

**Universidad de las Ciencias Informáticas
Facultad 3**



**Título: Pruebas de Caja Negra al Módulo Administración
Financiera del Sistema de Registros y Notarías**

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO INFORMÁTICO.

Autor: Nemury Silega Martínez.

Tutor: Ing. Michael González Jorrín.

Consultante: Ing. Irina Napal Torres.

**Caracas, Distrito Federal, Venezuela.
Mayo, 2007.**

AGRADECIMIENTOS

A Michael por su apoyo en todo momento...

A Irina por su ayuda cuando más lo necesitaba.....

A Rolan por su ayuda y comprensión.....Muchas gracias.....por la madrugadera.....

A Ameirys, Luis y Blanco por su labor titánica arreglando el documento, sin ustedes hubiese sido imposible.....además por darme aliento cuando pensaba que no se podía.

A Luis y Ameyrys por mantener el núcleo unido y ayudarme cuando lo necesité, y por cocinar bien.....

A Dayma por su apoyo y ayuda cuando estaba a punto de colapsar.....

A todos los profesores que he tenido, por ayudarme y por obligarme a esforzarme.....

A todos los compañeros que colaboraron en la realización de este trabajo.....

DEDICATORIA

Este trabajo está dedicado

Especialmente a la memoria de mi madre quien ha sido mi inspiración en todo momento y no ha dejado de estar un segundo en mi corazón y en mi mente..... Te extraño mucho.....

A mi hermano, que lo quiero con toda mi alma y mi corazón.....

A mi tía Julita quien ha sido mi gran apoyo desde..... y a la que quiero con todo mi alma y mi corazón.

A mi padre que lo quiero con todo mi corazón y siempre ha estado cuando lo he necesitado para darme su apoyo.....

A toda mi familia que siempre me ha apoyado..... Los quiero mucho.

A todos mis amigos, que han compartido las cosas buenas y las malas a mi lado.....

A mi queridísima Dana, por su comprensión, su cariño, su apoyo.....Te quiero muchísimo.....
Sin tí no hubiese sido igual.....Un beso eterno para tí

RESUMEN

En el presente trabajo se diseña la propuesta de un Plan de Pruebas que sirva de guía a las Pruebas de Aceptación del Módulo Administración Financiera del Sistema de Registros y Notarías. En este Plan de Pruebas se recogen todos los elementos que se deben tener en cuenta a la hora de realizar las mismas. En el trabajo se hace un recorrido por todos los elementos y conceptos más importantes relacionados con las Pruebas de Software. Se lleva a cabo una caracterización del Plan de Pruebas propuesto y se comprueba que contiene todos los elementos necesarios mediante una Lista de Comprobación. También se hace una descripción de las Pruebas realizadas a Administración Contable con el fin de aplicar algunos elementos contenidos en el Plan de Pruebas y de esta forma validar su eficacia, debido a que ambos Sistemas tienen una gran similitud. Mediante estas Pruebas se demostró la factibilidad de algunos elementos del Plan de Pruebas debido a que permitieron que las Pruebas desarrolladas se llevaran a cabo satisfactoriamente, logrando los objetivos definidos. De manera general lo más importante que se persigue con el Plan de Pruebas propuesto es que guíe el desarrollo de las Pruebas de Aceptación para demostrar que el Sistema cumple con todos los Requerimientos establecidos entre los Clientes y el equipo del proyecto.

TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO 1: Fundamentación teórica.....	5
1.1. Definición de Pruebas de Software.....	5
1.1.1. ¿Cuál es el objetivo de la Prueba?	6
1.1.2. Diseño de Casos de Prueba	7
1.1.3. Rasgos distintivos y principios.	9
1.1.4. ¿Cuán difícil puede ser una demostración?.....	10
1.1.5. ¿Qué atributos definen a una buena Prueba?.....	11
1.2. Formas de probar un Software	12
1.2.1. ¿Pueden ser exhaustivas las Pruebas de Caja Blanca?	13
1.3. Pruebas de Caja Negra.....	14
1.3.1. ¿Que se persigue con las Pruebas de Caja Negra?	15
1.3.2. ¿Cómo ejecutar las Pruebas de Caja Negra?	16
1.4. Estrategias de Prueba de Software	18
1.5. Fases de las Pruebas.	18
1.6. ¿Cuándo terminar las Pruebas?.....	22
1.7. Plan de Pruebas.....	23
1.8. Las Pruebas en la actualidad.....	25
Conclusiones.	28
CAPÍTULO 2: Descripción del Plan de Pruebas.....	29
2.1. Introducción.....	29
2.2. Objetivos del Plan de Pruebas.	29
2.3. Objetivos de las Pruebas.....	29
2.4. Alcance de las Pruebas.....	30
2.5. Estrategia de evolución del Plan	31
2.6. Roles y responsabilidades.	31
2.7. Escenario de Pruebas.....	32
2.8. Recursos del Sistema.	33
2.9. Recursos de Software.	34
2.10. Requerimientos a probar.....	35
2.11. Estrategia de las Pruebas de Aceptación.....	35
2.12. Descripción del flujo de trabajo.....	35
3. Descripción de las Estrategias y tipos de Pruebas.....	37
3.1. Criterios de aceptación del producto y evaluación de las Pruebas.....	43
3.2. Evaluación de las Pruebas.....	45
3.3. Artefactos de entrega.....	45
3.4. Cronograma	46
3.5. Diseño de los Casos de Prueba.....	46

3.6. Elementos a tener en cuenta en las Pruebas.....	52
3.7. ¿Cómo se asegurará la realización de unas buenas Pruebas?.....	54
Conclusiones.	56
CAPÍTULO 3: Aplicación del Plan de Pruebas.....	57
3.1. Recursos utilizados y escenario de Pruebas.	57
3.2. Ejecución de las Pruebas.....	58
3.3. Resultado de las Pruebas.	62
3.4. Evaluación de Administración Contable	64
3.5. Resumen de las Pruebas realizadas	65
3.6. Evaluación del Plan de Pruebas.	65
Conclusiones	67
CONCLUSIONES.	68
RECOMENDACIONES.....	69
GLOSARIO.....	70
BIBLIOGRAFÍA.....	72
ANEXOS	73

ÍNDICE DE FIGURAS

Figura 1: Representación de Pruebas de Caja Blanca y negra.....	13
Figura 2: Fases de las Pruebas.	19
Figura 3: Curva predecida.	23
Figura 4: Escenario de Prueba	33
Figura 5: Flujo de Trabajo	36
Figura 6: <i>Escenario de Pruebas</i>	58

INTRODUCCIÓN.

Con el desarrollo y difusión que en la actualidad han alcanzado la informática y las tecnologías de la información cada vez es mayor la cantidad de Sistemas que son automatizados en todos los países y, por supuesto, la Contabilidad no está exenta de esta tendencia. El área de la Contabilidad y las Finanzas es una de las más importantes de una empresa, como es Planteado en varias bibliografías y es evidenciado en la vida diaria en cualquier entidad tanto de producción como de servicio.

Con la automatización de los Sistemas Contables y Financieros, se les incorpora rapidez y mejoras en el almacenamiento de la información; las Cuentas se realizarán de manera certera y podrán ser consultadas en cualquier momento teniendo incluso históricos de esos procesos; Aunque no debe perderse de vista que en este tipo de aplicación, como en cualquier otro, debe ir de la mano el tema de la detección y erradicación de los errores que se puedan cometer durante el desarrollo de la aplicación. Según Pressman: *“Debido a la incapacidad humana de trabajar y comunicarse de forma perfecta, el desarrollo de Software debe ir acompañado de una actividad que garantice la Calidad”* (Pressman 1998).

Teniendo en cuenta que la Prueba de Software es: “...comprobar la respuesta de un Sistema a los estímulos y comparar esa respuesta a un estándar. Dado un cierto Software y una Lista de las funciones que se supone que haga, descubrir si las realiza de la forma que se describe. Adicionalmente, descubrir si hace otras cosas que no se describan...” (Hutcheson 2003), queda bien determinado que las Pruebas constituyen un elemento relevante para certificar la Calidad de lo que se automatiza. Para mitigar las situaciones de desavenencias entre lo que debía ser y lo que se automatiza es necesaria la realización de Pruebas de Software, las cuales deben realizarse durante y al finalizar la producción del mismo, antes de ponerse en uso por sus usuarios finales.

El Servicio Autónomo de Registros y Notarías, se encuentra automatizando todos los Procesos Registrales de los Registros Públicos y Mercantiles, está previsto que se diseñe un Módulo de Administración Financiera. Debido a la importancia del área donde se aplica es necesario que el Módulo sea probado y de esa forma validar el cumplimiento de los Requisitos Funcionales y No

Funcionales que se establecieron previamente por los Clientes. Estas Pruebas contribuyen al mejoramiento del Sistema para el momento en que sea usado por los usuarios finales y a minimizar la aparición de fallas que conlleven a consecuencias negativas como por ejemplo: pérdida de información, desvío de recursos, pérdidas materiales, entre otras, pero cuando se realizan dichas Pruebas de Aceptación si no son debidamente Planificadas puedes que se manifiestan algunos problemas que impidan que la Pruebas puedan cumplir sus objetivos, dentro de los problemas están: la insuficiencia de recursos para llevar a cabo las Pruebas, el personal no es el indicado o no conoce cuál es su labor, no existe una Estrategia a seguir, desorganización durante la realización de las Pruebas, no se conocen cuales son los entregables, no se cuenta con un calendario que permita ir controlando la evolución de las Pruebas.

Teniendo en cuenta esta situación se ha determinado la existencia del siguiente Problema Científico: ¿Cómo realizar un Plan que guíe el proceso de Pruebas de Aceptación y permita comprobar que el Módulo de Administración Financiera del Sistema de Registros y Notarías cumple con los Requisitos definidos en el análisis del propio Sistema?

Estableciendo como Objeto de Estudio, las Pruebas de Software al Módulo de Administración Financiera y basando esta investigación en el Campo de Acción de las Pruebas de Aceptación del Módulo de Administración Financiera.

Para resolver este Problema Científico la presente investigación tiene como Objetivo General: el diseño de un Plan de Pruebas de Administración Financiera y para darle cumplimiento a dicho objetivo se han definidos los Objetivos Específicos siguientes:

- Definir el alcance y objetivo de las Pruebas de Administración Financiera.
- Realizar una propuesta para el flujo de trabajo, la Planificación del tiempo y recursos necesarios para las actividades de Prueba.
- Validar una parte del Plan de Pruebas propuesto, a través de su aplicación a un Módulo con características similares para verificar su efectividad, así como la factibilidad de usar el procedimiento y las técnicas de Pruebas propuestas en Administración Financiera.
- Analizar las Pruebas realizadas a Administración Contable para verificar la factibilidad del uso de los métodos y las técnicas empleadas para su aplicación en las Pruebas a Administración

Financiera, así como para hacer estimaciones sobre el tiempo, los recursos y los datos de los Casos de Pruebas.

Definiendo como Hipótesis que con el diseño de un Plan de Pruebas de Aceptación adecuado que presente las técnicas necesarias y guíe las Pruebas solicitadas para demostrar que el Módulo de Administración Financiera del Sistema de Registros y Notarías cumple con los Requerimientos definidos entre los Clientes y el equipo de desarrollo de dicho Software.

Para el logro de los objetivos Planteados en este trabajo se proponen las tareas de investigación siguientes:

- Realizar un estudio de los elementos fundamentales sobre las Pruebas de Aceptación de un Software.
- Realizar un estudio de los elementos que constituyen un Plan de Pruebas de Aceptación.
- Estudiar la documentación existente sobre Administración Financiera para identificar sus particularidades y recopilar elementos necesarios para el diseño del Plan de Pruebas necesario.
- Definir cada uno de los elementos para el Plan de Pruebas de Aceptación del Módulo de Administración Financiera.
- Diseñar los Casos de Pruebas necesarios para Administración Contable.
- Aplicar las Pruebas al Módulo de Administración Contable.
- Evaluar los resultados obtenidos en las Pruebas realizadas a Administración Contable.
- Establecer elementos coincidentes entre las Pruebas realizadas al Módulo de Administración Contable y las Pruebas necesarias a desarrollar en el Módulo de Administración Financiera, para determinar aquellos elementos aplicables a esta última.
- Proponer las recomendaciones para la aplicación del Plan de Pruebas en Administración Financiera en función de los elementos coincidentes detectados.

El siguiente trabajo se encuentra estructurado como se describe a continuación:

Capítulo 1: Se realiza un análisis de los conceptos fundamentales relacionados con las Pruebas de Software. Además se hace un análisis de todos los elementos que constituyen un Plan de Pruebas.

Capítulo 2: Se describe el Plan de Pruebas diseñado y cada una de las partes que lo componen. Además se describen aquellas Pruebas realizadas al Modulo de Administración Contable y los Casos de Pruebas diseñados.

Capítulo 3: Se realiza un análisis de los resultados de las Pruebas realizadas al Módulo de Administración Contable. Además se presenta una valoración de la factibilidad de los algunos elementos definidos en el Plan de Pruebas y aplicados a las Pruebas de Administración Contable.

CAPÍTULO 1: Fundamentación teórica.

En el desarrollo de este capítulo se abordarán los principales elementos y conceptos relacionados con las Pruebas de Software, y se analiza la actualidad de las Pruebas en el mundo. Se definen todos los elementos que constituyen un Plan de Pruebas así como los tipos de Pruebas que se desarrollan según el momento de realizarlas.

1.1. Definición de Pruebas de Software

“Nunca se dará suficiente importancia a la Prueba del Software y sus implicaciones en la Calidad del Software” (Pressman 1998). Esta frase tiene total vigencia en la actualidad debido a que una buena parte de las personas que están dentro del desarrollo de Software coinciden en que las Pruebas tienen gran importancia, pero en la práctica no se le da la prioridad requerida. Durante el desarrollo de Software se realizan actividades de producción donde son grandes las posibilidades de que aparezcan errores humanos. Los errores pueden empezar a darse desde el principio del proceso, en el que los objetivos pueden estar identificados de forma errónea o imperfecta, así como posteriores pasos de diseño y desarrollo. Debido a la incapacidad humana de trabajar y comunicarse de forma perfecta, el desarrollo de Software debe ir acompañado de una actividad que garantice la Calidad (Pressman 1998). Las Pruebas tienen un papel preponderante en el aseguramiento de la Calidad del Software ya que es la actividad encargada de demostrar que el mismo hace de la manera correcta todo lo que espera el Cliente.

A continuación se proponen algunos conceptos de Prueba:

1. Comprobar la respuesta de un Sistema a los estímulos y comparar esa respuesta a un estándar. Evaluar la Calidad de la respuesta con respecto a la estándar. Dado un cierto Software y una Lista de las funciones que se supone que haga, descubrir si las realiza de la forma que se describe. Adicionalmente, descubrir si hace otras cosas que no se describan (Hutcheson 2003)
2. Las Pruebas del Software son el proceso de ejecución de un programa con la intención de descubrir un error (Pressman 1998).

3. IEE define las Pruebas como una actividad en la cual un Sistema o componente es ejecutado bajo condiciones específicas, se observan o almacenan los resultados y se realiza una evaluación de algún aspecto del Sistema o componente (IEEE 1991).

Lo antes mencionado es una representación de los conceptos manejados en el mundo del desarrollo de Software. El desarrollo de este trabajo se centrará en la definición elaborada por Pressman, se le puede adicionar que la Prueba de Software es un elemento crítico para la garantía de la Calidad del Software y representa una revisión final de las especificaciones, del diseño y de la codificación.

1.1.1. ¿Cuál es el objetivo de la Prueba?

Para poder comprender el objetivo de las Pruebas de Software a continuación se proponen varios conceptos y se hace un breve análisis de cada uno:

El propósito de la Prueba de Software es asegurar que los Sistemas de Software funcionen como se espera cuando sean usados por Clientes y usuarios (Tian 2005)

La dificultad que tiene este concepto es que la Prueba no es la responsable de que el Software funcione como esperan los Clientes, la Prueba no puede asegurar la ausencia de defectos, sólo puede demostrar que existen defectos en el Software (Pressman 1998) es decir que las Pruebas lo que pueden hacer es demostrar si cumple o no con lo esperado por los Clientes.

La siguiente definición enuncia mejor cual es el objetivo de las Pruebas:

- Un buen Caso de Pruebas es aquel que tiene una alta probabilidad de mostrar un error no conocido hasta entonces.
- Una Prueba tiene éxito si descubre un error no detectado hasta entonces (Pressman 1998).

Con estos elementos se define cual es el objetivo fundamental de las Pruebas de Software, el siguiente enuncia algunas actividades que pueden considerarse como parte de los objetivos de las Pruebas:

- Encontrar y Documentar defectos en la Calidad de Software.

- aconsejar sobre la Calidad percibida del Software.
- Validar y Probar las asunciones hechas en el diseño y especificación de requerimientos a través de una demostración concreta (RUP 2003).

Todo lo Planteado anteriormente es muy importante ya que actualmente es muy común que los Sistemas de Software formen parte como un elemento más de otros Sistemas, y cualquier fallo puede traer costosas consecuencias (materiales e incluso humanas), esto motiva a crear Pruebas minuciosas y bien Planificadas. Existen organizaciones de desarrollo de Software que emplean entre el 30 y 40 por ciento del tiempo de un proyecto en las Pruebas (RUP 2003). En casos críticos como el control de tráfico aéreo, control de reactores nucleares, entre otros. Las Pruebas pueden costar de 3 a 5 veces más que el resto de los pasos de la Ingeniería del Software juntos (Pressman 1998). Cuando queda algún error sin descubrir durante las Pruebas en estos casos críticos las consecuencias pueden ser catastróficas, por esto la necesidad de darle a las Pruebas la importancia que requiere.

Tomando como referencia los objetivos Planteados anteriormente se pudiera concluir que la Prueba va a tener éxito si descubre errores en el Sistema. Como ventaja secundaria las Pruebas demuestran hasta qué punto el Sistema cumple con las especificaciones y los Requisitos de Rendimiento y Seguridad implícitos. Además los datos que se van recogiendo durante la Prueba dan una medida de la fiabilidad del Software y de alguna manera una indicación de la Calidad del Sistema (Pressman 1998). También es una buena oportunidad que tiene el equipo de desarrollo para que se demuestre cuán potente es el Software, y que los defectos encontrados se puedan corregir antes de que el Cliente comience a usar el Sistema.

1.1.2. Diseño de Casos de Prueba.

Caso de Pruebas es una condición a ser probada que incluye su propia identificación y la respuesta esperada (Hutcheson 2003). Un diseño de los Casos de Prueba es fundamental para el éxito de las Pruebas.

Los Casos de Pruebas definen un conjunto específico de entradas de Pruebas, ejecución de condiciones y resultados esperados (RUP 2003).

Otro concepto de Caso de Prueba es: un caso de Prueba especifica una forma de probar el Sistema, incluyendo la entrada o resultado con la que se ha de probar y las condiciones bajo las que ha de probarse (Jacobson, Booch et al. 1999).

La última definición es muy adecuada para Caso de Prueba porque a diferencia del primer concepto, también tiene en cuenta las condiciones en las cuales se Prueba el Sistema. En este libro también se definen algunos Casos de Pruebas comunes que se aplican en la práctica:

- Probar un caso de uso o un escenario específico del mismo. Un caso de Prueba de este tipo incluye la verificación del resultado de la interacción entre los actores y el Sistema, que se satisfacen las precondiciones y poscondiciones especificadas por el caso de uso.
- Probar la realización de caso de Uso-Diseño o un escenario específico de la realización. Un Caso de Prueba de este tipo puede incluir la verificación de la interacción entre los componentes que implementan dicho Caso de Uso (Jacobson, Booch et al. 1999).

El primer caso se refiere a las Pruebas del tipo de Caja Negra y el segundo al de Caja Blanca, más adelante se abundará sobre estos tipos de Prueba de Software.

Los Casos de Pruebas son muy importantes para las Pruebas, por eso se deben documentar y revisar bien: Documentando los Casos de Pruebas permiten que sean revisados para que sean completos y correctos. Esto adquiere mayor relevancia cuando el Sistema es muy complejo y donde las entradas, condiciones de ejecución y resultados esperados son particularmente complejos (RUP 2003).

Como se ha mencionado anteriormente las Pruebas del Software o de otros productos puede requerir tanto esfuerzo como el diseño inicial, pero la mayoría de las veces esto es obviado por los Ingenieros de Software y realizan Diseños de Casos de Prueba que parezcan adecuados pero que tienen pocas posibilidades de encontrar errores y no tienen en cuenta que: las Pruebas son solamente tan buenas como los Casos de Prueba (Lewis 2005), es decir que si no se diseñan buenos Casos de Prueba, la Prueba va a tener pocas posibilidades de tener éxito, o sea de encontrar nuevos errores.

1.1.3. Rasgos distintivos y principios.

Se debe tener en cuenta que *“la Prueba no puede asegurar la ausencia de defectos, sólo puede demostrar que existen defectos en el Software”* (Pressman 1998). Esa afirmación es de gran relevancia porque aunque no se encuentren defectos en la Prueba no quiere decir que el Sistema esté libre de ellos, por lo tanto la presencia de defectos no puede tomarse como responsabilidad del equipo de Prueba, no obstante la misma debe ser vista como una oportunidad del equipo de desarrollo para demostrar que el Software desarrollado cumple con las especificaciones Planteadas por el Cliente.

Para poder desarrollar un adecuado proceso de Pruebas es muy útil conocer algunos de sus principios enunciados en (Pressman 1998):

- A todas las Pruebas se le debería poder hacer un seguimiento hasta los Requisitos del Cliente. Como se ha visto el principal objetivo es encontrar errores. Para el Cliente los errores más graves son los que le impiden al Sistema cumplir sus Requisitos.
- Las Pruebas deberían Planificarse mucho antes de que empiecen. La Planificación de las Pruebas pueden comenzar tan pronto como esté completo el Modelo de Requisitos. La definición detallada de los Casos de Prueba puede empezar una vez que se haya aprobado el modelo de diseño. Por tanto, se puede Planificar y diseñar todas las Pruebas antes de generar ningún código.
- El principio de Pareto es aplicable a la Prueba del Software. El principio de Pareto implica que al 80 por ciento de todos los errores descubiertos durante las Pruebas surgen al hacer un seguimiento de sólo el 20 por ciento de todos los Módulos del programa. El problema está en aislar estos Módulos sospechosos y probarlos.
- Las Pruebas deberían empezar por lo pequeño y progresar hacia lo más grande. Las primeras Pruebas Planeadas y ejecutadas en general se centran en Módulos individuales del programa y a medida que avanzan las Pruebas, se concentran en encontrar errores en grupos integrados de Módulos y finalmente al Sistema entero.

- No son posible las Pruebas exhaustivas. Incluso en un programa pequeño la cantidad de permutaciones de caminos es muy grande, por lo que es imposible cubrir todas las combinaciones de caminos. Sin embargo es posible cubrir adecuadamente la lógica del programa y asegurarse de que se han aplicado todas las condiciones del diseño procedimental.
- Para ser más efectivas, las Pruebas deberían ser conducidas por un equipo independiente. Se ha demostrado que el Ingeniero de Software que desarrolló el Sistema no es el más indicado para realizar las Pruebas al Sistema.

Una Prueba que cumpla con estos principios tiene una alta probabilidad de cumplir el objetivo fundamental, que es encontrar errores y es muy conveniente chequear hasta qué punto se manifiestan estos principios en la Planificación y desarrollo de las Pruebas, para lograrlo se pueden utilizar las Listas de Comprobación.

1.1.4. ¿Cuán difícil puede ser una demostración?

Es muy útil que cuando se desarrolle un Software se piense hasta qué punto es fácil de demostrar que el Software hace de la manera correcta todo lo que debe hacer. Este tema se conoce como facilidad de la Prueba, Pressman formula una definición que puede ayudar a entender en qué consiste:

“La facilidad de la Prueba de Software es simplemente lo fácil que se puede probar un programa de computadora. Como la Prueba es tan profundamente difícil, merece la pena saber qué se puede hacer para hacerlo más sencillo. A veces los programadores no están dispuestos a hacer cosas que faciliten el proceso de Prueba y una Lista de Comprobación de posibles puntos de diseño, características, etc., puede ser útil a la hora de negociar con ellos” (Pressman 1998)

Debido a la relevancia que cobra, existen incluso métricas para chequear cuán fácil puede ser un Software de probar, esto lo propone (Pressman 1998):

- *Operatividad:* Cuanto mejor funcione, más eficientemente se puede probar. Se logra cuando el Sistema tiene pocos errores, porque los errores añaden sobrecarga de análisis y generación de informes al proceso de Prueba. Es importante también que no existan errores que

bloqueen la ejecución de las Pruebas y que el Producto evolucione en fases funcionales o sea que permita un desarrollo o Pruebas simultáneos.

- *Observabilidad:* Lo que ves es lo que Pruebas.
- *Controlabilidad:* Cuanto mejor se pueda controlar el Software, más se puede automatizar y optimizar. Se cumple cuando todos los resultados posibles se pueden generar a través de algunas combinaciones de entrada, los ejecutores de la Prueba pueden controlar directamente los estados y las variables del Hardware y del Software. Los formatos y resultados son consistentes y estructurados.
- *Capacidad de descomposición:* Controlado el ámbito de las Pruebas se puede aislar más rápidamente los problemas y llevar a cabo mejores Pruebas de Regresión. Se logra cuando el Sistema esté construido con Módulos independientes y estos se puedan probar independientemente.
- *Simplicidad:* Cuanto menos haya que probar, más rápidamente podremos probarlo. Debe cumplir con la simplicidad funcional, estructural de código.
- *Estabilidad:* Cuanto menos cambios, menos interrupciones a las Pruebas. Los cambios del Software deben ser infrecuentes y controlados además que no invaliden las Pruebas y que el Software se recupere bien de los fallos.
- *Facilidad de comprensión:* Cuanta más información tengamos, más inteligentes serán las Pruebas. El personal que realizará la Prueba debe entender perfectamente el diseño, así como las dependencias entre los componentes internos, externos y compartidos. Se deben comunicar los cambios del diseño y la documentación debe ser accesible instantáneamente así como bien organizada, detallada, específica y exacta.

Todo lo mencionado anteriormente debe ser usado por el equipo de desarrollo para desarrollar un Sistema que no sea difícil de comprobar si cumple con los Requerimientos por los cuales fue desarrollado.

1.1.5. ¿Qué atributos definen a una buena Prueba?

Todos los investigadores le atribuyen diferentes características para las Pruebas, la mayoría coinciden en qué una Prueba es adecuada si tiene los siguientes atributos, definidos en (Pressman 1998):

- *Alta probabilidad de encontrar un error:* El responsable de las Pruebas debe entender el Software e intentar desarrollar una imagen mental de cómo podría fallar el Sistema.
- *No puede ser redundante:* Es significativo porque a veces se piensa que la Prueba debe cubrir todo el Sistema sin comprender lo costoso que puede ser esto. Por eso no se debe realizar una Prueba que tenga el mismo propósito que otra.
- *La mejor de la cosecha:* Debido a las limitaciones de tiempo y recursos, cuando existen un grupo de Pruebas con propósito similar se debe escoger la que tenga la más alta probabilidad de descubrir una clase entera de errores.
- *Ni demasiado sencilla ni demasiado compleja:* Aunque se pueden combinar una serie de Pruebas en un Caso de Prueba, los posibles efectos secundarios de este enfoque puede enmascarar errores. Por eso cada Prueba debe realizarse de forma separada.

Todos estos elementos son muy significativos y deben ser chequeados constantemente para comprobar que el proceso de Prueba que se sigue es el correcto.

1.2. Formas de probar un Software.

Cualquier producto de Ingeniería puede probarse mediante las Pruebas de Caja Blanca y Caja Negra, las que se definen como:

- *Caja negra:* Conociendo la función específica para la que fue diseñado el producto, se pueden llevar a cabo Pruebas que demuestren que cada función es completamente operativa, y al mismo tiempo buscando errores en cada función (Pressman 1998).
- *Caja blanca:* Conociendo el funcionamiento del producto, se pueden desarrollar Pruebas que aseguren que todas las piezas encajan, o sea, que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada (Pressman 1998).

La siguiente figura muestra en qué consisten.

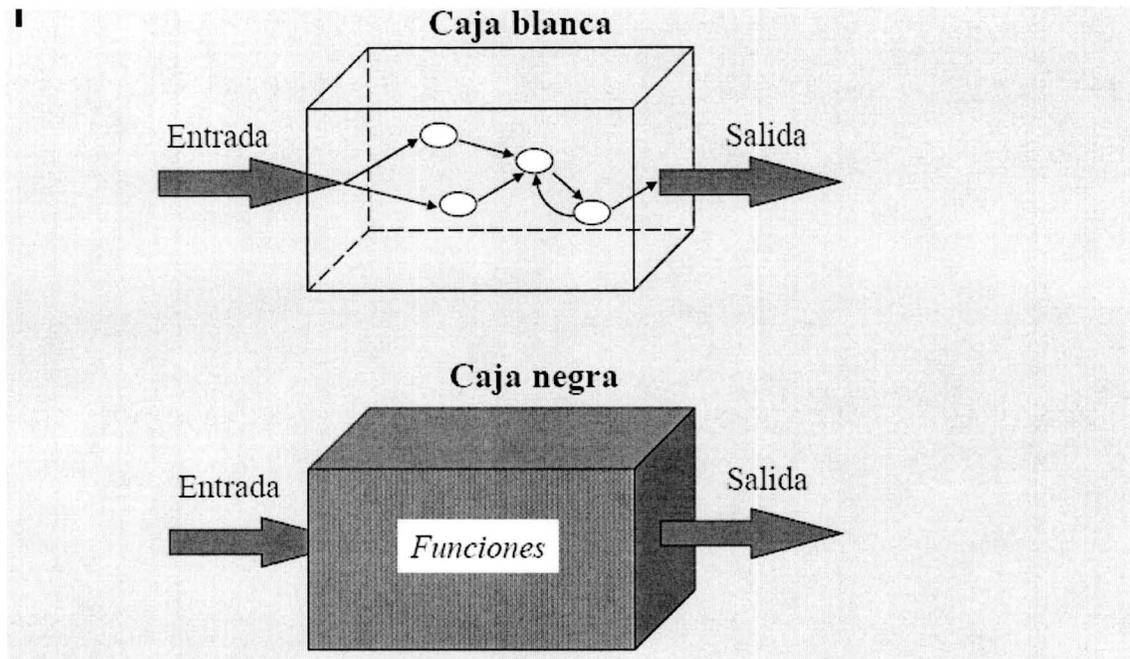


Figura 1: Representación de Pruebas de Caja Blanca y negra.

1.2.1. ¿Pueden ser exhaustivas las Pruebas de Caja Blanca?

La Prueba de Caja Blanca se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del Software proponiendo Casos de Pruebas que ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para detectar si el estado real coincide con el esperado o mencionado.

Se podría pensar que mediante las Pruebas de Caja Blanca se podrían obtener programas libre de errores, solo habrían que realizar Casos de Pruebas que examinen todos los caminos lógicos del programa exhaustivamente, el único problema es que se estaría violando uno de los principios de la Pruebas y de hecho es humanamente imposible, en el ejemplo se ilustra que sucedería si se intentara hacer esto:

Considere un programa de 100 líneas de código en lenguaje C. Después de la declaración de algunos datos básicos, el programa contiene dos bucles que se ejecutan de 1 a 20 veces cada uno, dependiendo de las especificaciones en la entrada. Dentro del bucle interior se necesitan cuatro

instrucciones `if - then - else`. Existen aproximadamente 104 posibles caminos que se pueden ejecutar en este programa.

Para poner de manifiesto el significado de este número, suponiendo que se ha desarrollado un procesador de Pruebas Mágico (mágico porque no existe tal procesador) para hacer una Prueba exhaustiva. El procesador puede desarrollar un caso de Pruebas, ejecutarlo y evaluar los resultados en un milisegundo. Trabajando las 24 horas del día, 365 días al año, el procesador trabajaría durante 3170 años para probar el programa. Esto irremediablemente causaría estragos en la mayoría de los Planes de desarrollo. La Prueba exhaustiva es imposible para los grandes Sistemas (Pressman 1998).

Queda demostrada la imposibilidad de realizar Pruebas de Caja Blanca exhaustivas, sin embargo no quiere decir que las Pruebas de Caja Blanca son impracticables. Se puede elegir y ejercitar una cantidad de caminos lógicos importantes. Se pueden comprobar las estructuras de datos más importantes para ver su validez y combinándose con el método de Caja Negra los resultados pueden ser satisfactorios.

1.3. Pruebas de Caja Negra.

Cuando se considera el Software de computadora, la Prueba de Caja Negra se refiere a las Pruebas que se llevan a cabo sobre la interfaz del Software. O sea, los Casos de Prueba pretenden demostrar que las funciones del Software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene. Una Prueba de Caja Negra examina algunos aspectos del modelo fundamental del Sistema sin tener mucho en cuenta la estructura lógica interna del Software (Pressman 1998).

El concepto enunciado da una definición muy completa de las Pruebas de Caja Negra, coincide en gran medida con lo Planteado por otros autores, por ejemplo en la siguiente definición:

La Prueba de Caja Negra o Prueba Funcional es una donde las condiciones de Pruebas son desarrolladas basadas en las funcionalidades del Sistema o del programa (Lewis 2005).

Las Pruebas de Caja Negra se centran en los Requisitos funcionales del Software, este tipo de Prueba le permite al equipo de Pruebas obtener conjuntos de Requisitos de entrada que ejerciten

completamente todos los Requisitos de un programa. Por eso requieren información sobre los datos de entrada y observar las salidas, pero no necesitan conocer como el programa o Sistema trabaja (Lewis 2005). Este tipo de Prueba se va a centrar en las funcionalidades del Sistema y compararla con las especificaciones. El que realiza las Pruebas de Caja Negra ve el Sistema como una Caja Negra y no se interesa por la estructura interna del Sistema.

Algunos piensan que la Prueba de Caja Negra es un alternativa a las Pruebas de Caja Blanca, sin embargo se trata de de un enfoque complementario que intenta descubrir diferentes tipos de errores que los métodos de Caja Blanca (Pressman 1998).

La principal ventaja de este tipo de Pruebas es que está dirigida a lo que el programa o Sistema se supone que haga, algo que es natural y entendido por todos (Lewis 2005) y la principal limitación, es la imposibilidad de realizarlas de forma exhaustiva, porque para eso habría que probar todas las condiciones de entrada y sus combinaciones. Además porque no existe conocimiento de la estructura interna o lógica del programa y puede que existan errores o descuidos de los programadores los cuales podrían no ser detectados con las Pruebas de Caja Negra (Lewis 2005).

1.3.1. ¿Que se persigue con las Pruebas de Caja Negra?

La Prueba de Caja Negra intenta encontrar diversos tipos de errores, que se pueden clasificar en las siguientes categorías: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento, errores de inicialización y de terminación (Pressman 1998).

A diferencia de la Prueba de Caja Blanca que se realiza previamente en el proceso de Pruebas, la de Caja Negra suele llevarse a cabo en fases posteriores de la Prueba ya que estas ignoran las estructuras de control y se centran en la información. Las Pruebas de Caja Negra se diseñan para responder las siguientes preguntas: ¿Cómo se Prueba la validez funcional? ¿Qué clases de entrada compondrán unos buenos Casos de Prueba? ¿Es el Sistema particularmente sensible a ciertos valores de entrada? ¿De qué forma están aislados los límites de una clase de datos? ¿Qué volúmenes y niveles de datos tolerará el Sistema? ¿Qué efectos sobre la operación del Sistema tendrán combinaciones específicas de datos? (Pressman 1998)

Mediante las técnicas de Prueba de Caja Negra para probar una operación se deben derivar suficientes Casos de Pruebas para verificar lo siguiente:

- Para cada valor válido usado como entrada, un valor apropiado fue retornado por la operación.
- Para cada valor inválido usado como entrada, solamente un valor apropiado o fue retornado por la operación.
- Para cada estado válido de entrada, un apropiado estado de salida ocurre.
- Para cada estado válido de entrada, apropiado estado de salida ocurre (RUP 2003).

1.3.2. ¿Cómo ejecutar las Pruebas de Caja Negra?

Como resultado de los años de experiencia acumulados en las Pruebas de Software, se han desarrollado diferentes técnicas de Pruebas, todas con sus características y objetivos. Algunas de las principales son: la partición equivalente, análisis de valores límite, Prueba de comparación y métodos basados en grafos.

La partición equivalente es un método de Prueba de Caja Negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar Casos de Prueba. Un caso de Prueba ideal descubre de forma inmediata una clase de errores (por ejemplo, proceso incorrecto de todos los datos de carácter) que, de otro modo, requerían la ejecución de muchos casos antes de detectar el error genérico (Pressman 1998).

Se ha comprobado que los errores tienden a darse más en los límites del campo de entrada que en el centro, razones por la cual se ha desarrollado el análisis de valores límites como técnica de Prueba, esta técnica lleva a definir Casos de Prueba que ejerciten los valores límites.

En lugar de seleccionar cualquier elemento de una clase de equivalencia, el Análisis de los valores límites lleva a la elección de Casos de Prueba en los extremos de la clase. En lugar de centrarse solamente en las condiciones de entrada, el análisis de valores límites obtiene Casos de Prueba también para el campo de salida (Pressman 1998).

Las directrices del análisis de valores límites son las siguientes:

1. Si una condición de entrada especifica un rango delimitado por los valores a y b , se deben diseñar Casos de Prueba para los valores a y b y para los valores justo por debajo y justo por encima de a y b , respectivamente.
2. Si una condición de entrada especifica un número de valores, se deben desarrollar Casos de Prueba que ejerciten los valores máximo y mínimo. También se deben probar los valores justo por encima y justo por debajo del máximo y del mínimo.
3. Aplicar las directrices 1 y 2 a las condiciones de salida. Por ejemplo supongamos que se requiere una tabla de temperatura/presión como salida de un programa de análisis de ingeniería. Se deben diseñar Casos de Prueba que creen un informe de salida que produzca el máximo (y el mínimo) número permitido de entradas en la tabla.
4. Si las estructuras de datos internas preestablecidas (por ejemplo, una matriz que tenga un límite definido de 100 entradas) hay que asegurarse de diseñar un caso de Prueba que ejercite la estructura de datos en sus límites (Pressman 1998).

La mayoría de los Ingenieros de Prueba han aplicado intuitivamente alguna forma de análisis de valores límites. Si se aplican las directrices expuestas anteriormente la Prueba de valores límite será más completa y aumentará la probabilidad encontrar errores.

Existen situaciones en que la fiabilidad del Software es absolutamente crítica, por ejemplo en el control de una Planta nuclear. En estos casos a menudo se utiliza Hardware y Software redundante para minimizar la posibilidad de error. En el caso del Software redundante consiste en que varios equipos desarrollan versiones independientes con las mismas especificaciones y se deben probar todas las versiones con los mismos datos de Prueba, para asegurar que todas generan salidas idénticas. Los investigadores han sugerido que para los casos críticos se deben desarrollar estas versiones aunque sólo se vaya a utilizar una de estas finalmente, estas versiones son la base de la técnica de Prueba de Caja Negra conocida como Prueba de Comparación o Prueba Mano a Mano.

A cada una de las versiones se le debe pasar como entrada varios Casos de Prueba diseñados mediante otra técnica de Caja Negra, si todas las salidas son idénticas se asume que todas las versiones, si la salida es diferente se investiga la causa de las de los defectos que proporcionan las diferencias. La comparación generalmente se puede llevar a cabo automáticamente.

Aunque parezca muy completa esta técnica no es perfecta y sus principales dificultades son: en primer lugar, si el error se encuentra en la especificación a partir de la cual se han desarrollado todas las versiones, lo más probable es que todas las versiones reflejen ese error y en segundo lugar si todas las versiones independientes producen resultados idénticos pero erróneos, la Prueba de comparación no detectará el error (Pressman 1998).

La técnica de comparación es muy costosa y no debe aplicarse para Sistemas que no sea tan crítico el tema de las Pruebas, aunque para algunos Sistemas esta técnica puede ser muy útil y dar garantía y seguridad del Software.

1.4. Estrategias de Prueba de Software.

Una Estrategia de Prueba integra las técnicas de diseños de Casos de Prueba en una serie de pasos bien Planificados que dan como resultado una correcta construcción del Software. Y lo que es más importante, una Estrategia de Prueba de Software proporciona un mapa a seguir para el responsable del desarrollo del Software, a la organización de control de la Calidad y al Cliente: un mapa que describe los pasos que hay que llevar a cabo como parte de la Prueba, cuándo se deben Planificar y realizar esos pasos, y cuánto esfuerzo, tiempo y recursos se van a requerir. Por tanto, cualquier Estrategia de Prueba debe incorporar la Planificación de la Prueba, el Diseño de Casos de Prueba, la ejecución de las Pruebas y la agrupación y evaluación de los datos resultantes (Pressman 1998).

Una Estrategia de Prueba debe ser suficientemente flexible para promover la creatividad y la adaptabilidad necesaria para adecuar la Prueba a todos los grandes Sistemas basados en Software. Al mismo tiempo, la Estrategia debe ser lo suficientemente rígida para promover un seguimiento razonable de la Planificación y la gestión a medida que progresa el proyecto.

1.5. Fases de las Pruebas.

Las Pruebas al igual que el proceso de Ingeniería de Software se pueden ver como una espiral (figura 2). La *Prueba de Unidad* comienza en el vértice de la espiral y se centra en cada unidad del Software, tal como está implementado en código fuente. La Prueba avanza, al moverse hacia fuera de la espiral, hasta llegar a la *Prueba de Integración*, donde el foco de atención es el diseño y la

construcción de la arquitectura del Software. Dando otra vuelta por la espiral hacia fuera, se encuentra la *Prueba de Validación*, donde se validan los Requisitos establecidos como parte del análisis de Requisitos del Software, comparándolos con el Sistema que ha sido construido. Finalmente se llega a la *Prueba del Sistema*, en la que se Prueba como un todo el Software y otros elementos del Sistema.

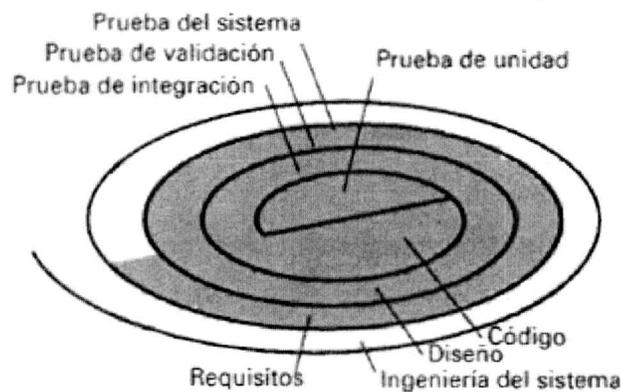


Figura 2: Fases de las Pruebas.

La *Prueba de Unidad* está enfocada a los elementos más pequeños del Software. Es aplicable a componentes representados en el Modelo de Implementación para verificar que los flujos de control y de datos están cubiertos, y que ellos funcionen como se espera. La Prueba de unidad siempre está orientada a Caja Blanca. Antes de iniciar cualquier otra Prueba es preciso probar el flujo de datos de la interfaz del componente, si los datos no entran correctamente, todas las demás Pruebas no tienen sentido. El diseño de Casos de Prueba de una unidad comienza una vez que se ha desarrollado, revisado y verificado en su sintaxis el código a nivel fuente.

La *Prueba de Integración* es ejecutada para asegurar que los componentes en el Modelo de implementación operen correctamente cuando son combinados para ejecutar un caso de uso. Se Prueba un paquete o un conjunto de paquetes del Modelo de implementación. Estas Pruebas descubren errores o elementos no completos en las especificaciones de las interfaces de los Paquetes. Esta Prueba debe ser responsabilidad de desarrolladores y de independientes, sin solaparse las Pruebas. Es el proceso de combinar y probar múltiples componentes juntos. El objetivo es tomar los componentes probados en unidad y construir una estructura de programa que esté de

acuerdo con lo que dicta el diseño. Se denomina integración incremental, cuando el programa se construye y se Prueba en pequeños segmentos en los que los errores son más fáciles de aislar y corregir, es más probable que se pueda probar completamente las interfaces y se puede aplicar un enfoque de Prueba sistemática.

Existen dos Estrategias de integración incremental: la integración descendente y la integración ascendente.

En la *Integración descendente* se integran los Módulos moviéndose hacia abajo por la jerarquía de control. Comenzando por el Módulo principal, los Módulos subordinados se van incorporando a la estructura bien, en forma primero en profundidad, que integra todos los Módulos de un camino de control principal de la estructura, o primero en anchura, que incorpora todos los Módulos directamente subordinados a cada nivel, moviéndose por la estructura de forma horizontal.

Este proceso se realiza en una serie de cinco pasos:

1. Se usa el Módulo de control principal como controlador de la Prueba, disponiendo de resguardos para todos los Módulos directamente subordinados al Módulo de Control Principal.
2. Dependiendo del enfoque de integración elegido se van sustituyendo los resguardos subordinados uno a uno por los Módulos reales.
3. Se llevan a cabo Pruebas cada vez que se integra un nuevo Módulo.
4. Tras terminar cada conjunto de Pruebas, se reemplaza otro resguardo con el Módulo real.
5. Se hace la Prueba de Regresión para asegurarse de que no se han introducido errores nuevos.

El programa continúa desde el paso 2 hasta que se haya construido la estructura del programa entero.

Con la *integración ascendente* comienza la construcción y la Prueba con los Módulos de los niveles más bajos de la estructura del programa, dado que los Módulos se integran de abajo hacia arriba, el proceso requerido de los Módulos subordinados a un nivel dado siempre está disponible y se elimina la necesidad de resguardos. Se puede implementar una Estrategia de integración ascendente mediante los siguientes pasos:

1. Se combinan los Módulos de bajo nivel en grupos que realicen una subfunción específica del Software.
2. Se escribe un controlador para coordinar la entrada y la salida de los Casos de Prueba.
3. Se Prueba el grupo.
4. Se eliminan los controladores y se combinan los grupos moviéndose hacia arriba por la estructura del programa.

A medida que la integración progresa disminuye la necesidad de controladores de Prueba diferentes. La selección de una Estrategia de Integración depende de las características del Software y de la Planificación del proyecto. Una buena alternativa es usar una mezcla de las dos Estrategias (Ascendente y Descendente) que use la descendente para los niveles superiores de la estructura, junto con la ascendente para los niveles subordinados. A medida que avanza la Prueba de integración, se deben identificar los Módulos críticos. Un Módulo crítico es aquel que presenta una o más de las siguientes características: varios Requisitos del Software, mayor nivel de control, complejo o propenso a errores, Requisitos de rendimiento muy definidos y Módulos críticos que deben ser probados lo antes posible.

Las Pruebas de Sistemas, se ejecutan cuando el Software está funcionando como un todo, es la actividad dirigida a verificar el producto final, después que todos los componentes de Software y Hardware han sido integrados. En un ciclo iterativo estas Pruebas ocurren más temprano, tan pronto como subconjuntos bien formados de comportamiento de caso de uso son implementados.

Existen diferentes tipos de Pruebas del Sistema, entre las cuales se encuentran:

- *Prueba de Recuperación:* Es una Prueba del Sistema que fuerza el fallo del Software de muchas formas y verifica que la recuperación se lleva a cabo apropiadamente.
- *Prueba de Seguridad:* Intenta verificar que los mecanismos de protección incorporados en el Sistema lo protegerán, de hecho, de acceso impropios.
- *Prueba de Resistencia:* Están diseñadas para enfrentar a los programas con situaciones anormales.
- *Prueba de Rendimiento:* Está diseñada para probar el rendimiento del Software en tiempo de ejecución dentro del contexto de un Sistema integrado.

- *Prueba de aceptación:* Prueba de aceptación del usuario es la Prueba final antes del Despliegue del Sistema. Su objetivo es verificar que el Software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el Software fue construido.

1.6. ¿Cuándo terminar las Pruebas?

El punto sobre cuándo terminar la Prueba no está totalmente definido debido a que no existen métodos cuantitativos exactos para las Pruebas, aunque los que existen, pueden dar una visión superior al que da la intuición. Una de las respuestas que se manejan cuando se pregunta en qué momento deben parar las Pruebas es: Se termina la Prueba cuando se agota el tiempo o el dinero para tal efecto. Esta respuesta aunque pareciera un poco sencilla, es real, lo que poco profesional.

Musa y Ackerman sugieren una respuesta basada en un criterio estadístico: *“No, no podemos tener la certeza absoluta de que el Software nunca fallará. Pero en base a un modelo estadístico de corte teórico y validado experimentalmente hemos realizado las Pruebas suficientes para decir, con un 95% de certeza, que la probabilidad de funcionamiento libre de fallo de 1000 horas de CPU, en un entorno definido de forma probabilística, es al menos 0.995.”* (Akerman 1989)

Existe un modelo llamado modelo logarítmico de Poisson de tiempo de ejecución. El cual adquiere la siguiente forma:

$$f(t) = (1/p) \ln(l_0 p t + 1) \quad (1)$$

Donde $f(t)$ es el número acumulado de fallos que se espera que produzcan una vez que se ha probado el Software durante una cierta cantidad de tiempo de ejecución.

l_0 es la intensidad de fallos inicial del Software (fallos por unidad de tiempo) al principio de la Prueba.

p es la reducción exponencial de la intensidad de fallo a medida que se encuentran los errores y se van haciendo las correcciones.

La intensidad de fallos instantánea, $l(t)$ se puede obtener mediante la derivada de $f(t)$:

$$l(t) = l_0 / (l_0 p t + 1) \quad (2)$$

Mediante la relación de la ecuación (2), los que realizan la Prueba pueden predecir la disminución de errores a medida que avanza la Prueba. La intensidad de error real se puede trazar junto a la *curva predecida* (Figura 3.). Si los datos reales recopilados durante la Prueba y el modelo logarítmico de Poisson de tiempo de ejecución están razonablemente cerca unos de otros, sobre un número de puntos de datos, el modelo se puede usar para predecir el tiempo de Prueba total requerido para alcanzar una intensidad de fallo aceptablemente baja.

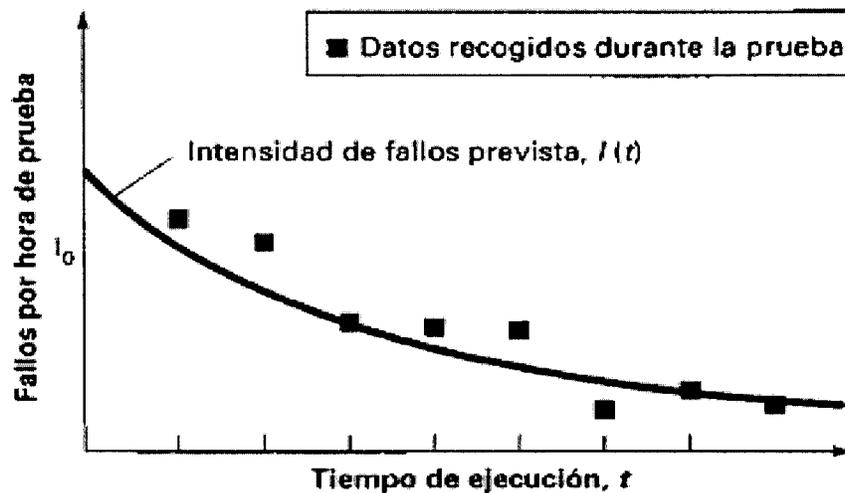


Figura 3: Curva predecida.

1.7. Plan de Pruebas

El Plan de Pruebas es uno de los elementos fundamentales a la hora de llevar a cabo el proceso de Prueba, es el que va a trazar el camino y determinará qué es lo necesario para hacerlo, además debe contemplar cual es el objetivo de la Prueba y hasta cuando se estará aplicando, la siguiente definición ilustra lo expresado anteriormente: "...el Plan de Pruebas describe la Estrategia, recursos y Planificación de la Prueba. La Estrategia de Prueba incluye la definición del tipo de Pruebas a realizar para cada iteración y sus objetivos, el nivel de cobertura de Prueba y de código necesario y el

porcentaje de Pruebas que deberán ejecutarse con un resultado específico.” (Jacobson, Booch et al. 1999)

Aunque en la definición anterior Jacobson y Booch exponen los elementos principales a tener en cuenta para la realización del Plan de Pruebas; estos no son totalmente explícitos, y se pueden complementar con Planteado por Lewis en su definición: “...*el Plan de Pruebas es como la Biblia de Software, además que define los objetivos de la Prueba, alcance, Estrategia y acercamiento, procedimientos de Prueba, ambiente de Pruebas, criterio de terminación de la Prueba, elementos a ser probados, la Prueba a ser desarrollada, el calendario de Pruebas, personal requerido, procedimientos para los reportes, asunciones, riesgos y Plan de contingencia.*” (Lewis 2005)

Un Plan de Pruebas, debe ser simple, completo, actual y accesible por los individuos apropiados para la retroalimentación y aprobación. Además, ha de cumplir con las siguientes características para ser un apropiado Plan de Prueba: fluir lógicamente y reducir al mínimo las Pruebas redundantes; demostrar cobertura funcional completa; proporcionar los procedimientos realizables para supervisar, seguir y divulgar el estado de las Pruebas; contener una definición clara de los papeles y de las responsabilidades de las partes implicadas; tener fechas de expedición y documentar claramente los resultados de la Prueba.

Para diseñar un correcto y efectivo Plan de Pruebas se deben seguir los siguientes pasos:

1. *Objetivos de las Pruebas:* Este paso asegura que todos los individuos responsables contribuyan a la definición de los criterios de la Prueba que serán utilizados.” *El elaborador de un Plan de Prueba determina qué va a ser logrado con la misma, las Pruebas específicas que se realizarán, las expectativas de la Prueba, los factores críticos de la Prueba, apremios, alcance del éxito de las Pruebas que se realizarán, los productos finales previstos de la Prueba, un reporte final del Sistema y las anotaciones y aprobación final. Los objetivos de las Pruebas deben ser revisados y aprobados” (Lewis 2005).*
2. *Determinar el alcance de la Prueba:* el que desarrolla la Estrategia de Prueba da un acercamiento a la forma en que cada Prueba será llevada a cabo. Esto incluye las técnicas de Prueba que serán usadas, procedimientos para coordinar las actividades de Pruebas con los desarrolladores, tales como reporte de defectos y seguimiento, seguimiento del progreso de

las Pruebas, estado de los reportes, recursos para la Prueba, riesgos, y una definición de las bases de la Prueba(Lewis 2005).

3. *Definir el ambiente de Pruebas:* se deben examinar las facilidades físicas para las Pruebas, se define el Hardware, Software y Red, se determina que herramientas automatizadas y herramientas de soporte son requeridas y define el soporte de ayuda requerido (Lewis 2005).
4. *Desarrollar las especificaciones de la Prueba:* el elaborador del Plan de Pruebas forma el equipo de Prueba para escribir las especificaciones de la Prueba, desarrolla el formato estándar para las especificaciones, divide el trabajo en tareas, asigna tareas a los miembros, e identifica las características a probar. El Equipo de Prueba documenta las especificaciones de Prueba para cada característica y referencia las especificaciones funcionales. También identifica las independencias y flujo de trabajo de las especificaciones y revisa las especificaciones de la Prueba(Lewis 2005)
5. *Calendario de Pruebas:* el desarrollador del Plan de Pruebas desarrolla un calendario de Pruebas basado en los recursos disponibles y horario del desarrollo, compara el calendario con los plazos, balance de recursos y demanda de carga de trabajo, define los grandes puntos de chequeo y desarrolla el Plan de contingencia(Lewis 2005)
6. *Revisión y aprobación del Plan:* el desarrollador del Plan de Pruebas Planifica una reunión de revisión con el Equipo de Prueba, revisa el Plan en detalle para asegurar que es completo y realizable y obtiene la aprobación para proceder.

1.8. Las Pruebas en la actualidad.

En la actualidad los Sistemas Informáticos han alcanzado un gran desarrollo, se aplican en la mayoría de las áreas de la vida diaria .Por este motivo se hace necesario desarrollar Sistemas con Calidad, para poder satisfacer las necesidades de los cada vez más exigentes Clientes .Las Pruebas constituyen un factor relevante para lograr la Calidad en los Sistemas y permitir que cuando estos lleguen a los Clientes finales cubran sus necesidades y contengan al menos los elementos mínimos necesarios para desempeñarse correctamente.

Prácticamente todas las instituciones y empresas de hoy día, si aún no lo están, tienen entre sus objetivos a mediano o largo plazo la automatización de sus procesos; y por consiguiente sus especialistas se preocupan por la Calidad de los productos informáticos que reciben de sus

proveedores, o crean ellos mismos; así como cada día se estudia más cómo lograr que cuando estos lleguen a las manos de los Clientes, cumplan con los Requisitos mínimos esperados.

Un ejemplo de esto son los eventos que se desarrollan en el Instituto Tecnológico de Informática de Valencia el cual ya organiza la “Cuarta Edición de las Jornadas Internacionales sobre Pruebas de Software”. El propósito de las mismas es divulgar entre las empresas del sector de las Tecnologías de la Información y las Comunicaciones (TICs) la importancia de la Pruebas de Software. Siempre se cuenta con la participación de expertos en Pruebas de Software, quienes exponen las tendencias actuales más relevantes en éste área y enseñan prácticas y conocimientos que directamente podrán aplicar en su trabajo diario.

El objetivo principal de esas jornadas es concienciar a las empresas del sector TICs sobre la importancia de las Pruebas de de Software en el mercado actual, e introducir los conceptos básicos que son útiles para cualquier compañía de Software. Consecuentemente, las jornadas pretenden explicar la importancia y los fundamentos de las Pruebas de Software y describir los procesos básicos que cada compañía moderna de Software debe implementar para garantizar cierto nivel de Calidad en sus productos de Software. Además se presentan y explican técnicas, modelos y metodologías que se pueden utilizar para llevar a cabo un proceso básico de testeo (Valencia 2007).

Además, como medidas para mejorar el alcance de las Pruebas se especifican la automatización de las Pruebas, un ejemplo es el proyecto EVOTEST que se desarrolla en ese mismo Instituto para la aplicación de algoritmos evolutivos para la generación automática de Casos de Pruebas.

La compañía GreenSQA que se dedica a ofrecer servicio de Prueba de Software realizó una encuesta donde preguntaban qué factores se considera inciden más en la Calidad de un producto de Software y el 26. 3% de las personas que habían participado seleccionaban a la Prueba como el factor de más importancia, el único factor que superaba a las Pruebas fueron los Recursos Humanos, evidenciando la importancia que se le da actualmente a las Pruebas en la garantía de la Calidad, esta encuesta está disponible en (GreenSQA 2007).

Como parte de la evolución que han tenido las Pruebas de Software se puede identificar el esfuerzo por mejorar el rendimiento de las Pruebas, para lograrlo hay empresas que se dedican a brindar los servicios de Pruebas y han desarrollado productos que automaticen las Pruebas, por

ejemplo la compañía Segue ha desarrollado un producto el cual automatiza las Pruebas y ha sido contratado por importantes compañías de desarrollo de Software como por ejemplo PalmOne, que es una destacada compañía de desarrollo de Software. Entre las características de SilkTest están los elementos del flujo de trabajo para la creación y personalización de Pruebas, acceso directo a la Base de Datos y validación, Sistema de recuperación para las Pruebas desatendidas y la capacidad de llevar a cabo Pruebas en diferentes plataformas y tecnologías con un conjunto de escritos de Pruebas.

De manera general en el mundo actualmente las Pruebas son desarrolladas por algunas compañías que se han especializado en el Control de la Calidad y las Pruebas de Software, un ejemplo es la compañía SQS S. A, la cual se especializa en servicios de Consultoría de Calidad de Software y Pruebas, además promocionan la Calidad de Software y para ello se realiza la organización de encuentros periódicos con profesionales o la celebración anual de las conferencias internacionales QA&TEST. Este evento se ha convertido en una cita ineludible al que acuden empresas de todo el mundo para exponer y conocer experiencias relacionadas con la Calidad del Software y las Pruebas.

Cuba en este momento está en pleno desarrollo de Software y como parte importante de ese desarrollo se encuentra la Universidad de las Ciencias Informáticas (UCI) donde se desarrollan Software de gran relevancia. Sin embargo según se pudo comprobar en entrevistas que se realizaron a Directivos confirman que en este momento no existe un proceso de Pruebas estándar que le permita tener control sobre el estado final del Software y que no se tiene una detección de los errores antes de llegar a los Clientes finales porque no se cuenta con la experiencia en Pruebas de otros productos. También aceptan que las Pruebas tienen una gran importancia para determinar el estado de las aplicaciones, porque casi siempre los jefes del desarrollo Plantean algunos elementos pero cuando se desarrollan las Pruebas no se cumplen.

Como parte de los esfuerzos por mejorar la Calidad de los productos desarrollados en Cuba se creó CALISOFT (Centro Nacional de Calidad de Software), el cual se encarga, por el momento, de la realización de Controles de Calidad y realización de las Pruebas. CALISOFT ha participado en las Pruebas de Aceptación a importantes productos desarrollados en la UCI, sirviendo de terceros entre

el Proveedor y el Cliente, viabilizando, organizando y desarrollando todo el proceso de dichas Pruebas, para lo cual prepara, presenta toda la documentación necesaria para ello.

Otro ejemplo de la importancia que actualmente se le da a las Pruebas, es la creación en la UCI del Laboratorio de Pruebas, para la realización de Pruebas a los proyectos, y poder liberarlos para ser entregados o presentados a sus Clientes y Usuarios Finales.

Se han creado los Grupos de Calidad en cada Proyecto, los que se encargan del Control de la Calidad en los proyectos desarrollados en cada Facultad y de la realización de las Pruebas. Para regir todo este proceso en la UCI, existe la Dirección de Calidad y Normas.

También se han desarrollado en Cuba eventos para exponer los avances que se han tenido en la Calidad y las Pruebas de Software, un ejemplo fue el taller “Perspectivas para una Industria Cubana de Software de Calidad” desarrollado durante el evento Informática 2007 donde se debatió sobre las perspectivas de la Industria del Software en Cuba y la importancia de la Calidad y las Pruebas de Software. Mediante la organización de esos eventos se puede apreciar la gran relevancia que en este momento se le da a la Calidad y las Pruebas de Software en Cuba y los esfuerzos que se realizan para mejorar los productos obtenidos por su Industria de Software.

Conociendo de la importancia que se le otorga a las Pruebas en el mundo, a continuación se realiza el análisis de los elementos relacionados con las Pruebas para lograr realizar unas Pruebas satisfactorias.

Conclusiones.

En este capítulo se han resumido los principales elementos referentes a las Pruebas de Software brindando los conceptos fundamentales a tener en cuenta para llevar a cabo un proceso de Prueba. Todos estos elementos constituirán la base de la realización del Plan de Pruebas de Aceptación para el Módulo de Administración Financiera. Es importante identificar cada uno de los tipos de Pruebas y en qué etapa de desarrollo se ejecutan, así como los principios y los objetivos generales de las Pruebas de Software. Todo lo expuesto anteriormente puede ser tomado como guía para profesionales que se inician en el mundo de la Calidad de Software.

CAPÍTULO 2: Descripción del Plan de Pruebas.

2.1. Introducción.

En el capítulo anterior se analizaron los conceptos fundamentales relacionados a las Pruebas de Software. Lo Planteado constituye una guía para el diseño del Plan de Pruebas y para el futuro proceso de Pruebas al Módulo de Administración Financiera.

En este capítulo se hace una descripción del Plan de Pruebas diseñado para realizar las Pruebas de Aceptación del Módulo de Administración Financiera. Se abordan brevemente cada uno de los puntos que se establecen en el Plan de Pruebas y se realiza un análisis de cómo se le da cumplimiento a los principios de las Pruebas, verificándose además que se cumplan las condiciones establecidas para lograr la efectividad esperada.

Por otra parte, se describen las Pruebas desarrolladas al Módulo de Administración Contable con el objetivo de validar algunos de los elementos definidos en el Plan de Pruebas diseñado.

2.2. Objetivos del Plan de Pruebas.

Para determinar los objetivos del Plan de Pruebas se procedió a la realización de consultas a las personas implicadas en el proceso de Pruebas, tomándose como referencia los objetivos Planteados por las diferentes bibliografías que se mencionaron en el capítulo anterior. Dentro de los principales objetivos que se definieron están: determinar el alcance de las Pruebas, definir las Estrategia y el flujo de trabajo a seguir, establecer los tipos de Pruebas a realizar, definir y describir los recursos humanos, de Hardware y Software que necesitan para el desarrollo de las Pruebas y la elaboración del cronograma de las Pruebas, además mencionar los criterios de aceptación y evaluación de las Pruebas realizadas. Se debe verificar el cumplimiento de los objetivos enunciados para el Plan de Pruebas, en este caso se utiliza una Lista de Comprobación¹.

2.3. Objetivos de las Pruebas.

Para determinar los objetivos de estas Pruebas se tiene presente el propósito fundamental de las Pruebas de Software enunciado en el capítulo precedente: detectar la mayor cantidad de errores en

¹ Ver Glosario de Términos

el Software en el menor tiempo y costo posible. Además se definen otros objetivos específicos que fueron determinados por las personas involucradas en la realización de las Pruebas en conjunto con el equipo de desarrollo y los directivos del proyecto. Dentro de los principales objetivos están: detectar y documentar la mayor cantidad de errores posibles en el Software, verificar el cumplimiento de la mayor cantidad posible de Requisitos Funcionales y no Funcionales ² establecidos para el Sistema; demostrar que el Sistema cumple con estos Requisitos elicitados y que los Clientes tengan la posibilidad de ver hasta qué punto el Sistema satisface sus expectativas; además que los desarrolladores observen cómo se comporta la aplicación en un entorno real.

2.4. Alcance de las Pruebas

En esta sección del Plan de Pruebas se especifican las etapas en que se dividen las Pruebas. En la primera etapa se realizarán las Pruebas de cada uno de los submódulos por separados, en la etapa siguiente se realiza la integración de cada uno de estos submódulos y en la última se propone la realización de Pruebas de Sistema, es decir se Prueba Administración Financiera como un Sistema, donde se realizan Pruebas de Seguridad, de Funcionalidad, entre otras. La división explicada le da cumplimiento a uno de los principios de las Pruebas, que Plantea que las Pruebas deben comenzar por lo pequeño y progresar hacia lo más grande, lo que permitirá que sea más fácil de identificar los errores en la primera etapa en cada uno de los submódulos y en las etapas posteriores se podrá verificar con mayor eficiencia la relación de estos submódulos e identificar los errores producidos en la relación de los submódulos. También hará más simples las Pruebas de Regresión y reducirá el número de errores en las últimas etapas donde es más difícil saber el origen de los defectos, esto ayudará a reducir el tiempo de Pruebas y como las personas que participan en el proceso de las Pruebas pertenecen al proyecto y tienen otras tareas que realizar les permitirá incorporarse más rápido a ellas. La posibilidad de dividir las Pruebas entra dentro de los elementos que facilitan las Pruebas introducidos en el capítulo 1 el cual Plantea que la capacidad de descomposición de un Sistema ayuda a facilitar las Pruebas. Controlado el ámbito de las Pruebas se puede aislar más rápidamente los problemas y llevar a cabo mejores Pruebas de Regresión.

² Ver Glosario de Términos

2.5. Estrategia de evolución del Plan de Pruebas

Aquí se establecen los mecanismos establecidos para chequear la evolución del Plan de Pruebas y las personas encargadas de validarlos. Este punto es fundamental, porque mediante dichos mecanismos se puede tener una idea de cómo se van desarrollando las Pruebas, además dará una idea de cuán bueno fue el Plan de Pruebas que se diseñó. Se incluye la descripción de los pasos a realizar para modificar el Plan de Pruebas y el cronograma en dependencia de la evolución de las Pruebas así como de los factores que puedan afectar su desarrollo. También servirá como retroalimentación para Pruebas posteriores que se deben realizar a otros Módulos, a medida que vaya evolucionando las Pruebas se pueden incorporar o quitar alguno de estos elementos con el fin de que se logre una mayor eficiencia y objetividad en las Pruebas.

2.6. Roles y Responsabilidades.

En este punto se describen los recursos humanos que intervendrán en el proceso de Pruebas, además sus roles y responsabilidades. Se tiene en cuenta uno de los elementos sobre la facilidad de las Pruebas definidos en el capítulo anterior que es la facilidad de comprensión. Cuanta más información se tenga, más inteligentes serán las Pruebas. Se debe hacer por parte del equipo de desarrollo una presentación formal donde se describa el Sistema y sus especificaciones, se seleccionará para la realización de las Pruebas por parte de los Clientes personas que tengan conocimiento de la contabilidad y que posean habilidades en el uso de Sistemas informáticos, por eso se han identificado dos especialistas en informática, y los demás funcionarios deben tener conocimiento sobre el tema de la contabilidad y las finanzas, además dentro del cronograma descrito en el Plan de Pruebas se especifica que se destinarán dos días para la familiarización y entrenamiento con el Sistema.

Los especialistas del equipo de Calisoft que intervendrán en las Pruebas también tendrán una previa preparación con el Sistema con el objetivo de que tengan conocimiento del funcionamiento del mismo y puedan ayudar a los Clientes en la ejecución de las Pruebas. Todas estas medidas permitirán que las Pruebas sean más objetivas y sea mayor la probabilidad de detectar no conformidades, además como los Clientes serán especialistas en informática o vinculados con la contabilidad y las finanzas, tendrán la posibilidad de comprobar si el Sistema cumple o no con lo que ellos esperan.

Otro factor importante en este punto es que los probadores no pertenecen al equipo de desarrollo del Sistema, esto le da cumplimiento a otro de los principios de las Pruebas que Plantea que para ser más efectivas, las Pruebas deberían ser conducidas por un equipo independiente. Existen dos momentos en los cuales se pudiera analizar este factor, el primero es durante las Pruebas de liberación en las cuales son realizadas por el personal de Calisoft y otros compañeros que aunque forman parte del proyecto tampoco pertenecen al equipo de desarrollo y el otro momento es durante las Pruebas de Aceptación con el Cliente donde además del los Clientes interviene el mismo equipo de Pruebas. Esto ha brindado buenos resultados en las Pruebas precedentes que se le han realizado a otros Módulos del proyecto, se han detectado errores que pasaron inadvertidas por los desarrolladores durante las Pruebas realizadas por ellos, porque como se explicó en el capítulo 1 es muy difícil para las personas que participan en el desarrollo de un Sistema encontrarle errores a lo que ellos mismo han realizado.

2.7. Escenario de Pruebas.

En este epígrafe se describe el escenario donde se realizarán las Pruebas, se han identificado las PC necesarias de acuerdo al alcance de las Pruebas. El escenario es controlado, con el objetivo que no se presenten factores que puedan generar algunos defectos que no tengan que ver con la aplicación, es decir problemas que pudieran aparecer por la instalación de otros programas o por algún acceso no controlado a través de la red. La configuración del escenario de Prueba se realiza a partir del Documento de Arquitectura donde se definen las características generales del Sistema.

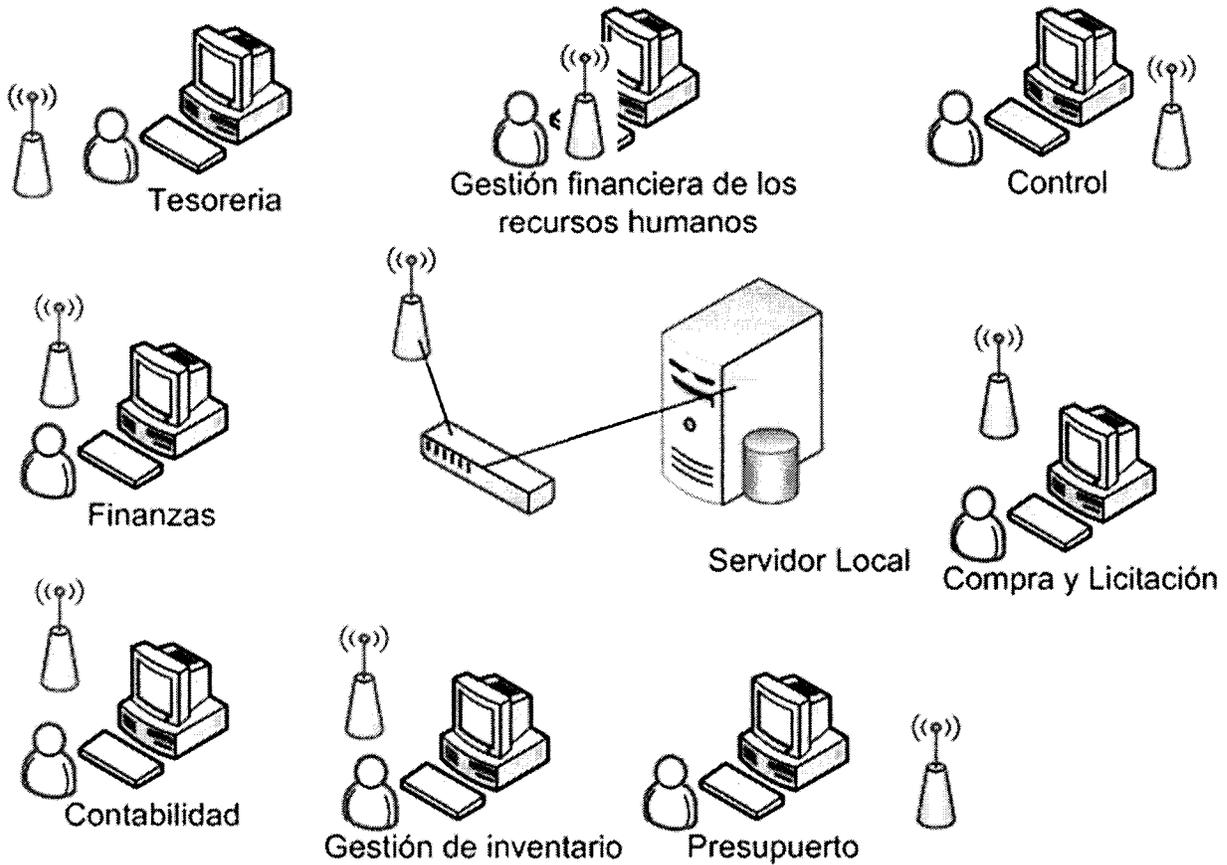


Figura 4: Escenario de Prueba

2.8. Recursos del Sistema.

Tomando como referencia las características de Hardware especificadas en el Documento de Arquitectura se describen los atributos que van a tener las computadoras, servidores e impresoras que se utilizarán para desarrollar las Pruebas, estas características coinciden con las que tendrán cuando se instale la aplicación en su entorno real, en la siguiente tabla se muestran los recursos especificados.

Recurso	Tipo
Servidor de base de datos	Pentium IV 3,00 GHz, 2 Gb RAM y 250 Gb de Disco Duro.
PC Cliente para Pruebas	Pentium IV 3,00 GHz, 512 Mb RAM y 80 Gb de Disco Duro.
Requerimientos especiales	Impresora HP LaserJet 2430 DTN
Red o subred	Red Inalámbrica

La ausencia de alguno de los recursos puede afectar el desarrollo de las Pruebas al igual que si se cambia alguna de las características especificadas, sobre todo porque entonces no coincide con las que se tendrá en el entorno real y puede que se manifieste alguna situación que pueda afectar o detener el proceso de Pruebas, que no se daría si se cumpliera con lo que se especificó anteriormente, por ejemplo si se utiliza otro tipo de servidor de base de datos con menos capacidad, las respuestas serán más lentas e incluso si se accede a la vez desde varios puntos, el servidor podría colapsar por lo que se tendría que detener las Pruebas hasta que se instale otro, esta situación aumentaría el tiempo de Prueba por lo tanto el costo, y habría que hacer modificaciones en el Plan de Pruebas y en el cronograma.

2.9. Recursos de Software.

En esta sección se especifica que los requerimientos de Software requeridos para la ejecución deben ser garantizados por el equipo de desarrollo. Lo que se ha realizado hasta ahora es la instalación en todas las máquinas de Pruebas una imagen que contiene todos los programas requeridos para la ejecución correcta de la aplicación. Al igual que se explicó anteriormente este es un punto que no se debe violar ya que de hacerse puede afectar el éxito de las Pruebas. En Pruebas que se han desarrollado a otros Módulos se han manifestado los problemas que esto puede traer, por ejemplo en una de las máquinas de Pruebas que no había sido instalada con la imagen definida, comenzaron a aparecer errores en la aplicación que tenían que ver con la ausencia de algunos programas requeridos por el Sistema y por la presencia de otros que afectaban el funcionamiento de la aplicación, dicha situación llevó a tener que instalar nuevamente la computadora, actividad que retrasó las Pruebas, aumentó el tiempo y afectó el Plan y el cronograma definido y se debe tener en cuenta que siempre que se prolongue el tiempo de Pruebas va a influir negativamente en el costo del proyecto.

2.10. Requerimientos a probar.

En este epígrafe en el Plan de Pruebas se especifica que en estos momentos no se pueden definir todos los Requisitos porque no han sido totalmente identificados, cuando sean identificados se anexarán al Plan de Pruebas. Aquí se manifiesta otro de los principios de las Pruebas descritos en el Capítulo 1 que Plantea que a todas las Pruebas se le debería poder hacer un seguimiento hasta los Requisitos del Cliente. Para la selección de los requerimientos a probar se deben revisar cada uno de los requerimientos especificados en los casos de usos, con el fin de que se compruebe de alguna manera todos los requerimientos especificados por el usuario. La Prueba de estos requerimientos demostrará cuan cerca está el Sistema de lo que se espera de él por parte de los Clientes así como le permitirá al equipo de desarrollo conocer cuáles son las no conformidades de los Clientes y resolverlas rápidamente. Más adelante en este capítulo se hace una descripción del diseño de los Casos de Pruebas para Administración Contable el cual es el único submódulo que tiene los requerimientos definidos.

2.11. Estrategia de las Pruebas de Aceptación.

En este epígrafe se describe el flujo de trabajo que será implementado durante todo el periodo de ejecución de las Pruebas, de igual forma se detalla las diferentes Pruebas que en cada una de las etapas y fases que serán realizadas.

2.12. Descripción del flujo de trabajo.

Aquí se hace una descripción del flujo de trabajo que se estará siguiendo diariamente durante las Pruebas, en la siguiente figura se describe dicho flujo de trabajo:

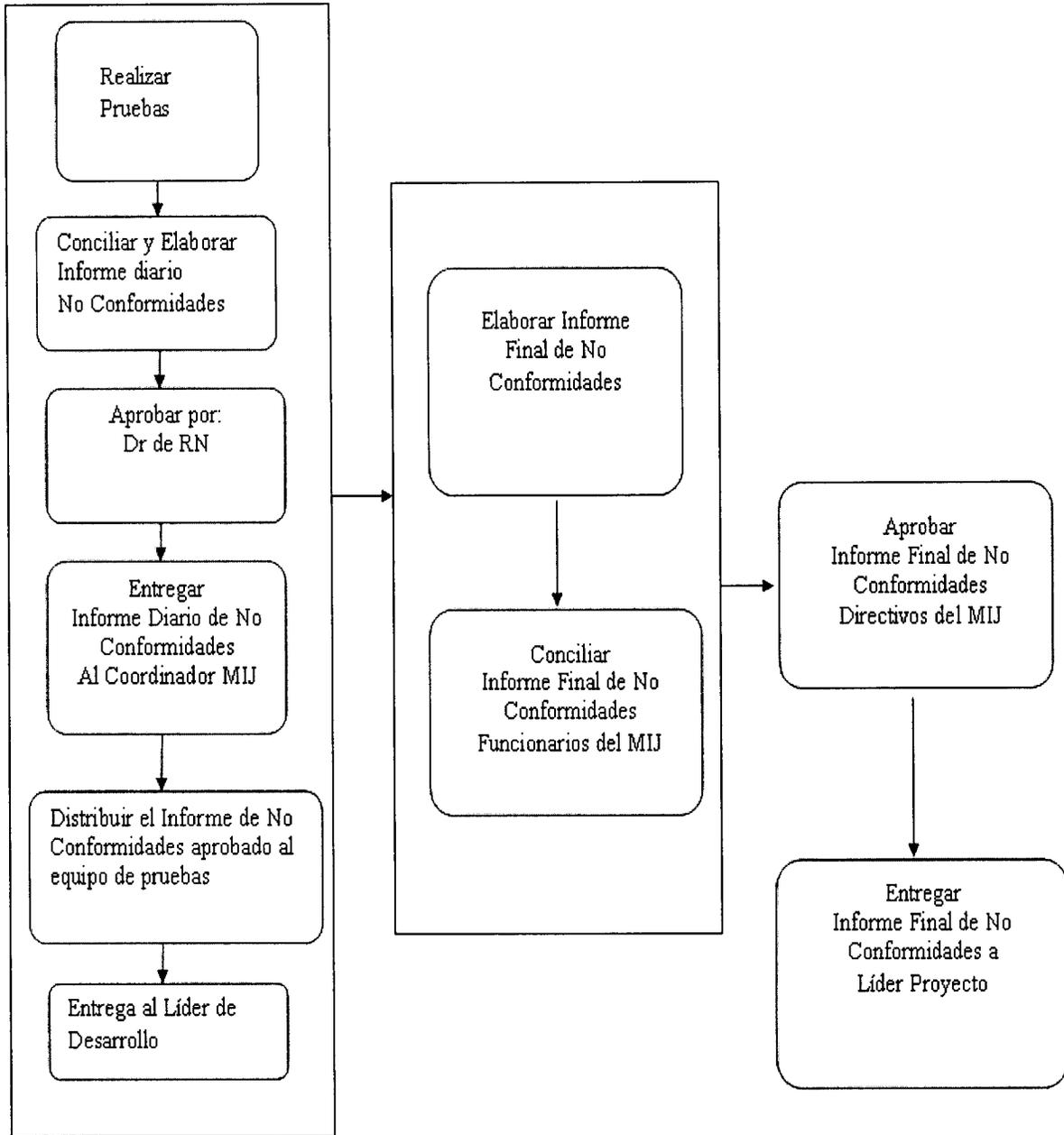


Figura 5: Flujo de Trabajo

El flujo de trabajo se inicia cuando los Funcionarios del Ministerio de Interior y Justicia (en lo adelante MIJ) comienzan a trabajar con el producto para probar sus funcionalidades, en la medida que detecten errores, molestias o incomodidades en el trabajo con el producto, estas serán anotadas de conjunto con el Especialista de CaliSoft que esté asignado, al finalizar el día se expondrán a todo

el equipo de Pruebas las no conformidades encontradas, para con ello lograr definir las de conjunto eliminando las repetidas y aclarando los términos para evitar ambigüedades y asignarle un único número que sirva como código de la misma. En todo momento el desarrollador deberá estar atento a cualquier solicitud de ayuda por parte de los probadores.

Al iniciar el día se realizará la conciliación entre los Funcionarios del MIJ que realizan las Pruebas y los Especialistas de CaliSoft para asegurar que las no conformidades encontradas el día anterior fueron descritas correctamente y a su vez asignarle el valor correspondiente a los atributos (tipo de no conformidad, estatus, categoría de falla y nivel crítico definidos en el epígrafe 4 de este documento), las mismas quedarán documentadas en el Informe Diario de No Conformidades el cual será entregado al concluir el día a Director de Registros y Notarías para que sea firmado y entregado a los coordinadores del proyecto del MIJ para su posterior conciliación.

En el caso de las No Conformidades que tengan que ver con insatisfacción de los Clientes con alguna forma del Sistema mostrar los elementos en la interfaz o mensajes (En próximos epígrafes se especificará la clasificación), se solicitará a los Clientes que ofrezcan una propuesta de cómo sería más conveniente mostrar esos elementos en la interfaz. Una vez conciliado el informe es entregado al Líder del proyecto para que este comience a ejecutar los arreglos acordados. Es de señalar que una vez aprobado el informe diario de no conformidades este no podrá ser modificado.

Al concluir el período de Prueba se realizará el Informe Final de No Conformidades, atendiendo a los Informes diarios, este será conciliado y firmado por ambas partes. Este informe será aprobado por los Directivos del MIJ de manera formal y además se especifica que para la elaboración del mismo se utilicen las Plantillas acordadas por ambas partes para lograr la estandarización de la documentación.

3. Descripción de las Estrategias y tipos de Pruebas.

Aquí se define que todas las Pruebas se realizarán de forma manual ya que como las Pruebas a desarrollar son con de Aceptación con el Cliente no es aconsejable que se realicen de forma automatizada, porque a los Clientes le interesa comprobar por ellos mismos el funcionamiento del Sistema. Además se especifica que se probarán todas las funcionalidades implementadas y se expresa que las Pruebas se harán en cada una de las etapas teniendo en cuenta las siguientes fases:

Primera Fase: organización del escenario de Pruebas y capacitación del equipo de Pruebas.

Segunda Fase: realización de Pruebas de funcionalidad.

El principal objetivo de este tipo de Prueba es medir la correspondencia entre el documento Requisitos funcionales y las funciones que realmente fueron implementadas en el Sistema. Además se realizarán Pruebas que permitan determinar si la lógica del flujo básico del Sistema es la correcta.

Se realizaran Pruebas de Regresión para chequear que los errores ya corregidos no se vuelvan a repetir y/o no se dañen funcionalidades que funcionaban correctamente.

Tercera Fase: se realizarán Pruebas de Seguridad con el objetivo de validar la protección de la aplicación sensible a entradas no deseadas.

Cuarta Fase: aplicación de la Lista de Comprobación y los principios de Calidad.

Quinta Fase: conciliación de los documentos entregables resultados de las Pruebas, Aprobación y firma de los mismos.

Pruebas a realizar.

También en el Plan de Pruebas se definen los tipos de Pruebas a Realizar, se hace una breve descripción, se Plantean los objetivos así como las técnicas y los criterios de aceptación de estas Pruebas, la selección de los tipos de Pruebas a realizar se hizo tomando como base que las Pruebas en cuestión son Pruebas de Aceptación con el Cliente y lo que se pretende es demostrar que el Sistema cumple con todos los requerimientos establecidos, tanto los funcionales como los no funcionales.

Pruebas funcionales: La Prueba de funcionalidad se enfoca en requerimientos para verificar que se corresponden directamente a casos de usos o funciones y reglas del negocio. Los objetivos de estas Pruebas son verificar la aceptación de los datos, el proceso, la recuperación y la implementación correcta de las reglas del negocio. Este tipo de Prueba se basa en técnicas de Caja Negra que consisten en verificar la aplicación y sus procesos interactuando por medio de la interfaz de usuario y analizar los resultados obtenidos. El Objetivo de la Prueba de estas Pruebas es asegurar

la funcionalidad apropiada de los Módulos, incluyendo el flujo de trabajo, entrada de datos, proceso y recuperación y se usa como técnica ejecutar cada proceso o función usando datos válidos y no válidos, para verificar lo siguiente:

- Se obtienen los resultados esperados cuando se usan datos válidos.
- Cuando se usan datos no válidos se despliegan los mensajes de error o advertencia apropiados.
- Se aplica apropiadamente cada regla del negocio.

Se define como criterio de aceptación que todas las Pruebas Planificadas sean realizadas y todos los defectos encontrados han sido debidamente identificados y documentados.

Prueba de interfaz del usuario: Esta Prueba verifica que la interfaz de la aplicación proporcione al usuario el acceso y navegación a través de las funciones apropiadas. Además asegura que los objetos presentes en la interfaz de usuario se muestren como se espera y conforme a los estándares establecidos y se verificará lo siguiente:

- Navegación a través de todas las funcionalidades, verificar que cada interfaz es amigable al usuario.
- Verificar las funciones de Ayuda Online.

Se realizarán con el objetivo de verificar que la navegación a través de los elementos que se están probando reflejen las funciones del negocio y los requerimientos solicitados, de acuerdo a los estándares establecidos, incluyendo las validaciones de los campos, métodos de acceso como los menús y las opciones de ayuda y la técnica será crear o modificar Pruebas para cada rol verificando la navegación y los estados de los objetos para cada interfaz y cada objeto contemplado dentro de la misma. Se definió como Criterio de aceptación que cada interfaz ha sido verificada exitosamente siendo consistente con una versión de referencia o estándar.

Prueba de seguridad y control de acceso :se enfoca en dos áreas de seguridad: la primera seguridad en el ámbito de aplicación, incluyendo el acceso a los datos y a las funciones de negocios asociadas a cada rol de usuario y la segunda seguridad en el ámbito de Sistema, incluyendo conexión, o acceso remoto al Sistema. La seguridad en el ámbito de aplicación asegura que, los usuarios solo accedan a los procesos de acuerdo al rol asignado. La seguridad en el ámbito de Sistema asegura que, solo los usuarios con derecho a acceder al Sistema son capaces de acceder a las aplicaciones y solo a través de los puntos de ingresos apropiados. Tiene como objetivo verificar que un usuario pueda acceder solo a las funciones o datos para los cuales su tipo de rol tiene permiso y verificar que solo los roles con acceso al Sistema y/o red y a las aplicaciones, puedan acceder a ellos. Se empleará la técnica de Identificar y hacer una Lista de cada tipo de usuario y las funciones y datos sobre las que cada tipo tiene permiso, crear Pruebas para cada tipo de usuario y verificar cada permiso creando operaciones específicas para cada tipo de usuario y modificar el tipo de usuario y volver a ejecutar las Pruebas para los mismos usuarios. En cada caso, verificar que las funciones o datos adicionales están correctamente disponibles o sean denegados. Se define como criterio de aceptación que para cada tipo de rol, las funciones y datos apropiados de la aplicación están disponibles y todas las operaciones funcionan como se espera y se ejecutan las Pruebas de funcionalidad de la aplicación.

Prueba de Fallas y Recuperación: aseguran que el Software puede recuperarse de fallas de Hardware, Software o mal funcionamiento de la red sin pérdida de datos o de integridad de los datos. La Prueba de Recuperación es un proceso en el cual el Sistema se expone a condiciones extremas, o condiciones simuladas, para causar falla, como fallas en dispositivos de Entrada/Salida. Los procedimientos de recuperación se invocan y el Sistema es monitoreado e inspeccionado para verificar que se recupera apropiadamente y que de igual forma se logra la recuperación de los datos. El objetivo es verificar que los procesos de recuperación (manual o automáticos) recuperen apropiadamente la base de datos y los diferentes Módulos del Sistema a un estado conocido y deseado. En la Prueba se incluyen los siguientes tipos de condiciones entre otros aun por definir:

- Interrupción de energía al Cliente
- Interrupción de energía al servidor
- Interrupción de comunicaciones mediante los servidores de la red
- Interrupción de comunicación o pérdida de energía de los discos del servidor o con los controladores
- Ciclos incompletos (procesos de filtro de datos interrumpidos, procesos de sincronización de datos interrumpidos)
- Elementos de datos en la base de datos inválidos o corruptos.

Como técnica se deben usar las Pruebas creadas para probar Funcionalidad y procesos de negocio para crear una serie de operaciones. Una vez logrado el punto de comienzo deseado, se deben realizar o simular las siguientes acciones, individualmente:

- Interrumpir la energía del Cliente: apagar la PC.
- Interrumpir la energía del servidor: simular o iniciar el proceso de apagado del servidor.
- Interrupción por medio de los servidores de red: simular o iniciar la pérdida de comunicación con la red (desconectar físicamente la comunicación o apagar el servidor de red o router)
- Interrumpir la comunicación o quitar la energía de los discos del servidor o sus controladores: simular o eliminar físicamente la comunicación con uno o más controladores de disco o los discos.

Una vez que se lograron o simularon estas condiciones, se deben invocar los procedimientos de recuperación. Las Pruebas de ciclos incompletos utilizan la misma técnica excepto que los procesos de bases de datos deben ser abortados a sí mismos o terminados prematuramente.

La última Prueba requiere que se logre un estado conocido de la base de datos. Se deben corromper manualmente campos de la base de datos, punteros y claves trabajando directamente sobre la base de datos (utilizando herramientas). Se deben ejecutar las Pruebas de Funcionalidad y procesos de negocio y verificar que los ciclos se completen. Se establecen como criterio de aceptación que todos los casos, los Módulos, la base de datos y el Sistema deben, en la realización de procedimientos de recuperación, volver a un estado conocido y deseable. Este estado incluye corrupción de datos limitada a los campos, punteros o claves corruptos conocidos, y reportes

indicando los procesos u operaciones que no se completaron debido a las interrupciones. Y se deben tener en cuenta que los procedimientos para desconectar cables (simulando falta de energía o pérdida de comunicación) no son deseables o factibles. Se pueden requerir métodos alternativos, como Software de diagnóstico. Se requieren los grupos de recursos de Sistemas, Bases de datos y Red. Estas Pruebas deben ejecutarse fuera del horario de trabajo normal o en una máquina aislada.

Prueba de diseño informacional: tiene como objetivo verificar que el diseño de la interfaz y de los reportes entre los diferentes submódulos que componen al Sistema mantenga las mismas pautas de diseño gráfico. Se deberá considerar el cumplimiento de aspectos tales como:

- Color
- Tipo y tamaño de la letra tanto para la entrada de datos como para la salida de los mismos.
- Similitud en el diseño gráfico y distribución de los componentes en las interface y reportes.
- Contenido de los mensajes que emite la aplicación para eventos como: errores, alertas, información, ayudas, etc.

Se visualizarán las interfaces para realizar las comparaciones, de igual forma se imprimirán los reportes para su chequeo. Se define como criterio de aceptación que se cumplen con las pautas establecidas.

Prueba de referencia cruzada se comprueba que las no conformidades detectadas en un Módulo serán provocadas en el resto para ver la reacción de este. Tiene el objetivo de verificar que los componentes genéricos implementados para su uso en todos los Módulos funcionen correctamente, para ello se deberá realizar una Lista de dichos componentes especificando la funcionalidad que implementa para poder aplicar las Pruebas. Al detectarse un error en una funcionalidad de un Módulo y que pudiera haber sido implementado en cualquiera de los restantes se tomaran las condiciones bajo las cuales ocurrió el mismo y se repetirá en el resto de los Módulos para comprobar si este es general a todo el Sistema.

Prueba exploratoria: consiste en hacer una revisión no detallada de la aplicación antes de comenzar el proceso de Pruebas establecido. Tiene como objetivo verificar que el Sistema está listo para ser probado y que no existen problemas con pasar por las distintas funcionalidades. Se chequear sin entrar al detalle las funcionalidades, para verificar que no exista ningún problema que

impida que se pruebe alguna y el criterio de aceptación es que no exista ningún problema que impida probar cada una de las funcionalidades.

3.1. Criterios de aceptación del producto y evaluación de las Pruebas.

En el Plan de Pruebas se definió el criterio de aceptación de las Pruebas, se tomó como base la experiencia en Pruebas anteriores y la opinión de personas que han estado implicados en procesos de Pruebas anteriormente, además los criterios del Cliente. Aunque no se aplicará en este proceso de Pruebas se explicará el método de Poisson para que el personal de Prueba conozca un método estadístico para determinar cuándo detener las Pruebas y en próximas Pruebas se pudiera aplicar dicho método, en dependencia de su aceptación. En la siguiente tabla se muestra los criterios a usar:

Etapas	Criterios de Aceptación
I	La cantidad de no conformidades detectadas sin solución inmediata no debe ser mayor de 5.
II	A definir.
III	A definir.

Para cada una de los submódulos definidos en la etapa I, se emitirá el documento Acta de Aceptación, el cual será debidamente firmado por las autoridades competentes del MIJ.

El paso a la etapa II de Prueba está condicionado por las combinaciones que necesariamente tienen que ser probadas en ese escenario y que están descritos en el Plan de Pruebas.

Como resultado de las Pruebas de Aceptación cada no conformidad deberá ser analizada para definir los atributos de estatus, tipo de no conformidad, categoría y nivel de la falla atendiendo a los valores definidos por el Cliente y que son descritos a continuación:

Estatus:

Valor	Descripción
0	Inconformidad No resuelta.
1	Inconformidad Resuelta
2	Inconformidad no satisfactoria.
3	Pendiente Cliente
4	Reincidencia de la falla
4	Reincidencia de la falla

También se hizo una clasificación de las fallas, hace referencia al nivel de dificultad de las mismas, estableciendo además un estimado para la solución en dependencia del nivel donde se ubique:

Categoría	Descripción	Ejemplo	Tiempo estimado para resolver la falla
1	Efectos no significativos en la funcionalidad y usabilidad del Sistema. Errores de forma (Amigabilidad del Sistema). Capa de Presentación.	Disconformidad del desarrollo/proceso formal. Disconformidad de la documentación. Disconformidad superficial del Sistema. Palabra mal escrita.	1 día.
2	La funcionalidad del Sistema no está afectada hasta tal punto que el Sistema no puede emplearse. Capa de Negocios.	Validaciones de campos, fechas, etc. Errores en los reportes.	2 días
3	El Sistema se detiene y no puede avanzar. Capa de Negocio y Capa de Datos.	Genera error al guardar un trámite.	1 semana

Nivel crítico.

Nivel de Importancia	Descripción	Justificación
3	Alta	Afecta procesos del negocio, e impide el correcto funcionamiento del proceso para la consecución del fin.
2	Media	Procesos de validación, esquemas, etc. Generación de reportes.
1	Baja	Errores ortográficos.

3.2. Evaluación de las Pruebas.

Se definen los parámetros que se tomarán en cuenta para la evaluación de las Pruebas, estos son:

1. Asistencia y puntualidad por parte del equipo de probadores (Funcionarios Adscritos del MIJ) en 80 por ciento. Para lograr esto se chequeara diariamente la asistencia mediante el registro de asistencia al laboratorio.
2. Cumplimiento del Plan de Pruebas de Aceptación y en especial del flujo de trabajo descrito.

La evaluación de las Pruebas quedará reflejada en el documento "Evaluación de las Pruebas de Aceptación" y el mismo será elaborado por CaliSoft de conjunto con los Especialistas en Informática del MIJ.

3.3. Artefactos de entrega.

En cada una de las actividades que se desarrolla se establecen los artefactos que se deben entregar, el incumplimiento en la entrega de alguno de estos artefactos afectará la Planificación de las Pruebas ya que hay algunas actividades que requieren la entrega de ciertos requerimientos y si estos no se entregan entonces la actividad se atrasará, a continuación se especifican los artefactos a entregar por cada una de las actividades.

Planificación de las Pruebas. Se describe la Estrategia de Prueba, los recursos humanos y de tiempo necesarios para acometerlos y los artefactos que se generan son : Plan de Pruebas de Aceptación y Lista de chequeo.

Ejecutar las Pruebas: Se ejecutan las Pruebas y se genera el Informe Diario de No conformidades.

Evaluar los resultados de las Pruebas: Se genera el Informe Final de No conformidades, Solicitudes de cambio y propuesta de la solución de las no conformidades y el Acta de aceptación del producto.

3.4. Cronograma

Debido a que todavía no se ha concluido de desarrollar el Módulo de Administración Financiera, no se tiene una idea exacta del tamaño del Sistema, por este motivo se consultó con las personas del equipo de desarrollo para que ellos hicieran una estimación y tomando como base lo Planteado por ellos y la opinión de personas involucradas en otros procesos de Pruebas se hizo una estimación del tiempo dedicado a cada actividad, a medida que se vaya avanzando en el desarrollo del Sistema se podrá actualizar este cronograma.

3.5. Diseño de los Casos de Prueba.

Los Casos de Prueba para Administración Contable se diseñaron partiendo de los Requisitos establecidos para el Sistema, dándole cumplimiento al principio de las Pruebas que Plantea que en las Pruebas se le debería poder hacer un seguimiento hasta los Requisitos del Cliente. En este caso sólo se realizaron los Casos de Pruebas para comprobar el submódulo Administración Contable, debido a que todavía no se han definido los requerimientos para los otros submódulos, según el resultado de los Casos de Pruebas diseñado se podrá aplicar los mismos elementos para el diseño de los Casos de Prueba para Administración Financiera. Para la selección de los Casos de Prueba se estuvieron contrastando dos de los principios de las Pruebas, el primero es el principio que Plantea que las Pruebas no pueden ser exhaustivas y el otro es el principio de Pareto que establece que revisando el 20 % del Software se pueden encontrar el 80% de los errores, la explicación que se le da es que no se puede revisar exhaustivamente todo el Software, sin embargo si se hace un estudio de los elementos que tengan más probabilidad de tener errores, además se identifican las funcionalidades que son implementadas de la misma forma en el Software se puede cumplir el principio de Pareto hasta cierto punto, por ejemplo en Administración Contable hay varios casos de usos que tienen la funcionalidad de buscar proveedor, si esta es probada una sola vez correctamente ya no se tendría que probar nuevamente. En el diseño de los Casos de Prueba se aplicaron las técnicas de partición equivalente y de valores límites en dependencia de las funcionalidades a probar, lo que ayudará también a eliminar las Pruebas redundantes, por ejemplo en un campo que permite 90 caracteres, habría que probar con tres tipos de datos: uno inferior a 90 pero cerca, uno igual a 90 y otro superior a 90. De existir algún error relacionado con este campo tiene una alta probabilidad de aparecer, si no se conociera esta técnica de valores límites lo más probable es que hubiese que

aplicar varias Pruebas para saber el origen del error. Para los Casos de Prueba se utilizó la Plantilla establecida por CALISOFT, la cual se describe a continuación:

Presentación

Esta sesión permite identificar que elementos de configuración se estarán probando y la versión del caso de Prueba, así como el nombre del proyecto al que se le está realizando las Pruebas y la versión del Módulo. La especificación de las versiones es muy importante porque permite identificar los errores por cada versión del producto y hacer un análisis de la cantidad de errores entre las versiones e ir sacando conclusiones con dicho análisis.

Descripción General

Aquí se hace una descripción de aspectos generales a tener en cuenta a la hora de realizar el diseño de las Pruebas, incidencias en el momento de su desarrollo y otros aspectos relevantes.

Pruebas a ser aplicadas:

Lista de las Pruebas diseñadas para ser aplicadas al Caso de Uso, así como las funcionalidades para las que se diseñaron las mismas.

Funcionalidades a revisar

Para cada una de las funcionalidades se especifican los siguientes elementos:

Descripción de la Funcionalidad:

Se hace una breve descripción de la funcionalidad.

Flujo Central:

Se describen los pasos que se deben realizar para llegar a la funcionalidad que se va a probar, lo que le permite al que va a ejecutar el Casos de Prueba saber que pasos debe realizar para llegar a la funcionalidad por lo que no es necesario que conozca al detalle el Sistema, además si existe algún error será más fácil para los desarrolladores saber exactamente donde está.

Condiciones de Ejecución:

Se especifican los prerrequisitos que se deben cumplir para que se pueda ejecutar el caso de uso y la funcionalidad en cuestión.

Iteraciones.

En la siguiente tabla se registran los datos con los cuales se Prueba la funcionalidad en cada una de las iteraciones, se especifica si los datos son válidos o no y la respuesta que se espera por parte del Sistema, cuando se ejecute el caso de Prueba se coloca lo que realmente hizo el Sistema y si hay alguna observación que hacer sobre la Prueba realizada.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
<Clases Válidas>	<Clases Inválidas>	<Resultado Esperado de la Prueba según lo Especificado en el Flujo del Caso de Uso>	<Resultado Real de la Prueba>	<Observaciones>

Registro de defectos y dificultades detectados

En esta tabla se colocan todas las no conformidades encontradas cuando se ejecutan las Pruebas.

Elemento	No	No conformidad	Aspecto correspondiente	Etapas de detección	Importancia	Recomendación
<Nombre del Elemento>	<1>	<Descripción de la No Conformidad>	<Descripción del Aspecto correspondiente>	<Etapas de detección del error>	<X>	<X>

Para cada no conformidad encontrada se introduce el elemento donde se detectó, se le otorga un número, y se describe brevemente, estas especificaciones permitirán que se puedan identificar todas

las no conformidades por el equipo de desarrollo y que tenga toda la información necesaria para resolverla, también le será más fácil al equipo de Prueba verificar si se resolvió o no la no conformidad. Además se especifica en que etapa se detectó la no conformidad y la importancia que se le otorga. Se puede hacer alguna recomendación o sugerencia para resolver la no conformidad.

Anexos

Aquí se colocarán fotos o cualquier elemento que ayuden a identificar o describir las no conformidades.

A continuación se muestra un ejemplo con uno de los Casos de Prueba que se diseñó:

Descripción General

Este caso de uso se refiere a otros cobros que se puedan presentar y no estén contemplados en ninguno de los casos anteriores, eso será lo que se comprobará en el presente caso de Prueba.

Pruebas a realizar

- Pruebas funcionales.
- Pruebas de interfaz de usuario.

Funcionalidades a probar

- Guardar nuevo documento.
- Anular un documento.

1. CPR 1: Guardar nuevo documento de asignación de presupuesto.

Descripción.

Flujo Central

- El actor inicia el caso de uso Otros Ingresos en el menú principal de la aplicación Finanzas, luego selecciona Banco, después Ingreso y finalmente Otros Ingresos.
- El Sistema muestra en la interfaz los documentos y sus propiedades.
- El actor crea un nuevo documento.
- El Sistema muestra en la interfaz los datos que serán editados o seleccionados:
 - Número o Nombre Cuenta Bancaria.

- Tipo de Documento.
 - Número Documento.
 - Importe.
 - Descripción del movimiento.
 - Acción a realizar.
 - Cuenta Contrapartida o Descripción Cuenta Contrapartida.
- El actor edita o selecciona los datos.
 - El actor selecciona la acción guardar.
 - El actor ordena aplicar la operación.
 - El Sistema actualiza la Lista de documentos.
 - El actor ordena aceptar la operación.
 - El Sistema acepta la operación.
 - El Sistema actualiza la base de datos.

Condiciones de Ejecución

Tiene que existir una cuenta bancaria, una cuenta contable y un tipo de documento.

Iteraciones.

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Número de Cuenta Bancaria: "0488-1111-11-11111111" Tipo de Documento: "Nota de crédito" Número de documento: "55" Importe: "6000000" Cuenta Contable de contrapartida: "11205" Acción a realizar: "Guardar"		Se guarda el nuevo documento		
	Número de Cuenta Bancaria: "0488-1111-11-11111111" Tipo de Documento: "Nota de crédito" Número de documento: "55" Importe: "6000000" Cuenta Contable de contrapartida: "11205" Acción a realizar: "Guardar"	El Sistema no permite escribir letras como importe.		

2. CPR 2: Anular un documento.**Descripción**

Se selecciona un documento y se selecciona la operación anular.

Flujo Central

- El actor inicia el caso de uso Otros Ingresos en el menú principal de la aplicación Finanzas, Banco, Ingreso, Otros Ingresos.
- El Sistema muestra en la interfaz los documentos y sus propiedades.
- El actor selecciona un documento en estado de confirmación.

- El actor selecciona la acción anular.
- El Sistema generará un comprobante contable de forma automática el cual puede ser visualizado.

Condiciones de Ejecución

Debe existir el documento a anular.

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Ordenar anular un documento.		Se anula el documento y se genera un comprobante contable.		

3.6. Elementos a tener en cuenta en las Pruebas.

En epígrafes anteriores se ha explicado como se le da cumplimiento a alguno de los principios de las Pruebas, en este epígrafe se explica cómo se asegurará la realización de unas buenas Pruebas. El equipo de desarrollo que interviene en el Módulo de Administración Financiera desconoce este tema, aunque han cumplido de manera inconsciente muchos de los lineamientos que se deben cumplir para facilitar las Pruebas, para comprobar hasta qué punto es fácil de probar el Sistema se aplicará una Lista de Comprobación donde se verifiquen los elementos que debe cumplir el Software para facilitar la demostración de sus Funcionalidades. Aquí se propondrá una Lista de elementos que se van a comprobar antes de que comiencen las Pruebas, a continuación se analiza cómo se comportan algunos de los puntos que facilitan el proceso de Pruebas.

- Operatividad. Pospuesto que los más interesados en que el Sistema funcione correctamente son los desarrolladores, pero por experiencias anteriores este tema ha afectado el desarrollo de las Pruebas, por eso en el Plan de Pruebas descrito se especifica que una de las Pruebas a realizar es la Prueba exploratoria que de una medida de la operatividad del Sistema y si está en condiciones de ser probado, esta opción puede ayudar a detectar y resolver algunos errores que podrían detener el proceso de errores, por ejemplo errores que no permitan seguir

probando la aplicación. Esta medida ayudará a reducir el tiempo dedicado a las Pruebas y que puedan ser más efectivas.

- Observabilidad. Específicamente como en Administración Financiera los datos fundamentadme son números, permite comprobar que para cada entrada exista una sola salida y que se pueda verificar cuando se produce algún valor no esperado en estas salidas, se tendrá en cuenta para el diseño de los Casos de Prueba, para que sea más simple saber cuándo hay salidas erróneas e identificar la causa.
- Controlabilidad. Será uno de los elementos a tener en cuenta para la Prueba exploratoria, además se estudia la factibilidad que tendría diseñar un componente de Pruebas que permitiera revisar la base de datos en el caso que alguna que las salidas que se ofrecen tenga problemas, también se estudia la posibilidad de utilizar algún componente que verifique si se dispone todas las facilidades de Hardware y Software requeridos por el Sistema, esto evitaría las interrupciones que en ocasiones se causan por la ausencia de un tipo de Hardware o Software específico y ayudaría a que las Pruebas consuman menos tiempo.
- Capacidad de descomposición. Se explicó cuando se analizaron las fases en las cuales se dividió las Pruebas, la medida además permitirá que sea más fácil detectar los errores en los submódulos en la primera fase y hará más simples las Pruebas de Regresión, además reducirá el número de errores en las Pruebas de Sistema donde es más difícil saber el origen de los defectos.
- Estabilidad. Este problema se ha manifestado en Pruebas anteriores a otros Módulos del Sistema de Registros y Notarías, y ha interrumpido las Pruebas porque se ha tenido que detener el proceso y en ocasiones comenzar desde el principio. Por ejemplo durante las Pruebas al Módulo de Mercantil por un cambio sencillo en la base de datos que se estaba usando, no se pudo probar en un día completo. En este caso se especificará a los desarrolladores que los cambios durante las Pruebas deben ser mínimos y cuando sea inminente realizarlos comunicárselo al jefe del equipo de Pruebas para que se pueda hacer un análisis de en cuanto afectará el proceso de Prueba y realizar los cambios pertinentes en la Planificación. De no cumplirse se va a afectar el tiempo de Pruebas por lo que aumentarán los costos del proyecto, porque las Pruebas consumen muchos recursos. Además hay que tener en cuenta que parte del equipo de Pruebas pertenece al proyecto y el tiempo que esté realizando Pruebas no puede efectuar otras actividades que también son importantes, lo

mismo sucede con los desarrolladores que mientras se están realizando las Pruebas ellos deben estar vinculados para aclarar cualquier inquietud que pudiera surgir. En el Plan de Pruebas se especifica que el proceso de Pruebas tiene varios ciclos y que en un mismo ciclo no deben haber actualizaciones del Sistema, porque cada vez que se hace una actualización hay que comenzar las Pruebas desde el principio para verificar si realmente se le dio solución a los errores reportados y detectar si en ese proceso se generaron nuevos errores.

- Facilidad de comprensión. Cuando se analizó los roles y responsabilidades se especificó qué medidas se tomaría para asegurar que se tenga toda la información necesaria y que las Pruebas sean inteligentes y objetivas.

Los elementos explicados anteriormente pueden ser negociados entre el equipo de Prueba y el equipo de desarrollo con el objetivo de que se tengan en cuenta para facilitar las Pruebas, permitirá que las Pruebas optimicen tiempo, recursos y que cumplan sus objetivos, algo que influirá de manera positiva en la Calidad del Sistema. Sin embargo la facilidad de las Pruebas debe ser usada por el equipo de Pruebas para centrarse en cumplir con el objetivo fundamental de las Pruebas que es detectar errores y demostrar la operatividad y capacidad del Sistema, y no se debe ver el tema como un elemento para facilitarle el trabajo a los probadores sino como un factor que combinado con otros que analizarán en epígrafes posteriores ayudará a realizar unas buenas Pruebas. Sólo una buena comunicación entre el equipo de desarrollo y el de Prueba permitirá que las Pruebas sean exitosas, por eso se han Planificado los mecanismos que ayudarán a la transmisión de información entre ambos equipos. También es muy importante hacerle entender al equipo de desarrollo la relevancia que adquiere desarrollar un Software que sea fácil de probar, ya que esto es lo que permite demostrar si realmente el Sistema cumple con todo lo que ellos Plantean y con lo que se exige por parte de los Clientes.

3.7. ¿Cómo se asegurará la realización de unas buenas Pruebas?

Teniendo en cuenta lo definido en el Capítulo1 existen atributos que caracterizan una buena Prueba. Para las Pruebas del Módulo de Administración Financiera hay que tener presente que tienen que ser lo más inteligentes posibles y con una alta probabilidad de encontrar errores, por eso se debe consultar a personas con conocimientos del tema de la contabilidad y las finanzas para el diseño de los Casos de Prueba, además hay que comprender que por la importancia que tiene la información

que se gestiona en este tipo de Sistema las Pruebas constituyen un elemento crítico y hay que lograr que las Pruebas sin llegar a ser redundantes abarquen gran parte del Sistema y se pueda comprobar las áreas con más probabilidad de error, y las más delicadas. Existe una Planificación de tiempo y recursos para el proceso de Pruebas en cuestión por lo que se le debe dar cumplimiento al Plan de Pruebas. Se hará un estudio de todas las funciones a probar y se determinarán los mejores Casos de Prueba y no puede haber Casos de Prueba que verifiquen lo mismo. Se debe lograr que la Prueba sin llegar a ser tan simple, en menor tiempo y recursos verifique el funcionamiento de gran parte del Sistema y que logre su objetivo fundamental que es detectar la mayor cantidad posible de errores en el Sistema. Con el fin de lograr todo lo explicado se propone una Lista de Comprobación que contenga una serie de elementos a tener en cuenta para la realización de unas buenas Pruebas, lo que ayudará a comprobar de manera sencilla hasta qué punto las Pruebas se desarrollan correctamente, además va a brindar una estadística de los errores que se van cometiendo y servirá para tenerlos en cuenta para mejorar el proceso y servirán como base para las Pruebas o los otros Módulos del Sistema, a continuación se propone una Lista de Comprobación, la cual debe irse actualizando según vaya avanzando las Pruebas.

Lista de Comprobación

- Se ha hecho un estudio y se ha identificado cómo puede fallar el Software.
- No existen redundancias en las Pruebas.
- Los Casos de Prueba son los mejores por cada caso de uso.
- Pruebas de forma separadas.
- Se Prueba cada caso de uso.
- Se documenta cada defecto detectado.
- Se verifica la solución a cada defecto reportado.
- Le llega correctamente la información al equipo de desarrollo.

Se comprobarán estos puntos y se determinarán cuales son los errores cometidos y se especificará la forma que serán resueltos, se analizarán con todo el equipo de Pruebas y se expondrán ideas de las posibles causas y como se asegurará en futuras Pruebas que no se cometan los mismos errores.

Conclusiones.

En este capítulo se hizo una descripción del Plan de Pruebas diseñado para el Módulo Administración Financiera. Este Plan propuesto sirve como guía en la realización de las Pruebas de Aceptación de Administración Financiera y puede ser un elemento importante en el éxito de las Pruebas, allí se recogen todos los elementos que se deben tener en cuenta para las Pruebas como son : los objetivos de la Prueba a desarrollar, el alcance, la Estrategia que se usará, la identificación de los escenarios de Pruebas y de los recursos necesarios, así como un cronograma donde se recoge el tiempo estimado para las actividades. Además se identificó como se garantiza el cumplimiento de los principios de las Pruebas de Software y de otros elementos que se deben tener en cuenta para la realización de las Pruebas. Una vez que el Plan de Pruebas sea aprobado se debe comprobar su cumplimiento, aunque se pueden realizar algunos cambios, estos deben ser bien justificados y aprobados por los responsables de todos los equipos implicados en las Pruebas. Esto ayudará a que las Pruebas cumplan sus objetivos y que se desarrollen en el tiempo establecido, contribuyendo de esta manera a la reducción del costo del proyecto y a la mejora de la calidad del Software. Se hizo la descripción de los Casos de Pruebas que se ejecutarán a Administración Contable con el fin de validar que el procedimiento utilizado es correcto y que se pueda aplicar en las Pruebas de Administración Financiera. Para el diseño de los Casos de Prueba se tuvo en cuenta en todo momento los principios de las Pruebas, y los elementos que caracterizan una buena Prueba y en el próximo capítulo se hará una descripción de la ejecución y de los resultados obtenidos.

CAPÍTULO 3: Aplicación del Plan de Pruebas.

Debido a que el Módulo de Administración Financiera no ha sido desarrollado aún no se puede probar el Plan diseñado totalmente y se realizó la Comprobación de la validez de algunos elementos definidos en el Plan de Pruebas mediante la ejecución de Pruebas al Módulo Administración Contable que es un Sistema muy similar al de Administración Financiera e incluso será uno de los submódulos que integrará a Administración Financiera, por lo que se podrá verificar parte de la Estrategia propuesta, el diseño de los Casos de Prueba, además estas Pruebas coinciden con parte de las Pruebas definidas para la primera etapa de las Pruebas de Administración Financiera.

Se hará una descripción de las Pruebas realizadas a Administración Contable, se pondrá como ejemplo uno de los Casos de Pruebas ejecutados y se hará un análisis de los resultados de las Pruebas aplicadas, además de una valoración del Sistema probado y de las Pruebas realizadas. Se analizará el Plan de Pruebas diseñado mediante una Lista de Comprobación.

3.1. Recursos utilizados y escenario de Pruebas.

Las Pruebas la realizó un probador durante 5 días, para ello se utilizó la PC de Prueba, la impresora y el servidor, el Hardware utilizado es el mismo que se utilizará para las Pruebas de Aceptación a Administración Financiera y el escenario también , la única diferencia es la cantidad de computadoras, a continuación se muestran sus características:

Recurso	Tipo
Servidor	Pentium IV 3,00 GHz, 2 Gb RAM y 250 Gb de Disco Duro.
PC Cliente para Pruebas	Pentium IV 3,00 GHz, 512 Mb RAM y 80 Gb de Disco Duro.
Requerimientos especiales	Impresora HP LaserJet 2430 DTN
Red o subred	Red Inalámbrica

El escenario que se utilizó se muestra en la siguiente gráfica:

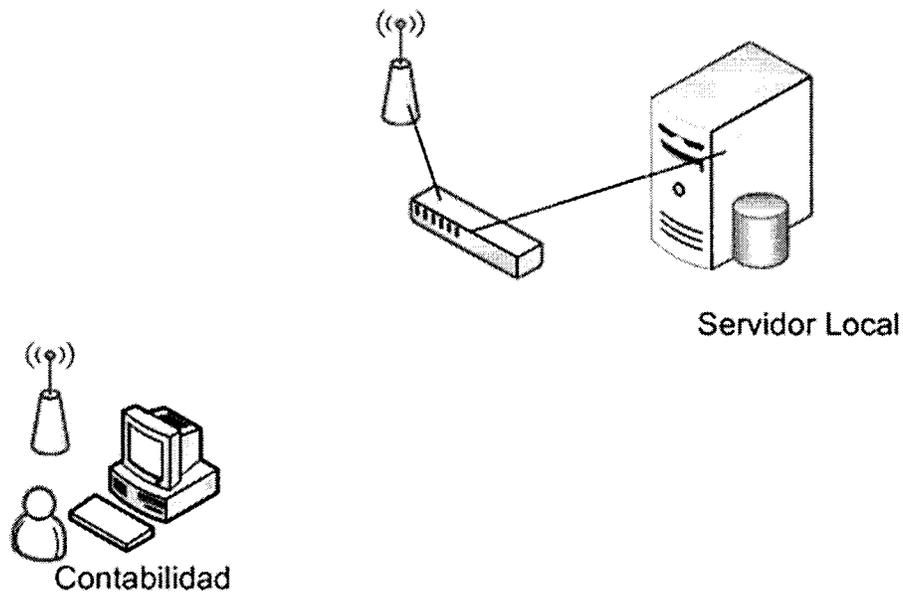


Figura 6: Escenario de Pruebas

3.2. Ejecución de las Pruebas.

Se diseñaron 15 Casos de Prueba y todos fueron ejecutados. Como parte de la Estrategia del Plan de Pruebas para Administración Financiera en estas Pruebas se realizó en primer lugar una Prueba exploratoria con el objetivo de verificar si el Sistema está en condiciones de ser probado, luego se llevaron a cabo Pruebas funcionales, de interfaz de usuario y de referencia cruzada, a continuación se muestra un ejemplo de un caso de Pruebas de los que se ejecutó, en el Anexo 2 se muestran, algunos de los Casos de Prueba ejecutados:

Descripción General

Este caso de uso consiste en Listar los comprobantes contables de cada día.

Pruebas a realizar:

- Pruebas Funcionales.
- Pruebas de interfaz de usuario.

Funcionalidades a probar:

- Visualizar reporte Diario General.
- Imprimir reporte.
- Exportar reporte.

CPR 1: Visualizar Reporte Diario General.**Descripción**

El Sistema muestra una tabla al usuario con el reporte diario general cuando se presiona el botón visualizar.

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación y el contador selecciona en el menú de la izquierda la opción: "Contabilidad", dentro de ella: "Reportes Contables", luego "Reportes Internos" y por último la opción: "Reporte Diario General".
- El Sistema muestra la interfaz: "Reporte Diario General".
- El usuario o contador selecciona el intervalo de fecha Desde-Hasta a buscar, y presiona el botón "Visualizar".
- El Sistema visualiza el reporte diario.

Condiciones de Ejecución

Debe existir el reporte diario general.

Iteraciones.

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Desde: "01/04/2007" Hasta: "20/04/2007"		El Sistema visualiza el reporte.	El Sistema visualiza el reporte	
	Desde: "01/05/2007" Hasta: "01/04/2007"	El Sistema no permite que la fecha de inicio sea mayor que la de fin.	El Sistema visualiza el reporte	
Desde: "01/09/2007" Hasta: "04/09/2007"		El Sistema debe informar que no se han encontrado resultados.	El Sistema debe informar que no se han encontrado resultados.	

CPR 2: Imprimir el Reporte.**Descripción**

Se comprueba la posibilidad de que se pueda imprimir el o los reportes seleccionados.

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación y el contador selecciona en el menú de la izquierda la opción: "Contabilidad", dentro de ella: "Reportes Contables", luego "Reportes Internos" y por último la opción: "Reporte Diario General".
- El Sistema muestra la interfaz: "Reporte Diario General".
- El usuario o contador selecciona el intervalo de fecha Desde Hasta a buscar y presiona el botón "Visualizar".
- El Sistema visualiza el reporte diario.
- El usuario presiona el botón "Imprimir".
- El Sistema presenta la interfaz: "Reporte".

- El usuario presiona el botón correspondiente a imprimir reporte.

Condiciones de Ejecución

Debe existir el reporte diario general.

Iteraciones.

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Presiona el botón correspondiente a imprimir		El Sistema imprime el reporte buscado.		

CPR 3: Exportar el Reporte.

Descripción

Se comprueba que se pueda exportar el reporte diario general a uno de los formatos anteriormente especificados.

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación y el contador selecciona en el menú de la izquierda la opción: "Contabilidad", dentro de ella: "Reportes Contables", luego "Reportes Internos" y por último la opción: "Reporte Diario General".
- El Sistema muestra la interfaz: "Reporte Diario General".
- El usuario o contador selecciona el intervalo de fecha Desde Hasta a buscar y presiona el botón "Visualizar".
- El Sistema visualiza el reporte diario.
- El usuario presiona el botón "Imprimir".
- El Sistema presenta la interfaz: "Reporte".
- El usuario presiona el botón correspondiente a "Exportar Reporte" y selecciona el formato en que lo desea exportar.

Condiciones de Ejecución

Deben existir reportes correspondientes a ese tipo de ingreso.

Iteraciones.

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Presiona el botón correspondiente a exportar.		El Sistema exporta el reporte buscado.	El Sistema exporta el reporte.	

3.3. Resultado de las Pruebas.

La siguiente tabla muestra el resumen de los errores detectados durante las Pruebas realizadas a Administración Contable, donde: CP i es el i -ésimo caso de Prueba y CPR i se refiere al i -ésimo requisito de Prueba.

Elemento	Nº	Descripción de la No conformidad	Aspecto correspondiente	Etapa de la detección	Importancia	Recomendaciones
Sistema	1	El Sistema permite que se entre fecha superior a la de sincronización	CPR1 del CP2	Ejecución de la Pruebas	Media	Revisar en todos los casos donde hay que introducir la fecha de sincronización.
Sistema	2	Cuando se ordena imprimir aparece un error	CPR2 del CP 12	Ejecución de las Pruebas	media	Revisar todas las funciones de impresión
Sistema	3	Cuando se da buscar y no se encuentran resultados no se informa	CPR3 del CP12	Ejecución de las Pruebas	baja	En todos los reportes, cuando no se encuentren resultados, se debe informar
Sistema	4	Cuando se ordena aceptar la nueva cuenta aparece un error	CPR1 del CP 13	Ejecución de las Pruebas	Alta	Revisar los roles y permisos.
Sistema	5	Cuando se crea un nuevo documento aparece un mensaje de error y no se puede guardar	CPR1 del CP 14	Ejecución de las Pruebas	alta	
Sistema	6	Se muestra un error y no permite guardar el documento	CPR2 del CP15	Ejecución de las Pruebas	Alta	
Documentación	7	En la descripción del caso de uso se especifica que activo es uno de los campos de los proveedores y en el Sistema no es así.	CPR1 del CP17	Ejecución de las Pruebas	baja	Revisar toda la documentación
Sistema	8	Ocurre el error cuando se manda a guardar el documento	CPR1 del CP 19	Ejecución de la Pruebas	Alta	
Sistema	9	Cuando se intenta aceptar la operación de confirmar el documento	CPR2 del CP 19	Ejecución de la Pruebas	Alta	
Sistema	100	Aparece un error cuando se manda a aplicar la operación de guardar	CPR1 del CP20	Ejecución de las Pruebas	Alta	

No conformidades

En la tabla se especifican cada uno de los errores detectados durante las Pruebas desarrolladas, la siguiente muestra el resumen de la cantidad y el nivel de importancia de los errores.

Nivel de Importancia	Defectos	Justificación
Alto	6	Afecta procesos del negocio, e impide el correcto funcionamiento del proceso para la consecución del fin.
Medio	2	Procesos de validación, esquemas, etc. Generación de reportes.
Bajo	2	Errores ortográficos ,etc.
Total	12	

Tabla 4: Resumen de Defectos

3.4. Evaluación de Administración Contable

Durante las Pruebas se detectaron 10 no conformidades, de las cuales, 4 son de importancia alta, eso significa que el Sistema todavía no está en condiciones de ser liberado. Sin embargo se pudo comprobar que el Sistema cumple con la mayoría de los requerimientos probados y permitió que se probara el comportamiento de una buena parte de las funcionalidades. En el próximo ciclo de Pruebas se verificará si se le dio solución a las no conformidades detectadas, además si no se han introducidos nuevos errores en el proceso de corrección. Como análisis general se puede decir que se debe dar importancia a los mensajes que se muestran cuando no se encuentran resultados en las búsquedas, o que no se puede realizar la búsqueda, además que se verifiquen los permisos para cada tipo de usuario. Las no conformidades y sugerencias se le hicieron llegar al equipo de desarrollo documentadas para que les facilite la corrección de los errores.

3.5. Resumen de las Pruebas realizadas

En la realización de las Pruebas se logró el objetivo fundamental de las Pruebas de Software, que es detectar errores y demostrar que el Sistema cumple las funcionalidades especificadas. En este caso específico donde las Pruebas realizadas se hicieron con el objetivo de comprobar algunos elementos definidos en el Plan de Pruebas se verificó que los Casos de Prueba permitieron que aparecieran defectos que tiene el Sistema, los cuales fueron documentados y entregados al equipo de desarrollo. Se demostró que el Sistema cumple con gran parte de los requerimientos especificados. Durante este proceso se pudo comprobar la efectividad de los Casos de Pruebas diseñados, aunque se deben mejorar en algunos aspectos, como son la previa identificación de de las funcionalidades con más probabilidad de que aparezcan errores, para lograrlo debe consultarse a los integrantes del equipo de desarrollo y de personas que tengan conocimiento de la contabilidad y las finanzas. También se comprobó parte de la Estrategia de Pruebas donde cada tipo de Prueba ejecutado cumplió su objetivo y se realizó parte de la primera etapa definida como parte del alcance de las Pruebas a Administración Financiera donde se realizaría Pruebas a cada uno de los submódulos por separado demostrándose que la primera etapa si se desarrolla correctamente tendrá buenos resultados. Se recomienda para la realización de las Pruebas de Administración Financiera aplicar las mismas técnicas para el diseño de los Casos de Pruebas y que se siga la misma Estrategia de Prueba, tomando como base la efectividad demostrada, además se debe cumplir todo lo especificado en el Plan de Pruebas en referencia a los recursos necesarios para la ejecución de las Pruebas y al escenario descrito. Se propone que se utilice una Lista de Comprobación cuando se estén ejecutando las Pruebas para ir chequeando cada punto, algo que permitirá que se compruebe la evolución de las Pruebas, además de ir llevando una estadística que sirva para la Planificación y la toma de decisiones en el futuro, esta Lista de Comprobación debe ir actualizándose y agregándole nuevos puntos o quitando alguno de los que tiene.

3.6. Evaluación del Plan de Pruebas.

Para determinar si el Plan de Pruebas de Pruebas diseñado está correcto se utilizará una Lista de Comprobación propuesta por (RUP 2003) que permita chequear todos los puntos que debe cumplir un Plan de Pruebas, a continuación se presenta la Lista de Comprobación:

Lista de Comprobación:

- El Plan claramente identifica el alcance de las Pruebas.
- El Plan de Pruebas claramente identifica los elementos usados para generar el contenido de Pruebas.
- Cada requerimiento del proyecto (como los establecidos en los casos de usos o las especificaciones complementarias) tiene asociado al menos un requerimiento de Prueba o una justificación de porque no constituye un requerimiento para las Pruebas.
- Todos los requerimientos para las Pruebas han sido identificados y especificado el tipo de Pruebas a ser implementado y ejecutado.
- Una Estrategia clara y concisa es documentada para cada tipo de Prueba a ser implementada y ejecutada. Para cada Estrategia de Pruebas, la siguiente información tiene que haber sido establecida
 - Nombre de la Prueba y cuál es su objetivo.
 - Una descripción de cómo la Prueba debe ser implementada y ejecutada.
 - Una descripción de las métricas y criterio a ser usado para la evaluación la Calidad de las metas. de la Prueba y completamiento de las Pruebas.
- Todos los recursos necesarios para implementar y ejecutar las Pruebas han sido identificados incluyendo, Hardware, Software y personal.
- Contiene un cronograma y los hitos para las actividades más importantes.
- Especifica los elementos creados por cada actividad de las Pruebas. (RUP 2003)

En el Plan de Pruebas diseñado se especifica el alcance de las Pruebas, se mencionan los elementos que se usó para generar el contenido de las Pruebas, se especifica de los requerimientos que se probarán y el tipo de Prueba a realizar. Para cada tipo de Prueba se especificó el nombre, objetivo y criterio de aceptación de la Prueba. Se identificaron los recursos necesarios para desarrollar las Pruebas, incluyendo recursos de Hardware, Software y los recursos humanos, de los cuales se describen además los roles y responsabilidades que desempeñarán durante las Pruebas. Se realizó un cronograma donde se especifica un tiempo para las actividades más importantes y se establecen los hitos. Se especifica las entradas y las salidas que tendrá cada actividad durante las Pruebas. Por lo tanto se puede Plantear que el Plan de Pruebas propuesto cumple con los principales

elementos que se especifican en la Lista de Comprobación analizada demostrando que está acorde con lo establecido.

Conclusiones

En este capítulo se Realizó una descripción de las Pruebas que se realizaron a Administración Contable, se explicó que se realizaron Pruebas a Administración Contable porque aún no se han definido los requerimientos para el Módulo de Administración Financiera , y se toma esta alternativa para validar algunos elementos del Plan de Pruebas propuesto. Se hizo un análisis de las Pruebas realizadas y se presentaron los resultados de dicha Prueba y se determinó que la Prueba se había desarrollado correctamente y que el mismo procedimiento de diseño de Casos de Pruebas y el procedimiento seguido así como la Estrategia de Prueba debía aplicarse cuando se vayan a realizar las Pruebas al Módulo de Administración Financiera, aunque mejorando algunos elementos señalados. Se hizo una evaluación de Administración Contable y se llegó a la conclusión que se debía someter a otro ciclo de Pruebas para verificar si se habían resuelto los errores detectados en las Pruebas realizadas y además verificar que no se introduzcan nuevos errores.

De manera general todos los elementos contenidos en el Plan de Pruebas que se aplicaron durante estas Pruebas dieron buenos resultados demostrando su factibilidad si se usa en las Pruebas a Administración Financiera.

CONCLUSIONES.

El Plan de Pruebas diseñado servirá de guía para la realización de las Pruebas de Aceptación del Módulo de Administración Financiera, debido a que en él se recogen todo los elementos que se deben tener en cuenta para la realización de las Pruebas.

- El Plan de Pruebas fue validado según los parámetros establecidos, verificándose que abarcaron todos los aspectos necesarios, y que por tanto los métodos utilizados para su confección fueron los adecuados.
- La ejecución de las Pruebas a Administración Contable permitieron validar la factibilidad de algunos elementos:
 - El diseño de los Casos de Pruebas fue efectivo, mediante su ejecución se lograron realizar Pruebas satisfactorias a las funcionalidades previstas, detectándose un importante número de errores en el Sistema probado.
 - El Plan de Pruebas diseñado, y aplicado al Módulo de Administración Contable (ajustándolo parcialmente a sus particularidades) fue validado en los aspectos de: tipos de Pruebas, técnica para el diseño de los Casos de Prueba y procedimiento de Prueba, corroborándose que de ser aplicado correctamente, permitirá guiar adecuadamente el proceso de Pruebas de manera que se puedan cumplir los objetivos de las Pruebas.
- Por lo tanto, se puede concluir que dada la efectividad demostrada en la aplicación parcial del Plan de Pruebas propuesto para el Módulo de Administración Financiera al Módulo de Administración Contable (por su gran similitud con el anterior), es factible aplicar este Diseño al Módulo de Administración Financiera, siempre que se cumpla con las especificaciones indicadas, y existe una gran probabilidad de que de utilizarse correctamente estos elementos contribuyan a un desarrollo satisfactorio de las Pruebas de Aceptación.

RECOMENDACIONES.

A partir de la investigación realizada y de los elementos tratados en este trabajo se recomienda:

- Aprobar, aplicar y evaluar el Plan de Pruebas de Aceptación propuesto y retroalimentarlo constantemente.
- Involucrar especialistas en el área de contabilidad y finanzas en la realización de los diseños de los Casos de Pruebas.
- Evaluar las Pruebas verificando el cumplimiento de los objetivos definidos.
- Investigar sobre las técnicas de automatización de Pruebas y aplicarlas, en caso de ser posible, al Módulo de Administración Financiera.
- Continuar profundizando en la Investigación sobre las normas y Requisitos establecidos para los Sistemas de este tipo en Venezuela, Cuba y otras comunidades de desarrollo de Software líderes en el área.

GLOSARIO.

Calidad:	Calidad de Software: Satisfacción de las necesidades de los usuarios.
Caso de Prueba:	Especificación de un caso para probar el Sistema, incluyendo que probar, con que entradas y resultados y bajo qué condiciones.
Cliente:	Persona, organización o grupo de personas que encarga la construcción de un Sistema, ya sea empezarlo desde cero, o mediante el refinamiento de versiones sucesivas.
Defecto:	Cualquier requerimiento, elemento de diseño o de implementación que si no es cambiado, causará un diseño, implementación, Prueba, uso, o mantenimiento inapropiado del producto.
Desarrollador:	Persona que trabaja directamente en el desarrollo de un proyecto de Software, dígase: analista, programador, diseñador, etc.
Falla:	Error en el funcionamiento que emite el Sistema.
Grupo de desarrollo:	Dos personas o más que trabajan para lograr una meta, objetivo o misión común, donde cada individuo tiene asignado un rol específico y donde el completamiento de la misión depende de los miembros del equipo.
Interfaz:	Una colección de operaciones que se usan para especificar el servicio de una clase o de un componente. Un juego nombrado de operaciones que caracterizan la conducta de un elemento. La Interfaz hombre-máquina es un canal comunicativo entre el usuario y el ordenador.

Lista de Comprobación: Una serie de pasos ó procedimiento que se deben seguir de forma precisa, pueden ser útil para comprobar un producto o proceso.

Proceso: Secuencia de actividades invocadas para producir un producto de Software.

Prueba: Ejecución de un programa para verificar si cumple con lo establecido.

Requisito funcional: Requisito que especifica una acción que debe ser capaz de realizar el Sistema, sin considerar restricciones físicas: requisito que especifica comportamiento de entrada y salida de un Sistema.

Requisito No funcional: Requisito que especifica propiedades del Sistema, restricciones del entorno o de implementación, rendimiento, dependencias de la plataforma, mantenibilidad, extensibilidad o fiabilidad. Requisito que especifica restricciones físicas sobre un requisito funcional.

Usuario: Persona que utiliza normalmente el Software.

BIBLIOGRAFÍA.

- Akerman, J. D. M. y. A. F(1989). "Quantifying Software Validation :When to stop Testing?."
- Boris, B(1995)."Black box testing".John Wiley and Sons, New York, 1995.
- Gilb, T(1995). What we fail to do in our current testing culture.
- Glass,R(1991)."Building Quality Software". Englewood Cliffs, NJ:Prentice Hall.
- GreenSQA (2007). Retrieved 20/04, 2007, from www.GreenSQA.com/index.php
- Myers,G(1979),"The art of Software testing", John Wiley and Sons, N.Y., 1979.
- Hutcheson, M. L(2003). "Software Testing Fundamentals: Methods and Metrics."
- IEEE (1991). "Metrics."
- Jacobson, I., G. Booch, et al(1999). "El Proceso Unificado del Desarrollo del Software ".
- Lewis, W. E(2005). Software Testing and Continuous Quality Improvement, AUERBACH.
- Napal,I., I. Reyes(2001)."Las Pruebas de Software, su aplicación al Config. Case ".
- Perry,W(1991)."Quality Assurance for Information System :Methods,Tools, and Technique".
Wellesley, MA:QED.
- Pressman, R. S(1998). "Ingeniería del Software.Un enfoque práctico".
- RUP (2003). "Ayuda del Rational Unified Process."
- SQS,(2007). Retrieved 20/04, 2007, from www.SQS.es/index.php.
- Thomas, Royer(1992)."Software Testing Management: Life on Critical Path". Englewood Cliffs.
- Shulmeyer,G(1990)."Zero Defect Software". New York:McGraw-Hill.
- Tian, J(2005). Software Quality Engineering, John Wiley & Sons.
- Valencia, I. T. d (2007). Retrieved 03/03, 2007, from www.iti.upv.es.

ANEXOS

Anexo 1: Plan de Pruebas de Aceptación del Módulo Administración Financiera.

Elemento de Configuración: Aplicación y Documentación

Versión 1. 0. 0. 0

Control de versiones:

Fecha	Versión	Descripción	Autor
20/04/07	1. 00	Pre Plan de Pruebas de Aceptación para conciliar con los involucrados.	Nemury Silega

Introducción

Este documento se confecciona con el objetivo de definir el Plan de las Pruebas de Aceptación para el Módulo de Administración Financiera del Sistema de Registros y Notarías. Aquí se establecerán los elementos y factores a tener en cuenta para la realización de las Pruebas. Este Plan será revisado y aprobado por los directivos del proyecto así como los Clientes. En la Planificación de las Pruebas se ha tenido en cuenta como se le dio cumplimiento a los principios de las Pruebas, también se chequean el cumplimiento de los elemento que definen unas buenas Pruebas.

1. 1. Objetivos del Plan de Pruebas

El objetivo del Plan de Pruebas de Aceptación es:

- Identificar los recursos necesarios.
- Definir el alcance de las Pruebas.
- Especificar las funcionalidades aprobar.
- Identificar los elementos de Pruebas y sus prioridades.
- Describir y recomendar las Estrategias de las Pruebas a ser empleadas.
- Definir el cronograma de las Pruebas.
- Establecer los criterios de aceptación de las Pruebas y las condiciones de parada.

1. 2. Objetivos de las Pruebas

- Detectar y documentar la mayor cantidad posible de errores en el Sistema.
- Verificar la concordancia de las funcionalidades del Sistema con los Requisitos establecidos por los Clientes.
- Verificar el cumplimiento de los Requisitos establecidos para los Sistemas contables y financieros que están soportados sobre las tecnologías de la informática.
- Verificar el cumplimiento de los Requisitos no funcionales.
- Analizar las insatisfacciones que tengan los Clientes con el Sistema.

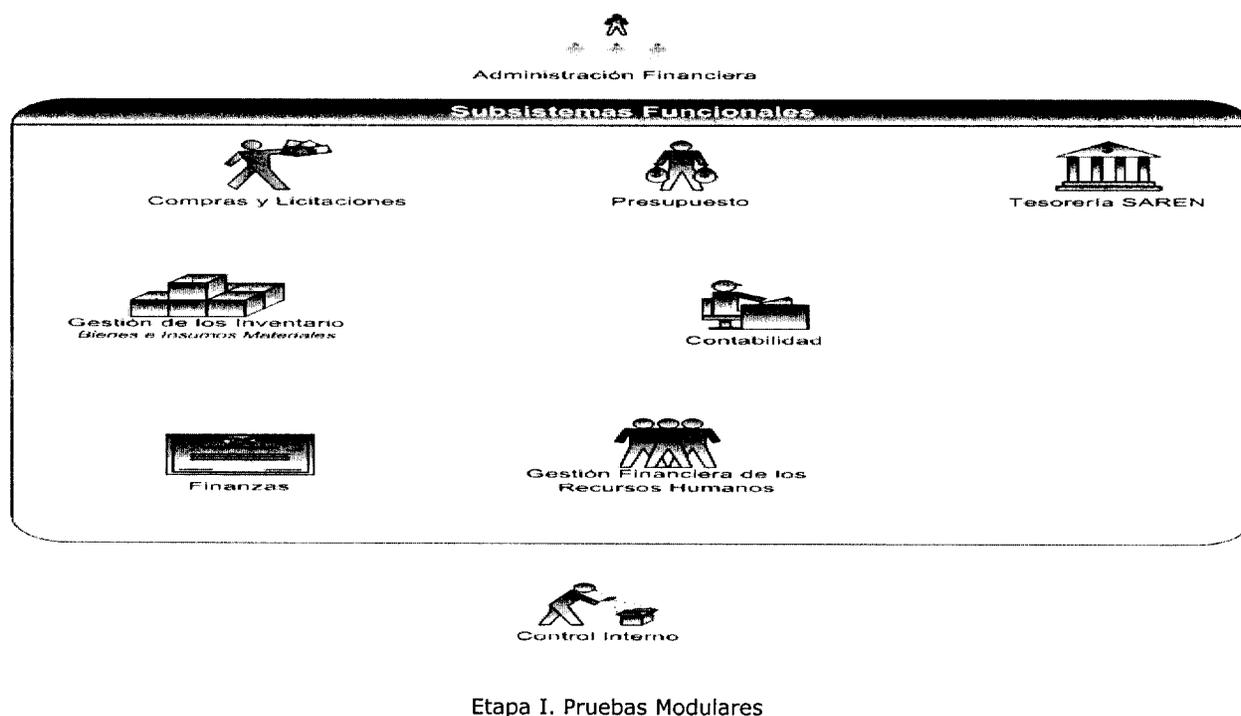
- Brindarle al equipo de desarrollo un Listado con las no conformidades detectadas y una clasificación de ellas, además de proponer una posible forma de solución tomando como base las opiniones de los usuarios.

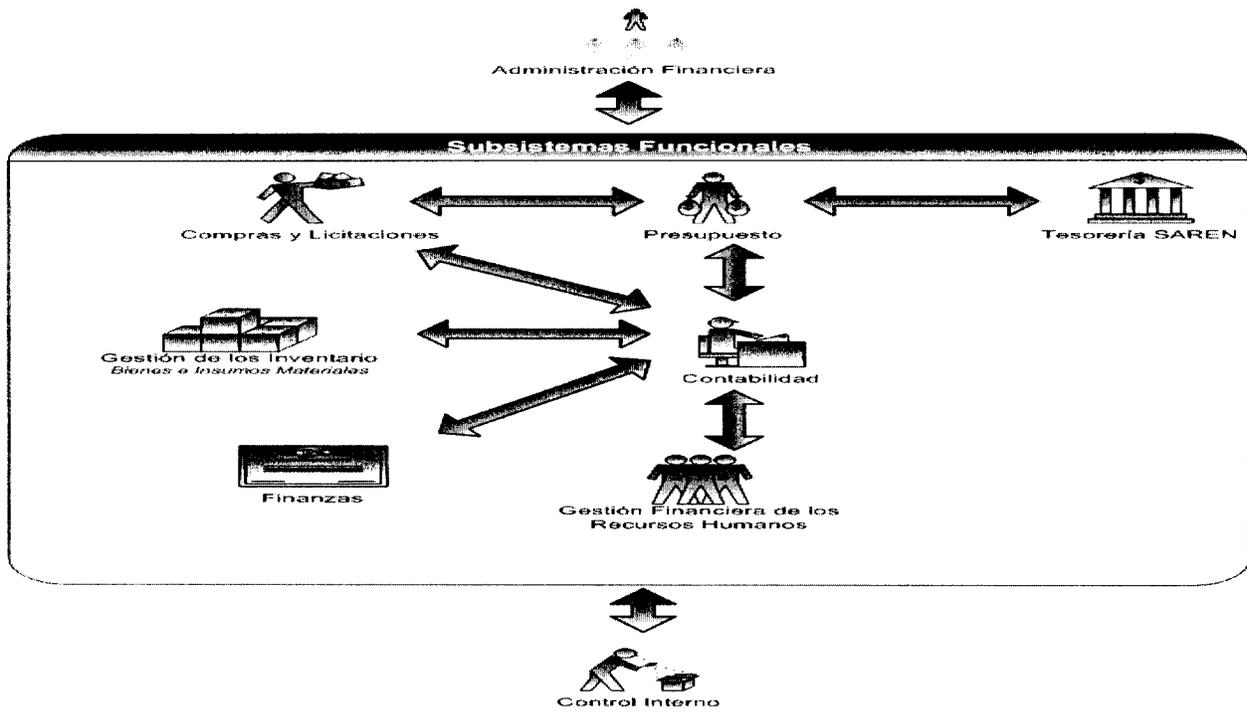
1. 3. Alcance:

Se proponen las siguientes etapas para la realización de las Pruebas:

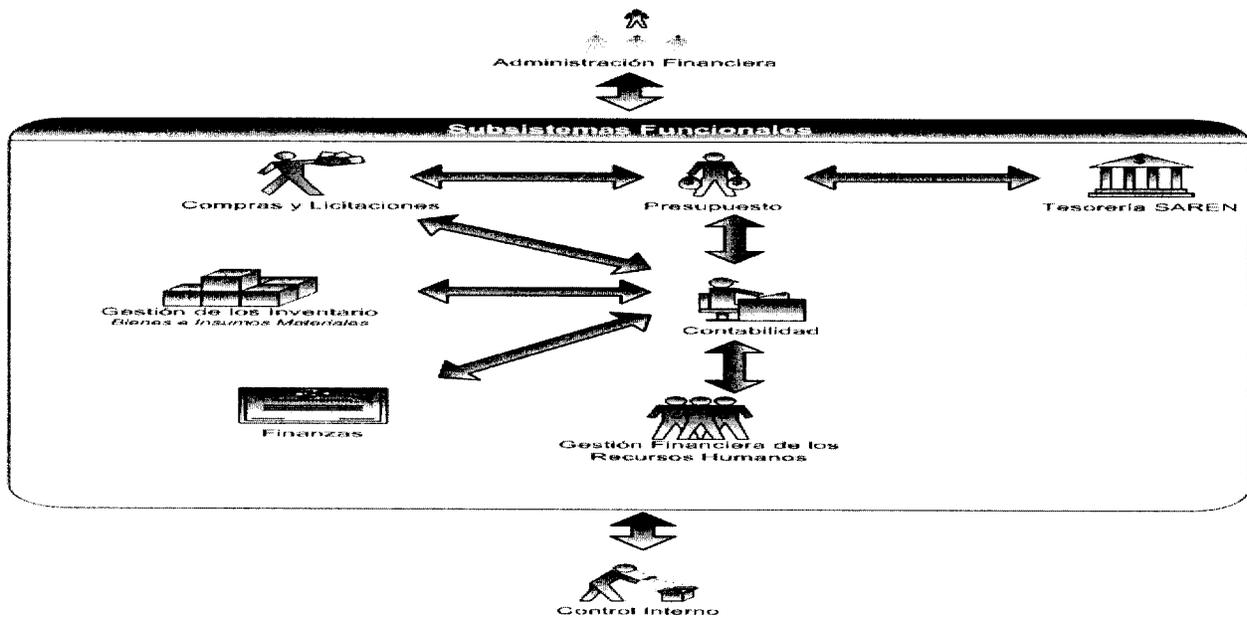
- Etapa I. Pruebas Modulares: Consiste en la Prueba de cada submódulo, aislado del resto del Sistema.
- Etapa II. Pruebas de Integración de primer nivel por oficinas: Consiste en probar los submódulos de forma integrada en la oficina.
- Etapa III. Pruebas de Módulo: Se realizan cuando se han integrado todas los submódulos a nivel de oficina y comienza la integración de todo el Sistema, su objetivo es comprobar que el Sistema satisface los Requisitos especificados.

Los siguientes diagramas muestran las etapas con sus escenarios correspondientes:





Etapa II. Pruebas de Integración a nivel de oficina.



Etapa III. Pruebas de Módulo.

1. 4. Identificación del proyecto

Los documentos usados para elaborar el Plan de Pruebas son los siguientes:

- Especificación de Casos de Uso.
- Especificación de Casos de Pruebas.
- Documento de Arquitectura.
- Informe de No Conformidades.

1. 5. Estrategia de evolución del Plan

El Plan será chequeado diariamente para realizar los ajustes necesarios:

- Al finalizar la jornada del día se reúne el equipo de Prueba para verificar el cumplimiento del Plan y realizar los ajustes necesarios en cuanto al cronograma.
- Se realizará un documento con las modificaciones al cronograma el cual será distribuido entre los participantes de las Pruebas para con ello comunicar los cambios.
- Las propuestas de modificaciones al Plan serán presentadas para su análisis en la reunión de chequeo diaria de conjunto MIJ (Ministerio del Interior y Justicia), CALISOFT y Líder del Proyecto de Registros y Notarías.

A continuación se relacionan los representantes de cada una de las partes que tendrán la responsabilidad de aprobar el Plan de Pruebas:

MIJ	Funcionario Designado por parte del MIJ
CALISOFT	Jefe de CALISOFT
Líder del Proyecto	Gerente del Proyecto

2. Recursos.

2. 1. Roles y responsabilidades.

En la siguiente tabla se muestran los roles, cantidad y responsabilidad que serán empleados en la realización de las Pruebas:

Rol	Cantidad	Responsabilidad
Responsable del proyecto Informático del MIJ	1	Controlar el Plan de Pruebas de Aceptación
Especialistas Informáticos del MIJ	2	Coordinar el trabajo de Pruebas con los directivos del MIJ. Monitorear el Plan de Pruebas de Aceptación. Probar la parte técnica del producto.
Especialistas de CaliSoft	4	Supervisar el trabajo de Pruebas, recogiendo las no conformidades para elaborar el informe de conjunto con los funcionarios del MIJ que ejecutaran las mismas y que será aprobado con los directivos del MIJ. Controlar, Monitorear y Ejecutar el Plan de Pruebas de Aceptación Diseñadores de Pruebas de Software. Probadores de Sistemas Informáticos. Evaluación del proceso de Pruebas y los resultados de las mismas.
Funcionarios adscritos a la Dirección General de Registros y Notarías del MIJ	10	Realizar las Pruebas al producto Verificar y Validar las funcionalidades de la aplicación. Elaboración de conjunto con los especialistas de CaliSoft el informe de las no conformidades.
Desarrollador	4	Instruir a todo el equipo de Prueba de cómo trabajar con el producto. Recepcionar el informe de no conformidades aprobado por los directivos del MIJ para dar respuesta a cada una de las no conformidades o solicitudes nuevas realizadas.

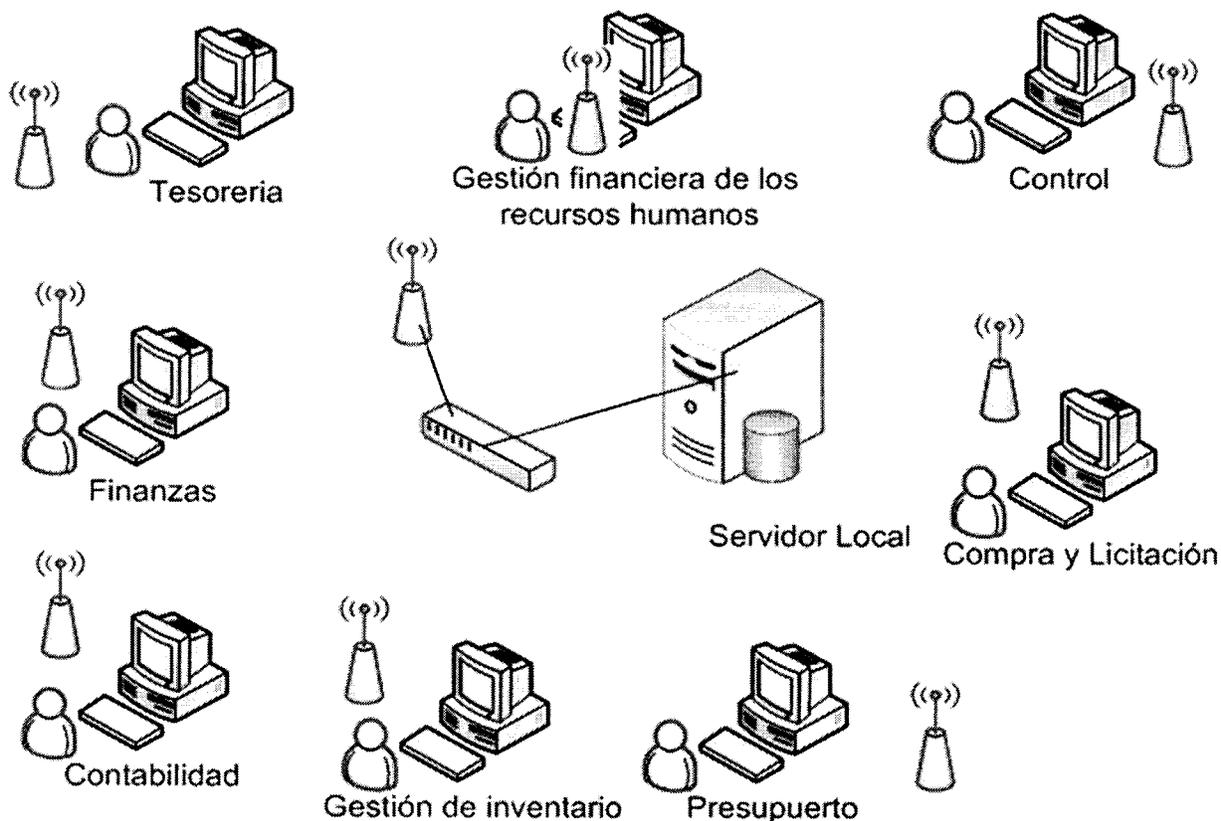
2. 2. Escenario de Pruebas.

Se simulará el escenario de una oficina donde estarán 8 PC, cada una con un submódulo de Administración Financiera, además estarán conectadas a una PC que es el servidor de base de datos y otra que es el servidor de la aplicación, como se muestra a continuación:

- 8 microcomputadoras en redes, de ellas:
- 1 PC para Presupuesto.

- 1 PC Contabilidad.
- 1 Para Control.
- 1 PC para Tesorería.
- 1PC para Finanzas
- 1PC para Compras y Licitaciones.
- 1 PC Para Gestión Financiera de los Recursos Humanos.
- 1 PC para Gestión de Inventario.
- 1 Servidor.
- Una impresora.

Las Pruebas serán realizadas en un ambiente controlado y no de cambio, la siguiente imagen muestra la configuración física descrita:



Escenarios de Pruebas

2. 3. Recursos de Sistema

En la siguiente tabla se establecen los recursos de Sistema necesarios para realizar la verificación. Es recomendable que el Sistema simule el entorno de producción, reduciendo los tamaños de bases de datos si fuera apropiado.

Recursos para la Etapa I, II y III

Recurso	Tipo
Servidor	Pentium IV 3,00 GHz, 2 Gb RAM y 250 Gb de Disco Duro.
PC Cliente para Pruebas	Pentium IV 3,00 GHz, 512 Mb RAM y 80 Gb de Disco Duro.
Requerimientos especiales	Impresora HP LaserJet 2430 DTN
Red o subred	Red Inalámbrica

2. 4. Recursos de Software.

Los Requisitos de Software que necesita el Sistema se aseguran mediante la instalación de todas las máquinas de Prueba con una imagen por el equipo de desarrollo. Esta imagen asegura que las computadoras tengan todos los programas necesarios para el Sistema, además esta misma medida se toma en las computadoras que se utilizarán cuando el Sistema se ponga en funcionamiento.

3. Requerimientos a probar:

Como en este momento Administración Financiera se encuentra en la fase de inicio no se han definido todos los Requisitos, por eso en este momento no se pueden anexar los Requisitos aprobar, cuando se identifiquen deben anexarse a este Plan de Pruebas.

4. Estrategia de las Pruebas de Aceptación.

A continuación se describirá el flujo de trabajo que será implementado durante todo el periodo de ejecución de las Pruebas, de igual forma se detalla las diferentes Estrategias que en cada una de las etapas y fases serán realizadas.

4. 1. Descripción del flujo de trabajo.

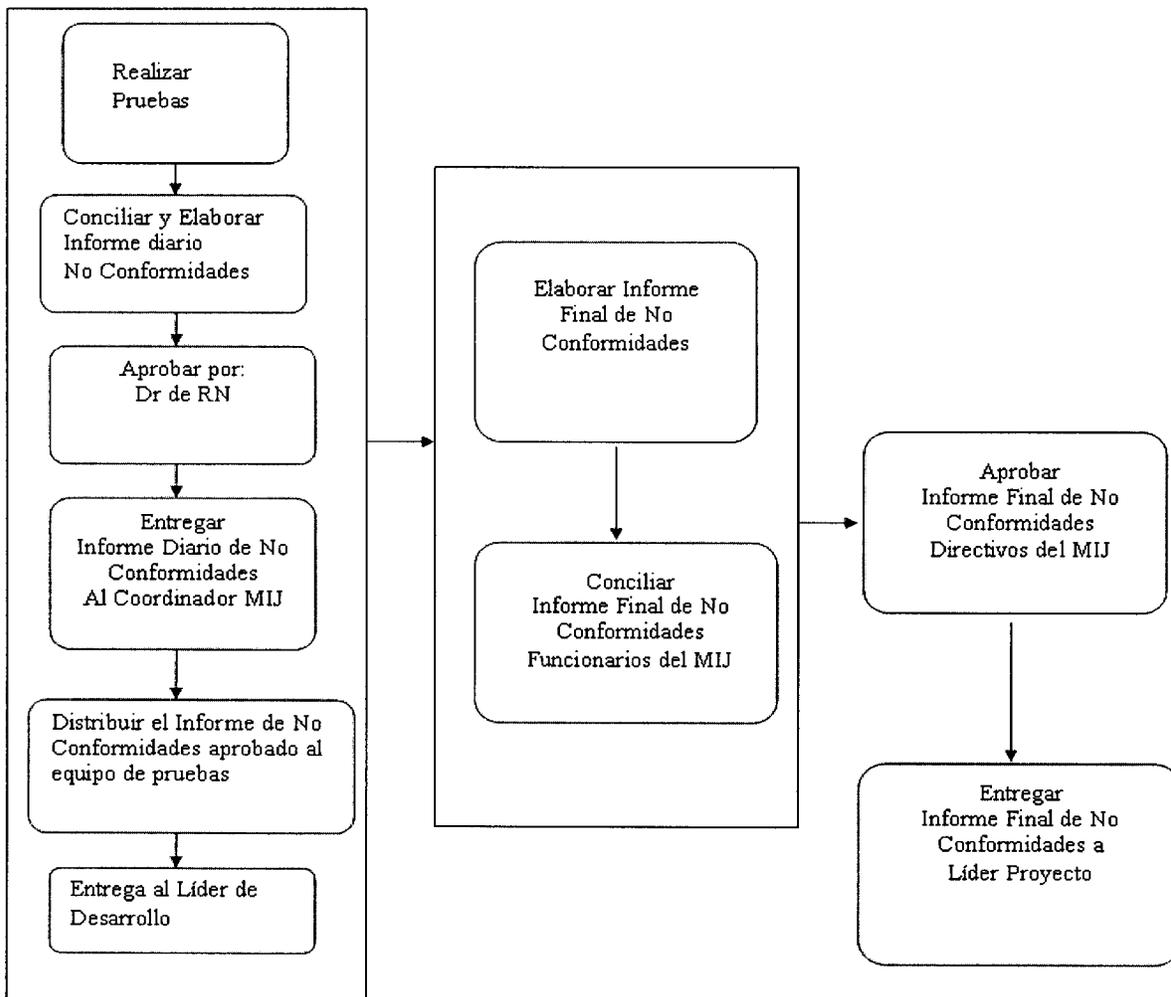
El flujo de trabajo se inicia cuando los Funcionarios del MIJ comiencen a trabajar con el producto para probar sus funcionalidades, en la medida que detecten errores, molestias o incomodidades en el trabajo con el producto, etc., estas serán anotadas de conjunto con el Especialista de CaliSoft que esté asignado, al finalizar el día se expondrán a todo el equipo de Pruebas las no conformidades encontradas, para con ello lograr definir las de conjunto eliminando las repetidas y aclarando los términos para evitar ambigüedades y asignarle un único número que sirva como código de la misma. En todo momento el desarrollador deberá estar atento a cualquier solicitud de ayuda por parte de los probadores.

Al iniciar el día se realizará la conciliación entre los Funcionarios del MIJ que realizan las Pruebas y los Especialistas de CaliSoft para asegurar que las no conformidades encontradas el día anterior fueron descritas correctamente y a su vez asignarle el valor correspondiente a los atributos (tipo de no conformidad, estatus, categoría de falla y nivel crítico definidos en el epígrafe 4 de este documento), las mismas quedarán documentadas en el Informe Diario de No Conformidades el cual será entregado al concluir el día a Director de Registros y Notarías para que sea firmado y entregado a los coordinadores del proyecto del MIJ para su posterior conciliación.

En el caso de las No Conformidades que tengan que ver con insatisfacción de los Clientes con alguna forma del Sistema mostrar los elementos en la interfaz o mensajes (En próximos epígrafes se especificará la clasificación), se solicitará a los Clientes que ofrezcan una propuesta de cómo sería más conveniente mostrar esos elementos en la interfaz. Una vez conciliado el informe es entregado al Líder del proyecto para que este comience a ejecutar los arreglos acordados. Es de señalar que una vez aprobado el informe diario de no conformidades este no podrá ser modificado.

Al concluir el periodo de Prueba se realizará el Informe Final de No Conformidades, atendiendo a los Informes diarios, este será conciliado y firmado por ambas partes. Este Informe Final de No Conformidades será aprobado por los Directivos del MIJ de manera formal.

En la siguiente figura se muestra de forma general el flujo de trabajo:



Flujo de trabajo

Para la elaboración de los Informes se utilizarán las Plantillas acordadas por ambas partes para lograr la estandarización de la documentación.

4. 2. Descripción de las Estrategias y tipos de Pruebas.

Todas las Pruebas se realizarán de forma manual y se probarán todas las funcionalidades implementadas en el Módulo según la metodología del Cliente.

Estas Pruebas se harán en cada una de las etapas teniendo en cuenta las siguientes fases:

- *Primera Fase:* Organización del escenario de Pruebas y capacitación del equipo de Pruebas.
- *Segunda Fase:* Realización de Pruebas de funcionalidad. El principal objetivo de este tipo de Prueba es medir la correspondencia entre el documento Requisitos funcionales y las funciones que realmente fueron implementadas en el Sistema. Además se realizarán Pruebas que nos permitan determinar si la lógica del flujo básico del Sistema es la correcta. Se realizaran Pruebas de Regresión para chequear que los errores ya corregidos no se vuelvan a repetir y/o no se dañen funcionalidades que funcionaban correctamente.
- *Tercera Fase:* Se realizarán Pruebas de Seguridad con el objetivo de validar la protección de la aplicación sensible a entradas no deseadas.
- *Cuarta Fase:* Aplicación de la Lista de chequeo y los principios de Calidad.
- *Quinta Fase:* Conciliación de los documentos entregables resultados de las Pruebas, Aprobación y firma de los mismos.

4. 2. 1. Pruebas funcionales.

La Prueba de funcionalidad se enfoca en requerimientos para verificar que se corresponden directamente a casos de usos o funciones y reglas del negocio. Los objetivos de estas Pruebas son verificar la aceptación de los datos, el proceso, la recuperación y la implementación correcta de las reglas del negocio. Este tipo de Prueba se basa en técnicas de Caja Negra que consisten en verificar la aplicación y sus procesos interactuando por medio de la interfaz de usuario y analizar los resultados obtenidos.

Objetivo de la Prueba:

Asegurar la funcionalidad apropiada de los Módulos, incluyendo el flujo de trabajo, entrada de datos, proceso y recuperación.

Técnica:

Ejecute cada proceso o función usando datos válidos y no válidos, para verificar lo siguiente:

- Se obtienen los resultados esperados cuando se usan datos válidos.
- Cuando se usan datos no válidos se despliegan los mensajes de error o advertencia apropiados.
- Se aplica apropiadamente cada regla del negocio.

Criterio de aceptación:

Todas las Pruebas Planificadas se realizaron. Todos los defectos encontrados han sido debidamente identificados y documentados.

4. 2. 2. Prueba de interfaz del usuario.

Esta Prueba verifica que la interfaz de la aplicación proporcione al usuario el acceso y navegación a través de las funciones apropiadas. Además asegura que los objetos presentes en la interfaz de usuario se muestren como se espera y conforme a los estándares establecidos. Se verificará lo siguiente:

- Navegación a través de todas las funcionalidades, verificar que cada interfaz es amigable al usuario.
- Verificar las funciones de Ayuda.

Objetivo de la Prueba:

Verificar que la navegación a través de los elementos que se están probando reflejen las funciones del negocio y los requerimientos solicitados, de acuerdo a los estándares establecidos, incluyendo las validaciones de los campos, métodos de acceso como los menús y las opciones de ayuda.

Técnica:

Crear o modificar Pruebas para cada rol verificando la navegación y los estados de los objetos para cada interfaz y cada objeto contemplado dentro de la misma.

Criterio de aceptación:

Cada interfaz ha sido verificada exitosamente siendo consistente con una versión de referencia o estándar establecido en la Lista de chequeo del Anexo 1A.

4. 2. 3. Prueba de seguridad y control de acceso.

La Prueba de Seguridad y Control de Acceso se enfoca en dos áreas de seguridad:

- Seguridad en el ámbito de aplicación, incluyendo el acceso a los datos y a las funciones de negocios asociadas a cada rol de usuario.
- Seguridad en el ámbito de Sistema, incluyendo conexión, o acceso remoto al Sistema.

La seguridad en el ámbito de aplicación asegura que, los usuarios solo accedan a los procesos de acuerdo al rol asignado.

La seguridad en el ámbito de Sistema asegura que, solo los usuarios con derecho a acceder al Sistema son capaces de acceder a las aplicaciones y solo a través de los puntos de ingresos apropiados.

Objetivo de la Prueba.

Seguridad en el ámbito de aplicación:

- Verificar que un usuario pueda acceder solo a las funciones o datos para los cuales su tipo de rol tiene permiso.
- Seguridad en el ámbito de Sistema:
- Verificar que solo los roles con acceso al Sistema y/o red y a las aplicaciones, puedan acceder a ellos.

Técnica:

Seguridad en el ámbito de aplicación:

- Identificar y hacer una Lista de cada tipo de usuario y las funciones y datos sobre las que cada tipo tiene permiso.

- Crear Pruebas para cada tipo de usuario y verificar cada permiso creando operaciones específicas para cada tipo de usuario.
- Modificar el tipo de usuario y volver a ejecutar las Pruebas para los mismos usuarios. En cada caso, verificar que las funciones o datos adicionales están correctamente disponibles o sean denegados.

Acceso en el ámbito de red:

- Este tipo de Pruebas está centrada en la seguridad de la red, la misma no será realizada por equipo de Pruebas, para ello el Cliente contratará una empresa que certifica la seguridad de la misma.

Criterio de aceptación:

Para cada tipo de rol, las funciones y datos apropiados de la aplicación están disponibles y todas las operaciones funcionan como se espera y se ejecutan las Pruebas de funcionalidad de la aplicación.

4. 2. 4. Prueba de Fallas y Recuperación

Las Pruebas de Fallas y Recuperación aseguran que el Software puede recuperarse de fallas de Hardware, Software o mal funcionamiento de la red sin pérdida de datos o de integridad de los datos.

La Prueba de Recuperación es un proceso en el cual el Sistema se expone a condiciones extremas, o condiciones simuladas, para causar falla, como fallas en dispositivos de Entrada/Salida. Los procedimientos de recuperación se invocan y el Sistema es monitoreado e inspeccionado para verificar que se recupera apropiadamente y que de igual forma se logra la recuperación de los datos.

Objetivo de la Prueba:

Verificar que los procesos de recuperación (manual o automáticos) recuperen apropiadamente la base de datos y los diferentes Módulos del Sistema a un estado conocido y deseado. En la Prueba se incluyen los siguientes tipos de condiciones entre otros aún por definir:

- Interrupción de energía al Cliente.
- Interrupción de energía al servidor.

- Interrupción de comunicaciones mediante los servidores de la red.
- Interrupción de comunicación o pérdida de energía de los discos del servidor o con los controladores.
- Ciclos incompletos (procesos de filtro de datos interrumpidos, procesos de sincronización de datos interrumpidos).
- Elementos de datos en la base de datos inválidos o corruptos.

Técnica:

Se deben usar las Pruebas creadas para probar Funcionalidad y procesos de negocio para crear una serie de operaciones. Una vez logrado el punto de comienzo deseado, se deben realizar o simular las siguientes acciones, individualmente:

- Interrumpir la energía del Cliente: apagar la PC.
- Interrumpir la energía del servidor: simular o iniciar el proceso de apagado del servidor.
- Interrupción por medio de los servidores de red: simular o iniciar la pérdida de comunicación con la red (desconectar físicamente la comunicación o apagar el servidor de red o router)
- Interrumpir la comunicación o quitar la energía de los discos del servidor o sus controladores: simular o eliminar físicamente la comunicación con uno o más controladores de disco o los discos.
- Una vez que se lograron o simularon estas condiciones, se deben invocar los procedimientos de recuperación.
- Las Pruebas de ciclos incompletos utilizan la misma técnica excepto que los procesos de bases de datos deben ser abortados a sí mismos o terminados prematuramente.
- La última Prueba requiere que se logre un estado conocido de la base de datos. Se deben corromper manualmente campos de la base de datos, punteros y claves trabajando directamente sobre la base de datos (utilizando herramientas). Se deben ejecutar las Pruebas de Funcionalidad y procesos de negocio y verificar que los ciclos se completen.

Criterio de aceptación:

En todos los casos, los Módulos, la base de datos y el Sistema deben, en la realización de procedimientos de recuperación, volver a un estado conocido y deseable. Este estado incluye corrupción de datos limitada a los campos, punteros o claves corruptos conocidos, y reportes indicando los procesos u operaciones que no se completaron debido a las interrupciones.

Consideraciones especiales

Los procedimientos para desconectar cables (simulando falta de energía o pérdida de comunicación) no son deseables o factibles. Se pueden requerir métodos alternativos, como Software de diagnóstico. Se requieren los grupos de recursos de Sistemas, Bases de datos y Red.

Estas Pruebas deben ejecutarse fuera del horario de trabajo normal o en una máquina aislada.

4. 2. 5 Prueba de diseño informacional.*Objetivo de la Prueba:*

Verificar que el diseño de la interfaz y de los reportes entre los diferentes submódulos que componen al Sistema mantenga las mismas pautas de diseño grafico.

Técnica:

A partir de la definición de las pauta del diseño informacional aplicados al Sistema, chequear que estas se cumplan en todos los Módulos. Se deberá considerar el cumplimiento de aspectos tales como:

- Color
- Tipo y tamaño de la letra tanto para la entrada de datos como para la salida de los mismos.
- Similitud en el diseño gráfico y distribución de los componentes en las interfaces y reportes.
- Contenido de los mensajes que emite la aplicación para eventos como: errores, alertas, información, ayudas, etc.

Se visualizarán las interfaces para realizar las comparaciones, de igual forma se imprimirán los reportes para su chequeo.

Criterio de aceptación:

Que se cumplan las pautas establecidas.

4. 2. 6. Prueba de referencia cruzada

Las no conformidades detectadas en un Módulo serán provocadas en el resto para ver la reacción de este.

Objetivo de la Prueba:

Verificar que los componentes genéricos implementados para su uso en todos los Módulos funcionen correctamente, para ello se deberá realizar una Lista de dichos componentes especificando la funcionalidad que implementa para poder aplicar las Pruebas.

Técnica:

Al detectarse un error en una funcionalidad de un Módulo y que pudiera haber sido implementado en cualquiera de los restantes se tomaran las condiciones bajo las cuales ocurrió el mismo y se repetirá en el resto de los Módulos para comprobar si este es general a todo el Sistema.

4. 2. 7 Prueba Exploratoria.

La Prueba exploratoria consiste en hacer una revisión no detallada de la aplicación antes de comenzar el proceso de Pruebas establecido.

Objetivo de la Prueba:

Verificar que el Sistema está listo para ser probado y que no existen problemas con pasar por las distintas funcionalidades.

Técnica:

Chequear sin entrar al detalle las funcionalidades, para verificar que no exista ningún problema que impida que se pruebe alguna funcionalidad.

Criterio de aceptación:

Que no exista ningún problema que impida probar cada una de las funcionalidades.

5. Criterios de aceptación del producto y evaluación de las Pruebas.

En la siguiente tabla se muestran los criterios de aceptación para cada una de las etapas.

Etapas	Criterios de Aceptación
I	La cantidad de no conformidades detectadas sin solución inmediata no debe ser mayor de 5 en cada Módulo.
II	A definir.
III	A definir.

Etapas

Para cada una de los submódulos definidos en la etapa I, se emitirá el documento Acta de Aceptación, el cual será debidamente firmado por las autoridades competentes del MIJ. El paso a la etapa II de Prueba está condicionado por las combinaciones que necesariamente tienen que ser probadas en ese escenario y que están descritos en el epígrafe 1. 2 de este documento. Como resultado de las Pruebas de Aceptación cada no conformidad deberá ser analizada para definir los atributos de estatus, tipo de no conformidad, categoría y nivel de la falla atendiendo a los valores definidos por el Cliente y que son descritos a continuación:

Valor	Descripción
0	Inconformidad No resuelta.
1	Inconformidad Resuelta
2	Inconformidad no satisfactoria.
3	Pendiente Cliente
4	Reincidencia de la falla

Nivel de falla

La categoría de la falla hace referencia al nivel de dificultad de la misma, estableciendo a su vez un tiempo estimado para la solución de la falla de acuerdo al nivel donde se encuentre ubicada.

Categoría	Descripción	Ejemplo	Tiempo estimado para resolver la falla
1	Efectos no significativos en la funcionalidad y usabilidad del Sistema. Errores de forma (Amigabilidad del Sistema). Capa de Presentación.	Disconformidad del desarrollo/proceso formal. Disconformidad de la documentación. Disconformidad superficial del Sistema. Palabra mal escrita.	1 día.
2	La funcionalidad del Sistema no está afectada hasta tal punto que el Sistema no puede emplearse. Capa de Negocios.	Validaciones de campos, fechas, etc. Errores en los reportes.	2 días
3	El Sistema se detiene y no puede avanzar. Capa de Negocio y Capa de Datos.	Genera error al guardar un trámite.	1 semana

Categoría

Nivel de Importancia	Descripción	Justificación
3	Alta	Afecta procesos del negocio, e impide el correcto funcionamiento del proceso para la consecución del fin.
2	Media	Procesos de validación, esquemas, etc. Generación de reportes.
1	Baja	Errores ortográficos.

Nivel de importancia

Tipo de No Conformidad:

Atendiendo a que no solamente se estarán revisando las no conformidades sino que también se están generando solicitudes de cambio ya sea por la introducción de un nuevo requerimiento como la modificación de uno existente se hace necesario introducir el término de tipo de no conformidad y los valores que podrá tomar serán:

Valor	Tipo de No Conformidad	Descripción
NC	No Conformidad	Errores
NR	Nuevo Requisito	Nuevos procesos a implementar, nuevos reportes, etc.
MR	Modificación de Requisito	Modificación a Requisitos ya implementados

Tipo de no conformidad

El tiempo de respuesta a los tipos de no conformidades cuyo tipo sea NR y MR de acuerdo con las necesidades del Cliente, deberá ser negociado entre los líderes de ambas partes, atendiendo al nivel de complejidad y el impacto que tiene dicha solicitud en el proyecto.

5. 1. Evaluación de las Pruebas.

Una vez concluido el período de Pruebas se realizará la evaluación de la misma siendo aprobada si se cumplen los siguientes criterios:

- Asistencia y puntualidad por parte del equipo de probadores (Funcionarios Adscritos del MIJ) en 80 por ciento. Para lograr esto se chequeara diariamente la asistencia mediante el registro de asistencia al laboratorio.
- Cumplimiento del Plan de Pruebas de Aceptación y en especial del flujo de trabajo descrito en el epígrafe 4. 1 de este documento.

La evaluación quedará reflejada en el documento “Evaluación de las Pruebas de Aceptación” y el mismo será elaborado por CaliSoft de conjunto con los Especialistas en Informática del MIJ.

Artefactos de entrega.

1. Planificación de las Pruebas.

Describir la Estrategia de Prueba, los recursos humanos y de tiempo necesarios para acometerlos.

Artefactos que se generan:

- Plan de Pruebas de Aceptación.
- Lista de chequeo.

2. Ejecutar las Pruebas

- Se ejecutan las Pruebas.

Artefactos que se generan:

- Informe Diario de No conformidades.

3. Evaluar los resultados de las Pruebas

Artefactos que se generan

- Informe Final de No conformidades.
- Acta de aceptación del producto.

6. Cronograma de trabajo

En la medida que sean aceptados los submódulos y vencidas las etapas se elaborará el cronograma de trabajo. Debido a que es desconocida la fecha de inicio de las Pruebas, en el cronograma se utilizan los días de Pruebas y no fechas específicas. Se le pidió al equipo de desarrollo que hiciera una estimación de la cantidad de casos de usos que tendría el Módulo, el número estimado fue de 100 casos de usos. Sobre esa base se hizo una estimación para cada actividad, para eso se consultó a personas que poseen experiencia en el desarrollo de Pruebas de Software y pueden hacer una estimación tomando como referencia el tiempo empleado en las Pruebas de otros Sistemas.

No.	Tarea	Fecha	Responsable	Participantes	Observaciones
	Organizar las Pruebas de Aceptación	Todo el período de Pruebas.	CaliSoft	Desarrolladores, CaliSoft, MIJ	
	Montaje del escenario de Pruebas.	Día 1 al 4	CaliSoft	Desarrolladores	
	Coordinaciones de inicio de las Pruebas de Aceptación y conciliación del pre-Plan de Pruebas	Día 1 al 2	CaliSoft	CaliSoft, MIJ	
	Definir el Plan de Pruebas de Aceptación	Día 4 al 10	CaliSoft	CaliSoft, MIJ	
	Aprobación del Plan de Pruebas de Aceptación y firma del mismo.	Día 10 al 11	CaliSoft, MIJ	CaliSoft, MIJ	
	Realizar Pruebas por Calisoft	Día 1 al 10	CaliSoft	Calisoft, MIJ, Desarrolladores	
	Organización del equipo de Prueba con Funcionarios adscritos a la Dirección de Registros y Notarías del MIJ.	Día 4 al 10	Calisoft	CaliSoft, MIJ, Desarrolladores	
	Inicio de las Pruebas	Día 11	CaliSoft	CaliSoft, MIJ, Desarrolladores	
	Realizar ciclo final de Pruebas.	Día 20 al 24	CaliSoft	CaliSoft, MIJ, Desarrolladores	
	Análisis de las no conformidades por parte del equipo de Pruebas.	Día 24 al 26	CaliSoft	CaliSoft, MIJ, Desarrolladores	Se realizará diariamente a las 4. 00 PM
	Elaborar y Conciliar el Informe diario de No Conformidades	Día 26 al 28	CaliSoft.	CaliSoft, MIJ	
	Entregar para aprobar y firmar el Informe diario de No Conformidades	Día 29	MIJ	MIJ	
	Distribuir el Informe diario de No Conformidades firmado al equipo de Pruebas	Día 30	MIJ	CaliSoft, MIJ y Líder del proyecto	La distribución del informe se realizará al inicio de la jornada de trabajo.
	Elaborar Informe final de No Conformidades	Día 31	CaliSoft	CaliSoft, MIJ	
	Conciliar, firmar y entregar el Informe Final de No Conformidades.	Día 32	CaliSoft	CaliSoft, Funcionarios MIJ que realizaron las Pruebas y Desarrolladores	

	Entrega del Informe Final de No Conformidades aprobado y firmado por el directivo competente del MIJ	Día 33	MIJ		
	Elaborar documento Evaluación de las Pruebas de Aceptación	Día 33 al 35	CaliSoft		
	Elaborar el cronograma de trabajo para la II Etapa	Día 35 al 37			
	Presentar el cronograma de trabajo para la II Etapa	Día 38			
	Aprobar el cronograma de trabajo de la II Etapa	Día 39			

Cronograma de Realización de las Pruebas.

Anexo 1A**EVALUACIÓN DE LA INTERFAZ DE USUARIO**

Proyecto _____ Módulo: _____

Control No: _____ Etapa: _____

Fecha: _____

Nivel	Evaluación	Eval	NP	Comentario
!!	¿La interfaz corresponde con las características, metas y nivel de experiencia de los usuarios de esta aplicación?			
!!	¿Está la información libre de errores gramaticales, deletreado, ortográficos y de los errores tipográficos? (estas clases de errores no solamente indican una carencia del control de Calidad, pero pueden producir realmente inexactitudes en la información.)			
!	¿Existe el menú de la "ayuda"?			
!	¿Los comandos y las opciones apropiadas están en cada menú?			
!!	¿Cada una de las opciones dan las respuestas apropiadas?			
!!	¿Se tiene ausencia de errores críticos al probar todas las opciones?			
!	¿Cada comando de menú tiene una secuencia hot-key la cuál lo invocará cuando sea apropiado?			
!	En cajas de diálogo "tabuladas", los nombres de la lengüeta o pestañas no son abreviaturas			
!	En cajas de diálogo "tabuladas", ¿las lengüetas o pestañas se pueden alcanzar vía combinaciones de hot-key apropiadas?			
!	¿El uso apropiado de la tecla escape es correcto? (la cuál debe cancelar cualquier cambio que se haya realizado).			
!	¿Los botones de aceptación y de cancelación están agrupados de forma separada de los otros botones de comandos?			
!	Los nombres de los botones de comandos no son abreviaturas			
!	Los nombres de los botones de comandos no son etiquetas técnicas, pero si nombres que tienen un significado para los usuarios del Sistema			
!	¿Los botones de comando son de tamaño y forma similares?			
!	¿Cada botón de comando se puede alcanzar vía combinación de hot-key? (excepto los botones de ACEPTAR y de CANCELAR).			

!	Los nombres de los botones de opción (botón de radio) no son abreviaturas			
!	¿Las cajas de opción, los botones de opción, y los botones de comando están agrupados lógicamente juntos en áreas claramente demarcadas?			
!	¿El color rojo no está siendo utilizado para destacar objetos activos de la interfase?			

Elementos de diseño

Nivel	Evaluación	Eval	NP	Comentario
!!	¿La aplicación funciona en el Hardware y Software propuesto?			
!	¿Ayuda la aplicación a descubrir y corregir los errores de los usuarios?			
!!	¿En la aplicación aparecen todas las funcionalidades especificadas en la documentación?			
!!	¿Ayuda la aplicación a que los usuarios lleven a cabo sus tareas?			

Atributos de Calidad

Nivel	Evaluación	Eval	NP	Comentario
!	¿Es simple el vocabulario utilizado?			
!	¿Se proporciona tiempo suficiente para realizar las entradas por teclado?			
!	¿Hay algún tipo de asistencia para los usuarios que hacen uso del Sistema por primera vez?			
!	¿Proporciona el Sistema un soporte apropiado a los usuarios más novatos?			
!	¿Se entienden la interfaz y su contenido?			
!	¿Asiste el Sistema al usuario de forma efectiva proporcionando el modo más efectivo de hacer las tareas en caso de que no haya una única forma de hacerlas?			
!!	¿Está diseñada la interfaz para facilitar la realización eficiente de las tareas de la mejor forma posible?			

!	¿Es fácil de utilizar el Sistema en la realización de tareas?			
!	¿Es apropiada la retro alimentación presentada por el Sistema?			
!	¿Actúa el Sistema en la prevención de errores?			
!!	¿Actúa el Sistema en la información de los errores?			
!	¿Se permite la utilización del ratón o el teclado?			
!	¿Permite al usuario interrumpir su tarea y continuar más tarde?			
!	¿Se utiliza mensajes y textos descriptivos?			
!!	¿Permite deshacer las acciones, e informar el estado?			
!!	¿Permite una cómoda navegación dentro del producto y una fácil salida de éste?			
!	¿Permite distintos niveles de uso del producto para usuarios con distintos niveles de experiencia?			
!!	¿Se proporciona información visual de dónde está el usuario, qué está haciendo y qué puede hacer a continuación?			
!	¿Se presenta al usuario la información que sólo necesita?			
!	¿Existen diferentes niveles de ayuda?			

Usabilidad

Nivel	Evaluación	Eval	NP	Comentario
!	¿Se recupera el Software ante fallas?			
!	¿La recuperación ante fallas se realiza en el tiempo requerido para la aplicación?			

Confiabilidad

Nive I	Evaluación	Eval	NP	Comentario
!	¿Las respuestas del Sistema se realizan en el tiempo requerido para la aplicación?			

Eficiencia

LEYENDA:

Nivel: Importancia del aspecto a evaluar

E: Evaluación

NP: No Procede

Comentario: Es obligatorio en las respuestas negativas

Las evaluaciones serán:

- Malo - propiedad no disponible(0)
- Satisfactorio - propiedad parcialmente disponible(2)
- + Bien - propiedad disponible(4)

++ Excelente - propiedad muy bien implementada(5)

El peso de la pregunta será:

!! Muy importante (5)

! Menos importante (3)

Anexo2 Casos de Prueba.

Anexo 2 A

Diseño de Casos de Prueba
Elemento de Configuración: Sistema, Documentación
Nombre del Caso: Reporte Contable del Libro Mayor

Registros y Notarías
Módulo Administración Contable
Versión

Registro de versiones:

Fecha	Versión	Descripción	Autor
20/04/07	<1. 0>	Caso de Prueba 1	Nemury Silega

Descripción General

Este caso de uso consiste en mostrar a través de un reporte a nivel de cuentas contables el saldo inicial, debe, haber y saldo final de las mismas. Partiendo de la información que van generando los comprobantes, en este caso de uso se contempla la posibilidad de visualizar o imprimir el reporte

Pruebas a realizar:

- Pruebas funcionales.
- Prueba de interfaz de usuario.

1. CPR 1: Mostrar libro mayor

Descripción

El Sistema muestra el Libro Mayor, luego, de ser seleccionados el nivel de cuentas y el intervalo de fecha.

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación y el contador selecciona en el menú de la izquierda la opción: "Contabilidad", "Reportes contables" y dentro de ella: "Reportes internos", luego "Reporte Libro Mayor".
- El Sistema muestra la interfaz: "Reporte libro mayor".
- Se selecciona el nivel de cuenta y el intervalo de fecha correspondiente.
- El usuario presiona el botón "Visualizar" en la parte inferior aparecen los datos.

Condiciones de Ejecución

Tienen que existir comprobantes contables. Antes de que se elija la opción de visualizar el reporte contable del libro mayor deben estar especificados los niveles de cuenta a acceder y los rangos de fechas posibles a ver el libro mayor (El rango de fecha no puede ser superior al almacenado en la sincronización).

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba
Nivel Cuenta: Rubro Desde: 01/04/2007 Hasta: 01/04/2007		Se obtiene el reporte contable del libro mayor generado listo para ser impreso.	Se obtiene el reporte contable del libro mayor generado listo para ser impreso.
	Nivel Cuenta: Prueba Desde: 01/01/2007 Hasta: 01/01/2007	El Sistema no permite la entrada de otros caracteres.	El Sistema no permite la entrada de otros caracteres.
	Nivel Cuenta: Rubro Desde: 01/05/2007 Hasta: 10/04/2007	El Sistema no permite que la fecha Desde sea mayor que la fecha Hasta.	El Sistema no permite la entrada de otros caracteres.
	Nivel Cuenta: Rubro Desde: 01/01/2006 Hasta: 05/01/2006	El Sistema no permite la entrada de una fecha posterior a la de sincronización.	El Sistema no hace nada

2. CPR 2: Imprimir reporte.

Descripción

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación y el contador selecciona en el menú de la izquierda la opción: "Reportes contables" y dentro de ella: "Reportes internos", luego "Reporte del libro mayor".
- El Sistema muestra la interfaz: "Reporte libro mayor".
- Se selecciona el nivel de cuanta y el intervalo de fecha correspondiente.
- El usuario presiona el botón "Visualizar" en la parte inferior aparecen los datos.
- El Sistema muestra el reporte.
- El usuario presiona el botón "Imprimir".
- El Sistema visualiza la interfaz "Reporte".
- El usuario presiona el botón correspondiente a imprimir reporte.

Condiciones de Ejecución

- Debe existir un reporte del libro mayor visualizado.

Iteraciones

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Ordenar imprimir		El Sistema imprime el reporte del libro mayor.	El Sistema imprime el reporte del libro mayor.	

Registro de defectos y dificultades detectados

Elemento	No	Descripción de la No conformidad	Aspecto correspondiente	Etapas de la detección	Importancia	Recomendaciones
Sistema	1	El Sistema permite que se entre fecha superior a la de sincronización	CPR1	Ejecución de la Pruebas	Media	Revisar en todos los casos

Anexo 2B

Diseño de Casos de Prueba
Elemento de Configuración: Sistema, Documentación
Nombre del Caso: Reporte Diario General

Registros y Notarías
Módulo Administración Contable
Versión

Registro de versiones:

Fecha	Versión	Descripción	Autor
20/04/07	<1. 0>	Caso de Prueba 2	Nemury Silega

Descripción General

Este caso de uso consiste en Listar los comprobantes contables de cada día.

Pruebas a realizar:

- Pruebas Funcionales.
- Pruebas de interfaz de usuario.

Funcionalidades a probar:

- Visualizar reporte Diario General.
- Imprimir reporte.
- Exportar reporte.

1. CPR 1: Visualizar Reporte Diario General.

Descripción

El Sistema muestra una tabla al usuario con el reporte diario general cuando se da Clic en el botón visualizar.

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación y el contador selecciona en el menú de la izquierda la opción: "Contabilidad", dentro de ella: "Reportes Contables", luego "Reportes Internos" y por último la opción: "Reporte Diario General".
- El Sistema muestra la interfaz: "Reporte Diario General".
- El usuario o contador selecciona el intervalo de fecha **Desde-Hasta** a buscar, y presiona el botón "Visualizar".
- El Sistema visualiza el reporte diario.

Condiciones de Ejecución

- Debe existir el reporte diario general.

Iteraciones

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Desde: "01/04/2007" Hasta: "20/04/2007"		El Sistema visualiza el reporte.	El Sistema visualiza el reporte	
	Desde: "01/05/2007" Hasta: "01/04/2007"	El Sistema no permite que la fecha de inicio sea mayor que la de fin.	El Sistema visualiza el reporte	
Desde: "01/09/2007" Hasta: "04/09/2007"		El Sistema debe informar que no se han encontrado resultados.	El Sistema informa que no se han encontrado resultados.	

2. CPR 2: Imprimir el Reporte.

Descripción

Se comprueba la posibilidad de que se pueda imprimir el o los reportes seleccionados.

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación y el contador selecciona en el menú de la izquierda la opción: "Contabilidad", dentro de ella: "Reportes Contables", luego "Reportes Internos" y por último la opción: "Reporte Diario General".
- El Sistema muestra la interfaz: "Reporte Diario General".
- El usuario o contador selecciona el intervalo de fecha Desde Hasta a buscar y presiona el botón "Visualizar".
- El Sistema visualiza el reporte diario.
- El usuario presiona el botón "Imprimir".
- El Sistema presenta la interfaz: "Reporte".
- El usuario presiona el botón correspondiente a imprimir reporte.

Condiciones de Ejecución

- Debe existir el reporte diario general.

Iteraciones

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Presiona el botón correspondiente a imprimir		El Sistema imprime el reporte buscado.		

3. CPR 3: Exportar el Reporte.

Descripción

Se comprueba que se pueda exportar el reporte diario general a uno de los formatos anteriormente especificados.

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación y el contador selecciona en el menú de la izquierda la opción: "Contabilidad", dentro de ella: "Reportes Contables", luego "Reportes Internos" y por último la opción: "Reporte Diario General".
- El Sistema muestra la interfaz: "Reporte Diario General".
- El usuario o contador selecciona el intervalo de fecha Desde Hasta a buscar y presiona el botón "Visualizar".
- El Sistema visualiza el reporte diario.
- El usuario presiona el botón "Imprimir".
- El Sistema presenta la interfaz: "Reporte".
- El usuario presiona el botón correspondiente a "Exportar Reporte" y selecciona el formato en que lo desea exportar.

Condiciones de Ejecución

- Deben existir reportes correspondientes a ese tipo de ingreso.

Iteraciones

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Presiona el botón correspondiente a exportar.		El Sistema exporta el reporte buscado.	El Sistema exporta el reporte.	

Anexo 2C

Diseño de Casos de Prueba
Elemento de Configuración: Sistema, Documentación
Nombre del Caso: Reporte Contable Balance General de Saldos

Registros y Notarías
Módulo Administración Contable
Versión

Registro de versiones:

Fecha	Versión	Descripción	Autor
20/04/07	<1. 0>	Caso de Prueba 3	Nemury Silega

Descripción General

Este caso de uso consiste en realizar un Reporte Contable del registro mostrando en cualquier nivel de las cuentas los saldos de las mismas en el período contable analizado y acumulado en el Ejercicio Fiscal. Partiendo de la información que van generando los comprobantes, en este caso de uso se contempla la posibilidad de visualizar e imprimir el reporte.

Pruebas a realizar

- Pruebas funcionales.
- Pruebas de interfaz de usuario.

Funcionalidades a probar.

- Visualizar Balance de Comprobación de Saldos
- Imprimir reporte.
- Exportar reporte.

1. CPR 1: Visualizar Balance de Comprobación de Saldos.

Descripción

Una vez seleccionado el nivel de cuenta y el intervalo de la fecha satisfactoriamente, si no existen cuentas con saldos inversos a su naturaleza el usuario puede mandar a visualizar el reporte.

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación y el contador selecciona en el menú de la izquierda la opción: "Contabilidad" y dentro de ella: "Reportes Contables", luego "Reportes Externos" y por último la opción "Balance de Comprobación".
- El Sistema muestra la interfaz: "Reporte Balance de Comprobación".
- El usuario o contador introduce los datos a insertar:
 - Nivel Cuenta.

- Desde.
 - Hasta.
- Presiona el botón “Visualizar”.
 - El Sistema muestra el reporte.

Condiciones de Ejecución

- La fecha y el nivel de cuenta deben existir en la base de datos.

iteraciones

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Nivel Cuenta: Cuenta Desde: 01/04/2007 Hasta: 02/04/2007		Se muestra el reporte contable del balance de Comprobación de saldos generado listo para ser impreso.	Se muestra el reporte contable del balance de Comprobación de saldos generado listo para ser impreso.	
	Nivel Cuenta: Prueba1 Desde: 01/04/2007 Hasta: 04/04/2007	El Sistema no permite la entrada de otros caracteres.	El Sistema no permite la entrada de otros caracteres.	
	Nivel Cuenta: Rubro Desde: 05/04/2007 Hasta: 01/04/2007	El Sistema no permite que el campo Desde sea mayor que Hasta .	El Sistema no permite que el campo Desde sea mayor que Hasta	

2. CPR 2: Imprimir Reporte de Balance de Comprobación de Saldos.

Descripción

El Sistema brinda la posibilidad de imprimir los reportes.

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación y el contador selecciona en el menú de la izquierda la opción: “Contabilidad” y dentro de ella: “Reportes Contables”, luego “Reportes Externos” y por último la opción “Balance de Comprobación”.

- El Sistema muestra la interfaz: “Reporte Balance de Comprobación”.
- El usuario o contador introduce los datos a insertar.
- Presiona el botón “Visualizar”.
- El Sistema muestra el reporte.
- El usuario presiona el botón “Imprimir”.
- El Sistema visualiza la interfaz “Reporte”.
- El usuario presiona el botón correspondiente a imprimir reporte.

Condiciones de Ejecución

- Debe existir un reporte de Balance de Comprobación de Saldos.

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Ordenar imprimir el reporte.		El Sistema imprime el reporte	El Sistema imprime el reporte	

3. CPR 3: Exportar el Reporte del Balance de Comprobación de Saldos.

Descripción

El Sistema ofrece la posibilidad de exportar los reportes a formato.

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación y el contador selecciona en el menú de la izquierda la opción: “Contabilidad” y dentro de ella: “Reportes Contables”, luego “Reportes Externos” y por último la opción “Balance de Comprobación”.
- El Sistema muestra la interfaz: “Reporte Balance de Comprobación”.
- El usuario o contador introduce los datos a insertar.
- Presiona el botón “Visualizar”.
- El Sistema muestra el reporte.
- El usuario presiona el botón “Imprimir”.
- El Sistema visualiza la interfaz “Reporte”.

- El usuario presiona el botón correspondiente a exportar el reporte y selecciona el formato deseado.

Condiciones de Ejecución

- El Sistema debe tener cuentas contables definidas y el botón “exportar” debe existir en la pantalla.

Iteraciones

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Seleccionar la opción de exportar y escoger el formato deseado		El reporte se exporta al formato solicitado.	El Sistema imprime el reporte	

Anexo 2D

Diseño de Casos de Prueba
Elemento de Configuración: Sistema, Documentación
Nombre del Caso: Reporte Contable Estado de Ganancia o Pérdida

Registros y Notarías
Módulo Administración Contable
Versión

Registro de versiones:

Fecha	Versión	Descripción	Autor
20/04/07	<1. 0>	Caso de Prueba 4	Nemury Silega

Descripción General

Este caso de Prueba consiste probar uso consiste en realizar un Reporte Contable donde se registren los ingresos y los gastos cuya diferencia resulta ser ganancia o pérdida en el período y en el acumulado del Ejercicio Fiscal. Partiendo de la información que van generando los comprobantes contables, en este caso de uso se contempla la posibilidad de visualizar e imprimir el reporte.

Pruebas a realizar

- Pruebas funcionales.
- Prueba de interfaz de usuario.

Funcionalidades a probar.

- Mostrar reporte de tipos de ingresos.
- Imprimir reporte.
- Exportar reporte.
-

1. CPR 1: Mostrar reportes correspondientes a los intervalos de fechas introducidos para la búsqueda de tipos de ingresos.

En la interfaz se muestran los campos de los intervalos de fecha y el nivel de estado, aquí se seleccionarán por estos aspectos los reportes de pérdidas y ganancias relacionados a los mismos.

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación y el contador selecciona en el menú de la izquierda la opción: "Contabilidad", dentro de ella: "Reportes Contables", luego "Reportes Externos" y por último la opción: "Estado de Ganancias o Pérdida".
- El Sistema muestra la interfaz: "Reporte de Estado de Ganancias o Pérdidas".
- El usuario selecciona Nivel de Cuenta y el intervalo de fecha Desde-Hasta a buscar y presiona el botón "Visualizar".
- El Sistema muestra los reportes correspondientes.

Condiciones de Ejecución

- La fecha y el nivel de cuenta deben estar configurados.

Iterecaciones

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Nivel Cuenta: CUENTA Desde: 01/04/2007 Hasta: 10/04/2007		Se muestra el reporte	Se muestra el reporte	
Nivel Cuenta: CUENTA Desde: 01/04/2007 Hasta: 01/04/2007		Se muestra el reporte	Se muestra el reporte	
	Nivel Cuenta: mal Desde: 01/04/2007 Hasta: 10/04/2007	No es permitido por el Sistema la entrada de otros caracteres	No es permitido por el Sistema la entrada de otros caracteres	
	Nivel Cuenta: ANALISIS Desde: 10/04/2006 Hasta: 20/04/2005	El campo desde no puede ser mayor al hasta .	El campo desde no puede ser mayor al hasta .	

2. CPR 2: Imprimir Reporte Contable de Estado de Ganancias o Pérdidas.

Descripción

Después de haber obtenido el reporte se brinda la posibilidad de imprimirlo.

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación y el contador selecciona en el menú de la izquierda la opción: "Contabilidad", dentro de ella: "Reportes Contables", luego "Reportes Externos" y por último la opción: "Estado de Ganancias o Pérdida".
- El Sistema muestra la interfaz: "Reporte de Estado de Ganancias o Pérdidas".
- El usuario o contador selecciona Nivel de Cuenta y el intervalo de fecha Desde Hasta a buscar y presiona el botón "Visualizar".

- El Sistema muestra los reportes correspondientes.
- El usuario presiona el botón "Imprimir".
- El Sistema visualiza la interfaz "Reporte".
- El usuario presiona el botón correspondiente a imprimir reporte.

Condiciones de Ejecución

- Debe existir un reporte contable de Estado de Ganancias o Pérdidas.

Iteraciones

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Seleccionar la opción imprimir.		Se imprime el reporte y el Sistema notifica que la operación ha sido realizada correctamente	Se imprime el reporte y el Sistema notifica que la operación ha sido realizada correctamente	

4. CPR 3: Exportar el Reporte Contable de Estado de Ganancias o Pérdidas.

Descripción

El Sistema ofrece la posibilidad de exportar los reportes a formato PDF, DOC, XLS.

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación y el contador selecciona en el menú de la izquierda la opción: "Contabilidad", dentro de ella: "Reportes Contables", luego "Reportes Externos" y por último la opción: "Estado de Ganancias o Pérdida".
- El Sistema muestra la interfaz: "Reporte de Estado de Ganancias o Pérdidas".
- El usuario o contador selecciona Nivel de Cuenta y el intervalo de fecha Desde Hasta a buscar y presiona el botón "Visualizar".
- El Sistema muestra los reportes correspondientes.
- El usuario presiona el botón "Imprimir".
- El Sistema visualiza la interfaz "Reporte".

- El usuario presiona el botón correspondiente a exportar el reporte y selecciona el formato deseado.

Condiciones de Ejecución

- Debe existir un reporte contable de Estado de Ganancias o Pérdidas.

Iteraciones

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba
Se selecciona la opción de importar y se escoge el formato deseado		Se exporta el reporte al formato especificado.	Se exporta el reporte al formato especificado.

Anexo 2E

Diseño de Casos de Prueba
Elemento de Configuración: Sistema, Documentación
Nombre del Caso: Reporte de Egresos de Caja

Registros y Notarías
Módulo Administración Contable
Versión

Registro de versiones:

Fecha	Versión	Descripción	Autor
20/04/07	<1. 0>	Caso de Prueba 5	Nemury Silega

Descripción General

Este caso de uso consiste en diseñar un reporte donde se registren todas las operaciones de egresos de caja que se realizaron en un intervalo de tiempo. Dicho reporte puede ser visualizado e impreso.

Pruebas a Realizar

- Prueba de funcionalidad.
- Prueba de interfaz de usuario.

Funcionalidades

- Visualizar egresos.
- Imprimir importe.
- Exportar reporte.

1. CPR 1: Visualizar Egresos.

Descripción

El Sistema permite visualizar los reportes de documentos de egreso.

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación se selecciona “Finanzas” luego “caja” y dentro de esta “Reporte egreso caja”
- El Sistema muestra la interfaz “Reporte de egreso de caja”
- Se selecciona un intervalo de fecha.
- Se presiona el botón “visualizar”
- Se selecciona el intervalo de fecha correspondiente:
 - Desde.
 - Hasta.
- El usuario presiona el botón “Visualizar” en la parte inferior aparecen los datos.
- En la parte inferior se muestran los egresos realizados.

Condiciones de Ejecución

- Deben estar creados los documentos de Egresos

Iteraciones.

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba
Desde: 01/04/2007 Hasta: 10/04/2007		Se muestran los egresos realizados.	Se muestran los egresos realizados
	Desde: 10/04/2007 Hasta: 01/04/2007	El Sistema no permite que el campo Desde sea mayor que el campo Hasta .	El Sistema no permite que el campo Desde sea mayor que el campo Hasta
	Desde: 01/01/2008 Hasta: 05/01/2008	El Sistema no permite la entrada de una fecha posterior a la de sincronización.	El Sistema no notifica nada

2. CPR 2: Imprimir reporte.

Descripción

Imprimir reporte seleccionado

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación se selecciona "Finanzas" luego "caja" y dentro de esta "Reporte egreso caja"
- El Sistema muestra la interfaz "Reporte de egreso de caja"
- Se selecciona un intervalo de fecha.
- Se presiona el botón "visualizar"
- Se selecciona el nivel de cuanta y el intervalo de fecha correspondiente.
- El usuario presiona el botón "Visualizar" en la parte inferior aparecen los datos.
- En la parte inferior se muestran los egresos realizados.
- Se selecciona el reporte dando clic sobre el.
- Se presiona el botón "imprimir"

Condiciones de Ejecución

- El usuario debe haber seleccionado al menos uno de los reportes mostrados por el caso de uso anterior, y haber dado Clic en el botón Imprimir.

Iteraciones

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba
Selecciona la opción de imprimir el reporte		el Sistema imprime el reporte buscado	el Sistema imprime el reporte buscado

3. CPR 3: Exportar reporte.

Descripción

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación se selecciona “Finanzas” luego “caja” y dentro de esta “Reporte egreso caja”
- El Sistema muestra la interfaz “Reporte de egreso de caja”
- Se selecciona un intervalo de fecha.
- Se presiona el botón “visualizar”
- Se selecciona el nivel de cuanta y el intervalo de fecha correspondiente.
- El usuario presiona el botón “Visualizar” en la parte inferior aparecen los datos.
- En la parte inferior se muestran los egresos realizados.
- Se selecciona el reporte dando clic sobre el.
- Se presiona el botón “imprimir”.
- El Sistema visualiza la interfaz “Reporte”.
- El usuario presiona el botón correspondiente a exportar el reporte y selecciona el formato deseado.

Condiciones de Ejecución

- Debe existir un reporte visualizado.

Iteraciones

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Seleccionar la opción de exportar		El Sistema exporta el reporte al formato deseado.	El Sistema exporta el reporte al formato deseado.	

Registro de defectos y dificultades detectados

Elemento	N^o	Descripción de la No conformidad	Aspecto correspondiente	Etapas de la detección	Importancia	Recomendaciones
Sistema	1	El Sistema no debería permitir una fecha superior a la de sincronización.	CPR1	Ejecución de las Pruebas	Media	Revisar para todos los casos

Anexo 2F

Diseño de Casos de Prueba
Elemento de Configuración: Sistema, Documentación
Nombre del Caso: Reporte de Egresos de Caja

Registros y Notarías
Módulo Administración Contable
Versión

Registro de versiones:

Fecha	Versión	Descripción	Autor
20/04/07	<1. 0>	Caso de Prueba 6	Nemury Silega

Descripción General

Este caso de Prueba consiste probar que se fijen las obligaciones de pago por factura recibida a los proveedores. Teniendo en cuenta los importes por concepto de retenciones a entidades recaudadoras.

Pruebas a realizar

- Pruebas Funcionales.
- Pruebas de interfaz de usuario.

Funcionalidades a probar

- Guardar una factura.
- Confirmar una factura.
- Cancelar una factura.
- Seleccionar una factura en estado de liquidación

1. CPR 1: Guardar una Factura.

Descripción

Este caso de Prueba brinda la posibilidad de guardar una factura.

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación y se selecciona en el menú de la izquierda la opción: "Finanzas", dentro de ella: "Obligaciones por Pagar" y por último la opción: "Gestionar Factura".
- El Sistema muestra la interfaz: "Gestionar Factura".
- El usuario selecciona un proveedor en la región correspondiente a "Buscar Proveedor" y posteriormente presiona el botón "Nueva" para crear una nueva factura.
- El Sistema muestra otra interfaz propia para insertar los datos de la factura:
 - No. de Factura.
 - No. de Control.

- Descripción del movimiento.
 - Sin derecho a crédito Fiscal.
 - Fecha de emisión.
 - Fecha de vencimiento.
 - Código Cuenta o Descripción de la Cuenta.
 - Acción a realizar
- El actor edita o selecciona los datos.
 - El actor ordena visualizar la interfaz donde se adiciona un concepto de gasto.
 - El Sistema muestra la interfaz para editar o seleccionar los siguientes datos:
 - Concepto de Contrapartida.
 - Importe.
 - El actor ordena adicionar el concepto de gasto.
 - El Sistema adiciona el concepto de gasto.
 - El actor ordena guardar la factura
 - El Sistema muestra un mensaje informando que la operación Guardar se ha ejecutado con éxito.

Condiciones de Ejecución

- El Sistema debe tener proveedores registrados.
- El Sistema debe tener cuentas en el Plan único de cuenta.

Iteraciones

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba
No. de Factura: "15" No. De control:"1" Fecha de Emisión: "10/04/2007" Fecha de Vencimiento: "10/04/2007" Descripción del movimiento:"Movimiento1" Concepto de Gasto: SERVICIOS Importe: "200000" Cuenta:"111111"		El Sistema guarda la factura satisfactoriamente.	El Sistema guarda la factura satisfactoriamente.
	No. de Factura: "35" No. De control:"1" Fecha de Emisión: "11/04/2007" Fecha de Vencimiento: "10/04/2007" Descripción del movimiento:"Movimiento1" Concepto de Gasto: SERVICIOS Importe: "200000" Cuenta:"111111"	El Sistema no permite que el campo Fecha de vencimiento sea menor que el campo Fecha de emisión.	El Sistema no permite que el campo Fecha de vencimiento sea menor que el campo Fecha de emisión.
	No. de Factura: "25" No. De control:"1" Fecha de Emisión: "10/04/2007" Fecha de Vencimiento: "10/04/2007" Descripción del movimiento:"Movimiento1" Concepto de Gasto: SERVICIOS Importe: "un millón" Cuenta:"111111"	El Sistema no permite que se introduzcan letras en el campo importe.	El Sistema no permite que el campo Fecha de vencimiento sea menor que el campo Fecha de emisión.
	No. de Factura: " " No. De control:"1" Fecha de Emisión: "10/04/2007" Fecha de Vencimiento: "10/04/2007" Descripción del movimiento:"Movimiento1" Concepto de Gasto: SERVICIOS Importe: "200000" Cuenta:"111111"	El Sistema no debe permitir el número de la factura como un campo vacío.	El Sistema no permite que el campo Fecha de vencimiento sea menor que el campo Fecha de emisión.

2. CPR 2: Confirmar una Factura.

Descripción

Este caso de Prueba brinda la posibilidad de confirmar una factura.

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación y el contador selecciona en el menú de la izquierda la opción: “Finanzas”, dentro de ella: “Obligaciones por Pagar” y por último la opción: “Gestionar Factura”.
- El Sistema muestra la interfaz: “Gestionar Factura”.
- El usuario o contador selecciona un proveedor en la región correspondiente a “Buscar Proveedor” y posteriormente presiona el botón “Nueva” para crear una nueva factura.
- El Sistema muestra otra interfaz propia para insertar los datos de la factura:
 - No. de Factura.
 - No. de Control.
 - Descripción del movimiento.
 - Sin derecho a crédito Fiscal.
 - Fecha de emisión.
 - Fecha de vencimiento.
 - Código Cuenta o Descripción de la Cuenta.
 - Acción a realizar
- El actor edita o selecciona los datos.
- El actor ordena visualizar la interfaz donde se adiciona un concepto de gasto.
- El Sistema muestra la interfaz para editar o seleccionar los siguientes datos:
 - Concepto de Contrapartida.
 - Importe.
- El usuario presiona el botón “Aceptar”.
- El Sistema muestra un mensaje notificando que la Factura se ha confirmado satisfactoriamente.

Condiciones de Ejecución

- El Sistema debe tener proveedores registrados.
- El Sistema debe tener cuentas en el Plan único de cuenta.
- El Sistema debe poseer escáner instalado.

Nota: Se realiza la misma Prueba anterior pero especificando la Acción como de Confirmar.

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones

3. CPR 3: Cancelar una Factura. Descripción

Se comprueba que se pueda cancelar una factura.

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación y el contador selecciona en el menú de la izquierda la opción: "Finanzas", dentro de ella: "Obligaciones por Pagar" y por último la opción: "Gestionar Factura".
- El Sistema muestra la interfaz: "Gestionar Factura".
- El Sistema muestra la interfaz para editar o seleccionar los siguientes datos.
 - No. de Factura.
 - No. de Control.
 - Descripción del movimiento.
 - Sin derecho a crédito Fiscal.
 - Fecha de emisión.
 - Fecha de vencimiento.
 - Código Cuenta o Descripción de la Cuenta.
 - Acción a realizar.
- El actor edita o selecciona los datos.
- El actor ordena visualizar la interfaz donde se adiciona un concepto de gasto.
- El Sistema muestra la interfaz para editar o seleccionar los siguientes datos:

- Concepto de Contrapartida.
- Importe.
- El actor edita o selecciona los datos.
- El actor ordena adicionar el concepto de gasto.
- El Sistema adiciona el concepto de gasto.
- El actor ordena aceptar la operación.
- El actor ordena guardar la factura.
- El actor ordena cancelar la operación.
- El Sistema cancela la operación.

Condiciones de Ejecución

El Sistema debe tener proveedores registrados.

El Sistema debe tener cuentas en el Plan único de cuenta.

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Ordenar cancelar la operación		El Sistema cancela la factura satisfactoriamente.	El Sistema cancela la factura satisfactoriamente.	

5. CPR 4: Anular una factura.

Descripción

Este caso de Prueba brinda la posibilidad de anular una factura.

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación y el contador selecciona en el menú de la izquierda la opción: "Finanzas", dentro de ella: "Obligaciones por Pagar" y por último la opción: "Gestionar Factura".
- El Sistema muestra la interfaz: "Gestionar Factura".
- El actor selecciona el proveedor.
- El actor selecciona la fecha en la que desea hacer el filtro de búsqueda.

- El Sistema muestra las facturas correspondientes al proveedor y la fecha.
- El actor selecciona una factura en estado de confirmación.
- El actor ordena abrir la factura.
- El Sistema muestra una interfaz con los datos de la factura seleccionada que no pueden ser modificados.
- El actor ordena anular la factura.
- El Sistema genera un comprobante contable y actualiza las obligaciones.

Condiciones de Ejecución

- El Sistema debe tener proveedores registrados.
- El Sistema debe tener cuentas en el Plan único de cuenta.

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Ordenar anular la factura		El Sistema anula la factura correctamente.	El Sistema anula la factura correctamente.	

4. CPR 4: Seleccionar una factura en estado de liquidación.

Descripción

Este caso de Prueba se comprueba si se puede seleccionar una factura en estado de liquidación.

Flujo Central

- El Sistema muestra la interfaz principal de la aplicación y el contador selecciona en el menú de la izquierda la opción: "Finanzas", dentro de ella: "Obligaciones por Pagar" y por último la opción: "Gestionar Factura".
- El Sistema muestra la interfaz: "Gestionar Factura".
- El actor selecciona el proveedor.
- El actor selecciona la fecha en la que desea hacer el filtro de búsqueda.
- El Sistema muestra las facturas correspondientes al proveedor y la fecha.

- El actor selecciona una factura en estado de liquidación.
- El actor ordena abrir la factura.
- El Sistema muestra una interfaz con los datos de la factura esta no puede sufrir ningún cambio.

Condiciones de Ejecución

- El Sistema debe tener proveedores registrados.
- El Sistema debe tener cuentas en el Plan único de cuenta.

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Ordenar mostrar factura en estado de liquidación		El Sistema muestra la factura en liquidación	El Sistema muestra la factura en liquidación	

Anexo 2 G

Diseño de Casos de Prueba
Elemento de Configuración: Sistema, Documentación
Nombre del Caso: Registrar Cuenta Bancaria

Registros y Notarías
Módulo Administración Contable
Versión

Registro de versiones:

Fecha	Versión	Descripción	Autor
20/04/07	<1. 0>	Caso de Prueba 7	Nemury Silega

Descripción General

Este caso de uso consiste en definir las cuentas bancarias que se utilizarán en el Sistema, mostrando además la conciliación de las mismas.

Pruebas a realizar.

- Pruebas funcionales.
- Pruebas de interfaz de usuario.

Funcionalidades a probar

- Configurar una cuenta bancaria.
- Conciliar cuenta.
- Eliminar cuenta.

1. CPR 1: Configurar una cuenta bancaria

Descripción

El Sistema muestra las cuentas existentes y permite adicionar nuevas cuentas.

Flujo Central

- El actor inicia el caso de uso Cuenta Bancaria en el menú principal de la aplicación Finanzas, Banco, Cuenta Bancaria.
- El Sistema muestra una interfaz con las Cuentas Bancarias creadas hasta el momento.
- El actor ordena crear una Cuenta Bancaria..
- El Sistema muestra en la interfaz los siguientes datos que serán editados o seleccionados.
 - Número
 - Nombre
 - Cuenta Contable o Descripción Cuenta Contable.
 - Saldo Inicial en Banco.
- El actor edita o selecciona los datos.

- El actor ordena aceptar la operación.
- El Sistema acepta la operación
- El Sistema actualiza la base de datos.
- Se presiona el botón “aplicar”
- Se crea la nueva cuenta.
- Se muestra en la parte inferior.
- Se presiona “cerrar” para salir.

Condiciones de Ejecución

- Solo se podrá asociar a una cuenta contable una Cuenta Bancaria.
- En el Plan único de cuentas deben estar definidas las cuentas.
- Solo se podrán eliminar las Cuentas Bancarias que no estén asociadas a ningún documento registrado por el Sistema.
- Deben estar configuradas las reglas contables.

Iteraciones.

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Número de la cuenta bancaria: "5555-5555-55-55555555" Nombre de la cuenta bancaria: "Pruebas" Cuenta contable asociada: "11205" Saldo en Banco Apertura: "5000000"		Se crea la nueva cuenta bancaria.	Se muestra un error informando que no se puede realizar la operación.	
Número de la cuenta bancaria: "nombre" Nombre de la cuenta bancaria: "Pruebas SAREN" Cuenta contable asociada: "45864789" Saldo en Banco Apertura: "5000000"		El Sistema no permite la entrada de letras en el número de cuenta.	El Sistema no permite la entrada de letras en el número de cuenta.	
Número de la cuenta bancaria: "1111-2222-33-44444444457862134" Nombre de la cuenta bancaria: "Pruebas SAREN" Cuenta contable asociada: "45864789" Saldo en Banco Apertura: "5000000"		El Sistema no permite más de 20 números en el campo de número de cuenta bancaria.	El Sistema no permite más de 20 números en el campo de número de cuenta bancaria.	

	<p>Número de la cuenta bancaria: "1111-2222-33-4444444444" Nombre de la cuenta bancaria: "Pruebas SAREN" Cuenta contable asociada: "45864789" Saldo en Banco Apertura: "mil "</p>	<p>El Sistema no permite la entrada de letras en el campo correspondiente al saldo en banco apertura.</p>	<p>El Sistema no permite la entrada de letras en el campo correspondiente al saldo en banco apertura.</p>	
--	--	---	---	--

2. CPR 2: Conciliar cuenta

Descripción

Se realizara una comparación entre el saldo bancario y el saldo del libro.

Flujo Central

- El actor inicia el caso de uso Cuenta Bancaria en el menú principal de la aplicación Finanzas, Banco, Cuenta Bancaria.
- El Sistema muestra una interfaz con las Cuentas Bancarias creadas hasta el momento.
- El actor ordena conciliar la Cuenta Bancaria seleccionada.
- El Sistema concilia la Cuenta Bancaria.
- El Sistema muestra una interfaz con un resumen de la Conciliación hecha a la Cuenta Bancaria seleccionada.
- El actor ordena visualizar la conciliación.
- El Sistema visualiza la conciliación.

Condiciones de Ejecución

- Para que la cuenta bancaria sea conciliada es necesario que esta esté salvada en la base de datos.

Iteraciones

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba
Ordenar conciliar		El Sistema muestra el resultado de la conciliación bancaria.	El Sistema muestra el resultado de la conciliación bancaria.

3. CPR 3 Eliminar Cuenta bancaria.

Descripción

Imprimir reporte seleccionado

Flujo Central

- El actor inicia el caso de uso Cuenta Bancaria en el menú principal de la aplicación Finanzas, Banco, Cuenta Bancaria.
- El Sistema muestra una interfaz con las Cuentas Bancarias creadas hasta el momento.
- El actor selecciona la Cuenta Bancaria.
- El actor ordena eliminar la cuenta seleccionada
- El Sistema elimina la Cuenta Bancaria.

Condiciones de Ejecución

- Debe existir la cuenta y sólo se elimina si no tiene operaciones asociadas.

Iteraciones

Clases válidas	Clases inválidas	Resultado Esperado	Resultado de la Prueba
Seleccionar una cuenta bancaria que tenga alguna operación asociada y ordenar eliminar		El Sistema notifica que no se puede eliminar porque tiene alguna operación asociada	El Sistema notifica que no se puede eliminar porque tiene alguna operación asociada
Ordenar eliminar la cuenta bancaria		Se elimina la cuenta bancaria	Se elimina la cuenta bancaria

Registro de defectos y dificultades detectados

Elemento	Nº	Descripción de la No conformidad	Aspecto correspondiente	Etapas de la detección	Importancia	Recomendaciones
Sistema	1	Cuando se ordena aceptar la nueva cuenta aparece un error.	CPR1	Ejecución de las Pruebas	Alta	Revisar los roles y permisos.