

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

**TÍTULO: Sistema de Gestión de Datos Geológicos. Módulo Registro
Minero. Rol Diseñador de Casos de Prueba**

AUTOR: Yisel Sánchez Arias.

TUTOR: Ing. Lisdaynet Gómez Espinosa.
CO-TUTOR: Ing. Daniel Burgos Hechavarría

Ciudad de la Habana, jueves, 24 de junio de 2010.
Año 52 de la Revolución

DEDICATORIA

A mis padres Reina y Carlos, por ser esos padres maravillosos que me lo han dado todo, por confiar siempre en mí y por hacerme la persona que soy hoy.

A mis dos hermanas y amigas: Yamilet por quererme tanto y estar siempre al pendiente de mí y Yenisel que a pesar de la distancia siempre me ha ayudado y querido mucho.

A mis sobrinos, que son mis niños bellos: Camila, Israel, José Carlos e Ismaelito de quienes aspiro ser guía y ejemplo y a los cuales amo mucho.

A toda mi familia que tanto apoyo me han dado durante toda la carrera para poder lograr mi sueño de graduarme a pesar de la distancia.

AGRADECIMIENTOS

En primer lugar a mis padres y a mis hermanas por siempre estar presente cuando más los he necesitado en mi vida, por ayudarme tanto y apoyarme en todas mis decisiones.

A mi tía Rosario que es como mi segunda madre, por abrirme las puertas de su casa, por haberme educado tanto en mis años de estudio, desde 5to grado hasta hoy, por darme amor y afecto, gracias a ella estoy hoy aquí.

A Elizabeth que ha sido para mí una de las personas más maravillosas que tengo en mi vida, siempre apoyándome, preocupándose siempre por mí, dándome consejos, terapias, psicologías y mucho amor. Le estaré eternamente agradecida, la quiero mucho.

A mi novio por estar conmigo en todo momento y por toda su ayuda en el trabajo, muchas gracias de corazón.

A todos mis primos, especialmente a Yani por ser otra hermana más, siempre contándonos todo y muy preocupada por mis cosas, también a mi primo Yovi, que tantas jabas de comida ha tenido que cargar para mí en estos años de universidad y que también se ha preocupado y ayudado en cuanto a mis estudios.

A toda mi familia en general, tanto de Bayamo, Veguita y Cayo Grande, porque todos me han brindado su amor incondicional.

A mi tutora, cotutor, tribunal y oponente, por toda su ayuda, que contribuyeron con su granito de arena a que todo saliera bien.

A todos mis amigos, amigas y compañeros de aula, desde la infancia hasta la universidad, a todos muchas gracias por estar siempre pendiente de mí, por quererme y soportarme.

A todas las personas que participaron en mi formación personal y profesional.

Muchas gracias Yisel.

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la facultad 9 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Yisel Sánchez Arias

Lisdaynet Gómez Espinosa

DATOS DE CONTACTO

Síntesis del Tutor

Nombre: Lisdaynet Gómez Espinosa.

Profesión: Ingeniero en Ciencias Informáticas.

Categoría docente: Adiestrado.

Año de graduado: 2008.

Correo: lespinosa@uci.cu

Breve descripción: Graduado en la Universidad de las Ciencias Informáticas, donde actualmente se desempeña en la facultad 9 como profesora del departamento de Sistemas Digitales. Imparte Sistema Operativo y Seguridad Informática, siendo la jefa de asignatura de ambas. Se desempeña en el departamento de Geoinformática perteneciente al centro GEySED como Jefa de Formación, Mercadotecnia y Negocio.

Síntesis del Co-tutor

Nombre: Daniel Burgos Hechavarría.

Profesión: Ingeniero Informático.

Categoría docente: Instructor.

Año de graduado: 2006.

Correo: dburgos@uci.cu

Breve descripción: Graduado en la Universidad de Ciego de Ávila, actualmente se desempeña en la facultad 9 como profesor del departamento de Ciencias Básicas. Imparte Investigación de Operaciones y Probabilidades y Estadísticas, siendo el jefe de asignatura de esta última.

Beneficios del proceso de diseño y ejecución de las pruebas Módulo: Registro Minero

El diseño y ejecución de las pruebas al módulo Registro Minero del Sistema de Gestión de Datos Geológicos (en lo adelante SGD) concluyó de forma satisfactoria, significando un paso importante en el fortalecimiento de su calidad como producto informático.

El diseño de las pruebas constituyó una herramienta importante para la detección de los errores que muy probablemente estarían presentes en el software. Se diseñaron 17 Casos de Prueba en correspondencia con los 17 Casos de Uso del módulo en cuestión los cuales fueron validados por los implementadores y corregidos y revisados según las listas de chequeo previstas para esto. Cada diseño fue realizado con buena calidad y seriedad, siendo fieles a las descripciones textuales de cada escenario de Caso de Uso.

Un segundo momento de la etapa de pruebas fue la ejecución de los diseños, el momento de la prueba en sí misma. Se probaron todos los diseños realizados, constituyendo el centro y eje principal del proceso, se detectaron un conjunto de No Conformidades (defectos) las cuales fueron devueltas a los implementadores para su pronta corrección, esto permitió una entrega más profesional con una posibilidad de error mucho mayor que la obtenida antes de comenzar las pruebas.

De forma general, el proceso de pruebas concluyó satisfactoriamente al poder detectar una cantidad significativa de defectos que fueron eliminados en un tiempo breve, acercando la solución informática a un software con alta calidad y rigor técnico.

2do jefe del Dpto. Geoinformática: Ing. Vladimir Martell Fernández.

OPINIÓN DEL TUTOR

Como tutora de esta tesis, opino que la autora se desempeñó durante el desarrollo de este trabajo de una manera ardua, profunda y con un alto grado de sacrificio. Marcando una alta independencia, responsabilidad y laboriosidad.

El trabajo desarrollado consistió, principalmente, en el diseño y la implementación de las pruebas al módulo Registro Minero, que forma parte del Sistema de Gestión de Datos Geológicos, para garantizar que el mismo se entregue libre de defectos.

Realizando un correcto trabajo investigativo que demuestra las habilidades desarrolladas con el trabajo de diploma, y que considero como muy importantes para el éxito del ejercicio integrador con que culminan sus estudios de la carrera.

Los artefactos que forman parte de su solución presentan una alta calidad científico-técnica en correspondencia con el nivel del rigor exigido. El documento está bien redactado y con buena ortografía, siguiendo un flujo de exposición del contenido que permite el buen entendimiento.

Considero que la diplomante ha defendido su trabajo de manera correcta en las distintas exposiciones que ha realizado, demostrando confianza y dominio en el tema y cumpliendo de los objetivos propuestos.

Por tanto, considero que la estudiante está apta para ejercer como Ingeniera en Ciencias Informáticas y propongo se le otorgue al trabajo de diploma la calificación de ___ puntos.

Ing. Lisdaynet Gómez Espinosa

Firma

1 de julio de 2010

Fecha

RESUMEN

Las pruebas de software son los procesos que permiten verificar y revelar la calidad de un producto software. Hoy día es crucial comprobar y evaluar la calidad de lo construido con el objetivo de minimizar el costo de su reparación. Mientras más rápido se detecte una falla, más barata es su corrección. En el centro productivo GEySED de la facultad 9 de la Universidad de las Ciencias Informáticas se desarrollan varios productos informáticos como es el Sistema de Gestión de Datos Geológicos. El módulo Registro Minero, forma parte de este sistema, el cual está en la etapa final de su fase de implementación por lo que para desplegarlo se hace necesario verificar si el mismo funciona correctamente.

El objetivo principal que persiguió este trabajo fue el elaborar todos los artefactos correspondientes al rol diseñador de casos de pruebas que define la metodología de Proceso Unificado de Desarrollo para realizarle las pruebas al módulo Registro Minero y así garantizar la calidad del producto desarrollado. Al encontrar los defectos en el software los desarrolladores pueden corregirlos a tiempo. Como resultado de la investigación, se diseñaron y aplicaron 27 casos de prueba al módulo. Además se realizó un Plan de Prueba el cual describe las estrategias, los recursos y la planificación para la realización de los casos de prueba.

Palabras Claves: caja blanca, caja negra, casos de prueba, pruebas, registro minero

Summary

Software testing is the process that allows checking and showing the quality of a software product. Today is critical to test and evaluate quality of products aiming at minimizing the cost of repairs. The sooner a failure is detected, the cheaper its correction. In the production center GEySED from School 9 in the University of Informatics Sciences are developed many software products such as the Management System of Geological Data. The Mining Register module is part of this system, which is in the final stage of its implementation phase and it is needed to check if it works correctly to display it.

The main objective of this research was to develop all the artifacts for the role designer of testing cases that defines the methodology of Rational Unified Process to carry out tests to the Mining Register module thus ensuring the quality of the product developed. When finding flaws in the software, the developers can correct it on time. As a result of the investigation, 27 testing cases were designed and applied to the module. It was also made a Testing Plan describing strategies, resources, and planning to carry out the testing cases.

TABLAS Y FIGURAS

Índice de Tablas

Tabla 1: Diseño de caja negra del Caso de Uso: Gestionar Medida.....	24
Tabla 2: Diseño de caja negra del Caso de Uso: Gestionar Compatibilización.....	26
Tabla 3: Diseño de caja negra del Caso de Uso: Gestionar Obligación.....	29
Tabla 4: Resultados al aplicar el caso de prueba: Gestionar Medida. SC2: Buscar Medidas	¡Error! Marcador no definido.
Tabla 5: Resultados al aplicar el caso de prueba: Gestionar Medida. SC3: Insertar Medidas	42
Tabla 6: Resultados al aplicar el caso de prueba: Gestionar Medida. SC4: Modificar Medidas	43
Tabla 7: Resultados al aplicar el caso de prueba: Gestionar Medida. SC5: Eliminar Medidas	44
Tabla 8: Resultados al aplicar el caso de prueba: Gestionar Compatibilizaciones SC2: Buscar Compatibilizaciones.	45
Tabla 9: Resultados al aplicar el caso de prueba: Gestionar Compatibilizaciones SC3: Insertar Compatibilizaciones.	46
Tabla 10: Resultados al aplicar el caso de prueba: Gestionar Compatibilizaciones SC4: Modificar Compatibilizaciones.....	47
Tabla 11: Resultados al aplicar el caso de prueba: Gestionar Compatibilizaciones SC5: Eliminar Compatibilizaciones.	48
Tabla 12: Resultados al aplicar el caso de prueba: Gestionar Obligaciones SC2: Buscar Obligaciones.....	48
Tabla 13: Resultados al aplicar el caso de prueba: Gestionar Obligaciones SC3: Insertar Obligaciones	50
Tabla 14: Resultados al aplicar el caso de prueba: Gestionar Obligaciones SC4: Modificar Obligaciones.....	51
Tabla 15: Resultados al aplicar el caso de prueba: Gestionar Obligaciones SC5: Eliminar Obligaciones.	52

Índice de Figuras

Figura 1: Fases y Flujos de Trabajo de RUP.....	11
Figura 2: Prueba de caja blanca.....	18
Figura 3: Prueba de caja negra	19
Figura 4: Grafo del caso de prueba de caja blanca: Gestionar Medidas.....	33
Figura 5: Grafo del caso de prueba de caja blanca: Gestionar Compatibilizaciones.....	37
Figura 6: Grafo del caso de prueba de caja blanca: Gestionar Obligaciones.....	39

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: Fundamentación Teórica.....	6
1.1 Introducción	6
1.2 Conceptos asociados al dominio del problema.....	6
1.2.1 Registro Minero.....	6
1.2.2 Rol	6
1.2.3 Prueba	7
1.2.4 Prueba de Software	7
1.2.5 Calidad.....	7
1.2.6 Calidad del software.....	8
1.3 Objeto de Estudio	8
1.3.1 Situación Problemática	8
1.4 Metodologías de desarrollo de software	9
1.4.1 Proceso Unificado de Desarrollo (RUP)	10
1.5 Proceso de pruebas de RUP.	13
1.5.1 Artefactos del flujo de trabajo de Prueba.....	14
1.5.2 Actividades del flujo de trabajo de Prueba	15
1.5.3 Tipos de Pruebas de Software	17
1.5.4 Pruebas de caja blanca.....	18
1.5.5 Pruebas de caja negra	19
1.6 Conclusiones Parciales.....	20
CAPÍTULO 2: Diseño y Aplicación de las pruebas	21
2.1 Introducción	21
2.2 Características a probar	21
2.3 Plan General de Pruebas	21
2.4 Objetivos	22
2.5 Estrategia de Pruebas	22
2.6 Diseño de casos de prueba de caja negra	23
2.6.1 Caso de prueba de caja negra: Gestionar Medidas.	23
2.6.2 Caso de prueba de caja negra: Gestionar Compatibilizaciones.	26
2.6.3 Caso de prueba de caja negra: Gestionar Obligaciones.	29
2.7 Diseño de casos de prueba de caja blanca.....	31

2.7.1	Caso de prueba de caja blanca: Gestionar Medidas	31
2.7.2	Caso de prueba de caja blanca: Gestionar Compatibilizaciones.....	34
2.7.3	Caso de prueba de caja blanca: Gestionar Obligaciones	38
2.8	Conclusiones parciales	40
	CAPÍTULO 3: Resultados Obtenidos	41
3.1	Introducción	41
3.2	Resultados al aplicar los casos de prueba al módulo Registro Minero.....	41
3.2.1	Caso de prueba Gestionar Medidas.....	41
3.2.2	Caso de prueba Gestionar Compatibilizaciones.....	45
3.2.3	Caso de prueba Gestionar Obligaciones.....	48
3.3	Resultados Generales:	53
3.4	Validación de expertos. Método Delphi.....	54
3.5	Conclusiones Parciales.....	57
	Conclusiones Generales.....	58
	Referencias bibliográficas.....	60
	Bibliografía consultada.....	62
	Glosario de términos	63
	Anexos	64

INTRODUCCIÓN

El desarrollo de software ha alcanzado en la actualidad niveles sorprendentes. Cada empresa o institución que se inicie en estos tiempos, encuentra un fuerte soporte en las tecnologías de la información y las comunicaciones; la consolidación de cualquier organización sin tener en su núcleo principal un sistema automatizado que ofrezca calidad y confianza en su desempeño, resulta inconcebible o con pocas esperanzas de prosperidad.

A nivel mundial, para controlar la calidad de un software, se realizan una serie de revisiones y pruebas durante el ciclo de desarrollo, para asegurar que cada producto cumpla con sus requisitos funcionales. El objetivo de la garantía de la calidad, es la gestión para informar sobre los datos necesarios del producto, pues se adquiere una visión más profunda y segura de que el producto está cumpliendo sus objetivos. Además, no se garantiza mediante las auditorías y controles sistemáticos que el resultado final del software sea satisfactorio.

Si no se garantiza una buena calidad del producto, este corre el riesgo de verse afectado por errores o fallas y podría colapsar en cuanto a su funcionamiento dicho sistema. Esta es una de las razones principales por la cual el proceso de pruebas es clave desde el mismo momento en que empieza el desarrollo de un software hasta que finaliza.

Una de las dificultades más arduas en la prueba, es saber cuándo terminar. No se ha demostrado una forma práctica para probar que cualquier pieza de software en el mundo esté exenta de errores, ni aunque haya pasado toda la etapa de pruebas.

Conceptos como la estabilidad, eficiencia, escalabilidad y seguridad están asociados a la calidad de un producto bien desarrollado, los cuales se garantizan con las pruebas realizadas al software, además de la robustez y fiabilidad con que pueda contar el producto una vez concluida la aplicación de dichas pruebas.

A pesar de la gran importancia que representan las pruebas de software, a nivel mundial no todas las empresas productoras las tienen en cuenta o le dan la importancia que merece. En España por ejemplo, los profesionales que desarrollan su actividad laboral entorno a las tecnologías informáticas, consideran que las pruebas de software,

muy valoradas por los expertos, son una tarea poco creativa, de escasa consideración dentro de las organizaciones y de poca proyección profesional. A estas afirmaciones se le debe sumar la falta de formación específica en pruebas de los profesionales dedicados a estas tareas y la poca importancia que las empresas dan a los técnicos dentro de los proyectos, al tratarse de un servicio que no siempre se puede facturar y del escaso calendario dedicado a las pruebas. [1]

Como parte de los esfuerzos de Cuba por ascender a planos superiores en la rama de la informática, han surgido diferentes entidades dedicadas al desarrollo de software. Las mismas deben garantizar la entrega de sus productos con la mayor calidad posible y para ello es sumamente importante no olvidar detalles ni etapas en el desarrollo de una aplicación. Una de estas etapas es la de pruebas, fundamental para aumentar la calidad de los productos y con ellos las posibilidades de insertarse entre las naciones reconocidas como exportadoras de software.

En Cuba se encuentra la Universidad de las Ciencias Informáticas (UCI), que es un Centro de Estudios Superiores dedicado a la formación de Ingenieros en Ciencias Informáticas y a la producción de proyectos informáticos. Tiene entre sus objetivos principales informatizar el país y desarrollar la industria del software para contribuir al desarrollo económico del mismo. Estratégicamente, la universidad está dividida en 9 facultades. Cada facultad cuenta con Centros Productivos que laboran sobre diferentes perfiles, estos son los encargados de desarrollar los productos sobre las temáticas relacionadas con sus perfiles.

El Centro Productivo Geoinformática y Señales Digitales (GEySED) de la facultad 9, cuenta con dos departamentos, Geoinformática y Señales Digitales. En el departamento de Geoinformática se desarrollan varios productos como el Sistema de Gestión de Datos Geológicos (SGDG), el cual está dividido en módulos. El módulo Registro Minero es de gran importancia para el país, pues en este sistema es donde se recoge toda la información referente al proceso de gestión y recuperación de la información de las solicitudes y de los derechos mineros. Este módulo tiene entre sus funciones, ordenar y fiscalizar la actividad geológica, minera y petrolera de la República de Cuba.

La realización de este sistema implica un gran compromiso por parte de cada uno de los miembros del equipo de desarrollo con el cliente, teniendo en cuenta que no se le debe

entregar una utilidad que contenga errores. Al entregar un producto inmaduro, trae como consecuencia una labor penosa y costosa de soporte, además de pérdida de imagen de la entidad, así como baja credibilidad de la misma en cuanto a la calidad de sus productos. Para poder competir en el mercado globalizado y altamente competitivo de las tecnologías de información y comunicación, la calidad representa un factor muy importante e indispensable. Por todo lo anteriormente expuesto, surge el **problema a resolver**: ¿Cómo lograr la entrega del módulo Registro Minero libre de defectos?

Es por esto que el **objeto de estudio** de la investigación es: El proceso de pruebas de sistemas de gestión de información; enmarcado en el **campo de acción**: Diseño e implementación de las pruebas del módulo Registro Minero que forma parte del Sistema de Gestión de Datos Geológicos.

Ya definidos los elementos anteriores de la investigación, se concreta el **objetivo general** de la investigación: Elaborar la documentación del proceso de pruebas correspondiente al módulo Registro Minero que forma parte del Sistema de Gestión de Datos Geológicos.

Se plantea como **idea a defender**: El diseño y la implementación de las pruebas al módulo Registro Minero, que forma parte del Sistema de Gestión de Datos Geológicos, garantizarán que el mismo se entregue libre de defectos.

Para el cumplimiento del objetivo general se trazaron las siguientes **tareas investigativas**:

- Caracterizar el proceso de pruebas según la metodología seleccionada.
- Caracterizar el Rol Diseñador de casos de prueba según la metodología seleccionada.
- Elaborar el Plan de Pruebas para el módulo Registro Minero.
- Diseñar los casos de prueba.
- Realizar las pruebas de caja blanca al módulo Registro Minero.
- Realizar las pruebas de caja negra al módulo Registro Minero.

- Caracterizar el estado de la implementación una vez finalizado el proceso de pruebas.
- Validar las pruebas realizadas.

Una vez concluida la investigación se esperan obtener los siguientes **resultados**: Evaluación y validación de la implementación del módulo Registro Minero a partir del desarrollo del proceso de pruebas.

Para darle cumplimiento a las tareas de la investigación se utilizan varios métodos científicos:

Métodos teóricos:

- **Método histórico-lógico:** Se utiliza para analizar los procesos de pruebas y la información referente a las pruebas, que se tienen hasta el momento, así como hacer un resumen de los aspectos más importantes a tener en cuenta para la investigación.
- **Método analítico-sintético:** Es el método utilizado en toda la investigación, pues se recoge información acerca del tema tratado, se hace un análisis de la bibliografía y se realiza síntesis de la misma.

Métodos Empíricos:

- **Observación:** Específicamente se realizará la observación al código, para el período de realización de las pruebas de caja blanca.
- **Encuestas:** Se realizarán las encuestas una vez realizada la validación de las pruebas mediante el método experto.

Población: está representada por todos los integrantes del proyecto para un total de 46 miembros.

Muestra: será seleccionada intencionalmente, con el fin de que los seleccionados cumplan con las características planteadas en la encuesta (ver anexo 4). Se escogen 7 especialistas del proyecto.

El contenido de la investigación queda definido en tres capítulos:

Capitulo 1: Fundamentación teórica: en este capítulo se investiga todo lo relacionado a las pruebas de software y se hace referencia sobre la metodología utilizada, el rol de Diseñador de casos de prueba, los tipos de pruebas que existen y cuáles serán aplicadas en la presente investigación.

Capitulo 2: Diseño y aplicación de las pruebas: en este capítulo se definen, diseñan y aplican los casos de prueba correspondientes al módulo.

Capitulo 3: Resultados obtenidos: se hace un resumen de los errores encontrados durante la aplicación de las pruebas.

CAPÍTULO 1: Fundamentación Teórica

1.1 Introducción

Todas las etapas definidas para el ciclo de vida de un software, tienen una determinada importancia. En el caso específico de la que comprende las pruebas de software, es una etapa que representa la auto-aprobación del producto, lo que constituye una de las principales acciones del equipo de desarrollo por garantizar que llegue con la mayor calidad a manos de los clientes, teniendo en cuenta las funcionalidades exigidas.

En este capítulo se brindará información acerca de las pruebas de software. También se describirá la metodología seleccionada para representar cada etapa del ciclo de vida del software, se caracterizará el proceso de pruebas y el rol de Diseñador de casos de prueba.

1.2 Conceptos asociados al dominio del problema

1.2.1 Registro Minero

Un Registro Minero es un sistema de control en el que figuran, entre otros, los datos relativos a los derechos concedidos a personas naturales y jurídicas para realizar actividades mineras. [2]

El Registro Minero es un sistema de inscripción, autenticidad y publicidad de los títulos mineros con el derecho a explorar y explotar el suelo y subsuelo de acuerdo al Código de Minas. [3]

Se puede definir un Registro Minero, como un sistema donde se controlan y autentican los derechos de exploración y explotación, concedidos a personas para realizar actividades mineras, regidos por la Ley de Minas. [4]

1.2.2 Rol

Papel que desempeña una persona o un grupo en cualquier actividad. [5]

Según la **RAE**¹ un rol es la función que alguien o algo cumple.

¹ **RAE**: Real Academia Española

En un proyecto informático, se define como la realización de una tarea específica por parte de una o varias personas, que está estrechamente relacionada con otras tareas que le dan cumplimiento al objetivo del proyecto.

1.2.3 Prueba

Según la RAE, una prueba es un examen que se hace para demostrar o comprobar los conocimientos o aptitudes de alguien.

La Prueba puede ser definida 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. [5]

1.2.4 Prueba de Software

Las pruebas de software, son los procesos que permiten verificar y revelar la calidad de un producto software. [6]

Prueba de software es la ejecución de un programa con la intención de descubrir un error. Además es una técnica experimental para la búsqueda de errores en los programas. [7]

Una prueba es un análisis exhaustivo que se hace del comportamiento del software ante determinadas condiciones de uso.

1.2.5 Calidad

Según la RAE la calidad es una propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor.

Propiedad o conjunto de propiedades inherentes a una persona o cosa que permiten apreciarla con respecto a las restantes de su especie. [5]

Calidad es un criterio de medida, basado en las características y particularidades de cada componente del entorno, que permite establecer un orden lógico.

1.2.6 Calidad del software

¿Qué es la calidad del software?

La calidad del software no es más que el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario. [8]

“La totalidad de las características de una entidad que influyen en su capacidad de satisfacer necesidades explícitas e implícitas”. [9]

El software en su gran extensión, como entidad intelectual, es más difícil de caracterizar que los objetos físicos. No obstante, sí existen las medidas de características de un programa. Entre estas propiedades se incluyen complejidad ciclométrica, cohesión, número de puntos de función, líneas de código y muchas otras. [10]

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

“El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas”. [9]

Por tanto, la calidad del software no es más que las características que se deben cumplir para satisfacer las necesidades del usuario.

1.3 Objeto de Estudio

Atendiendo a las apreciaciones realizadas de la situación existente, para la presente investigación se ha definido como objeto de estudio el proceso de pruebas de sistemas de gestión de información. Es de vital importancia conocer los detalles de este proceso, en aras de lograr la mayor eficacia del resultado.

1.3.1 Situación Problemática

En la Universidad de Ciencias Informáticas se encuentra la Facultad 9, dentro de la cual se halla el Centro Productivo GEySED, que cuenta con dos departamentos. Uno de ellos es el de Geoinformática, donde se desarrollan varios productos como el Sistema de Gestión de Datos Geológicos (SGDG). Esta aplicación está dividida en diferentes

módulos, entre ellos el módulo Registro Minero. Este es de gran importancia para el país, en el mismo se controla toda la información referente al proceso de gestión y recuperación de la información de las solicitudes y de los derechos mineros. Este módulo tiene entre sus funciones ordenar y fiscalizar la actividad geológica, minera y petrolera de la República de Cuba. Dicho módulo acaba de ser implementado por parte del equipo de desarrollo, por lo que para desplegarlo se hace necesario verificar que el mismo funcione correctamente.

La realización de este sistema implica un gran compromiso por parte de cada uno de los miembros del equipo con el cliente, teniendo en cuenta que no se le debe entregar un sistema con errores y que cuente con la calidad requerida. El hecho de entregar un producto inmaduro, trae como consecuencia una labor penosa y costosa de soporte por parte del equipo de desarrollo con los clientes, además de pérdida de imagen de la entidad en cuanto a la calidad de sus productos, tanto a nivel nacional como internacional, así como una baja de la credibilidad de la misma. Es por tal motivo que surge la necesidad de desarrollar un proceso de pruebas que haga posible corregir las fallas en el software antes de su liberación.

1.4 Metodologías de desarrollo de software

Una metodología es un conjunto de procedimientos, técnicas, herramientas y soporte documental que ayuda a los desarrolladores a realizar un nuevo software, además de constituir una asistencia fundamental cuando se da mantenimiento o actualización a un software implementado. [11]

El desarrollo de software no es, sin dudas, una tarea fácil. Como resultado a este problema, ha surgido una alternativa desde hace mucho: las Metodologías de desarrollo. Las cuales imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen desarrollando un proceso detallado con un fuerte énfasis en planificar, inspirado por otras disciplinas de la ingeniería.

Durante el ciclo de vida del software, han existido diferentes generaciones de metodologías :[11]

- Desarrollo Convencional (sin metodología).

- Desarrollo Estructurado.
- Desarrollo Orientado a Objeto.

Existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Un ejemplo de ellas son las propuestas tradicionales o robustas centradas específicamente en el control del proceso. Estas han demostrado ser efectivas y necesarias en un gran número de proyectos, sobre todo aquellos proyectos de gran tamaño (respecto a tiempo y recursos).

Dentro de las metodologías robustas o también llamadas fuertes, se encuentra el Proceso Unificado de Desarrollo, conocido como RUP², por sus siglas en inglés, la cual es una de las más utilizadas en estos tiempos y es la que se aplica en la presente investigación.

1.4.1 Proceso Unificado de Desarrollo (RUP)

Para confeccionar la representación de la arquitectura se tomó como metodología de desarrollo RUP, que define la representación arquitectónica mediante las 4+1 vistas arquitectónicas: la vista de casos de uso, la vista lógica, la vista de procesos, la vista de implementación y la vista de despliegue. [12]

RUP es una metodología para la ingeniería de software, que va más allá del mero análisis y diseño orientado a objetos, para proporcionar una familia de técnicas que soportan el ciclo completo de desarrollo de software. El resultado es un proceso basado en componentes, dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental. [13]

² RUP: Rational Unified Process

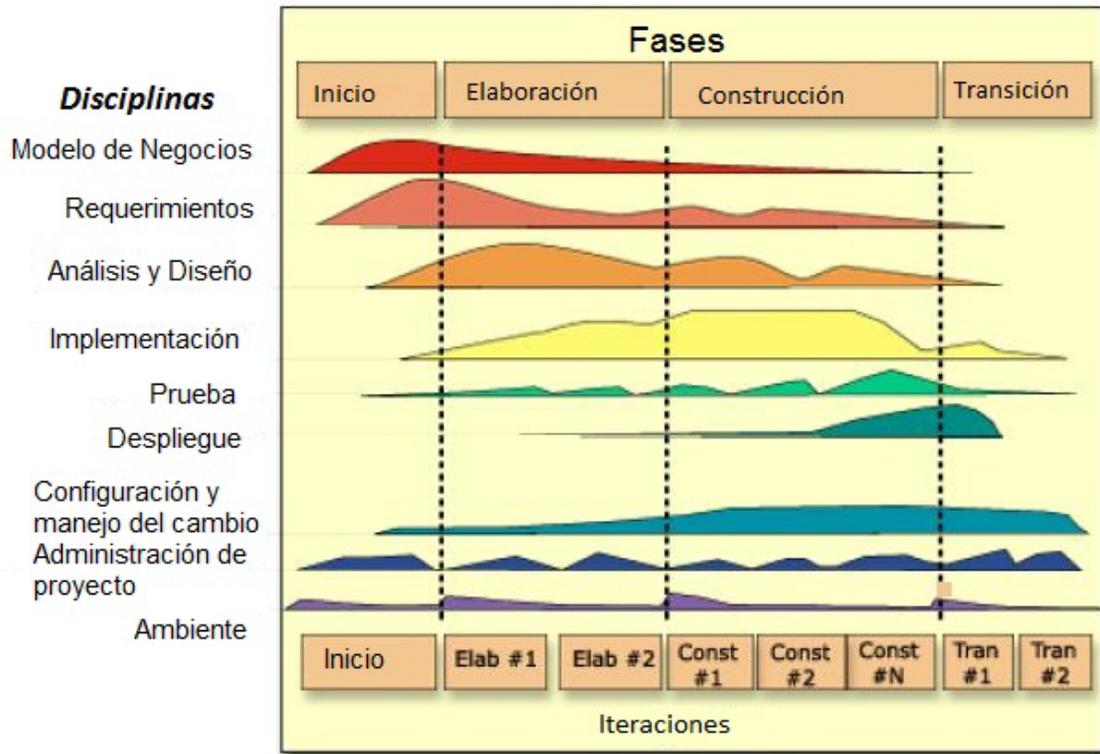


Figura 1: Fases y Flujos de Trabajo de RUP

Características principales de RUP

- **Guiado por los Casos de Uso:** en RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba. Los Casos de Uso constituyen un elemento integrador y una guía de trabajo. [14]
- **Centrado en la Arquitectura:** presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento. Existe una interacción entre los Casos de Uso y la arquitectura. Los Casos de Uso deben encajar en la arquitectura cuando se llevan a cabo y la arquitectura debe permitir el desarrollo de todos los Casos de Uso requeridos, actualmente y en el futuro. Esto provoca que tanto arquitectura como Casos de Uso deban evolucionar en paralelo durante todo el proceso de desarrollo de software. [14]
- **Iterativo e Incremental:** Durante todo el proceso de desarrollo se producen versiones incrementales (que se acercan al producto terminado) del producto en

desarrollo. El proceso iterativo e incremental consta de una secuencia de iteraciones. Cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Cada iteración se analiza cuando termina. Se puede determinar si han aparecido nuevos requisitos o han cambiado los existentes, afectando a las iteraciones siguientes. [14]

RUP se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo constituye una versión del sistema. Cada ciclo consta de cuatro fases:

- **Inicio:** Definir el alcance del proyecto.
- **Elaboración:** Planificar el proyecto, elaborar una arquitectura base.
- **Construcción:** Construir el sistema.
- **Transición:** Transición de los usuarios.

Cada fase se subdivide en iteraciones. En cada iteración se desarrolla en secuencia, un conjunto de disciplinas o flujos de trabajos, los que están divididos en dos grupos.

De Ingeniería:

- **Modelación del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización. [10]
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen. [10]
- **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar. [10]
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación. [10]
- **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida. [10]

- **Instalación:** Produce release³ del producto y realiza actividades (empaquete, instalación y asistencia a usuarios) para entregar el software a los usuarios finales. [10].

De apoyo:

- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes. [10]
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones. [10]
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.[10]

1.5 Proceso de pruebas de RUP.

Las pruebas de software son los procesos que permiten verificar y revelar la calidad de un producto, se integran dentro de las diferentes fases del ciclo del software dentro de la ingeniería de software. Así se ejecuta un programa y mediante técnicas experimentales se trata de descubrir qué errores tiene. RUP define cuatro trabajadores en el flujo de trabajo de Pruebas.

- **Trabajador: diseñador de casos de prueba**

Un diseñador de casos de prueba debe garantizar la integridad del modelo de pruebas, asegurando que el modelo cumple con su propósito. Planea las pruebas, selecciona y describe los casos de prueba. Además, es el responsable de los artefactos casos de prueba, Procedimientos de Prueba, Plan de Pruebas y Evaluación de Pruebas. [11]

³ **Release:** Es una versión de un producto.

➤ **Trabajador: ingeniero de componentes**

Son responsables de los componentes de prueba que automatizan algunos de los procedimientos de prueba. [11]

➤ **Trabajador: ingeniero de pruebas de integración**

Son los responsables de realizar las pruebas de integración que se necesitan para cada construcción producida en el flujo de implementación. Documentan los defectos que aparecen como resultados de las pruebas de integración. [11]

➤ **Trabajador: ingeniero de pruebas de sistema**

Un ingeniero de pruebas de sistema es responsable de realizar las pruebas de sistema necesarias, sobre una construcción que muestra el resultado de una iteración completa. Las pruebas de sistema se llevan a cabo para verificar interacciones entre los actores y el sistema. [11]

1.5.1 Artefactos del flujo de trabajo de Prueba

Los siguientes artefactos son los que se definen para el flujo de trabajo de Prueba de acuerdo a la metodología RUP: [15]

➤ **Artefacto: Modelo de pruebas**

Describe cómo se prueban los componentes en el modelo de implementación. El modelo de pruebas es una colección de casos de prueba, procedimientos de prueba y componentes de prueba.

➤ **Artefacto: Caso de prueba**

Un caso de prueba específica 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.

➤ **Artefacto: Procedimiento de prueba**

Un procedimiento de prueba especifica cómo realizar uno o varios casos de prueba o partes de estos.

➤ **Artefacto: Componente de prueba**

Un componente de prueba automatiza uno o varios procedimientos de prueba o partes de ellos.

➤ **Artefacto: Plan de Prueba**

El Plan de Prueba describe las estrategias, recursos y planificación de la prueba. La estrategia incluye definición de tipo de pruebas a realizar por iteración, sus objetivos, nivel de cobertura y código necesario.

➤ **Artefacto: Defecto**

Un defecto es una anomalía del sistema. Un defecto puede ser utilizado para localizar cualquier elemento que los desarrolladores necesitan registrar como síntoma de un problema.

➤ **Artefacto: Evaluación de prueba**

Una evaluación de prueba es una caracterización que se realiza de los resultados obtenidos luego de realizada la prueba.

1.5.2 Actividades del flujo de trabajo de Prueba

Planificar Prueba: planificar los esfuerzos de prueba en una iteración:

- Describir una estrategia de prueba: qué tipo de pruebas ejecutar, cómo y cuándo y cómo determinar si el esfuerzo tiene éxito.
- Estimar los requisitos (recursos humanos y de sistema)
- Planificar el esfuerzo de prueba.

Diseñar Prueba: caso de prueba y procedimiento de prueba.

- Identificar y describir los casos de prueba.

- Identificar los procedimientos de prueba especificando cómo realizar los casos de prueba.

Implementar Prueba: automatizar procedimientos de prueba creando componentes de prueba.

Realizar Pruebas de Integración: casos de prueba para verificar que los componentes interaccionan entre sí de la forma apropiada después de haber sido integrados en una construcción. Pasos: [15]

- Realizar pruebas de integración relevante, manual o ejecutar algún componente de prueba.
- Comparar los resultados obtenidos con los esperados e investigar los que no coinciden.
- Informar defectos a los Ingenieros de componentes.
- Informar defectos a los diseñadores de pruebas.

Realizar Pruebas de Sistema: verifican que el comportamiento externo del sistema satisface los requisitos establecidos por los clientes y futuros usuarios del mismo.

- Probar que el sistema funciona correctamente como un todo.
- Si los CU satisfacen los requerimientos de los actores.
- Probar con diferentes configuraciones, diferentes cargas al sistema, diferentes número de actores y distintos tamaños de BD.

Evaluar las Pruebas: evaluar los esfuerzos de la prueba en una iteración. Se usan distintas métricas que permiten determinar el nivel de calidad del software y qué cantidad de pruebas es necesario realizar. [15]

1.5.3 Tipos de Pruebas de Software

En epígrafes anteriores se ha abordado sobre la importancia de la etapa de pruebas durante el ciclo de vida del software. Como es lógico, no todos los software tienen las mismas características, ni están desarrollados con los mismos propósitos. Es por esto que el tipo de prueba a realizar dependerá del tipo de implementación que se estará llevando a cabo.

Los diferentes tipos de pruebas permitirán llegar a los más mínimos detalles en algunos casos, mientras que en otras situaciones, el resultado de la prueba será más superficial. Es por esto que se debe conocer bien el tipo de software al que se le aplicarán las pruebas y luego, determinar el tipo de prueba a realizar.

A continuación se muestran los **niveles de pruebas**:

➤ **Prueba de unidad o unitaria**

Prueban el diseño y el comportamiento de cada uno de los componentes una vez construidos. [11]

➤ **Prueba de Integración**

Comprueban la correcta unión de los componentes entre sí a través de sus interfaces, y si cumplen con la funcionalidad establecida. [11]

➤ **Prueba de Sistema**

Prueban a fondo el sistema, comprobando su funcionalidad e integridad globalmente, en un entorno lo más parecido posible al entorno final de producción. [11]

➤ **Prueba de Aceptación**

Verifican si el sistema cumple con todos los requisitos indicados y permite que los usuarios del sistema den el visto bueno definitivo. [11]

En los procesos de pruebas se le pueden aplicar a los productos diferentes tipos de comprobación, de acuerdo a las necesidades que este tenga, que hacen posible la acentuación de la calidad de un software. Algunas de las pruebas son las de caja negra (CN) y de caja blanca (CB), de bucles, de flujo de datos; pruebas manuales y automáticas; pruebas unitarias, de integración, funcionales y de rendimiento; pruebas alfa y pruebas beta; pruebas de stress.

Existen otros tipos de prueba, pero no siempre es necesario realizarse todas, o no siempre se tiene la posibilidad de realizarlas todas. Las pruebas que se le realizarán al módulo Registro Minero son las pruebas de caja negra y caja blanca, las cuales se caracterizan a continuación.

1.5.4 Pruebas de caja blanca

Las pruebas de caja blanca se centran en la estructura de control del programa. Se obtienen casos de prueba que aseguren que durante la prueba se han ejecutado, por lo menos una vez, todas las sentencias del programa y que se ejercitan todas las condiciones lógicas. [11]

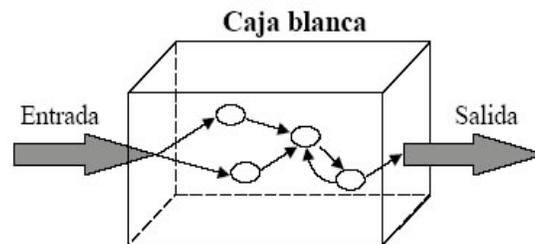


Figura 2: Prueba de caja blanca

Técnicas de caja blanca.

- La prueba del camino básico, hace uso de grafos de programa (o matrices de grafos) para obtener el conjunto de pruebas linealmente independientes que aseguren la total cobertura.
- La prueba de condiciones es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en el módulo de un programa. [11]

- La prueba de flujo de datos representa el flujo de control lógico de un programa y se utiliza para trazar más fácilmente los caminos de éste. (Cada nodo representa una o más sentencias procedimentales y cada arista representa el flujo de control). [10]
- La prueba de los bucles complementa a otras técnicas de caja blanca, proporcionando un procedimiento para ejercitar bucles de distintos grados de complejidad.

Debido a las características que presenta, se escogió para la presente investigación aplicar la técnica de camino básico, la cual permite al Diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizan, que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa.

1.5.5 Pruebas de caja negra

Las pruebas de caja negra son diseñadas para validar los requisitos funcionales sin fijarse en el funcionamiento interno de un programa. Las técnicas de prueba de caja negra se centran en el ámbito de información de un programa, de forma que se proporcione una cobertura completa de prueba. [11]

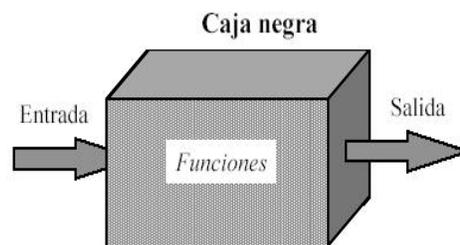


Figura 3: Prueba de caja negra

Métodos de caja negra

- Métodos de pruebas basados en grafo: exploran las relaciones entre los objetos del programa y su comportamiento.
- La partición equivalente divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- El análisis de valores límite prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- La prueba de la tabla ortogonal suministra un método sistemático y eficiente para probar sistemas con un número reducido de parámetros de entrada.[10]

En el caso del método de partición equivalente, que se aplicará como parte de la prueba de caja negra, fue seleccionado teniendo en cuenta que un caso de prueba ideal descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. El objetivo de partición equivalente es reducir el posible conjunto de casos de prueba en uno más pequeño, manejable que evalúe bien el software.

1.6 Conclusiones Parciales

Como resultado del análisis realizado en el capítulo, queda definido que se realizarán las pruebas de caja negra, dentro de ella se escogió el método de partición equivalente, y pruebas de caja blanca seleccionando la técnica de camino básico. Todas las pruebas se realizarán en el nivel de sistema ya que en el mismo se prueban diferentes criterios, como son: rendimiento, robustez, seguridad, usabilidad y funcionalidad.

Se determinó realizar los dos tipos de pruebas, para garantizar que la búsqueda de posibles errores sea más profunda y así disminuir en gran medida la posibilidad de errores en el sistema.

CAPÍTULO 2: Diseño y Aplicación de las pruebas

2.1 Introducción

En el capítulo anterior se efectuó un análisis de los principales conceptos asociados a las pruebas de software. Como parte fundamental de la presente investigación, teniendo en cuenta las definiciones ya realizadas, en el Capítulo 2 se diseñarán y aplicarán todos los casos de prueba correspondientes al módulo Registro Minero. Como resultado se obtendrá un Plan de Pruebas, que fungirá como guía fundamental para la etapa de evaluación del software.

2.2 Características a probar

Para determinar los aspectos que se evaluarán en un software, se debe tener en cuenta el tipo de producto, además de qué se pretende probar. Siempre se deben tener en cuenta algunas características presentes en cada aplicación, como es el caso del cumplimiento con las funcionalidades requeridas, ejecución sin errores, facilidad de comprensión para el usuario y si cuenta con una documentación detallada y específica. Otros aspectos medibles que se tienen en cuenta son los relacionados con el código del programa y que permiten conocer si está bien estructurado y es reutilizable.

2.3 Plan General de Pruebas

La necesidad de comprobar el correcto funcionamiento de un producto hace que sea necesario un Plan de Pruebas, el cual describe las estrategias, recursos y planificación con el cual se procederá a realizar una serie de ensayos que permitan obtener resultados correctos y erróneos con el fin de analizar el proceso de ejecución. Al realizar este conjunto de pruebas se determinará si el programa es erróneo, sobre todo en casos extremos y particulares, tanto si estos fallos se producen por una mala implementación del programa o bien por un uso específico que realiza el usuario.

El aspecto más importante para realizar la planificación de este conjunto de pruebas es abarcar con ellas todos los requisitos que debe cumplir el programa y que por tanto responda correctamente a las funcionalidades que se le solicitan inicialmente. Puesto que en el documento de Especificación de Requisitos de Software ya se ha realizado una evaluación de las funcionalidades que debe incluir el programa, se toma este documento de referencia para desarrollar el Plan de Pruebas del sistema.

El objetivo general del plan es establecer la cronología y condiciones para la aplicación de las pruebas, de manera que se obtiene un sistema que pueda ser completado con una recepción total de los interesados y entrar en operación con la totalidad de las funcionalidades requeridas para su funcionamiento. [16]

El propósito del Plan de Pruebas es explicitar el alcance, enfoque, recursos requeridos, calendario, responsables y manejo de riesgos de un proceso de pruebas. [16] En el Plan de Pruebas que se definió para el módulo Registro Minero se identifican los elementos que serán probados.

2.4 Objetivos

El objetivo principal de realizarle las pruebas al módulo Registro Minero es garantizar la calidad del producto desarrollado, aplicando un conjunto de casos de prueba que serán definidos, siendo la principal razón encontrar defectos en el software, para que los desarrolladores puedan corregirlos. Una vez aplicados los casos de prueba se debe tener en cuenta que, esta tuvo éxito si se descubren errores en el sistema y fracasa si hay errores y no son detectados.

2.5 Estrategia de Pruebas

La Estrategia de Prueba es una guía a la hora de realizarle las pruebas a un producto. Esta vela por que el producto tenga los requerimientos que el cliente pidió. La Estrategia de Prueba de un Software integra un conjunto de actividades que describen los pasos que hay que llevar a cabo en un proceso de prueba: la planificación, el diseño de casos de prueba, la ejecución y los resultados, tomando en consideración cuánto esfuerzo y recursos se van a requerir, con el fin de obtener como resultado una correcta construcción del software.

Para llevar a cabo el proceso de prueba del software en la presente investigación, la estrategia planteada fue la siguiente:

- Realizar una investigación acerca del proceso de diseño y aplicación de los casos de prueba.
- Identificar en el documento de Modelo de Sistema, los CU a los que se va a aplicar los casos de prueba.
- Realizar el diseño de los casos de prueba.

- Aplicar las pruebas a los casos de uso del sistema ya implementados.
- Obtención de resultados.
- Creación de la lista de defectos.

2.6 Diseño de casos de prueba de caja negra

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. Lo que se prueba puede venir dado por un requisito o colección de requisitos del sistema.

Un caso de prueba basado en un caso de uso define típicamente una prueba del sistema como caja negra, es decir, una prueba del comportamiento observable externamente del sistema. [11]

Las pruebas de caja negra tienen como objetivo principal comprobar la funcionalidad del sistema. Para la aplicación segura de las mismas, se aplica la técnica de partición de equivalencia que resulta efectiva cuando se desea comprobar la validez de cada entrada en los diferentes módulos.

2.6.1 Caso de prueba de caja negra: Gestionar Medidas.

Módulo Registro Minero

Nombre del CU: Gestionar Medidas

Descripción General: Una vez dentro del sistema el administrador puede insertar, modificar o eliminar los datos de una medida.

Secciones a probar en el Caso de Uso:

Buscar Medidas

Insertar Medidas

Modificar Medidas

Eliminar Medidas

Tabla 1: Diseño de caja negra del Caso de Uso: Gestionar Medida.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC1: Seleccionar Opción.	EC1.1: Mostrar opciones.	Se muestran dos opciones “Buscar Medidas” y “Adicionar Medidas”, también se muestra un listado de todas las medidas que hay registradas en el sistema, en las cuales se puede también modificar dicha medida o eliminar.
SC2: Buscar Medidas	EC 2.1: Se marca la opción buscar	El sistema muestra una lista con todas las medidas y sus datos.
	EC 2.2: Se introduce el texto por el cual se desea buscar una medida.	El sistema muestra un listado con todas las medidas y sus datos que tienen en cualquiera de sus campos el texto entrado.
	EC 2.3 : Los datos entrados son incorrectos	El sistema no muestra nada.
	EC 2.4: Generar documento Word.	Se genera un documento Word con los resultados obtenidos en una búsqueda determinada.
	EC 2.5: Generar documento PDF.	Se genera un documento PDF con los resultados obtenidos en una búsqueda determinada.
SC3: Insertar Medida	EC 3.1: Se adiciona una medida correctamente	El sistema muestra un formulario para insertar datos. Primero se selecciona el nombre del derecho minero, y a partir de ahí se muestran los datos a insertar: Fecha de imposición de la medida, fecha límite, medida aplicada y las observaciones.

	EC 3.2: : Los datos entrados son incorrectos	El sistema muestra un mensaje especificando el error, y no se realiza el adicionar una medida.
	EC 3.3: Se cancela la inserción.	Después que inserte los datos puedo cancelar, y el sistema no inserta ningún dato y cierra la interfaz de insertar.
SC4: Modificar Medidas	EC 4.1: Se modifica una medida correctamente.	El sistema muestra un formulario para modificar con los siguientes datos: nombre del derecho minero, medida aplicada, fecha de imposición, fecha límite, fecha en que se cumple la medida, y observaciones.
	EC 4.2: : Los datos no son correctos	El sistema manda un mensaje especificando el error, y no se modifican los datos.
	EC 4.3: Se cancela la modificación.	Luego de haber modificado los datos, se puede cancelar dicha modificación, y el sistema no modifica y cierra la interfaz de modificar.
SC5: Eliminar una Medida	EC5.1: Se elimina una medida.	El sistema da la opción de eliminar, se selecciona en la lista la medida que se desea eliminar, el sistema muestra un mensaje “¿Está seguro que desea eliminarlo?” (La cual tiene dos opciones, aceptar y cancelar), si da en la opción aceptar, se elimina y el sistema debe mostrar un mensaje de confirmación, informando que se ha eliminado correctamente una medida.
	EC5.2: No se elimina	Si no se desea eliminar luego de haber apretado el botón eliminar, el sistema muestra un mensaje “¿Está seguro que desea eliminarlo?” (La cual tiene dos opciones, aceptar y cancelar), si da en la opción cancelar, no se elimina. El sistema debe de mostrar un mensaje de confirmación informando que no se ha eliminado.

2.6.2 Caso de prueba de caja negra: Gestionar Compatibilizaciones.

Módulo Registro Minero

Nombre del CU: Gestionar Compatibilizaciones.

Descripción General: Una vez dentro del sistema el administrador puede insertar, modificar o eliminar los datos de una compatibilización.

Secciones a probar en el Caso de Uso:

Buscar Compatibilización

Insertar Compatibilización

Modificar Compatibilización

Eliminar Compatibilización

Tabla 2: Diseño de caja negra del Caso de Uso: Gestionar Compatibilización.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC1: Seleccionar Opción.	EC1.1: Mostrar opciones.	Se muestran dos opciones “Buscar Compatibilización” y “Adicionar Compatibilización”, también se muestra un listado de todas las compatibilizaciones que hay registradas en el sistema, en las cuales se puede también modificar o eliminar.
SC2: Buscar Compatibilizaciones	EC 2.1: Se marca la opción buscar	El sistema muestra una lista con todas las compatibilizaciones y sus datos.
	EC 2.2: Se introduce el texto por el cual se desea buscar una compatibilización.	El sistema muestra un listado con todas las compatibilizaciones y sus datos que tienen en cualquiera de sus campos el texto entrado.
	EC 2.3 : Los datos entrados son incorrectos	El sistema no muestra nada.

	EC 2.4: Generar documento Word.	Se genera un documento Word con los resultados obtenidos en una búsqueda determinada.
	EC 2.5: Generar documento PDF.	Se genera un documento PDF con los resultados obtenidos en una búsqueda determinada.
SC3: Insertar Compatibilizaciones	EC 3.1: Se adiciona una compatibilización correctamente	El sistema muestra un formulario para insertar datos. Primero se selecciona la solicitud en un formulario, luego se escoge la fecha de envió y la fecha límite, se ponen observaciones que no son obligatorias y el organismos. Cuando se introducen los datos el sistema los adiciona.
	EC 3.2: Los datos entrados son incorrectos	El sistema manda un mensaje de error, especificando cuales datos son los incorrectos.
	EC 3.3: Se cancela la inserción.	Después que inserte los datos puedo cancelar, y el sistema no adiciona ningún dato y cierra la interfaz de insertar.
SC4: Modificar Compatibilizaciones	EC 4.1: Se modifica una Compatibilizaciones correctamente.	El sistema muestra un formulario para modificar con los siguientes datos: nombre de solicitud, fecha de envió, limite y de recibo, observaciones y organismos. Se selecciona la solicitud y el organismo que compatibiliza y modifica los datos. El sistema modifica los datos.
	EC 4.2: Los datos no son correctos	El sistema manda un mensaje especificando el error, y no se modifican los datos.
	EC 4.3: Se cancela la modificación	Luego de haber modificado los datos, se puede cancelar dicha modificación, y el sistema no modifica y cierra la interfaz de modificar

SC5: Eliminar Compatibilizaciones	EC 5.1: Se elimina una Compatibilización	El sistema da la opción de eliminar, se selecciona en la lista la compatibilización que se desea eliminar, el sistema muestra un mensaje “¿Está seguro que desea eliminarlo?” (La cual tiene dos opciones, aceptar y cancelar), si da en la opción aceptar, se elimina y el sistema debe mostrar un mensaje de confirmación, informando que se ha eliminado correctamente una compatibilización.
	EC 5.2: No se elimina	Si no se desea eliminar luego de haber apretado el botón eliminar, el sistema muestra un mensaje “¿Está seguro que desea eliminarlo?” (La cual tiene dos opciones, aceptar y cancelar), si da en la opción cancelar, no se elimina. El sistema debe de mostrar un mensaje de confirmación informando que no se ha eliminado.

2.6.3 Caso de prueba de caja negra: Gestionar Obligaciones.

Módulo Registro Minero

Nombre del CU: Gestionar Obligaciones

Descripción General: Una vez dentro del sistema el administrador puede insertar, modificar o eliminar los datos de una obligación.

Secciones a probar en el Caso de Uso:

Buscar Obligación

Insertar Obligación

Modificar Obligación

Eliminar Obligación

Tabla 3: Diseño de caja negra del Caso de Uso: Gestionar Obligación.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC1: Seleccionar Opción.	EC1.1: Mostrar opciones.	Se muestran dos opciones “Buscar Obligación” y “Adicionar Obligación”, también se muestra un listado de todas las obligaciones que hay registradas en el sistema, en las cuales se puede también modificar o eliminar.
SC2: Buscar Obligaciones	EC 2.1: Se marca la opción buscar	El sistema muestra una lista con todas las obligaciones y sus datos.
	EC 2.2: Se introduce el texto por el cual se desea buscar una obligación.	El sistema muestra un listado con todas las obligaciones y sus datos que tienen en cualquiera de sus campos el texto entrado.
	EC 2.3 : Los datos entrados son incorrectos	El sistema no muestra nada.

	EC 2.4: Generar documento Word.	Se genera un documento Word con los resultados obtenidos en una búsqueda determinada.
	EC 2.5: Generar documento PDF.	Se genera un documento PDF con los resultados obtenidos en una búsqueda determinada.
SC3: Insertar Obligaciones	EC 3.1: Se adiciona una obligación correctamente	El sistema muestra un formulario para insertar datos. Primero se selecciona el nombre del derecho minero, y a partir de ahí se muestran los datos a insertar: Tipo de obligación, fecha de presentación y de cumplimiento, y las observaciones.
	EC 3.2: Los datos entrados son incorrectos	El sistema muestra un mensaje especificando el error, y no se realiza el adicionar una obligación.
	EC 3.3: Se cancela la inserción.	Después que inserte los datos puedo cancelar, y el sistema no inserta ningún dato y cierra la interfaz de insertar.
SC4: Modificar Obligación	EC 4.1: Se modifica una obligación correctamente.	El sistema muestra un formulario para modificar con los siguientes datos: Tipo de obligación, fecha de presentación y de cumplimiento, y las observaciones
	EC 4.2: Los datos no son correctos.	El sistema manda un mensaje especificando el error, y no se modifican los datos.
	EC 4.3: Se cancela la modificación.	Luego de haber modificado los datos, se puede cancelar dicha modificación, y el sistema no modifica y cierra la interfaz de modificar.
SC5: Eliminar una obligación	EC5.1: Se elimina una obligación.	El sistema da la opción de eliminar, se selecciona en la lista la obligación que se desea eliminar, el sistema muestra un mensaje "¿Está seguro que desea eliminarlo?" (La cual tiene dos opciones, aceptar y cancelar), si da en la opción

		aceptar, se elimina y el sistema debe mostrar un mensaje de confirmación, informando que se ha eliminado correctamente una obligación.
	EC5.2: No se elimina	Si no se desea eliminar luego de haber apretado el botón eliminar, el sistema muestra un mensaje “¿Está seguro que desea eliminarlo?” (La cual tiene dos opciones, aceptar y cancelar), si da en la opción cancelar, no se elimina. El sistema debe de mostrar un mensaje de confirmación informando que no se ha eliminado dicha obligación.

2.7 Diseño de casos de prueba de caja blanca

El diseño de los casos de prueba de caja blanca especifica una prueba del sistema, es decir, una prueba de la interacción interna de los componentes del sistema. Para la realización de dicha prueba se siguió la técnica de diseño del camino básico, la cual permite obtener una medida de la complejidad lógica de un diseño.[11]

2.7.1 Caso de prueba de caja blanca: Gestionar Medidas.

Paso 1: Dibujar el grafo asociado a partir del código fuente:

```
public function executeBuscarMedidasPage()
{
1   $texto = $this->getRequestParameter('txtTexto');
1   $inicio = $this->getRequestParameter('txtInicio');
1   $cantidad = $this->getRequestParameter('txtCantidad');
1   $a= TmedidaPeer::obtenerMedidasTexto($texto, $cantidad, $inicio);
1   $t= TmedidaPeer::CantidadMedidasTexto($texto);
1   $array_json = '{}';
1   $Json = array();
1   $i = 0;
```

```

2   foreach ($a as $elem)
      {
3       if($elem["fecha_cumplimiento"] == null)
          {
4           $real = "Sin cumplir";
          }
          else
          {
5           $real = $elem["fecha_cumplimiento"];
          }
6       $derecho = TmedidaPeer::buscarDM($elem["idmedidas"]);
6       $Json[$i++] = array (
7       "Derecho Minero" => $derecho->getNombreDm(),
7       "Fecha de Imposici&oacute;n" => $elem["fecha_imposicion"],
7       "Fecha real de Cumplimiento" => $real,
7       "Fecha tope de Cumplimiento" => $elem["fecha_limite"],
7       "id" => $elem["idmedidas"]
          );
      }
8       $j = 0;
8       $titulos = array(0 => "Derecho Minero",
9       1 => "Fecha de Imposici&oacute;n",
9       2 => "Fecha tope de Cumplimiento",
9       3 => "Fecha real de Cumplimiento",
9       4 => "Acciones" ,
9       5 => "1",
9       6 => "2",
9       7 => "3");

10      $acciones= array(
11      array("accion" => "medida/modificar/id/",
11      "img" => '../..../images/icon/edit.png'),
11      array("accion" => "medida/eliminar/",
11      "img" => ' "1",
11         "mensaje" => "Est&aacute; seguro que desea eliminar el medida",
11         "titulo" => "Eliminar Medida" ),
11         array("funcion" => "ajaxDetallesMedida",
11         "img" => '../..../images/icon/filter.png')
12     );
12     $Json[$i++] = $acciones;
12     $Json[$i++] = $titulos;
12     $Json[$i++] = array("total" => $t);
12     $array_json = json_encode($Json);
13     $this->getResponse()->setHTTPHeader("application/json");
14     echo $array_json;
15     return sfView::NONE; }

```

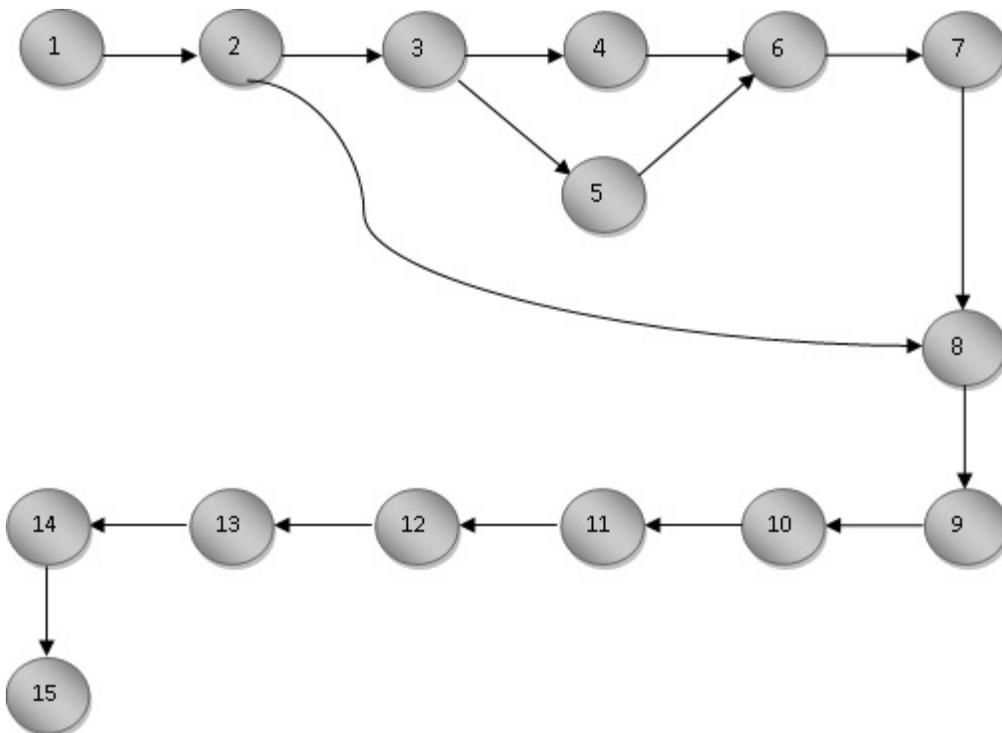


Figura 4: Grafo del caso de prueba de caja blanca: Gestionar Medidas

Paso 2: Cálculo de complejidad ciclomática de Gestionar Medidas.

$$V(G) = \text{Número de Aristas} - \text{Número de Nodos} + 2$$

$$V(G) = 16 - 15 + 2$$

$$V(G) = 3$$

Paso 3: Caminos básicos

CB1: 1 – 2 – 3 – 4 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15

CB2: 1 – 2 – 3 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15

CB3: 1 – 2 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15

Paso 4: Caso de prueba para el camino básico.

CB1: 1 – 2 – 3 – 4 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15

Caso de prueba: Buscar medidas

Entrada: Criterio de búsqueda ejemplo buscar = Registro Minero

Resultado Esperado: No se encuentra la medida.

Resultado de la prueba: Insatisfactorio

2.7.2 Caso de prueba de caja blanca: Gestionar Compatibilizaciones.

Paso 1: Dibujar el grafo asociado a partir del código fuente

```
public function executeBuscarCompatPage()
{
1     $texto = $this->getRequestParameter('txtTexto');
1     $inicio = $this->getRequestParameter('txtInicio');
1     $cantidad = $this->getRequestParameter('txtCantidad');
1     $a= TcompatibilizacionPeer::obtenerCompatTexto($texto, $cantidad,
        $inicio);

1     $t= TcompatibilizacionPeer::CantidadCompatTexto($texto);

1     $array_json = '{}';
1     $Json = array();
1     $i = 0;
2     foreach ($a as $elem)
        {
3         if($elem["fecha_respuesta"]==null)
```

```

    {
4      $resp = "Sin Respuesta";
    }
    else
    {
5      $resp = $elem["fecha_respuesta"];
    }
6      $solicitud =
TcompatibilizacionPeer::getNombreSolicitud($elem["idcompatibilizacion"]);
6      $Json[$i++] = array (
7      "Solicitud" => $solicitud->getNombre(),
7      "Fecha de Envio" => $elem["fecha_envio"],
7      "Organismo" => $elem["organismo"],
7      "Fecha de Respuesta" => $resp,
7      "Fecha L&iacute;mite" => $elem["fecha_limite"],
7      "id" => $elem["idcompatibilizacion"],

    );
}
8      $j = 0;
8      $titulos = array(0 =>"Solicitud" ,
9          1 => "Organismo",
9          2 => "Fecha de Envio",
9          3 => "Fecha L&iacute;mite",
9          4 => "Fecha de Respuesta",
9          5 => "Acciones" ,
9          6 => "1",
9          7 => "2",
9          8 => "3");

10     $acciones= array(
        array("accion" => "compatibilizacion/modificar/id/", "img" =>
        '../..../images/icon/edit.png')
        array("accion" => "compatibilizacion/eliminar/",

```

```

        "img" => ' "1",
        "mensaje" => "Est&aacute; seguro que desea eliminar la
        compatibilizaci&oacute;n",
        "titulo" => "Eliminar Compatibilizaci&oacute;n" ),
        array("funcion" => "ajaxDetallesCompatibilizacion",
        "img" => '../../images/icon/filter.png')
    );
10     $Json[$i++] = $acciones;
10     $Json[$i++] = $titulos;
10     $Json[$i++] = array("total" => $t);
10     $array_json = json_encode($Json);
11     $this->getResponse()->setHTTPHeader("application/json");
12     echo $array_json;
13     return sfView::NONE;
}

```

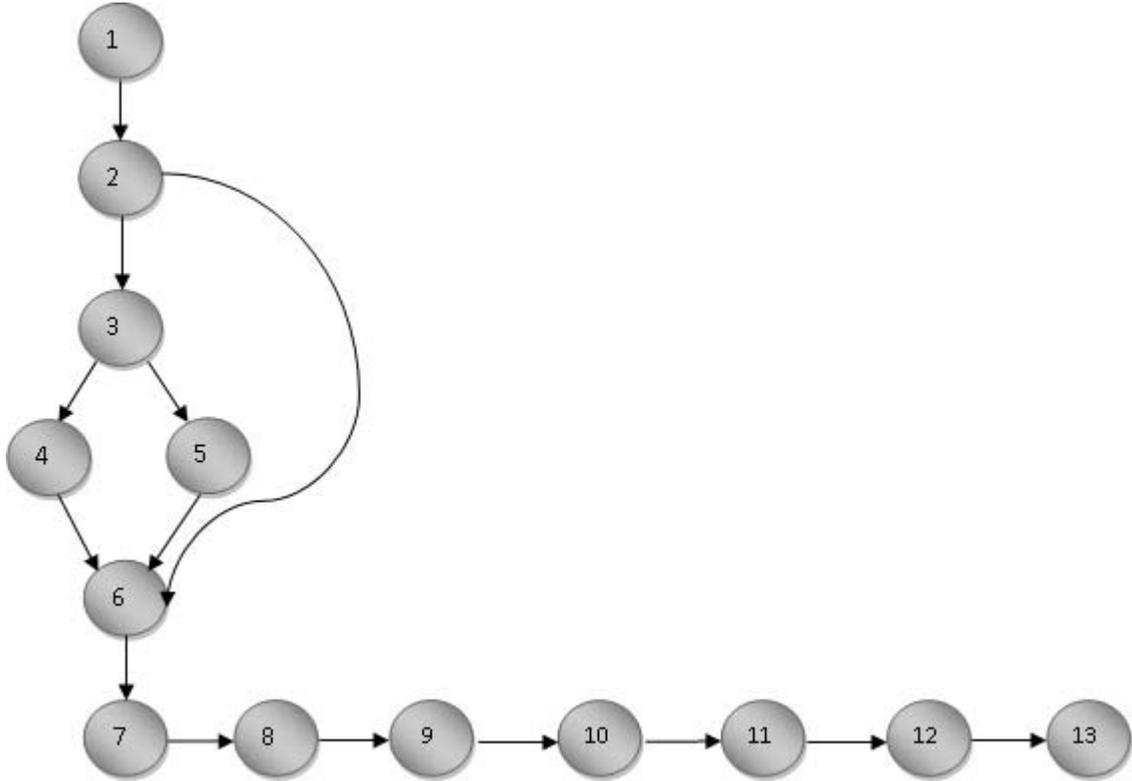


Figura 5: Grafo del caso de prueba de caja blanca: Gestionar Compatibilizaciones.

Paso 2: Calcular la Complejidad Ciclomática

$V(G) = \text{Número de Aristas} - \text{Número de Nodos} + 2$

$V(G) = 14 - 13 + 2$

$V(G) = 3$

Paso 3: Determinar los Caminos Independientes

Caminos Básicos

CB1: 1_2_3_4_6_7_8_9_10_11_12_13

CB2: 1_2_3_5_6_7_8_9_10_11_12_13

CB3: 1_2_6_7_8_9_10_11_12_13

Paso 4: casos de prueba del Camino Básico

CB2: 1_2_3_5_6_7_8_9_10_11_12_13

Caso de prueba: Buscar Compatibilizaciones

Entrada: Criterio de búsqueda ejemplo buscar = UCI

Resultado Esperado: Se encuentra la compatibilización.

Resultado de la prueba: Satisfactorio

2.7.3 Caso de prueba de caja blanca: Gestionar Obligaciones

Paso 1: Dibujar el grafo asociado a partir del código fuente

```
public static function registrarObligacion($idDerecho, $tipoOblig, $fechaPres, $observac,  
$frecuencia, $trimestre)
```

```
{  
1     Propel::getConnection()->executeQuery('set search_path to concesionariom');  
  
2     $obligacion = new Tobligacion();  
  
3     $obligacion->setIdDerechoMinero($idDerecho);  
3     $obligacion->setFechaPresentacion($fechaPres);  
3     $obligacion->setTipoOblig($tipoOblig);  
3     $obligacion->setObservaciones($observac);  
3     $obligacion->setFrecuencia($frecuencia);  
4     if($frecuencia == '3')  
        {
```

```

5     $year = split('-', $fechaPres);
6     if($trimestre != '4')
7     {
8         $obligacion->setTrimestre($trimestre.'/'.$year[2]);
9     }
10    else
11    {
12        $obligacion->setTrimestre($trimestre.'/'.$year[2]+1));
13    }
14    $obligacion->save();
15 }

```

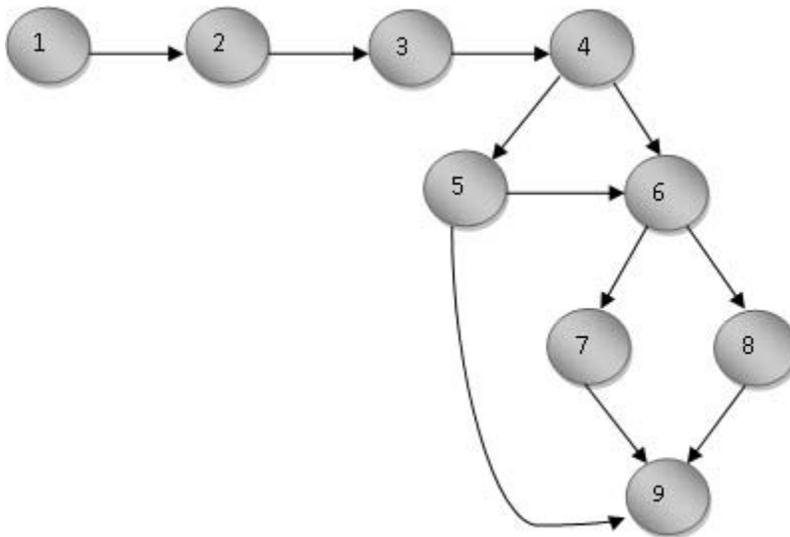


Figura 6: Grafo del caso de prueba de caja blanca: Gestionar Obligaciones

Paso 2: Calcular la Complejidad Ciclomática

$V(G) = \text{Número de Aristas} - \text{Número de Nodos} + 2$

$V(G) = 10 - 9 + 2$

$V(G) = 3$

Paso 3: Determinar los Caminos Independientes

CB1: 1_2_3_4_5_9

CB2: 1_2_3_4_5_6_7_9

CB3: 1_2_3_4_5_6_8_9

Paso 4: casos de prueba del Camino Básico

CB3: 1_2_3_4_5_6_8_9

Caso de prueba: Registrar Obligación

Entrada: Criterio de búsqueda ejemplo adicionar:

Nombre del derecho minero: Derecho obligatorio, Tipo de obligación: ITE Anual, fecha de presentación: 25-2-2010, fecha tope: 1-03-010, observaciones: cumplimiento de la prueba.

Resultado esperado: Se adicionó correctamente.

Resultado de la Prueba: Satisfactoria.

2.8 Conclusiones parciales

Durante el desarrollo del segundo capítulo del trabajo investigativo, se llevaron a cabo las pruebas de caja blanca y caja negra. Para las mismas, se tuvieron en cuenta los estudios realizados en el capítulo anterior, aprovechando los conocimientos adquiridos en el mismo para la elaboración del Plan de Pruebas.

La aplicación de las pruebas permitió evaluar el estado de la implementación del sistema, sirviendo de guía para las acciones inmediatas de corrección. A partir de las tareas de la investigación cumplidas hasta este momento, están creadas las condiciones para describir los resultados obtenidos, los que serán referidos en el próximo capítulo.

CAPÍTULO 3: Resultados Obtenidos

3.1 Introducción

En los capítulos anteriores se analizaron y aplicaron los procesos de pruebas al módulo Registro Minero. Es necesario para la presente investigación documentar los resultados obtenidos en la misma, para que sean aprovechados a cabalidad y de esta forma se apliquen las funcionalidades para las que fueron creadas las pruebas de software. En el presente capítulo, se exponen todos los errores detectados al aplicar los casos de prueba al sistema y así poder corregirlos antes de que sea liberado el producto. Esta documentación agilizará el proceso de corrección de errores.

3.2 Resultados al aplicar los casos de prueba al módulo Registro Minero.

En esta sección del documento se muestran los resultados obtenidos al aplicar cada uno de los casos de prueba diseñados en el capítulo 2.

3.2.1 Caso de prueba Gestionar Medidas.

SC 1: Buscar Medidas

Tabla 4: Resultados al aplicar el caso de prueba: Gestionar Medida. SC2: Buscar Medidas

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 2.1: Se marca la opción buscar	El sistema muestra el listado con todos los medidas existente en el sistema.	Satisfactorio	<ul style="list-style-type: none">• Principal• Registro Minero• Gestión• Medidas• Buscar
EC 2.2: Se introduce el texto por el cual se desea buscar una obligación.	El sistema debe de mostrar una lista con todas las medidas que tienen en alguno de sus campos el texto igual al entrado.	No Satisfactorio	<ul style="list-style-type: none">• Principal• Registro Minero• Gestión• Medidas• Buscar
EC 2.3 : Los datos entrados son incorrectos	El sistema no muestra nada en el resultado de la búsqueda.	Satisfactorio	<ul style="list-style-type: none">• Principal• Registro Minero• Gestión• Medidas• Buscar

EC 2.4: Generar documento Word.	El sistema genera un documento Word con los resultados de la búsqueda previamente realizada.	Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Medidas Buscar
EC 2.5: Generar documento PDF.	El sistema genera un documento PDF con los resultados de la búsqueda previamente realizada.	Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Medidas Buscar

SC 2: Insertar Medidas

Tabla 5: Resultados al aplicar el caso de prueba: Gestionar Medida. SC3: Insertar Medidas

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 3.1: Se adiciona una medida correctamente	El sistema inserta los datos de la Medida.	Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Medidas Adicionar
EC 3.2: Los datos entrados son incorrectos	El sistema muestra un mensaje con todos los errores y no se realiza la inserción de los datos.	No Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Medidas Adicionar
EC 3.3: Se cancela la inserción.	El sistema cancela la inserción de una nueva medida.	Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Medidas Adicionar • Cancelar

SC 3: Modificar Medidas

Tabla 6: Resultados al aplicar el caso de prueba: Gestionar Medida. SC4: Modificar Medidas

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 4.1: Se modifica una medida correctamente	El sistema modifica los datos de la medida seleccionada.	Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Medidas • Modificar
EC 4.2: Los datos entrados son incorrectos	El sistema muestra un mensaje con todos los errores y no se realiza la modificación de los datos.	Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Medidas • Modificar
EC 4.3: Se cancela la modificación.	El sistema cancela la modificación de la medida que se había seleccionado	Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Medidas • Modificar • Cancelar

SC 4: Eliminar Medidas

Tabla 7: Resultados al aplicar el caso de prueba: Gestionar Medida. SC5: Eliminar Medidas

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 5.1: Se elimina una medida	Se elimina la medida satisfactoriamente	No Satisfactorio.	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Medidas • Eliminar • Aceptar
EC 5.2: No se elimina	No se llega a eliminar, ha sido cancelada	Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Medidas • Eliminar • Cancelar

3.2.2 Caso de prueba Gestionar Compatibilizaciones.

SC 1: Buscar Compatibilizaciones

Tabla 8: Resultados al aplicar el caso de prueba: Gestionar Compatibilizaciones SC2: Buscar Compatibilizaciones.

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 2.1: Se marca la opción buscar	El sistema muestra el listado con todos las compatibilizaciones existente en el sistema.	Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Compatibilizaciones • Buscar
EC 2.2: Se introduce el texto por el cual se desea buscar una obligación.	El sistema debe de mostrar una lista con todas las compatibilizaciones que tienen en alguno de sus campos el texto igual al entrada.	Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Compatibilizaciones • Buscar
EC 2.3 : Los datos entrados son incorrectos	El sistema no muestra nada en el resultado de la búsqueda.	Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Compatibilizaciones • Buscar
EC 2.4: Generar documento Word.	El sistema genera un documento Word con los resultados de la búsqueda previamente realizada.	Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Compatibilizaciones • Buscar

EC 2.5: Generar documento PDF.	El sistema genera un documento PDF con los resultados de la búsqueda previamente realizada.	Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Compatibilizaciones • Buscar
-----------------------------------	---	---------------	--

SC 2: Insertar Compatibilizaciones

Tabla 9: Resultados al aplicar el caso de prueba: Gestionar Compatibilizaciones SC3: Insertar Compatibilizaciones.

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 3.1: Se adiciona una compatibilizaciones correctamente	El sistema inserta los datos de las compatibilizaciones.	Satisfactorio.	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Compatibilización • Adicionar
EC 3.2: Los datos entrados son incorrectos.	El sistema muestra un mensaje con todos los errores y no se realiza la inserción de los datos.	No Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Compatibilización
EC 3.3: Se cancela la adicción.	Se cancela la adicción de los datos.	Satisfactoria	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Compatibilización • Adicionar

SC 3: Modificar Compatibilizaciones

Tabla 10: Resultados al aplicar el caso de prueba: Gestionar Compatibilizaciones SC4: Modificar Compatibilizaciones.

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 4.1: Se modifica una Compatibilización correctamente.	El sistema modifica los datos de las compatibilizaciones.	Satisfactorio.	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Compatibilización • Modificar
EC 4.2: Los datos no son correctos	El sistema muestra un mensaje especificando el error, y no se realiza la modificación de los datos.	No Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Compatibilización • Modificar
EC 4.3: Se cancela la modificación	Se cancela la modificación de los datos	Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Compatibilización • Modificar • Cancelar

SC 4: Eliminar Compatibilizaciones

Tabla 11: Resultados al aplicar el caso de prueba: Gestionar Compatibilizaciones SC5: Eliminar Compatibilizaciones.

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 5.1: Se elimina una Compatibilización.	El sistema elimina una compatibilización	Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Compatibilización • Eliminar • Aceptar
EC 5.2: No se elimina	El sistema no elimina una compatibilización	Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Compatibilización • Eliminar • Cancelar

3.2.3 Caso de prueba Gestionar Obligaciones.

SC 1: Buscar Obligaciones

Tabla 12: Resultados al aplicar el caso de prueba: Gestionar Obligaciones SC2: Buscar Obligaciones.

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 2.1: Se marca la opción buscar	El sistema muestra el listado con todos los obligaciones existente en el sistema.	Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Obligaciones • Buscar

<p>EC 2.2: Se introduce el texto por el cual se desea buscar una obligación.</p>	<p>El sistema debe de mostrar una lista con todas las obligaciones que tienen en alguno de sus campos el texto igual al entrado.</p>	<p>No Satisfactorio</p>	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Obligaciones • Buscar
<p>EC 2.3 : Los datos entrados son incorrectos</p>	<p>El sistema no muestra nada en el resultado de la búsqueda.</p>	<p>Satisfactorio</p>	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Obligaciones • Buscar
<p>EC 2.4: Generar documento Word.</p>	<p>El sistema genera un documento Word con los resultados de la búsqueda previamente realizada.</p>	<p>Satisfactorio</p>	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Obligaciones • Buscar • Generar Word
<p>EC 2.5: Generar documento PDF.</p>	<p>El sistema genera un documento PDF con los resultados de la búsqueda previamente realizada.</p>	<p>Satisfactorio</p>	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Obligaciones • Buscar • Generar PDF

SC 2: Insertar Obligaciones

Tabla 13: Resultados al aplicar el caso de prueba: Gestionar Obligaciones SC3: Insertar Obligaciones

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 3.1: Se adiciona una obligación correctamente	El sistema inserta los datos de la Obligación.	Satisfactorio.	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Obligaciones • Adicionar
EC 3.2: Los datos entrados son incorrectos	El sistema muestra un mensaje especificando el error, y no se realiza la inserción de los datos.	No Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Obligaciones
EC 3.3: Se cancela la inserción.	El sistema cancela la inserción de una nueva obligación.	No Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Obligaciones • Adicionar • Cancelar

SC 3: Modificar Obligaciones

Tabla 14: Resultados al aplicar el caso de prueba: Gestionar Obligaciones SC4: Modificar Obligaciones.

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 4.1: Se modifica una obligación correctamente	El sistema modifica los datos de la Obligación seleccionada.	Satisfactorio.	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Obligaciones • Modificar
EC 4.2 : Los datos entrados son incorrectos	El sistema muestra un mensaje especificando el error, y no se realiza la modificación de los datos.	No satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Obligaciones • Modificar
EC 4.3: Se cancela la modificación.	El sistema cancela la modificación de la obligación que se había seleccionado	No satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Obligaciones • Modificar • Cancelar

SC 4: Eliminar Obligaciones

Tabla 15: Resultados al aplicar el caso de prueba: Gestionar Obligaciones SC5: Eliminar Obligaciones.

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 5.1: Se elimina una obligación	Se elimina la obligación satisfactoriamente	No satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Obligaciones • Eliminar • Aceptar
EC 5.2: No se elimina	No se llega a eliminar, ha sido cancelada	Satisfactorio	<ul style="list-style-type: none"> • Principal • Registro Minero • Gestión • Obligaciones • Eliminar

3.3 Resultados Generales:

En el proceso de ejecución de las pruebas, de forma general, se encontraron errores en la mayoría de los Casos de Uso del módulo Registro Minero. La detección de estos errores en este momento, posibilitará la corrección a tiempo por parte del equipo de desarrollo. Los errores encontrados con mayor frecuencia fueron los siguientes:

- El usuario no necesita iniciar sesión en el sistema para poder realizar las acciones de buscar, adicionar, modificar y eliminar.
- No se muestran los mensajes de confirmación luego de realizar acciones como eliminar. Ejemplo: cuando se realiza el caso de prueba Gestionar Obligaciones, la acción del escenario Eliminar una obligación, este no muestra el mensaje de que se ha eliminado satisfactoriamente.
- Existen algunos botones que están mal implementados, como es el caso del escenario Buscar, que aún no ejecuta su función como debe ser.
- La mayoría de los campos donde se debe de entrar datos, no están validados. Permite poner números en campos que son sólo para letras y viceversa.
- Se detectaron errores en la revisión del artefacto Especificación de Requisitos, mediante la lista de chequeo que se seleccionó para inspeccionarle.
- Se evidenció claramente los serios problemas que existen en cuanto a la descripción de los casos de uso. Se pudo ver durante el desarrollo de la planificación y ejecución de las pruebas, que hay funcionalidades que no están descritas previamente, sin embargo están implementadas, o por el contrario, algunas que están descritas y no están desarrolladas.
- De los 17 casos de pruebas que se diseñaron, 14 fueron de caja negra y 3 de Caja Blanca, de los cuales 5 fueron satisfactorios en la primera iteración de las pruebas al módulo, y 12 insatisfactorios
- De los 14 casos de prueba de caja negra que se aplicaron, 5 de ellos eran críticos, los cuales presentaban el mismo error, adicionaban y modificaban incorrectamente los datos entrados.

- En total se detectaron 27 errores en todos los casos de prueba que se aplicaron, incluyendo las 4 no conformidades encontradas en la revisión del documento Especificación de Requisitos.

Se llevó a cabo una segunda iteración de las pruebas al módulo, con el objetivo de verificar que el equipo de desarrollo le había dado solución a los errores detectados. Hay que tener en cuenta que al corregir los errores anteriores, existe la posibilidad de que se generen otros nuevos, por lo tanto siempre hay que verificar cada iteración del módulo.

Al realizarse la tercera iteración de las pruebas, se comprobó que ya el módulo ha corregido el 100% de los errores detectados en las anteriores iteraciones de las pruebas al software.

3.4 Validación de expertos. Método Delphi.

En la presente investigación, se ha señalado como uno de los objetivos, validar de forma externa los resultados alcanzados al aplicar los casos de prueba al módulo Registro Minero. Para ello se utilizó el método Delphi, el cual consiste en la utilización sistemática del juicio intuitivo de un grupo de expertos para obtener un consenso de opiniones. Resulta imprescindible que estas opiniones no se encuentren filtradas o influenciadas por criterios de algunos expertos. Este método resulta más efectivo si se garantiza: el anonimato, la retroalimentación controlada y la respuesta estadística de grupo.

Según [17] las **Ventajas del método Delphi** son las siguientes:

- Permite la formación de un criterio con mayor grado de objetividad.
- La toma de decisiones, sobre la base de los criterios de expertos, obtenido por éste, tiene altas probabilidades de ser eficiente.
- Evita conflictos entre expertos al ser anónimo, (lo que constituye un requisito imprescindible para garantizar el éxito del método) y crea un clima favorable a la creatividad.

- El experto se siente involucrado plenamente en la solución del problema y facilita su implantación. De ello es importante el principio de voluntariedad del experto en participar en la investigación.
- Garantiza libertad de opiniones (por ser anónimo y confidencial). Ningún experto debe conocer que a su igual se le está solicitando opiniones.

¿Cuáles son sus desventajas?

Sus desventajas más significativas están dadas en que:

- Es muy laborioso y demanda tiempo su aplicación, debido a que se requiere como mínimo de dos vueltas para obtener el consenso necesario.
- Es costoso en comparación con otros, ya que requiere del empleo de: tiempo de los expertos, hojas, impresoras, teléfono y correo.
- Precisa de buenas comunicaciones para economizar tiempo de búsqueda y recepción de respuestas.
- Debe ser llevado a cabo por un grupo de análisis: los expertos como tales.

De aquí la necesidad de aplicar varias vueltas, buscar técnicas variadas de análisis para obtener un consenso y pruebas estadísticas para determinar su grado de confiabilidad y pertinencia.

¿Cómo se seleccionan los expertos y qué instrumentos se aplican?

Para la aplicación práctica del método, es necesario considerar metodológicamente dos aspectos fundamentales: selección del grupo de expertos a encuestar y la elaboración del cuestionario o los cuestionarios. Pero ante todo: ¿A quiénes se pueden considerar expertos?

Se define como experto el individuo en sí, grupo de personas u organizaciones capaces de ofrecer valoraciones conclusivas de un problema en cuestión y hacer recomendaciones respecto a sus momentos fundamentales con un máximo de competencia.

Como requisito básico para la selección de un experto, se debe de tener en cuenta que éste tenga experiencia en el tema a consultar, dado por sus años de trabajo y que puedan ser complementados con: conocimientos teóricos adquiridos a través de las distintas formas de superación y grado académico o científico alcanzado en relación al tema, entre otros.

Para la selección de los expertos que responderían las preguntas en la presente investigación, se tuvo en cuenta una serie de aspectos sumamente importantes en la persona que tendrá como tarea emitir un criterio que permita probar si el trabajo realizado fue satisfactorio. Estos deben caracterizarse por su responsabilidad, integridad, honestidad, discreción y confiabilidad.

Luego de haber realizado la encuesta (anexo # 4), esta arrojó los siguientes resultados:

El 85.7% de los encuestados, manifiestan una calificación de 5 puntos para la funcionalidad de las pruebas realizadas y la eficiencia en la aplicación de las mismas. Los artefactos generados fueron aceptados totalmente por todos los especialistas, por su gran utilidad como resultado del proceso de pruebas.

3.5 Conclusiones Parciales

Al terminar este capítulo, se puede señalar que el proceso de pruebas se llevó a cabo en el tiempo planificado, siguiendo las disposiciones del Plan de Prueba confeccionado. Los casos de prueba desarrollados, mostraron muchos de los errores del sistema, lo que evidencia la eficiencia de los mismos.

Es importante resaltar que los resultados obtenidos no fueron utilizados para demeritar la labor realizada por los implementadores del módulo. Este es el momento de poner en práctica el objetivo principal de las pruebas de software, que es corregir los errores del sistema antes de que este llegue a los clientes.

Conclusiones Generales

Durante el desarrollo de la investigación, se realizó un estudio de los principales aspectos referentes al rol de diseñador de casos de prueba, se analizaron diseños realizados por especialistas con experiencia en el rol y se efectuó estudio del estado del arte del objeto de estudio. Estas tareas permitieron caracterizar la aplicación sobre la que se va a incidir, además de identificar la mejor forma de realizar las pruebas al software con el objetivo de obtener resultados con la mayor calidad posible.

Luego del trabajo realizado, al diseñar, implementar y aplicar las pruebas de software al módulo Registro Minero, se obtuvieron resultados positivos. Se adquirió un amplio conocimiento del proceso de pruebas existente, así como del Plan de Pruebas que se realiza para su elaboración, la estrategia que se traza para su aplicación y los pasos que hay que seguir para que todo este proceso se realice de forma correcta.

Se diseñaron y aplicaron 17 casos de prueba y a raíz de esto se realizó un documento de No Conformidades donde quedaron registrados los errores encontrados en el módulo. Como resultado de las pruebas realizadas, se generaron artefactos que documentan debidamente las faltas halladas, lo que permitirá facilitar y agilizar el proceso de corrección y perfeccionamiento del sistema.

Dada la cantidad de errores detectados, el tiempo de realización de las pruebas y los artefactos generados, se puede señalar que las pruebas tuvieron resultados satisfactorios. El diseño de las mismas se realizó de manera correcta y puede ser utilizado en otros sistemas similares. Con la entrega de los resultados al equipo de desarrolladores, culmina el cumplimiento de los objetivos propuestos, ya que el sistema será liberado sin errores y con una buena calidad.

Recomendación

Una vez concluido el proceso de pruebas y consciente de la importancia que este proceso representa dentro del ciclo de vida de una aplicación se recomienda:

- Realizarle otras pruebas al sistema como son las pruebas de stress y de carga.
- Fomentar el uso de la documentación en todo el ciclo de vida del proyecto, para facilitar otros procesos que garantizan una mayor calidad del software, como es el caso del proceso de pruebas.

Bibliografía

Referencias bibliográficas

1. Las pruebas de calidad de software, una asignatura pendiente en las empresas españolas. 8 de septiembre del 2008 [cited; Available from: <http://www.noticias.com/tecnologias-empresa-espanolas-tecnologia-universidad-tendencias-tic.1702>. 1
2. López, Y.T., Diseño y aplicación de las pruebas al producto Primicia, plataforma de televisión informativa. 2009, Universidad de las Ciencias Informáticas: La Habana. 2
3. Marco Legal Minero. 2009 [cited; Available from: http://www.upme.gov.co/guia_ambiental/carbon/gestion/politica/marco/marco.htm#6.REGISTROMINERO. 3
4. Ley de Minas...1977, 23 de Enero de 1995 se publicó en la Gaceta Oficial
5. Diccionario de la Lengua Española.[cited; Available from: <http://www.wordreference.com/definicion>. 5
6. *Pruebas de Software*. 2009 [cited; Available from: http://es.wikipedia.org/wiki/Pruebas_de_software.
7. *Pruebas de Software*. 2009 [cited; Available from: <http://Lsi.ugr.es/pruebas+de+software>
8. IEEE Standard Glossary of Software Engineering Terminology. 1990
9. IEEE, Std. 610. 1990.
10. ISO 8402. 1994.
11. Pressman, R. Ingeniería del Software, un enfoque practico. 2002 [cited.
12. Ronquillo, D.E., *ARQUITECTURA DE SOFTWARE*. 14/05/2009
13. Metodologías de desarrollo del Software. 2009 [cited; Available from: <http://www.scribd.com/doc/2050925/metodologias-de-desarrollo-software>.
14. Ivar Jacobson, G.B., James Rumbaugh, El Proceso unificado de Desarrollo de Software.
15. Teoría 14. Proceso Unificado. 2009 [cited; Available from: http://dc.exa.unrc.edu.ar/nuevodc/materias/sistemas/2008/Teoricos/TEORIA_14_PU_Implementacion_Prueba-en%204.pdf. 15

16. UNIVERSIDAD POLITÉCNICA DE MADRID, F.D.I. PLAN MAESTRO DE PRUEBAS, Ingeniería en computación e informática, NSGT- NUEVO SISTEMA DE GESTIÓN DE TITULACIÓN. [Cited; Available from: <http://www ldc.usb.ve/~teruel/ci4713/clases2001/planPruebas.html>].
- 17 Ing. Arabel Moráguez Iglesias, El método Delphi 05-2006 [cited; Available from: <http://www.gestiopolis.com>]

Bibliografía consultada

- Construyendo software de calidad. 2009 [cited; Available from: http://www.elguille.info/colabora/NET2005/Percynet_ConstruyendoSoftCalidad.htm].
- Enríquez, A.M.B., Proceso Metodológico de Auditoría Informática aplicado a la evaluación y seguimiento de Sistemas de Gestión desarrollados con el estándar de modelado UML, Tesis de Maestría en Ingeniería Informática. 2002, Universidad de Oriente La Habana Cuba – Universidad Autónoma Tomás Frías, Potosí-Bolivia.
- Equipoteccelaya, Importancia de la calidad del software. 3 de febrero 2009.
- Fernández, V.M., ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE. 20/03/2009.
- Fernández., V.M., MODELO DE SISTEMA. 20/03/2009.
- Granda, I.M.F., La importancia de las pruebas de usabilidad. 9 de enero del 2008.
- [3, 12-18]Prado, E.R. Casi todas las pruebas del software. 2007 [cited; Available from: <http://www.sistedes.es/sistedes/pdf/2007/pris-07-raja-ctps.pdf>].
- PRUEBA DE COMPONENTES DE SOFTWARE BASADAS EN EL MODELO DE JAVABEANS. . ABRIL DE 2001. [cited; UNIVERSIDAD AUTÓNOMA DE TLAXCALA DEPARTAMENTO DE INGENIERÍA Y TECNOLOGÍA. UNIDAD DE ESTUDIOS DE POSGRADO]. Available from: <http://www.cs.man.ac.uk/~velascop/publ/Tesis.pdf>.
- Sanz, L.f., Un sondeo Sobre la practica actual de pruebas de software en España, Depto. de Sistemas Informáticos, Universidad Europea de Madrid. España.
- Servicios de pruebas funcionales, Gestión de Prueba. 2009 [cited; Available from: <http://es.testhouse.net/index.php/servicios/pruebas-funcionales/gestion-de-pruebas>].
- Software, I.D.d.i.d. Pruebas de Software. 21 de setiembre de 2005 [cited; Available from: <http://athenea.ort.edu.uy/publicaciones/ingsoft/ortsf/seminarios05/prueba.htm>].
- Universidad del Valle, c. Técnicas de Pruebas de Software. 27 de agosto del 2007 [cited; Available from: <http://eisc.univalle.edu.co/materias/TPS/objetivos.html>].

- Uruguay, U.d. Pruebas de Software. 2009 [cited; Available from: <http://www.ort.edu.uy/index.php?q=pruebas+de+software>].

Glosario de términos

Release: Es una versión de un producto, preparado para publicarse como versión definitiva a menos que aparezcan errores que lo impidan.

Prueba de stress: Identifican problemas con el sistema cuando hay recursos insuficientes o cuando hay competencia por los recursos.

Prueba alfa: Es la primera versión del programa, la cual es enviada a los verificadores para probarla. Se utilizan el término alfa informalmente para referirse a una fase donde un producto todavía es inestable.

Prueba beta: Representa generalmente la primera versión completa del programa informático o de otro producto, que es posible que sea inestable pero útil para que las demostraciones internas y las inspecciones previas seleccionen a clientes.

Anexos

Anexo # 1 Foto del sistema, Caso de Uso Gestionar Medidas.

The screenshot shows a web browser window with the URL 'Conectando la ciudad' and the application 'SGDG'. The page title is 'Sistema de Gestión de Datos Geológicos'. The date 'Lunes, 8 de Marzo del 2010' is displayed. The breadcrumb trail is 'Inicio > Concesionario Mineral > Pagina Principal'. The left sidebar contains a menu with 'Gestión' expanded, showing options like 'Solicitante', 'Solicitud de Concesión', 'Solicitud Derecho Minero', 'Compatibilización', 'Obligación', 'Dictámenes', 'Medidas', and 'Resolución o Acuerdos'. Below this are 'Consultas' and 'Reportes'. The main content area is titled 'Gestionar Medidas' and has two tabs: 'Buscar' and 'Adicionar'. The 'Adicionar' tab is active, showing a form for 'Datos de la nueva Medida'. The form fields are: 'Derecho Minero' (dropdown menu with 'Derecho Minero' selected), 'Fecha de Imposición:' (calendar icon, value '8-03-2010'), 'Fecha Límite:' (calendar icon, value '8-03-2011'), 'Medida Aplicada:' (text input field), and 'Observaciones:' (text area). At the bottom of the form are 'Registrar' and 'X' buttons.

Anexo # 2 Foto del sistema, Caso de Uso Gestionar Obligaciones.

The screenshot shows a web browser window with the URL 'Conectando la ciudad' and the application 'SGDG'. The page title is 'Sistema de Gestión de Datos Geológicos'. The date 'Lunes, 8 de Marzo del 2010' is displayed. The breadcrumb trail is 'Inicio > Concesionario Mineral > Pagina Principal'. The left sidebar contains a menu with 'Gestión' expanded, showing options like 'Solicitante', 'Solicitud de Concesión', 'Solicitud Derecho Minero', 'Compatibilización', 'Obligación', 'Dictámenes', 'Medidas', and 'Resolución o Acuerdos'. Below this are 'Consultas' and 'Reportes'. The main content area is titled 'Gestionar Obligaciones' and has two tabs: 'Buscar' and 'Adicionar'. The 'Adicionar' tab is active, showing a form for 'Datos de la nueva Obligación'. The form fields are: 'Derecho Minero:' (dropdown menu with '---Seleccione---' selected), 'Tipo de Obligación:' (dropdown menu with '---Seleccionar Derecho---' selected), 'Fecha de Presentación:' (calendar icon, value '8-03-2010'), 'Fecha Cumplimiento:' (calendar icon, value '8-03-2011'), 'Frecuencia de la Obligación:' (text input field), and 'Observaciones:' (text area). At the bottom of the form are 'Registrar' and 'X' buttons.

Anexo # 3 Foto del sistema, Caso de Uso Gestionar Compatibilización.

The screenshot shows a web browser window with the SGD G logo and the slogan "El conocimiento geológico de Cuba al servicio de la sociedad". The date "Lunes, 8 de Marzo del 2010" is displayed. The breadcrumb trail reads "Usted está en: Inicio > Concesionario Mineral > Pagina Principal". The main content area is titled "Gestionar Compatibilización" and features a sidebar menu with options like "Gestión", "Solicitante", "Solicitud de Concesión", "Solicitud Derecho Minero", "Compatibilización", "Obligación", "Dictámenes", "Medidas", "Resolución o Acuerdos", "Consultas", and "Reportes". The "Compatibilización" option is selected. The main form includes fields for "Organismo", "Fecha de Envío" (8-03-2010), "Solicitud" (Seleccione), and "Fecha Límite" (8-04-2010), along with an "Observaciones" text area and a "Registrar" button.

Anexo # 4 Encuesta de validación de la investigación. Método Delphi.

Encuesta para la validación de la aplicación de los casos de prueba al módulo Registro Minero

Datos del Especialista.

Categoría Científica:

Centro Laboral:

Cargo que desempeña:

En las siguientes preguntas, otorgue calificaciones en base a 5.

Calificación

Funcionalidad de las pruebas realizadas.....

Eficiencia en la aplicación de las pruebas.....

Utilidad de los artefactos generados.....

En las siguientes preguntas otorgue la calificación marcando con una X la respuesta que usted considera adecuada.

SI

NO

¿En su opinión, las Pruebas de Software tienen gran importancia dentro del ciclo de vida de un software?

¿Considera usted que los casos de prueba realizados permitirán entregar un producto libre de defectos?

¿Cree que el proceso de pruebas debería llevarse a cabo en los demás proyectos de la facultad o la universidad?

