sean aplicadas las pruebas de usuario. Estas pruebas son realizadas a nivel de capa de negocio en los productos y servicios comprendidos dentro del sistema de calidad. Su diseño y ejecución se apoyan en la herramienta JUnit.
- ✓ Todos los proyectos productivos de la Facultad Siete.

### 3.2.3 Descripción General del Procedimiento

Para la realización de este procedimiento se siguieron una serie de pasos muy importantes para llevar a cabo el mismo, los cuales están definidos por una sucesión cronológica de operaciones que se relacionan entre sí. Además se generan artefactos que son importantes para el correcto completamiento de este procedimiento.

La elaboración de este procedimiento para normalizar las Pruebas de Integración en la fase de Depuración y Cambios se hace necesaria, ya que se logra, que todas las interfaces estén bien conectadas, es una satisfacción tanto para el equipo de desarrollo como para el cliente.

Para realizar las Pruebas de Integración se deben seguir por las siguientes fases (**Ver Figura #1.9**):

- ✓ Fase #1: Análisis de los requerimientos de integración.
- ✓ Fase #2: Diseño de Casos de Prueba.
- ✓ Fase #3: Validar y ejecutar el diseño de los Casos de Prueba.

### 3.2.4 Roles del Procedimiento

- ✓ **Diseñador de Casos de Prueba:** Es el encargado de diseñar cada uno de los Casos de Prueba.
- ✓ **Jefe de Prueba:** Es el responsable de validar los diseños de Casos de Prueba, recibe toda la documentación que se necesita para poder definir las Pruebas de Integración y verifica que se realicen correctamente todas las pruebas.

- ✓ **Probador:** Es el que realiza cada una de las pruebas y ejecuta los Casos de Prueba generando un documento de No Conformidades donde se registran todos los defectos encontrados.

### 3.2.5 Artefactos de entrada

- ✓ **Documento de Requerimientos:** Documento donde se comunica de manera precisa todos los requerimientos del sistema, tanto requerimientos funcionales como no funcionales.
- ✓ **Documento de Reglas de Integración:** En este documento se especifica a nivel técnico la descripción de cada regla de integración. Es útil para tomar de allí el nombre de las clases que representan cada observación, los nombres y tipos de los parámetros que reciben y las acciones que ejecutan. Es importante tener en cuenta este documento ya que siguiendo sus lineamientos se pueden disminuir los errores en los nombres y tipos de los parámetros que frecuentemente se introducen en los Casos de Prueba de integración.
- ✓ **Documento de Configuración del Ambiente de Pruebas:** En este documento se describe detalladamente las configuraciones que deben efectuarse previamente para poder diseñar y ejecutar adecuadamente las pruebas unitarias. Si se van a realizar pruebas unitarias que involucren la interacción con otros sistemas presentes en la organización, es fundamental describir en el documento de Configuración del Ambiente de Pruebas, cómo se comunican los sistemas involucrados, la información de entrada y salida que se maneja, y las configuraciones previas que deben realizarse en cada uno de los sistemas para la ejecución de las pruebas unitarias.

### 3.2.6 Artefactos de salida

- ✓ **Diseño de Casos de Prueba:** Documento donde se especifican todas las secciones a probar con cada uno de sus escenarios, donde se diseña el caso de pruebas tanto para datos válidos como no válidos.
- ✓ **Documento Plan de Pruebas de Integración:** Este Plan de Pruebas tiene como finalidad dictar los pasos a seguir para realizar un conjunto de Pruebas de Integración. Este plan se basa en su totalidad en pruebas del tipo Caja Negra.
- ✓ **Documento de No Conformidades:** Documento donde se registran todos los defectos y errores encontrados en la realización de pruebas.

- ✓ **Documento Informe Parcial de Resultados:** Informe donde se recogen todos los resultados obtenidos durante la realización de pruebas en cada una de las fases y cada vez que se concluye con cada uno de los procedimientos que se proponen en esta investigación antes de llegar a realizar pruebas de aceptación se elabora dicho informe. **(Ver Anexo #6)**

### 3.2.7 Fases del Procedimiento:

Primeramente antes de comenzar a desarrollar todo el procedimiento se recomienda que en los grupos internos de calidad de cada Área Temática se lleven a cabo tipos de Pruebas de Integración a lo largo del desarrollo del software para que una vez ya pasado al proyecto de Calidad de la Facultad Siete el mismo llegue libre de algunos errores que se pueden detectar en esferas menores.

Dentro de las pruebas a realizar en el entorno de la integración, con una descripción breve y algunos ejemplos se encuentran:

#### **Prueba Funcional**

**Descripción:** Se asegura de que el elemento funcione como se ha especificado. Esto también puede incluir aspectos de conectividad del servidor de aplicaciones y de base de datos que no tienen relación con la seguridad.

**Ejemplo:** Comprobar que la lista desplegable se rellena correctamente y que el valor seleccionado persiste.

#### **Prueba Facilidad de uso y acceso**

**Descripción:** Se asegura de que el usuario pueda ver y usar el elemento de la manera prevista. También puede incluir pruebas en exploradores de terceros. Se asegura de que las personas con problemas visuales, auditivos o de movilidad puedan usar las características.

**Ejemplo:** Asegurarse de que el control aparezca correctamente en circunstancias de varios tamaños de página y que los usuarios puedan usar el control en exploradores de terceros (si es aplicable).

#### **Prueba de Seguridad**

**Descripción:** Se asegura de que se pueda obtener acceso al elemento de la manera prevista y que el usuario pueda realizar todas las tareas según se han diseñado y únicamente esas tareas. Deben realizarse pruebas de las capacidades de inicio de sesión único y los recortes de interfaz de usuario. Deben realizarse pruebas de intervención del usuario, pruebas de autenticación y autorización, y pruebas contra problemas de seguridad, como scripting entre sitios, desbordamientos de búfer y ataques de denegación de servicio.

**Ejemplo:** Asegurarse de que los usuarios puedan iniciar sesión en la página, usar el control para ver las propiedades de archivo desde un almacén específico y que no puedan consultar las propiedades de archivo desde almacenes a los que no están autorizados a tener acceso.

#### **Prueba de Facilidad de administración**

**Descripción:** Se asegura de que el elemento o la página envíen los eventos correctos o escriba las entradas correctas en los registros especificados.

**Ejemplo:** Asegurarse que el control genere los eventos correctos durante una prueba de inserción de errores.

#### **3.2.7.1 Fase #1: Análisis de los requerimientos de integración**

Dentro de las actividades a realizar en esta fase se encuentran:

- ✓ Recibir documentos de apoyo.
- ✓ Verificar que las Reglas del Negocio funcionen acorde a los requerimientos planteados.

Para **analizar los requerimientos del proyecto** se deben apoyar en todos los documentos que pueden ayudar a definir las Pruebas de Integración los cuales son recibidos por el Jefe Prueba del Proyecto de Calidad de la Facultad Siete, que se realizarán sobre la infraestructura de integración para verificar que cada una de las reglas de negocio funcione acorde a los requerimientos planteados. De esta manera se garantiza la completitud de las pruebas definidas y un proceso de calidad que realmente apoye la detección temprana de errores dentro del grupo de trabajo y promueva el desarrollo de software de alta calidad.

Los documentos por los cuales se deben apoyar son:

- ✓ **Documento de Requerimientos.**
- ✓ **Documento de reglas de integración.**

### 3.2.7.2 Fase #2: Diseño de Casos de Prueba

Para comenzar a realizar las actividades en esta fase el equipo de desarrollo debe haber entregado el documento de Configuración de Ambiente de Pruebas bien estructurado donde describa detalladamente las configuraciones que deben efectuarse previamente para poder diseñar y ejecutar adecuadamente los Casos de Prueba.

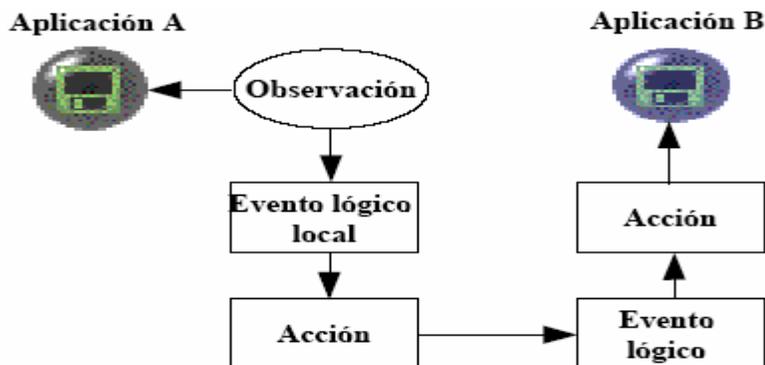
Dentro de las actividades a realizar en esta fase se encuentran:

- ✓ Planificar talleres y cursos optativos para capacitar al personal encargado de realizar los diseños de Casos de Prueba.
- ✓ Diseñar los Casos de Prueba.
- ✓ Realizar Pruebas de Observación a la aplicación.
- ✓ Realizar Pruebas de Verificación a la aplicación.

Para realizar el **diseño y ejecución de Pruebas de Integración**, se utilizará la herramienta JUnit. Esta es un framework de pruebas de regresión. Con esta herramienta es posible **diseñar y ejecutar diferentes Casos de Prueba** de una manera sencilla y rápida. Estos Casos de Prueba pueden ser ejecutados repetidamente cuando sea necesario, optimizando el tiempo requerido para efectuar pruebas de regresión. Esta es la actividad que ayuda a asegurar que los cambios (debido a pruebas u otros motivos) no introduzcan un comportamiento no deseado o errores adicionales, por ello prueban nuevamente componentes evaluados anteriormente, asegurando que los cambios introducidos en el software no han afectado su funcionamiento.

Utilizando JUnit se deben realizar las siguientes pruebas sobre cada una de las reglas de negocio de integración, teniendo en cuenta en cada uno de ellos las condiciones de ejecución válida e inválida de acuerdo a las Reglas del Negocio establecidas:

- ✓ Pruebas sobre la observación que se realiza sobre la aplicación, para verificar que ésta se esté realizando adecuadamente y se genere el evento lógico esperado.
- ✓ Pruebas para verificar la correcta ejecución de la acción que se genera como consecuencia de la observación. Debe verificarse que se genere adecuadamente el evento lógico que será enviado por la infraestructura a las aplicaciones interesadas.
- ✓ Pruebas para verificar la correcta ejecución de la acción que se genera como consecuencia del evento lógico recibido.



**Diagrama conceptual de la ejecución de una regla de integración.**

En todos los casos se recomienda tener en cuenta Casos de Prueba exitosos, que consideren situaciones de excepción y datos inválidos elaborados por el Diseñador de Casos de Prueba.

Para facilitar la ejecución de las Pruebas de Integración es necesario elaborar los scripts sql para configurar la base de datos con la información que se requiere para ejecutar las pruebas.

**3.2.7.3 Fase #3: Validar y ejecutar el diseño de los Casos de Prueba**

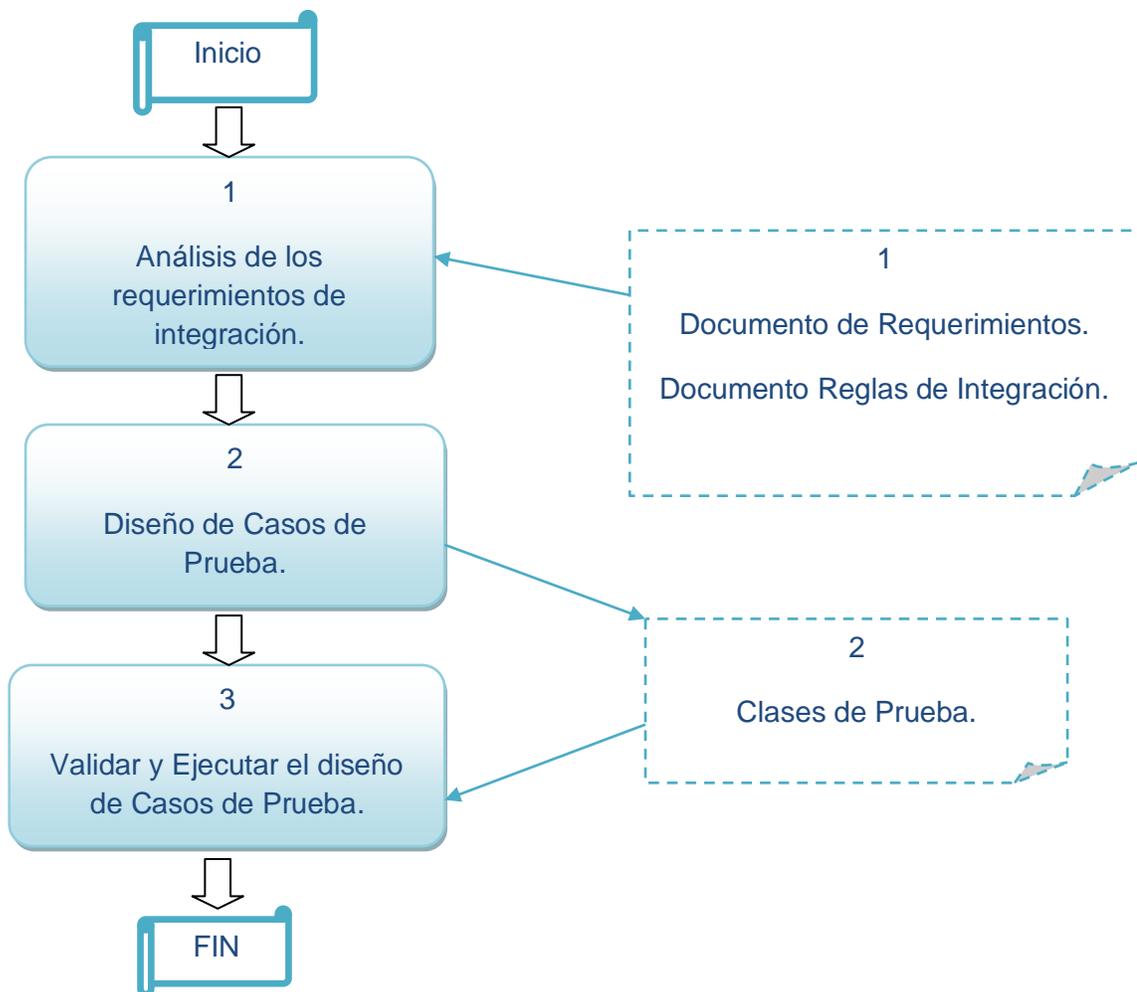
Dentro de las actividades a realizar en esta fase se encuentran:

- ✓ Validar el diseño de Casos de Prueba.
- ✓ Ejecutar el diseño de Casos de Prueba.

Para **validar e inspeccionar los Casos de Prueba definidos**, se verifica que se hayan contemplado todas las pruebas necesarias, que no existen pruebas duplicadas o implícitas en otras y que el tiempo estimado no sobrepasa la fecha límite de ejecución.

Se recomienda que la inspección de los Casos de Prueba la efectúe una persona diferente a quien diseñó los Casos de Prueba inicialmente en este caso el Jefe de Prueba.

En la integración se revisa además el redireccionamiento de la aplicación, así como la funcionalidad de los botones para retroceder y adelantar en las páginas. Puesto que una vez integrados los componentes, las direcciones cambian, así como su camino de acceso. Todos los defectos detectados durante la integración se registrarán en una planilla de no conformidades la cual es confeccionada por el probador que se anexará al expediente del proyecto y que tendrán que ser analizadas por el equipo de desarrollo, para valorar y procesar los defectos detectados en el proceso de integración.



### **Figura 1.9 Diagrama de Actividades del Procedimiento.**

Al concluir este proceso de integración con todo su correcto funcionamiento, la aplicación se encuentra en condiciones de pasar a la etapa de pruebas del sistema.

#### **3.3 Fase Depuración y Cambios; Ejecución de las Pruebas de Sistema.**

Dentro de las Pruebas de Sistema a realizar en este procedimiento se encuentran las pruebas de carga, volumen, seguridad y estabilidad como ya se han mencionado en el capítulo #1 de esta investigación. Las Pruebas de Sistema son muy importantes ya que mediante las mismas se comprueba que el sistema ofrezca todas sus funcionalidades recogidas en su especificación.

##### **3.3.1 Propósito:**

- ✓ Garantizar el cumplimiento de los requerimientos planteados en el marco del proyecto en cuanto a los tiempos de respuesta que debe mantener el sistema para cada caso de uso cuando se experimentan peticiones de un número N de usuarios y M transacciones por unidad de tiempo sobre dicho caso de uso.
- ✓ Definir todas las actividades relacionadas con la ejecución de las pruebas de seguridad y las responsabilidades individuales para cada tarea.
- ✓ Definir todas las actividades relacionadas con la ejecución de las pruebas de carga y las responsabilidades individuales para cada tarea.
- ✓ Definir todas las actividades relacionadas con la ejecución de las pruebas de volumen y estabilidad y las responsabilidades individuales para cada tarea.
- ✓ Garantizar el éxito de todas las Pruebas de Sistema realizadas.

##### **3.3.2 Alcance:**

- ✓ Una aplicación completa integrada.
- ✓ Focalizada en características que están presentes sólo al nivel de todo el sistema.
- ✓ Todos los proyectos de la Facultad Siete.

### 3.3.3 Descripción General de Procedimiento

La elaboración de este procedimiento para normalizar las Pruebas de Sistema en la fase de Depuración y Cambios se hace necesaria ya que lograr que todas las funcionalidades del sistema funciones correctamente es una gran meta y una gran satisfacción para el cliente.

Para la realización del mismo se siguieron una serie de pasos que dieron lugar a la generación de artefactos importantes donde se recogen mayor parte de los defectos encontrados, además de que se definieron las responsabilidades de cada integrante en la realización de dichas pruebas.

Este procedimiento consta de tres fases dentro de las que se encuentran:

- ✓ Fase #1: Inicio de las Pruebas de Sistema.
- ✓ Fase #2: Ejecución de las Pruebas de Sistema.
- ✓ Fase #3: Conclusión de las Pruebas de Sistema.

### 3.3.4 Roles del Procedimiento

- ✓ **Jefe de Pruebas:** Es el encargado de impartir todos los cursos y talleres junto con un equipo que defina de dos o tres probadores además de ser el responsable de cada una de las pruebas que se realicen.
- ✓ **Líder de Proyecto:** Es el jefe del equipo de desarrolladores el cual recibe toda la información por parte del equipo de pruebas además de entregar todo lo referente al expediente de proyecto de la aplicación junto con el Documento Nivel de Acceso.
- ✓ **Probador:** Encargado de ejecutar todas las pruebas y probar los Casos de Prueba diseñados.
- ✓ **Desarrollador:** Forma parte del equipo de desarrollo encargado de desarrollar la aplicación y de erradicar todos los errores descritos en el Documento de No Conformidades.
- ✓ **Diseñador de Casos de Prueba:** Es responsable de diseñar e implementar cada uno de los casos de prueba.

### 3.3.5 Artefactos de Entrada

- ✓ **Documento Nivel de Acceso:** Este documento contiene información acerca de la cantidad máxima de usuarios que soporta la aplicación, además de todos los usuarios con sus contraseñas para poder acceder a los distintos niveles de la misma.
- ✓ **Documento Especificación de Casos de Uso del Sistema:** En este documento se detallan bien todas las actividades y pasos a seguir en cada uno de los casos de usos, es decir se realiza una descripción textual de cada uno de los casos de uso del sistema, además de conocer sus pre condiciones, pos condiciones y sus actores.
- ✓ **Documento de Requerimientos:** Documento donde se comunica de manera precisa todos los requerimientos del sistema, tanto requerimientos funcionales como no funcionales.

### 3.3.6 Artefactos de Salida

- ✓ **Documento de No Conformidades:** Este documento se genera con el propósito de especificar todos los defectos encontrados en la realización de cada una de las Pruebas de Sistema, para así guiar al equipo de desarrollo a erradicar los errores encontrados.
- ✓ **Documento Diseño de Casos de Prueba:** Documento donde se especifican todas las secciones a probar con cada uno de sus escenarios, donde se diseña el caso de pruebas tanto para datos válidos como no válidos.
- ✓ **Documento Plan de Pruebas del Sistema:** Este Plan de Pruebas tiene como finalidad dictar los pasos a seguir para realizar un conjunto de Pruebas de Sistema.
- ✓ **Documento Informe Parcial de Resultados:** Informe donde se recogen todos los resultados obtenidos durante la realización de pruebas en cada una de las fases y cada vez que se concluye con cada uno de los procedimientos que se proponen en esta investigación antes de llegar a realizar pruebas de aceptación se elabora dicho informe.

### 3.3.7 Fases del Procedimiento:

#### 3.3.7.1 Fase #1: Inicio de las Pruebas de Sistema.

Dentro de esta fase se realizan actividades como:

- ✓ Planificar Talleres para capacitar al personal encargado de realizar las Pruebas de Sistema.

- ✓ Entregar toda la documentación pertinente para la realización de las Pruebas de Sistema.

Antes de comenzar a realizar todas estas pruebas a cualquier aplicación el personal debe estar bien capacitado para obtener un resultado satisfactorio de las mismas. Para esto se decide previamente planificar una serie de encuentros con los encargados de llevar a cabo estas acciones donde se realizarán talleres y cursos optativos para fortalecer el conocimiento de cómo realizar cada una de estas pruebas.

Para el comienzo de todas estas pruebas primeramente el grupo de desarrollo debe entregar un documento denominado (**Documento Nivel de Acceso**) donde muestre la cantidad máxima de usuarios para la que está diseñada la aplicación, es decir especificar el alcance en cuanto a cantidad de usuarios del sistema y además mostrar la tendencia real de conexiones simultáneas. También debe existir en este documento un juego de datos para permitirles a los ingenieros de pruebas moverse por toda la aplicación sin problemas, además de facilitar todos los usuarios y contraseñas de los mismos para todos los niveles de acceso del sistema.

### 3.3.7.2 Fase #2: Ejecución de las Pruebas de Sistema.

En cuanto a las **pruebas de Seguridad** se evalúa que el sistema se protege contra ataques, internos o externos, no autorizados, esta prueba requiere sofisticadas técnicas y herramientas, además de que el personal que la va a realizar este bien preparado ya que la aplicación puede recibir ataques en todos los sentidos. Este personal debe especializarse en este tipo de pruebas y explotar todas las vías que existen para atacar el sistema y apoderarse de la información para así poder fortalecer la seguridad de los sistemas.

Además de preparar al personal para estas pruebas se debe prever que existan las condiciones para la realización de las mismas, es decir, condicionar al laboratorio de calidad con todas las herramientas y personas capacitadas para el uso de dichas herramientas que simularán las pruebas de seguridad, dichas herramientas se mencionan en el capítulo #1 de esta investigación.

Con estas pruebas de seguridad se garantiza encontrar todos los bugs o llamadas vulnerabilidades que el sistema pueda presentar y que el fabricante no tiene conocimiento o no conoce de la solución de los mismos.

Para lograr el descubrimiento de estos fallos se suelen emplear, entre otras, unas herramientas que se dedican a lanzar peticiones mal formadas de forma automática. A estas se les llama fuzzers, y pueden estar enfocadas tanto a protocolos de red, como a formatos de archivos y sistemas de ficheros. Son de gran utilidad para auditores informáticos y desarrolladores de software, entre otros, para evitar intrusiones indeseadas unos y para obtener un producto seguro los otros.

Para el funcionamiento de dichos fuzzers se proponen seguir las siguientes etapas:

- ✓ **Obtención de datos:** Dependiendo del tipo de fuzzing deseado y según la implementación de la herramienta, se obtendrán los datos a enviar de una lista estática almacenada en archivos o en el propio código fuente, o se generan en el mismo momento según las configuraciones efectuadas. Este proceso se puede realizar únicamente al inicio de la sesión o justo antes de cada envío.
- ✓ **Envío de datos:** Una vez que se dispone de la información que se desea enviar a la aplicación objetivo, se realizará el proceso, dependiendo, por ejemplo, de si se hace a través de una red informática o de si se quiere realizar únicamente un chequeo local.
- ✓ **Análisis:** Después de realizado el envío, solo quedará esperar los resultados del fuzzing. Si no se espera ninguna respuesta por parte del objetivo, se deberá estar alerta por si se produce un comportamiento inesperado. Si en cambio, se recibe una respuesta, entonces en este momento se comprobará si ésta indica un comportamiento normal o si, por el contrario, el ataque ha tenido éxito y la aplicación ha quedado inestable.

Dependiendo de los datos que se envíen al objetivo se puede hablar de diferentes técnicas de fuzzing. Se suelen englobar en dos grupos, la mutación y la generación de datos.

#### ✓ **Mutación**

Es un método bastante rápido y efectivo, y consiste en partir de una entrada válida y realizar ciertas mutaciones de esos datos, con la intención de que sigan siendo válidas para la aplicación pero lo suficientemente inesperadas para lograr el objetivo.

#### ✓ **Generación**

Se trata de un proceso más lento que el anterior, porque se deben crear los datos antes de enviarlos, pero puede descubrir fallos que el otro método obviaba.

Normalmente, los datos obtenidos también se sustituirán en una entrada válida del objetivo, ya que si se envía simplemente un dato aleatorio, se corre el riesgo de que el programa lo deseche sin ni siquiera procesarlo.

Otras de las pruebas a realizar en esta fase son las **pruebas de carga** y son el tipo más sencillo de pruebas de rendimiento. Una prueba de carga se realiza generalmente para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperada. Esta carga puede ser el número esperado de usuarios concurrentes utilizando la aplicación y que realizan un número específico de transacciones durante el tiempo que dura la carga. Esta prueba puede mostrar los tiempos de respuesta de todas las transacciones importantes de la aplicación. En fin con estas pruebas se busca identificar las condiciones de carga pico en las cuales el programa fallará en manejar las cargas de procesamiento requeridas dentro del período de tiempo requerido.

La tecnología de las pruebas de rendimiento utiliza una o más computadoras (PC) o servidores para actuar como peticionarios. Cada uno emula la presencia de un número de usuarios y cada uno ejecuta una secuencia automática de las interacciones (registrada como una secuencia de comandos, o como una serie de scripts para simular los distintos tipos de uso por parte de los usuarios) con la aplicación cuyo rendimiento se pone a prueba. Por lo general, una PC actúa como gestor de prueba, coordinando, recopilando las métricas de cada uno de los ejecutores y acumulando los datos de rendimiento para la realización de los informes.

La secuencia habitual es incrementar la carga, comenzando con un pequeño número de usuarios virtuales y aumentando el número durante un período hasta alcanzar el máximo. El resultado de la prueba muestra la forma en que el rendimiento varía con la carga, mostrando como el número de usuarios modifica el tiempo de respuesta.

Para la realización de estas pruebas de carga se pueden seguir una serie de pasos importantes dentro de los que se encuentran:

- ✓ **Analizar y seleccionar los casos de uso representativos del proyecto.**

Es fundamental tener en cuenta que no todos los casos de uso que forman parte de un proyecto de software, son susceptibles para aplicarles pruebas de carga, debido a que la cantidad de usuarios

concurrentes que en producción harán peticiones sobre tal caso de uso es relativamente pequeña. De igual manera la cantidad de transacciones por unidad de tiempo no es lo suficientemente significativa para realizar pruebas de carga sobre el caso de uso.

Para seleccionar adecuadamente los casos de uso a los cuáles deben aplicarse pruebas de carga, debe apoyarse en la información consignada previamente en el Formato de Análisis de Carga por Caso de Uso (**Ver Anexo #4**). A partir de esta información es posible determinar cuáles son los casos de uso más representativos en cuanto a carga operacional, para que les sean aplicadas las pruebas de carga pertinentes.

✓ **Diseñar los planes de prueba en JMeter.**

Una vez seleccionados los casos de uso sobre los cuáles se van a aplicar las pruebas de carga, se procede a realizar el diseño de los planes de pruebas en la herramienta JMeter.

JMeter es una herramienta de software libre que ofrece la posibilidad de realizar pruebas de carga sobre diferentes aspectos de una aplicación desarrollada en Java. JMeter es fácilmente configurable y permite conocer los tiempos de respuesta experimentados por una aplicación cuando se tiene un número N de usuarios y el número real de transacciones procesadas por unidad de tiempo. Los resultados pueden ser visualizados en diferentes formatos lo cual facilita las labores de análisis para el probador.

Una ventaja de JMeter es que permite realizar pruebas de carga sobre cada una de las capas que conforman la aplicación a probar. De esta manera, en un momento dado, es sencillo localizar el foco que está generando contención y está afectando los tiempos de respuesta de un caso de uso específico. Esta herramienta debe estar instalada en la máquina que controle las peticiones como amo y como esclavo en las máquinas que sean receptoras de dichas peticiones para así simular la carga de conexiones que debe soportar la aplicación en tiempo real.

✓ **Validar y efectuar la aprobación del diseño de los Casos de Prueba.**

En este paso se pretende validar e inspeccionar los planes de prueba definidos, verificando que se hayan contemplado todas las pruebas necesarias, que los datos de prueba están completos y la información es correcta.

Con esta carga de conexiones se procede a ejecutar las **pruebas de estabilidad**, que se realizarán observando el comportamiento de los servicios con la carga de conexiones por un tiempo determinado, al final el tiempo de respuesta no debe variar, y cada servicio debe mantenerse sin soportar cambios. Esta prueba se hace para determinar si la aplicación puede aguantar una carga esperada extendida. Generalmente esta prueba se realiza para determinar si hay algún escape de memoria en la aplicación.

La **prueba de volumen** sujeta al elemento de prueba a grandes cantidades de datos, para determinar si se alcanzan los límites que hacen fallar al software. También identifica la carga máxima continua o volumen que el elemento de prueba puede manejar por un período dado.

Para la ejecución de estas pruebas primeramente se planifica mediante el tipo de servicio que brinde la aplicación, donde se planificaría una demanda que llene la aplicación en cuanto a datos. Por ejemplo se pediría desde varias conexiones la generación de un reporte determinado varias veces y al mismo tiempo la búsqueda de algunos datos específicos para que sean mostrados en la pantalla, de forma tal que la aplicación tenga que usar un gran volumen de datos para satisfacer la petición del cliente y así identificar su carga máxima.

### **3.3.7.3 Fase #3: Conclusión y Validación de las Pruebas de Sistema.**

En esta última fase se realizan actividades como:

- ✓ Generar el Documento de No Conformidades.
- ✓ Generar el Documento Diseño de Casos de Prueba.
- ✓ Validar las pruebas realizadas.

La buena planificación de todas estas pruebas depende de los datos que se entreguen en el expediente de proyecto al comienzo de las mismas, por esta razón se deben entregar todo lo relacionado con el producto bien reglamentado y estructurado. Además, se debe generar un documento de No Conformidades en el cual se registren todos los errores encontrados en la ejecución de las pruebas antes mencionadas, así como la generación de los diseños de Casos de Prueba con sus clases válidas e inválidas.

Al concluir esta etapa de pruebas, las no conformidades serán entregadas al grupo de desarrollo para su depuración y cambios, es decir este grupo será el encargado de erradicar las no conformidades encontradas en una primera iteración, para luego volver a ser procesados los mismos diseños de Casos de Prueba para verificar su correcta implementación o detección de nuevos fallos. Este ciclo será repetido mientras que el proyecto de Calidad esté detectando no conformidades.

Además de las pruebas que se realicen, deben ser validadas para comprobar si fueron exitosas o no, para garantizar el producto libre de errores en estas fases. El Jefe de Prueba es el encargado de validar si fueron triunfantes o no las pruebas que se llevaron a cabo.

Después de realizadas todas estas actividades la aplicación está en condiciones de pasar a la fase de Depuración y Cambios (Ejecución de las Pruebas de Aceptación), para así ser liberada por el proyecto de Calidad de la Facultad Siete.

### **3.4 Fase Depuración y Cambios; Ejecución de las Pruebas de Aceptación y/o Liberación. Pruebas de Aceptación.**

Estas pruebas las realiza el cliente. Son básicamente Pruebas Funcionales sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario. Estas pruebas no se realizan durante el desarrollo, pues sería impresentable al cliente; sino que se realizan sobre el producto terminado e integrado o pudiera ser una versión del producto o una iteración funcional pactada previamente con el cliente.

La experiencia muestra que aún después del más cuidadoso proceso de pruebas por parte del desarrollador, quedan una serie de errores que sólo aparecen cuando el cliente comienza a usarlo.

Como política a seguir, el cliente siempre tiene razón. Exponer que los requisitos no estaban claros, puede aclarar aparentemente la situación, pero ciertamente no deja satisfecho al cliente.

Una Prueba de Aceptación puede ir desde un informal Caso de Prueba hasta la ejecución sistemática de una serie de pruebas bien planificadas. Las Pruebas de Aceptación pueden tener lugar a lo largo de semanas o meses, descubriendo así errores latentes que pueden ir degradando el funcionamiento del

sistema. Estas pruebas son importantes, pues definen el paso de nuevas fases del proyecto como el despliegue y mantenimiento.

Las pruebas anteriormente mencionadas como en el capítulo #1 se dividen en dos grupos:

**Prueba Alfa:** Se lleva a cabo, por un cliente, en el lugar de desarrollo. Se usa el software de forma natural con el desarrollador como observador del usuario. Las Pruebas Alfa se llevan a cabo en un entorno controlado. Para que tengan validez, se debe primero crear un ambiente con las mismas condiciones que se encontrarán en las instalaciones del cliente. Una vez logrado esto, se procede a realizar las pruebas y a documentar los resultados.

**Prueba Beta:** Se lleva a cabo por los usuarios finales del software en los lugares de trabajo de los clientes. A diferencia de la Prueba Alfa, el desarrollador no está presente. Así, la Prueba Beta es una aplicación "en vivo" del software en un entorno que no puede ser controlado por el desarrollador. El cliente registra todos los problemas (reales o imaginarios) que encuentra durante esta prueba, e informa a intervalos regulares al desarrollador.

Como resultado de los problemas informados durante la prueba anterior, el desarrollador del software lleva a cabo modificaciones y así prepara una versión del producto de software para toda clase de clientes.

Esta es la última fase del ciclo de vida de una aplicación web y comienza cuando las distintas áreas temáticas solicitan al Proyecto de Calidad de la Facultad Siete la aceptación o liberación del producto.

#### 3.4.1 Propósito:

- ✓ Asegurar que se tengan en cuenta todos los Casos de Prueba posibles para validar la solución informática para que en el momento de entregar el producto este cuente con un nivel de aceptación apropiado.
- ✓ Definir todas las actividades relacionadas con la ejecución de las Pruebas de Aceptación, las responsabilidades individuales para cada tarea, los recursos y los prerrequisitos que deben ser considerados en el esfuerzo de pruebas.
- ✓ Garantizar el buen funcionamiento de las Pruebas de Aceptación utilizadas.

### 3.4.2 Alcance:

- ✓ Todos los proyectos productivos de la Facultad Siete.
- ✓ En este caso particular las Pruebas de Aceptación están enfocadas en verificar el correcto funcionamiento de todos los requisitos funcionales.

### 3.4.3 Descripción General de Procedimiento

Para la realización del siguiente procedimiento se siguieron una serie de pasos, los cuales están definidos por una sucesión cronológica de operaciones que se relacionan entre sí. Además se generan artefactos que son importantes para el correcto completamiento de este procedimiento.

La elaboración del procedimiento para normalizar las Pruebas de Aceptación en la fase de Depuración y Cambios se hace necesaria, ya que lograr que el producto tenga una aceptación del cliente final es importante para la liberación del mismo. Para la realización de este procedimiento se consta con las siguientes fases:

- ✓ Fase #1: Inicio del Proceso de Pruebas de Aceptación.
- ✓ Fase #2: Revisión de Documentos y Diseño de Casos de Prueba.
- ✓ Fase #3: Aplicación de los Casos de Prueba y Listas de Chequeo.
- ✓ Fase #4: Liberación del Software.

### 3.4.4 Roles del Procedimiento

- ✓ **Probador:** Es el que realiza cada una de las pruebas y ejecuta los Casos de Prueba generando un documento de No Conformidades donde se registran todos los defectos encontrados.
- ✓ **Desarrollador:** Es el encargado de recoger en un documento todos los defectos encontrados por el cliente en el lugar de desarrollo cuando se están realizando las Pruebas de Aceptación de tipo alfa.
- ✓ **Diseñador de Casos de Prueba:** Es el encargado de diseñar cada uno de los Casos de Prueba.
- ✓ **Jefe de Prueba:** Es el responsable de verificar que se realicen correctamente todas las pruebas a desarrollar a lo largo del procedimiento.
- ✓ **Usuario Final:** Es la persona a la que va destinada el producto una vez que ha superado las fases de desarrollo correspondientes y es el encargado de realizar Pruebas Alfa y Beta.

### 3.4.5 Artefactos de Entrada

- ✓ **Cronogramas de entrega y liberación:** Documento donde quedan planificados todos los cronogramas de entrega y de liberación de cada artefacto, será un documento que se genere tanto con el equipo de desarrollo como con el equipo de calidad.
- ✓ **Documento Especificación de Casos de Uso:** En este documento se detallan las actividades y pasos a seguir en cada uno de los casos de usos, donde se realiza una descripción textual de cada uno de los casos de uso del sistema, además de conocer sus precondiciones, pos condiciones y sus actores.
- ✓ **Documento Manual de Usuario:** El objetivo que persigue este documento es dar a conocer a los usuarios finales las características y las formas de funcionamiento del software.
- ✓ **Documento Manual de Instalación:** Es una guía donde se explican los pasos a seguir por el usuario para la instalación del software.
- ✓ **Glosario de Términos:** Documento donde se tiene como propósito recoger las principales definiciones y acrónimos empleados en la realización del software.
- ✓ **Listas de Chequeo:** Es un conjunto de condiciones que debe cumplir previamente el documento o aplicación a probar. Están basadas en la identificación de las técnicas de prueba para evaluar cada subcaracterística de las características de calidad. Además de que una lista de chequeo es un formulario de preguntas, que dependen del objetivo para el cual son usadas.
- ✓ **Documento con Juego de Datos:** Documento donde se especifican datos que sean válidos en la base de datos (datos ya probados) para la realización de las Pruebas de Aceptación.

### 3.4.6 Artefactos de Salida

- ✓ **Documento Plan de Pruebas de Aceptación:** Este plan de pruebas tiene como finalidad dictar los pasos a seguir para realizar las Pruebas de Aceptación. Este plan se basa en su totalidad en pruebas de Caja Negra.
- ✓ **Documento Resumen de Evaluación de Pruebas:** Documento donde se recogen todos los defectos encontrados diariamente en la realización de Pruebas Alfa. Es también el documento a partir del cual se realizan las solicitudes de cambio.
- ✓ **Documento de No Conformidades:** Documento donde se registran todos los defectos y errores encontrados en la realización de pruebas con sus respectivas evidencias.

- ✓ **Documento Informe Final de Prueba:** Documento donde se recoge el análisis de los resultados arrojados durante todo lo sucedido en las diferentes etapas del proceso de prueba y la evaluación del usuario final. (**Ver Anexo#5**)

### 3.4.7 Fases del Procedimiento:

#### 3.4.7.1 Fase #1: Inicio del Proceso de Pruebas de Aceptación.

En esta fase se realizan unas series de actividades dentro de las que se encuentran:

- ✓ Comunicar la propuesta de realización de Pruebas de Aceptación.
- ✓ Planificar cronogramas de entrega y liberación.
- ✓ Elaborar el Plan de Pruebas.

El proceso de Pruebas de Aceptación está basado en la Universidad de las Ciencias Informáticas (UCI) en la presencia de un tercero confiable en este caso el Laboratorio de Pruebas y Certificación que está integrado por el grupo de calidad de la universidad, el cual se encarga de diseñar, guiar y dirigir las Pruebas de Aceptación.

La Facultad Siete consta de un laboratorio de calidad encargado de garantizar el éxito de su producción, además de que todo el software que en él se libere llegue a calidad central con el menor número de errores.

Este proceso de Pruebas de Aceptación debe comenzar cuando cada área temática de la Facultad Siete le comunica al laboratorio de calidad antes mencionado la propuesta de la realización de las Pruebas de Aceptación a un determinado producto de la misma, luego de este paso el equipo de desarrollo junto al equipo de calidad se reúnen y planifican los cronogramas de entrega y liberación de cada artefacto, esto será firmado por ambos grupos y se elabora el Plan de Pruebas en el que se recoge todo lo referente a las pruebas tanto por parte de los desarrolladores como por los probadores, todo lo que se menciona en el proceso que se comienza a describir, aquí está incluido en el Plan de Pruebas, pues este es el que se adjunta lo referente a estas pruebas.

Luego el equipo de desarrollo entrega al equipo de calidad un expediente, este contiene el producto software a probar, el documento de especificación de casos de uso, el manual de usuario y de instalación, un glosario de términos, y en caso de no estar incluidos en la especificación de casos de uso, un

documento que contiene los requerimientos y a que caso de uso corresponde cada uno de ellos. También deberán entregar un documento con los usuarios, permisos, nombres y demás detalles necesarios que se deban conocer para la instalación del software y no estén incluidos en los manuales.

El equipo de desarrollo debe garantizar además, un producto estable en el laboratorio de prueba que el equipo de calidad ha preparado para las mismas, durante los días que se vayan a realizar las pruebas. Para montar este laboratorio el equipo de desarrollo deberá entregar al equipo de calidad con antelación los requerimientos mínimos de hardware y software para que la aplicación funcione.

Según las necesidades y características del software que se esté probando se necesitará además un documento con algunos juegos de datos, con el objetivo de que el probador cuente con todo lo necesario y utilice datos específicos.

Antes de comenzar las pruebas se crea un expediente del producto, este incluye todo lo entregado por los desarrolladores y se le irá incluyendo todo lo que se genere durante el proceso de pruebas.

#### **3.4.7.2 Fase #2: Revisión de Documentos y Diseño de Casos de Prueba.**

Dentro de las actividades a llevar a cabo en esta segunda fase se tienen:

- ✓ Realizar pruebas al Manual de Usuario y Manual de Instalación.
- ✓ Realizar Pruebas Funcionales.
- ✓ Diseñar Casos de Prueba.

Las primeras pruebas que se realizan serán las pruebas al Manual de Instalación, los probadores instalarán un producto que nunca han visto, guiándose solo por el Manual de Instalación, si no pueden hacerlo dejará claro que el manual no cuenta con todo lo necesario o hay algún problema, una vez instalado el software, se comenzarán las pruebas al Manual de Usuario, mediante este los probadores deben ser capaces de moverse por una aplicación que no conocen y comprenderla, así como saber hacia donde ir y que hacer. Pero esto no es lo único que los probadores revisan, además verifican otros detalles como ortografía, redacción, e incluso a veces aunque algo esté bien redactado pueden sugerirlo de una mejor manera, pues consideran que así el usuario final lo comprenderá mejor.

Otras de las pruebas a aplicar son las “Pruebas Funcionales”, las cuales mediante la técnica de “Caja Negra” verifican si el software cumple con los requerimientos definidos por el usuario. Para esto se diseñan los Casos de Prueba, los mismos se realizan a partir de los casos de uso, de forma tal que al final del diseño se tendrá un Caso de Prueba por cada caso de uso existente. A medida que se diseñan los Casos de Prueba se probará el documento de especificación de caso de uso, se comprobará que lo que se encuentra en la aplicación corresponde a lo descrito en el documento, así como la ortografía y la redacción.

Estas pruebas se pueden hacer de forma paralela, se designa un grupo de probadores para hacer las pruebas a los manuales, mientras que otro grupo diseña los Casos de Prueba facilitando así el trabajo del probador y ahorrando tiempo de prolongación del producto.

Si un software está compuesto por varios módulos, siendo muy complejo de probar una vez ya integrado, entonces se prueban primero los módulos del mismo por separado y luego se diseña un Caso de Prueba de integración para probar el software como un todo. Para esto el equipo de desarrollo deberá entregar con el expediente del producto un documento donde exponga cómo los datos manejados de un módulo afectan a otros.

#### **3.4.7.3 Fase #3: Aplicación de los Casos de Prueba y Listas de Chequeos.**

Para esta fase del procedimiento se llevan a cabo actividades como:

- ✓ Probar los Casos de Prueba diseñados.
- ✓ Aplicar las Listas de Chequeo.

Una vez diseñados los Casos de Prueba el equipo de calidad comienza a aplicar los mismos, en muchas ocasiones a medida que se diseñan los Casos de Prueba el software es probado casi en su totalidad y se recogen las no conformidades que aparecen, esto ahorra tiempo, proporciona una mayor cantidad de pruebas y permite una mayor detección de errores y defectos.

Luego de aplicadas las pruebas se obtiene como resultado de las mismas el documento de No Conformidades, en el mismo están todos los defectos, errores y sugerencias realizadas al equipo de

desarrollo, así como también queda documentado en el Caso de Prueba, el resultado de todas las pruebas aplicadas.

Otra técnica que se aplica es la utilización de listas de chequeo para medir atributos de las interfaces y de calidad con los que debe contar el software, estas listas de chequeo se aplican para obtener un resultado cuantitativo de estos atributos y características, el contenido de la lista de chequeo puede variar según las características del producto que se este probando.

El equipo de calidad entrega estos resultados al equipo de desarrollo, el cual trabajará en función de estos, para perfeccionar al máximo el software. Este proceso se repetirá cuantas veces sea necesario y se establecerá una retroalimentación entre los dos equipos de trabajo en cuanto a los defectos encontrados, cuando se entreguen versiones nuevas que solucionen errores señalados por el equipo de calidad el equipo de desarrollo deberá entregar un documento adjunto a la nueva versión, que especifique exactamente que parte se cambió en cada artefacto, para evitar volver a indicar errores ya señalados, y aún no corregidos. Este documento puede ser el mismo documento de No Conformidades con modificaciones en sus columnas según se plantea en las "Bases para la comunicación con un equipo de desarrollo" De esta forma el equipo de calidad se retroalimenta de las correcciones realizadas y sirve de guía a los desarrolladores para el cumplimiento con los cambios necesarios que implican las no conformidades.

Por las características del proceso antes descrito se puede decir que el objetivo de utilizar un tercero confiable en este proceso de Pruebas de Aceptación es lograr un producto con una alta calidad y con la menor cantidad de errores posibles antes de presentarlo al usuario final.

#### **3.4.7.4 Fase #4: Liberación del Software.**

Una vez logrado esto, se crea el laboratorio de pruebas donde el usuario final probará el software, ya que como se había mencionado antes, hay cosas que solo el usuario final puede detectar. Estas pruebas corresponden a pruebas de tipo alfa, están guiadas por miembros del equipo de calidad y contarán con la presencia de un miembro del equipo de desarrollo, porque pueden surgir dudas, o el usuario puede querer saber algo específico, que solo el desarrollador puede responder, pues el probador aunque ha llegado a

conocer el software, no sabe como se hizo el mismo, sino como funciona, Además pueden estar presentes algunos especialistas funcionales, estos son personas que pertenecen a la parte del cliente pero no son el usuario final, conocen bien el negocio y se encargan de contactar a los usuarios finales para las pruebas y participan también en las mismas, es con ellos que se definen los días que se dedicarán a las pruebas, los horarios u otra cosa que se necesite.

Para esta etapa del proceso de pruebas los probadores refinan una vez más los Casos de Prueba con la última versión del software, que es el resultado de todas las pruebas anteriores.

Para probar, el usuario final se guiará por los Casos de Prueba ya refinados, así como necesitará el documento según las características del software con los datos que contiene la Base de Datos, aplicará el Caso de Prueba del sistema y las listas de chequeo.

Primeramente se le explica al usuario los objetivos de las pruebas y como debe de realizarlas, a partir de esto los miembros del equipo de calidad recogerán en un documento denominado "Resumen de Evaluación de Pruebas" todos los problemas detectados diariamente y estos resultados se expondrán en una reunión al concluir el día, este Resumen puede realizarse al concluir todas las pruebas, es decir al final de este proceso pero realmente sería un gran problema ya que si algún apunte no se entiende bien entonces no se sabría detectar cuándo ocurrió y realizando este Resumen diariamente si se podría identificar.

Luego de terminados todos los días previstos para las pruebas los probadores realizan las solicitudes de cambio a partir del "Resumen de Evaluación de Pruebas", estas solicitudes se entregan a los desarrolladores los cuales redactarán un documento en que expondrán cuáles proceden a realizarse y cuáles no, así como las causas de estas decisiones. También se realizará el resumen de las listas de chequeo, donde se obtiene un resultado cuantitativo de las mismas.

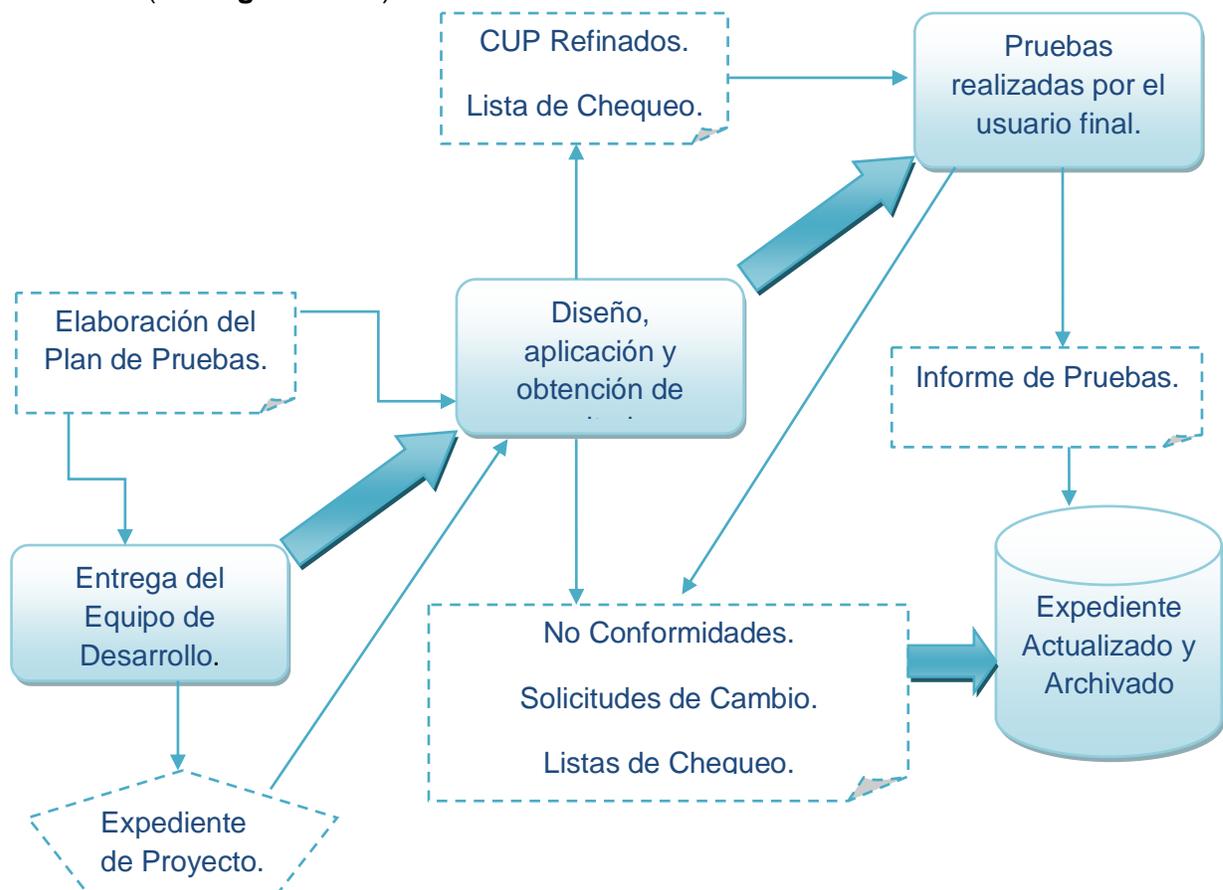
Después de esto se procede a realizar una reunión con todos los especialistas funcionales, probadores y el equipo de desarrollo para discutir este documento, donde se firma el documento que se ha venido realizando "Resumen de Evaluación de Pruebas" afirmando así que existe un mutuo acuerdo de todos los señalamientos realizados junto a las solicitudes de cambio, también se firmará el sumario de las listas de chequeo.

Luego los desarrolladores trabajarán nuevamente pero ahora sobre las solicitudes de cambios realizadas hasta una segunda presentación del software al usuario final y así este proceso se repetirá las veces que sea necesario, con la diferencia de que ya no se diseñarán nuevos Casos de Prueba sino el equipo de calidad refina los ya realizados y los vuelve aplicar así como verificará si estas solicitudes fueron llevadas a cabo.

En la próxima presentación al usuario final el mismo aplicará nuevamente los Casos de Prueba, revisará las solicitudes de cambio para ver si fueron modificadas y volverá a aplicar las listas de chequeo, el resultado de estas será comparado con el de las aplicadas anteriormente.

Si es un software complejo compuesto por diferentes módulos se aplicará este proceso para cada uno de ellos, para cuando termine el mismo el equipo de calidad ya habrá elaborado un Caso de Prueba de Integración y lo aplicará para probar el software integrado, y se volverá a presentar esto al usuario final para que aplique este Caso de Prueba y haga los señalamientos que considere.

Al terminar todo este proceso de prueba se obtendrá el expediente de proyecto ya actualizado con todo lo realizado en cuanto a las Pruebas de Aceptación además de la realización de un Informe Final de Prueba el cual recoge todo lo sucedido en las diferentes etapas del proceso de pruebas y la aceptación por parte del usuario final. (Ver Figura #1.10)



**Figura #1.10 Actividades del Plan de Pruebas de Aceptación.**

En este capítulo, mediante las revisiones bibliográficas, entrevistas a especialistas y análisis de la información se llegó a las siguientes conclusiones:

- ✓ Se debe definir y describir el proceso a seguir para realizar las Pruebas de Aceptación con el cliente en una versión preliminar.
- ✓ Es necesario definir el flujo de comunicación necesario entre el equipo de desarrollo y el grupo de calidad para lograr terminar con éxito la etapa de pruebas.
- ✓ Resulta conveniente definir la documentación empleada y generada en el proceso de Pruebas de Aceptación.
- ✓ Se debe realizar la propuesta de procedimiento para normalizar las Pruebas de Integración, Sistema y Aceptación o Liberación mejorando los procedimientos ya definidos en otras tesis anteriores.

## Conclusiones

Al culminar esta investigación se le ha dado cumplimiento al objetivo general y a las tareas investigativas que se trazaron al principio de este trabajo. Se han tomado las propuestas de procedimientos para cada una de las fases de Depuración y Cambios que propone el modelo en W para mejorar la calidad de todas las aplicaciones y aumentar las expectativas del cliente hacia el producto. Por lo que se arribaron a las siguientes conclusiones:

- ✓ Se definieron procedimientos para normalizar las pruebas durante las fases de Depuración lo que permite garantizar el éxito en el desarrollo de estas fases en el ciclo de vida de las Aplicaciones Web.
- ✓ Con el diagnóstico realizado se pudo conocer el estado actual de los proyectos en cuanto a la realización de Pruebas de Integración, Sistema y Aceptación. Además, se comprobó que no se realizan pruebas de este tipo y se demostró la necesidad de realizar procedimientos que normalizaran estas pruebas.
- ✓ La inserción de nuevas pruebas en cada una de las fases de Depuración y Cambios y el uso de herramientas propuestas después del análisis realizado permitirá un mayor control de la calidad.
- ✓ Debido a la gran importancia del uso de las herramientas en la realización de pruebas en Aplicaciones Web, se recomienda el uso de JMeter y JUnit ya que cumplen con los estándares de producción de la universidad.

### Recomendaciones

Al concluir el presente trabajo de diploma las autoras recomiendan:

- ✓ Aplicar y dar seguimiento a los procedimientos propuestos en cada fase de Depuración y Cambios en los proyectos productivos de la Facultad Siete que desarrollen Aplicaciones Web y en general en la Universidad de las Ciencias Informáticas.
- ✓ Planificar programas que capaciten al personal encargado de llevar a cabo las Pruebas de Integración, Carga, Volumen, Seguridad, Estabilidad y Aceptación en cuanto a las herramientas propuestas para la realización de las mismas y los pasos a seguir.
- ✓ Aplicar este procedimiento en algunos proyectos pilotos, lo que permitirá recoger y analizar los resultados de la aplicación validando así la propuesta de los procedimientos.
- ✓ Se recomienda realizar un nuevo diagnóstico de la producción de la Facultad Siete, luego de período de tiempo de la aplicación de los procedimientos propuestos. Para ello se propone realizar una nueva entrevista para valorar si en los proyectos que aplicaron estos procedimientos se erradicaron los problemas existentes y así poder validar la propuesta realizada.

## Referencias Bibliográficas

1. Gidis. *Gidis*. [En línea] Universidad Nacional de La Pampa, 21 de octubre de 1999. [http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad\\_software.PDF](http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF).
2. Teleformacion. *Teleformacion*. [En línea] UCI, 2007. Conferencia\_Implementacion.pdf. Habana: UCI: Teleformación, 2007.
3. Angelfire. *Angelfire*. [En línea] 2 de noviembre de 2001. <http://www.angelfire.com/my/jimena/ingsoft/guia9.htm>.
4. ETSIIT. *ETSIIT*. [En línea] Universidad de Granada. <http://lsi.ugr.es/~ig1/docis/pruso.pdf>.
5. ETSIIT. *ETSIIT*. [En línea] Universidad de Granada. <http://lsi.ugr.es/~ig1/docis/pruso.pdf>.
6. [En línea] Universidad de Alicante, 9 de febrero de 2008. <http://geeks.ms/blogs/mllopis/archive/2008/02/09/191-en-qu-233-consiste-el-proceso-de-pruebas-de-software.aspx>.
7. ETSIIT. *ETSIIT*. [En línea] Universidad de Granada. [http://lsi.ugr.es/~arroyo/inndoc/doc/pruebas/pruebas\\_d.php](http://lsi.ugr.es/~arroyo/inndoc/doc/pruebas/pruebas_d.php).
8. JACOBSON, G. B., J. RUMBAUGH. *El proceso unificado de desarrollo de software. Quinta edición. Madrid, 2002. 464 p. 84-7829-036-2*. España : s.n., 2002.
9. Teleformacion. *Teleformacion*. [En línea] UCI. [http://teleformacion.uci.cu/file.php/259/Curso\\_2007-2008/Materiales\\_Basicos/Materiales\\_Prueba/Material\\_de\\_caja\\_b\\_y\\_caja\\_n.pdf](http://teleformacion.uci.cu/file.php/259/Curso_2007-2008/Materiales_Basicos/Materiales_Prueba/Material_de_caja_b_y_caja_n.pdf).
10. [En línea] [https://www.u-cursos.cl/ingenieria/2005/1/CC51A/1/material\\_docente/objeto/65585](https://www.u-cursos.cl/ingenieria/2005/1/CC51A/1/material_docente/objeto/65585).
11. Software Guru. *Software Guru*. [En línea] 2007-2009. <http://www.sg.com.mx/content/view/350/>.
12. Idem a la referencia 10. [En línea]
13. Ciria. *Ciria*. [En línea] Universidad de las Americas Puebla, 27 de marzo de 2009. [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/morales\\_h\\_sp/capitulo5.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/morales_h_sp/capitulo5.pdf).
14. Application LifeCycle Solutions. *Application LifeCycle Solutions*. [En línea] 2002-2007. <http://www.als-es.com/home.php?location=herramientas/entorno-pruebas>.
15. Idem a la referencia 14. [En línea]
16. Idem a la referencia 14. [En línea]

17. Techerald.com. *Techerald.com*. [En línea] 5 de Diciembre de 2008. <http://techerald.com/view.jsp?jmeter-una-herramienta-para-pruebas-de-carga-aplicaciones-web-051220083713.html>.
18. *Ingeniería del Software: Un enfoque práctico*. 3ra Edición. págs. Pág. 26-30.
19. *Idem a la referencia 18*.
20. *Idem a la referencia 18*.
21. *Idem a la referencia 18*.
22. *Idem a la referencia 18*.
23. Carolina. *Carolina*. [En línea] 14 de febrero de 2005. <http://carolina.terna.net/ingsw2/Datos/Cascada-ModeloV.doc>.
24. *Idem a la referencia 23*. [En línea]
25. Kinetia. *Kinetia*. [En línea] <http://www.kynetia.es/calidad/metodologia-de-pruebas.html>.
26. *lenguacastellana*. *lenguacastellana*. [En línea] <http://www.memo.com.co/fenonino/aprenda/castellano/castellano3.html>.
27. RePRIS. *RePRIS*. [En línea] Universidad Politecnica de Valencia, mayo de 2007. <http://in2test.lsi.uniovi.es/repris/actividades/TestingJTS2007.pdf>.
28. Universidad Distrital Francisco Jose de Caldas. *Universidad Distrital Francisco Jose de Caldas*. [En línea] 2007. <http://www.udistrital.edu.co/comunidad/grupos/arquisoft/fileadmin/Estudiantes/Pruebas/HTML%20-%20Pruebas%20de%20software/node55.html>.
29. usuario:usuario. *usuario:usuario*. [En línea] ISP UVED, 2009. <http://www.isp.fuac.edu.co/wis/wish/doku.php?id=usuario:usuario>.
30. [En línea] <http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm#s3>.

## Bibliografía

- [En línea]. - Universidad de Alicante, 9 de febrero de 2008. - diciembre de 2008. - <http://geeks.ms/blogs/mllopin/archive/2008/02/09/191-en-qu-233-consiste-el-proceso-de-pruebas-de-software.aspx>.
- [En línea]. - 2008. - [https://www.u-cursos.cl/ingenieria/2005/1/CC51A/1/material\\_docente/objeto/65585](https://www.u-cursos.cl/ingenieria/2005/1/CC51A/1/material_docente/objeto/65585).
- [En línea]. - <http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm#s3>.
- Angelfire** [En línea] // Angelfire. - 2 de noviembre de 2001. - noviembre de 2008. - <http://www.angelfire.com/my/jimena/ingsoft/guia9.htm>.
- Application LifeCycle Solutions** [En línea] // Application LifeCycle Solutions. - 2002-2007. - 2008. - <http://www.als-es.com/home.php?location=herramientas/entorno-pruebas>.
- Calidad en Ingeniería de Software** [En línea]. - <http://dmi.uib.es/~bbuades/calidad/index.htm>.
- Carolina** [En línea] // Carolina. - 14 de febrero de 2005. - 2008. - <http://carolina.terna.net/ingsw2/Datos/Cascada-ModeloV.doc>.
- Ciria** [En línea] // Ciria. - Universidad de las Americas Puebla, 27 de marzo de 2009. - [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/morales\\_h\\_sp/capitulo5.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/morales_h_sp/capitulo5.pdf).
- Definicion** [En línea] // Definicion. - <http://www.definicion.org/procedimiento>.
- ETSIIT** [En línea] // ETSIIT. - Universidad de Granada. - noviembre de 2008. - <http://lsi.ugr.es/~ig1/docis/pruso.pdf>.
- ETSIIT** [En línea] // ETSIIT. - Universidad de Granada. - noviembre de 2008. - <http://lsi.ugr.es/~ig1/docis/pruso.pdf>.
- ETSIIT** [En línea] // ETSIIT. - Universidad de Granada. - 2008. - [http://lsi.ugr.es/~arroyo/inndoc/doc/pruebas/pruebas\\_d.php](http://lsi.ugr.es/~arroyo/inndoc/doc/pruebas/pruebas_d.php).
- Gidis** [En línea] // Gidis. - Universidad Nacional de La Pampa, 21 de octubre de 1999. - noviembre de 2008. - [http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad\\_software.PDF](http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF).
- Ingeniería del Software: Un enfoque práctico. 3ra Edición. págs. Pág. 26-30.** [Libro].

**INGENIERIA INDUSTRIAL (INSTITUTO SUPERIOR POLITECNICO ECHEVARRIA)** [Publicación periódica]. - Ciudad de La Habana : INST. SUPERIOR POLITECNICO JOSE ANTONIO ECHEVERRIA : LA HABANA. - Vol. 024.

**JACOBSON, G. B., J. RUMBAUGH. El proceso unificado de desarrollo de software. Quinta edición. Madrid, 2002. 464 p. 84-7829-036-2** [Libro]. - España : [s.n.], 2002.

**Kinetia** [En línea] // Kinetia. - 2008. - <http://www.kynetia.es/calidad/metodologia-de-pruebas.html>.

**lenguacastellana** [En línea] // lenguacastellana. - <http://www.memo.com.co/fenonino/aprenda/castellano/castellano3.html>.

**Qualitrain** [En línea] // Qualitrain. - enero de 2009. - <http://www.qualitrain.com.mx/index.php/Procesos-de-software/Aseguramiento-de-la-calidad-de-software.html>.

**RePRIS** [En línea] // RePRIS. - Universidad Politecnica de Valencia, mayo de 2007. - <http://in2test.lsi.uniovi.es/repris/actividades/TestingJTS2007.pdf>.

**Software Guru** [En línea] // Software Guru. - 2007-2009. - <http://www.sg.com.mx/content/view/350/>.

**SPRI** [En línea] // SPRI. - [www.spri.es/ddweb/inicio/cursos/DD/ct/UNIDAD%201.pdf](http://www.spri.es/ddweb/inicio/cursos/DD/ct/UNIDAD%201.pdf).

**Techerald.com** [En línea] // Techerald.com. - 5 de Diciembre de 2008. - 2008. - <http://techerald.com/view.jsp?jmeter-una-herramienta-para-pruebas-de-carga-aplicaciones-web-051220083713.html>.

**Teleformacion** [En línea] // Teleformacion. - UCI, 2007. - noviembre de 2008. - Conferencia\_Implementacion.pdf. Habana: UCI: Tele formación, 2007.

**Teleformacion** [En línea] // Teleformacion. - UCI. - 2008. - [http://teleformacion.uci.cu/file.php/259/Curso\\_2007-2008/Materiales\\_Basicos/Materiales\\_Prueba/Material\\_de\\_caja\\_b\\_y\\_caja\\_n.pdf](http://teleformacion.uci.cu/file.php/259/Curso_2007-2008/Materiales_Basicos/Materiales_Prueba/Material_de_caja_b_y_caja_n.pdf).

**Un enfoque actual sobre la calidad del software** [En línea] // Un enfoque actual sobre la calidad del software. - [http://bvs.sld.cu/revistas/aci/vol3\\_3\\_95/aci05395.htm](http://bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm).

**Universidad Distrital Francisco Jose de Caldas** [En línea] // Universidad Distrital Francisco Jose de Caldas. - 2007. - <http://www.udistrital.edu.co/comunidad/grupos/arquisoft/fileadmin/Estudiantes/Pruebas/HTML%20-%20Pruebas%20de%20software/node55.html>.

**usuario:usuario** [En línea] // usuario:usuario. - ISP UVED, 2009. - <http://www.isp.fuac.edu.co/wis/wish/doku.php?id=usuario:usuario>.

## Glosario de Términos

**Área Temática:** Se denomina área temática al conjunto de proyectos que poseen características comunes, es decir que todos se realizan con el mismo fin determinado y para el mismo perfil para así organizar mejor la producción de la Facultad Siete.

**Aplicación:** En informática, una aplicación es un tipo de programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo.

**Calidad:** La calidad de software es la aptitud de un producto o servicio para satisfacer las necesidades del usuario. Es la cualidad de todos los productos, no solamente de equipos sino también de programas.

**Ciclo de Vida:** Tiempo de vida de un producto desde que comienza a elaborarse en sus inicios hasta su liberación cuando es aceptado por el usuario final.

**Depuración y Cambios:** Es el nombre de fases propuestas por el modelo en W donde se realizan Pruebas de Integración, Sistema y Aceptación.

**Fase:** Parte Homogénea de un sistema que está en contacto con otra del mismo, pero que están separadas por una frontera bien definida.

**Listas de Chequeo:** Herramientas utilizadas para chequear características generales del producto que no se verifican a través de las pruebas.

**Normalizar:** En este caso se quiere normalizar las pruebas durante las fases de Depuración y Cambios, entonces significaría lograr estandarizar que la realización de estas pruebas se ajuste a los procedimientos y reglas definidas en las propuestas realizadas.

**PC:** Computadora.

**Proceso:** Un proceso puede ser definido como un conjunto de actividades enlazadas entre sí que, partiendo de uno o más inputs (entradas) los transforma, generando un output (resultado).

**Propuesta:** Es algo que se propone con el propósito de ponerlo en práctica y mejorar lo que está definido.

**Procedimiento:** Sucesión cronológica de operaciones concatenadas entre sí, que se constituyen en una unidad de función para la realización de una actividad o tarea específica dentro de un ámbito predeterminado de aplicación. Todo procedimiento involucra actividades y tareas del personal, determinación de tiempos de métodos de trabajo y de control para lograr el cabal, oportuno y eficiente desarrollo de las operaciones.

**Pruebas de Liberación:** Conjunto de pruebas y revisiones desarrolladas por el grupo de calidad de la Facultad Siete y posteriormente por el equipo de calidad central de la universidad para garantizar la satisfacción del cliente y un software libre de errores.

**Proyecto Productivo:** Se refiere a un grupo de trabajo, que produce resultados observables y productivos beneficiando al cliente. Los miembros del mismo juegan varios roles para así estar mas organizados y obtener mejores resultados.

**Requerimientos:** Son todos los requisitos que debe cumplir el software para así llegar a satisfacer las expectativas del cliente. La definición de requerimientos marca el inicio de todo proceso y determina en gran medida el resultado del mismo.

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

**Usuario:** Persona que utiliza determinado software o hardware mediante el cual obtiene un servicio.

## Anexos

### Anexo#1

#### Entrevista a realizar en los proyectos productivos de la Facultad Siete.

Nombre de Proyecto: \_\_\_\_\_ Área Temática: \_\_\_\_\_

Nombre del Entrevistado: \_\_\_\_\_

¿Cómo considera la calidad de las aplicaciones que se realizan en este proyecto?: E\_\_\_ B\_\_\_ R\_\_\_ M\_\_\_

¿Existe el grupo interno de calidad en el proyecto? Si\_\_\_ No\_\_\_

¿Cómo funciona el mismo?

---

---

---

---

¿Sabe usted las fases existentes en todo el ciclo de vida del software? Si\_\_\_ No\_\_\_

De ser positiva la respuesta anterior.

¿En cuáles fases se le realizan pruebas al software?

---

---

Si el grupo de calidad interno del proyecto funciona.

¿Qué tipo de pruebas se le realizan al software y con que frecuencia?

---

---

---

¿En su proyecto se aplica algún tipo de Listas de chequeo? Si\_\_\_ No\_\_\_

¿Conoce alguna herramienta para la realización de pruebas? Si\_\_\_ No\_\_\_

Si la Respuesta es si

¿Se aplica Alguna en su proyecto? Si \_\_\_ No \_\_\_ \_

¿Cuál(es)? \_\_\_\_\_

¿Sabe usted lo que es un plan de pruebas? Si \_\_\_ No \_\_\_ No sé \_\_\_

Según sus conocimientos, las Pruebas de Aceptación las realiza:

\_\_\_ El desarrollador \_\_\_ El grupo de calidad interno \_\_\_ El usuario \_\_\_ El grupo de calidad de la facultad

\_\_\_ El cliente final.

¿En su proyecto se lleva algún tipo de registro de defectos detectados durante las pruebas realizadas?

Si \_\_\_ No \_\_\_

## Anexo#2

### Plan de Pruebas

Un plan de pruebas incluye:

1. Identificador del plan.

Se identifica el nombre del plan de pruebas a desarrollar.

2. Alcance

Indica el tipo de prueba y las propiedades.

3. Estrategia

Se describen las técnicas y herramientas usadas para crear el procedimiento, así como la secuencia de pasos que definieron el mismo.

4. Procedimientos Especiales

Identifica el grafo de las tareas necesarias para preparar y ejecutar las pruebas.

5. Tangibles

Explicita los documentos a entregarse al culminar el proceso previsto por el plan.

6. Recursos

Especifica las propiedades necesarias y deseables del ambiente de prueba.

## Anexo#3 Plantilla de No Conformidades

Nombre del proyecto

Versión 1.x

## Plantilla de No Conformidades

Elemento a probar &lt;Nombre del artefacto a probar&gt;

Versión 1.x

## Control de versiones:

Fecha	Versión	Descripción	Autor
<00/00/0000>	<1.0>	<Descripción>	<Autor>

## Descripción General

## Elementos probados

## Elementos no probados y causas

## 1. Registro de defectos y dificultades detectados

Elemento	No	No conformidad	Aspecto correspondiente	Etapas de detección	Clasificación	Estado NC	Respuesta del Equipo Desarrollo
<Nombre del Elemento>	<1>	<Descripción de la No Conformidad>	<Descripción del Aspecto correspondiente>	<Etapas de detección del error>	S: Significativa NS: No Significativa R: Recomendación	[Se coloca el estado de la NC y la fecha, cada vez que se revise se deja el estado anterior y se coloca el nuevo con la fecha en que se revisó.] RA:	[Esta columna se comienza a llenar a partir de la 2da iteración, y es responsabilidad del equipo de desarrollo, quien especifica la conformidad con lo encontrado o no y en caso de no proceder la no conformidad explica por qué.]

## Anexo #4

### Análisis de Carga por Caso de Uso

**1. Información General:** En esta sección se especifica información general acerca del proyecto sobre el que se van a aplicar las pruebas de carga.

- Producto: Se debe escribir el nombre del producto
- Versión: indica la versión en la cuál se ejecutarán las pruebas.
- Entorno: Escriba el ambiente en el cual se van a realizar las pruebas, por ejemplo Desarrollo, Soporte o Clientes o los equivalentes de acuerdo a como cada proyecto los haya definido.
- Responsable: Persona encargada de realizar la estimación de usuarios concurrentes y número de transacciones por unidad de tiempo para cada caso de uso que forma parte del proyecto.

### 2. Estimación de Carga por Caso de Uso

<b>Caso de Uso</b>	<b>No. de Usuarios concurrentes</b>	<b>No. de Transacciones</b>	<b>Unidad de Tiempo (Transacciones)</b>
<Nombre del caso de uso>	<Cantidad estimada de usuarios concurrentes> Promedio: Pico:	<Cantidad estimada de transacciones por unidad de tiempo> Promedio: Pico:	< Unidad de tiempo sobre la cuál se van a medir las transacciones >

## Anexo #5

## Documento Informe Final de Prueba

## INFRAESTRUCTURA PRODUCTIVA

**INFORME FINAL DE PRUEBA****Número de control:**

**Proyecto:** *(Nombre de la entidad a revisar en este caso nombre del proyecto)*

**No Conformidades:**

*00 DE MES DE 0000*

**Control del Documento**

	<b>Nombre</b>	<b>Cargo</b>
Redactado por	<i>Nombre y Apellidos</i>	Revisor líder
Revisado por	<i>Nombre y Apellidos</i>	Director de Calidad
Aprobado por	<i>Nombre y Apellidos</i>	Director General

## Índice de Contenidos

Resumen.....	#
Alcance.....	#
Metodología.....	#
Acta de Evaluación.....	#

### Resumen

*Breve resumen del informe final de prueba.*

### Alcance

*En la sección del informe que se refiere al alcance, se debe expresar la profundidad y el informe debe describir el alcance de las pruebas del revisor y presentar los resultados de esas pruebas, así como hacer referencia a informes separados que contengan dicha información.*

*Al presentar los resultados de dichas pruebas, el revisor debe informar sobre las irregularidades, incumplimientos significativos y situaciones que deben considerarse.*

### Metodología

*Definir los métodos de obtención de información que se realizaron.*

### Acta de Evaluación

Resumen de la Revisión

La revisión se desarrolló entre los días # y #, fue *calidad* el ambiente, permitió identificar que el proyecto está *bien* pero se realizaron # No Conformidades las cuales # tienen importancia alta. Se destaca la cooperación del equipo de proyecto en la realización de la revisión y la cooperación de todos.

#### EVALUACIÓN:

*En esta evaluación primeramente se da un resumen de la cantidad de no conformidades encontradas con sus respectivas evidencias y una vez concluido este resumen se da la evaluación de aceptado o no aceptado por parte del usuario.*

## Anexo #6

## Informe Parcial de Resultados

Etapa	Elemento a revisar	Resultados Obtenidos
1era Iteración	Comportamiento de la Asistencia	<Se especifica la cantidad de revisores que deben de llevar a cabo la tarea a realizar y si todos asistieron al cumplimiento de la tarea>
	Nombre de la Tarea(s)	<Se especifican los resultados de las tarea o tareas (si se realizaron en el tiempo especificada para la misma) y la calidad con que se efectuaron las mismas.
2da Iteración		

Cant. Significativas	NC	Cant. Significativas	NC	No	Cant. Recomendaciones	Observaciones
[Se especifica la cantidad de Significativas]	NC	[Se especifica la cantidad de Significativas]	NC	No	[Se especifica la cantidad de Recomendaciones]	[En caso de que existan NC pendientes o generalizadas se explica en esta columna]

## Causas de los Resultados Obtenidos Negativamente

Resultados obtenidos de Cada Elemento a Revisar	Causas
<Aquí se pone el apesto negativo que afecto el desarrollo de las pruebas o revisiones>  <b>Nota:</b> Se ponen en una fila distinta cada resultado.	<Aquí se ponen las causas o justificación de por qué dicho resultado negativo >