

Universidad de las Ciencias Informáticas



Facultad # 9



Sistema de Gestión de Datos Geológicos. Módulo Inventario de Petróleo y Gas.

Rol Diseñador de Casos de Prueba.

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autor: Tony Moisés Amarán Fernández

Tutor: Ing. Dayris Espinosa Ronquillo

La Habana, Mayo 2010

“Año 52 de la Revolución”

DECLARACIÓN DE AUTORÍA

Yo: **Tony Moisés Amarán Fernández** declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) a hacer uso del mismo en su beneficio.

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

Firma del Autor

Firma del Tutor

AGRADECIMIENTOS

Primeramente agradezco a mi tutora Dayris Espinosa, la cual me guió desde el primer momento de este trabajo de diploma. También a mi tribunal, Gerdys, Héctor, Lisset, Odalys, Dianet y Yudiel, gracias por todas las recomendaciones y revisiones que posibilitaron que la investigación quedara con mayor calidad.

A mi familia, la cual es la razón principal por la que me he esforzado tanto a lo largo de estos años. A mi madre Mariana Fernández, a la cual le estaré eternamente agradecido por todo su amor y valor de seguir adelante cada día de esta vida para ver a sus hijos crecer; a mi padre Rafael Moisés, sus consejos los recordaré por siempre; a mi hermana, la más linda de todas; a mis abuelos, el amor de los abuelos no se puede resumir en tan pocas líneas, se necesita mucho más que eso, gracias por todo; a mis tíos, los segundos padres; a mis primos, los segundos hermanos.

Agradezco también a Julio (el teacher), un gran amigo, por todo su apoyo a lo largo de más de 15 años de amistad y en especial en los 5 años de mi carrera y a su esposa Yarielys, gracias a los dos.

A mi novia Yanet por todo su amor, apoyo y comprensión a lo largo de 2 años de una hermosa relación, por su amor incondicional y su respeto, haciendo de mi una mejor persona cada día.

A todos mis profesores, por enseñarme todo lo que se durante mis estudios, en especial a mi profesora Rosa Elena (Chelena) profesora de matemática en la EMCC de Pinar del Río, la cual siempre confió en que podría conseguir este título. “El que persevera triunfa”.

A mis amigos del proyecto SGDG y profesores por tanto apoyo durante la mayor parte de mi carrera, en especial a Vladimir, Joel, Alberto, Reinaldo, Yamile, Yamil, Yudiel, Rocny, Rubén, Eddy y Liester.

A mis amigos Dany y Samy, por ser los causantes de tan buenos recuerdos durante mi estancia en los camilitos. A todos mis amigos durante mi estancia en la Universidad, mis compañeros de aula, a todos, en especial a Alain, Eyonys y Adrian, sin los cuales no se pudiera conocer el verdadero significado de la amistad. A mi hermano Armando “Mandy”. A todos mis amigos en mi barrio, en especial a Yosniel, el Niñito, el Chino y Edel, amigos por siempre.

A Fidel.

DEDICATORIA

A mi madre. Gracias por existir. Este título es para ti. Todos mis logros son tuyos.

A mi padre. Gracias por todos tus consejos y apoyo.

A mi hermana. Sigue así de linda e inteligente.

A mis abuelos. Muchas gracias por todo lo que han hecho por mí durante toda mi vida, los amo.

A mi familia. Ustedes son la razón por la cual he llegado tan lejos en la vida.

A mis amigos de toda la vida.

RESUMEN

Los errores en el proceso de desarrollo de un software son muy comunes desde el inicio del ciclo de vida del mismo. La posibilidad de que se cometan fallos en su realización es muy elevada desde el momento en que se levantan los requerimientos técnicos del software, hasta los posteriores pasos de análisis y diseño llegando a la implementación. Debido a esto se hace necesario que se verifique la calidad del software una vez que el equipo de desarrollo haya concluido con la implementación.

El presente trabajo tiene entre sus principales objetivos desarrollar las pruebas al módulo Inventario de Petróleo y Gas de la Oficina Nacional de Recursos Minerales (ONRM) que está siendo automatizado mediante el proyecto Sistema de Gestión de Datos Geológicos (SGDG) perteneciente a la Facultad 9 de la Universidad de las Ciencias Informáticas, primera universidad creada al calor de la Batalla de Ideas. La aplicación de un proceso de pruebas a este módulo, permitirá encontrar errores que no han sido detectados por los desarrolladores y que son de vital importancia corregirlos para que la aplicación llegue a manos de los clientes con la mayor calidad posible.

Palabras Clave: Calidad, ONRM, Pruebas, SGDG

Tabla de contenido

Introducción	1
CAPÍTULO 1	4
FUNDAMENTACIÓN TEÓRICA	4
1.1 Introducción.....	4
1.2 ¿Qué es la calidad del software?	4
1.3 Modelos de calidad.....	4
1.3.1 Norma ISO 9000-2000.....	5
1.3.2 Modelo de Capacidad y Madurez (CMM según sus siglas en inglés).....	5
1.3.3 ISO/IEC TR 15504.....	6
1.4 Metodologías de desarrollo de software.....	7
1.4.1 ¿Qué es RUP?	7
1.4.2 Desarrollar software iterativamente	9
1.4.3 Administrar los requerimientos	9
1.4.4 Utilizar arquitecturas basadas en componentes.....	9
1.4.5 Verificar la calidad del software	10
1.5 Introducción a las pruebas de software.....	10
1.6 Pruebas según RUP.....	11
1.7 Rol de Ingeniero de Pruebas o Diseñador de Casos de Prueba	12
1.8 Principios de las pruebas de software.....	12
1.9 Características de una buena prueba	13
1.10 Diseño de Casos de Prueba	13
1.11 Prueba de caja blanca.....	14
1.12 Técnicas de prueba de caja blanca.....	15
1.12.1 Prueba de camino básico.....	15
1.12.2 Prueba de condición	15
1.12.3 Prueba del flujo de datos	16
1.12.4 Prueba de bucles.....	16
1.13 Prueba de caja negra	16
1.14 Técnicas de prueba de caja negra	17
1.14.1 Partición equivalente.....	17
1.14.2 Análisis de valores límites.....	18
1.14.3 Prueba de comparación.....	19
1.14.4 Prueba de tabla ortogonal.....	19
Conclusiones	20
CAPÍTULO 2	21
PLANIFICACIÓN DEL PROCESO DE PRUEBA	21
2. Introducción	21
2.1 Módulo Inventario de Petróleo y Gas	21

2.2 Características a probar.....	21
2.3 Plan de Prueba.....	22
2.3.1 Objetivos	22
2.3.2 Recursos para las pruebas	23
2.3.3 Requerimientos generales	23
2.3.4 Pasos a seguir en las pruebas o estrategia	23
2.3.5 Diseño de Casos de Prueba de caja negra.....	24
2.3.6 Diseño de Casos de Prueba de caja blanca	29
2.5 Manual de Usuario y Especificación de Requerimientos	37
2.5.1 Listas de Chequeo	37
Conclusiones.....	38
CAPÍTULO 3	39
RESULTADOS OBTENIDOS.....	39
3. Introducción	39
3.1 Resultados de la aplicación de las pruebas al módulo	39
3.1.1 Caso de Prueba Gestionar Prospecto.	39
3.1.2 Caso de Prueba Gestionar Bloque.....	41
3.1.3 Caso de Prueba Gestionar Datos Primarios de Prospecto.....	44
3.1.4 Caso de Prueba Gestionar Datos de Producción Mensual.	44
3.2 Errores en el documento Especificación de Requerimientos	45
3.3 Principales errores detectados en el proceso de pruebas	49
3.4 Validación de expertos. Método Delphi.	50
Conclusiones.....	51
CONCLUSIONES	53
RECOMENDACIONES	54
Bibliografía referenciada	55
Bibliografía consultada	56
GLOSARIO DE TÉRMINOS	58
ANEXOS	59

ÍNDICE DE TABLAS

Tabla 1: Diseño del CP del CU: Gestionar Prospecto	26
Tabla 2: Diseño del CP del CU: Gestionar Yacimiento.....	27
Tabla 3: Diseño del CP del CU: Gestionar Bloque	29
Tabla 4: Resultados del CP: Gestionar Prospecto. SC1: Mostrar Prospecto.....	39
Tabla 5: Resultados del CP: Gestionar Prospecto. SC2: Adicionar Prospecto	40
Tabla 6: Resultados del CP: Gestionar Prospecto. SC3: Modificar Prospecto	41
Tabla 7: Resultados del CP: Gestionar Prospecto. SC4: Eliminar Prospecto	41
Tabla 8: Resultados del CP: Gestionar Bloque. SC1: Mostrar Bloque.....	41
Tabla 9: Resultados del CP: Gestionar Bloque. SC2: Adicionar Bloque	42
Tabla 10: Resultados del CP: Gestionar Bloque. SC3: Modificar Bloque	43
Tabla 11: Resultados del CP: Gestionar Bloque. SC4: Eliminar Bloque.....	43
Tabla 12: Resultados del CP: Gestionar Datos Primarios de Prospecto. SC2: Gestionar Datos	44
Tabla 13: Resultados del CP: Gestionar Datos de Producción Mensual. SC2: Adicionar Datos.....	45
Tabla 14: Lista de Chequeo para el documento Especificación de Requerimientos.....	49
Tabla 15: Registro de defectos y dificultades detectados.....	68

ÍNDICE DE FIGURAS

Figura 1: Prueba de caja blanca	15
Figura 2: Prueba de caja negra.....	17
Figura 3: Grafo de flujo de datos para el método Buscar Prospecto.....	31
Figura 4: Grafo de flujo de datos para el método Modificar Yacimiento.....	33
Figura 5: Grafo de flujo de datos para el método Datos del Prospecto.....	36
Figura 6: Nombre incorrecto en Gestionar Prospecto	68
Figura 7: Nombre incorrecto en Gestionar Pozo Natural	68
Figura 8: Error número extenso. La interfaz colapsa	69
Figura 9: Mensaje distinto a los demás del sistema	69
Figura 10: Botón con formato y nombre distintos	70
Figura 11: No muestra los Yacimientos.....	70
Figura 12: Mensaje distinto a los demás del sistema	71

Introducción

El creciente desarrollo del software a nivel mundial y el aumento acelerado de las Tecnologías de la Información y las Comunicaciones (TIC) ha significado un cambio radical en la mayoría de las empresas del mundo, las cuales automatizan sus procesos y las tareas que se realizan en la misma. En la actualidad se considera que las TIC es el mayor impulsor de la economía a nivel mundial en el siglo actual.

La Oficina Nacional de Recursos Minerales (ONRM) evidencia de una forma muy concreta el desarrollo de las TIC en las empresas cubanas. Pertenece al Ministerio de la Industria Básica pero su esfera de influencia y acción se extiende a todos los órganos y organismos de la administración central del estado.

La ONRM es la entidad que vela por el aprovechamiento racional de los recursos minerales del país y constituye el órgano que controla el proceso concesionario, ordenando y fiscalizando la actividad geológica, minera y petrolera de la República de Cuba. (ONRM, 2008)

Actualmente está desarrollando el Programa Nacional de Informatización del Conocimiento Geológico (PNICG), con el objetivo de informatizar el conocimiento geológico existente en nuestro país. Un equipo de desarrollo de la Facultad 9 de la Universidad de las Ciencias Informáticas (UCI) automatiza estos procesos mediante un proyecto llamado Sistema de Gestión de Datos Geológicos (SGDG).

Situación Problemática

El proyecto SGDG cuenta con 8 módulos, entre los cuales se encuentra Inventario de Petróleo y Gas. Este es el encargado de controlar y almacenar toda la información correspondiente a los recursos de petróleo y gas con que cuenta el país en la actualidad. La información que en él se maneja contiene un alto valor económico, político y social. Dicho módulo acaba de ser implementado por parte del equipo de desarrollo y para poder desplegarlo se hace necesario verificar que el mismo se entregue libre de defectos a los usuarios finales.

Antes de liberar el producto se debe comprobar la correcta implementación y elaboración de los requisitos funcionales presentados por los clientes de la ONRM. Mediante esta comprobación se podrá lograr que el producto final quede con la calidad requerida, verificando si los requerimientos planteados por el cliente fueron cumplidos y documentando las fallas encontradas en este proceso. El objetivo principal de esta

comprobación es lograr que la calidad del módulo sea la mayor posible antes de ser entregado a los clientes.

Debido a esto se plantea el siguiente **Problema a resolver**: ¿Cómo lograr la entrega del módulo Inventario de Petróleo y Gas libre de defectos?

Dando paso así al siguiente **Objeto de Estudio**: El proceso de pruebas de sistemas de gestión de información.

Se define como **Objetivo General**: Elaborar la documentación técnica del proceso de pruebas correspondiente al módulo Inventario de Petróleo y Gas que forma parte del Sistema de Gestión de Datos Geológicos.

Enmarcando el **Campo de acción** en: El diseño e implementación de las pruebas del módulo Inventario de Petróleo y Gas.

Se plantea como **Idea a defender** de la investigación: El diseño y la implementación de las pruebas al módulo Inventario de Petróleo y Gas que forma parte del Sistema de Gestión de Datos Geológicos garantizarán que el mismo se entregue libre de defectos.

Tomando en cuenta lo antes mencionado, se presentan las siguientes **Tareas de la Investigación**:

- 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 Inventario de Petróleo y Gas.
- Diseñar los Casos de Prueba.
- Realizar las pruebas de caja negra al módulo Inventario de Petróleo y Gas.
- Realizar las pruebas de caja blanca al módulo Inventario de Petróleo y Gas.
- Caracterizar el estado de la implementación una vez finalizado el proceso de pruebas.
- Validar las pruebas realizadas.

Métodos científicos

En la primera etapa del estudio, se procedió a la revisión bibliográfica nacional e internacional, ordenando el conocimiento ya existente y satisfaciendo las necesidades de búsqueda, basado en los métodos teóricos a través del análisis de los principales documentos de interés. El método **Analítico - Sintético** permitió descomponer el tema, posibilitando un mejor estudio de todos los aspectos obtenidos para la elaboración de la investigación. El método **Inductivo – Deductivo**, facilitó interpretar los análisis de datos, así como descubrir las regularidades y relaciones entre los distintos componentes de la investigación para poder realizar las generalizaciones que parten por un lado de los análisis **Histórico – Lógico** a través del cual se profundizó en las tendencias, regularidades y cualidades que presenta el objeto de estudio.

Además al final de la investigación se utilizó dos métodos empíricos, el método **Encuesta** para darle validación externa a los resultados expuestos a través de una encuesta y el de **Observación** para comprobar si los requerimientos fueron completados satisfactoriamente.

La presente investigación está compuesta por 3 capítulos, de los cuales a continuación se brinda una breve descripción de lo que trata en particular cada uno de ellos.

Capítulo 1: FUNDAMENTACIÓN TEÓRICA. En este capítulo están sentadas las bases de la investigación, brindando conceptos y caracterizando el proceso de prueba según la metodología seleccionada.

Capítulo 2: PLANIFICACIÓN DEL PROCESO DE PRUEBA. En este capítulo se planificó el proceso de pruebas que se realizó. Se brindan elementos sobre el Plan de Pruebas y su objetivo, así como la realización de los Casos de Prueba de caja blanca y caja negra.

Capítulo 3: RESULTADOS OBTENIDOS. Al final de la investigación se exponen los resultados que arrojó el proceso de pruebas en el módulo y se validan las pruebas realizadas.

CAPÍTULO 1

FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

El capítulo 1 correspondiente a la presente investigación, caracteriza el proceso de pruebas según la metodología Proceso Unificado de Desarrollo (RUP por su nombre en inglés), además brinda una serie de características correspondientes al rol de Diseñador de Casos de Prueba según RUP.

A continuación se brindan una serie de conceptos los cuales están vinculados a la presente investigación para una mejor comprensión de la misma.

1.2 ¿Qué es la calidad del software?

La calidad del software está dada por el conjunto de cualidades que lo caracterizan, determinando su utilidad y existencia.

En la actualidad los equipos de desarrollo llevan a cabo grandes esfuerzos para realizar software que tengan la calidad requerida y que cumplan con los requerimientos y las funcionalidades que fueron planteadas por el cliente en un inicio.

¿Cómo se define la calidad del software?

“Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente.” (Pressman, 2005)

La misma va a ser sinónimo de corrección, confiabilidad, eficiencia, mantenibilidad, portabilidad, usabilidad, flexibilidad, así como la integridad y seguridad del producto.

1.3 Modelos de calidad

En la actualidad ha aumentado la complejidad con que se desarrollan los sistemas de software para la industria, pues se ha hecho difícil generar productos que cumplan fielmente con las expectativas de los clientes. Para corresponder a esta situación, han surgido una serie de herramientas y modelos, que

facilitan a las organizaciones, encargadas de las tecnologías de la información, generar productos que cumplan con las expectativas del cliente e incluso las sobrepasen y además prometen ser la solución a los problemas de calidad, costo y tiempos de desarrollo; se pueden mencionar modelos de calidad como la norma ISO 9000-2000, la ISO/IEC TR 15504 y el modelo CMM (Capability Maturity Model). (ALARCÓN, 2004)

A continuación se hace una breve descripción de algunos de estos modelos, con algunas de sus ventajas y desventajas:

1.3.1 Norma ISO 9000-2000

La Norma ISO 9000-2000 es la única edición de normas ISO 9000 que está en vigor a partir del 15 de diciembre de 2003. Es una norma internacional destinada a evaluar la capacidad de la organización para cumplir los requisitos del cliente, los reglamentarios y los propios de la organización. Está enfocada a la satisfacción de los clientes, a la mejora continua y a la integración de los Sistemas de Gestión de Calidad (SGC) en los procesos que se realizan en la organización. (ALARCÓN, 2004)

Ventajas

- Tiene un mecanismo de certificación bien establecido.
- Está disponible y es conocida.

Desventajas

- No es específica para la industria de software.
- No es fácil de entender.
- No está definida como un conjunto de procesos.
- No es fácil de aplicar.

1.3.2 Modelo de Capacidad y Madurez (CMM según sus siglas en inglés)

Es un modelo de calidad, que se utiliza para fomentar mejoras, medir y controlar los procesos de desarrollo de software. Mediante el mismo se clasifican las empresas en niveles de madurez. Estos niveles van a permitir conocer la madurez de los procesos que se realizan para producir software. Son 5 niveles: Inicial, Repetible, Definido, Cuantitativamente Gestionado, Optimizado. (ALARCÓN, 2004)

Ventajas

- Es específica para el desarrollo y mantenimiento de software.

- Es definida como un conjunto de áreas claves de procesos.
- Tiene un modelo de evaluación.
- Desde 1998 empezó a popularizarse en México.
- Existen organizaciones evaluadas.

Desventajas

- Es un modelo extranjero, no internacional.
- No es fácil de aplicar (pensado para organizaciones grandes).
- La mejora no está enfocada directamente a los objetivos de negocio.
- La evaluación es costosa y no tiene período de vigencia.
- Se está abandonando a favor de CMM-I.

1.3.3 ISO/IEC TR 15504

Define el modelo de referencia de procesos de software, así como las capacidades de procesos que constituyen la base para la evaluación de procesos de software. Este estándar permite analizar los resultados obtenidos de la evaluación contra los modelos de capacidad deseados. Está compuesto de 9 partes de las cuales la 2, 3 y 9 son normativas y las demás informativas. (ALARCÓN, 2004)

Ventajas

- Específico para el desarrollo y mantenimiento de software.
- Definido como un conjunto de procesos.
- Orientado a mejorar los procesos para contribuir a los objetivos del negocio.

Desventajas

- No es práctico ni fácil de aplicar.
- Tiene solamente lineamientos para un mecanismo de evaluación.
- Todavía no es norma internacional.

En la actualidad el desarrollo del software a nivel mundial, propiciado por el rápido crecimiento de las TIC, hace que las aplicaciones desarrolladas tengan un alto nivel de robustez y una mayor complejidad. Debido a esto la calidad del software desarrollado tiene que ser la mayor posible y para lograrla no se debe solamente tener en cuenta los modelos planteados, también se es necesario enfocarse en la realización de las pruebas al software que van a permitir detectar errores ocultos hasta el momento en la aplicación.

Una vez detectados se podrán documentar y corregir, logrando una aplicación con una mayor calidad y lista para ser entregada a los clientes.

1.4 Metodologías de desarrollo de software

Para lograr una mejor calidad en el producto o software que se está desarrollando, se debe realizar basándose en una metodología de desarrollo de software, debido a que las mismas definen un conjunto de criterios de desarrollo que guían la forma en la que se aplica la ingeniería de software.

El objetivo de la misma es controlar de manera transparente todo el proceso de desarrollo del software, así se podrá producir el mismo, en el tiempo esperado y con el costo que se tenía planificado desde un inicio. Este aspecto es fundamental, pues en ocasiones los proyectos se realizan sin ninguna metodología invirtiéndose más de lo planificado, atrasándose de forma general el proyecto y con ello la calidad del mismo se ve afectada en todo momento.

El proyecto SGDG se ha desarrollado basado en la metodología de desarrollo de software RUP. Esta es la metodología en la cual se centró la presente investigación para asegurar la calidad del módulo.

1.4.1 ¿Qué es RUP?

“RUP es un proceso de desarrollo de software que captura las mejores prácticas del conocimiento de líderes en Ingeniería de Software y proporciona a los equipos de desarrollo guías, estándares y recomendaciones para la construcción de software de alta calidad. Las mejores prácticas de desarrollo de software están documentadas como principios clave.” (IT Institute, 2009)

RUP provee un enfoque de ingeniería de software para una mejor asignación de tareas así como las responsabilidades dentro del equipo de desarrollo en que se encuentra el proyecto. El principal objetivo del mismo, es asegurarse que el proyecto termine con una alta calidad para que sea entregado al usuario final libre de defectos, logrando que sea capaz de satisfacer las necesidades planteadas por él anteriormente y que el tiempo de desarrollo del mismo sea factible.

Una de las ventajas que ofrece RUP es que aumenta la calidad del equipo de desarrollo, pues permite sin importar cuál sea la responsabilidad de cada individuo, entrar a la misma base de datos de conocimientos. Esto proporciona que cada cual pueda compartir el mismo lenguaje, la misma visión y que se centren en el mismo proceso acerca de cómo desarrollar el software.

RUP describe como aplicar efectivamente enfoques comprobados comercialmente para el desarrollo de software. Estos enfoques son llamados "mejores prácticas" pues los mismos son utilizados en la industria del software por distintas organizaciones de gran éxito en la actualidad. (Sanchez, 2007)

RUP divide en 4 fases el desarrollo del software:

- Inicio. El objetivo en esta etapa es determinar la visión del proyecto.
- Elaboración. En esta etapa el objetivo es determinar la arquitectura óptima.
- Construcción. En esta etapa el objetivo es llegar a obtener la capacidad operacional inicial.
- Transición. El objetivo es llegar a obtener el release¹ del proyecto.

Cada una de estas etapas se va desarrollando mediante un ciclo de iteraciones, el cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una de las iteraciones se establecen en función de la evaluación de las iteraciones precedentes. (Sanchez, 2007)

RUP va pasando por diferentes flujos de trabajos como son: Modelado de negocio, Requisitos, Análisis y Diseño, Implementación, Prueba, Despliegue y en 3 flujos de trabajo llamados de apoyo: Gestión de Configuración y Cambios, Gestión de Proyecto y Entorno de Desarrollo.

Características esenciales de RUP:

- Dirigido por los Casos de Uso
- Centrado en la arquitectura
- Iterativo e incremental

RUP provee a cada uno de los miembros del equipo las guías de proceso, mentores de herramientas y plantillas, evidenciándose así las mejores prácticas: (Metodologías - RUP, 2007)

- Desarrollar iterativamente
- Administrar requerimientos
- Arquitectura basada en componentes
- Modelizar visualmente

¹ Se refiere a liberar la versión de un producto software.

- Verificar calidad
- Controlar cambios

A continuación se dará una breve explicación de algunas de las prácticas que se consideran importantes para lograr un producto software con una mayor calidad.

1.4.2 Desarrollar software iterativamente

Debido a la complejidad actual de los nuevos productos que se necesitan a nivel mundial, ya no es posible trabajar de forma secuencial: definir primero el problema completo, posteriormente diseñar toda la solución, construir el software y finalmente, testear el producto. Para lograr una mayor calidad en el producto final se hace necesario otro tipo de enfoque, que sea capaz de detectar en forma temprana los desajustes en el software, detectando las inconsistencias en cada una de las etapas del desarrollo del mismo. A través de las iteraciones se hace posible refinar sucesivamente el software llegando a una solución efectiva luego de múltiples iteraciones acotadas en complejidad. (Metodologías - RUP, 2007)

1.4.3 Administrar los requerimientos

“Los requerimientos son las condiciones o capacidades que el sistema debe conformar. La Administración de Requerimientos es un enfoque sistemático para hallar, documentar, organizar y monitorear los requerimientos cambiantes de un sistema.” (Metodologías - RUP, 2007)

La Administración de Requerimientos permite:

- Que las comunicaciones estén basadas en requerimientos claramente definidos.
- Que los requerimientos puedan ser priorizados, filtrados y monitoreados.
- Que sea posible realizar evaluaciones objetivas de funcionalidad y rendimiento.
- Que las inconsistencias se detecten más fácilmente.

RUP describe como obtener, organizar y documentar la funcionalidad y restricciones requeridas, así como documentar y monitorear las alternativas y decisiones.

1.4.4 Utilizar arquitecturas basadas en componentes

RUP muestra la forma en que se puede llevar a cabo una arquitectura flexible, que sea capaz de acomodarse a los cambios en el desarrollo del software, que sea comprensible intuitivamente y además

promueva de forma más efectiva la reutilización del software. Soporta el desarrollo de software basado en componentes: módulos no triviales que completan una función clara. RUP provee un enfoque sistemático para definir una arquitectura utilizando componentes nuevos y preexistentes.

1.4.5 Verificar la calidad del software

Se hace necesario evaluar la calidad de todo software antes de ser liberado a los usuarios finales, respecto a diferentes aspectos, como son: sus requerimientos de funcionalidad, confiabilidad y rendimiento. La actividad esencial en este aspecto son las pruebas, que van a permitir encontrar las fallas en el software. RUP asiste en las diferentes etapas por la que pasa este proceso de pruebas, como son: planeamiento, diseño, implementación, ejecución y evaluación de todos los tipos de pruebas.

“El aseguramiento de la calidad se construye dentro del proceso, en todas las actividades, involucrando a todos los participantes, utilizando medidas y criterios objetivos, permitiendo así detectar e identificar los defectos en forma temprana.” (Metodologías - RUP, 2007)

1.5 Introducción a las pruebas de software

“Las pruebas del software es un elemento crítico para la garantía de calidad del software...” (Pressman, 2005). Una vez que se ha concluido con la implementación se deben de realizar las pruebas al software, para descubrir y corregir al máximo los errores presentes en el mismo, antes de la entrega al cliente. El Ingeniero de Pruebas es el encargado de crear una serie de Casos de Prueba y pretenderá con ellos destruir por completo el software, de esta forma será capaz de encontrar errores, los cuales podrán ser corregidos por el equipo de desarrollo.

El proceso de pruebas al software debe de ser constante a lo largo del ciclo de vida del mismo, así se logrará obtener un producto de alta calidad para ser entregado al usuario final, cumpliendo con sus necesidades en todo momento. El objetivo de esta etapa de pruebas es lograr que el producto elaborado salga con la mayor calidad posible. Las pruebas no necesariamente tienen que verse reflejadas al final de la implementación del proyecto, sino que desde un inicio del ciclo de vida del mismo, se pueden ir realizando las mismas. Las pruebas pueden ser enfocadas en dos aspectos, en la lógica interna del software y en las funcionalidades externas.

Para la realización de las pruebas se establecen objetivos que se han de cumplir en aras de que las pruebas sean realizadas de forma satisfactoria, entre ellos se encuentran: Una prueba tiene éxito si descubre un error no detectado hasta entonces; La prueba es el proceso de ejecución de un programa con la intención de descubrir un error; Mediante la realización de las pruebas no se garantiza la ausencia de defectos, sino que se puede demostrar que existen defectos en el software; Un buen Caso de Prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces. (Pressman, 2005)

1.6 Pruebas según RUP

El flujo de trabajo Prueba según RUP es el que se encarga de evaluar la calidad del producto que se está desarrollando por parte del equipo de desarrollo. RUP plantea que el mismo no se realiza para rechazar o aceptar el producto al final del proceso, sino que la misma debe de ir integrada al proyecto durante todo el ciclo de vida. (Int. RUP, 2005)

Los objetivos de este flujo de trabajo son:

- Encontrar y documentar defectos en la calidad del software.
- Generalmente asesora sobre la calidad del software percibida.
- Provee la validación de los supuestos realizados en el diseño y especificación de requisitos por medio de demostraciones concretas.
- Verificar las funciones del producto de software según lo diseñado.
- Verificar que los requisitos tengan su apropiada implementación.

Con la realización del Plan de Pruebas es que se comienzan las actividades en este flujo de trabajo y va a ser clave para la realización de las pruebas con éxito. Este documento contiene los objetivos generales y específicos de las pruebas, las estrategias y los diferentes recursos para poder proceder a la realización de las mismas.

En este flujo de trabajo se pasa por diferentes etapas, primeramente se planifica que es lo que hay que probar, después se diseña cómo se va a hacer, se implementa lo necesario para llevarlos a cabo, se ejecutan en los niveles necesarios y en un final se obtienen los resultados, de forma que la información obtenida sirva para ir refinando el producto a desarrollar. (Int. RUP, 2005)

1.7 Rol de Ingeniero de Pruebas o Diseñador de Casos de Prueba

Debido a la necesidad de lograr que los software tengan la mayor calidad posible, los equipos de desarrollo se ven obligados a contar con un grupo que esté especializado en pruebas de calidad, para que esté asegurada en todo momento la calidad del software y que llegue al usuario final con la calidad requerida y con el menor número de errores posibles. El responsable de la realización de las pruebas en un equipo de desarrollo es el Ingeniero de Pruebas.

“Una vez que se ha generado el código, comienzan las pruebas del programa. El proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales; es decir, realizar las pruebas para la detección de errores y asegurar que la entrada definida produce resultados reales de acuerdo con los resultados requeridos.” (Pressman, 2005)

El Ingeniero de Pruebas de software es el encargado de identificar y posteriormente definir las pruebas requeridas y de llevar a cabo un conjunto de pruebas de software, así como de documentar las inconformidades halladas durante la realización de las mismas. Conjuntamente con el equipo de desarrollo deben de identificarse las causas principales de estas no conformidades, definiendo así las estrategias para mejorar estos procesos y con ello el software final.

Este rol también se refiere en distintas ocasiones como el *diseñador de casos de prueba*. El mismo es el responsable de: (Hanz, 2007)

- Identificar los elementos de prueba que se evaluarán por el esfuerzo de la prueba.
- Definir las pruebas apropiadas requeridas y cualquier dato de prueba asociado.
- Recopilar y manejar los datos de prueba.
- Evaluar el resultado de cada ciclo de prueba.

1.8 Principios de las pruebas de software

Antes de la aplicación de métodos para diseñar los Casos de Prueba efectivos, los Ingenieros de Software deberán entender los principios básicos que guían las pruebas de software. Según Pressman estos principios son:

- A todas las pruebas se les debería poder hacer un seguimiento hasta los requisitos del cliente.

- Las pruebas deberían planificarse antes de comenzar.
- El principio de Pareto es aplicable a la prueba del software. El 80% de los errores está en el 20% de los módulos. Hay que identificar esos módulos y probarlos muy bien.
- Las pruebas deberían empezar por “lo pequeño” y progresar hacia “lo grande”.
- No son posibles las pruebas exhaustivas.
- Para ser más efectivas, las pruebas deberían ser conducidas por un equipo independiente.

1.9 Características de una buena prueba

Algunos autores como son [Kaner, Falk y Nguyen]² establecen los siguientes atributos para llevar a cabo una “buena” prueba. (Pressman, 2005)

- Una buena prueba tiene una alta probabilidad de encontrar un error. Para alcanzar este objetivo, la persona responsable de la prueba debe entender el software e intentar desarrollar una imagen mental de cómo podría fallar.
- Una buena prueba no debe ser redundante. El tiempo y los recursos para las pruebas son limitados. No existen motivos para realizar una prueba que tiene el mismo propósito que otra. Todas las pruebas deberían tener un propósito diferente (incluso si es sutilmente diferente).
- Una buena prueba debería ser “la mejor de la cosecha”. En un grupo de pruebas que tienen propósito similar, se debería emplear la prueba que tenga la más alta probabilidad de descubrir una clase entera de errores.
- Una buena prueba no debería ser ni demasiado sencilla ni demasiado compleja.

1.10 Diseño de Casos de Prueba

El diseño de Casos de Prueba para el software o para cualquier otro producto de ingeniería puede llegar a requerir tanto esfuerzo como el propio diseño inicial del producto en sí. Pero aún así los Ingenieros de Software a menudo tratan a las pruebas como algo sin importancia mediante el desarrollo de Casos de Prueba que “parecen adecuados”, pero que en realidad tiene pocas garantías de ser completos. Teniendo en cuenta el objetivo de las pruebas, se han de diseñar Casos de Prueba que tengan la mayor probabilidad de encontrar el mayor número de errores posibles con la mínima cantidad de esfuerzo y

² Kaner, C., J. Falk y H.Q. Nguyen. Autores del libro “Testing Computer Software”

tiempo posible. También se debe tener en cuenta de que existe una sola regla para el diseño de los Casos de Prueba que es cubrir todas las posibilidades, sin hacer demasiados. Cualquier producto de ingeniería o de cualquier otro campo en específico se puede probar mediante una de estas dos formas: *prueba de caja blanca* o *prueba de caja negra*. Para la realización de las pruebas de caja blanca se necesita conocer las especificidades del código que sustenta el software, mientras que en las de caja negra únicamente se necesita saber cuáles son los objetivos o las funcionalidades en la interfaz, es decir, lo que el código ha de proporcionar. (Pressman, 2005)

Para verificar que la entrega del módulo Inventario de Petróleo y Gas se hará libre de defectos y con la mayor calidad posible, se procederá a la realización de pruebas de caja blanca y caja negra a este módulo.

1.11 Prueba de caja blanca

Las pruebas de caja blanca, denominadas también *pruebas de caja de cristal*, es un método de diseño de Casos de Prueba que usa la estructura de control del diseño procedimental para obtener los Casos de Prueba. Mediante los métodos de prueba de caja blanca, los ingenieros del software pueden obtener Casos de Prueba que: (Pressman, 2005)

- Garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo.
- Ejerciten todas las decisiones lógicas en sus vertientes verdaderas y falsas.
- Ejecuten todos los bucles en sus límites y con sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

Una pregunta razonable en este punto sería ¿por qué se gasta tiempo y energía probando la lógica del programa cuando se puede ir a probar si las funcionalidades se hacen de forma correcta mediante las pruebas de caja negra? No siempre es así pues en ocasiones se hace necesaria la realización de las pruebas de caja blanca por diferentes situaciones, entre las cuales están: (Pressman, 2005)

- Los errores lógicos y las suposiciones incorrectas son inversamente proporcionales a la probabilidad de que se ejecute un camino del programa.
- Existen ocasiones en las que se cree que un camino lógico tiene pocas posibilidades de ejecutarse aún cuando este se puede ejecutar de forma normal.

- Los errores tipográficos son aleatorios. Muchos de estos errores solo podrán ser detectados mediante esta prueba.

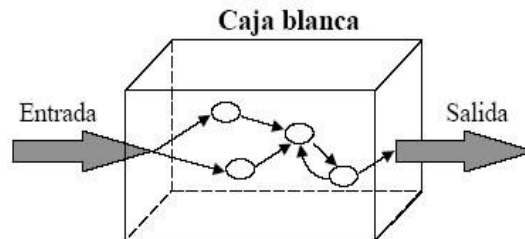


Figura 1: Prueba de caja blanca

1.12 Técnicas de prueba de caja blanca

Existen diferentes técnicas de prueba que están enmarcadas dentro de las pruebas de caja blanca, a continuación se brinda una breve explicación de las mismas.

1.12.1 Prueba de camino básico

Este método permitirá al Diseñador de Casos de Prueba tener una medida de la complejidad lógica y a la vez podrá utilizarla para tenerla como guía para la definición de los caminos básicos posible de ejecución. Una vez que se considere el método de camino básico necesariamente se tendrá que introducir una sencilla notación que será la encargada de representar el flujo de control, la misma se denominará *grafo de flujo*. El grafo de flujo representa el flujo de control lógico mediante una notación que logra representar las distintas instrucciones en el código fuente del programa. (Pressman, 2005)

1.12.2 Prueba de condición

Esta prueba es un método que ejercita las condiciones lógicas contenidas en el módulo de un programa. Los tipos posibles de componentes en una condición pueden ser: un operador lógico, una variable lógica, un par de paréntesis lógicos (que rodean a una condición simple o compuesta), un operador relacional o una expresión aritmética. Cuando una condición es incorrecta, entonces es incorrecto al menos un componente de la condición. Así, los tipos de errores de una condición pueden ser los siguientes: (Pressman, 2005)

- Error en un operador lógico (existencia de operadores lógicos incorrectos / desaparecidos / sobrantes).
- Errores en variables lógicas.
- Errores en paréntesis lógicos.
- Errores en operadores relacionales.
- Errores en una expresión aritmética.

1.12.3 Prueba del flujo de datos

El método de prueba de flujo de datos selecciona caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa. Una sencilla estrategia de prueba de flujo de datos se basa en requerir que se cubra al menos una vez cada cadena de definición-uso (cadena DU). Aun así se ha demostrado que con la realización de la misma no se garantiza el cubrimiento de una ramificación en situaciones raras como las construcciones when-else en las que la parte then no tiene ninguna definición de variable y no existe la parte else. En esta situación, la prueba DU no cubre necesariamente la rama else de la sentencia if superior. Estas estrategias son útiles para seleccionar caminos de prueba de un programa que contenga sentencias if o de bucles anidados. (Pressman, 2005)

1.12.4 Prueba de bucles

A pesar de que los bucles son la piedra angular de la inmensa mayoría de los algoritmos implementados en el software, se les presta normalmente poca atención cuando se llevan a cabo las pruebas del software. La prueba de bucles es una técnica de prueba de caja blanca que se centra exclusivamente en la validez de las construcciones de bucles [BEI90]. Los mismos se pueden definir en distintas clases de bucles: bucles simples, bucles concatenados, bucles anidados y bucles no estructurados. (Pressman, 2005)

1.13 Prueba de caja negra

Las pruebas de caja negra, también denominadas *pruebas de comportamiento*, se centran en los requisitos funcionales del software. Las pruebas de caja negra van a permitir un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales del programa. La prueba de caja negra no es una alternativa a las técnicas de prueba de caja blanca. (Pressman, 2005)

Las pruebas de caja negra intentan encontrar errores en las siguientes categorías:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

Aplicando las técnicas de pruebas de caja negra se podrá obtener un conjunto de Casos de Prueba que satisfagan los siguientes criterios:

- Casos de Prueba que reduzcan, en un coeficiente que es mayor que uno, el número de Casos de Prueba adicionales que se deben diseñar para alcanzar una prueba razonable.
- Casos de Prueba que digan algo sobre la presencia o ausencia de clases de errores en lugar de errores asociados solamente con la prueba que estamos realizando.

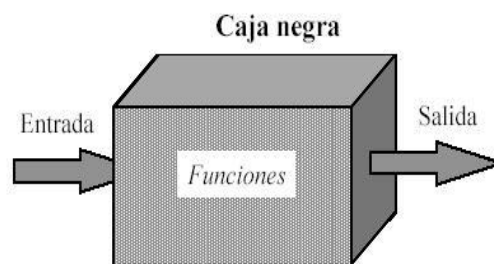


Figura 2: Prueba de caja negra

1.14 Técnicas de prueba de caja negra

El método de prueba de caja negra contiene diferentes técnicas que son utilizadas en la actualidad para probar las funcionalidades de un software. A continuación se muestra una breve explicación de estas técnicas.

1.14.1 Partición equivalente

Es un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar Casos de Prueba. Un Caso de Prueba ideal descubre de forma

inmediata una clase de errores que de otro modo estos requerían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de Casos de Prueba que descubran clases de errores, reduciendo de esa forma el número total de Casos de Prueba que hay que desarrollar. (Pressman, 2005)

Quien va a representar un conjunto de estados válidos o no válidos para condiciones de entrada se denomina *clase de equivalencia*. Una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica. Las clases de equivalencia se pueden definir de acuerdo con las siguientes directrices:

- Si una condición de entrada especifica un rango, se define una clase de equivalencia válida y dos no válidas.
- Si una condición de entrada requiere un valor específico, se define una clase de equivalencia válida y dos no válidas.
- Si una condición de entrada especifica un miembro de un conjunto, se define una clase de equivalencia válida y una no válida.
- Si una condición de entrada es lógica, se define una clase de equivalencia válida y una no válida.

1.14.2 Análisis de valores límites

Los errores tienden a darse con más frecuencia en los límites del campo de entrada que en el “centro”. Para ello se ha desarrollado una técnica de *análisis de valores de límites* (AVL) para ser aplicadas a las pruebas. Su análisis lleva a una elección de Casos de Prueba que ejerciten los valores límite. El mismo es una técnica de diseño de Casos de Prueba que complementa a la partición equivalente. En vez de seleccionar cualquier elemento de una clase de equivalencia, el AVL lleva a la elección de Casos de Prueba en los “extremos” de la clase. Así mismo en lugar de centrarse solamente en las condiciones de entrada, el AVL obtiene Casos de Prueba también para el campo de salida [MYE79]. (Pressman, 2005)

Las directrices de AVL son similares en muchos aspectos a las que proporciona la partición equivalente:

- Si una condición de entrada especifica un rango delimitado por los valores a y b, se deben diseñar Casos de Prueba para los valores a y b, además para los valores justo por debajo y justo por encima de a y b, respectivamente.

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

La gran parte de los Ingenieros del Software llevan a cabo de forma intuitiva alguna forma de AVL. Con la aplicación de las directrices que se acaban de exponer, la prueba de límites será más completa y, por tanto, tendrá una mayor probabilidad de detectar errores.

1.14.3 Prueba de comparación

Hay situaciones en las que la fiabilidad del software es algo absolutamente crítico. En ese tipo de aplicaciones a menudo se utiliza hardware y software redundante y de esta forma se puede minimizar la posibilidad de error. Cuando se desarrolla un software redundante varios equipos de ingeniería del software separados desarrollan versiones independientes de una aplicación, usando las mismas especificaciones. Posteriormente se procede a probar todas las versiones con los mismos datos de prueba, para verificar si todas proporcionan una salida idéntica, luego se ejecutan todas al mismo tiempo y se comparan los resultados en tiempo real. (Pressman, 2005)

1.14.4 Prueba de tabla ortogonal

La prueba de la tabla ortogonal es aplicable a problemas en que el dominio de entrada es relativamente pequeño pero demasiado grande para posibilitar pruebas exhaustivas. Este método es particularmente útil para encontrar errores asociados con fallos localizados, una categoría de error asociada con defectos de la lógica dentro de un componente software. (Pressman, 2005)

Conclusiones

Mediante el análisis realizado en este capítulo se muestra la importancia que tienen las pruebas del software para lograr un producto de calidad. Se han mencionado distintos modelos de calidad además de caracterizar la metodología RUP y el proceso de pruebas que ella establece, llegando al Rol de Ingeniero de Pruebas. Posteriormente se ha analizado los principios y las características de una buena prueba, así como los diseños de Casos de Prueba. También han quedado evidenciados los métodos y las técnicas de pruebas que se emplean para la realización de las mismas. Toda esta base propicia una buena selección de técnicas para la realización de las pruebas y que al realizar las mismas se hagan con la calidad requerida y así poder tener conocimiento de los errores que presenta la aplicación y poder corregirlos antes de entregarla a los clientes.

CAPÍTULO 2

PLANIFICACIÓN DEL PROCESO DE PRUEBA.

2. Introducción

En el presente capítulo se desarrollan una serie de procedimientos que se obtienen luego de la fundamentación teórica planteada en el capítulo 1. Estos posibilitarán la realización de diferentes artefactos para su posterior aplicación, dándole cumplimiento a los objetivos planteados en el presente trabajo de diploma y logrando así una buena aplicación de las pruebas al módulo Inventario de Petróleo y Gas. Se utilizaron las plantillas definidas por RUP para la confección de los diferentes artefactos. Mediante las mismas se logra una mayor organización y estandarización del trabajo, trayendo consigo esto que el proceso de pruebas se realice con la mayor calidad posible, aumentando las posibilidades de detectar errores en el sistema.

2.1 Módulo Inventario de Petróleo y Gas

El módulo Inventario de Petróleo y Gas administra y provee la información sobre el estado de los recursos y las reservas de hidrocarburos propiciando y garantizando el uso racional de las mismas y gestiona la información referida a los prospectos y pozos con sus datos primarios, yacimientos, datos de producción mensual y reservas calculadas de petróleo y gas. Anualmente, edita y distribuye el Balance del Estado de los Recursos y las Reservas de Petróleo y Gas del país. El módulo cuenta con 12 casos de uso diseñados para darle cumplimiento a los requerimientos planteados por el cliente en un inicio. Se le realizarán las pruebas a 6 de ellos.

Para evitar que este producto se entregue con errores a los clientes, se le realizarán pruebas de caja blanca mediante la técnica de camino básico y de caja negra a través de la técnica de partición equivalente al módulo teniendo en cuenta los casos de uso diseñados.

2.2 Características a probar

Existen varias características a la hora de probar un software, sobre todo por las condiciones del producto en sí y también con lo que la prueba a realizar pretenda probar.

Las características más generales a tener en cuenta son las siguientes: si la aplicación cuenta con una interfaz comprensible para el usuario, si la aplicación cumple con los requerimientos planteados al inicio por los clientes, si la documentación es legible a los usuarios y sobre todo que el software no de errores a la hora de ejecutarlo.

Se puede medir también si el código del software es reutilizable y que esté bien estructurado y legible a la vista de otros programadores. Esto posibilita que se le pueda dar mantenimiento, incluso por otro equipo de desarrolladores, logrando así que la aplicación permanezca activa por mucho más tiempo.

2.3 Plan de Prueba

El objetivo del Plan de Pruebas es establecer con anterioridad una guía para poder realizar las pruebas de forma satisfactoria y con la mayor calidad posible. En el Plan de Pruebas que se definió para el módulo Inventario de Petróleo y Gas están identificados los elementos que serán probados en cuestión. También aparecen detallados los Casos de Uso a los cuales se le realizarán las pruebas y la estrategia de pruebas que se aplicará, logrando con esto un mejor diseño de los Casos de Prueba, aumentando la posibilidad de encontrar errores en el sistema.

Los Casos de Prueba definidos están compuestos por 3 bloques, primero el propósito de la prueba, donde se menciona en detalle lo que se quiere lograr con esa prueba en específico, segundo los pasos de ejecución de la prueba, que no es más que la guía de pasos a seguir para realizar la prueba y ver si muestra el resultado esperado al finalizar los pasos y por último el resultado que se espera que es donde se ve si la prueba que se realizó no fue exitosa o se realizó con éxito, en este caso se detectó un fallo en el sistema.

2.3.1 Objetivos

El objetivo fundamental de aplicar las pruebas al módulo Inventario de Petróleo y Gas es detectar fallas o errores en el sistema, mediante un conjunto de Casos de Prueba que están definidos con anterioridad y poder corregir estos errores, logrando que el sistema quede con una mayor calidad en su funcionamiento, obteniéndose un sistema que satisfaga las necesidades de los clientes de la ONRM.

Teniendo en cuenta que un buen Caso de Prueba es aquel que tiene una altísima probabilidad de detectar un error que ha sido pasado por alto por parte del equipo de desarrolladores, se deben diseñar los mismos de forma minuciosa para lograr un buen Caso de Prueba.

2.3.2 Recursos para las pruebas

Para poder desarrollar las pruebas se hace necesaria una serie de recursos tanto de software como de hardware, los cuales se presentan a continuación:

Software:

- Windows 2000 NT, Windows XP y Linux, Ubuntu 7.x
- Sistema de Gestión de Base de datos PostgreSQL versión 8.3
- Servidor Web: Apache

Hardware

- 1 GB de Memoria RAM como mínimo

2.3.3 Requerimientos generales

Los requerimientos planteados por los clientes en un inicio son los elementos fundamentales que se deben tener en cuenta a la hora de realizar los Casos de Prueba. Ellos son los que van a determinar las funcionalidades del sistema, pues constituyen condiciones o capacidades que este debe cumplir y forman parte de la documentación del proyecto desde sus inicios. También teniendo en cuenta que el cliente al recibir su producto pretende ver que las funcionalidades en el sistema se realicen de manera correcta, verificar la correcta implementación de los requerimientos funcionales es una de las funciones principales en el proceso de pruebas que se le realiza al módulo.

Estos requerimientos se agrupan en Casos de Uso y a partir de ellos los desarrolladores del sistema los implementan.

2.3.4 Pasos a seguir en las pruebas o estrategia

Antes de comenzar a planear las pruebas se hace necesario tener conocimiento de todo lo que ocurre en el módulo, así como los requisitos planteados por los clientes en un inicio, para saber cómo funciona. Con

esto se garantiza una mejor aplicación de las pruebas, debido a que se preparan los Casos de Prueba de una forma más lógica y precisa. El proceso de pruebas comienza haciendo una selección de las técnicas y métodos mediante las cuales se van a realizar las mismas. Se decidió realizar por el método de pruebas de caja blanca la técnica de camino básico a 3 Casos de Uso críticos del sistema y con el método de caja negra utilizando la técnica de partición equivalente a 6 Casos de Uso. Posteriormente se procede a la realización del diseño de los Casos de Prueba. Una vez elaborados todos los artefactos que son necesarios para llevar a cabo el proceso de pruebas, se procede a la realización de las mismas. Se probará el software para comprobar que las funcionalidades implementadas están en correcto estado, en caso de detectarse errores, estos serán registrados. Finalizado el proceso de pruebas se documentarán los resultados de las pruebas, generando el documento de No Conformidades, el cual contendrá los errores detectados.

2.3.5 Diseño de Casos de Prueba de caja negra

Las pruebas de caja negra verifican las funcionalidades del sistema. Con la aplicación de la técnica de partición equivalente en el módulo se podrá comprobar si los requerimientos planteados por el cliente en un inicio y descritos en Casos de Uso, fueron cumplidos en su totalidad sin fallos o errores.

A continuación se muestran los Casos de Prueba de caja negra para 3 Casos de Uso:

Nombre del Caso de Uso: Gestionar Prospecto

Descripción general: El CU comienza cuando el Operador desea adicionar, modificar o eliminar un prospecto. El sistema brinda las opciones pertinentes al Operador. El Operador selecciona la opción deseada, realiza la operación y termina el CU con la actualización de los datos.

Precondiciones: El usuario debe estar autenticado como administrador del módulo.

Secciones a probar en el Caso de Uso:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC1: Gestionar Prospecto	EC 1.1: Mostrar pestañas	Se muestran dos pestañas: Buscar y Adicionar. En la pestaña de Buscar, se muestran las opciones correspondientes: (modificar, eliminar, detalles)

SC2: Mostrar Prospectos	EC 2.1: Se muestran los prospectos.	Se ha verificado la sección del administrador como válida. Se mostrarán todos los Prospectos existentes.
	EC 2.2: Se muestra solamente el listado de Prospectos.	Se debe de mostrar el listado de Prospectos existentes en el sistema con la opción de buscar.
SC3: Adicionar Prospecto	EC 3.1: Se adiciona el Prospecto.	El prospecto es único en el sistema, el Prospecto es adicionado y mostrado en el listado de Prospectos.
	EC 3.2: El administrador no adiciona el Prospecto.	La interfaz de Adicionar Prospecto es cerrada por el administrador.
	EC 3.3: Datos Incorrectos.	Los datos que se requieren para adicionar un Prospecto son incorrectos.
	EC 3.4: Sesión de administrador inválida.	El Prospecto no se puede adicionar ya que la sesión de administrador es inválida.
	EC 3.5: Actualiza la lista de los Prospectos.	La lista de los Prospectos existentes en el sistema se actualiza.
SC4: Modificar Prospecto	EC 4.1: Se modifica el Prospecto.	El Prospecto es modificado y mostrado en el listado de Prospectos.
	EC 4.2: El administrador no modifica el Prospecto.	La interfaz de Modificar Prospecto es cerrada por el administrador.
	EC 4.3: Datos Incorrectos.	Los datos que se requieren para modificar un Prospecto son incorrectos.
	EC 4.4: Sesión de administrador inválida.	El Prospecto no se puede modificar ya que la sesión de administrador es inválida.
	EC 4.5: Actualiza la lista de los Prospectos.	La lista de los Prospectos existentes en el sistema se actualiza.
SC5: Eliminar Prospecto	EC5.1: Eliminando un Prospecto.	El administrador selecciona el Prospecto que desea eliminar, elimina el Prospecto y el sistema actualiza el listado.
	EC5.2: No se confirma la eliminación del Prospecto	El administrador no confirma la eliminación del Prospecto seleccionado.

	EC5.3: Sesión del administrador inválida	La sesión del Administrador es inválida.
--	--	--

Tabla 1: Diseño del CP del CU: Gestionar Prospecto

Nombre del Caso de Uso: Gestionar Yacimiento

Descripción general: El CU comienza cuando el Operador desea adicionar, modificar o eliminar un yacimiento. El sistema brinda las opciones pertinentes al Operador. El Operador selecciona la opción deseada, realiza la operación y termina el CU con la actualización de los datos.

Precondiciones: El usuario debe estar autenticado como administrador del módulo.

Secciones a probar en el Caso de Uso:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC1: Gestionar Yacimiento	EC 1.1: Mostrar pestañas	Se muestran dos pestañas: Buscar y Adicionar. En la pestaña de Buscar, se muestran las opciones correspondientes: (modificar, eliminar, detalles)
SC2: Mostrar Yacimientos	EC 2.1: Se muestran los Yacimientos.	Se ha verificado la sección del administrador como válida. Se mostrarán todos los Yacimientos existentes.
	EC 2.2: Se muestra solamente el listado de Yacimientos.	Se debe de mostrar el listado de Yacimientos existentes en el sistema con la opción de buscar.
SC3: Adicionar Yacimiento	EC 3.1: Se adiciona el Yacimiento.	El Yacimiento es adicionado y mostrado en el listado de Yacimientos.
	EC 3.2: El administrador no adiciona el Yacimiento.	La interfaz de Adicionar Yacimiento es cerrada por el administrador.
	EC 3.3: Datos Incorrectos.	Los datos que se requieren para adicionar un Yacimiento son incorrectos.
	EC 3.4: Sesión de administrador inválida.	El Yacimiento no se puede adicionar ya que la sesión de administrador es inválida.

	EC 3.5: Actualiza la lista de los Yacimientos.	La lista de los Yacimientos existentes en el sistema se actualiza.
SC4: Modificar Yacimiento	EC 4.1: Se modifica el Yacimiento.	El Yacimiento es modificado y mostrado en el listado de Yacimientos.
	EC 4.2: El administrador no modifica el Yacimiento.	La interfaz de Modificar Yacimiento es cerrada por el administrador.
	EC 4.3: Datos Incorrectos.	Los datos que se requieren para modificar un Yacimiento son incorrectos.
	EC 4.4: Sesión de administrador inválida.	El Yacimiento no se puede modificar ya que la sesión de administrador es inválida.
	EC 4.5: Actualiza la lista de los Yacimientos.	La lista de los Yacimientos existentes en el sistema se actualiza.
SC5: Eliminar Yacimiento	EC5.1: Eliminando un Yacimiento.	El administrador selecciona el Yacimiento que desea eliminar, elimina el Yacimiento y el sistema actualiza el listado.
	EC5.2: No se confirma la eliminación del Yacimiento	El administrador no confirma la eliminación del Yacimiento seleccionado.
	EC5.3: Sesión del administrador inválida	La sesión del Administrador es inválida.

Tabla 2: Diseño del CP del CU: Gestionar Yacimiento

Nombre del Caso de Uso: Gestionar Bloque

Descripción general: El CU comienza cuando el Operador desea adicionar, modificar o eliminar un bloque. El sistema brinda las opciones pertinentes al Operador. El Operador selecciona la opción deseada, realiza la operación y termina el CU con la actualización de los datos.

Precondiciones: El usuario debe estar autenticado como administrador del módulo.

Secciones a probar en el Caso de Uso:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
----------------------	--------------------------	---------------------------------

SC1: Gestionar Bloque	EC 1.1: Mostrar pestañas	Se muestran dos pestañas: Buscar y Adicionar. En la pestaña de Buscar, se muestran las opciones correspondientes: (modificar, eliminar, detalles)
SC2: Mostrar Bloques	EC 2.1: Se muestran los Bloques.	Se ha verificado la sección del administrador como válida. Se mostrarán todos los Bloques existentes.
	EC 2.2: Se muestra solamente el listado de Bloques.	Se debe de mostrar el listado de Bloques existentes en el sistema con la opción de buscar.
SC3: Adicionar Bloque	EC 3.1: Se adiciona el Bloque.	El Bloque es adicionado y mostrado en el listado de Bloques.
	EC 3.2: El administrador no adiciona el Bloque.	La interfaz de Adicionar Bloque es cerrada por el administrador.
	EC 3.3: Datos Incorrectos.	Los datos que se requieren para adicionar un Bloque son incorrectos.
	EC 3.4: Sesión de administrador inválida.	El Bloque no se puede adicionar ya que la sesión de administrador es inválida.
	EC 3.5: Actualiza la lista de los Bloques.	La lista de los Bloques existentes en el sistema se actualiza.
SC4: Modificar Bloque	EC 4.1: Se modifica el Bloque.	El Bloque es modificado y mostrado en el listado de Bloques.
	EC 4.2: El administrador no modifica el Bloque.	La interfaz de Modificar Bloque es cerrada por el administrador.
	EC 4.3: Datos Incorrectos.	Los datos que se requieren para modificar un Bloque son incorrectos.
	EC 4.4: Sesión de administrador inválida.	El Bloque no se puede modificar ya que la sesión de administrador es inválida.
	EC 4.5: Actualiza la lista de los Bloques.	La lista de los Bloques existentes en el sistema no se actualiza.
SC5: Eliminar Bloque	EC5.1: Eliminando un Bloque.	El administrador selecciona el Bloque que desea eliminar, elimina el Bloque y el sistema actualiza el listado.

	EC5.2: No se confirma la eliminación del Bloque	El administrador no confirma la eliminación del Bloque seleccionado.
	EC5.3: Sesión del administrador inválida	La sesión del Administrador es inválida.

Tabla 3: Diseño del CP del CU: Gestionar Bloque

2.3.6 Diseño de Casos de Prueba de caja blanca

Las pruebas de caja blanca ejercitan que se cumplan todos los caminos lógicos del programa. Con la aplicación de la técnica de camino básico se realiza un grafo de flujo del código fuente seleccionado previamente.

Los pasos para la realización de esta técnica son los siguientes:

Paso 1. Generar el grafo de flujo de datos a partir del código/diseño procedimental (Nodos, Aristas, Regiones)

Paso 2. Calcular la complejidad ciclomática $V(G)$: métrica del software que da una medición de la complejidad de un programa.

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

$$V(G) = P \text{ (Nodos predicados)} + 1$$

$$V(G) = \text{Número de regiones}$$

Paso 3. Definir un conjunto básicos de caminos de ejecución a partir de la complejidad ciclomática.

Paso 4. Generar un Caso de Prueba para cada camino de ejecución.

Se muestra a continuación 3 Casos de Prueba utilizando esta técnica:

Caso de Prueba CB: Buscar Prospecto

```
public function executeBuscarProspecto()
{
```

```
1      $texto = $this->getRequestParameter('txtTexto');
1      $inicio = $this->getRequestParameter('txtInicio');
1      $cantidad = $this->getRequestParameter('txtCantidad');
1      $prospectos = TprospectonaturalPeer::BuscarProspectoTexto($texto, $cantidad, $inicio);
1      $t = TprospectonaturalPeer::CantidadProspectoTexto($texto);
1      $array_json = '{}';
1      $Json = array();
1      $i = 0;
2      foreach ($prospectos as $selem)
        {
3          $Json[$i++] = array (
                "Nombre" => $selem["nombre"],
                "Solicitante" => $selem["nombres"],
                "id"    => $selem["id"]
            );
        }
4      $titulos = array(0 =>"Nombre" ,
                1 => "Solicitante",
                2 => "Acciones",
                3 => "1" ,
                4 => "2",
                5 => "3");
5      $acciones= array(
                array("funcion" => "ajaxModificarProspecto",
                    "img" => '../..../images/icon/edit.png'),
```

```

array("img" => ' "1",
      "accion" => "/balancep/prospecto/eliminar/",
      "mensaje" => "Estas seguro que desea eliminarlo",
      "titulo" => "Eliminar Prospecto"),
array("img" => '../../../../images/icon/filter.png',
      "funcion" => "ajaxDetallesProspecto" )
);

```

```

6   $Json[$i++] = $acciones;
6   $Json[$i++] = $titulos;
6   $Json[$i++] = array("total" => $t);
6   $array_json = json_encode($Json);
6   $this->getResponse()->setHTTPHeader("application/json");
6   echo $array_json;
6   return sfView::NONE;
6 }

```

Paso 1: Generar Grafo de flujo de datos

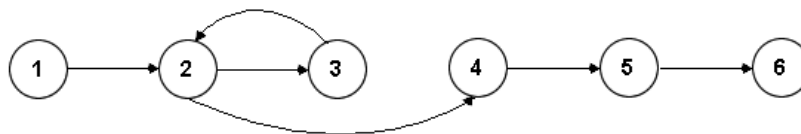


Figura 3: Grafo de flujo de datos para el método Buscar Prospecto

Paso 2: Cálculo de complejidad ciclomática

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

$V(G) = 6 - 6 + 2 = 2$

Paso 3: Caminos básicos

CB1: 1 2 4 5 6

CB2: 1 2 **3** 2 4 5 6

Paso 4: Caso de prueba para el camino básico.

CB1: 1 2 3 2 4 5 6

Caso de prueba: Buscar Prospecto.

Entrada: Una cadena de texto.

Resultado esperado: Se muestran los datos que coinciden con esa cadena de texto.

Resultado de la Prueba: Satisfactoria

Caso de Prueba CB: Modificar Yacimiento

```
public function executeModificar()
{
1     $yac_id = $this->getRequestParameter('yac', null);
2     if ($yac_id == null)
3         $this->redirect ('yacimiento/yacimiento');
4     if ($this->getRequest()->getMethod() == sfRequest::POST)
        {
5         $nombre = $this->getRequestParameter('txtNombre');
5         $hidrocarburo = $this->getRequestParameter('cbxHidrocarburo');
5         $solicitante_id = $this->getRequestParameter('cbxSolicitante');
```

```

5         TyacimientonaturalPeer::ActualizarYacimiento($yac_id,      $nombre,      $hidrocarburo,
$solitante_id);
        }
6         $this->yac = TyacimientonaturalPeer::ObtenerYacimientoNatural($yac_id);
6         $this->solicitantes = TsolicitantePeer::ObtenerSolicitantes();
6         return sfView::SUCCESS;
6     }
    
```

Paso1: Generar Grafo de flujo de datos.

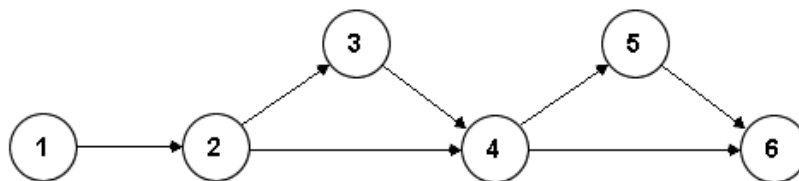


Figura 4: Grafo de flujo de datos para el método Modificar Yacimiento

Paso 2: Cálculo de complejidad ciclomática

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

$$V(G) = 7 - 6 + 2 = 3$$

Paso 3: Caminos básicos

CB1: 1 2 4 6

CB2: 1 2 3 4 6

CB3: 1 2 3 4 5 6

Paso 4: Caso de prueba para el camino básico.

CB3: 1 2 3 4 5 6

Caso de Prueba: Modificar Yacimiento.

Entrada: Nombre, Hidrocarburo, Solicitante.

Resultado esperado: Se modifican los datos del yacimiento.

Resultado de la Prueba: No satisfactorio. No se pudo comprobar ya que no se puede acceder a la sección de modificar yacimiento. El enlace se encuentra roto.

Caso de Prueba CB: Datos del Prospecto

```
public function executeDatos()
{
1     $this->prospecto_id = $this->getRequestParameter('prosp', null);
2     if ($this->prospecto_id == null)
3         $this->redirect('datosprospecto/index');
4     if ($this->getRequest()->getMethod() == sfRequest::POST)
        {
5         $porosidad_efectiva = $this->getRequestParameter('txtPorosidadEfectiva');
5         $saturacion_efectiva = $this->getRequestParameter('txtSaturacionEfectiva');
5         $cantidadHidrocarburo = $this->getRequestParameter('txtCantidadHidrocarburo');
5         $espesor_efectivo = $this->getRequestParameter('txtEspesorEfectivo');
5         $coeficiente = $this->getRequestParameter('txtCoeficiente');
5         $area = $this->getRequestParameter('txtArea');
5         $metodo_utilizado = $this->getRequestParameter('cbxMetodoUtilizado');
5         $datos_validos = $this->getRequestParameter('hddDatosValidos');
5         $probabilidad_exito = $this->getRequestParameter('txtProbabilidadExito');
5         $ses_factible = ($this->getRequestParameter('rdbFactibilidadEconomica') == '1');
```

```
5         $costo = $this->getRequestParameter('txtCosto');

6         if (TprospectonaturalPeer::EsProspectoDePetroleo($this->prospecto_id))
        {
7             $factor_gaseoso = $this->getRequestParameter('txtFactorGaseoso');
7             TdatosprospectopetPeer::AsignarDatosProspectoNaturalPetroleo($this->prospecto_id, $porosidad_efectiva, $saturacion_efectiva,
7             $cantidadHidrocarburo, $espesor_efectivo, $coeficiente, $area, $metodo_utilizado,
7             $datos_validos,
7             $probabilidad_exito, $es_factible, $costo, $factor_gaseoso);
        }
        else
        {
8             TdatosprospectoPeer::AsignarDatosProspectoNatural($this->prospecto_id,
8             $porosidad_efectiva, $saturacion_efectiva,
8             $cantidadHidrocarburo, $espesor_efectivo, $coeficiente, $area, $metodo_utilizado,
8             $datos_validos,
8             $probabilidad_exito, $es_factible, $costo);
        }
9         $cantidad_intervalos = $this->getRequestParameter('hddCantidadIntervalos');
10        for ($i = 1; $i <= $cantidad_intervalos; $i++)
        {
11            $intervalo_inicio = $this->getRequestParameter('profundidad_i_'.$i, null);
12            if ($intervalo_inicio != null)
            {
13                $intervalo_fin = $this->getRequestParameter('profundidad_f_'.$i);
13                $porosidad = $this->getRequestParameter('porosidad_'.$i);
```

```

13         $saturacion = $this->getRequestParameter('saturacion_'. $i);
13         $permeabilidad = $this->getRequestParameter('permeabilidad_'. $i);
13         $espesor = $this->getRequestParameter('espesor_'. $i);
13         TprospectonaturalPeer::RegistrarIntervalo($this->prospecto_id,
13         $intervalo_inicio, $intervalo_fin, $porosidad, $saturacion, $permeabilidad, $espesor);
        }
    }
}

14 $this->datos_prospecto = TdatosprospectoPeer::ObtenerDatosProspectoNatural($this->prospecto_id);
14 $this->intervalos = TprospectonaturalPeer::ObtenerIntervalos($this->prospecto_id);
14 $this->metodos = TlistaOficialPeer::ObtenerMetodosDeCalculo();
14 return sfView::SUCCESS;
14 }
    
```

Paso1: Generar Grafo de flujo de datos.

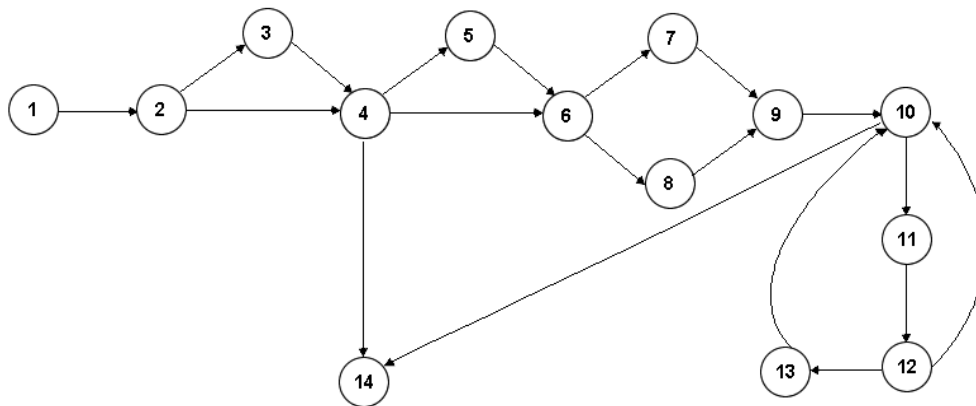


Figura 5: Grafo de flujo de datos para el método Datos del Prospecto

Paso 2: Cálculo de complejidad ciclomática

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

$$V(G)=19 - 14 + 2 = 7$$

Paso 3: Caminos básicos

CB1: 1 2 4 14

CB2: 1 2 4 **6 7 9 10** 14

CB3: 1 2 **3** 4 6 7 9 10 14

CB4: 1 2 3 4 **5** 6 7 10 14

CB5: 1 2 3 4 5 6 **8** 9 10 14

CB6: 1 2 3 4 5 6 8 9 10 **11 12** 10 14

CB7: 1 2 3 4 5 6 8 9 10 11 12 **13** 10 14

Paso 4: Caso de Prueba para el camino básico.

CB1: 1 2 3 4 5 6 8 9 10 11 12 13 10 14

Caso de Prueba: Datos Prospecto.

Entrada: Intervalo 1, Intervalo 2, %Porosidad, %Saturación, Permeabilidad, Espesor Efectivo.

Resultado esperado: Se guardan los datos del prospecto.

Resultado de la Prueba: Satisfactoria

2.5 Manual de Usuario y Especificación de Requerimientos

Dentro de los artefactos que se encuentran en el módulo y que serán liberados en conjunto con él, están el Manual de Usuario y la Especificación de Requerimientos. Debido a esto se hace necesario comprobar la calidad de los mismos mediante una Lista de Chequeo, en la cual estarán recogidas un conjunto de preguntas y mediante estas se podrá validar que dichos artefactos se encuentran elaborados de forma correcta.

2.5.1 Listas de Chequeo

“Se entiende por lista de chequeo a un listado de preguntas, en forma de cuestionario que sirve para verificar el grado de cumplimiento de determinadas reglas establecidas a priori con un fin determinado.” (Bichachi, 2002)

Existen diferentes formas de realizar una Lista de Chequeo, el contenido y extensión de las mismas es variado, las hay muy breves y también muy extensas, complejas y sencillas, lo cierto es que no

siempre las listas de chequeo más completas y exhaustivas son las mejores ni las más exitosas. En ocasiones es preferible redactar un cuestionario breve y fácil de responder seleccionando los puntos más claros para realizar las preguntas.

Algunas de las claves del éxito de una lista de chequeo, en cuanto a aceptación e incorporación para su uso, es que tenga estas características: (Bichachi, 2002)

- Quien deba responderla la entienda fácilmente.
- No consuma mucho tiempo llenar el formulario, (solo el estrictamente necesario para responderla a conciencia).
- Quien debe leer las respuestas, también lo pueda hacer de una manera rápida y clara (evitando incluso el problema de redacciones confusas).

Se puede hacer uso de alguna Lista de Chequeo ya elaborada o se puede diseñar una propia ateniéndose a las peculiares necesidades de la organización. De ser posible, siempre es preferible elegir esta última opción, debido a que redactando la lista se puede contemplar de manera más precisa los intereses de quienes van a usarlas y las particularidades de la institución donde se va a aplicar. (Bichachi, 2002)

Conclusiones

En este capítulo se detallaron las características del módulo Inventario de Petróleo y Gas, el cual fue sometido al proceso de pruebas. Se realizaron los artefactos que estaban definidos como entregables, dentro de ellos el Plan de Pruebas que el mismo va a tener como objetivo fundamental establecer una línea inicial para que el diseño y la aplicación de las pruebas se ejecuten de manera satisfactoria. También se diseñaron los Casos de Prueba tanto de caja blanca como de caja negra. Se diseñaron 10 Casos de Prueba de caja negra, utilizando la técnica de partición equivalente y 3 de caja blanca, con la técnica de camino básico.

CAPÍTULO 3

RESULTADOS OBTENIDOS

3. Introducción

Cuando se realiza el proceso de pruebas a una aplicación, es posible detectar fallas en el sistema, las cuales se registran en el documento de No Conformidades y este se le entrega al equipo de desarrollo para que le dé solución a las mismas.

En el presente capítulo se expondrán algunos de los resultados obtenidos al aplicar las pruebas al módulo.

3.1 Resultados de la aplicación de las pruebas al módulo

3.1.1 Caso de Prueba Gestionar Prospecto.

SC2: Mostrar Prospectos

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC2.1 Se muestran los prospectos.	El sistema muestra el listado con todos los Prospectos existente en el sistema.	Satisfactoria	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Prospecto
EC2.2 Se muestra solamente el listado de Prospectos.	Si no se ha iniciado sesión como administrador, debe mostrar solamente el listado de los Prospectos con la opción de buscar.	No Satisfactoria	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Prospecto

Tabla 4: Resultados del CP: Gestionar Prospecto. SC1: Mostrar Prospecto

SC 3: Adicionar Prospecto

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 3.1 Se adiciona el Prospecto.	El sistema adiciona el nuevo Prospecto y actualiza el listado de los Prospectos y restablece los valores por defectos de los campos de la interfaz.	Satisfactoria	<ul style="list-style-type: none"> • Inventario Petróleo • Gestión • Prospecto • Adicionar

EC 3.2 El administrador no adiciona el Prospecto.	El administrador cancela la operación. El sistema regresa al listado de Prospectos.	No Satisfactoria	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Prospecto • Adicionar
EC 3.3 Datos Incorrectos.	El sistema da un mensaje de error y no adiciona el Prospecto	No Satisfactoria	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Prospecto • Adicionar
EC 3.4 Actualiza la lista de los Prospectos.	Una vez que se adiciona un nuevo Prospecto, el sistema debe de actualizar la lista de los prospectos existentes en el sistema, mostrando el adicionado recientemente.	Satisfactoria	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Prospecto • Adicionar

Tabla 5: Resultados del CP: Gestionar Prospecto. SC2: Adicionar Prospecto

SC 4: Modificar Prospecto

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 4.1 Se modifica el Prospecto.	En los campos Solicitante, Nombre e Hidrocarburo están los datos actuales del Prospecto. El administrador actualiza los campos. El sistema actualiza el listado de los Prospectos.	No Satisfactoria (El enlace no funciona, el vínculo está roto, por tal motivo no se puede probar los demás escenarios para esta sección.)	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Prospecto • Modificar
EC 4.2 El administrador no modifica el Prospecto.	El administrador cancela la operación. El sistema regresa al listado de Prospectos.	No Satisfactoria	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Prospecto • Modificar
EC 4.3 Datos Incorrectos.	El sistema da un mensaje de error y no modifica el Prospecto.	No Satisfactoria	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Prospecto • Modificar

EC 4.4 Actualiza la lista de los Prospectos.	Se actualiza la lista de los Prospectos y el Prospecto modificado se ve en la lista.	No Satisfactoria	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Prospecto • Adicionar
---	--	------------------	---

Tabla 6: Resultados del CP: Gestionar Prospecto. SC3: Modificar Prospecto

SC 5: Eliminar Prospecto

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC5.1 Eliminando un Prospecto	Se elimina el Prospecto seleccionado y se actualiza el listado de Prospectos existentes.	No Satisfactoria	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Prospecto • Eliminar
EC5.2 No se confirma la eliminación del Prospecto	Se cierra la interfaz de confirmación de eliminación y no se elimina el Prospecto seleccionado.	Satisfactoria	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Prospecto • Eliminar

Tabla 7: Resultados del CP: Gestionar Prospecto. SC4: Eliminar Prospecto

3.1.2 Caso de Prueba Gestionar Bloque.

SC 2: Mostrar Bloque

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC2.1 Se muestran los Bloques.	El sistema muestra el listado con todos los Bloques existentes.	Satisfactoria	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Bloque
EC2.2 Se muestra solamente el listado de Bloques.	Si no se ha iniciado sesión como administrador, debe mostrar solamente el listado de los Bloques con la opción de buscar.	No Satisfactoria	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Bloque

Tabla 8: Resultados del CP: Gestionar Bloque. SC1: Mostrar Bloque

SC 3: Adicionar Bloque

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 3.1 Se adiciona el Bloque.	El sistema adiciona el nuevo Bloque y actualiza el listado de los Bloques y restablece los valores por defectos de los campos de la interfaz.	Satisfactoria	<ul style="list-style-type: none"> Principal Inventario Petróleo Gestión Bloque Adicionar
EC 3.2 El administrador no adiciona el Bloque.	El administrador cancela la operación. El sistema regresa al listado de Bloques.	No Satisfactoria	<ul style="list-style-type: none"> Principal Inventario Petróleo Gestión Bloque Adicionar
EC 3.3 Datos Incorrectos.	El sistema da un mensaje de error y no adiciona el Bloque.	No Satisfactoria	<ul style="list-style-type: none"> Principal Inventario Petróleo Gestión Bloque Adicionar
EC 3.4 Actualiza la lista de los Bloques.	El sistema actualiza el listado de los Bloques existentes en el sistema, ya que el que se adiciona se muestra.	Satisfactoria	<ul style="list-style-type: none"> Principal Inventario Petróleo Gestión Bloque Adicionar

Tabla 9: Resultados del CP: Gestionar Bloque. SC2: Adicionar Bloque

SC 4: Modificar Bloque

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 4.1 Se modifica el Bloque.	En los campos Solicitante, Nombre e Hidrocarburo están los datos actuales del Bloque. Se actualizan los campos. El sistema actualiza el listado de los Bloques.	No Satisfactoria (El enlace para modificar no funciona. No se pueden probar las demás sesiones de este escenario por este motivo.)	<ul style="list-style-type: none"> Principal Inventario Petróleo Gestión Bloque Modificar

EC 4.2 El administrador no modifica el Bloque.	El administrador cancela la operación. El sistema regresa al listado de Bloques.	No Satisfactoria	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Bloque • Modificar
EC 4.3 Datos Incorrectos.	El sistema da un mensaje de error y no modifica el Bloque.	No Satisfactoria	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Bloque • Modificar
EC 4.4 Sesión de administrador inválida.	Si no se ha iniciado sesión como administrador, el sistema no debe de mostrar la opción de Modificar Bloque. Solamente mostrar la lista de los Bloques en el sistema y una opción de búsqueda.	No Satisfactoria	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Bloque
EC 4.5 Actualiza la lista de los Bloques.	Se actualiza la lista de los Bloques y el Bloque modificado se ve en la lista.	No Satisfactoria	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Bloque • Modificar

Tabla 10: Resultados del CP: Gestionar Bloque. SC3: Modificar Bloque

SC 5: Eliminar Bloque

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC5.1 Eliminando un Bloque.	Se elimina el Bloque seleccionado y se actualiza el listado de Bloques existentes.	No Satisfactoria	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Bloque • Eliminar
EC5.2 No se confirma la eliminación del Bloque	Se cierra la interfaz de confirmación de eliminación y no se elimina el Bloque seleccionado.	Satisfactoria	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Bloque • Eliminar

Tabla 11: Resultados del CP: Gestionar Bloque. SC4: Eliminar Bloque

3.1.3 Caso de Prueba Gestionar Datos Primarios de Prospecto.

SC 2: Gestionar Datos primarios de Prospecto

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 2.1 Se gestionan datos del Datos primarios de Prospecto	El sistema gestiona los datos del Datos primarios de Prospecto. El sistema muestra el listado de los Datos primarios de Prospectos existentes.	Satisfactorio	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Primarios • Gestionar
EC 2.2 El administrador no gestiona los datos del Datos primarios de Prospecto.	El administrador cancela la operación. El sistema regresa al listado de Datos primarios de Prospectos.	No Satisfactorio	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Primarios • Gestionar
EC 2.3 Datos Incorrectos.	El sistema da un mensaje de error y no adiciona el Datos primarios de Prospecto.	No Satisfactoria	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Primarios • Gestionar

Tabla 12: Resultados del CP: Gestionar Datos Primarios de Prospecto. SC2: Gestionar Datos

3.1.4 Caso de Prueba Gestionar Datos de Producción Mensual.

SC 2: Adicionar Datos de Producción Mensual

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 2.1 Se adicionan Datos de Producción Mensual	El sistema adiciona los Datos Generales y actualiza el listado de los Datos de Producción Mensual y restablece los valores por defectos de los campos de la interfaz.	No Satisfactorio	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Producción • Adicionar

EC 2.2 El administrador no adiciona el Datos de Producción Mensual	El administrador cancela la operación. El sistema regresa al listado de Datos de Producción Mensual.	No Satisfactorio	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Producción • Adicionar
EC 2.3 Datos Incorrectos.	El sistema da un mensaje de error y no adiciona el Datos de Producción Mensual.	Satisfactorio	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Producción • Adicionar
EC 2.4 No se actualiza la lista de los Datos de Producción Mensual.	El sistema no actualiza el listado de los Datos de Producción Mensual existentes en el sistema, ya que el que se adiciona no se muestra.	No Satisfactorio (No se puede comprobar, ya que el adicionar no funciona)	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Producción • Adicionar

Tabla 13: Resultados del CP: Gestionar Datos de Producción Mensual. SC2: Adicionar Datos

3.2 Errores en el documento Especificación de Requerimientos

Mediante la aplicación de la Lista de Chequeo al documento Especificación de Requerimientos se realizaron 2 señalamientos. Básicamente fueron errores de falta de ortografía, detectándose 3 de ellos, empañando la buena calidad y estructura que contiene el documento. Estos errores fueron informados de inmediato al equipo de desarrollo para corregir el documento.

A continuación se muestra la Lista de Chequeo aplicada con sus resultados:

Estructura del documento					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
crítico	1. ¿Está el documento acorde con a la plantilla estándar del proyecto o del expediente de	0			

	proyecto?				
crítico	2. ¿Contiene las secciones obligatorias definidas en el expediente? (Ver Expediente de Proyecto)	0			
Elementos definidos por la metodología					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
crítico	1. ¿Están todos los requisitos redactados de forma simple y clara para aquellos que vayan a consultarlo en un futuro?	0			
	2. ¿Debería especificarse algún requisito con más detalle?	0			
	3. ¿Debería especificarse algún requisito con menos detalles?	0			
	4. ¿Todos los requisitos identificados	0			

	se centran en lo que el sistema debe hacer y no como el sistema debe hacerlo?				
crítico	5. ¿Han sido abordadas e identificadas los valores de entradas y salidas?	0			
	6. ¿Han sido incluidos las respuestas válidas y no válidas de los valores de entrada?	0			
	7. ¿Se han identificado los requerimientos de software y de hardware?	0			
	8. ¿Han sido identificadas las restricciones de diseño e implementación?	0			
	9. ¿Han sido identificadas las restricciones de interfaz externa?	0			
	10. ¿Los requerimientos de	0			

	soporte y usabilidad se han identificados?				
	11. ¿Se han identificado los requerimientos de seguridad (confidencialidad, integridad, disponibilidad)?	0			
Semántica del documento					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
crítico	1. ¿Ha identificado errores ortográficos?	1		3	Existe una palabra que comienza con minúscula después de un punto (2.4 Gestionar Yacimientos (.estos)). Existen dos palabras unidas sin espacio entre ellas (5.1 Eficiencia (de15 segundos)). Existe una inconsistencia con el estándar de los números (8. En vez de tener 8.1 tiene 1.1 y en otro formato)
crítico	2. ¿Se entiende claramente lo que se ha especificado en el documento?	0			
	3. ¿El número de página que aparece en el índice coincide con el	1		13	No coincide el número especificado en el índice con el del pie de página del documento.

	contenido que se refleja realmente en dicha página?				
	4. ¿El total de páginas que aparecen en las reglas de confidencialidad coincide con el total de páginas que tiene el documento?	0			

Tabla 14: Lista de Chequeo para el documento Especificación de Requerimientos

3.3 Principales errores detectados en el proceso de pruebas

En la primera etapa de aplicación de las pruebas se detectaron un total de 34 No Conformidades al módulo en la aplicación y 1 No Conformidad en el Caso de Prueba de caja blanca: Modificar Yacimiento. También se realizaron 2 señalamientos mediante la aplicación de la Lista de Chequeo al documento Especificación de Requerimientos, detectándose un total de 37 No Conformidades en el módulo en general. Dentro de los principales errores encontrados están los siguientes:

- El usuario no necesita iniciar sesión en el sistema para poder realizar la acción de adicionar, modificar y eliminar.
- Existen muchos enlaces que no están funcionando correctamente. Ejemplo de ellos es el caso de los vínculos para modificar los datos, que no funcionó en ninguno de los 6 Casos de Prueba de caja negra que se le aplicaron al módulo.
- Se muestran mensajes que no cumplen con el diseño de interfaz de usuario definido para el módulo.
- Existen errores en la validación de los datos. En los campos de entrada de datos, se puede introducir prácticamente cualquier cadena de texto, no importa cuál sea, la aplicación las acepta, aún cuando se nota claramente que no pueden ser aceptadas por cuestiones lógicas.
- Los mensajes de confirmación para eliminar los datos seleccionados no realizan acción alguna.

- Se detectaron errores mediante la aplicación de la Lista de Chequeo al artefacto Especificación de Requerimientos, obteniendo la evaluación de regular. Estos errores son faltas de ortografía que empañan la buena calidad y estructura que presenta el documento.

3.4 Validación de expertos. Método Delphi.

Entre los objetivos de la presente investigación se encuentra el de validar de forma externa los resultados alcanzados en la misma. Para llevar a cabo esta tarea, se hace necesario que cada experto revisando el trabajo previamente hecho, responda la siguiente encuesta:

1. ¿Considera usted que las técnicas de prueba aplicadas al módulo Inventario de Petróleo y Gas fueron las correctas?
2. ¿Considera usted que el proceso de pruebas realizado garantizará que el módulo se entregue libre de defectos?
3. ¿Considera que la investigación se pudiera utilizar como base para la realización del proceso de pruebas en los proyectos existentes en la Facultad?
4. En un rango de 1 a 5 valore la calidad de la investigación.
5. En un rango de 1 a 5 valore la facilidad de comprensión de la investigación.

Para darle respuesta a estas preguntas se seleccionaron 7 expertos. La selección fue a través de una serie de características y cualidades que ellos deben de poseer, entre las cuales se encuentran:

- Responsabilidad.
- Competencia.
- Creatividad.
- Seriedad.
- Honestidad.
- Capacidad de análisis.

Este proceso tiene como objetivo darle validez a la solución propuesta mediante la aplicación del cuestionario a la muestra seleccionada. Mediante la respuesta de este grupo de expertos se puede mejorar el proceso de aplicación de pruebas propuesto. La encuesta es enviada a los expertos mediante correo electrónico especificándoles que la misma es anónima y que tienen un plazo de 7 días para darle respuesta. Conjuntamente con la encuesta se envía el documento de la investigación, así como los entregables definidos en la misma para que la puedan consultar y estudiársela antes de responder las preguntas.

Mediante la aplicación de la encuesta se obtuvieron los siguientes resultados:

ENCUESTA	Expertos						
# Pregunta	1	2	3	4	5	6	7
1	No	Si	Si	No	Si	Si	Si
2	Si	Si	Si	Si	Si	Si	Si
3	Si	Si	Si	Si	Si	Si	Si
4	5	5	5	4	5	5	5
5	4	5	5	4	5	5	5

Como se evidencia en la tabla anterior, los expertos en sentido general consideran que la presente investigación se realizó con calidad, además consideran que las técnicas y los métodos utilizados para la realización del proceso de pruebas al módulo fueron las correctas. Dos expertos consideran que se deberían de haber realizado también pruebas de seguridad al módulo, así como las de carga y estrés. Plantean que estas son necesarias debido a la información que se maneja en él y a la cantidad de peticiones a las que tendrá que responder al mismo tiempo el sistema.

Conclusiones

Se han realizado las pruebas al módulo mediante los Casos de Prueba diseñados anteriormente. Esto permitió la detección de 37 No Conformidades en el mismo. Se han detectado errores en el artefacto Especificación de Requisitos mediante la Lista de Chequeo que se definió para aplicársele, principalmente errores de faltas de ortografía en el documento.

Se realizó un documento de No Conformidades donde quedaron registradas estas fallas presentes en el módulo. En un plazo de 72 horas el equipo de desarrollo debe darle respuesta a las No Conformidades presentadas, dando paso así a otra iteración del proceso de pruebas.

CONCLUSIONES

Para liberar un producto al concluir su desarrollo, se hace necesario verificar la calidad del mismo. En la presente investigación se le realizaron las pruebas al módulo Inventario de Petróleo y Gas, las cuales arrojaron errores que presentaba el módulo, evitándose así que llegara a las manos de los clientes en ese estado. De esta forma se le dio cumplimiento a los objetivos de la investigación, llegando a las siguientes conclusiones:

- Se verificó la correcta implementación de los requerimientos planteados por los clientes de la ONRM en un inicio, mediante las pruebas de caja blanca, utilizando la técnica de camino básico y las pruebas de caja negra, mediante la técnica de partición equivalente.
- Se logró una mayor organización en el equipo de pruebas para una correcta aplicación de las pruebas al módulo, mediante la elaboración del Plan de Pruebas.
- Se diseñaron Casos de Prueba para poder verificar las funcionalidades, así como los grafos de flujo de datos que posibilitaron comprobar los caminos lógicos en las funciones del código.
- Mediante la aplicación de las pruebas utilizando los Casos de Prueba diseñados se detectaron errores que son significativos y que no fueron detectados por los desarrolladores. En general el proceso de pruebas arrojó a la luz un total de 37 No Conformidades.
- Se detectaron 2 errores en el artefacto Especificación de Requisitos mediante la aplicación de la Lista de Chequeo. Principalmente faltas de ortografía que empañan la buena calidad y estructura del documento.
- Se documentaron los errores mediante un documento de No Conformidades para tener control sobre ellos y darle seguimiento por parte del equipo de pruebas y el equipo de desarrollo.

RECOMENDACIONES

Luego de haber finalizado el proceso de pruebas al módulo aplicándole las pruebas de caja negra con la técnica de partición equivalente y las pruebas de caja blanca con la técnica de camino básico y de haber comprobado que las mismas han arrojado a la luz errores en el sistema que no habían sido detectados en un inicio, se recomienda:

- Que se le realicen pruebas de rendimiento, de carga y de estrés al módulo.
- Que se continúen realizando iteraciones de las pruebas una vez que se finalice la implementación de los Casos de Uso que faltan.

Bibliografía referenciada

ALARCÓN, Armando Silva. 2004. Modelos de calidad. La industria del software en México. *Enterate*. [En línea] Universidad Nacional Autónoma de México, 01 de 2004. [Citado el: 17 de 06 de 09.] <http://www.enterate.unam.mx/Articulos/2004/Enero/modelos.htm>.

Bichachi, Dra. Diana Susana. 2002. Universidad del Salvador. [En línea] 2002. [Citado el: 3 de 4 de 2010.] http://www.salvador.edu.ar/vrid/iiefgs/tr_check_list.pdf.

Hanz. 2007. H@nz ...el Geek. [En línea] 19 de 4 de 2007. [Citado el: 28 de 11 de 2009.] <http://hancocchi.net/el-rol-del-analista-en-rup/>.

Int. RUP. 2005. Portal Desarrollo Software. [En línea] 22 de 09 de 2005. [Citado el: 24 de 11 de 2009.] <https://pid.dsic.upv.es/C1/Material/Documentos Disponibles/Introducción a RUP.doc>.

IT Institute. 2009. IT Institute - Avanced Information Technology Center . [En línea] 2009. [Citado el: 28 de 11 de 2009.] http://it-institute.org/index.php?option=com_content&task=view&id=145&Itemid=75.

Metodologías - RUP. 2007. Consultoría en Metodologías de Desarrollo de Software - RUP y las mejores prácticas para el desarrollo de software. *Hista Internacional S.A.* [En línea] 27 de 2 de 2007. [Citado el: 24 de 11 de 2009.] <http://www.histaintl.com/servicios/consulting/rup.php>.

ONRM. 2008. Oficina Nacional de Recursos Minerales. [En línea] 2008. [Citado el: 20 de 4 de 2009.] <http://www.onrm.minbas.cu>.

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

Sanchez, María A. Mendoza. 2007. Informatizate. [En línea] 7 de 6 de 2007. [Citado el: 28 de 11 de 2009.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

Bibliografía consultada

Acuña, César Javier. Kybele. Grupo de Investigación. [En línea] [Citado el: 09 de 12 de 2009.] <http://kybele.escet.urjc.es/documentos/ISG/%5BISG-2006-07%5DPuebasSoftware.pdf>.

ALARCÓN, Armando Silva. 2004. Modelos de calidad. La industria del software en México. *Enterate*. [En línea] Universidad Nacional Autónoma de México, 01 de 2004. [Citado el: 17 de 06 de 09.] <http://www.enterate.unam.mx/Articulos/2004/Enero/modelos.htm>.

Anna. C Grimán, María Pérez, Luis. E Mendoza. 2003. LISI. [En línea] 2003. [Citado el: 03 de 12 de 2009.] http://www.lisi.usb.ve/publicaciones/02%20calidad%20sistemica/calidad_09.pdf.

Bichachi, Dra. Diana Susana. 2002. Universidad del Salvador. [En línea] 2002. [Citado el: 3 de 4 de 2010.] http://www.salvador.edu.ar/vrid/iiefigs/tr_check_list.pdf.

ConexionIT. 2008. ConexionIT Comunidad de Profesionales de IT . [En línea] 20 de 12 de 2008. [Citado el: 28 de 11 de 2009.] <http://www.conexionit.com/blog/metodologias/que-es-rup.html>.

Espinosa, Dayris. 2009. *ARQUITECTURA DE SOFTWARE*. [Expediente del proyecto] 2009.

Fernández, José Almendros. 2009. Molinux. [En línea] 27 de 2 de 2009. [Citado el: 24 de 11 de 2009.] https://forja.molinux.info/frs/download.php/203/Spela_PRU_PlandePruebas_v3.0.doc.

Hanz. 2007. H@nz ...el Geek. [En línea] 19 de 4 de 2007. [Citado el: 28 de 11 de 2009.] <http://hancocchi.net/el-rol-del-analista-en-rup/>.

Int. RUP. 2005. Portal Desarrollo Software. [En línea] 22 de 09 de 2005. [Citado el: 24 de 11 de 2009.] <https://pid.dsic.upv.es/C1/Material/Documentos Disponibles/Introducción a RUP.doc>.

IT Institute. 2009. IT Institute - Avanced Information Technology Center . [En línea] 2009. [Citado el: 28 de 11 de 2009.] http://it-institute.org/index.php?option=com_content&task=view&id=145&Itemid=75.

Jose R. Vilar. 2008. El papel de las TIC en la empresa del SXXI. *Tecnobiz*. [En línea] 2008. [Citado el: 20 de 4 de 2009.] <http://www.tecnobiz.com/el-papel-de-las-tic-en-las-empresas>.

Metodologías - RUP. 2007. Consultoría en Metodologías de Desarrollo de Software - RUP y las mejores prácticas para el desarrollo de software. *Hista Internacional S.A.* [En línea] 27 de 2 de 2007. [Citado el: 24 de 11 de 2009.] <http://www.histaintl.com/servicios/consulting/rup.php>.

Myers, Glenford J. 1979. *The Art of the Software Testing.* 1979.

ONRM. 2008. Oficina Nacional de Recursos Minerales. [En línea] 2008. [Citado el: 20 de 4 de 2009.] <http://www.onrm.minbas.cu>.

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

Racionales, Tecnologías. 2007. Tecnologías Racionales. [En línea] 27 de 02 de 2007. [Citado el: 05 de 04 de 2010.] <http://www.tecnologiasracionales.com/servicios/consulting/rup.php>.

Salazar, Lisset. 2008. *ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE.* [Expediente del proyecto] 2008.

Sanchez, María A. Mendoza. 2007. Informatizate. [En línea] 7 de 6 de 2007. [Citado el: 28 de 11 de 2009.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

Zúñigas, Nieleidys. 2008. *MODELO DE SISTEMA.* [Expediente del proyecto] 2008.

GLOSARIO DE TÉRMINOS

CMM: Modelo de Capacidad y Madurez

CP: Casos de Prueba

CU: Caso de Uso

ONRM: Oficina Nacional de Recursos Minerales

PNICG: Programa Nacional de Informatización del Conocimiento Geológico

Requerimiento: Necesidad o solicitud

RUP: Proceso Unificado de Desarrollo

SGDG: Sistema de Gestión de Datos Geológicos

Testear: Probar

TIC: Tecnologías de la Información y las Comunicaciones

UCI: Universidad de las Ciencias Informáticas

ANEXOS

Anexo 1

Registro de defectos y dificultades detectados

Elemento	No	No conformidad	Aspecto correspondiente	Etapas de detección	Significativa	No Significativa	Recomendación	Estado NC	Respuesta del Equipo Desarrollo
Aplicación	1	Si no se ha iniciado sesión como administrador, el sistema no debe de mostrar las opciones de adicionar, modificar, ni de eliminar	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Prospecto 	1ra etapa	X			PD – (Pendiente)	
Aplicación	2	Cuando se cancela la opción de adicionar, el sistema no te regresa a la lista de los prospectos, se queda en la misma interfaz cuando debería de regresar a la lista	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Prospecto • Adicionar 	1ra etapa	X			PD – (Pendiente)	
Aplicación	3	Cuando en el campo del nombre se introducen caracteres especiales, o incorrectos, no se debería de adicionar el Prospecto. Ejemplo de caracteres en el nombre: ♦♣♠♦. Ver anexo 1	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Prospecto • Adicionar 	1ra etapa	X		Cuando ingrese a la sesión de adicionar, que en el campo de selección del Solicitante, me aparezca uno seleccionado por defecto, como se encuentra el campo de selección de	PD – (Pendiente)	

ANEXOS

							Hidrocarburo		
Aplicación	4	El sistema no muestra la interfaz para modificar el Prospecto, el enlace no funciona. No se pudo comprobar los otros aspectos del Caso de Prueba debido a este error.	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Prospecto • Modificar 	1ra etapa	X		Arreglar los vínculos rotos en la aplicación.	PD – (Pendiente)	
Aplicación	5	El sistema no elimina el Prospecto, muestra el mensaje para eliminarlo, confirmas su eliminación y aún así no lo elimina	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Prospecto • Eliminar 	1ra etapa	X			PD – (Pendiente)	
Aplicación	6	Si no se ha iniciado sesión como administrador, el sistema no debe de mostrar las opciones de adicionar, modificar, ni de eliminar	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Yacimiento 	1ra etapa	X			PD – (Pendiente)	
Aplicación	7	Cuando se cancela la opción de adicionar, el sistema no te regresa a la lista de los yacimientos, se queda en la misma interfaz cuando debería de regresar a la lista	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Yacimiento • Adicionar 	1ra etapa	X			PD – (Pendiente)	

ANEXOS

Aplicación	8	<p>Cuando en el campo del nombre se introducen caracteres especiales, o incorrectos, no se debería de adicionar el Yacimiento. Ejemplo de caracteres en el nombre: ♦♣♠♦</p>	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Yacimiento • Adicionar 	1ra etapa	X		<p>Cuando ingrese a la sesión de adicionar, que en el campo de selección del Solicitante, me aparezca uno seleccionado por defecto, como se encuentra el campo de selección de Hidrocarburo</p>	PD – (Pendiente)	
Aplicación	9	<p>El sistema no muestra la interfaz para modificar el Yacimiento, el enlace no funciona. No se pudo comprobar los otros aspectos del Caso de Prueba debido a este error.</p>	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Yacimiento • Modificar 	1ra etapa	X		<p>Arreglar los vínculos rotos en la aplicación.</p>	PD – (Pendiente)	
Aplicación	10	<p>El sistema no elimina el Yacimiento, muestra el mensaje para eliminarlo, confirmas su eliminación y aún así no lo elimina</p>	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Yacimiento • Eliminar 	1ra etapa	X			PD – (Pendiente)	
Aplicación	11	<p>Si no se ha iniciado sesión como administrador, el sistema no debe de mostrar las opciones de adicionar, modificar, ni de eliminar</p>	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Bloque 	1ra etapa	X			PD – (Pendiente)	

ANEXOS

Aplicación	12	Cuando se cancela la opción de adicionar, el sistema no te regresa a la lista de los Bloques, se queda en la misma interfaz cuando debería de regresar a la lista	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Bloque • Adicionar 	1ra etapa	X			PD – (Pendiente)	
Aplicación	13	Cuando en el campo del nombre se introducen caracteres especiales, o incorrectos, no se debería de adicionar el Bloque. Ejemplo de caracteres en el nombre: ♦♣♠♦	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Bloque • Adicionar 	1ra etapa	X			PD – (Pendiente)	
Aplicación	14	El sistema no muestra la interfaz para modificar el Bloque, el enlace no funciona. No se pudo comprobar los otros aspectos del Caso de Prueba debido a este error.	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Bloque • Modificar 	1ra etapa	X		Arreglar los vínculos rotos en la aplicación.	PD – (Pendiente)	
Aplicación	15	El sistema no elimina el Bloque, muestra el mensaje para eliminarlo, confirmas su eliminación y aún así no lo elimina	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Bloque • Eliminar 	1ra etapa	X			PD – (Pendiente)	

ANEXOS

Aplicación	16	Si no se ha iniciado sesión como administrador, el sistema no debe de mostrar las opciones de adicionar, modificar, ni de eliminar	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Pozo 	1ra etapa	X			PD – (Pendiente)	
Aplicación	17	Cuando se cancela la opción de adicionar, el sistema no te regresa a la lista de los Pozos, se queda en la misma interfaz cuando debería de regresar a la lista	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Pozo • Adicionar 	1ra etapa	X			PD – (Pendiente)	
Aplicación	18	Cuando en el campo del nombre se introducen caracteres especiales, o incorrectos, no se debería de adicionar el Pozo. Ejemplo de caracteres en el nombre: ♦♣♠♦. Ver anexo 2	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Pozo • Adicionar 	1ra etapa	X		En este caso los Pozos que se adicionan se ponen en el inicio de la lista, cuando en varias de las funcionalidades implementadas se muestran al final de la lista. Establecer un orden para todos.	PD – (Pendiente)	
Aplicación	19	No se muestra un botón cancelar para salir de esta interfaz en caso de que no se desee gestionar los datos.	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Pozo • Gestionar 	1ra etapa	X		Poner un botón de cancelar.	PD – (Pendiente)	

ANEXOS

Aplicación	20	El sistema valido que sean números los que se están entrando, pero en el caso de que se le entre un número grande, ejemplo 40 o más 1, aun así los introduce y si se pudiera tener ese número aun así la interfaz en un primer momento se colapsa, mostrando el número de una manera que no es la adecuada. Ver anexo 3, anexo 4	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Pozo • Gestionar 	1ra etapa			Los mensajes de esta interfaz no están acorde con los utilizados hasta el momento en la aplicación.	PD – (Pendiente)	
Aplicación	21	El enlace se encuentra roto, por lo que es imposible probar los demás escenarios para esta sección	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Pozo • Modificar 	1ra etapa	X		Arreglar todos los enlaces rotos de la aplicación.	PD – (Pendiente)	
Aplicación	22	El sistema muestra el mensaje para eliminar el Pozo y cuando se confirma el mensaje para eliminar, no se elimina	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Pozo • Eliminar 	1ra etapa	X			PD – (Pendiente)	

ANEXOS

Aplicación	23	Si no se ha iniciado sesión como administrador, el sistema no debe de mostrar las opciones de gestionar	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Datos Primarios Prospecto 	1ra etapa	X			PD – (Pendiente)	
Aplicación	24	No existe un botón de cancelar la gestión de los datos en caso de que se decida no hacerlo	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Datos Primarios Prospecto • Gestionar Datos 	1ra etapa	X			PD – (Pendiente)	
Aplicación	25	Se encuentra bien validado para números pero en caso de que se introduzca un número muy grande, el sistema no lo valida y debido a esto la interfaz colapsa en un inicio. El mensaje de eliminar no está acorde con los demás mensajes del sistema. Ver anexo 4, Anexo 5	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Datos Primarios Prospecto • Gestionar Datos 	1ra etapa	X		EL botón de Guardar datos no se muestra con el estándar para el diseño de interfaz de usuario definido en el módulo. También muestra el nombre de Guardar Datos del Pozo, cuando debería ser Guardar Datos del Prospecto	PD – (Pendiente)	

ANEXOS

Aplicación	26	Si no se ha iniciado sesión como administrador, el sistema no debe de mostrar las opciones de gestionar. No se muestra la opción de buscar, ni el paginado que es estándar para las demás funcionalidades	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Producción 	1ra etapa	X			PD – (Pendiente)	
Aplicación	27	No aparecen los datos de los Yacimientos en la lista desplegable, existiendo Yacimientos en el sistema. Lo mismo con el CoboBox de Pozo Natural. Ver anexo 6	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Producción 	1ra etapa	X			PD – (Pendiente)	
Aplicación	28	El botón de cancelar no funciona, no te lleva a la interfaz que contiene una lista con los datos	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Producción • Adicionar 	1ra etapa	X			PD – (Pendiente)	
Aplicación	29	El enlace no funciona, por tal motivo no se pueden comprobar las demás secciones de este escenario	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Producción • Modificar 	1ra etapa	X		Arreglar el enlace	PD – (Pendiente)	

ANEXOS

Aplicación	30	Elimina correctamente, pero el mensaje que muestra no es el que se utiliza en la aplicación en las demás funcionalidades. Ver anexo 7	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Producción • Eliminar 	1ra etapa	X				PD – (Pendiente)	
Aplicación	31	Si no se ha iniciado sesión como administrador, el sistema no debe de mostrar las opciones de gestionar. No se muestra la opción de buscar, ni el paginado que es estándar para las demás funcionalidades	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Reservas Calculadas 	1ra etapa	X				PD – (Pendiente)	
Aplicación	32	No funciona el Adicionar. El botón de siguiente no funciona. No se pueden probar las demás funcionalidades	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Reservas Calculadas • Adicionar 	1ra etapa	X				PD – (Pendiente)	
Aplicación	33	El enlace del modificar no funciona. No se pueden probar las demás funcionalidades	<ul style="list-style-type: none"> • Principal • Inventario Petróleo • Gestión • Reservas Calculadas • Modificar 	1ra etapa	X				PD – (Pendiente)	

ANEXOS

Aplicación	34	Muestra el mensaje para eliminar y cuando se confirma no elimina los datos	<ul style="list-style-type: none"> Principal Inventario Petróleo Gestión Reservas Calculadas Eliminar 	1ra etapa	X			PD – (Pendiente)	
------------	----	--	--	-----------	---	--	--	------------------	--

Tabla 15: Registro de defectos y dificultades detectados

Anexo 2

SGDG Sistema de Gestión de Datos Geológicos
"El conocimiento geológico de Cuba al servicio de la sociedad"
Martes, 16 de Marzo del 2010
Inicio Portal Iniciar Sesión
Usted está en: Inicio > Balance del petróleo > Módulo para la gestión de prospectos >
Balance-Petróleo Gestionar Prospecto

Nombre	Solicitante	Acciones
Prospecto 2	PetroCuba	[Editar] [Eliminar] [Ver]
asd	yhtjt	[Editar] [Eliminar] [Ver]
PruebaHR	Gasrom	[Editar] [Eliminar] [Ver]
vhghgh	yhtjt	[Editar] [Eliminar] [Ver]
aaaa	tghj	[Editar] [Eliminar] [Ver]
#sddf5*	Petrolsa	[Editar] [Eliminar] [Ver]
Esto No Sirveee	yhtjt	[Editar] [Eliminar] [Ver]
	Petrolsa	[Editar] [Eliminar] [Ver]

Figura 6: Nombre incorrecto en Gestionar Prospecto

Anexo 3

Gestionar Pozos Naturales

Nombre	Yacimiento	Solicitante	Acciones
Tampoco sirve ¿porque de primero?	loco	Eddy	[Editar] [Eliminar] [Ver]
Tampoco sirve	loco	Eddy	[Editar] [Eliminar] [Ver]
Tampoco sirve	loco	Eddy	[Editar] [Eliminar] [Ver]
aaaaa	loco	Eddy	[Editar] [Eliminar] [Ver]
locura	loco	Eddy	[Editar] [Eliminar] [Ver]
gdsgh	loco	Eddy	[Editar] [Eliminar] [Ver]
dd	loco	Eddy	[Editar] [Eliminar] [Ver]
ssssiiii	sdf	Leyaaa	[Editar] [Eliminar] [Ver]
eee	figu	PDV5a	[Editar] [Eliminar] [Ver]
prueba	zzzz	PetroleoG	[Editar] [Eliminar] [Ver]

Figura 7: Nombre incorrecto en Gestionar Pozo Natural

Anexo 6

Gestionar Datos de Prospecto Natural

Intervalos

Intervalo	% Porosidad	% Saturación	Permeabilidad	Espesor Efectivo	Acciones
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="+"/>
3	3	3	3	3	<input type="button" value="X"/>

Datos generales

Probabilidad de Éxito:

Porosidad Efectiva:

Cantidad Hidrocarburo (Q):

Área:

Coefficiente de Recálculo:

Factibilidad Económica: Sí No

Saturación Efectiva:

Espesor efectivo:

Costo:

Método Utilizado:

Datos Datos no validados

Figura 10: Botón con formato y nombre distintos

Anexo 7

SGDG Sistema de Gestión de Datos Geológicos

"El conocimiento geológico de Cuba al servicio de la sociedad"

Martes, 16 de Marzo del 2010

Inicio Portal Iniciar Sesión

Usted está en: Inicio > Balance del petróleo > Módulo para la gestión de acumulados de producción >

Balance-Petróleo

Gestionar Acumulados de Producción

Buscar Adicionar

Datos Generales

Solicitante:

Yacimiento natural:

Pozo natural:

Gestión

- Prospectos
- Yacimientos
- Bloques
- Pozos natural
- Primarios de Prospectos
- Producción
- Reservas calculadas

Consultas

Reportes

Figura 11: No muestra los Yacimientos

Anexo 8

The screenshot displays the SGDg (Sistema de Gestión de Datos Geológicos) web application. The header includes the logo and the slogan "El conocimiento geológico de Cuba al servicio de la sociedad". The date is "Martes, 16 de Marzo del 2010". The user is logged in as "Usted está en: Inicio > Balance del petróleo > Módulo para la gestión de acumulados de producción". The main content area is titled "Gestionar Acumulados de Producción" and features a table with the following data:

Fecha	Pozo	Yacimiento	Solicitante	Producción Mensual	Acciones
26-02-2002	prueba	zzzz	zzzz	5	[Edit] [Delete] [List]

A confirmation dialog box is overlaid on the table, with the title "La página en http://sgdg.uci.cu dice:" and the message "¿Está seguro que desea eliminar el acumulado seleccionado?". The dialog has "Aceptar" and "Cancelar" buttons.

Figura 12: Mensaje distinto a los demás del sistema