Universidad de las Ciencias Informáticas

“Facultad 3”

Título: “Pruebas de Software para el módulo Servicio Autónomo en el proyecto Registros y Notarías”.

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autor(es): Dayma Direntau Batista

Adriana Román Herrera

Tutor(es): Ing. Yanosky Ríos La Hoz

16 de junio del 2007.

“Año 49 de la Revolución”

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

_____	_____	_____
Adriana Román Herrera	Dayma Direntau Batista	Ing. Yanosky Ríos La Hoz
Autor	Autor	Tutor

Dedicatoria

A mi mimi por creer en mí y estar siempre conmigo, por ser tan unidas, por haber vencido todos los obstáculos que nos ha puesto la vida.

A la memoria de mis abuelitos, que hubiesen disfrutado mucho este momento, nunca los olvidaré, fueron los mejores.

A mi titi, por brindarme su amor y apoyarme en todo, por malcriarme, por enseñarme a madurar y ser independiente, por tener confianza en mí.

A mi tío, mis primos, por querer verme ingeniera.

A Lidia y Bruno por su apoyo incondicional en todo momento.

Adriana

A mi mamá porque sin su ayuda y apoyo en estos momentos de mi vida no hubiese llegado hasta aquí.

A mi papá y mi hermano por impulsarme siempre a realizar las cosas de una mejor manera y servirme de ejemplo.

A Darién por incentivarme a seguir superándome en todo momento y hacerle frente a todos los problemas desde los inicios de mi carrera.

A mi abuelita querida y mis tíos por ser tan buenos conmigo y quererme tanto.

Dayma

Agradecimientos

Agradecemos:

A nuestro tutor por estar siempre dispuesto a ayudarnos a pesar de estar tan ocupado.

A Darién y Osmín por guiarnos, que fuera de nosotras sin ellos.

Al profe Michael que nos sirvió de mucha ayuda en todo momento para la culminación de la tesis.

A Yada por estar siempre a nuestro lado y apoyarnos en todo.

A Iosmel por molestarlo en tantas ocasiones para configurar el entorno de pruebas.

A los integrantes del módulo Servicio Autónomo, a Daynier, Marbys, Yusbel.

A Denis, Nemury y Blanco por aclararnos nuestras dudas en los momentos críticos.

A todas aquellas personas de las sedes de DESOFT que colaboraron con nosotras en la encuesta realizada, especialmente a Rodney.

Resumen

La base de la calidad del software es el cumplimiento de los requisitos que solicita el cliente, de modo que se logre la satisfacción de éste con la elaboración del producto.

Para lograr este objetivo, es favorable el uso de métricas, herramientas, metodologías que controlen la calidad durante todo el proceso de desarrollo ya que actualmente un producto de software, para su inserción en el mercado debe contar con este requisito.

Antes de que el software se le entregue al usuario final es necesario realizar pruebas con el objetivo de detectar errores de la aplicación y la documentación; este proceso resulta de gran importancia ya que da una medida de la calidad del producto siempre que se lleve a cabo de forma apropiada.

Muchas empresas en el mundo han definido sus propias metodologías y herramientas que permiten elevar el nivel de calidad del software y llevar a cabo un proceso de pruebas de forma más efectiva con la detección y corrección temprana de los errores.

Cuba también ha visto la importancia de que los productos tengan calidad por lo cual se esfuerza cada día más para obtenerla y así poder introducirse en el mercado.

En la Universidad de Ciencias Informáticas (UCI) se ha venido desarrollando un software para la informatización de los Registros Mercantiles e Inmobiliarios en Venezuela. Este proyecto consta de cuatro módulos y uno de ellos es Servicio Autónomo el cual controla las actividades sobre los registros.

Debido a esto, el presente trabajo se centra en la realización de las pruebas de software a dicho módulo, con el objetivo de encontrar la mayor cantidad posible de errores en el mismo para lograr un producto fiable. Para ello se utiliza la metodología RUP, donde se realiza una planificación inicial de la pruebas definiendo los requerimientos de prueba, la estrategia a seguir, el cronograma, los roles y los recursos necesarios para probar la aplicación en un ambiente lo más cercano posible al entorno del usuario final. Se diseñan y realizan pruebas funcionales y se hace un análisis de los resultados reales y los esperados para corregir los errores encontrados.

Palabras claves

Proceso de pruebas de software.

Tabla de Contenidos

Introducción

La Ingeniería de Software nace a mediados de 1968 como una “disciplina de aplicación inteligente de principios probados, técnicas, lenguajes y herramientas para la creación y mantenimiento, dentro de un coste razonable, de software que satisfaga las necesidades de los usuarios” (1). Surge debido al proceso tradicional de ensayo y error utilizado por los programadores, el cual se dificultó con el avance del hardware y software y la vinculación de la informática a las diferentes ramas de la sociedad. Este hecho conllevó a que se produjera un software sin calidad debido a (2): “la falta de entendimiento de los requisitos, fallas en los estimados de tiempo y esfuerzo al programar, variaciones en la productividad de los desarrolladores, dificultad en separar el diseño y la construcción del software, rápido crecimiento de su demanda”.

Esta disciplina ha contribuido al desarrollo de diferentes metodologías como: Rational Unified Process (RUP) y Extreme Programing (XP), consideradas las más importantes, con el objetivo de mejorar el proceso de desarrollo de software.

Actualmente, la humanidad está inmersa en un proceso de constante cambio y evolución de la tecnología, esto conlleva a que muchos países creen productos de software cada vez más competitivos para su inserción en el mercado incluyendo a Cuba, quien ha definido como estrategia la creación de la Universidad de Ciencias Informáticas (UCI) como centro de estudios y desarrollo de software. Para lograr dicho objetivo es necesario aplicar adecuadamente las metodologías, procedimientos, estándares, herramientas que permitan desarrollar, con la calidad requerida, el proceso de software y así obtener un producto de gran aceptación para el cliente.

En la UCI se están desarrollando una serie de proyectos de diferente magnitud y complejidad, entre ellos un software para la informatización de los Registros y Notarías de Venezuela al cual están vinculados un grupo de profesores y estudiantes de 4to y 5to año de la facultad 3. Este proyecto presenta cuatro módulos: Administración Contable, Mercantil, Inmobiliario y Servicio Autónomo. Este último controla la parte legal y administrativa de los registros mercantiles e inmobiliarios. En dicho módulo se vio afectada, en los inicios de la fase de elaboración, la calidad del producto, pues no se siguió debidamente la metodología propuesta ya que no se llevó a cabo una buena planificación del proyecto, trayendo consigo la no elaboración de un plan de pruebas, es decir, éstas no se introdujeron desde el inicio del proceso de desarrollo, no se identificaron los riesgos ni se trazó una estrategia para realizarlas, no se definieron roles

ni las actividades que deberían tener asignados, provocando desorganización y menor productividad en el trabajo. Se produjo una mala gestión de requisitos lo cual dificultó el diseño de las pruebas. Debido a esto se definieron fechas de entrega con las cuales no se pudo cumplir, provocando que el tiempo de su realización disminuyera en los primeros ciclos de pruebas de dicha fase, no se encontraron todos los defectos dificultando el proceso de depuración, no se contó con herramientas de automatización que pudieran acelerar el proceso generándose los casos de prueba de forma manual lo que consumió la mayor cantidad de tiempo y aumento de los costos. Debido a esta situación, se ha definido el siguiente problema:

La carencia de un proceso de pruebas bien definido está afectando la calidad en el módulo Servicio Autónomo.

Objeto de Estudio

Proceso de pruebas de software.

Campo de Acción

Proceso de pruebas de software durante el desarrollo en el módulo Servicio Autónomo perteneciente al proyecto Registros y Notarías.

Hipótesis

Si se aplica un proceso de pruebas bien definido en el módulo Servicio Autónomo se podrá obtener un producto de aceptada calidad.

El autor define como objetivo general de la tesis:

Aplicar correctamente el proceso de pruebas propuesto para el módulo Servicio Autónomo que permita obtener un producto de aceptada calidad.

Partiendo del objetivo general se han definido los siguientes objetivos específicos:

- Organizar el proceso de pruebas de software para el módulo.

- Establecer las pruebas de software de tipo funcional.
- Evaluar los resultados de la ejecución de las pruebas.

Para el cumplimiento de los objetivos específicos se han determinado las siguientes tareas de investigación:

1. Revisar la documentación técnica del proyecto necesaria para las pruebas.
2. Analizar las plantillas definidas por la UCI y RUP para las pruebas.
3. Analizar los recursos disponibles del proyecto.
4. Definir el plan de pruebas detallando los requerimientos de prueba, la estrategia a seguir, recursos, los entregables y el cronograma.
5. Diseñar la configuración del entorno de pruebas.
6. Implantar el entorno de pruebas para realizar pruebas funcionales.
7. Apoyarse en el jefe de módulo y el responsable de base de datos para la instalación de la aplicación.
8. Consultar a los integrantes del módulo Servicio Autónomo para conocer los datos reales que se utilizarán en las pruebas.
9. Detallar los datos de pruebas.
10. Diseñar los casos y procedimientos de pruebas utilizando la técnica de caja negra.
11. Ejecutar las pruebas funcionales en el módulo.
12. Comparar los resultados esperados y reales que se obtienen de la ejecución de las pruebas.

El trabajo consta de los siguientes capítulos:

El capítulo 1 se centra en el fundamento teórico del objeto de estudio, se abordan aspectos importantes como la implicación que tiene la prueba en la calidad del software, se define su concepto, sus objetivos y principios. Otra de las temáticas que se tratan es el estado del arte a nivel internacional, nacional y en la UCI. Se realiza una descripción de las técnicas y tipos de pruebas existentes y cómo influye

negativamente el surgimiento de la orientación a objeto en el proceso de prueba. Se realiza además, un estudio de las metodologías para analizar cómo llevan a cabo dicho proceso.

El capítulo 2 se basa en la actividad de planificación del proceso de pruebas, donde se describe el producto y se realizan dos artefactos propuestos por la metodología RUP: la configuración del entorno de pruebas para especificar los recursos necesarios que permitan probar el software en un ambiente similar al real, y el plan de pruebas donde se definen los requisitos a probar, la estrategia a seguir, el cronograma, los entregables, los roles y procedimientos a seguir dentro del proceso.

En el capítulo 3 se realiza el diseño, ejecución y evaluación de las pruebas de acuerdo a la planificación de las mismas para lo cual se desarrollan los siguientes artefactos: los datos de pruebas, el modelo de pruebas que contiene el diseño de los casos de pruebas, así como el procedimiento a seguir para ejecutarlos, y el sumario de evaluación de las pruebas donde se expone un resumen de los resultados de las pruebas, éstas se evalúan de acuerdo a la métrica basada en los requisitos del software y se realiza un análisis de los defectos encontrados.

Capítulo 1: Fundamento Teórico

1.1. Introducción

Con el objetivo de llevar a cabo un proceso apropiado referente a las pruebas de software, en el presente capítulo se hace un análisis de los conceptos de prueba dados por varios autores, así como su importancia en la calidad del software, se dan a conocer sus objetivos, principios que la rigen y una descripción del proceso. Se detallan las técnicas, los tipos de prueba que se realizan para métodos convencionales y software orientado a objetos, se establece una comparación entre las metodologías RUP y XP para definir cuál se adecua más a las características del módulo. Se hace un estudio del estado del arte a nivel nacional, internacional y en la Universidad de Ciencias Informáticas para saber cuáles son las tendencias fundamentales y el nivel de importancia que se le da a dicho proceso.

1.2. Pruebas en la Calidad del Software

La calidad del software “es la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente” (3).

Hoy en día se buscan no solo software que funcionen sino que tengan alta calidad. Aunque en muchas ocasiones no se tome muy en serio, esta disciplina tiene gran importancia tanto para el cliente como para los desarrolladores ya que un software de baja calidad puede prolongar el proceso de desarrollo del producto de software y hacerlo más engorroso, lo que conlleva en muchos casos al fracaso del proyecto.

Una de las actividades críticas en el aseguramiento de la calidad es la prueba, la cual para Myers (4), es el “proceso de ejecutar un programa con el fin de encontrar errores”. El nombre “prueba”, además de la actividad de probar, se puede utilizar para designar “un conjunto de casos y procedimientos de prueba” (5).

“La prueba (flujo de trabajo) es el flujo de trabajo fundamental cuyo propósito general es comprobar el resultado de la implementación mediante las pruebas de cada construcción, incluyendo tanto

construcciones internas como intermedias, así como las versiones finales del sistema que van a ser entregadas a terceras partes" (6).

Se considera que la prueba de software es un proceso para su verificación y validación, es decir, el producto que se obtiene implementa sus funciones correctamente y se valida la funcionalidad del sistema, esta validación está basada en la observación de casos de pruebas, los resultados se registran y evalúan, para mejores resultados se pueden ejecutar en un entorno de prueba similar al entorno en que se vaya a desplegar el sistema, se deben aplicar incluyendo pruebas de las construcciones internas, intermedias y las versiones finales del sistema, para de esta forma ir probando de lo más simple a lo más complejo, haciéndose más fácil y menos costoso el descubrimiento de los errores.

Para que el proceso de pruebas sea totalmente efectivo es necesario tener en cuenta cuáles son los objetivos de las mismas, los cuales se detallan a continuación.

1.3. Objetivos de la Prueba

Los objetivos de la prueba de software según Glen Myers (4) son:

- La prueba es el proceso de ejecución de un programa con la intención de descubrir un error.
- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- Una prueba tiene éxito si descubre un error no detectado hasta entonces.

Dados estos objetivos, se estima que es de gran importancia su cumplimiento ya que dicho proceso posibilita no solo la detección de errores de la aplicación, verificando el cumplimiento de los requisitos funcionales y no funcionales, errores ortográficos, abreviaturas, la funcionalidad de vínculos, botones, la ayuda, sino también se verifica la correspondencia de la documentación con la aplicación, dígase especificaciones de casos de uso del sistema, manuales de usuario y de instalación. Se verifica además, la lógica interna del programa, pues un buen resultado de la implementación no solo depende de la codificación sino también de las pruebas. La realización de un proceso adecuado facilita que no aumenten los costos y tiempo del proyecto, que disminuyan los riesgos asociados y además que se eleve el nivel de calidad del producto, de esta forma, una vez que el mismo llegue a manos del cliente, va a tener la menor cantidad de errores dando como resultado la satisfacción del usuario final.

Se considera que para obtener un buen caso de prueba se debe tener en cuenta también, que no se hagan pruebas que tengan el mismo propósito, es decir, redundantes ya que se pierde tiempo en el diseño y ejecución de las mismas.

Para el descubrimiento de los errores es necesario conocer con claridad los conceptos de error, defecto y fallo, pues en algunas bibliografías (5) se dice que el error es un defecto, sin embargo en el presente trabajo se considera que sus significados son diferentes:

Error**:** Es un descuido efectuado por una persona que interviene en el desarrollo del software lo que conduce a un fallo en el sistema.

Fallo: Es el incumplimiento o la realización incorrecta de alguna funcionalidad que debe desempeñar el sistema.

Defecto**:** Es un desperfecto que tiene el software el cual se debe registrar para poder encontrar el error que lo originó y el desarrollador que debe corregirlo.

Se estima que el error es producido por humanos y conlleva a la aparición de defectos en el software, lo que luego conduce a que el sistema no haga lo que debe, es decir, a un fallo.

Una vez conocidos los objetivos de las pruebas, se deben seguir una serie de principios para que el proceso tenga calidad, éstos se explican a continuación.

1.4. Principios de la Prueba

Para la realización de las pruebas, Davis A. define entre otros principios (7):

- A todas las pruebas se les debería poder hacer un seguimiento hasta los requisitos de los clientes (trazabilidad).
- Las pruebas deberían planificarse antes de que empiecen.
- El principio de Pareto es aplicable a la prueba del software (“donde hay un defecto, hay otros”).
- Las pruebas deberían empezar por “lo pequeño” y progresar hacia “lo grande”.
- No son posibles las pruebas exhaustivas.
- Para ser más efectivas, las pruebas deberían ser conducidas por un equipo independiente.
- Se deben evitar los casos de prueba no documentados ni diseñados con cuidado.

- No deben realizarse planes de prueba suponiendo que prácticamente no hay defectos en los programas y, por tanto, dedicando pocos recursos a las pruebas.

Se considera que estos principios reflejan cómo se debe realizar el proceso de pruebas de la forma más adecuada, pues es fundamental tener una visión general de cómo se va a llevar a cabo el mismo, trazar una estrategia que permita ir probando la aplicación a medida que se vayan obteniendo sus funcionalidades para ir corrigiendo los defectos que aparezcan. Es importante además que los casos de prueba sean diseñados correctamente de forma que se tengan en cuenta la mayor cantidad de combinaciones posibles que pueden generar defectos así como documentarlos cuidadosamente para el posterior entendimiento por parte de los desarrolladores. El motivo de confiar en que se ha hecho lo correcto es el mayor riesgo de no realizar las pruebas como se debe, es por eso que resulta de gran beneficio que las pruebas no sólo sean ejecutadas por los desarrolladores sino también por un grupo independiente.

El proceso de pruebas define un conjunto de actividades que lo guían y facilitan la detección temprana de un gran número de defectos siempre que se lleve a cabo adecuadamente. Éstas se explican en el epígrafe siguiente.

1.5. Proceso de Prueba

Las actividades que se llevan a cabo dentro del proceso de pruebas son:

- Planificación de las pruebas: Para realizar esta actividad es necesario conocer las características del proyecto y del software, de forma que se puedan definir los roles y recursos del sistema para un apropiado entorno de pruebas, estrategia a seguir, concretar aquellos requerimientos que van a ser probados, los entregables y el cronograma.
- Diseño de las pruebas: Esta actividad requiere de técnicas adecuadas para el diseño de casos y procedimientos de prueba.
- Implementación de las pruebas: Esta actividad puede o no realizarse, lo cual depende de que se cuente o no con herramientas adecuadas para automatizar las pruebas. Es conveniente tener en cuenta que no siempre es necesario automatizarlas, generalmente esto se hace para las que van a

ser usadas en otros ciclos de prueba y para las complejas, que al generar defectos se deben repetir para verificar la corrección de los mismos y se pierde tiempo del propio proceso.

- Ejecución de las pruebas: Una vez que han sido diseñados los casos de prueba, se procede a la ejecución de las mismas con el objetivo de detectar errores.
- Evaluación de los resultados: Se verifica la correspondencia que existe entre los resultados esperados y reales y se obtienen estadísticas de los errores.
- Análisis de errores: Se analizan los errores encontrados dándoles una clasificación para detectar las causas más frecuentes de su aparición y se predice la fiabilidad del software.

1.5.1. Proceso de depuración

Este proceso comienza al iniciarse la ejecución de las pruebas, una vez que se hayan encontrado los errores, el objetivo es localizar el origen del mismo, corregirlo y volver a repetir las pruebas para comprobar que se arregló y no afectó otras funcionalidades. Si no se encuentra el origen pero se deduce cuál pueda ser, se diseñan y ejecutan nuevos casos de prueba.

Con el objetivo de conocer el grado de conocimiento respecto al proceso de pruebas en las empresas que se dedican al desarrollo de software y profundizar en las nuevas tendencias existentes en este sentido, en el siguiente epígrafe se aborda el estado del arte de la calidad y pruebas a nivel internacional, nacional y en la Universidad de las Ciencias Informáticas.

1.6. Estado actual

Actualmente el mundo está en una transición a una sociedad del conocimiento donde el software es el intermediario cada vez más grande entre la información y la inteligencia humana por lo cual han surgido gran cantidad de empresas dedicadas a la producción del mismo, no obstante muchas de ellas no han alcanzado la cultura organizacional necesaria para realizar esta tarea de forma adecuada, es decir que no han logrado que su proceso de desarrollo de software se realice con la calidad requerida, lo cual ya no es una ventaja en el mercado sino un requisito que debe cumplir todo producto de software.

Capítulo 1: Fundamento Teórico

Un pilar fundamental para garantizar la calidad lo constituyen las pruebas de software, no obstante según PROFit[1] "el 75% de las organizaciones que se dedican al desarrollo de software tienen un nivel de madurez bajo en la realización de esta tarea" (8), ya que lo ven como un proceso opcional, aislado del proceso global de desarrollo. Sin embargo hay empresas que se han dado cuenta de la importancia de las pruebas de software y se han centrado en estudiar la disciplina para proponer mejoras a la misma y de esta forma, incrementar la calidad del proceso de desarrollo. Es por esto que hoy en día han surgido dos tendencias fundamentales en este sentido:

- Vincular el proceso de pruebas a lo largo del ciclo de vida de desarrollo de software, para ello se han creado una gran cantidad de metodologías, modelos y estándares por empresas como PROFit la cual "ha desarrollado un modelo de verificación y validación V&V basado en el estándar IEEE Std. 1012-1998 y que además cumple con los requisitos especificados en el modelo CMMI[2], uno de los más extendidos en las áreas de desarrollo. Este modelo tiene su propio ciclo de vida, que se ejecuta en paralelo con el ciclo de vida clásico del desarrollo. Así, la verificación y validación (V&V) abarca a todas las fases de éste, de modo que hay una V&V de los Requisitos, una V&V del Diseño, una V&V de la Construcción, entre otros" (8). Una de las empresas que usa este modelo es SQS la cual es una compañía líder en servicios de Consultoría de Calidad de Software y Pruebas.

Otra de las metodologías creadas para la mejora del proceso de pruebas en el desarrollo del software es la BaQEM[3], definida por GreenSQA (empresa de ingeniería que brinda servicios de aseguramiento de calidad de pruebas de software) para hacer pruebas funcionales a productos de software, con el objetivo de utilizar un mecanismo que sea efectivo para la evaluación de la calidad de los productos de software desarrollados en ParqueSoft[4]. Esta metodología también es en todo el ciclo de vida del producto.

---

[1] Programa de Fomento de la Investigación Técnica.

[2] Capability Maturity Model Integration (Modelo de Integración de la Capacidad de Madurez del proceso de desarrollo de software).

[3] Business Application Quality Evaluation Method.

[4] Parque Tecnológico de Software: Clúster de empresas de base tecnológica especializadas en la industria del conocimiento, a través del desarrollo de productos, soluciones y servicios de software.

Se estima que estos métodos para la corrección y prevención de defectos dan una medida del avance que se está promoviendo en esta área, son métodos que están vinculados a todas las fases del proceso de desarrollo, es decir que se revisan todos los artefactos de salidas de las fases que son fundamentales para el flujo de trabajo de pruebas y que denotan la calidad del producto, esto es muy importante para la disminución del encuentro tardío de los errores en el software de forma considerable ya que en el análisis y el diseño es donde se introducen más de la mitad de los errores.

- La otra tendencia respecto a las pruebas de software es la automatización de sus procesos para lo cual se han creado una serie de herramientas como JUnit (para aplicaciones en Java) y NUnit (para aplicaciones en .NET) que ayudan al diseño y ejecución de las pruebas de caja blanca, el JMeter y Empirix sirven para la realizacion de pruebas funcionales, de carga, entre otras.

Se considera que estas herramientas ayudan en gran medida a la realización de las pruebas de software ya que en nuestros días los sistemas que se necesitan son cada vez más grandes y éstos llegan al límite de la capacidad humana, además refuerza el desarrollo de las pruebas ya que permite que se ejecuten al mismo tiempo que el software que se desea desarrollar disminuyendo tiempo, esfuerzo y costo.

Cuba también ha visto la posibilidad de lograr un espacio en el mercado internacional del software por lo que le ha dado gran prioridad a esta rama de la industria para que se emplee como la fuente de ingreso más significativa en la economía del país. Existen una serie de empresas creadas con este objetivo y una de ellas es la Empresa Nacional de Software (DESOFT) que está destinada a evaluar la eficiencia y eficacia en la gestión de las organizaciones cubanas brindado soluciones en tecnologias, consiste en ser líder en soluciones informáticas integrales en Cuba, tiene aproximadamente 150 clientes y reconocimiento en el mercado internacional, su función es la informatización de la sociedad cubana. Esta empresa presenta una oficina central y una sede en cada provincia que se subordinan a ésta.Una de las tareas que se realiza en la oficina central, es la creación de normas como la ISO/IEC 90003 (una aplicación de la norma ISO/IEC 9001) y la ISO/IEC 2207 (define los procesos de software) para la calidad del producto software las cuales deben ser aplicadas en todas las sedes de DESOFT. Otra de ellas es Segurmática, Empresa de Seguridad Informática en Cuba, que se encarga de la elaboración de antivirus para evitar los ataques de programas malignos. Softel es otra de las empresas cubanas productora y comercializadora de software, con la capacidad de desarrollar sistemas informáticos en una amplia gama de aplicaciones como el turismo, la medicina y la gestión empresarial.

Capítulo 1: Fundamento Teórico

Con el objetivo de saber si se realiza un proceso de pruebas bien definido en Cuba, se ha realizado una encuesta (Ver Anexo 1) dirigida a las empresas de desarrollo de software.

De un total de 25 trabajadores encuestados, los cuales pertenecen a las sedes de DESOFT de Villa Clara, Cienfuegos, Provincia Habana, Ciudad Habana, Matanzas, y a las empresas Segurmática y Softel, se obtubieron los siguientes resultados:

- El 100% le conceden importancia alta a las pruebas de software.
- El 68% utiliza la metodología RUP, el 4% usa la metodología XP y el 28%, otras.
- El 60% lleva a cabo la actividad de planificación, el 96% realiza diseño, el 16%, implementación, el 100% efectúa la ejecución y el 60%, la evaluación de los resultados.
- Sólo el 36% realiza todas las actividades de las pruebas a lo largo del ciclo de vida del software.
- El 72% afirma que en su empresa existe un equipo dedicado a la realización de las pruebas de software los cuales tienen un rol específico.
- El 36% afirma que en su empresa se definen ciclos de prueba a realizar por cada iteración.
- El 44% utilizan como estrategia, probar desde la unidad más pequeña hasta el sistema completo y el 56%, sólo probar el sistema completo.
- El 92% utiliza plantillas para generar los documentos del proceso de prueba.
- El 84% documenta los defectos encontrados y los clasifica de acuerdo a la prioridad de los mismos.
- El 44% realiza pruebas de unidad, el 56% de integración, el 76% lleva a cabo pruebas de sistema, el 96%, funcionales y el 36%, pruebas de aceptación.
- Sólo el 16% utiliza herramientas de automatización de las pruebas.

En la Universidad de las Ciencias Informáticas (UCI) no se tenía experiencia en el proceso de calidad del software debido al corto tiempo desarrollándose en este campo, lo cual provocó que no se llevara a cabo un proceso de pruebas adecuadamente, sin embargo, en estos últimos tiempos se ha visto un interés creciente en este sentido. De acuerdo a estudios realizados en este tema se llegó a la conclusión de trazar una estrategia para mejorar la calidad de los productos, para de ésta forma alcanzar un alto nivel de madurez e introducirse en el mercado del software. En aras de mejorar el proceso de prueba como un elemento de la calidad se estableció un laboratorio de pruebas para crear un entorno adecuado, además

de automatizar los procesos. Se creó un grupo central que controlara y liberara los productos de software con calidad (CaliSoft) y grupos en todas las facultades que respondieran al grupo central con el objetivo de que se interesaran en este tema y se realizaran pruebas a los proyectos de cada facultad. En cada uno de ellos se define la estrategia de pruebas a seguir de acuerdo a sus características propias para obtener la mayor cantidad de defectos posibles y corregirlos en el ciclo de vida del mismo, por lo cual el siguiente epígrafe se centra en el estudio de las técnicas y niveles de pruebas que pueden ser usados para una mejor realización de las mismas y con esto aumentar la calidad del software.

1.7. Técnicas de diseño de casos de pruebas.

Las técnicas de diseño de casos de pruebas se utilizan con el objetivo de facilitar la búsqueda de errores con el mínimo consumo de tiempo y recursos garantizando, de esta forma, la obtención de un producto correcto.

Casos de prueba: Caso de prueba es un artefacto que explica la forma de probar el sistema, detallando las entradas, salidas y las condiciones de la prueba que se va a realizar. Para su mejor realización se debe redactar el procedimiento de pruebas y de esta forma ayudar al ingeniero de pruebas a la ejecución de las mismas.

Para la elaboración de los casos de prueba, se utilizan diferentes técnicas de diseño de casos de prueba:

1.7.1. Técnica de caja blanca:

Se centra en la estructura interna del programa (código) para la obtención de casos de prueba, garantizando que se ejecuten al menos una vez, todos los caminos independientes de un módulo, que se prueben dichos caminos desde el punto de vista lógico, es decir, en sus vertientes verdadera y falsa, los bucles en sus límites y con sus límites operacionales y las estructuras internas de datos para asegurar su validez (3).

Existen varias técnicas de caja blanca, éstas son:

- Prueba del camino básico o cubrimiento: En este tipo de pruebas se elaboran grafos de flujo, se determina su complejidad ciclomática para obtener el conjunto básico de caminos linealmente

independientes y de esta forma los casos de prueba del conjunto básico con el objetivo de ejecutar al menos una vez cada sentencia del programa.

- Prueba de condiciones: Ejercita las condiciones lógicas contenidas en el módulo de un programa para descubrir errores de operadores lógicos, relacionales, paréntesis lógicos o expresiones aritméticas. Esta técnica permite además detectar otros errores dentro de dicho programa.

  Existen algunas estrategias de pruebas de condiciones que permiten dar una orientación para la generación de pruebas adicionales del programa, entre otras se pueden mencionar:

  - Prueba de ramificación: Se centra en la necesidad de ejecutar una sentencia determinada, al menos una vez, en sus vertientes verdadera y falsa.
  - Prueba del dominio: Requiere de la realización de tres o cuatro pruebas para una expresión relacional.
  - BRO[5]: Esta se basa en las dos técnicas anteriores garantizando que se detecten errores de ramificaciones y de operadores relacionales en una condición dada.

- Prueba de bucles: Con el uso de esta técnica se validan las construcciones de bucles con diferentes grados de complejidad:

  - Bucles simples: Se realizan una serie de pruebas donde “n” es el número máximo de pasos permitidos por el bucle:
    - Pasar por alto totalmente el bucle.
    - Pasar una sola vez por el bucle.
    - Pasar dos veces por el bucle.
    - Hacer m pasos por el bucle, con m<n.
    - Hacer n-1, n y n+1 pasos por el bucle.
  - Bucles anidados: Para estos bucles se comienza la prueba por el bucle más interior.
    - Comenzar por el bucle más interior. Establecer o configurar los demás bucles con sus valores mínimos.

[5] Branch and Relational Operator.

- ❖ Llevar a cabo las pruebas de bucles simples para el bucle más interior, mientras se mantienen los parámetros de iteración (por ejemplo, contador del bucle) de los bucles externos en sus valores mínimos. Añadir otras pruebas para valores fuera de rango o excluidos.
- ❖ Progresar hacia fuera, llevando a cabo pruebas para el siguiente bucle, pero manteniendo todos los bucles externos en sus valores mínimos y los demás bucles anidados en sus valores típicos.
- ❖ Continuar hasta que se hayan probado todos los bucles.

➢ Bucles concatenados: Estos se pueden probar mediante el enfoque de los bucles simples si los bucles concatenados son independientes, en caso contrario, se usa el enfoque de los bucles anidados.

➢ Bucles no estructurados: Este tipo debe rediseñarse para que se ajusten a las construcciones de programación estructurada (3).

1.7.2. Técnica de caja negra:

Se centra en el análisis de las funcionalidades del sistema para el diseño de los casos de prueba, de modo que se verifiquen las entradas y salidas.

Existen varias técnicas de prueba de caja negra las cuales se mencionan a continuación:

- Método de prueba basado en grafos: Este tipo de prueba se basa en el análisis de las relaciones entre objetos del programa y su comportamiento. Se comienza con la creación de un grafo que modele estas relaciones definiéndose los nodos (objetos) y sus pesos (atributos), después se precisan los enlaces entre los nodos (relaciones) y sus pesos (describe alguna característica de un enlace), luego se diseñan casos de prueba con el fin de encontrar errores en las relaciones.
- Partición equivalente: Esta técnica utiliza las clases de equivalencia para el diseño de los casos de prueba, es decir, la entrada de una serie de datos válidos y no válidos, cubriendo en cada caso el máximo número de entradas.

  Pasos para identificar clases de equivalencia:

  ➢ Identificar el contenido de los datos de entrada del sistema.

- A partir de estos datos se definen las clases de equivalencia que tienen datos válidos y erróneos.

Algunas reglas para identificar las clases de equivalencia:

- Para cada rango de valores se especifica una clase válida y dos no válidas, es decir, la clase válida es el dato contenido en el rango de valores y las no válidas representan los datos que están fuera de ese rango.
- Si se especifica un número de valores, se crea una clase válida y dos no válidas, la clase válida la representa el valor correcto y las otras, números mayor y menor que dicho valor.
- Si una condición de entrada es lógica, se especifica una clase válida y otra no válida.
- Si se especifica un conjunto de valores admitidos y el programa trata de forma distinta cada uno de ellos, se especifica una clase válida por cada valor y otra no válida (3).

- Análisis de valores límites: Esta técnica complementa la partición equivalente pero no solo utiliza las condiciones de entrada sino también las de salida, selecciona, además, los casos de prueba en los extremos de la clase.

  Reglas para identificar casos:

  - Si una condición de entrada especifica un rango delimitado por los valores a y b, se deben generar casos para a y b y casos no válidos justo por debajo y justo por encima de a y b, respectivamente.
  - Si una condición de entrada especifica un número de valores, se deben desarrollar casos de prueba que ejerciten los valores máximo y mínimo, uno más el máximo y uno menos el mínimo.
  - Aplicar las directrices anteriores a las condiciones de salida.
  - Si las estructuras de datos internas tienen límites preestablecidos, hay que asegurarse de diseñar un caso de prueba que ejercite la estructura de datos en sus límites (3).

- Prueba mano a mano o de comparación: Esta técnica se utiliza en situaciones críticas en las que el software debe ser sumamente fiable. En estos casos se desarrolla una serie de versiones del programa siguiendo las mismas especificaciones a las cuales se les realiza pruebas con los mismos datos de entrada para asegurar la misma salida, en caso que se obtenga la misma salida, significa que todas las implementaciones son correctas, en caso contrario, es necesario revisar todas las versiones para encontrar la causa de las diferencias.
- Prueba de la tabla ortogonal: Esta técnica se aplica en pequeños dominios de entrada.

Las técnicas especificadas anteriormente son utilizadas para la realización de diversos tipos de pruebas en diferentes etapas del ciclo de desarrollo de software, a los cuales se hará referencia en el siguiente epígrafe.

1.8. Tipos de prueba.

1.8.1. Pruebas de Verificación.

- Pruebas Unitarias

Las pruebas unitarias se centran en verificar un componente software o módulo de un programa. Este tipo de prueba puede comprender desde un módulo hasta un grupo de módulos, incluso un programa completo y utiliza la técnica de caja blanca.

- Pruebas de Integración.

Estas pruebas tienen como objetivo comprobar el buen funcionamiento del programa completo con la integración de todos los módulos probados con las pruebas de unidad.

Existen varios tipos de integración, éstos son:

- Integración incremental: Se combina el módulo a probar con los que ya se han probado anteriormente. Esto posibilita que se puedan encontrar y corregir mejor los errores.
  - Integración ascendente: Comienza la integración de los módulos de los niveles más bajos de la estructura del programa hasta llegar al módulo principal del mismo. Primeramente se combinan los módulos en grupos que cumplan una función determinada, se crea para cada grupo, un módulo impulsor que es un programa que simula la llamada a los módulos, la entrada de datos de prueba y la recogida de los resultados. Luego de haber probado cada impulsor de los grupos, se eliminan y se sustituyen por los módulos de nivel superior de la jerarquía.
  - Integración descendente: Es el proceso inverso al de integración ascendente, se inicia con el módulo principal siguiendo una jerarquía de control hacia abajo de la forma “primero en profundidad” o “primero a lo ancho”, hasta llegar a la integración con los módulos de más bajo nivel.

- ❖ Integración mixta o sándwich: Este tipo de integración combina la integración ascendente con la descendente, utilizando la primera para los niveles más bajos del programa y la segunda para los niveles superiores.

Cada vez que se añade un nuevo módulo, el sistema cambia por lo que es necesaria la ejecución de un conjunto de pruebas que han sido realizadas anteriormente después de haber hecho cambios en el software para ver si las funcionalidades ya probadas siguen funcionando correctamente, a este proceso se le llama prueba de regresión. Esta prueba debe centrarse en los módulos críticos, es decir, en aquellos módulos de los cuales dependen varios requisitos del software, son más complejos y tienen un mayor nivel de control.

- ➢ Integración no incremental (big bang): Este tipo de integración se centra en probar el sistema en su conjunto después de haber probado cada módulo de forma independiente. Requiere menos tiempo de máquina para las pruebas y existe una posibilidad mayor de probar los módulos en paralelo.

1.8.2. Pruebas de Validación.

- Pruebas Funcionales

Estas pruebas son realizadas una vez que se hayan implementado las funcionalidades de un software determinado con el objetivo de ver la correspondencia con los requisitos funcionales previamente establecidos. Se apoya en la técnica de caja negra.

- Pruebas de aceptación.

Estas pruebas son realizadas por el usuario final con el objetivo de verificar el cumplimiento de los requisitos para determinar si el sistema está listo para ser usado finalmente en el entorno del usuario. Para ello se realizan dos tipos de pruebas de aceptación:

- ➢ Prueba alfa: Esta prueba la realiza el cliente en el ambiente de desarrollo para verificar el correcto funcionamiento del sistema junto con el desarrollador quien recoge cualquier error encontrado.
- ➢ Prueba beta: Esta prueba la realiza el cliente en el entorno final de producción, donde él se encarga de recoger todos los errores encontrados e informárselo después al desarrollador.

1.8.3. Pruebas de sistema.

Se prueba todo el sistema integrado de hardware y software para comprobar su correcto funcionamiento.

Para esto se realizan, entre otras:

- Prueba de recuperación: Con el objetivo de saber si los procedimientos de recuperación son los más apropiados para que no se pierdan los datos si el sistema falla, se fuerza al fallo para que los usuarios vuelvan a cargar y recuperar a partir de una copia de respaldo.
- Prueba de seguridad: Se llevan a cabo pruebas para comprobar que la política de seguridad impedirá accesos no autorizados al sistema que puedan poner en riesgo el mismo. Consiste en tratar de violar esta política intentando conseguir contraseñas, introducir virus, impedir el acceso al sistema del personal autorizado a usarlo, entre otros.
- Prueba de resistencia: Se generan condiciones anormales para ver cuánto puede soportar el sistema, es decir, lo ejecuta de forma que demande muchos recursos, continuas interrupciones, efectuar grandes búsquedas de datos residentes en disco, entre otros.
- Prueba de rendimiento: Se prueba el rendimiento del software en tiempo de ejecución.

Los tipos de prueba ya explicados se realizan para los métodos convencionales, pero con el surgimiento del paradigma orientado a objeto se han introducido nuevos conceptos que provocan diferentes problemas para la realización de las pruebas de software, no teniendo la cobertura necesaria para ellas. Las causas de esta situación se detallan en el siguiente epígrafe.

1.9. Aspectos a considerar para la prueba del Software Orientado a Objetos (POO).

Con el surgimiento de la Orientación a Objeto, la cual se puede considerar como “un marco para la ingeniería del software basado en objetos y clases” (3), surgieron nuevos conceptos importantes para la programación del software como por ejemplo los objetos y las clases, donde los objetos tienen sus propiedades (atributos) que los identifican y los hacen únicos, sus comportamiento (métodos) y se encargan de modelar el mundo real; las clases son definiciones de las propiedades y comportamiento de un tipo de objeto concreto y la instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas, surgieron, además, la encapsulación, la abstracción, la herencia y el polimorfismo. Estos

conceptos resultan muy ventajosos para la programación, no siendo así para las pruebas de software ya que estas características dificultan el trabajo en esta fase y mucho más si los sistemas son complejos.

El encapsulamiento permite que toda la información de una clase quede oculta bajo el nombre del objeto con el cual se va a referenciar dicha clase, y la abstracción permite que se definan algunos métodos y el tipo de datos que devuelve, para luego programarlos en la subclase que vaya a utilizar dichos métodos, estos conceptos dan lugar a que la información de la clase quede oculta al ingeniero de pruebas, lo cual es una desventaja ya que le imposibilita ver con claridad las violaciones de requisitos que se pueden cometer en la programación, por lo que dificulta la eficiencia de las pruebas de software pues la información de los objetos son necesarias para la realización de las mismas, entonces hay que usar otras técnicas que hacen el trabajo más engorroso como por ejemplo estas clases pueden proporcionar operaciones incorporadas para conocer los valores de los atributos, pero una imagen instantánea del estado del objeto puede ser difícil de adquirir. Además las clases abstractas no se pueden probar ya que éstas no pueden ser instanciadas, sólo se pueden probar las instancias concretas de una clase abstracta. (9).

La herencia es otra de las características del software actual, es una de las diferencias claves con respecto a los productos convencionales, permiten que las subclases importen un comportamiento especializado a partir de una base común de elementos proporcionados por las superclases, lo cual agrava el problema del encapsulamiento ya que una subclase puede heredar el comportamiento de una superclase sin detectar que ésta incumple los requisitos que fueron definidos.

La herencia propicia en cierta medida el reuso, pues se puede reutilizar el código de las superclases repetidas veces, esto no implica que este código de la subclase esté exento de errores, el reuso genera más pruebas pues no se puede garantizar que el código que se está reutilizando no tenga errores porque no es posible determinar que todos los caminos posibles hayan sido probados con anterioridad. Además, la herencia múltiple complica mucho más las pruebas incrementando el número de contexto para los que se requiere la prueba. Si la subclase instanciada de una superclase se utiliza dentro del mismo dominio del problema, es probable que el conjunto de casos de prueba derivados de la superclase pueda usarse para las pruebas de la subclase, pero si la subclase se usa en un contexto enteramente diferente, los casos de pruebas de la superclase tienen muy poca probabilidad de ser aplicados, por lo que se tendrá que realizar un nuevo conjunto de pruebas.

Polimorfismo: En términos generales, “capacidad de una entidad de tener varias formas según el contexto. Específicamente en la POO, habilidad de definir un método en clases distintas según sus necesidades” (3)

El Polimorfismo es una característica que reduce en gran medida el esfuerzo necesario para extender un sistema OO. Según dice Alejandro Teruel (10) “cada vez que se instancia un objeto en los métodos de las clases, como resultado del polimorfismo debe realizarse una prueba separada, lo que aumenta el tiempo de realización de las mismas, el costo y el esfuerzo”.

El manejo de excepciones es propenso a errores, por lo cual es necesario obligar su ejecución para de esta forma probarlo aunque es un elemento difícilmente controlable y forzarlo puede ser difícil.

Estos son los aspectos más significativos encontrados en bibliografías, pero existen otros más como las invocaciones implícitas, el paso de mensajes que provocan que el objeto receptor se comporte de una forma determinada cuando se ejecuta una operación, este es un punto fundamental para realizar las pruebas.

Estos problemas han dado paso a la definición de nuevas estrategias para los tipos de pruebas de software, a las cuales se hace referencia en el siguiente epígrafe.

1.10. Estrategias para las pruebas de software OO.

Como se ha dicho anteriormente todo sistema informático debe ser probado antes de ser entregado al usuario final con el objetivo de garantizar su calidad, para ello se ha adaptado los métodos convencionales considerando las características del software orientado a objeto.

La estrategia clásica para las pruebas de software comienza con probar de lo pequeño hasta lo más grande, por lo que se comienza con la prueba de unidad, luego se progresa con la prueba de integración y termina con la prueba de validación del sistema.

1.10.1. Pruebas de Unidad en el contexto OO

En los sistemas tradicionales, las pruebas de unidad se basan en probar las unidades de programas compilables más pequeñas (por ejemplo módulos, subrutinas, procedimientos, componentes), cuando se habla de software orientado a objeto la unidad para las pruebas cambia, ya que todo queda encapsulado en las clases, convirtiéndose así en la menor unidad comprobable, por lo que la prueba está

fundamentalmente dirigida a probar las clases y el estado y comportamiento de los objetos, es decir se encarga de verificar que esta unidad trabaje correctamente de forma independiente antes de realizar la integración del sistema.

1.10.2. Pruebas de integración en el contexto orientado a objeto.

El objetivo principal de las pruebas de integración es detectar las fallas de interacción entre las distintas clases que componen al sistema.

En este tipo de pruebas los métodos convencionales de integración descendente (top-down) y ascendentes (bootom-up) no son de mucha utilidad ya que el software orientado a objeto no tiene una estructura jerárquica, por lo que se utilizan dos estrategias diferentes.

La primera es la prueba basada en hilos el cual menciona que “la funcionalidad de esta prueba es integrar el conjunto de clases requeridas, para responder a una entrada o un suceso al sistema donde se crean casos de prueba para cada clase en la unidad, con lo cual un hilo de este conjunto se ejercita. Cada hilo se integra y se prueba individualmente” (3).

La segunda es la integración por dependencia, se comienza integrando el sistema con las clases independientes y luego con las dependientes hasta haber integrado el sistema completo.

1.10.3. Pruebas de sistema para software orientado a objeto.

A nivel de sistemas, al igual que en el software tradicional, la validación del software orientado a objeto se concentra en las entradas y salidas reconocidas desde el sistema que son visibles para el usuario.

Debido a los cambios que se introdujeron con la orientación a objeto, el aumento de los tamaños de los productos de software y una alta demanda de software con calidad, han surgido diferentes metodologías que solucionan estos problemas para software orientado a objetos.

1.11. Metodologías.

El uso de las metodologías constituye un paso significativo para la elaboración de un proyecto, permite además, definir una guía para desarrollar un proceso de software con un nivel de calidad elevado. A lo largo de la historia han surgido diversas metodologías con diferentes enfoques: metodologías tradicionales como RUP y metodologías ágiles como XP que responden a dichos propósitos.

1.11.1. Proceso Unificado

Rational Unified Process (RUP) es una metodología que se centra en controlar de forma rigurosa el proceso de desarrollo de software. Define dicho proceso como iterativo e incremental, las iteraciones se basan en los pasos de los flujos de trabajo y los incrementos, en versiones incrementales que se acercan al producto terminado. Está dirigido por casos de uso ya que éstos sirven para especificar los requisitos del sistema, guían el diseño, la implementación y las pruebas. Está centrado en la arquitectura para tener una clara perspectiva del sistema y de sus partes más relevantes.

RUP divide el proceso en cuatro fases: Inicio, Elaboración, Construcción y Transición, las cuales guían cada flujo de trabajo (secuencia de actividades que se realizan y se obtiene un resultado de valor observable para un actor individual o del negocio): Modelamiento del negocio, Requerimientos, Análisis y Diseño, Implementación, Pruebas, Instalación, Administración de proyecto, Gestión de Configuración y Cambios y Ambiente.

Ciclo de vida de pruebas

Las pruebas en RUP están enfocadas a la evaluación y determinación del grado de calidad del producto.

Durante la fase de inicio, se pueden planificar inicialmente las pruebas a ejecutar a partir de la especificación de casos de uso del sistema que se define en el flujo de trabajo de requerimientos. En la fase de elaboración, el análisis y diseño del software posibilitan la determinación, de forma más específica, de las pruebas a realizar. En esta fase se prueba la línea ejecutable de la arquitectura una vez que se hayan implementado las primeras funcionalidades, a las cuales se les realizan pruebas unitarias y funcionales. En la construcción, se concluye la implementación de todas las funcionalidades que debe tener el software a las cuales también se le hacen pruebas unitarias. De esta forma se obtiene un producto listo para su utilización, el cual es sometido a pruebas de integración, de sistema y funcionales.

En la fase de Transición se realizan pruebas de aceptación del cliente. En estas fases se deben corregir los defectos encontrados gestionando los cambios y realizando pruebas de regresión.

En RUP se definen diferentes niveles de pruebas yendo desde la menor unidad del programa con la realización de pruebas unitarias, haciendo pruebas de integración luego de integrar cada unidad probada, después la ejecución de pruebas de sistema y por último pruebas de validación.

Esta metodología aplica varias de las mejores prácticas en el desarrollo de software, entre otras, la verificación de la calidad. Para evaluar la misma se tienen en cuenta los atributos de calidad de Hewlett-Packard, conocidos con el acrónimo de FURPS, éstos son: funcionalidad, usabilidad, fiabilidad, rendimiento y soporte. Para cada uno de ellos, se ejecutan o implementan una serie de pruebas en diferentes niveles, tal como se muestra a continuación:

Atributos de Calidad	**Tipos de Prueba**
Funcionalidad	Funcionales, Seguridad, Volumen
Usabilidad	Usabilidad
Fiabilidad	Integridad, Estructura, Resistencia
Rendimiento	Estándar de comparación, Contención, Carga, Perfil de Rendimiento
Soporte	Configuración, Instalación

Para medir la completitud de las pruebas de software, en RUP se utilizan métricas de cobertura expresadas en la cobertura de pruebas basadas en los requerimientos y casos de prueba analizando las pruebas planeadas, implementadas, ejecutadas y satisfactorias con respecto al número de requerimientos definidos, y cobertura basada en la ejecución del código, verificando qué cantidad de líneas de código han sido ejecutadas para las pruebas mediante las coberturas de código para flujos de control, es decir se prueban líneas de código, declaraciones, caminos y otros elementos y para flujos de datos donde se prueba, por ejemplo, que los datos hayan sido definidos antes de ser usados.

RUP presenta dificultades con las restricciones de tiempo y esfuerzo por lo que no resulta ser el más adecuado para muchos de los proyectos actuales donde el entorno del sistema es muy cambiante, y en

donde se exige reducir drásticamente los tiempos de desarrollo manteniendo una alta calidad. Se genera gran cantidad de documentación como resultado de la planificación en cada flujo de trabajo.

1.11.2. Metodología Ágil: XP

Extreme Programing (XP) es una metodología ágil pensada para ser utilizada en proyectos cortos con requerimientos muy cambiantes, es una de las metodologías de desarrollo de software más exitosas en este sentido. Dedica escaso tiempo a la preparación y planificación, e invierte la mayor parte del tiempo en el desarrollo, probando y rectificando hasta conseguir el resultado adecuado. Un desarrollo mediante programación extrema está compuesto por una serie de iteraciones cortas. En XP no existe una fase de requisitos, en su lugar, al comienzo de cada iteración, se lleva a cabo el juego de la planificación.

Juego de la planificación: Se establece una comunicación frecuente del cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.

Historias de Usuario: Son tarjetas de papel usadas con el objetivo de describir los requisitos del software, tanto funcionales como no funcionales. Con esta técnica los programadores pueden realizar la implementación en pocas semanas.

Uno de los pilares de esta metodología es el proceso de pruebas. Los roles referentes a este proceso son:

- Programador: Produce el código del sistema y es el responsable de llevar a cabo las pruebas unitarias.
- Cliente: Realiza las pruebas funcionales, escribe las historias de usuario y les asigna prioridad de acuerdo a su importancia dentro del negocio.
- Encargado de pruebas (Tester): Escribe junto al cliente las pruebas funcionales. Ejecuta las pruebas recogiendo luego los resultados para que sean vistos por el personal de desarrollo y es responsable de las herramientas de soporte para pruebas.

Pruebas del sistema

XP divide las pruebas del sistema en dos grupos:

- Pruebas unitarias
- Pruebas de aceptación o pruebas funcionales

Pruebas de Aceptación

Las pruebas de aceptación son más importantes que las pruebas unitarias dado que significan la satisfacción del cliente con el producto desarrollado y el final de una iteración y el comienzo de la siguiente.

Sin embargo, el cliente no tiene la formación adecuada para desarrollar buenas pruebas de aceptación. Por ejemplo, el cliente, en la mayoría de los casos sabe que es lo que quiere que haga la aplicación de forma correcta, pero puede no ser capaz de desarrollar un conjunto de pruebas que garantice la total cobertura de la funcionalidad especificada en la historia de uso, limitándose a probar que el sistema hace lo que debe sin verificar todas sus variantes.

Existen en la actualidad muchas herramientas para desarrollar pruebas de aceptación, lo suficientemente sencillas para que un cliente pueda manejarlas. Sin embargo estas herramientas son inútiles si el cliente es incapaz de diseñar un conjunto completo de pruebas que verifiquen toda la funcionalidad del sistema y no solo una parte o con unos valores concretos.

1.12. Conclusiones

En este capítulo se ha descrito el proceso de pruebas de software definiendo importantes conceptos para su mejor entendimiento y su estado del arte a nivel internacional, nacional y en la UCI, se han explicado los niveles de pruebas y las técnicas tradicionales usadas, así como las consecuencias que provoca la realización de software orientado a objeto en la disciplina de pruebas, y se han explicado diferentes metodologías que pueden ser usadas para controlar y mejorar el proceso de desarrollo del software y con esto aplicar un adecuado proceso de pruebas. De acuerdo a este estudio se puede llegar a las siguientes conclusiones:

- En la actualidad se ha visto un interés creciente de las empresas existentes en la producción de software con calidad y con esto a la realización de las pruebas a sus productos, como uno de los requisitos básicos para garantizar la mejora continua de la calidad, ya que cada vez los clientes son

más exigentes en este sentido, esto es un gran paso de avance ya que anteriormente se producían software que sólo funcionaran sin tener en cuenta los errores que tuviesen.

- Se considera que en Cuba, según la encuesta realizada en algunas empresas de desarrollo de software, se le concede gran importancia a las pruebas pero no en todas existe un proceso bien definido. En esta disciplina influye el uso de metodologías para tener un mayor control de las actividades que se realizan dentro del proceso de desarrollo. Las empresas cubanas han visto la gran utilidad de las metodologías pero no todas las usan de forma apropiada ya que inician las pruebas al final del proceso de desarrollo, lo cual provoca atrasos en la entrega del producto final, aumento del esfuerzo y costos del proyecto, la insatisfacción del cliente, produciendo de esta forma, un software de baja calidad. La mayoría de las empresas no cuenta con un equipo de pruebas al que se le asigne un rol específico que pudiera facilitar la detección de mayor cantidad de errores. De acuerdo a las características de cada proyecto, en muchas ocasiones no es necesario realizar todas las actividades del proceso ni todos los tipos de prueba, de lo cual depende la estrategia a usar. La mayoría de las empresas usan plantillas y documentan los defectos, lo cual resulta muy importante para la estandarización de la información y el posterior entendimiento de los defectos por parte de los desarrolladores para que puedan corregirlos con mayor facilidad. Un alto por ciento de estas empresas no usa herramientas para la automatización de dicho proceso que pudiera proporcionar mayor rapidez en la realización del mismo.
- Se piensa además, que es importante que las personas que se vinculen a dicho proceso deban ser especialistas en el tema, y que además conozcan bien el negocio para evitar que se prolongue la realización de las pruebas debido a que hay que capacitar a aquellos que no sean del proyecto para que tengan una noción general del negocio.
- Se usará RUP como metodología para el desarrollo del software orientado a objeto ya que su utilización es factible en proyectos como éste, de gran tamaño. Los requisitos funcionales determinan la funcionalidad del sistema mediante la elaboración de casos de uso, lo cual facilita el diseño de casos de prueba y automatización de las mismas; es iterativo e incremental lo que posibilita que las pruebas no se apliquen al final del proyecto ya que en cada iteración puede ser aplicado al menos un ciclo de pruebas, tiene el proceso de pruebas bien definido.

- En el proceso de pruebas definido por RUP se generan una gran cantidad de artefactos. Para el módulo sólo se tendrán en cuenta los siguientes, que a consideración de las autoras son los más importantes pues son determinantes en cada una de las actividades del proceso: el plan de pruebas, configuración del entorno de pruebas, datos de prueba, modelo de pruebas y sumario de evaluación de las pruebas.

Capítulo 2: Planificación del Proceso de Pruebas de Software.

2.1. Introducción

Este capítulo se centra en la planificación de las pruebas de la cual se generan dos artefactos: el plan de pruebas general y el entorno de configuración de las mismas los cuales dan paso a un efectivo diseño, ejecución y evaluación de los resultados de las pruebas. Para la obtención del plan de pruebas se parte del plan de iteraciones del proyecto, ya que se utiliza como guía para definir los ciclos de pruebas por las iteraciones planificadas y los objetivos que debe tener cada uno de ellos, en el proyecto no se elaboró de manera formal el plan de iteraciones por lo cual la planificación de las pruebas se ha efectuado de acuerdo a la visión que se tiene de cómo se ha ido realizando el software desde sus inicios.

La planificación del proceso de pruebas es una actividad de suma importancia, ya que en ella se consideran todas las pruebas posibles para detectar la mayor cantidad de defectos en el producto, teniendo en cuenta los recursos disponibles, este último aspecto es crítico ya que en muchas ocasiones no se disponen de muchos recursos para el flujo de trabajo de pruebas, por lo cual es indispensable la tarea de planificación para administrarlos de forma eficiente. Además en esta actividad se define el esfuerzo que se va a emplear en las pruebas, tiempo de duración y cantidad de personas involucradas dependiendo de las características y la complejidad del software que se desea probar, por lo cual es necesario realizar un breve descripción del producto como primer paso para la comprensión de los artefactos que se desarrollan como resultado de esta tarea.

2.2. Descripción del producto

En Venezuela se han detectado problemas para la realización de las tareas que se llevan a cabo en los Registros y Notarías, para lo cual se ha buscado como solución a los mismos el desarrollo de un sistema que sea capaz de estandarizar la gestión de las oficinas de Registros y Notarías, para garantizar la certeza, confiabilidad y seguridad jurídica de los actos protocolizables de conformidad con las disposiciones legales que los regulan, así como ejercer el control y la debida supervisión que permita

brindar servicios eficaces, y lograr equitativos ingresos y la redistribución de los excedentes en función de planes de beneficio social.

Uno de los módulos del producto, Servicio Autónomo, es una aplicación Web sumamente importante para los registros mercantiles e inmobiliarios en Venezuela ya que controla todos los movimientos que se llevan a cabo dentro de los mismos por lo que debe tener un alto nivel de seguridad y ser fiable. Esta aplicación está compuesta por subsistemas:

- Subsistema oficina: Se encarga de la creación, búsqueda y modificación de las oficinas creadas, permite mostrar los datos de las oficinas, así como cambiar el estado de activación a una determinada oficina, configurar puntos para las aplicaciones locales en los diferentes registros existentes (Inmobiliarios, Mercantiles o de Servicio Autónomo), creación, modificación y configuración de roles de usuarios ya existentes para una oficina determinada, configurar los nomencladores que serán usados por los diferentes módulos, además se muestra un historial de los usuarios creados anteriormente y sus modificaciones, un historial de las oficinas, configura la jurisdicción de cada una de las oficinas, realiza reportes de usuarios y búsquedas de personas.
- Subsistema documento jurídico: Permite insertar y/o levantar medidas de prohibición ya insertadas a los documentos que se consultarán en las oficinas a la hora de realizar los actos, adicionar y/o anular exenciones, insertar una disposición legal, permite también adicionar cualquier tipo de documentos y se pueden observar todos los documentos insertados anteriormente.
- Subsistema de parametrización: Permite eliminar, insertar y modificar plantillas de actos mercantiles e inmobiliarios y estos también se pueden ordenar, conceptos de pagos que luego serán adicionados a los actos, plantillas de conceptos de pagos donde se definen pasos y en ellos se concretan sus entes recaudadores y a cada paso se le agregan conceptos de pagos, además se pueden agregar, modificar y eliminar recaudos para los registros mercantiles e inmobiliarios y los elementos que luego serán asociados a estos recaudos. Se puede observar e imprimir de acuerdo a una fecha determinada los reportes de los actos de los registros inmobiliarios y mercantiles.
- Subsistema seguridad: Permite entrar a la aplicación con un usuario y contraseña designado, además se puede cambiar la contraseña de un usuario determinado.
- Subsistema de búsquedas: Permite buscar personas, inmuebles y compañías de acuerdo a un criterio especificado.

Luego de una comprensión del producto se puede dar paso al desarrollo de los artefactos de planificación.

2.3. Artefacto: Plan de Pruebas

El plan de pruebas da una visión global del proceso de pruebas y constituye un resultado de la planificación de las mismas, teniendo en cuenta el alcance de la prueba, los requerimientos que se van a probar, tanto funcionales como no funcionales, la estrategia a seguir, las herramientas necesarias para facilitar la realización del proceso de pruebas, los recursos humanos y del sistema, el cronograma y los entregables.

2.3.1. Alcance

Se proponen las siguientes etapas para la realización de las pruebas al módulo Servicio Autónomo, las cuales se ejecutan de acuerdo al número que tienen, ya que se probará desde la parte más pequeña del sistema hasta probar el software completo.

1. Etapa I: Pruebas de unidad con la técnica de caja blanca: Consiste en probar los subsistemas de forma independiente.
2. Etapa II: Pruebas de integración: Consiste en probar los subsistemas de forma integrada, se van integrando los subsistemas independientes y luego los que dependen de ellos para probar que funcionan correctamente en conjunto.
3. Etapa III: Pruebas funcionales con la técnica de caja negra: Consiste en probar que el software cumple con las funcionalidades establecidas por los requisitos funcionales. Revisión de los documentos de especificación de casos de usos de sistema, manual de usuario (si el ciclo de pruebas es de la fase de construcción) contra la aplicación.
4. Etapa IV: Pruebas de sistema: Se centra en probar los requisitos no funcionales del producto.
5. Etapa V: Pruebas de Regresión: Consiste en la revisión de la aplicación para detectar la eliminación de errores en ciclos de pruebas anteriores y examinar la no introducción de otros nuevos.

Se deben efectuar las etapas en orden ascendente es decir desde la etapa I hasta la etapa IV y la etapa V se realiza luego de la corrección de defectos de cada una de las etapas ejecutadas anteriormente.

Los siguientes diagramas muestran las etapas con los escenarios de pruebas:

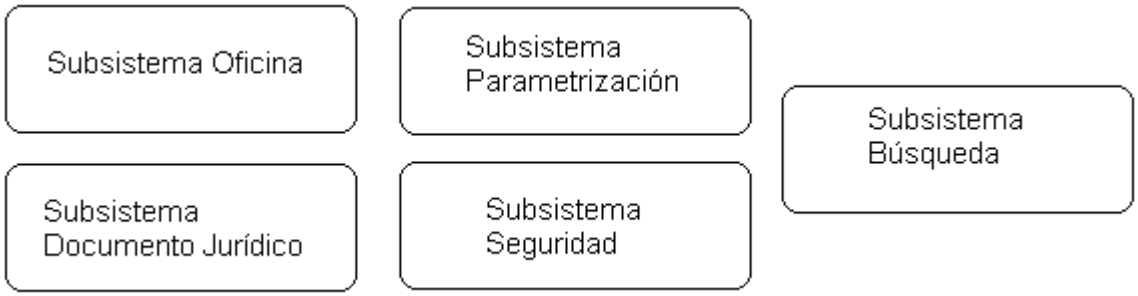

Figura 1 Submódulos para la Etapa I: Prueba de unidad

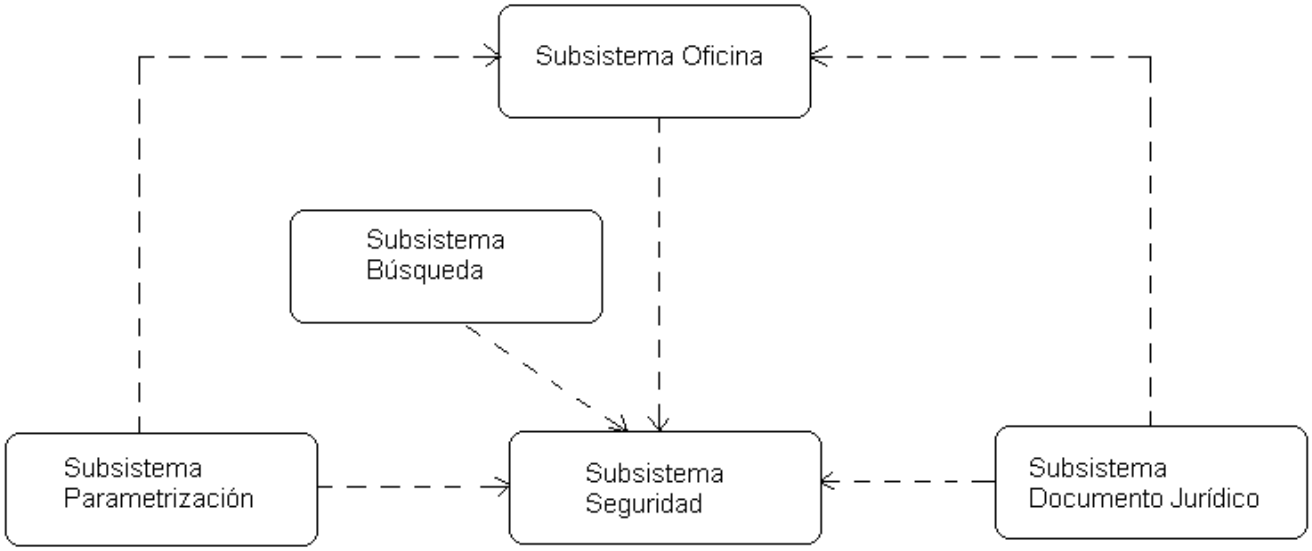

Figura 2 Integración por submódulos para la Etapa II

Lista de Riesgos

Riesgo	Estrategia de Mitigación	Plan de Contingencia
1. No contar con una planificación adecuada para el proceso de pruebas. 2. No disponer de los recursos necesarios	1. Elaborar un plan y estrategia de pruebas. 2. Configurar el entorno de pruebas para la estrategia definida. 3. Realizar cursos de	1. Establecer una nueva estrategia para la realización de las pruebas. 2. Realizar las pruebas definidas en el ambiente de desarrollo.

de hardware para la prueba. 3. Falta de conocimiento del personal. **4.** No contar con un repositorio para almacenar los artefactos que se generan dentro del proceso de pruebas. **5.** Afectaciones con el fluido eléctrico. **6.** Retrasos de los entregables requeridos para el proceso de pruebas. **7.** Cambios en miembros del equipo de pruebas. **8.** Dificultad en obtener los datos de pruebas. **9.** Inadecuada gestión de cambios. 10. Cambio en las funcionalidades de alto impacto del producto.	capacitación y pruebas para la incorporación de nuevos miembros al equipo antes de integrarlos al proyecto. 4. Crear un repositorio con las políticas de acceso bien definidas. 5. Contar con una UPS para las afectaciones eléctricas. 6. Revisar el estado de los entregables cada cierto tiempo para que se cumplan con los cronogramas establecidos. 7. Motivar a los miembros del equipo de pruebas. 8. Recopilar toda la información necesaria para la obtención de los datos de pruebas. 9. Verificar que el proceso de gestión de cambios se realice de forma correcta. 10. Realización de las revisiones técnicas formales a los requisitos funcionales.	3. Realizar cursos intensivos de capacitación. 4. Crear un repositorio con las políticas de acceso bien definidas. 5. Trabajar más horas para tratar de realizar la tarea en el tiempo previsto. 6. Definir nuevas fechas de entrega e ir revisando el estado de los entregables para que se cumplan las nuevas fechas establecidas. 7. Realizar cursos de capacitación para los nuevos integrantes del equipo de pruebas. 8. Recopilar toda la información necesaria para la obtención de los datos de pruebas. 9. Realizar las pruebas de regresión. 10. Reajuste del plan y el cronograma de las pruebas.

2.3.2. Referencias

- Documento Visión
- Especificaciones de casos de uso del sistema

2.3.3. Procedimiento para la modificación del plan

Al comenzar la jornada laboral se realiza una breve reunión con el equipo de pruebas para revisar el cumplimiento del plan en la jornada del día anterior y exponer los percances que imposibilitaron el cumplimiento del mismo. Se reajusta el cronograma de las pruebas en caso de ser necesario.

En el inicio de la fase de construcción se realiza una reunión con el Jefe de proyecto, jefe de módulo, Jefe de calidad interna y Jefe del equipo de pruebas del módulo para la analizar si habrá modificación en la misión de evaluación de acuerdo con lo cual se modifica el plan, se incorporan las nuevas funcionalidades implementadas y se ajusta el cronograma.

Las propuestas de modificaciones del plan serán presentadas para su análisis en reunión de chequeo con el Jefe de proyecto, Jefe de módulo, Jefe de calidad interna y un miembro del equipo de Calisoft para su aprobación.

En caso de haber modificaciones en el cronograma de pruebas se realizará un documento con las mismas, que será distribuido entre los participantes de las pruebas para hacer oficial los cambios.

2.3.4. Requisitos de Prueba

Nivel de Validación

Prueba de Funcionalidad: Funcionales

Caso de uso: Administrar usuarios

- Crear un nuevo usuario en la oficina.
- Establecer el estado del usuario:
  - Activo (es el estado inicial de un usuario al crearse).
  - Inactivo (al inactivar un usuario este no puede volver al estado Activo nunca más).
  - Suspendido (estado temporal).
- Realizar búsqueda de oficinas por cualquiera de los siguientes parámetros:
  - Código.
  - Municipio.
  - Estado.
  - Tipo de oficina.
- Mostrar listado de oficinas encontradas, indicando el nombre de éstas.
- Permitir realizar búsquedas de personas por los siguientes criterios:

  - Primer Nombre.
  - Segundo Nombre (si lo tiene).
  - Primer apellido.
  - Segundo Apellido.
  - Letra de cédula.
  - Número de cédula.
- Mostrar personas encontradas en la búsqueda mostrando:
  - Primer Nombre.
  - Segundo Nombre (si lo tiene).
  - Primer apellido.
  - Segundo Apellido.
  - Letra de cédula.
  - Número de cédula.
  - Sexo.
  - Estado civil.

Caso de uso: Búsqueda de Personas.

- Realizar la búsqueda de personas.
- Visualizar los detalles de la persona seleccionada.
- Visualizar las compañías e inmuebles relacionados con la persona seleccionada.

Caso de uso: Ver Reporte de Oficina.

- Listar las oficinas cumpliendo con la búsqueda efectuada.
- Mostrar los reportes.

Caso de uso: Crear Oficina

- Permitir crear una oficina con los siguientes datos:
  - Tipo de Oficina (Requerido).

- Estado (Requerido).
- Municipio (Requerido).
- Jurisdicción (No requerido).
- Nombre de la oficina (Requerido).
- Código de la oficina (Requerido).
- RIF.
- Categoría.

- Permitir guardar los datos del Documento Jurídico que avala la creación de la misma.
  - Institución que emite el documento (Requerido).
  - Tipo de documento que publica (Requerido).
  - Título del documento (Requerido).
  - Tipo de documento (Requerido).
  - Observaciones del documento (Requerido).
  - Número del documento (No requerido).
  - Fecha de emisión del documento (No requerido).
- Permitir que el documento jurídico almacene la fecha en que fue insertado a la base de datos (fecha actual).

Caso de uso: Configurar Documentos Jurídicos (Medidas de Prohibición).

- Registrar en el sistema los documentos jurídicos.

Caso de Uso: Actualizar Datos de la Oficina

- Permitir actualizar los siguientes datos de la oficina:
  - Nombre (Requerido).
  - Dirección.
  - Teléfono.
  - Fax.
  - RIF.

- Categoría.

Caso de Uso: Administrar Oficinas

- Permitir cambiar el estado de activación de las oficinas.
- Permitir guardar los datos del documento jurídico que avale la acción de cambiar el estado de activación de estas.
- Permitir migrar los datos de la oficina cerrada a otra con el mismo tipo de registro.

Caso de Uso: Configurar Plantillas de Actos

- El sistema debe permitir configurar plantillas de actos.

Caso de Uso: Búsqueda de Personas Jurídicas

- Debe permitir realizar la búsqueda de compañías.
- Debe visualizar los detalles de la compañía seleccionada.
- Debe visualizar las personas e inmuebles relacionados con la compañía seleccionada.

Caso de Uso: Búsqueda de Inmuebles

- Debe permitir realizar la búsqueda de inmuebles.
- Debe visualizar los detalles del inmueble seleccionado.
- Debe visualizar las compañías y personas relacionadas con el inmueble seleccionado.

Nivel de Sistema

Prueba de Funcionalidad: Seguridad

- Verificar que solo el usuario de Servicio Autónomo: saren, sea el que pueda acceder al sistema.

Prueba de Rendimiento: Carga

Estas pruebas no se llevarán a cabo ya que el propósito es que dicha aplicación sea usada siempre por pocos usuarios.

Prueba de Funcionalidad: Volumen

Estas pruebas no se llevarán a cabo debido a que la base de datos tiene un gran volumen de almacenamiento y para ponerla al menos al 90% de su capacidad deben pasar unos cuantos años.

Prueba de Fiabilidad: Usabilidad

- Verificar que las interfaces sean de fácil acceso para los usuarios finales.
- Verificar que las interfaces se correspondan con los estándares web con el uso de la herramienta W3C Validator 0.7.4.
- Verificar que haya sido elaborada la ayuda en línea y que describa con claridad las instrucciones a seguir para trabajar con el sistema.
- Comprobar que las interfaces sean amigables para el usuario teniendo en cuenta que:
  - Las ventanas del sistema contengan bien estructurados los datos, y de esta forma permitan la interpretación correcta e inequívoca de la información.
  - La interfaz contenga teclas de función, teclas de atajo y menús desplegables que faciliten y aceleren su utilización.
  - El diseño de la interfaz de usuario busque la ejecución de acciones de una manera rápida, minimizando los pasos a dar en cada proceso.
  - El sistema use una norma que permita la distinción visual entre los elementos de la ventana a través del uso de colores, así como otras técnicas.
  - La corrección de errores de introducción de datos sea clara y fácil de realizar.
  - Todos los textos y mensajes en pantalla aparezcan en idioma castellano.
  - Su funcionamiento sea intuitivo y que requiera de información mínima.

Prueba de Fiabilidad: Integridad

- Verificar que los datos de la oficina se hayan insertado correctamente.
- Verificar que una misma oficina no sea creada más de una vez.
- Verificar que las modificaciones de los datos de las oficinas sean guardadas correctamente.

- Verificar que un mismo usuario no sea creado más de una vez.
- Verificar que las modificaciones de los datos de los usuarios sean guardadas correctamente.
- Verificar que las plantillas de actos mercantiles sean creadas correctamente.
- Verificar que las modificaciones de las plantillas de los actos mercantiles sean guardadas correctamente.

Prueba de Fiabilidad: Estructura

- Verificar que se muestre el contenido apropiado en cada página.
- Verificar que no haya vínculos rotos, perdidos o incorrectos en la aplicación utilizando el W3C Validator 0.7.4.

Prueba de Fiabilidad: Resistencia (stress)

- Verificar el funcionamiento del sistema con 2G de memoria RAM en el servidor web.

Prueba de Soporte: Configuración

- Verificar el funcionamiento del sistema con el navegador Microsoft Internet Explorer 5 o superior.
- Verificar que el sistema corra con el sistema operativo RetHat Enterprise Linux Advanced Server 4.0
- Verificar que el sistema sea compatible con el Entorno de Ejecución de Java (JRE) 1.05.0_06

Prueba de Soporte: Instalación

En Máquina Cliente:

- Verificar la instalación del sistema operativo Windows XP SP 2
- Verificar la instalación del navegador Microsoft Internet Explorer 5 o superior.
- Verificar la instalación de la impresora y sus drivers.
- Verificar la instalación de los drivers del scanner.

En Servidor Web:

- Verificar la instalación del sistema operativo RetHat Enterprise Linux Advanced Server 4.0
- Verificar la instalación y configuración del Apache Tomcat 0.27
- Verificar la instalación del Entorno de Ejecución de Java (JRE) 1.05.0_06 o superior.

En Servidor de Base de Datos:

- Verificar la instalación del sistema operativo RetHat Enterprise Linux Advanced Server 4.0
- Verificar la instalación del Sistema Gestor de Base de Datos Oracle 10g.

2.3.5. Estrategia de Prueba

Esta estrategia precisa cómo conducir las pruebas de forma apropiada, especificando sus niveles, los tipos que se llevan a cabo dentro de cada nivel, con sus objetivos, técnicas y herramientas que las soporten, criterios de completitud y algunas consideraciones especiales. Se precisa, además, el criterio de evaluación que se va a utilizar.

2.3.5.1. Descripción del procedimiento interno de corrección de defectos.

El procedimiento comienza con la ejecución de las pruebas de unidad e integración (etapa I y II) por parte de los desarrolladores, en estas etapas los probadores son distribuidos con los desarrolladores para apoyar la elaboración de las no conformidades de acuerdo a los defectos encontrados, en las demás etapas los probadores son los responsables de la ejecución de las pruebas.

Cuando se obtienen las no conformidades en cada una de las etapas de pruebas, éstas pasan por un proceso de aprobación, donde se tiene en cuenta la calidad de la redacción de manera que sean entendibles para los desarrolladores, que no estén repetidas, luego se clasifican y se les otorga un número de acuerdo a la prioridad que tenga. Finalmente se obtiene una lista de no conformidades aprobadas por los jefes de proyecto, de módulo y de calidad interna y se guardan las no conformidades en un repositorio.

Luego las no conformidades aprobadas son clasificadas en solicitud de cambios o en notificación de cambio, en caso de que sea la última opción el Jefe de módulo asigna las no conformidades a los miembros del equipo de desarrollo, éstas se ubican en el TRAC[6] con el miembro del equipo al que le corresponde y el estado de la misma (abierta, cerrada), se realiza un cronograma para la corrección de los defectos y se establecen las fechas para las próximas pruebas.

Si la no conformidad se clasifica como solicitud de cambio se procede a la solicitud del cambio en un proceso formal.

Una vez se haya verificado la correcta depuración, el software junto con la documentación necesaria es entregada al equipo de Calisoft para la realización de las pruebas de liberación.

Las no conformidades son redactadas en la plantilla definida por Calisoft.

2.3.5.2. Tipos de Prueba

Nivel de Unidad

Objetivos	Verificar que la menor unidad del sistema esté programada correctamente de forma independiente.

[6] Sitio web utilizado para la planificación del proyecto.

Técnica	Utilizar la técnica de caja blanca haciendo uso de la prueba del camino básico para el diseño de los casos de pruebas y la herramienta JUnit para la ejecución de las pruebas.
Criterio de completitud	Las pruebas planeadas han sido ejecutadas. Todos los errores identificados han sido corregidos.
Consideraciones especiales	

Nivel de Integración

Objetivos	Verificar que la menor unidad del sistema funcione correctamente junto a todo el sistema integrado.
Técnica	Utilizar caja negra haciendo uso de la prueba de partición equivalente y análisis de los valores limites.
Criterio de completitud	Las pruebas planeadas han sido ejecutadas. Todos los defectos identificados han sido corregidos.
Consideraciones especiales	

Nivel de Validación

Pruebas funcionales

Objetivos	Validar que el sistema se corresponda con los requisitos funcionales definidos, la entrada y salida de datos.
Técnica	Ejecutar cada caso de prueba utilizando la técnica de caja negra, haciendo uso de las clases de equivalencia y el análisis de valores límites.
Criterio de completitud	Las pruebas planeadas han sido ejecutadas. Todos los defectos identificados han sido corregidos.
Consideraciones especiales	

Nivel de Sistema

Pruebas de seguridad

Objetivos	• Nivel de Seguridad de Sistema: Verificar que solo los usuarios creados para el uso del sistema sean los que acceden al mismo. • Nivel de Seguridad de la Aplicación: Verificar que cada usuario acceda a las funcionalidades y datos correspondientes de acuerdo al rol que cumple dentro del sistema.
Técnica	• Utilizar la técnica similar a la de las pruebas funcionales identificando los usuarios que pueden acceder al sistema, sus funcionalidades y los que no tienen acceso al mismo de modo que se pruebe el comportamiento del sistema para ambos casos.
Criterio de completitud	• Solo los usuarios definidos para el uso del sistema acceden a las funciones que les corresponde y todas las transacciones son ejecutadas correctamente.
Consideraciones especiales	

Pruebas de Usabilidad

Objetivos	• Verificar que en la aplicación sea fácil la navegación para los usuarios, con los contenidos bien organizados y que las funciones sean utilizadas por éstos de manera sencilla. • Verificar que se cumplan con los estándares web y que existan métodos de acceso.
Técnica	• Utilizar la herramienta W3C Validator para verificar que se cumple con los estándares web. • Crear pruebas para cada interfaz, de forma que se verifique la navegación y métodos de acceso.
Criterio de completitud	• Todas las interfaces han cumplido con los estándares web. • La aplicación ha tenido una cómoda navegación para los usuarios y rápido acceso a las

	funcionalidades.
Consideraciones especiales	

Prueba de Integridad

Objetivos	Asegurar que los procesos y métodos de acceso a la base de datos funcionan de forma correcta.
Técnica	Crear datos válidos y no válidos para verificar la respuesta de dichos procesos y métodos de acceso. Inspeccionar la base de datos para asegurarse que los datos son correctos.
Criterio de completitud	Todos los métodos de acceso y procesos funcionan como fueron diseñados sin corrupción de datos.
Consideraciones especiales	

Prueba de Estructura

Objetivos	Verificar el funcionamiento de los vínculos y el contenido de cada uno de ellos.
Técnica	Usar la herramienta W3C Validator para comprobar si todos los vínculos funcionan correctamente.
Criterio de completitud	Todos lo vínculos funcionan de forma apropiada. La información es adecuada para cada página.
Consideraciones especiales	

Prueba de resistencia (stress)

Objetivos	Verificar que el sistema funcione sin errores con memoria RAM mínima establecida.
Técnica	Utilizar una memoria RAM de 2G para verificar el funcionamiento del sistema.
Criterio de completitud	El sistema funciona sin ningún error bajo esta condición

	de stress.
Consideraciones especiales	

Prueba de Configuración

Objetivos	Verificar que el sistema funcione bajo las configuraciones de software y hardware requeridas.
Técnica	Verificar el funcionamiento del navegador Internet Explorer, del sistema operativo Linux y del JRE.
Criterio de completitud	El sistema funciona correctamente con estos elementos.
Consideraciones especiales	

Pruebas de instalación

Objetivos	Verificar que el software pueda ser instalado bajo las siguientes condiciones: Una nueva instalación a una nueva máquina en la cual nunca se haya instalado la aplicación.
Técnica	Iniciar la instalación
Criterio de completitud	Se realiza la instalación correctamente
Consideraciones especiales	

2.3.6. Criterio de Evaluación

Para evaluar la completitud de las pruebas, se utilizará la métrica de cobertura de pruebas basada en los requerimientos del software, con esta métrica se analizan los resultados de las pruebas planeadas, ejecutadas, y de éstas, las satisfactorias con respecto a los requerimientos de prueba definidos en el plan. Además se utiliza la métrica de la cobertura de la prueba basada en el código, la cual da una medida de cuanto ha sido cubierto el código por las pruebas, es decir mide el código ejecutado por la prueba en comparación con el código que no ha sido ejecutado.

2.3.7. Herramientas

	Herramientas	**Versión**
Administración del proyecto	Microsoft Project Microsoft Word	2003 2007
Configuración entorno de pruebas	Microsoft Visio	2003

2.3.8. Recursos

Roles

Recursos Humanos		
Trabajador	Cantidad mínima de recursos	Responsabilidades
Jefe de calidad interna.	1	• Planificar los cursos de capacitación para el equipo de pruebas. • Establecer las reglas para la entrega del producto a Calisoft. • Aceptación del plan de pruebas.
Desarrollador de software	5	• Diseño de los casos de pruebas de las pruebas de unidad y de integración. • Implementación de las pruebas con la herramienta JUnit. • Ejecución de las pruebas de unidad e integración. • Realizar la depuración.
Jefe del equipo de pruebas	1	• Revisar los artefactos generados en el proceso de pruebas. • Verificar el cumplimiento del plan de pruebas. • Proponer mejoras para el proceso de pruebas.

Diseñador de pruebas	1	• Definir la estrategia de pruebas. • Especificar la configuración del entorno de pruebas.
Analista de pruebas	2	• Diseñar casos de prueba • Detallar los datos de prueba
Administrador de prueba	1	• Definir el plan de pruebas • Resumir la evaluación de las pruebas.
Probador (tester)	2	• Ejecutar las pruebas. • Elaborar documento de no conformidades.

Sistema

Recursos del Sistema	
Recurso	**Número de Serial**
Servidor Web	55274641948249623314
Servidor de Base de Datos	55274641948249623007
2 PC clientes (conectadas a la LAN y con conexión a Internet)	55274641948249623893 55274643186528523976
4 UPS	DED7061892 DED7062769 DED7061425 DED7062731
Repositorio para las pruebas	

2.3.9. Hitos del Proyecto

Las tareas que se detallan a continuación se enmarcan en la fase de elaboración donde se desarrollará un ciclo de prueba. El cronograma de las tareas de los ciclos de pruebas que se realizarán en la fase de construcción está por definir.

Tarea	Esfuerzo (por día)	Fecha Inicio	Fecha Fin
Planear las pruebas	En todo el proceso de pruebas.	diciembre 2005	Diciembre 2006
Instaurar el entorno de pruebas	Ciclo I	Semana 3 de mayo 2006	Semana 4 de mayo 2006
Diseñar las pruebas	Ciclo I	Semana 1 de enero 2006	Semana 3 de enero 2006
		Semana 1 de abril	Semana 3 de abril 2006
Implementar las pruebas	Ciclo I	Semana 4 de enero 2006	Semana 1 de febrero 2006
Ejecutar las pruebas	Ciclo I	Semana 2 de febrero 2006	Semana 3 de febrero de 2006
		Semana 4 de abril 2006	Semana 1 de mayo 2006
		Semana 1 de junio 2006	Semana 2 de junio 2006
Evaluar las pruebas	Ciclo I	Semana 4 de febrero 2006	Semana 1 de marzo 2006
		Semana 2 de mayo 2006	Semana 3 de mayo 2006
		Semana 1 de junio 2006	Semana 2 de junio 2006
Depuración	Ciclo I	Semana 2 de febrero 2006	Semana 4 de marzo 2006
		Semana 4 de abril 2006	Semana 4 de mayo 2006

2.3.10. Entregables

Entregables	Trabajador	Revisado por	Fecha límite
Plan de Pruebas	Adriana Román	Jefe del equipo	Semana 3 de

	Dayma Direntau	de prueba.	diciembre 2005
Entorno de Configuración de las pruebas	Dayma Direntau	Jefe del equipo de prueba.	Semana 3 de diciembre 2005
Datos de Pruebas	Adriana Román	Jefe del equipo de prueba.	Semana 4 de diciembre 2005
Modelo de Pruebas	Dayma Direntau Adriana Román	Jefe del equipo de prueba.	Semana 2 de junio 2006
Documento de no conformidades	Dayma Direntau	Jefe del equipo de prueba.	Semana 2 de junio 2006
Resumen de Evaluación de las pruebas	Adriana Román	Jefe del equipo de prueba.	Semana 2 de junio 2006

2.4. Artefacto: Configuración del Entorno de Prueba

En la configuración del entorno de prueba se detalla la distribución de hardware y software para llevar a cabo las pruebas, periféricos de entrada y salida y sus drivers correspondientes, herramientas que faciliten la prueba.

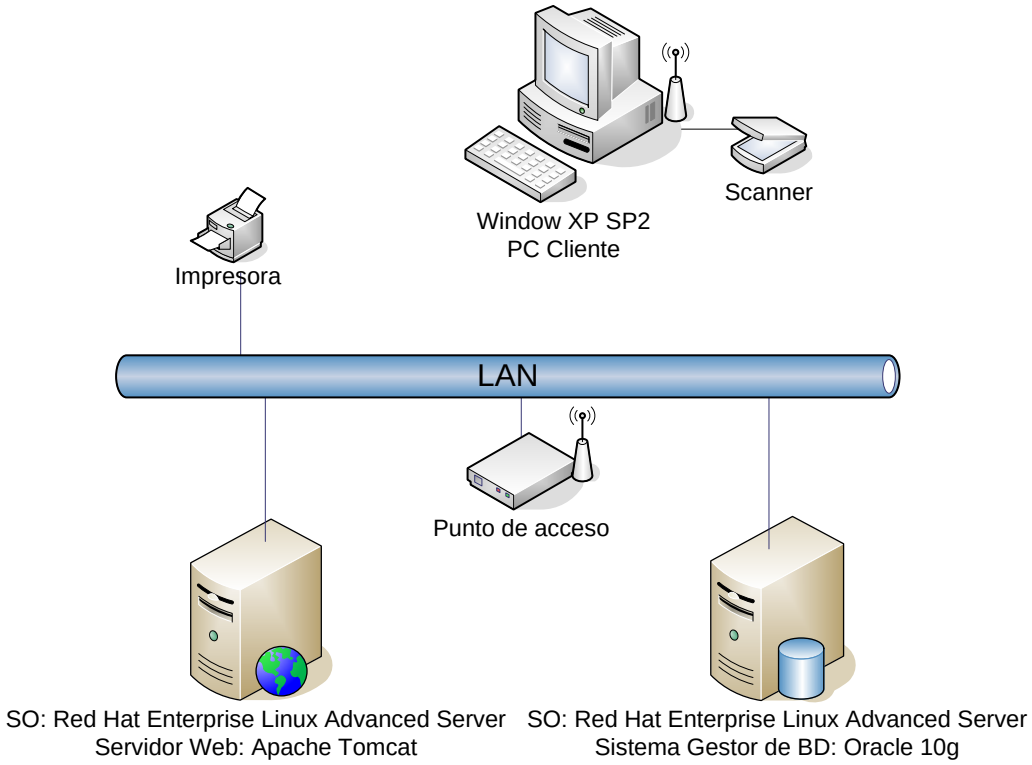

Figura 3. Configuración del entorno de pruebas

Nombre de propiedad	**Breve descripción**
Nombre	Entorno de prueba Servicio Autónomo
Descripción	El entorno de prueba consta de un servidor Web para la publicación de la aplicación, el cual tiene como sistema operativo Red Hat Enterprise Linux Advanced Server 4.0, el apache Tomcat 0.27 y el Entorno de Ejecución de Java (JRE) 1.05.0_06. Además tiene un servidor de base de datos con sistema operativo Red Hat Enterprise Linux Advanced Server 4.0, y con Oracle 10g como Sistema Gestor de Bases de Datos. Conjuntamente tiene equipos clientes, los cuales tienen instalado Windows XP SP2 con Kaspersky Anti-Virus for Windows WorkStation 5.0, y conectado por un puerto USB un Scanner para realizar las tareas de escaneo de la aplicación. Asimismo hay una impresora compartida para realizar todas las tareas de impresión, para ello se deben instalar los drivers requeridos de la impresora y del escáner. Se usa como herramientas de automatización de las pruebas el JUnit para las pruebas unitarias y el Validator W3C

	para verificar que el sitio Web cumpla con los estándares correspondientes, el cual debe de estar publicado en el Apache Tomcat. Se utilizará una red local (LAN) a la cual se conectarán el servidor de base de datos, el servidor Web, la impresora y un punto de acceso para la conexión de máquinas clientes inalámbricas.
Propósito	La configuración del entorno de pruebas tiene como objetivo principal crear un ambiente cercano al real, de forma tal que las pruebas no se lleven a cabo en el entorno de desarrollo del software, esto es muy importante ya que permite ver como se va a comportar el software en su ambiente verdadero, detectar los defectos y evaluar las pruebas con mayor facilidad.
Hardware Requerido	**2 modelos del tipo que se muestra a continuación, uno para el Servidor Web y otro para el Servidor de Bases de Datos.** Microprocesador Intel Pentium 4 CPU 3.0 GHz Dual Core 2 memorias de 1GB PC1600 ECC DDR 72GB 15K U320 Pluggable Hard Drive WW HBA FCA 2214 **PC cliente** HP Laserjet 2420N HP Scanjet 5590 con ADF DC5000 SFF P4/2.8 256MB 40GB CD WXPP (1 x Intel P4) Monitor CRT 15" S5500 Tarjeta de Red Inalámbrica PCI WL500 802.11b/g 54Mbps Low Profile
Software Requerido	Red Hat Advanced Server 4.0. Entorno de Ejecución de Java (JRE) 1.05.0_06. Apache Tomcat 0.27. Windows XP SP2. Oracle 10g. Drivers para la impresora y escáner. Microsoft Internet Explorer 5.0 o superior. hp_scanjet_HP 5550. Kaspersky Anti-Virus for Window WorkStation 5.0 Validator W3C 0.7.4
Procedimiento de	Crear copias de seguridad de la imagen de cada equipo y un backup de la Base de Datos diariamente. De esta forma estos elementos se pueden usar para lograr una

restauración y recuperación	restauración y recuperación de la configuración del entorno de prueba, lo cual facilita el trabajo.

2.5. Conclusiones

Luego de realizar el plan de pruebas y la configuración del entorno de pruebas, se ha arribado a las siguientes conclusiones:

- La planificación de las pruebas de software es una de las actividades fundamentales para la buena ejecución de las mismas.
- Si se planifica una buena estrategia de pruebas, es decir comenzar probando las unidades más pequeñas hasta integrar todo el sistema y probarlo completamente, y ésta se lleva a cabo correctamente, se puede garantizar una mayor calidad del proceso y por ende una mayor calidad de producto final.
- Los requisitos que serán probados tienen que estar incluidos en el plan de pruebas como uno de los aspectos indispensables, ya que demuestra que los que sean excluidos del mismo no son objetivos del ciclo de pruebas que se desea realizar, de esta forma se garantiza que en la ejecución de las pruebas no se olvide ninguno de los requisitos planeados.
- El plan permite definir qué tipos de pruebas y en qué momento es preciso su realización de acuerdo a las características del módulo.
- La realización del cronograma de pruebas permite tener precisión de los días en que se realizará cada tarea del proceso, lo cual posibilita una mejor organización del personal de acuerdo a las actividades que deben realizar.
- La asignación de actividades por roles conlleva a que el personal del proyecto esté orientado en cuanto a lo que debe hacer en cada momento, por lo cual el trabajo se realiza de forma eficiente y sin pérdida de tiempo siendo responsabilidad de una persona en concreto.
- Que haya un jefe del equipo de prueba sirve para que guíe y revise el trabajo.
- Una correcta configuración del entorno de pruebas administra de forma eficiente los recursos destinados para esta tarea ya que en muchas ocasiones no se cuenta con los necesarios.

Capítulo 3: Diseño, ejecución y análisis de los resultados.

3.1. Introducción

De acuerdo a las etapas definidas en el plan de pruebas, este capítulo se centra en la etapa III referente a las pruebas funcionales, en el cual se realiza el diseño de las mismas como un aspecto fundamental para su ejecución. Estas actividades se incluyen en el modelo de pruebas; este artefacto contiene casos de prueba que utilizan como entrada los datos de prueba como clases válidas y otros datos como clases no válidas y los procedimientos que explican cómo se ejecuta el caso de prueba. Para ello, se utilizaron 10 casos de uso, de los cuales se identificaron 26 casos de prueba. De los resultados reales obtenidos de su ejecución, en el sumario de evaluación se resumen los errores más frecuentes y se utiliza la métrica de cobertura de pruebas basada en los requerimientos para hacer un análisis de la completitud de las pruebas, haciendo una estimación de las que han sido planeadas, ejecutadas y de éstas, las satisfactorias con respecto a los requisitos definidos en el plan. Se analizan, además, los defectos encontrados, haciendo una clasificación de los mismos según la prioridad (urgente, alto, normal, bajo), el grado de severidad (crítico, alto, medio, bajo), origen (el componente donde reside el problema) y estado (arreglado, probado, asignado, pendiente) con el objetivo de que los desarrolladores los conozcan y los puedan solucionar. La corrección de los defectos posibilita el aumento de la fiabilidad del software dando una medida de la calidad del mismo y su correspondencia con los requerimientos establecidos.

3.2. Artefacto: Datos de Pruebas

Este artefacto es de gran importancia para poder ejecutar los casos de prueba, ya que especifica un listado de datos reales referentes al negocio, los cuales se utilizan como entradas, salidas y precondiciones. (Ver Anexo 2)

3.3. Artefacto: Modelo de Pruebas

El modelo de pruebas se genera como parte de dicho proceso para diseñar e implementar las pruebas. Éste contiene el diseño de los casos de prueba, los procedimientos de prueba y los componentes de prueba. Estos últimos no se utilizan en la tesis para las pruebas funcionales.

3.3.1. Caso de uso Administrar Usuarios

A este Caso de Uso se le realizaron las siguientes pruebas:

- Caso de prueba “Adicionar Nueva Persona”
- Caso de Prueba “Crear Usuario”
- Caso de prueba “Buscar usuarios por oficina”
- Caso de prueba “Buscar usuarios por persona”
- Caso de Prueba “Modificar Usuario”
- Caso de Prueba “Cambiar Estado de Usuario”

3.3.1.1. CPR 1: < Caso de prueba “Adicionar Nueva Persona” >

3.3.1.1.1. Descripción de la Funcionalidad:

Este caso de prueba describe cómo se realiza la inserción de nuevas personas.

3.3.1.1.2. Flujo Central:

- En el menú superior, pulse la opciones Configuración, Usuarios, Administrar Usuarios y luego de realizar la búsqueda del usuario, pulse el botón Crear. En la interfaz Crear Usuario, luego de haber obtenido el resultado de la búsqueda de personas, pulse el enlace Nueva Persona.
- Puede seleccionar el campo Cédula o Pasaporte, en caso de que elija el primero, debe escoger la letra de cédula e insertar el número, en caso contrario, inserte el pasaporte.
- Se muestran, además los siguientes campos a llenar: Primer Nombre, Segundo Nombre, Primer Apellido, Segundo Apellido, Estado civil, Sexo y País.
- Pulse el botón Aceptar Datos.

3.3.1.1.3. Iteraciones.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Letra de Cédula: V **Número de Cédula:** 67.567.233 **Primer Nombre:** Ángela **Segundo Nombre:** María **Primer Apellido:** Pérez **Segundo Apellido:** Rodríguez **Estado Civil:** Soltera **Sexo:** Femenino **País:** Venezuela De clic en el botón Aceptar Datos		Se adiciona la nueva persona	No Satisfactorio	
Letra de Cédula: Vacío **Número de Cédula:** 67.567.233 **Primer Nombre:** Ángela **Segundo Nombre:** Vacío **Primer Apellido:** Pérez **Segundo Apellido:** Vacío **Estado Civil:** Soltera **Sexo:** Femenino **País:** Venezuela De clic en el botón Aceptar Datos		Se adiciona la nueva persona	No Satisfactorio	

Letra de Cédula: Vacío **Número de Cédula:** 67.567.233 **Primer Nombre:** Vacío **Segundo Nombre:** María **Primer Apellido:** Vacío **Segundo Apellido:** Rodríguez **Estado Civil:** Soltera **Sexo:** Femenino **País:** Venezuela De clic en el botón Aceptar Datos		Se adiciona la nueva persona	No Satisfactorio	
Pasaporte:ATF12 **Primer Nombre:** Ángela **Segundo Nombre:** María **Primer Apellido:** Pérez **Segundo Apellido:** Rodríguez **Estado Civil:** Soltera **Sexo:** Femenino **País:** Venezuela De clic en el botón Aceptar Datos		Se adiciona la nueva persona	No Satisfactorio	
	Letra de Cédula: Vacío **Número de Cédula:** Vacío **Primer Nombre:**	Se muestra una ventana con el siguiente mensaje: "Especifique	No Satisfactorio	

	Vacío **Segundo Nombre:** Vacío **Primer Apellido:** Vacío **Segundo Apellido:** Vacío **Estado Civil:** Vacío **Sexo:** Vacío **País:** Vacío De clic en el botón Aceptar Datos	el número de cédula, al menos un nombre y un apellido, el estado civil, sexo, y país"		
	Letra de Cédula: V **Número de Cédula:** Vacío **Primer Nombre:** Vacío **Segundo Nombre:** Vacío **Primer Apellido:** Vacío **Segundo Apellido:** Vacío **Estado Civil:** Vacío **Sexo:** Vacío **País:** Vacío De clic en el botón Aceptar Datos	Se muestra una ventana con el siguiente mensaje: "Especifique el número de cédula, al menos un nombre y un apellido, el estado civil, sexo, y país"	No Satisfactorio	
	Letra de Cédula: Vacío **Número de Cédula:** 67.567.233 **Primer Nombre:** Vacío **Segundo Nombre:** Vacío	Se muestra una ventana con el siguiente mensaje: "Especifique al menos un nombre y un apellido, el	No Satisfactorio	

	Primer Apellido: Vacío **Segundo Apellido:** Vacío **Estado Civil:** Vacío **Sexo:** Vacío **País:** Vacío De clic en el botón Aceptar Datos	estado civil, sexo, y país"		
	Letra de Cédula: Vacío **Número de Cédula:** Vacío **Primer Nombre:** Ángela **Segundo Nombre:** Vacío **Primer Apellido:** Vacío **Segundo Apellido:** Vacío **Estado Civil:** Vacío **Sexo:** Vacío **País:** Vacío De clic en el botón Aceptar Datos	Se muestra una ventana con el siguiente mensaje: "Especifique el número de cédula, al menos un apellido, el estado civil, sexo, y país"	No Satisfactorio	
	Letra de Cédula: Vacío **Número de Cédula:** Vacío **Primer Nombre:** Vacío **Segundo Nombre:** María **Primer Apellido:** Vacío	Se muestra una ventana con el siguiente mensaje: "Especifique el número de cédula, al menos un apellido, el estado civil,	No Satisfactorio	

	Segundo Apellido: Vacío **Estado Civil:** Vacío **Sexo:** Vacío **País:** Vacío De clic en el botón Aceptar Datos	sexo, y país"		
	Letra de Cédula: Vacío **Número de Cédula:** Vacío **Primer Nombre:** Vacío **Segundo Nombre:** Vacío **Primer Apellido:** Pérez **Segundo Apellido:** Vacío **Estado Civil:** Vacío **Sexo:** Vacío **País:** Vacío De clic en el botón Aceptar Datos	Se muestra una ventana con el siguiente mensaje: "Especifique el número de cédula, al menos un nombre, el estado civil, sexo, y país"	No Satisfactorio	
	Letra de Cédula: Vacío **Número de Cédula:** Vacío **Primer Nombre:** Vacío **Segundo Nombre:** Vacío **Primer Apellido:** Vacío **Segundo Apellido:** Rodríguez	Se muestra una ventana con el siguiente mensaje: "Especifique el número de cédula, al menos un nombre, el estado civil, sexo, y país"	No Satisfactorio	

	Estado Civil: Vacío **Sexo:** Vacío **País:** Vacío De clic en el botón Aceptar Datos			
	Letra de Cédula: Vacío **Número de Cédula:** Vacío **Primer Nombre:** Vacío **Segundo Nombre:** Vacío **Primer Apellido:** Vacío **Segundo Apellido:** Vacío **Estado Civil:** Soltera **Sexo:** Vacío **País:** Vacío De clic en el botón Aceptar Datos	Se muestra una ventana con el siguiente mensaje: "Especifique el número de cédula, al menos un nombre y un apellido, el sexo, y país"	No Satisfactorio	
	Letra de Cédula: Vacío **Número de Cédula:** Vacío **Primer Nombre:** Vacío **Segundo Nombre:** Vacío **Primer Apellido:** Vacío **Segundo Apellido:** Vacío **Estado Civil:** Vacío	Se muestra una ventana con el siguiente mensaje: "Especifique el número de cédula, al menos un nombre y un apellido, el estado civil y el país"	No Satisfactorio	

	Sexo: Femenino **País:** Vacío De clic en el botón Aceptar Datos			
	Letra de Cédula: Vacío **Número de Cédula:** Vacío **Primer Nombre:** Vacío **Segundo Nombre:** Vacío **Primer Apellido:** Vacío **Segundo Apellido:** Vacío **Estado Civil:** Vacío **Sexo:** Vacío **País:** Venezuela De clic en el botón Aceptar Datos	Se muestra una ventana con el siguiente mensaje: "Especifique el número de cédula, al menos un nombre y un apellido, el estado civil y el sexo"	No Satisfactorio	
	Pasaporte: Vacío **Primer Nombre:** Ángela **Segundo Nombre:** María **Primer Apellido:** Pérez **Segundo Apellido:** Rodríguez **Estado Civil:** Soltera **Sexo:** Femenino **País:** Venezuela De clic en el botón Aceptar Datos	Se muestra una ventana con el siguiente mensaje: "Especifique el pasaporte"	No Satisfactorio	

	Letra de Cédula: Vacío **Número de Cédula:** -455 **Primer Nombre:** 45 **Segundo Nombre:** 45 **Primer Apellido:** 12 **Segundo Apellido:** 12 **Estado Civil:** Vacío **Sexo:** Vacío **País:** Vacío De clic en el botón Aceptar Datos	El sistema no permite la entrada de valores negativos para el número de cédula, ni números para los nombres ni apellidos.	No Satisfactorio	
	Letra de Cédula: Vacío **Número de Cédula:** 54.822.1236 **Primer Nombre:** adroenfnfiquestarmetipierto()jhiu **Segundo Nombre:** adroenfnfiquestarmetipierto()jhiu **Primer Apellido:** adroenfnfiquestarmetipierto()jhiu **Segundo Apellido:** adroenfnfiquestarmetipierto()jhiu **Estado Civil:** Vacío **Sexo:** Vacío **País:** Venezuela De clic en el botón Aceptar Datos	El sistema no permite la entrada de más de 8 dígitos, ni para los nombres y apellidos, la entrada de más de 30 caracteres (a-z, A-Z).	No Satisfactorio	

	Letra de Cédula: Vacío **Número de Cédula:** ast() **Primer Nombre:** Vacío **Segundo Nombre:** Vacío **Primer Apellido:** Vacío **Segundo Apellido:** Vacío **Estado Civil:** Vacío **Sexo:** Vacío **País:** Vacío De clic en el botón Aceptar Datos	El sistema no permite la entrada de ningún carácter para el número de cédula.	No Satisfactorio	
De clic en el botón Cancelar		Se muestra una ventana con el siguiente mensaje: "¿Realmente desea cancelar?"	No Satisfactorio	

3.3.1.2. CPR 2: < Caso de prueba “Crear Usuario” >

3.3.1.2.1. Descripción de la Funcionalidad:

Este caso de prueba describe cómo se lleva a cabo la creación de un usuario a una persona determinada.

3.3.1.2.2. Flujo Central:

- En el menú superior, pulse la opciones Configuración, Usuarios, Administrar Usuarios y luego se realiza la búsqueda del usuario. Si esta lista no arroja ningún resultado o desea crear uno nuevo, pulse el botón Crear. Se muestra una nueva interfaz para la creación del usuario.
- Para la búsqueda de personas debe especificar algún criterio de búsqueda: Primer Nombre, Segundo Nombre, Primer Apellido, Segundo Apellido, Letra de Cédula o Número de Cédula.
- Luego pulse el botón Buscar. Se muestra en la Lista de Personas, la(s) persona(s) que concuerdan con el criterio especificado.
- En la parte inferior inserte los siguientes datos: Nombre de Usuario, Correo electrónico, Teléfono, Cargo, y Estado de Activación
- Pulse el botón Crear Usuario.

3.3.1.2.3. Condiciones de Ejecución:

Deben existir personas en la base de datos.

3.3.1.2.4. Iteraciones.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones

Primer Nombre: Alveria **Segundo Nombre:** Josefina **Primer Apellido:** Brito **Segundo Apellido:** Quijada **Letra de Cédula:** V **Número de Cédula:** 5.121.479 De clic en el botón Buscar		Se muestra la persona especificada.	No Satisfactorio	
	Primer Nombre: Vacío **Segundo Nombre:** Vacío **Primer Apellido:** Vacío **Segundo Apellido:** Vacío **Letra de Cédula:** Vacío **Número de Cédula:** Vacío De clic en el botón Buscar	Se muestra una ventana con el siguiente mensaje: "Especifique algún parámetro de búsqueda de personas."	No Satisfactorio	

Primer Nombre: Alveria **Segundo Nombre:** Vacío **Primer Apellido:** Vacío **Segundo Apellido:** Vacío **Letra de Cédula:** Vacío **Número de Cédula:** Vacío De clic en el botón Buscar		Se muestra la persona deseada.	No Satisfactorio	
Primer Nombre: Vacío **Segundo Nombre:** Josefina **Primer Apellido:** Vacío **Segundo Apellido:** Vacío **Letra de Cédula:** Vacío **Número de Cédula:** Vacío De clic en el botón Buscar		Se muestra la persona deseada.	No Satisfactorio	
Primer Nombre: Vacío **Segundo Nombre:** Vacío **Primer Apellido:** Brito **Segundo Apellido:** Vacío **Letra de Cédula:** Vacío **Número de**		Se muestra la persona deseada.	No Satisfactorio	

Cédula: Vacío De clic en el botón Buscar				
Primer Nombre: Vacío **Segundo Nombre:** Vacío **Primer Apellido:** Vacío **Segundo Apellido:** Quijada **Letra de Cédula:** Vacío **Número de Cédula:** Vacío De clic en el botón Buscar		Se muestra la persona deseada.	No Satisfactorio	
Primer Nombre: Vacío **Segundo Nombre:** Vacío **Primer Apellido:** Vacío **Segundo Apellido:** Vacío **Letra de Cédula:** V **Número de Cédula:** Vacío De clic en el botón Buscar		Se muestra la persona deseada.	No Satisfactorio	
Primer Nombre: Vacío **Segundo Nombre:** Vacío **Primer Apellido:** Vacío **Segundo Apellido:** Vacío		Se muestra la persona deseada.	No Satisfactorio	

Letra de Cédula: Vacío **Número de Cédula:** 5.121.479 De clic en el botón Buscar				
	Primer Nombre: 545 **Segundo Nombre:** 434 **Primer Apellido:** 437 **Segundo Apellido:** 784 **Letra de Cédula:** V **Número de Cédula:** ytyt()	El sistema no permite para la entrada de los nombres y apellidos, números, ni para el número de cédula, la entrada de caracteres.	Satisfactorio	
	Primer Nombre: adroenfnfiquestarmetipierto()jhiu **Segundo Nombre:** adroenfnfiquestarmetipierto()jhiu **Primer Apellido:** adroenfnfiquestarmetipierto()jhiu **Segundo Apellido:** adroenfnfiquestarmetipierto()jhiu **Letra de Cédula:** V **Número de Cédula:** 5.121.47933	El sistema no permite para la entrada de nombres y apellidos, la inserción de más de 30 caracteres, ni otros caracteres que no sean (a-z, A-Z) ni poner más de 8 dígitos en el número de cédula.	Satisfactorio	

	Primer Nombre: Vacío **Segundo Nombre:** Vacío **Primer Apellido:** Vacío **Segundo Apellido:** Vacío **Letra de Cédula:** Vacío **Número de Cédula:** -456	El sistema no permite la entrada de valores negativos en el número de cédula.	Satisfactorio	
Nombre de Usuario: abritoq **Correo electrónico:** abritoq@as.com **Teléfono:**6789jk4 **Cargo:** Contador I **Roles Disponibles:** Reportes, Revisar Prohibiciones **Roles Asignados:** Revisar Prohibiciones De clic en el botón Crear Usuario		Se muestra una ventana especificando el siguiente mensaje: “El usuario se ha creado satisfactoriamente”.	No Satisfactorio	

Nombre de Usuario: abritoq **Correo electrónico:** Vacío **Teléfono:** Vacío **Cargo:** Vacío **Roles Disponibles:** Reportes, Revisar Prohibiciones **Roles Asignados:** Revisar Prohibiciones De clic en el botón Crear Usuario		Se muestra una ventana especificando el siguiente mensaje: "El usuario se ha creado satisfactoriamente".	No Satisfactorio	
	Nombre de Usuario: abritoq **Correo electrónico:** Vacío **Teléfono:** Vacío **Cargo:** Vacío **Roles Disponibles:** Reportes, Revisar Prohibiciones **Roles Asignados:** Vacío De clic en el botón Crear Usuario	El sistema muestra el siguiente mensaje:" Especifique los roles asignados".	No Satisfactorio	
	Nombre de Usuario: Vacío **Correo electrónico:** abritoq@as.com **Teléfono:** Vacío **Cargo:** Vacío	Se muestra una ventana con el siguiente mensaje:" Especifique el nombre de usuario. Especifique los	No Satisfactorio	

	Roles Disponibles: Reportes, Revisar Prohibiciones **Roles Asignados:** Vacío De clic en el botón Crear Usuario	roles asignados".		
	Nombre de Usuario: Vacío **Correo electrónico:** Vacío **Teléfono:**6789jk4 **Cargo:** Vacío **Roles Disponibles:** Reportes, Revisar Prohibiciones **Roles Asignados:** Vacío De clic en el botón Crear Usuario	Se muestra una ventana con el siguiente mensaje:" Especifique el nombre de usuario. Especifique los roles asignados".	No Satisfactorio	
	Nombre de Usuario: Vacío **Correo electrónico:** Vacío **Teléfono:** Vacío **Cargo:** Contador I **Roles Disponibles:** Reportes, Revisar Prohibiciones **Roles Asignados:** Vacío De clic en el botón Crear Usuario	Se muestra una ventana con el siguiente mensaje:" Especifique el nombre de usuario. Especifique los roles asignados".	No Satisfactorio	

	Nombre de Usuario: Vacío **Correo electrónico:** Vacío **Teléfono:** Vacío **Cargo:** Vacío **Roles Disponibles:** Reportes, Revisar Prohibiciones **Roles Asignados:** Revisar Prohibiciones De clic en el botón Crear Usuario	Se muestra una ventana con el siguiente mensaje:" Especifique el nombre de usuario".	No Satisfactorio	
	Nombre de Usuario: abritoq **Correo electrónico:** abritoqcom **Teléfono:**6789jk4 **Cargo:** Contador I **Roles Disponibles:** Reportes, Revisar Prohibiciones **Roles Asignados:** Revisar Prohibiciones De clic en el botón Crear Usuario	Se muestra una ventana con el siguiente mensaje:" El formato de correo electrónico está incorrecto"	No Satisfactorio	
	Nombre de Usuario: alveriabritoquijada **Correo electrónico:** Vacío **Teléfono:** Vacío	El sistema no permite la entrada de más de 16 caracteres para el nombre de usuario.	Satisfactorio	

	Cargo: Vacío **Roles Disponibles:** Reportes, Revisar Prohibiciones **Roles Asignados:** Vacio			
	Nombre de Usuario: abritoq **Correo electrónico:** Vacío **Teléfono:** -456 **Cargo:** Vacio **Roles Disponibles:** Reportes, Revisar Prohibiciones **Roles Asignados:** Vacio	El sistema no permite la entrada de valores negativos para el número de teléfono.	Satisfactorio	
De clic en el botón Cancelar		Se muestra una ventana con el siguiente mensaje: "¿Realmente desea cancelar?"	Satisfactorio	

3.3.1.2.5. Registro de defectos y dificultades detectados

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Interfaz	1	No se puede realizar la búsqueda de personas para asignarles un	Se encuentra en la interfaz "Crear Usuario". Esta sección pertenece al caso de uso	Prueba	X	

		usuario pues se muestra un error (Ver Anexo 3 Fig. 1). El botón Cancelar de la interfaz que muestra el error, no funciona.	Administrar Usuario			

3.3.1.3 CPR 3: < Caso de prueba “Buscar usuarios por oficina” >

3.3.1.3.1. Descripción de la Funcionalidad:

Este caso de prueba describe cómo se realiza la búsqueda de usuarios de una oficina determinada, la cual se busca según diferentes criterios.

3.3.1.3.2. Flujo Central:

- En el menú superior, seleccione la opción Configuración, dentro de la misma, elija la opción Usuarios y luego Administrar Usuarios. Se muestra una interfaz para configurar los mismos.
- En el campo Buscar Usuarios por, seleccione Oficinas.
- La búsqueda de oficinas puede ser por Criterios o Código. Si elije Criterios debe seleccionar alguno de los campos siguientes: Estado, Municipio o Tipo de oficina. En caso de que elija Código, debe especificar el mismo en el campo Código.
- Pulse el botón Buscar.
- Después de haberlo pulsado, se muestra en Listado de Oficinas, una o varias oficinas según el criterio de búsqueda especificado.
- De doble clic sobre la oficina que desee para que en la parte derecha se muestren los datos de la misma: Nombre, Dirección, Estado de Activación, Tipo de Oficina, Teléfono, Fax, Estado, Municipio y Código.

- En la parte inferior se muestra un Listado de Usuarios como resultado de la búsqueda, al dar doble clic sobre uno de ellos, se muestra en la parte derecha: Usuario, Primer Nombre, Segundo Nombre, Primer Apellido, Segundo Apellido, Letra de cédula, Número de cédula, Sexo, Estado civil, Estado de activación y Pertenece a la oficina.

3.3.1.3.3. Condiciones de Ejecución:

Deben existir oficinas y usuarios en la base de datos.

3.3.1.3.4. Iteraciones.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Buscar usuarios por: Oficinas **Buscar oficinas por:** Criterio **Estado:** Amazonas **Municipio:** Alto Orinoco **Tipo de Oficina:** Mercantil Se da clic en el botón Buscar		Se muestra la oficina existente.	Satisfactorio	Cuando se selecciona Buscar oficina por Criterio, no se debería mostrar el campo Código.

	Buscar usuarios por: Oficinas **Buscar oficinas por:** Criterio **Estado:** Vacío **Municipio:** Vacío **Tipo de Oficina:** Vacío Se da clic en el botón Buscar	Se muestra la ventana con el mensaje siguiente: "Especifique algún criterio de búsqueda".	Satisfactorio	
Buscar usuarios por: Oficinas **Buscar oficinas por:** Criterio **Estado:** Amazonas **Municipio:** Vacío **Tipo de Oficina:** Vacío Se da clic en el botón Buscar		Se muestra la oficina existente.	Satisfactorio	
Buscar usuarios por: Oficinas **Buscar oficinas por:** Criterio **Estado:** Amazonas **Municipio:** Vacío		Se muestra la oficina existente.	Satisfactorio	

Tipo de Oficina: Mercantil Se da clic en el botón Buscar				
Buscar usuarios por: Oficinas **Buscar oficinas por:** Criterio **Estado:** Vacío **Municipio:** Vacío **Tipo de Oficina:** Mercantil Se da clic en el botón Buscar		Se muestra la oficina existente.	Satisfactorio	
Buscar usuarios por: Oficinas **Buscar oficinas por:** Código **Código:** 123 Se da clic en el botón Buscar		Se muestra la oficina existente.	Satisfactorio	
	Buscar usuarios por: Oficinas **Buscar oficinas por:** Código **Código:** Vacío	Se muestra una ventana con el siguiente mensaje:"Especifique el código de la oficina".	Satisfactorio	
	Buscar usuarios por: Oficinas **Buscar oficinas**	El sistema no permite introducir valores negativos para el código de la oficina.	Satisfactorio	

	por: Código **Código:** -123			
	Buscar usuarios por: Oficinas **Buscar oficinas por:** Código **Código:** 1234	El sistema no permite introducir más de tres dígitos para el código de la oficina.	Satisfactorio	
	Buscar usuarios por: Oficinas **Buscar oficinas por:** Código **Código:** 1yt	El sistema no permite introducir caracteres para el código de la oficina.	Satisfactorio	
De clic sobre la oficina mostrada como resultado de la búsqueda.		Se muestran los datos de la oficina, en la lista de usuarios, el usuario correspondiente y el botón Crear.	Satisfactorio	
De clic sobre el usuario mostrado como resultado de la búsqueda		Se muestran los datos del usuario y los botones Modificar y Cambiar de Estado	Satisfactorio	
Se da clic en el botón Cancelar		Se una ventana con el siguiente mensaje: "¿Realmente desea cancelar?"	Satisfactorio	

3.3.1.4 CPR 4: < “Caso de prueba “Buscar usuarios por Personas” >

3.3.1.4.1. Descripción de la Funcionalidad:

Este caso de prueba describe como se realiza la búsqueda de personas que tienen usuarios asignados.

3.3.1.4.2. Flujo Central:

- En el menú superior, seleccione la opción Configuración, dentro de la misma, elija la opción Usuarios y luego Administrar Usuarios. Se muestra una interfaz para configurar los mismos.
- En el campo Buscar Usuarios por, seleccione Personas.
- Para buscar a una persona según un criterio específico, puede seleccionar alguno de los siguientes campos: Número de Cédula, Primer Nombre, Segundo Nombre, Primer Apellido o Segundo Apellido y luego pulsar el botón Buscar.
- Después de haberlo pulsado, se muestra en Listado de Personas, una o varias personas.
- Una vez que de clic sobre una persona determinada, en la parte derecha se muestran los datos de la misma: Primer Nombre, Segundo Nombre, Primer Apellido, Segundo Apellido, Letra de Cédula, Número de Cédula, Sexo y Estado Civil.
- En la parte inferior, en Listado de Usuarios, se muestra además, el(los) usuario(s) que esa persona tiene asignado. Al dar clic sobre éste, se muestra en la parte derecha: Usuario, Primer Nombre, Segundo Nombre, Primer Apellido, Segundo Apellido, Número de cédula, Pasaporte, Sexo, Estado civil, Estado de activación, Roles y Pertenece a la oficina.

3.3.1.4.3. Condiciones de Ejecución:

Deben existir personas y usuarios asignados a éstas en la base de datos.

3.3.1.4.4. Iteraciones.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Buscar usuarios por: Personas **Letra de Cédula:** V **Número de Cédula:** 5.121.479 **Primer Nombre:** Alveria **Segundo Nombre:** Josefina **Primer Apellido:** Brito **Segundo Apellido:** Quijada De clic en el botón Buscar		Se muestra la persona deseada.	Satisfactorio	
	Buscar usuarios por: Personas **Letra de Cédula:** Vacío **Número de Cédula:** Vacío **Primer Nombre:** Vacío **Segundo Nombre:** Vacío **Primer Apellido:** Vacío **Segundo Apellido:** Vacío De clic en el botón Buscar	Se muestra una ventana con el mensaje siguiente: "Especifique algún criterio de búsqueda de usuarios".	Satisfactorio	
Buscar usuarios por: Personas **Letra de Cédula:** V **Número de Cédula:** Vacío **Primer Nombre:**		Se muestra la persona deseada.	Satisfactorio	

Vacío **Segundo Nombre:** Vacío **Primer Apellido:** Vacío **Segundo Apellido:** Vacío De clic en el botón Buscar				
Buscar usuarios por: Personas **Letra de Cédula:** Vacío **Número de Cédula:** 5.121.479 **Primer Nombre:** Vacío **Segundo Nombre:** Vacío **Primer Apellido:** Vacío **Segundo Apellido:** Vacío De clic en el botón Buscar		Se muestra la persona deseada.	Satisfactorio	
Buscar usuarios por: Personas **Letra de Cédula:** Vacío **Número de Cédula:** Vacío **Primer Nombre:** Alveria **Segundo Nombre:** Vacío **Primer Apellido:** Vacío **Segundo Apellido:** Vacío		Se muestra la persona deseada.	Satisfactorio	

De clic en el botón Buscar				
Buscar usuarios por: Personas **Letra de Cédula:** Vacío **Número de Cédula:** Vacío **Primer Nombre:** Vacío **Segundo Nombre:** Josefina **Primer Apellido:** Vacío **Segundo Apellido:** Vacío De clic en el botón Buscar		Se muestra la persona deseada.	Satisfactorio	
Buscar usuarios por: Personas **Letra de Cédula:** Vacío **Número de Cédula:** Vacío **Primer Nombre:** Vacío **Segundo Nombre:** Vacío **Primer Apellido:** Brito **Segundo Apellido:** Vacío De clic en el botón Buscar		Se muestra la persona deseada.	Satisfactorio	
Buscar usuarios por: Personas **Letra de Cédula:** Vacío **Número de Cédula:**		Se muestra la persona deseada.	Satisfactorio	

Vacío **Primer Nombre:** Vacío **Segundo Nombre:** Vacío **Primer Apellido:** Vacío **Segundo Apellido:** Quijada De clic en el botón Buscar				
	Buscar usuarios por: Personas **Letra de Cédula:** V **Número de Cédula:** ytyt **Primer Nombre:** 545 **Segundo Nombre:** 434 **Primer Apellido:** 437 **Segundo Apellido:** 784	El sistema no permite para la cédula, la entrada de caracteres ni para los demás criterios, la entrada de números.	Satisfactorio	
	Buscar usuarios por: Personas **Letra de Cédula:** Vacío **Número de Cédula:** -456 **Primer Nombre:** Vacío **Segundo Nombre:** Vacío **Primer Apellido:** Vacío **Segundo Apellido:** Vacío	El sistema no permite la entrada de valores negativos en el número de cédula.	Satisfactorio	

	Buscar usuarios por: Personas **Letra de Cédula:** V **Número de Cédula:** 5.121.47933 **Primer Nombre:** adroenfnfiquestar metipierto()jhiu **Segundo Nombre:** adroenfnfiquestar metipierto()jhiu **Primer Apellido:** adroenfnfiquestar metipierto()jhiu **Segundo Apellido:** adroenfnfiquestar metipierto()jhiu	El sistema no permite poner más de 8 dígitos en el número de cédula, ni para la entrada de nombres y apellidos, la inserción de más de 30 caracteres, ni otros caracteres que no sean (a-z, A-Z).	Satisfactorio	
De doble clic sobre el nombre de la persona que aparece como resultado de la búsqueda.		Aparecen los datos de la persona y se muestra el usuario correspondiente.	Satisfactorio	
De doble clic sobre el usuario		Aparecen sus datos y los botones Modificar y Cambiar Estado.	Satisfactorio	
De clic en el botón Cancelar		Se una ventana con el siguiente mensaje: "¿Realmente desea cancelar?".	Satisfactorio	

3.4. Sumario de Evaluación de las pruebas

3.4.1. Resumen de los resultados de las pruebas

Para la realización de las pruebas de software se contaron con 26 requisitos de tipo funcional.

La cobertura de pruebas planeadas respecto a los requerimientos de prueba definidos en el plan ha sido completada al 100%.

De las pruebas planeadas, las ejecutadas solo cubrieron el 81% de los requisitos funcionales.

De las pruebas planeadas, las ejecutadas que fueron exitosas solo cubrieron el 27% de los requisitos funcionales.

Los principales problemas que se han detectado durante la ejecución de las pruebas de software han sido producto de la interacción de la aplicación con la base de datos en las búsquedas y en la persistencia de la información, de forma más específica, en la búsqueda de personas naturales, jurídicas, inmuebles, documentos jurídicos y al intentar guardar los datos referente a la modificación de roles de usuarios, cambio de su estado de activación, creación de oficinas, cambio de estado de la misma, inserción de medidas de prohibición y de plantillas de actos mercantiles.

3.4.2. Cobertura de la prueba

3.4.2.1. Planificación

Se contaron con 26 casos de prueba para realizar las pruebas funcionales para un total de 26 pruebas planeadas en términos de casos de prueba.

El resultado de la cobertura de prueba para la actividad de planificación se muestra a continuación:

Cobertura de pruebas (planeadas) $=T^P/$ RfT= 26/26= 100%

T^P - Pruebas planeadas expresadas en casos de prueba.

RfT- Requisitos de prueba

3.4.2.2. Ejecución

Se ejecutaron 21 pruebas funcionales, 7 de las cuales fueron exitosas.

Los resultados de la cobertura de pruebas ejecutadas fueron los siguientes teniendo en cuenta que de 26 casos de prueba se obtuvo la siguiente clasificación:

Ejecutados, satisfactorios – 6 casos de prueba

Ejecutados, no satisfactorios -15 casos de prueba

No ejecutados, no satisfactorios -5 casos de prueba

Cobertura de pruebas (ejecutadas) = T^X/ RfT = 21/26 = 81%

Cobertura de pruebas (exitosas)= T^S/ RfT =7/26= 27%

3.4.3. Análisis de Defectos

No.	Descripción de la No Conformidad	Fuente (Opción de la Aplicación)	Prioridad	Severidad	Estado
1	Para la creación de un usuario no se puede realizar la búsqueda de personas pues se muestra una interfaz de error.	Configuración/ Usuarios/ Administrar Usuarios/Botón Crear.	Urgente	Crítica	Pendiente de arreglar por el equipo de desarrollo.
2	Para la modificación de los roles de un usuario se genera una interfaz de error al pulsar el botón Aplicar cambios para guardar los mismos.	Configuración/ Usuarios/ Administrar Usuarios/Botón Modificar.	Alta	Alta	Pendiente de arreglar por el equipo de desarrollo.

3	Para cambiar el estado de activación del usuario, se muestra una interfaz de error al intentar aplicar los cambios.	Configuración/ Usuarios/ Administrar Usuarios/Botón Cambiar Estado.	Alta	Alta	Pendiente de arreglar por el equipo de desarrollo.
4	Al pulsar el botón Buscar para efectuar la búsqueda de personas dado un criterio específico, se muestra una interfaz de error.	Búsquedas/Personas	Alta	Alta	Pendiente de arreglar por el equipo de desarrollo.
5	En la creación de una oficina se muestra una interfaz de error al pulsar el botón Aceptar.	Configuración/Oficinas/ Crear Oficinas.	Urgente	Crítica	Pendiente de arreglar por el equipo de desarrollo.
6	Para administrar los documentos Jurídicos, al seleccionar un tipo específico de organismo del Tipo de Documento Circular, aparecen todos los tipos de organismo.	Documentos/Documentos Jurídicos/Medidas de Prohibición	Normal	Media	Pendiente de arreglar por el equipo de desarrollo.
7	Se muestra una interfaz de error al insertar una medida de prohibición.	Documentos/Documentos Jurídicos/Medidas de Prohibición	Urgente	Crítica	Pendiente de arreglar por el equipo de desarrollo.

8	En el tracto documental se muestra una interfaz de error al pulsar el botón Buscar para realizar la búsqueda de los documentos jurídicos.	Documentos/Documentos Jurídicos/Medidas de Prohibición/Tracto Documental	Alta	Alta	Pendiente de arreglar por el equipo de desarrollo.
9	Al levantar una medida de prohibición, en el momento de referenciar un documento, se muestra una página de error al pulsar el botón Buscar para realizar la búsqueda de los documentos jurídicos.	Documentos/Documentos Jurídicos/Medidas de Prohibición/Documento Referenciado	Urgente	Crítica	Pendiente de arreglar por el equipo de desarrollo.
10	Al intentar actualizar los datos de una oficina determinada, se muestra una página de error.	Configuración/Oficinas/Actualizar Datos	Alta	Alta	Pendiente de arreglar por el equipo de desarrollo.
11	En el cambio de estado de una oficina, el sistema no valida la entrada de un código con número mayor de tres dígitos.	Configuración/Oficinas/Administrar Oficinas	Normal	Media	Pendiente de arreglar por el equipo de desarrollo.
12	Para cambiar el estado de una oficina	Configuración/Oficinas/Administrar Oficinas/Botón	Alta	Alta	Pendiente de arreglar por el equipo de desarrollo.

	determinada cuando se pulsa el botón Aceptar se muestra una interfaz de error y no se guardan los cambios.	Cambiar			
13	En la creación de plantillas de actos mercantiles cuando se da clic en el botón Salvar y no se selecciona el paso límite se muestra una interfaz de error.	Configuración/Parametrización/Plantillas de Actos/Botón Adicionar Plantillas	Alta	Alta	Pendiente de arreglar por el equipo de desarrollo.
14	En la creación de plantillas de actos mercantiles cuando se da clic en el botón Eliminar de un Concepto de Pago se muestra una interfaz de error.	Configuración/Parametrización/Plantillas de Actos/Botón Adicionar Plantillas	Alta	Alta	Pendiente de arreglar por el equipo de desarrollo.
15	Cuando se va a realizar la búsqueda de compañías y se da clic en el botón Buscar se muestra una interfaz de error.	Búsquedas/Personas Jurídicas	Alta	Alta	Pendiente de arreglar por el equipo de desarrollo.
16	Cuando se va a realizar la búsqueda de inmuebles se da clic en el botón Buscar y se muestra una interfaz de error.	Búsquedas/ Inmuebles.	Alta	Alta	Pendiente de arreglar por el equipo de desarrollo.

17	En todos los casos en la interfaz de error que se muestra, el botón Cancelar no funciona.	Interfaz de error	Normal	Media	Pendiente de arreglar por el equipo de desarrollo.

Después de haber clasificado cada defecto según la prioridad del mismo, se han encontrado 4 con prioridad urgente, 10 con alta prioridad y 3 con prioridad normal, tal como se muestra en la siguiente figura:

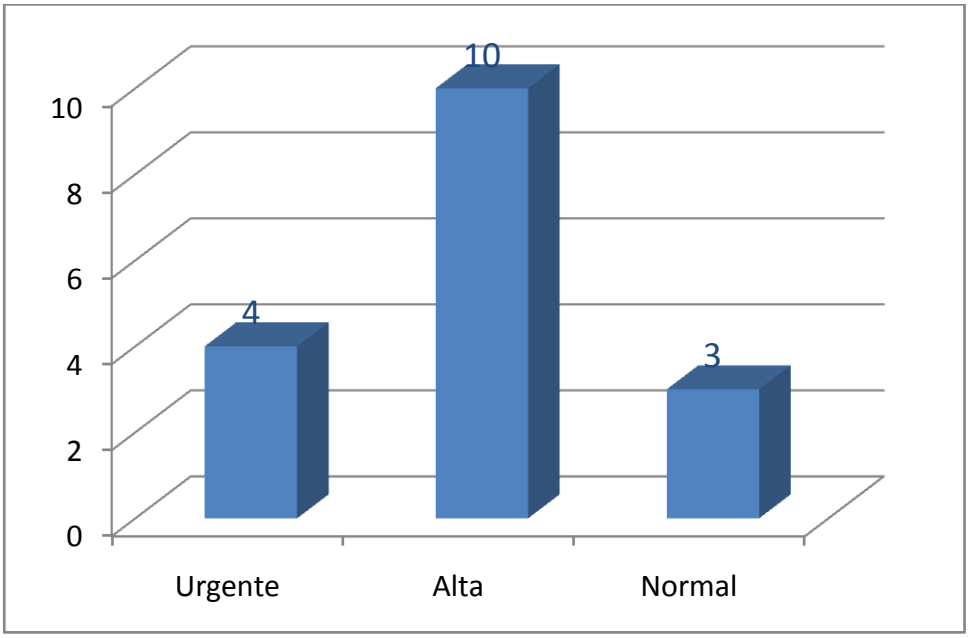

Figura 4. Clasificación de los defectos según la prioridad.

3.5. Conclusiones

Se ha arribado a las siguientes conclusiones:

- Los datos de prueba permiten al equipo de prueba interactuar con la aplicación de la forma más real posible, de forma que se encuentren y corrijan aquellos posibles errores que podían ser detectados por el usuario final.

- Se diseñaron los casos de prueba correctamente dando cumplimiento a cada requisito funcional definido en el plan.
- El software no es fiable pues tiene un gran número de defectos de alta prioridad, razón por la cual aumenta la probabilidad de que el sistema falle.
- Se estima que se debe llevar a cabo un proceso adecuado de depuración ya que la mayoría de los casos de pruebas que son ejecutados, no son satisfactorios y otros, a causa de los errores encontrados no han podido ser ejecutados. Este proceso es difícil ya que en muchas ocasiones no se puede encontrar el origen del defecto y del mismo depende, en gran medida, el aumento de los costos del proyecto, de ahí la importancia de prevenirlos.
- Los cinco casos de prueba que no pudieron ser ejecutados serán probados en el próximo ciclo de pruebas.
- Se considera conveniente clasificar los defectos ya que de esta forma se definen aquellos que son urgentes de corregir pues representan funcionalidades críticas del sistema y que además, afectan otras que son dependientes de éstas. Así los desarrolladores se centran en corregirlos de inmediato.

Conclusiones Generales

Luego de haber aplicado el proceso de pruebas propuesto, se ha arribado a las siguientes conclusiones:

- El proceso de pruebas se organizó de forma apropiada planificando los recursos disponibles, roles, cronograma, requerimientos a probar, procedimiento a seguir para la corrección de los errores y los entregables.
- Se diseñaron las pruebas de tipo funcional con la técnica de caja negra que cubrieron la mayor cantidad de combinaciones posibles para la detección de los defectos.
- Se compararon los resultados reales con los esperados donde la mayoría de los casos de prueba no fueron satisfactorios, de esta forma se evidenció que se encontraron más errores que en el ciclo anterior de pruebas, observando la mejoría de dicho proceso.
- Se aplicó correctamente el proceso de pruebas propuesto por RUP para el módulo Servicio Autónomo garantizando, de esta forma, la mejora de la calidad del software.

Recomendaciones

Luego de haber aplicado el proceso de pruebas en el módulo Servicio Autónomo en una de sus fases de desarrollo, se recomienda:

- Llevar a cabo un proceso adecuado para la detección y corrección de los errores por parte de los desarrolladores.
- Realizar el proceso de pruebas de la forma propuesta en todo el ciclo de vida del software.
- Hacer un estudio de las herramientas de automatización para pruebas funcionales que sean gratis y fáciles de usar.

Bibliografía

1. Wikipedia. [En línea] 20 de Noviembre de 2006. http://es.wikipedia.org/wiki/Ingeniería_de_software.

2. Los Antecedentes. [En línea] http://www.flexwiki.com/default.aspx/FlexWiki/LosAntecedentes.html.

3. **Pressman, Roger S.** *Ingeniería del Software, Un enfoque práctico.* 2005.

4. **Myers.** *The art of software testing.* 1979.

5. Diseño de la interfaz de usuario: Pruebas del software. [En línea] http://html.rincondelvago.com/diseno-de-la-interfaz-de-usuario_pruebas-del-software.html..

6. **Jacobson, Ivar.** *Proceso unificado del desarrollo del software.* 2003.

7. **Davis, A.** *201 Principles of Software Development.* 1995.

8. Profit.Consultoría/Testing de Software. [En línea] http://www.profit.es/21verifi.html#menu..

9. Introducción a la Programación y los Lenguajes Orientados a Objetos. [En línea] http://www.lsi.uned.es/lp/IntroPoo.pdf..

10. **Teruel, Alejandro.** Dificultades específicas a la prueba de software orientado a objetos. [En línea] 2 de Marzo de 2003. http://www.ldc.usb.ve/~teruel/ci4713/clases2001/dificultadesTOO.html..

11. **Hernán, Schenone Marcelo.** *Diseño de una Metodología Agil.* Buenos Aires : s.n., 2004.

12. *EN, EFICIENCIA DE LA CARACTERIZACIÓN DE TÉCNICAS DE PRUEBA. Eficiencia de la caracterización de técnicas de prueba en un contexto real de aplicación: ParqueSoft.*

13. **Meyer, Bertrand.** *Fundamentos de Ingeniería de Software. .*

14. **Villalba, Gurutze Miguel.** *Hacia la generación automática de pruebas para Jdeveloper a partir de especificaciones UML.* [En línea] Octubre de 2005. http://www.cuore.es/documentacion/congreso2005/ponencias/documentos/univeuropeamadrid.ppt.

15. **Ramírez, Ramón Alfonso.** *Interacción Humano Computadora y el proceso de desarrollo de software ¿Necesidad o Sugerencia?* 2006.

16. **Díaz, Ricardo Ruben Franco.** *Metodología para el desarrollo de aplicaciones orientadas a objeto. s.l. : myGnet.* 2006.

17. Monografías. [En línea] Noviembre de 2006. http://www.monografias.com/trabajos20/pruebas-de-software/pruebas-de-software.shtml..

18. **Luzuriaga, Juan Manuel.** Monografías.Inspecciones de software. [En línea] 1997. http://www.monografias.com/trabajos6/isof/isof.shtml..

19. Software Quality Systems S.A. [En línea] 2006. http://www.sqs.es/es/about/index.php..

20. **Elizondo, Perla Inés Velazco.** Prueba en el componente de software basado en el modelo Java Beans. [En línea] 2001. www.cs.man.ac.uk-~velascop-publ-Tesis1.mdi.

21. **Ivar Jacobson, Grady Booch, James Rumbaugh.** Ayuda Rational Unified Process. [En línea] 2003.

22. **Collado, Manuel.** *Pruebas de Software.* [http://lml.ls.fi.upm.es/ftp/ed2/0203/Apuntes/pruebas.ppt] Marzo, 2003.

23. **Isabel Blank, Larissa Herrera, Miguel Ortiz.** *Pruebas Funcionales.* [http://carolina.terna.net/ingsw3/datos/Pruebas_Funcionales.pdf] Mayo 2005.

24. **UAM.** *Pruebas.* [http://www.ii.uam.es/~is2/transparencias/pruebas.pdf]

25. *Técnicas de Verificación y Pruebas.* [http://trevinca.ei.uvigo.es/~rlaza/teoria/Verificacion.pdf]

26. **rbaeza@dcc.uchile.cl.** *Probando Software.* [http://www.dcc.uchile.cl/~rbaeza/inf/testeo.html]

27. *Pruebas de Sotware.* [http://lsi.ugr.es/~ig1/docis/pruso.pdf]

28. **Ros, Joaquin Nicolas.** *Capítulo 5. Pruebas de Software.* [http://dis.um.es/~jnicolas/09BK_FIS.html] 2006.

29. *Herramientas de Programación.* [http://boo.dif.um.es/Estudios/Asignaturas/Planes/Programas/down/ITIG/06BE.pdf]

30. **ELIZONDO, PERLA INÉS VELASCO.** *TRABAJO DE DIPLOMA: PRUEBA DE COMPONENTES DE SOFTWARE BASADAS EN EL MODELO JAVA BEANS .* [http://www.cs.man.ac.uk/~velascop/publ/Tesis.pdf] Abril 2001.

31. **Aplicada, ISCA- Ingeniería de Software y Calidad.** *Productos IBM Rational.* [http://www.isca.com.ve/productos.htm]

32. *Jornada sobre Testeo de Software.* [http://web.iti.upv.es/~squac/JTS/JTS2005/contenido.html] 2005.

33. **Pérez, Carlos García.** *JUnit4. Pruebas de Software Java.* [http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=junit4]

34. **Vicente, Jose María Toribio.** *JMeter.* [http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=jmeter] 2005-04-17.

35. *Programación Extrema-Wikipedia.* [http://es.wikipedia.org/wiki/Programaci%C3%B3n_Extrema] 2007.

36. **José H. Canós, Patricio Letelier yMª Carmen Penadés.** *Métodologías Ágiles en el Desarrollo de Software.* [http://www.willydev.net/descargas/prev/TodoAgil.Pdf]

37. **Martínez, Nelson Medinilla.** *En busca de respuestas para la ingeniería de software.* [http://is.ls.fi.upm.es/udis/docencia/proyecto/docs/FilosofiaIS.pdf] 2005.

38. *Guía de Seguridad de Windows Server 2003.* [http://www.microsoft.com/spain/technet/security/prodtech/windowsserver2003/w2003hg/s3sgapxd.mspx] 27/12/05.

39. **Biraj Rath, NIIC.** *Microsoft Enterprise Services Preparación técnica para el comercio electrónico.* [http://www.microsoft.com/latam/technet/articulos/200201/art08/] 2000.

40. *SQS consolida su actividad de testeo de calidad de software.* [http://www.sqs.es/archivos/estrategia_1_15nov05.pdf]

41. *Rational Unified Process(RUP).* [http://64.233.167.104/search?q=cache:QudLw8VatakJ:https://pid.dsic.upv.es/C1/Material/Documentos%2520Disponibles/Introducci%C3%B3n%2520a%2520RUP.doc+historia+RUP&hl=es&ct=clnk&cd=1&gl=cu]

Glosario

D

Documento jurídico:

Es el documento redactado por un abogado, donde se refleja cuál es la operación que se va a realizar sobre un inmueble, se especifican los datos del mismo inmueble y de las personas involucradas en la operación, que se lleva a la oficina para ser registrado. El usuario lo presenta para ingresarlo al proceso registral, de él emana toda la información que comprobará el abogado revisor y el funcionario de prohibiciones chequeará las posibles prohibiciones que pesen sobre las personas involucradas en la operación o sobre el inmueble.

Documento Referenciado:

Es el documento al que el usuario le va a suspender una prohibición o una exención.

E

Exención:

Figura establecida en las Leyes de la República mediante la cual el estado otorga ventaja a los beneficiados en el no pago total o parcial de impuestos, tasas o contribuciones.

Entorno de Ejecución de Java (JRE)

Un programa destinado a la Plataforma Java necesita dos componentes en el sistema donde se va a ejecutar: una máquina virtual de Java (JVM), y un conjunto de librerías para proporcionar los servicios que pueda necesitar la aplicación. La JVM que proporciona Sun Microsystems, junto con su implementación de las librerías estándar, se conocen como Java Runtime Environment (JRE) o Entorno en tiempo de ejecución para Java. El JRE es lo mínimo que debe contener un sistema para poder ejecutar una aplicación Java sobre el mismo. Para el desarrollo de programas se ofrece un paquete de utilidades y herramientas conocido como JSDK (Java Software Development Kit).

I

Inmueble:

Son todas aquellas posesiones como casas o fincas que es imposible de trasladar sin ocasionar daños a los mismos, porque forman parte de un terreno o están ancladas a él.

P

Personas Naturales:

A efectos de la obligación tributaria, se consideran personas naturales a los seres humanos sujeto de derechos y obligaciones, las sociedades conyugales, las sucesiones indivisas y las herencias yacentes.

Personas Jurídicas:

Ser o entidad capaz de derechos y obligaciones aunque no tiene existencia individual física; como las corporaciones, asociaciones, sociedades y fundaciones.

Prohibición:

Impedimento judicial, medida cautelar, que impide realizar una operación inmobiliaria, ya sea porque pese sobre el la persona o sobre el inmueble.

R

Recaudos:

Documentos, comprobantes, avales, certificaciones, constancias, etc. que deben acompañar a los documentos a la hora de presentarlos, para conferirle valor legal al proceso y respaldar las operaciones contenidas en el mismo.

S

Suspensión:

Medida que levanta, anula, termina una prohibición existente.

Servicio Autónomo:

Se refiere a la oficina de registro.

T

Tracto Documental:

Documentos a referenciar a la prohibición o exención en curso.