

Universidad de las Ciencias Informáticas

Facultad 4 y 8



**Plan de Pruebas de Módulo para la Unidad de Compatibilización,
Integración y Desarrollo de Productos Informáticos para la Defensa.**

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

AUTORES:

Yudisleidis Báez Ortiz.

Annalie Pacheco Carrió.

TUTOR:

Ing. Yusmary Trujillo Rodríguez.

CIUDAD DE LA HABANA, JULIO 2008.



DECLARACIÓN DE AUDITORÍA

Declaramos ser las autoras del presente trabajo de diploma y reconocemos la Unidad de Compatibilización, Integración y Desarrollo de Productos Informáticos para la Defensa (UCID) los derechos patrimoniales del mismo, con carácter exclusivo.

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

Autores:

Annalie Pacheco Carrió.

Yudisleidis Báez Ortiz.

Tutor:

Ing. Yusmary Trujillo Rodríguez.



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

Ernesto Che Guevara.

De Annalie:

A mi abuelo Mañi a quien quiero mucho por ayudarme a realizar este sueño y por ser mi padre. A mi mamá por su apoyo en todo momento, tanto bueno como malo, por escucharme, ser mi ejemplo y ser tan especial para mí. A mi hermano por el que siempre me preocupo y le doy gracias a dios porque me vea graduada. A mi novio a quien amo mucho, que siempre ha estado a mi lado por ser mi gran apoyo, por quererme y ayudarme a superar las vicisitudes por las que he pasado. A mis tíos Reno y Yami y mis primos Adrian y Annie por darme preocuparse por mí. A la melliza por ayudarme y atenderme en los momentos que la he necesitado. A mi compañera de tesis Yudisleidis Báez por su ayuda y apoyo que aunque nos conocemos hace poco tiempo hemos sabido confiar la una en la otra. A todas aquellas personas que me han ayudado a continuar estos cinco años, que han estado al pendiente de mis problemas y que han sabido brindarme su mano. A mi tutora Yusmary Trujillo.

A todas las personas que han hecho realidad el sueño de crear una Universidad como la que hoy tenemos...

Yudisleidis:

A mi mamá que siempre estuvo en los malos y buenos momentos de mi vida, a quien amo y respeto. A mi papá que siempre tuve su comprensión y su ejemplo en todo momento, siempre he querido ser como el. A mi abuelita Melba, a mi tía Tere, a mi tío Rey agradecerles por sus consejos, su cariño, a mi primo Diego que aunque es muy pequeño siempre he querido ser un ejemplo para el. A mi amiga Arietty que más que amiga es hermana para mí. A las hermanas Cros, Taimi, Tania, Tamara que siempre me han dado su apoyo. A mi tío Coca y Salva por todo su cariño. A María que siempre me ha tenido presente. A mi familia y a todos mis amigos y amigas que compartieron trabajo, amor, sonrisas, consejos y momentos inolvidables, en especial aquellas personitas que estuvieron a mi lado cuando más lo necesite María Caridad, Yuricel y Suleydis. A mi familia en Holguín que siempre me ayudaron en lo que pudieron...a la familia Blaya gracias. A mi tutora Yusmary Trujillo, a mi amiga, profesora y compañera Violena.

A mi compañera de tesis Annalie Pacheco que a pesar del poco tiempo que nos conocemos la puedo catalogar como mi amiga ahora y siempre.

En fin para no olvidar a nadie, a todos muchas gracias.

De Annalie:

A mi abuela Mima a quien la vida no le dio la oportunidad de verme graduada pero se que si estuviera conmigo se sentiría muy orgullosa de verme convertida en una profesional. A mi mamá Farita a quien quiero mucho, para que se sienta muy orgullosa de mí. A mi hermano Raulier (Papucho) para el que siempre he querido ser un ejemplo, para que en un futuro se convierta en un profesional también. A mi abuelo Mañi por ser el motor impulsor en mi vida, por creer en mí, porque a pesar que es mi abuelo tomó el lugar de mi padre, porque gracias a el nunca me ha faltado nada en la vida ni material ni espiritual. A mi novio Michel Hernández por ser especial para mí, porque en estos tres años he aprendido mucho de el. A mis tíos Reno y Yami y mis primos Adrian y Annie porque son especiales para mí, para que mis primos tomen mi ejemplo y se conviertan en profesionales. Dedico este trabajo de diploma a ustedes de forma especial por ayudarme a convertir mi sueño en realidad.

De Yudisleidis:

A mi mamita linda que siempre ha sido su anhelo ver a su niña graduada, que siempre ha estado a mi lado tanto en los malos, como en los buenos momentos, que me ha dado fuerza para enfrentar la vida, a mi papito que ha luchado para poder estar a mi lado en estos momentos, que siempre ha sido un sueño para mi ser como él, a mis abuelitos Rolando, Olga y Macho que aunque hace mucho que no están a mi lado, se que me han acompañado siempre, a mi abuelita Melba y mi tía Tere que siempre estarán orgullosa de mí. Dedico este trabajo de diploma a ustedes de forma especial por ayudarme a convertir mi sueño en realidad.

RESUMEN

En el presente trabajo se aborda la realización de un Plan de Pruebas¹ de Módulo que servirá de guía a cada probador, el cual será aplicado a los diversos módulos que se realizan en la Unidad de Compatibilización, Integración y Desarrollo de Productos Informáticos para la Defensa.

Para la realización del mismo se efectuó un estudio en la UCID llegándose a la conclusión de que no existe una estrategia² de prueba que logre guiar a cada uno de los probadores cuando se le asigna un sistema a verificar. La UCID no cuenta con Listas de Chequeo que verifiquen los documentos que se plasman en el Expediente de Proyecto por lo que se decidió realizar Listas de Chequeo que comprueben la documentación que se crea por Flujos de Trabajo, y de manera funcional se realizaron plantillas para Pruebas de Funcionalidad, Interfaz de Usuario, Atributos de Calidad y Diseño de Casos de Prueba con el objetivo de lograr una correcta aplicación del Procedimiento de Pruebas por parte del equipo de trabajo del Proyecto de Calidad de la UCID.

Como resultado de este trabajo se obtiene una estrategia de prueba que permite obtener un exitoso procedimiento³ al probar determinados sistemas.

Palabras Claves: Plan de Pruebas, Listas de Chequeo, Diseño de Casos de Prueba, Pruebas Funcionales, Interfaz de Usuario y Atributos de Calidad.

ÍNDICE

RESUMEN.....	VII
INDICE.....	VIII
INDICE DE TABLAS	XII
INTRODUCCION	1
Capítulo 1.Fundamentación Teórica.	5
1.1 Introducción	5
1.2 Ingeniería de pruebas.	6
1.2.1 Antecedentes de las pruebas destinadas a un software.....	6
1.2.2 Prueba de Software	6
1.2.3 ¿Qué es probar?.....	7
1.3 Las pruebas en el desarrollo de un software.....	8
1.3.1 Objetivo de las pruebas	8
1.3.2 Principios de pruebas.	8
1.3.3 Características de las pruebas.....	8
1.3.4 Estrategia de Prueba.	9
1.3.4.1 Características genéricas de la Estrategia de Prueba:.....	9
1.3.4.2 Los objetivos de la Estrategia de Prueba son:	9
1.3.4.3 La Estrategia de Prueba suele seguir estas etapas:	10
1.4 Elementos fundamentales del Proceso de Pruebas.....	10
1.4.1 Niveles de pruebas	10
1.5 Métodos de pruebas	13
1.6 Tipos de pruebas	15
1.7 Primera Fase	16
1.7.1 Pruebas de Módulo.....	16
1.7.1.1 Revisión de los Manuales entregables al Usuario.....	16
1.7.1.2 Revisión de la documentación en el Flujo de Trabajo de Negocio.	17
1.7.1.3 Revisión de la documentación en el Flujo de Trabajo Requisitos.....	17
1.7.1.4 Revisión de la documentación en el Flujo de Trabajo Análisis y Diseño.	18

1.7.1.5	Revisión de la documentación en el Flujo de Trabajo Implementación.	18
1.7.1.6	Pruebas de Funcionalidad y Diseño de Casos de Prueba.....	19
1.7.1.7	Pruebas para verificar Interfaz de Usuario y Atributos de Calidad.....	21
1.8	Segunda Fase	22
1.8.1	Revisión para certificar la corrección de errores detectados en la Primera Fase.....	22
1.8.2	Revisión para detectar nuevos errores introducidos.....	22
1.9	Conclusiones	23
Capítulo 2.Propuesta de guía de revisiones técnicas y prueba.....		24
2.1	Introducción	24
2.2	El proceso de pruebas en RUP (Proceso Unificado de desarrollo).....	24
2.3	Listas de Chequeo	30
2.3.1	Criterio Real de cada Lista de Chequeo.....	30
2.3.2	Revisión de los Manuales entregables al Usuario.....	31
2.3.3	Listas de Chequeo para la revisión de la documentación en el Flujo de Trabajo del Negocio.....	31
2.3.4	Listas de Chequeo para la revisión de la documentación del Flujo de Trabajo Requisitos.....	35
2.3.5	Listas de Chequeo para la revisión de la documentación del Flujo de Trabajo Análisis y Diseño.....	36
2.3.6	Listas de Chequeo para la revisión de la documentación del Flujo de Trabajo Implementación.....	38
2.3.7	Listas de Chequeo para la documentación de las Pruebas de Funcionalidad.....	38
2.3.8	Pruebas para verificar la documentación de la Interfaz de Usuario y Atributos de Calidad.....	39
2.4	Diseño de Casos de Prueba	39
2.6	Documento de No Conformidad.....	43
2.7	Plan de Pruebas	44
2.7.1	Estructura del Plan de Pruebas de Módulo	45
2.7.2	Estrategia de prueba.....	45
2.7.3	Métodos de prueba.....	46

2.7.4	Técnicas de pruebas de Caja Negra.....	46
2.7.5	Configuración del entorno de prueba.....	47
2.8	Procedimientos de prueba.....	47
2.9	Elementos a revisar del Modelo de Negocio.....	48
2.10	Elementos a revisar en Requisitos.....	48
2.10.1	Modelo de Casos de Uso del Sistema.....	48
2.10.2	Especificación de Requisitos.....	49
2.10.3	Plan de Gestión de Requisitos.....	50
2.11	Elementos a revisar en Análisis y Diseño.....	50
2.11.1	Modelo de Análisis.....	51
2.11.2	Modelo de Diseño.....	51
2.11.3	Documento de Arquitectura de Software.....	52
2.12	Elementos a revisar en Implementación.....	53
2.12.1	Modelo de Despliegue.....	53
2.13	Manuales a revisar del sistema.....	54
2.13.1	Manual de Usuario.....	54
2.13.2	Manual de Instalación.....	55
2.13.3	Manual de Tecnología.....	55
2.14	Elementos a revisar mediante las Pruebas de Funcionalidad.....	55
2.14.1	Pruebas Funcionales.....	55
2.14.2	Prueba de Seguridad.....	55
2.14.3	Prueba de Volumen.....	55
2.15	Elementos a revisar en la Interfaz de Usuario y en los Atributos de Calidad que este presenta.....	56
2.15.1	Interfaz de Usuario.....	56
2.15.2	Atributos de Calidad.....	56
2.16	Elementos del Plan de Prueba.....	56
2.16.1	Plan de Prueba.....	56

2.17	Plantilla de Diseño de Casos de Prueba.....	57
2.17.1	Diseño de Casos de Prueba	57
2.18	Conclusiones	57
Capítulo 3.Resultados al aplicar las Pruebas		58
3.1	Introducción.....	58
3.2	Elementos a probar, análisis de los resultados de las pruebas realizadas del Flujo de Trabajo de Negocio.....	58
3.3	Elementos a probar, análisis de los resultados de las pruebas realizadas del Flujo de Trabajo de Requisitos.....	67
3.4	Elementos a probar, análisis de los resultados de las pruebas realizadas del Flujo de Trabajo de Análisis y Diseño.....	68
3.5	Elementos a probar, análisis de los resultados de las pruebas realizadas al Flujo de Trabajo de Implementación.....	68
3.6	Elementos a probar, análisis de los resultados de las pruebas realizadas en los Manuales.....	69
3.7	Elementos a probar y análisis de los resultados de las Pruebas al Sistema.....	69
3.8	Segunda Fase. Respuesta a las No Conformidades.....	70
3.9	Resultados Obtenidos.....	74
3.10	Conclusiones	76
RECOMENDACIONES		78
REFERENCIAS BIBLIOGRÁFICAS		79
BIBLIOGRAFÍAS.....		80
ANEXOS.....		81
Anexo#1:	Proceso de Prueba.....	81
Anexo # 2:	Plan de Prueba de Módulo.....	82
Anexo # 3:	Plan de Prueba de Módulo.....	92
Anexo # 4:	Diseño de Casos de Prueba.....	101
GLOSARIO DE TÉRMINOS.....		106

ÍNDICE DE TABLAS

Tabla 1: Cantidad de valores que presenta la Lista de Chequeo.30

Tabla 2 : Por ciento Real de la Lista de Chequeo a través del Criterio de Evaluación.75

ÍNDICE DE FIGURAS

Figure 2: Por ciento reales de cada artefacto evaluado75

INTRODUCCIÓN

El uso de la tecnología es un factor determinante en los niveles de eficiencia y competitividad tanto desde el punto de vista empresarial como personal. A nivel mundial, la informática y las comunicaciones han logrado alcanzar una gran relevancia, debido al desarrollo de nuevas tecnologías que caracterizó las últimas décadas del siglo XX y que mantienen un ritmo en ascenso durante este siglo. La informática se ha convertido en una base imprescindible para el trabajo del resto de las ciencias y de cualquier actividad del mundo actual, abarcando todas las esferas de trabajo.

Las tecnologías de la informática y las comunicaciones han logrado tan alto grado de desarrollo que hoy esta en la mayoría de las actividades del ser humano: en la educación, la industria, el comercio, las finanzas, la investigación, la salud, el deporte y la cultura. Hoy en día, conocer la tecnología y utilizarla además de construir un privilegio, es una necesidad.

A la hora de elaborar cualquier producto software, es muy importante destacar lo imprescindible que se torna formar un sistema, que cumpla con la aceptación de los requisitos dictados por el cliente, además de cumplir con los requisitos propios del producto lo cual esta referenciado en la Calidad del Software. En nuestros días la Calidad del Software constituye varios criterios los cuales están enmarcados en la gran responsabilidad que define la entrega de un producto de alto nivel, del cual se obtenga un software fiable, económico y que funcione en máquinas reales.

Es necesario destacar que la Calidad de un Software es un logro importante ya que el aseguramiento de la calidad de un sistema no radica únicamente en la calidad del producto sino también de la calidad del proceso⁴. Para ello, es necesario tener en cuenta un enfoque sistémico que permita adoptar una calidad orientada a la eficiencia/efectividad, tanto de los procesos como del producto, desde las perspectivas del cliente y del usuario. Tomando en cuenta que, desde un punto de vista general un elemento crítico para definir la calidad de un software y de gran importancia a la hora de detectar sus defectos son las Pruebas.

Las Pruebas de Software en aplicaciones han evolucionado mucho más allá de simplemente depurar la aplicación y posteriormente migrarla a producción para convertirse en un proceso integrado, automatizable y administrable que se realiza desde el momento del inicio de la codificación y se extiende hasta la entrega final al cliente. Las Pruebas de Software constituyen la principal fuente de

información sobre el nivel de calidad del sistema, y la confianza que deben tener los responsables en sus procesos. Aplicar una estrategia de pruebas es un paso imprescindible a la hora de liberar un sistema, es por ello que se torna tan importante crear un equipo de trabajo que ponga en práctica un procedimiento de pruebas que verifique el correcto funcionamiento del producto que se esta realizando.

Actualmente, en la Universidad de las Ciencias Informáticas (UCI), se creó un centro destinado a las FAR que se denominó Unidad de Compatibilización, Integración y Desarrollo de Productos Informáticos para la Defensa (UCID) el cual fue inaugurado el pasado año 2007. En dicha institución a raíz de su premura no existe un procedimiento de prueba que oriente al personal de calidad de cómo llevar a cabo un proceso organizado de las diferentes pruebas por fases.

En la UCID se lleva a cabo el proceso de revisiones técnicas al Expediente del Proyecto a través de los conocimientos de los elementos que componen cada plantilla gestionada en este expediente y el nivel de profesionalidad de los revisores técnicos, confeccionándose para la constancia de la revisión registros de No Conformidades para cada entregable en caso de inconformidades o recomendaciones. Para una mayor rapidez y eficiencia de este proceso lo mejor sería aplicar una Lista de Chequeo a cada entregable que mida su Calidad y que no se queden elementos importantes sin probar, con las cuales no se cuenta en este momento.

En el Flujo de Trabajo de Pruebas se realizan pruebas funcionales con el objetivo de comprobar la funcionalidad del sistema; para ejecutar este proceso correctamente y que se puedan probar la mayor cantidad de combinaciones de datos posibles se diseñan casos de prueba para cada caso de uso. No se cuenta con Listas de Chequeo que puedan medir los Atributos de Calidad y la Interfaz de Usuario, además que permitan comprobar si el Manual de Usuario, de Instalación y Tecnología pueden emplearse con el fin con que fueron realizados.

Esta **situación problemática** presupone:

- Falta de especificación de los objetivos y estrategias a seguir a la hora de aplicar las Pruebas de Módulo.
- Desconocimiento del personal de la importancia de aplicar un Proceso de Prueba organizado en los módulos de la UCID.
- No existe una guía de revisión para los documentos establecidos en el Expediente de Proyecto implantado en la UCID.

Estas limitantes conllevan a que las condiciones laborales del proyecto de Calidad de la Unidad de Compatibilización, Integración y Desarrollo de Productos Informáticos para la Defensa (UCID) no sean las ideales. Lo cuál motiva a plantear el siguiente problema de investigación:

Problema de Investigación:

¿Cómo mejorar el procedimiento de prueba en la Unidad de Compatibilización, Integración y Desarrollo de Productos Informáticos para la Defensa (UCID).?

Para resolver el problema antes identificado es necesario establecer como **objeto de estudio** el Proceso de Pruebas, en el desarrollo de software.

De ello se deriva que, el **campo de acción** que abarca este trabajo es: Laboratorio de Calidad de la Unidad Compatibilización, Integración y Desarrollo de Productos Informáticos para la Defensa (UCID).

Hipótesis:

Si se aplica el Plan de Pruebas de Módulo, entonces se logrará mayor organización en la realización del Proceso de Pruebas de Módulo.

El **objetivo general** que se persigue con la realización de este trabajo es: Establecer un Plan de Pruebas de Módulo en la Unidad de Compatibilización, Integración y Desarrollo de Productos Informáticos para la Defensa (UCID).

Para dar cumplimiento al objetivo general se realizaron las siguientes **tareas de la investigación**:

- Realizar un estudio sobre los aspectos teóricos conceptuales y prácticos de las diferentes pruebas aplicadas a un software.
- Realizar el estudio de planes de pruebas aplicadas a diversos sistemas en la Universidad de las Ciencias Informáticas (UCI).
- Realizar el estudio y el análisis del Expediente de Proyecto propuesto en el Laboratorio de Liberación de la Universidad de las Ciencias Informáticas.
- Realizar un estudio de las disímiles Pruebas de Módulo que se aplican en la Universidad de las Ciencias Informática.
- Realizar un estudio de las disímiles Listas de Chequeo y Diseño de Casos de Prueba que se aplican en la Universidad de las Ciencias Informática.

Acciones:

- Confeccionar Listas de Chequeo.
- Confeccionar Documento de No Conformidad.
- Confeccionar Criterios de Evaluación para las Listas de Chequeo.
- Diseñar Casos de Prueba.
- Evaluar los resultados de las Pruebas de Módulo.

Según observaciones y entrevistas a los equipos revisores del centro se identificó que no existe un Plan de Pruebas de Módulo que organice y regule las pruebas aplicadas a los disímiles proyectos, para llevar a cabo las revisiones no se cuenta con una guía o una Lista de Chequeo que permita evaluar la Calidad de los documentos generados en cada flujo de trabajo del proceso de ingeniería del software. Por lo que se puntualizó la realización de un Plan de Pruebas de Módulo que oriente la evaluación de la calidad, Listas de Chequeo para cada plantilla del Expediente del Proyecto y un procedimiento de pruebas que garantice la eficiencia del producto desarrollado.

Este trabajo está dividido en tres capítulos que contienen toda la información referente a la investigación realizada y a las pruebas efectuadas a los distintos módulos de los proyectos Comités Militar, Contabilidad Material y Planificación Material y Financiera, la información está distribuida de la siguiente manera:

Capítulo 1: Fundamentación Teórica. En este capítulo se exponen los distintos tipos de pruebas que serán aplicados para la solución del problema.

Capítulo 2: Propuestas de guía y revisiones técnicas y pruebas.

- En este capítulo se hará referencia a las Listas de Chequeo realizadas para cada documento.
- Criterios de Evolución realizados para cada una de las Listas de Chequeo.
- Procedimiento de Prueba. Diseño de Casos de Prueba

Capítulo 3: Resultados al aplicar las Pruebas. En este capítulo se hará referencia al resultado obtenido al aplicarle las pruebas a los módulos Centro de Balance, Reclutamiento, Recepción y Reserva de los proyectos Comités Militar, Contabilidad Material y Planificación Material y Financiera.

Fundamentación Teórica

1.1 Introducción

En el presente capítulo se abordarán las tendencias⁵ y técnicas a estudiar para desarrollar la propuesta de solución. En el mismo se expondrán las diferentes definiciones de los tipos de pruebas, niveles, estrategias y procesos, entre otros aspectos que serán de gran utilidad para el desarrollo del trabajo. Este Proceso de Pruebas está compuesto por dos fases, la Primera Fase provee una visión de las pruebas que luego serán aplicadas y la Segunda Fase proporciona una pequeña revisión a los errores encontrados en la fase anterior.

A lo largo de este capítulo se pretende definir aspectos importantes y necesarios para el desarrollo del trabajo. No se podría comenzar la realización de las Pruebas de Módulo sino tenemos presente: ¿Qué es la Calidad de Software? y ¿Qué es el Proceso de Prueba?

¿Qué es la Calidad de Software?

Definiciones:

- La concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares⁶ de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente [1].
- Concordancia del software producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente que desea el usuario [2].
- El grado con que un sistema, componente o proceso cumple los requisitos especificados y las necesidades o expectativas del cliente o usuario [3].
- La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad. [4].

Con el estudio de los conceptos anteriormente expuestos se puede llegar a la conclusión, que la Calidad de Software son un conjunto de condiciones la cual el sistema o software debe cumplir, con el objetivo que este plasme todos los requisitos impuestos por el usuario para que el mismo quede satisfecho con el trabajo de los desarrolladores.

¿Qué es el Proceso de Prueba?

El Proceso de Prueba es un proceso usado para identificar posibles fallos de implementación, calidad, o usabilidad de un programa de ordenador, tiene como objetivo asegurar la aceptación de un sistema, a través de la verificación del cumplimiento de todos los requerimientos especificados y acordados, así como también el cumplimiento de los estándares de presentación y uso de la aplicación establecida. El Proceso de Prueba repasa las ideas principales sobre las Pruebas del Software. Los pasos a seguir en este proceso se encuentra especificado detalladamente en el Anexo 1.

1.2 Ingeniería de pruebas.

1.2.1 Antecedentes de las pruebas destinadas a un software.

El Proceso de Pruebas de Software era considerado como una actividad que desarrollaba el programador para encontrar las fallas a los productos que iba construyendo. Con el paso de los años se ha hecho evidente la importancia que tiene desarrollar productos de software empleando el mínimo de tiempo y costo, que a su vez cuenten con la Calidad requerida, por lo tanto se ha tomado como propósito principal evaluar la funcionalidad del software respecto a los requisitos establecidos en el inicio del proyecto.

Actualmente las especificaciones de pruebas se realizan al mismo tiempo que el diseño del software, por lo que se propone iniciar el análisis de las pruebas junto con el análisis del software. Se plantea entonces realizar sondeos preventivos que permitan ejecutar las pruebas tan pronto el software este listo y con ello no solo descubrir los errores sino también poder evitarlos.

En nuestros días se calcula que la fase o proceso de pruebas representa más de la mitad del coste de un programa, en vistas de que para el mismo se requiere de un tiempo similar al de la programación, lo que obviamente acarrea⁷ un alto costo económico [5].

1.2.2 Prueba de Software

Definiciones:

- “Una actividad en la cual un sistema o uno de sus componentes se ejecuta en dos o más circunstancias previamente especificadas, los resultados se observan y registran y se realiza una evaluación de algún aspecto”. Probar, por lo tanto, es el proceso de ejecutar un programa con el fin de encontrar errores o fallas. [6].
- “La prueba del software es un elemento crítico para la garantía de calidad del software y representa una revisión de las especificaciones del diseño y de la codificación”. [2].

- “Un conjunto de herramientas, técnicas y métodos que hacen la excelencia del desempeño de un programa. Las técnicas para encontrar problemas en un programa son extensamente variadas y van desde el uso del ingenio por parte del personal de prueba hasta herramientas automatizadas que ayudan a aliviar el peso y el costo de tiempo de esta actividad”. [7].
- “Las pruebas consisten en la verificación dinámica del comportamiento de un programa en una serie finita de casos de pruebas”. [8].

Con estas definiciones se puede concluir que las Pruebas de Software son una serie de actividades que se realizan con el propósito de encontrar los errores en un sistema de software, y así poner a prueba el comportamiento del mismo.

1.2.3 ¿Qué es probar?

Probar un programa es ejercitarlo con la peor intención de encontrarle fallas. La necesidad de hacer las pruebas se evidencia en que son enormes las posibilidades de que se cometan errores por los desarrolladores, dada la imposibilidad de trabajar y comunicarse de manera perfecta, e incluso en ocasiones estas fallas pueden estar dadas por cuestiones de infraestructura, o sea que el hardware, el sistema operativo o las herramientas que se utilizan durante el proceso de desarrollo del software introduzca algunos defectos o simplemente permitan que el programador introduzca errores que no sean detectados. [9].

Definiciones:

- “Una actividad en la cual un sistema o uno de sus componentes se ejecuta en circunstancias previamente especificadas, los resultados se observan y registran y se realiza una evaluación de algún aspecto”. [10].
- “Proceso de ejecutar un programa con el fin de encontrar errores”. [11].

Probar un programa es ver si las condiciones en que se crearon para la funcionalidad del sistema se correspondan con sus expectativas. Aunque en la prueba lo fundamental es encontrar y exponer las debilidades en el software. Podemos suponer que por muy bien que este el trabajo pueden existir dificultades aunque mínimas ya que las pruebas no pueden asegurar la ausencia de defectos; solo pueden demostrar que existen defectos en el software.

1.3 Las pruebas en el desarrollo de un software

1.3.1 Objetivo de las pruebas

Las pruebas tienen como objetivo encontrar los defectos que presente el software, una prueba tiene éxito si descubre un defecto y esta fracasa si hay defectos y no se descubren. El objetivo específico de la fase de pruebas es encontrar la mayor cantidad posible de errores, con un mínimo de costo y tiempo. Entre sus funciones están:

- Encontrar y documentar los defectos que puedan afectar la Calidad del Software.
- Validar que el software trabaje como fue diseñado.
- Validar y probar en forma de demostración los requisitos que debe cumplir el software.
- Validar que los requisitos fueron implementados correctamente.

1.3.2 Principios de pruebas.

Para realizar una buena aplicación de los diferentes métodos de prueba que existen, se hace necesario que los ingenieros que trabajan en esta línea tengan en cuenta una serie de principios que ayudarán y guiarán en la realización del Diseño de los Casos de Prueba, a continuación se hará referencia a los más relevantes dentro los estudiados en las bibliografías:

- Se debe hacer un seguimiento hasta ver si se cumplen los requisitos del cliente.
- Las pruebas deberán planificarse mucho antes de que empiecen.
- 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 exhaustivas⁸.
- Para ser más efectivas, las pruebas deberían ser conducidas por un equipo independiente.
- Se deben evitar los casos de prueba no documentados ni diseñados con cuidado.
- No deben realizarse planes de prueba suponiendo que prácticamente no hay defectos en los programas y, por tanto, dedicando pocos recursos a las pruebas.

1.3.3 Características de las pruebas.

Hay ciertas características que son vitales para que una prueba sea calificada como idónea⁹, entre ellas se pueden citar las siguientes:

- Ha de tener una alta probabilidad de encontrar un fallo.

- No debe ser redundante. Si ya funciona, no lo probamos más.
- No debería ser ni demasiado sencilla ni demasiado compleja. Si es muy sencilla no aporta nada, si es muy compleja a lo mejor no sabemos lo que ha fallado.

1.3.4 Estrategia de Prueba.

Una estrategia integra técnicas de casos de prueba en una serie de pasos planeados que llevan a la construcción exitosa del software. Describe los pasos que hay que llevar a cabo como parte de la prueba, cuándo se deben planificar y realizar estos pasos, y cuánto esfuerzo, tiempo y recursos se van a requerir. Cualquier estrategia de prueba debe incorporar la planeación de las pruebas, el Diseño de Casos de Prueba, la ejecución de pruebas y la agrupación y evaluación de los datos resultantes.

Una estrategia de prueba ha de ser lo suficientemente flexible para promover la creatividad y el ajuste necesario a los sistemas de software, y lo suficientemente rígido para promover una planeación razonable y un seguimiento administrativo a medida que el proyecto progresa.

1.3.4.1 Características genéricas de la Estrategia de Prueba:

- a) Las pruebas se inician a nivel módulo y se van realizando hacia la integración del sistema completo.
- b) Diferentes técnicas de prueba son apropiadas en diferentes puntos del tiempo.
- c) Las pruebas se llevan a cabo por el desarrollador del software o por un grupo independiente de pruebas.
- d) La depuración¹⁰ debe ser incluida en cualquier estrategia de prueba.
- e) Debe incluir pruebas de bajo y alto nivel.

1.3.4.2 Los objetivos de la Estrategia de Prueba son:

- Planificar las pruebas necesarias en cada iteración¹¹, incluyendo las pruebas de unidad, integración y las pruebas de sistema. Las pruebas de unidad y de integración son necesarias dentro de la iteración, mientras que las pruebas de sistema son necesarias sólo al final de la misma.
- Diseñar e implementar las pruebas creando los casos de prueba que especifican qué probar, cómo realizar las pruebas y creando, si es posible, componentes de prueba ejecutables para automatizarlas.
- Realizar diferentes pruebas y manejar los resultados de cada prueba sistemáticamente. Los productos de desarrollo de software en los que se detectan defectos son probados nuevamente

y posiblemente devueltos a otra etapa, como diseño o implementación, de forma que los defectos puedan ser corregidos.

1.3.4.3 La Estrategia de Prueba suele seguir estas etapas:

Debe comenzar con pruebas a nivel de módulo y continuar hacia la integración del sistema completo y su instalación y se culmina con la aceptación del producto por parte del cliente.

De forma más específica:

- Se comienza en la prueba de cada módulo, que normalmente la realiza el propio personal de desarrollo en su entorno.
- Con el esquema del diseño del software, los módulos probados se integran para comprobar sus interfaces en el trabajo conjunto (prueba de integración).
- El software totalmente ensamblado se prueba como un conjunto para comprobar si cumple o no tanto los requisitos funcionales como los requisitos de rendimientos, seguridad, etc. (prueba funcional o de validación¹²).
- El software ya validado se integra con el resto del sistema (por ejemplo, elementos mecánicos, interfaces electrónicas, etc.) para probar su funcionamiento conjunto (prueba del sistema).
- Por último, el producto final se pasa a la prueba de aceptación para que el usuario compruebe en su propio entorno de explotación si lo acepta como está o no (prueba de aceptación).

1.4 Elementos fundamentales del Proceso de Pruebas.

1.4.1 Niveles de pruebas

Las pruebas que se le realizarán a un sistema están divididas en diferentes niveles, en los cuales se efectuarán diversas actividades con el objetivo de obtener un software con mayor calidad.

➤ Pruebas Unitarias

Prueba el diseño y el comportamiento de cada uno de los componentes una vez construidos. Una prueba unitaria es una forma de probar el correcto funcionamiento de un módulo. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. El objetivo de estas pruebas es aislar cada parte del programa y mostrar que las partes individuales son correctas.

Para que una prueba unitaria sea efectiva se deben cumplir los siguientes requisitos:

- **Automatizable**¹³: no debería requerirse una intervención manual.
- **Completas**: deben cubrir la mayor cantidad de código.

- **Repetibles o Reutilizables:** no se deben crear pruebas que sólo puedan ser ejecutadas una sola vez. También es útil para integración continua.
- **Independientes:** la ejecución de una prueba no debe afectar a la ejecución de otra.
- **Profesionales:** las pruebas deben ser consideradas igual que el código, con la misma profesionalidad, documentación, etc.

Es importante darse cuenta de que las pruebas unitarias no descubrirán todos los errores. Por definición, sólo prueban las unidades por sí solas. Por lo tanto, no descubrirán errores de integración, problemas de rendimiento y otros problemas que afectan a todo el sistema en su conjunto.

➤ **Pruebas de Integración**

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

Las pruebas de integración se realizan durante la construcción del sistema, involucran a un número creciente de módulos y terminan probando el sistema como conjunto. Estas pruebas se pueden plantear desde un punto de vista estructura¹⁴ o funcional.

Las pruebas estructurales de integración son similares a las pruebas de caja blanca; pero trabajan a un nivel conceptual superior. En lugar de referirnos a sentencias del lenguaje, nos referiremos a llamadas entre módulos. Se trata pues de identificar todos los posibles esquemas de llamadas y ejercitarlos para lograr una buena cobertura de segmentos o de ramas.

Las pruebas funcionales de integración son similares a las pruebas de caja negra. Aquí trataremos de encontrar fallos en la respuesta de un módulo cuando su operación depende de los servicios prestados por otros módulos. Según nos vamos acercando al sistema total, estas pruebas se van basando más y más en la especificación de requisitos del usuario.

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

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

➤ **Pruebas de Sistema**

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

Verifica que cada elemento encaja de forma adecuada y que se alcanza la funcionalidad y el rendimiento del sistema total. La prueba del sistema está constituida por una serie de pruebas

diferentes cuyo propósito primordial es ejercitar profundamente el sistema. Algunas de estas pruebas son:

- Prueba de validación: Proporciona una seguridad final de que el software satisface todos los requisitos funcionales y de rendimiento. Además, valida los requisitos establecidos comparándolos con el sistema que ha sido construido. Durante la validación se usan exclusivamente técnicas de prueba de caja negra.
- Prueba de recuperación: Fuerza un fallo del software y verifica que la recuperación se lleva a cabo apropiadamente.
- Prueba de seguridad: Verificar los mecanismos de protección.
- Prueba de resistencia: Enfrenta a los programas a situaciones anormales.
- Prueba de rendimiento: Prueba el rendimiento del software en tiempo de ejecución.
- Prueba de instalación: Se centra en asegurar que el sistema software desarrollado se puede instalar en diferentes configuraciones hardware y software y bajo condiciones excepciones, por ejemplo con espacio de disco insuficiente o continuas interrupciones.
- Pruebas de Implantación: Comprueba el correcto funcionamiento del sistema dentro del entorno real de producción (rendimiento, copias de seguridad, etc.).

➤ Pruebas de Aceptación

Las pruebas de aceptación verifican si el sistema cumple con todos los requisitos indicados y permite que los usuarios del sistema den el visto bueno definitivo.

Estas pruebas las realiza el cliente. Son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario. Estas pruebas no se realizan durante el desarrollo, sino una vez pasada todas las pruebas de integración por parte del desarrollador.

La experiencia muestra que aún después del más cuidadoso proceso de pruebas por parte del desarrollador, quedan una serie de errores que sólo aparecen cuando el cliente se pone a usarlo. El cliente siempre tiene razón. Decir que los requisitos no estaban claros, o que el manual es ambiguo puede salvar la cara; pero ciertamente no deja satisfecho al cliente.

Por estas razones, muchos desarrolladores ejercitan unas técnicas denominadas "pruebas alfa" y "pruebas beta". Las pruebas alfa consisten en invitar al cliente a que venga al entorno de desarrollo a probar el sistema. Se trabaja en un entorno controlado y el cliente siempre tiene un experto a mano para ayudarle a usar el sistema y para analizar los resultados.

Las pruebas beta vienen después de las pruebas alfa, y se desarrollan en el entorno del cliente, un entorno que está fuera de control. Aquí el cliente se queda a solas con el producto y trata de encontrarle fallas (reales o imaginarios) de los que informa al desarrollador. Las pruebas alfa y beta son habituales en productos que se van a vender a muchos clientes. La experiencia muestra que estas prácticas son muy eficaces.

1.5 Métodos de pruebas

Los métodos de prueba son un enfoque sistemático que ayuda a encontrar buenos conjuntos de casos de prueba, y a determinar su grado de cobertura.

Existen dos enfoques alternativos:

- **Caja negra:** se comprueban las funcionalidades sin tener en cuenta la estructura interna.
- **Caja blanca:** se comprueban los componentes internos (módulos, subprogramas, bucles, condiciones, etc.).

Estos enfoques deben ser combinados para obtener un conjunto de casos de prueba razonablemente riguroso utilizando métodos de caja negra y completar el conjunto con casos de prueba generados con métodos de caja blanca con la vista puesta en las partes del programa algorítmicamente más complejas.

➤ Método de prueba de Caja Negra

La elección de los casos de prueba no se va a basar en el conocimiento que se tenga acerca de la estructura del objeto, sino en el conocimiento acerca de la funcionalidad deseada. Casi siempre el número de combinaciones de entradas (válidas o inválidas) es muy elevado. Las pruebas de caja negra exhaustivas casi siempre son imposibles.

Algunos métodos de caja negra son:

- Clases de equivalencia.
- Análisis de valores límite.
- Grafo causa-efecto.

Las pruebas de caja negra son las pruebas que se lleva a cabo sobre la interfaz del software, es decir este pretende demostrar que las funciones son operativas que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como la integridad de la información externa se mantiene, es completamente indiferente el comportamiento interno y la estructura del programa.

La prueba de caja negra intenta encontrar errores de las siguientes categorías:

- Funciones incorrecta o ausente.
- 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.

Para realizar esta prueba se debe tener la Especificación del Programa o el Manual del Usuario, estos determinan lo que el programa tiene que hacer. Se debe realizar al menos una prueba de todas las funcionalidades del sistema para poder determinar correctamente la funcionalidad del mismo.

➤ **Método de prueba de Caja Blanca o Transparente**

El programa que se desea probar se ve como una caja transparente; la elección de los casos de prueba se va a basar en el conocimiento que se tenga acerca de la estructura del programa.

Dependiendo del grado de cobertura que se desea lograr (ordenados de menos a más exhaustivos):

- Cobertura de sentencias.
- Cobertura de decisiones.
- Cobertura de condiciones.
- Cobertura de decisiones/condiciones.
- Cobertura de condiciones múltiples.
- Cobertura de caminos.

Las pruebas de caja blanca son mucho más amplias y normalmente se denominan pruebas de cobertura o pruebas de caja transparente, al total de pruebas de caja blanca se les llama cobertura ya que es un número porcentual que indica cuanto código del programa se ha probado.

La idea de estas pruebas es diseñar un Plan de Pruebas en las que se vaya ejecutando sistemáticamente el código hasta que haya ocurrido todo o la gran mayoría de el.

Existen diferentes métodos de caja blanca como son:

- **La prueba del camino básico:** Esta prueba le permite al diseñador obtener una medida de complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución.

- **La prueba de condición:** Es un método de diseño de casos de prueba que prepara las condiciones lógicas contenidas en el módulo de un programa.
- **La prueba de flujo de datos:** Se selecciona caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa.
- **La prueba de bucles:** Es una técnica de prueba de caja blanca que se centra exclusivamente en la validez de las construcciones de bucles.

1.6 Tipos de pruebas

Existen diferentes tipos de pruebas:

➤ Prueba Unitaria

Es una forma de probar, de asegurar que cada uno de los módulos funcione correctamente por separado, el objetivo de estas pruebas es aislar, cada parte del programa y mostrar que las partes individuales son correctas.

➤ Prueba de Integración

Se realiza cuando se han integrado todos los módulos, y su objetivo es comprobar que el sistema satisface los requisitos del usuario, tanto los funcionales como los no funcionales. Esta prueba presenta tres estrategias descendente, ascendente y sandwich.

➤ Pruebas de Validación

Permite evaluar el sistema o parte de este durante o al final del desarrollo para determinar si satisface los requisitos iniciales. Esta se basa en los criterios de aceptación, pruebas alfa (entorno de desarrollo) y pruebas beta (entorno del cliente).

➤ Pruebas de Sistema

Robustez: Determinan la capacidad del programa para soportar entradas incorrectas.

Seguridad: Se determinan los niveles de permiso de usuario, las operaciones de acceso al sistema y acceso de datos.

Resistencia: Determinan hasta donde puede soportar el programa determinadas condiciones extremas.

Rendimiento: Determinan si los tiempos de respuesta, el espacio que ocupa el modulo en disco o en memoria, el flujo de datos que genera a través de una canal de comunicaciones.

Usabilidad: Se determina la calidad de la experiencia de un usuario en la forma que éste interactúa con el sistema, se considerarla facilidad de uso y el grado de satisfacción del usuario.

Instalación: Se determinan las operaciones de arranque y actualización del software.

1.7 Primera Fase

Antes de comenzar a realizar las pruebas de la Primera Fase se debe tener bien claro si el producto al que se le realizarán las pruebas es complejo y esta compuesto por varios módulos lo que se probarán cada módulo por separado hasta obtener el resultado esperado por los probadores.

Las Pruebas de Módulo se dividen en dos partes:

Las Pruebas de Módulo Independiente se realizan con el objetivo de encontrar y reparar fallas con el comportamiento de los módulos independientes.

Las Pruebas de Módulo Integrado tienen por objetivo encontrar y reparar fallas en el comportamiento de los módulos integrados, así como el control y el manejo de datos de manera global.

1.7.1 Pruebas de Módulo.

Dentro de Pruebas de Módulo se realizarán diversas pruebas las cuales integraran en conjunto la Primera Fase del Plan de Prueba de Módulo , lo que conlleva una vez terminada esta fase satisfactoriamente pasar a la Segunda Fase del mismo.

Estas permiten comprobar el módulo individualmente por parte del diseñador y del implementador. En estas pruebas también incluiremos los niveles pruebas de desarrollador, prueba independiente y prueba de unidad, en los cuales se desarrollaran diversos Procesos de Pruebas.

- **Prueba de desarrollador:** Es la prueba diseñada e implementada por el equipo de desarrollo.
- **Prueba independiente:** Es la prueba que es diseñada e implementada por alguien independiente del grupo de desarrolladores.
- **Prueba de unidad:** Es la prueba enfocada a los elementos más pequeño del software.

Los resultados de las pruebas realizadas en la Primera y Segunda Fase quedarán certificados en el Documento de No Conformidad el cual poseerá una estructura similar para cada una de ellas.

1.7.1.1 Revisión de los Manuales entregables al Usuario.

Los Manuales entregables al Usuario de la aplicación deben ser revisados antes de que este sea suministrado. Por lo que se le efectuará una amplia revisión a diferentes manuales como son:

➤ Manual de Usuario

El manual de usuario pretende facilitar al usuario el aprendizaje y manejo de los distintos sistemas a los que hace referencia. Contiene las explicaciones necesarias para localizar las normas y sentencias

que se necesitan, además de enseñar de la forma más clara posible las múltiples opciones que ofrece el sistema.

➤ **Manual de Instalación**

El manual de instalación fue creado por la necesidad de eliminar aquellos pequeños errores de instalación que en su mayoría no son tan pequeños ya que pueden dejar las computadoras inhabilitadas para realizar cualquier tarea y perder toda la información que en esta se encuentre. Este manual nos permite tener un apoyo escrito y pasos oficiales para instalar un sistema determinado logrando resultados satisfactorios.

➤ **Manual de Tecnología**

El manual de tecnología nos facilita conocer un poco más sobre el sistema antes de realizar alguna operación en el, es una guía para seguir cuando queremos comenzar a trabajar con un software nuevo. Este manual nos ayuda a dar respuesta aquellas preguntas que por primera vez se realizan cuando tenemos un nuevo sistema en la computadora.

1.7.1.2 Revisión de la documentación en el Flujo de Trabajo de Negocio.

La revisión de la documentación creada en el Flujo de Trabajo de Negocio encierra todos los artefactos que se obtienen como resultado del Modelamiento del Negocio así como lo referente a los Modelos Casos de Uso del Negocio y Modelo del Dominio.

En esta se realizará una revisión a los Modelos de los Casos de Uso del Negocio donde se deben describir los Procesos del Negocio de determinada empresa en términos de casos de uso y actores del negocio que se corresponden con los procesos del negocio y los clientes respectivamente.

Se realizará una revisión del Modelo de Dominio que debe describir como colaboran los trabajadores y las entidades del negocio dentro de Flujo de Trabajo de Negocio.

1.7.1.3 Revisión de la documentación en el Flujo de Trabajo Requisitos.

La revisión de la documentación creada en el Flujo de Trabajo Requisitos es una de las más importantes, pues en esta se debe tener en cuenta los Requisitos Funcionales y los Requisitos No Funcionales.

Se debe tener presente que los **Requisitos Funcionales** son capacidades o condiciones que el sistema debe cumplir y que los **Requisitos No Funcionales** son propiedades o cualidades que el producto debe tener, como son las características que hacen el producto más atractivo, usable, rápido o confiable.

Las pruebas de requisitos se realizarán para tener una explicación clara, precisa y completa del problema que facilite el análisis de errores y la generación de casos de prueba. Esta se realizará con el objetivo de asegurar la corrección, coherencia y exactitud de los requisitos. Durante el proceso de verificación de requisitos una persona designada por el equipo de Aseguramiento de Calidad revisará el documento de especificaciones de requisitos. En este proceso se deben seguir los siguientes pasos:

- Se determina si los objetivos son claros, verificables y necesarios (entre otros).
- Se define la funcionalidad del software esperada.
- Debe verificar el documento de requisitos.
- Revisan cada requisito (consistencia, ambigüedad, etc.)

1.7.1.4 Revisión de la documentación en el Flujo de Trabajo Análisis y Diseño.

En la revisión de la documentación creada en el Flujo de Trabajo Análisis y Diseño se tendrán en cuenta Modelo de Diseño, Modelo de Análisis y Documento de Arquitectura. Estos documentos examinan la situación del sistema con el propósito de mejorar el procedimiento y especifican las características del producto. En este flujo de trabajo las documentaciones creadas muestran una amplia visión del funcionamiento del sistema.

1.7.1.5 Revisión de la documentación en el Flujo de Trabajo Implementación.

En la Revisión de la documentación creada en el Flujo de Trabajo de Implementación se verifica que los elementos del diseño se implementen en término de componentes y que se encuentren agrupados de acuerdo a los nodos específicos en el Modelo de Despliegue.

La Revisión de la documentación del Flujo de Trabajo de Implementación tiene como objetivo principal:

- Verificar la planificación de los subsistemas que deben ser implementados y en que orden han sido integrados, verificando también el Plan de Integración.
- Comprobar el orden para implementar los elementos del subsistema.
- Verificar los subsistemas.

En el Flujo de Trabajo Implementación se debe realizar una revisión a los diagramas de Despliegue y de Componentes, estos conforman un modelo de implementación, el cual muestra la configuración de los componentes hardware, los procesos, los elementos de procesamiento en tiempo de ejecución y los objetos que existen en tiempo de ejecución.

1.7.1.6 Pruebas de Funcionalidad y Diseño de Casos de Prueba.

Una vez desarrollada la aplicación, los desarrolladores necesitan verificar que esta libre de anomalías¹⁵ y que se ha logrado el objetivo en su diseño por lo que una vez realizadas las especificaciones de casos de uso, se procede a diseñar casos de prueba para que el probador verifique la correcta funcionalidad del sistema.

Al realizar las pruebas de Funcionalidad y de Diseño de Casos de Prueba se debe redactar el Documento de No Conformidad para dejar plasmado todos los errores encontrados en dichas pruebas y diversas recomendaciones que serán de gran ayuda a los desarrolladores del sistema.

➤ **Diseño y Refinamiento de Casos de Prueba.**

El Refinamiento de Casos de Prueba se realiza con el fin de garantizar que no solo se ha considerado las especificaciones propiamente funcionales de las aplicaciones, sino también se cuenta con los requisitos de los clientes y de los usuarios finales de la misma. De forma general se recopilarán las herramientas adecuadas para llevar a cabo los casos de prueba correspondiente en cada aplicación.

En los Diseños de Casos de Prueba se tienen en cuenta los principales métodos de prueba:

- **Prueba de Caja Negra**
- **Prueba de Caja Blanca**

➤ **Pruebas de Funcionalidad.**

Las pruebas de Funcionalidad determina la extensión en la que la aplicación satisface los requisitos funcionales antes expuestos, incluyendo navegación, entrada de datos, procesamiento y obtención de resultados. Este incluye la verificación de que la aplicación se ajusta a las especificaciones y ejecuta correctamente todas las funciones requeridas en una variedad de dispositivos y plataformas. Estas técnicas aseguran que la aplicación funcionará apropiadamente en el mundo real. En las pruebas de funcionalidad existen pruebas que permiten realizar un mejor desempeño y resultado de la misma.

- **Pruebas Funcionales.**

Mediante estas pruebas se verificará el procesamiento, recuperación e implementación apropiada a las reglas del negocio, se verificará también la adecuada aceptación de datos, se asegurará el trabajo apropiado de los requisitos funcionales incluyendo la navegación, entrada de datos y obtención de resultados.

Al realizar estas pruebas se deben tener en cuenta los siguientes aspectos que el sistema debe cumplir:

- Que se aplique apropiadamente cada regla de negocio.

- Que los resultados esperados ocurran cuando se usen datos válidos.
- Que sean desplegados los mensajes apropiados de error y precaución cuando se usan datos inválidos.

- **Pruebas de Seguridad.**

Las pruebas de seguridad en el sistema se centran en la realización de un informe una vez realizada las pruebas de funcionalidades del mismo donde se recogerán las vulnerabilidades¹⁶ existentes y sus soluciones, recomendaciones de la conservación de la seguridad del sistema. Su objetivo principal es proteger el sistema de ataques externos defendiendo el perímetro, sin tener en cuenta que los mayores riesgos son los internos y los problemas existentes radican en la baja calidad del software que se desarrolla. Estas pruebas garantizan que los usuarios estén restringidos a funciones específicas o su acceso este limitado únicamente a los datos que está autorizado a acceder, que solo aquellos usuarios autorizados a acceder al sistema son capaces de ejecutar, realizar funciones en el mismo.

Las pruebas de seguridad se realizan con el fin de determinar la protección del sistema contra accesos internos o externos no autorizados. Para ello se probará acceder al sistema de diferentes formas hasta obtener las fallas de seguridad, se deben verificar cada permiso creando transacciones específicas para cada tipo de usuario.

- **Pruebas de Volumen.**

Las pruebas de volumen es el tipo de prueba que se enfoca en certificar la capacidad del sistema de manejar volúmenes de datos extremos. Estas pruebas tienen como objetivo fundamental verificar que la aplicación funcione correctamente bajo los siguientes escenarios:

- Que estén conectados más de un usuario al sistema y todos ejecutando la misma función por un periodo extendido.
- Que se realicen múltiples consultas simultáneamente.

Las pruebas de volumen determinan si el programa puede manejar los volúmenes requeridos de datos, solicitudes de conexión, entre otros. Para la realización de esta prueba se deben usar múltiples clientes corriendo la misma prueba o pruebas complementarias para producir el peor caso de volumen por un periodo extendido. Se debe utilizar un tamaño máximo de la Base de Datos y múltiples clientes para correr consultas simultáneamente para periodos extendidos.

1.7.1.7 Pruebas para verificar Interfaz de Usuario y Atributos de Calidad.

Para verificar la interfaz de usuario y los atributos de calidad se propone realizar un chequeo organizado y examinar cuidadosamente todas las áreas importantes del sistema.

➤ **Interfaz de Usuario.**

La interfaz de usuario es la forma en que los usuarios pueden comunicarse con una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo. La interface de usuario es el aspecto más importante de cualquier aplicación. Una aplicación sin un interfaz fácil, impide que los usuarios saquen el máximo rendimiento del programa.

Sus principales funciones son:

- Manipulación de archivos y directorios.
- Herramientas de desarrollo de aplicaciones.
- Información de estado.
- Configuración de la propia interfaz y entorno.
- Control de acceso.
- Sistema de ayuda interactivo.

Dentro de las interfaces de usuario se distinguen básicamente dos tipos:

- Una interfaz de hardware, a nivel de los dispositivos utilizados para ingresar, procesar y entregar los datos: teclado, ratón y pantalla visualizadora.
- Una interfaz de software, destinada a entregar información acerca de los procesos y herramientas de control, a través de lo que el usuario observa habitualmente en la pantalla.

➤ **Atributos de Calidad.**

Resulta casi imposible aceptar todos los atributos de calidad, pues para poder alcanzar ciertos atributos impide que se logren otros y es cuando se dice que entran en conflicto. Por lo que se deben priorizar y determinar cuales son los verdaderamente importantes para el cliente y cuales están dispuestos a sacrificar para alcanzar su objetivo.

Algunos atributos que componen la Calidad son:

- **Correcto** (Que el programa este libre de errores).
- **Usable** (Fácil de Usar).
- **Eficiente** (El sistema responde las llamadas del usuario correctamente).
- **Confiable** (La habilidad del programa para cumplir con todas sus funciones de forma constante teniendo un largo periodo de tiempo entre fallas).

- **Integro** (El grado en que un sistema previene el acceso no autorizado de usuarios a funciones o datos del programa).
- **Adaptable** (La capacidad del programa para adaptarse a medios ambientes diferentes a aquellos en los cuales fue diseñado o pensado para operar. Las computadoras se hacen obsoletas rápidamente, cada que cambiamos de computadora viene con un sistema operativo diferente y nuevas herramientas de escritorio).
- **Exacto** (Las aplicaciones que realizan operaciones numéricas son un buen ejemplo de la aplicación de este criterio).
- **Robusto** (Capacidad del Sistema para continuar operando en circunstancias extremas como puede ser captura de datos inválidos, excesiva carga de trabajo).

1.8 Segunda Fase

En esta Segunda Fase se realizarán diversas revisiones tanto para validar los errores existidos en la Primera Fase, como para erradicar aquel nuevo que se introdujo y no ha sido detectado. Esta fase se realizará tantas veces sea necesario hasta obtener un resultando satisfactorio que satisfaga las necesidades del cliente y que sea lo que este aspiraba a obtener.

1.8.1 Revisión para certificar la corrección de errores detectados en la Primera Fase.

La revisión de los errores detectados en la Primera Fase se realizará a través de la aplicación de las pruebas que se habían realizados anteriormente y apoyándose en los Documentos de No Conformidad.

1.8.2 Revisión para detectar nuevos errores introducidos.

La revisión para detectar nuevos errores introducidos se realizará con el mismo seguimiento que se realizaron en la Primera Fase sin omitir ninguna de estas pruebas, pues puede que en dicha prueba que omitamos se encuentre el error que determine la efectividad del producto.

El equipo de calidad entrega los resultados obtenidos a los diferentes módulos, posteriormente el equipo de desarrollo trabaja en función de estos para perfeccionar al máximo el software. Este proceso se iterará tantas veces sea necesario. Cuando los desarrolladores solucionen los errores encontrados por el equipo de calidad este deberá entregar un documento que especifique exactamente que parte se cambió en cada artefacto, para evitar volver a indicar errores ya señalados y aun no corregidos.

1.9 Conclusiones

En este capítulo se profundizó en el conocimiento de algunos conceptos y especificaciones necesarias para la comprensión de diversas pruebas que serán aplicadas a un sistema. Se detallaron los tipos de pruebas actuales que se le aplica a un software antes de ser liberado en la etapa de Prueba de Módulo y los documentos que deben ser elaborados tanto por el equipo de calidad como por el cliente. Además se realizó un análisis completo para determinar las pruebas que se le realizarán a diferentes artefactos. En base a esto se llega a la conclusión de que se debe seguir la secuencia de las pruebas expuestas sin llegar a omitir ninguna de estas, como los pasos necesarios de la documentación del sistema que nos brinda gran información de la misma, las cuales presentan diferentes estructuras que serán plasmadas a lo largo del desarrollo de este trabajo.

Capítulo 2

Propuesta de guía de revisiones técnicas y pruebas.

2.1 Introducción

En el presente capítulo se establecen propuestas de Listas de Chequeo para los diferentes tipos de pruebas que se llevarán a cabo en los distintos módulos con que cuenta la Unidad de Compatibilización, Integración y Desarrollo de Productos Informáticos para la Defensa, así como Criterios de Evolución que permitirán evaluar el resultado de las mismas.

Se instauran disímiles Listas de Chequeo para los Manuales entregables al Usuario, para la documentación de los distintos Flujos de Trabajos, para la Interfaz de Usuario, los Atributos de Calidad, como también para las Pruebas de Funcionalidades. Teniendo en cuenta que en cada prueba realizada se localizarán una gran cantidad de no conformidades, estas serán documentadas en el Documento de No Conformidad el cual será expuesto en este capítulo.

Otro de los aspectos que tendremos presente son los elementos a probar en cada fase ,a los cuales se le aplicarán las pruebas expuestas en el Capítulo 1, así como la propuesta de la Estructura del Plan de Prueba y el Diseño de Casos de Prueba para el módulo que se estará probando en el Laboratorio de Calidad de la UCID.

2.2 El proceso de pruebas en RUP (Proceso Unificado de desarrollo).

- **Aspectos generales, artefactos, trabajadores y actividades. Metodología.**

De forma general se puede decir que RUP es una metodología muy potente de desarrollo de software, que describe como reutilizar los componentes existentes o implementar nuevos componentes con tareas bien definidas, el que nos lleva a obtener un sistema fácil de mantener al que se le puede ir incrementando las posibilidades de reutilización. Asegurando la producción de software de alta calidad con un costo y tiempo predecible para el usuario.

RUP divide el proceso de desarrollo en ciclos, donde se obtiene un producto al final de cada ciclo. Cada ciclo se divide en cuatro Fases: Inicio, Elaboración, Construcción, y Transición. Cada fase concluye con un hito bien definido donde deben tomarse ciertas decisiones.

En varios de los documentos estudiados se plantea la importancia que le da RUP a la ingeniería de pruebas, en aras de lograr la máxima calidad durante el desarrollo del producto en construcción, por ello define claramente aspectos como los objetivos de las pruebas, los principales artefactos y

trabajadores, así como las actividades que deben desarrollar los ingenieros de pruebas en las 4 fases del ciclo de vida del proyecto, a todo esto haremos referencia a continuación:

La realización de pruebas persigue varios objetivos que consideramos sean de vital importancia, por lo que se hace necesario:

- Planificar las pruebas necesarias en cada iteración, incluyendo las pruebas de integración y las pruebas de sistema. Las pruebas de integración son necesarias para cada construcción dentro de la iteración, mientras que las pruebas de sistema son necesarias sólo al final de la iteración.
- Diseñar y crear las pruebas creando los casos de prueba que especifican qué probar, creando los procedimientos de prueba que especifican cómo realizar las pruebas y creando, si es posible, componentes de prueba ejecutables para automatizar las pruebas.
- Realizar las diferentes pruebas y manejar los resultados de cada prueba sistemáticamente. Las construcciones en las que se detectan defectos son probadas de nuevo y posiblemente devueltas a otro flujo de trabajo, como diseño o implementación, de forma que los defectos importantes puedan ser arreglados.

Los principales artefactos definidos en RUP son:

1. Modelo de pruebas: Este modelo describe principalmente cómo se prueban los componentes ejecutables (como las construcciones) en el modelo de implementación con pruebas de integración y de sistema. Describe además cómo deben probarse aspectos específicos del sistema (interfaz de usuario, manual del usuario, etc.). El modelo de pruebas es un conjunto de casos de prueba, procedimientos de prueba y componentes de prueba.

2. Caso de prueba: Un caso de prueba especifica una forma de probar el sistema, incluyendo la entrada o resultado con la que se ha de probar y las condiciones bajo las que ha de probarse. Puede derivarse de un caso de uso del modelo de casos de uso o de una realización de caso de uso del modelo de diseño.

- Una **prueba de caja negra**, evalúa el comportamiento externamente observable del sistema, se puede realizar a partir de un caso de prueba que especifica cómo probar un caso de uso del modelo de casos de uso (verificación del resultado de la interacción entre los actores y el sistema, satisfacción de las precondiciones, pos condiciones y secuencias de acciones especificadas por el caso de uso).
- Una **prueba de caja blanca**, evalúa la interacción interna entre los componentes del sistema, se puede realizar a partir de un caso de prueba que especifica cómo probar una realización de

un caso de uso de diseño o un escenario específico de la realización. Puede incluir la verificación de la interacción entre los componentes que implementan dicho caso de uso.

- **Pruebas de instalación:** Se realizan para verificar que el sistema puede ser instalado en la plataforma del cliente y que funcionará correctamente después de instalado.
- **Pruebas de configuración:** Se realizan para verificar que el sistema funciona bien en diferentes configuraciones, por ejemplo diferentes configuraciones de red.
- **Pruebas negativas:** Intentan provocar que el sistema falle para poder así revelar sus debilidades.
- **Pruebas de tensión o estrés:** Indican problemas con el sistema cuando hay recursos insuficientes o cuando hay competencia por los recursos.

3. Procedimiento de prueba: Especifica cómo realizar uno o varios casos de prueba o partes de éstos.

4. Componente de prueba: Automatiza uno o varios procedimientos de prueba o parte de ellos. Pueden ser programados o utilizar una herramienta de automatización de pruebas.

5. Plan de Pruebas: Describe la estrategia, recursos y planificación de la prueba. La estrategia de prueba incluye la definición del tipo de prueba a realizar en cada iteración y sus objetivos.

6. Defecto: Es una anomalía del sistema, descubierto en una revisión o un fallo del software.

7. Evaluación de pruebas: Es una evaluación de los resultados de los esfuerzos de prueba.

Los trabajadores definidos en RUP son:

1. Diseñador de pruebas: Planifica las pruebas. Es responsable de la integridad del modelo de pruebas asegurando que el modelo cumple con su propósito. Del modelo de pruebas deben definir y describir los casos de prueba y los procedimientos de prueba. Además son los responsables de la evaluación de las pruebas de integración y de sistema cuando éstas se ejecuten.

2. Ingeniero de componentes: Son responsables de los componentes de pruebas que automatizan algunos de los procedimientos de pruebas.

3. Ingeniero de pruebas de integración: Son los responsables de realizar las pruebas de integración, las cuales se realizan para verificar que los componentes integrados en una construcción funcionan correctamente juntos, y se derivan a menudo de los casos de pruebas que especifican cómo probar realizaciones de casos de uso de diseño. Debe documentar los defectos en las pruebas de integración.

4. Ingeniero de pruebas de sistema: Son los responsables de realizar las pruebas de sistema sobre los ejecutables. Estas pruebas se llevan a cabo fundamentalmente para verificar las interacciones entre los actores y el sistema. Estas pruebas se derivan a menudo de los casos de pruebas que

especifican cómo probar los casos de uso, también se usan para probar el sistema como un todo. Debe documentar los defectos en las pruebas de sistema.

Las principales actividades definidas en RUP se explican a continuación:

1. Planificar pruebas

Al planificar las pruebas, se planifican los esfuerzos de prueba en una iteración y debe llevarse a cabo las siguientes tareas:

- Describir una estrategia de prueba: cuantos flujos básicos y alternativos deben ser probados, cuantas pruebas se realizarán automatizadas y cuántas manuales.
- Estimar los requisitos para el esfuerzo de la prueba: recursos humanos, sistemas necesarios, etc.
- Planificar el esfuerzo de prueba.

Para obtener el Plan de Prueba, la estrategia de pruebas y los requisitos necesarios de pruebas de una iteración los ingenieros de prueba deben partir de los artefactos de entrada: requisitos adicionales, modelo de casos de uso, modelo de análisis, modelo de diseño, modelo de implementación, descripción de la arquitectura (vistas arquitectónicas de los modelos).

2. Diseñar pruebas

Al diseñar las pruebas deben tenerse los siguientes propósitos:

- **Identificar y describir los casos de prueba para cada construcción**

Para diseñar las pruebas de cada construcción los ingenieros de prueba deben partir de los artefactos de entrada: requisitos adicionales, modelo de casos de uso, modelo de análisis, modelo de diseño, modelo de implementación, descripción de la arquitectura (vistas arquitectónicas de los modelos) y del Plan de Pruebas (estrategia y planificación). De esa forma deben obtener los casos de pruebas y los procedimientos de pruebas.

- **Diseño de Casos de Prueba de Integración.**

Deben ser diseñados los casos de prueba de integración para verificar que los componentes interaccionan entre sí de la forma apropiada después de ser integrados en una construcción. Estos casos de prueba pueden ser derivados de las realizaciones de los casos de uso de diseño (un diagrama de secuencia es parte de la realización del caso de uso diseño) o los diagramas de interacción de las realizaciones de casos de uso, ya que estas realizaciones describen cómo interaccionan las clases y los objetos, y por tanto cómo interaccionan los componentes.

Los diseñadores de pruebas buscan combinaciones de entradas, salida y estado inicial de sistema que den lugar a escenarios interesantes que empleen las clases (y por tanto los componentes) que participan en los diagramas.

- **Diseño de Casos de Prueba de Regresión.**

Algunos de los casos de pruebas de construcciones anteriores pueden ser usados para pruebas de regresión en construcciones subsiguientes, aunque no todos pueden ser utilizados para pruebas de regresión. Los casos de pruebas, para ser usados en pruebas de regresión, deben ser suficientemente flexibles para ser resistentes a cambios, esta flexibilidad debe ser cuidadosa ya que la conversión de un caso de prueba a caso de prueba de regresión supone un esfuerzo de desarrollo extra, luego se debe convertir casos de prueba a casos de prueba de regresión sólo cuando el esfuerzo merezca la pena.

Identificar y estructurar los procedimientos de prueba especificando cómo realizar los casos de prueba. Se debe reutilizar procedimientos de prueba existentes tanto como sea posible, realizando las modificaciones adecuadas. Los diseñadores de prueba deben crear procedimientos de pruebas que puedan ser reutilizados en varios casos de prueba.

3. Implementar pruebas

El propósito de la implementación de pruebas es automatizar los procedimientos de prueba, creando componentes de pruebas cuando sea posible.

Los componentes de pruebas usan por lo general grandes cantidades de datos de entrada y generan grandes cantidades de datos de salida. Para visualizar estos datos en ocasiones es útil utilizar hojas de cálculo y bases de datos.

4. Realizar pruebas de integración

Esta actividad garantiza que se realicen las pruebas de integración necesarias para cada una de las construcciones creadas en una iteración y se recopilan los resultados de las pruebas. Estas pruebas se llevan a cabo con los siguientes pasos:

- Realizar las pruebas de integración realizando los procedimientos de pruebas manualmente o ejecutando los componentes de pruebas si existen.
- Comparar los resultados de las pruebas con los esperados e investigar los resultados de las pruebas que no coinciden con los esperados.
- Informar de los defectos a los ingenieros de componentes.
- Informar de los defectos a los diseñadores de pruebas, quienes usarán los defectos para evaluar los resultados del esfuerzo de prueba.

5. Realizar prueba de sistema

La prueba de sistema puede empezar cuando las pruebas de integración indican que el sistema satisface los objetivos de calidad de integración fijados en el Plan de Pruebas de la iteración, debe realizarse con pasos análogos a las pruebas de integración.

6. Evaluar prueba

El propósito de esta actividad es evaluar los esfuerzos de prueba en cada iteración. Para evaluar los resultados de pruebas estos deben ser comparados con los objetivos esbozados en el Plan de Pruebas. Los diseñadores de pruebas preparan métricas que les permiten determinar el nivel de Calidad de Software y qué cantidad de pruebas es necesario hacer.

Luego de analizar el proceso de pruebas que define RUP, se decide utilizar para la ejecución de las pruebas el proceso que describe el mismo, puesto que ha sido la metodología de desarrollo de software escogida por el arquitecto del equipo de desarrollo, para aplicar en todo el ciclo de vida. Además RUP es una metodología que se puede aplicar en proyectos grandes y pequeños, muy potente para el desarrollo de software basada en UML (Lenguaje Unificado de Modelado), que describe como reutilizar los componentes existentes o implementar nuevos componentes existentes con tareas bien definidas, llevándonos a obtener un sistema fácil de mantener al que se le puede ir incrementando las posibilidades de reutilización.

Una de las grandes ventajas que tiene RUP son sus casos de uso y sus casos de prueba, ya que los casos de uso facilitan las tareas de programación y los casos de prueba garantizan un Plan de Pruebas bastante bueno y robusto. Otra ventaja y que constituye una de sus características principales es su enfoque iterativo, implicando con esto que se estará evaluando a lo largo de todo el proyecto, con el fin de encontrar defectos lo antes posible, y poder así reducir el costo por la detección de defectos.

El Proceso Unificado del Software es nuestra guía a la hora de realizar pruebas a los documentos con respecto a la Ingeniería. RUP proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es garantizar la producción de programas de alta calidad que se ajusten a las necesidades de sus usuarios finales, dentro de un previsible calendario propuesto. RUP ofrece las mejores prácticas a través de modelos y herramientas para todo el ciclo de vida del software. Esta base de conocimiento nos permite trabajar con mayor facilidad a la hora de aplicar pruebas es por ello que lo utilizamos como guía de trabajo en el transcurso de la aplicación del proceso de pruebas a los diferentes módulos de la UCID.

2.3 Listas de Chequeo

Las Listas de Chequeo son creadas con el objetivo de facilitar el trabajo de los probadores y obtener un mejor resultado en cada una de las pruebas que serán realizadas. Es una guía con que puede contar el equipo de prueba para lograr que los usuarios finales se sientan satisfechos, esta permite cubrir los espacios más comunes del problema y examina cuidadosamente todas las áreas importantes considerando que las mejoras pueden ser planificadas.

Las Listas de Chequeo creadas verifican cada una de las plantillas expuestas en el Expediente de Proyecto implantado en la UCID que exhibe la documentación creada por cada Flujo de Trabajo.

2.3.1 Criterio Real de cada Lista de Chequeo

En cada Lista de Chequeo se deberá obtener un criterio real a partir de las evaluaciones que realizan cada uno de los probadores. Este criterio será calculado de la siguiente forma:

Cant 5	Cant 4	Cant 2	Cant 0
X	X	X	X

Tabla 1: Cantidad de valores que presenta la Lista de Chequeo.

- **Calcular la cantidad Real.**

$$\text{Real} = \text{Cant } 5 * 5 + \text{Cant } 4 * 4 + \text{Cant } 2 * 2 + \text{Cant } 0 * 0$$

$$\text{Real} = X$$

- **Calcular la cantidad Ideal.**

$$\text{Ideal} = (\text{Cant } 5 + \text{Cant } 4 + \text{Cant } 2 + \text{Cant } 0) * 5$$

$$\text{Ideal} = X$$

- **Por ciento Real de la Lista de Chequeo a través del Criterio de Evaluación.**

$$\text{Real} \text{----} X$$

$$\text{Ideal} \text{----} 100$$

$$X = (\text{Real} * 100) / \text{Ideal}$$

$$X = X$$

2.3.2 Revisión de los Manuales entregables al Usuario.

➤ Manual de Usuario

Para llevar a cabo la revisión del Manual de Usuario se propone una Lista de Chequeo de Manual de Usuario, que fue realizada con el objetivo de verificar si la información que se encuentra en dicho documento es la necesaria para que el usuario utilice correctamente la aplicación y que esta se encuentre de forma descifrable para aquellos que no presentan conocimiento de informática.

➤ Manual de Instalación

Para llevar a cabo la revisión del Manual de Instalación se propone en este capítulo una Lista de Chequeo de Manual de Instalación, que fue diseñada con el objetivo de verificar que este contenga la información necesaria para implementar dicha aplicación. Dentro de este documento se encuentran las instrucciones para la puesta en marcha del sistema y las normas de utilización del mismo.

➤ Manual de Tecnología

Para realizar la revisión del Manual de Tecnología se propone una Lista de Chequeo con el fin de verificar toda aquella información que es determinante para que el sistema sea ejecutado correctamente. Este manual permite tener un conocimiento previo de todos los requisitos tanto de hardware como de software que debe presentar el programa que se quiere manipular en ese instante.

2.3.3 Listas de Chequeo para la revisión de la documentación en el Flujo de Trabajo del Negocio.

Para la revisión de los documentos creados en Flujo de Trabajo del Negocio en la Primera Fase de prueba, se propone la Lista de Chequeo del Modelo de Casos de Uso del Negocio y la Lista de Chequeo del Modelo del Dominio.

Se propone la revisión del Modelo de Casos de Uso del Negocio, ya que este permite visualizar el alcance del sistema, representando lo que abarca y cuáles son sus límites. Así mismo, modela las actividades y procesos que serán ejecutados en dicho sistema, señala gráficamente las funciones y metas que persigue el negocio, y también permite identificar cuáles son los entregables y roles dentro del desarrollo del mismo.

La Lista de Chequeo del Modelo de Casos de Uso del Negocio se realiza con el objetivo de verificar cada aspecto del modelo expuesto, así como los actores que intervienen en el mismo, los casos de usos identificados, las relaciones existentes, entre otros elementos imprescindibles en dicha modelación.

Se muestra a continuación un ejemplo de estructura de las Listas de Chequeo creadas, tomando como modelo la Lista de Chequeo del Modelo de Casos de Uso del Negocio.

UCI - FAR

Modelo de Casos de Uso del Negocio

Proyecto	Módulo	Etapa	Versión	Fecha
<nombre del proyecto>	<nombre del módulo>	<etapa en que se realiza>	<1.0>	<00/00/00>

Control de Versiones:

Fecha	Versión	Descripción	Autor
<00/00/00>	<1.0>	<>	<nombre del que aplicó la lista>

No.	Evaluación	Eval.	NP.	Comentario
1	¿Se realizó el Modelo de Negocio, así como su breve descripción?			
2	¿Se han especificado los actores del negocio?			
3	¿El nombre que presenta el actor del negocio, es el rol que este desempeña en el mismo?			
4	¿Se especificaron correctamente los trabajadores del negocio?			
5	¿Se realizó correctamente el diagrama de CU de Negocio?			
6	¿Se especificaron correctamente los casos de usos, así como su descripción?			
7	¿Se describe en la tabla los detalles del caso de uso en fusión de acción del actor y respuesta del sistema?			

Diagrama de Actividad

No.	Evaluación	Eval.	NP.	Comentario
8	¿Se realizó el diagrama de Actividades?			
9	¿Los Diagramas de Actividad tienen bien representados su único estado inicial y al menos un estado final?			
10	¿Las actividades transitan siempre a: otra actividad, una decisión, una barra de sincronización o un estado final?			
11	¿Las actividades están precedida por: otra actividad, una decisión, una barra de sincronización o un estado inicial?			
12	¿En el diagrama quedan claro el orden en que ocurren las actividades y además quedan claro cuando el orden de las actividades no es significativo (pueden ejecutarse en paralelo) utilizando para ello, barras de sincronización?			
13	¿Se utilizan barra de sincronización para representar caminos alternativos?			
14	¿Todas las actividades que están descritas dentro de un Diagrama de Actividad son posibles?			
15	¿Las calles sólo se corresponden con Unidades Organizativas Trabajadores de Negocio o Actores de Negocio?			
16	Una misma Entidad de Negocio puede ser utilizada como entrada (con un mismo estado) en más de una actividad.			
17	Siempre que una actividad del diagrama transforme una Entidad de Negocio debe señalarse ésta como entrada y salida pero con dos estados diferentes.			
18	Varias actividades no pueden hacer igual			

	transformación a la misma Entidad de Negocio.			
19	¿El flujo de actividades del diagrama es claro y fácil de comprender?			
20	¿Para indicar en el Diagrama de Actividad la ejecución de un caso de uso extendido o incluido se colocaron las actividades con el mismo nombre del caso de uso? Se sugiere resaltar dicha actividad			
21	¿Están resaltadas dentro del diagrama aquellas actividades que se pretenden automatizar?			

Diagrama de Clase del Modelo de Objeto del Negocio

No.	Evaluación	Eval.	NP.	Comentario
22	¿Se realizó el Modelo de Objeto?			
23	¿Se incluyeron: las Entidades de Negocio, la relación que existe entre ellas y los Trabajadores de Negocio que interactúan con dichas entidades?			

Balance entre artefactos

No.	Evaluación	Eval.	NP.	Comentario
24	¿Cada caso de uso tiene asociado al menos un Diagrama de Actividad que lo describa?			
25	¿Las Entidades de Negocio que aparecen en los Diagramas de Actividades aparecen en el Diagrama de Clases del Modelo Objeto de Negocio?			
26	¿Los Trabajadores de Negocio que aparecen en los Diagramas de Actividades manipulando alguna Entidad de Negocio aparecen en el Diagrama de Clases del Modelo Objeto de Negocio?			

Notas:

<Insertar nota si es necesario>

Leyenda

Eval. : Evaluación que se le otorga al aspecto a evaluar: [0, 2, 4,5,].

NP. : No procede (se marca con una X la pregunta que no será necesaria realizarla en el sistema que se le esta aplicando la lista de chequeo).

Comentario: Es obligatorio en las evaluaciones distintas de 5 puntos.

Las evaluaciones serán:

-Malo: propiedad parcialmente disponible. (0)

--Satisfactorio: propiedad parcialmente disponible. (2)

+ Bien: propiedad parcialmente disponible. (4)

++Excelente: propiedad muy bien implementada. (5)

Criterio de evaluación:

() Sin modificaciones.

() Pequeñas modificaciones.

() Grandes modificaciones.

() Nueva elaboración.

Conclusión

	Aceptado
	Diferido
	No Aceptado

Se propone además la revisión del Modelo del Dominio en la Primera Fase, ya que este es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema, conceptos del mundo real y no de los componentes de software.

La Lista de Chequeo del Modelo del Dominio tiene como propósito comprobar cada uno de los componentes del modelo realizado, así como los objetos capturados del negocio, la relación que existe entre dichos objetos en el diagrama, entre otros aspectos que se deben ser verificados en el modelo correspondiente. Ver Anexo 3.

2.3.4 Listas de Chequeo para la revisión de la documentación del Flujo de Trabajo Requisitos.

Para la revisión de los documentos creados en el Flujo de Trabajo de Requisitos en la Primera Fase, se propone la Lista de Chequeo de Especificación de Requisitos, la Lista de Chequeo del Modelo Casos de Uso del Sistema y la Lista de Chequeo de Plan de Gestión de Requisitos.

La Especificación de Requisitos nos permiten obtener una descripción completa del comportamiento del sistema que se va a desarrollar, fija los requisitos funcionales y los no funcionales de sistema, especifica la funcionalidad, rendimiento, interfaz, restricciones de diseño, determina también los atributos, entre otros aspectos necesarios para obtener una buena descripción de los requisitos.

La Lista de Chequeo que se aplicará en la plantilla de Especificación de Requisitos tiene como propósito verificar la captura de los requisitos funcionales y los no funcionales, incluyendo en los no funcionales cada una de sus clasificaciones, se verifica además si estos son aplicables al sistema, entre otros aspectos significativos que la Especificación debe tener presente en el documento. Ver Anexo 4.

La plantilla del Modelo de Casos de Uso del Sistema ayuda a describir lo que el sistema debe hacer, es decir muestra las distintas operaciones que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (usuarios u otras aplicaciones).

La Lista de Chequeo que se aplicará al Modelos de Casos de Uso del Sistema tiene como objetivo verificar todos los aspectos que debe tener presente dicho modelo, así como sus acápite, artefactos, la identificación de los actores del sistema, los casos de uso del sistema y las relaciones entre ellos. Ver Anexo 6.

Otra de las propuestas para lograr un resultado satisfactorio en la revisión del Flujo de Trabajo de Requisitos es la plantilla del Plan de Gestión de Requisitos, ya que este tiene como objetivo gestionar los requisitos asociados en los proyectos, así como identificar inconsistencias entre requisitos y los planes de proyecto y productos de trabajo.

La Lista de Chequeo que se aplicará en el documento de Plan de Gestión de Requisitos tiene como objetivo verificar todos los aspectos que deben tener presente dicho modelo, así como sus acápite, artefactos, los responsables de las actividades, las herramientas de software para la Gestión de Requisitos y los procedimientos a seguir en el desarrollo del software. Ver Anexo 5.

2.3.5 Listas de Chequeo para la revisión de la documentación del Flujo de Trabajo Análisis y Diseño.

Para la revisión de los documentos creados en el Flujo de Trabajo Análisis y Diseño se propone la Lista de Chequeo del Modelo de Análisis, la Lista de Chequeo del Modelo de Diseño y la Lista de Chequeo para el Documento de Arquitectura de Software.

Se propone para la revisión la plantilla Modelo de Análisis, ya que en término general se puede definir que este modelo establece la realización de los casos de uso en las clases, pasando desde una representación en términos de análisis hacia una de diseño de acuerdo al avance del proyecto. Consiste en obtener una visión que se preocupa en ver que hace el sistema de software.

La Lista de Chequeo que se aplicará en el documento Modelo de Análisis tiene como objetivo verificar que el modelo anteriormente expuesto presente todos los aspectos, además los principales elementos que dicho modelo debe tener presente. La Lista de Chequeo presenta también la identificación de las clases interfaz, controladora y la entidad, así como componentes de los diagramas de interacción. Ver Anexo 9.

Se propone la revisión de la plantilla Modelo de Diseño, ya que es la etapa técnica que consiste en producir un modelo o representación técnica del software que se va a desarrollar. Este permite que se construya un sistema que satisfaga determinada especificación del mismo, que se ajuste a las limitaciones impuestas por el medio de destino y que respete requisitos sobre forma, rendimiento utilización de recursos.

La Lista de Chequeo que se aplicará en el documento Modelo de Diseño permite verificar cada uno de los componentes que debe tener el modelo, así como las clases de diseño, su descripción, relaciones, la identificación de los atributos y las operaciones. Ver Anexo 8.

Se toma como propuesta para la revisión de la documentación en el Flujo de Trabajo Análisis y Diseño el Documento de Arquitectura de Software, ya que este es una especificación de las ideas principales del diseño, proporciona una descripción entendible de la arquitectura del sistema software y sirve como medio de comunicación entre el arquitecto de software y otros miembros de equipo del proyecto con respecto a las decisiones arquitectónicamente significativas que se han tomado en el mismo. Contiene varias vistas que muestran aspectos distintos del sistema como son: Vista de Casos de Uso, Vista Lógica, Vista de procesos, Vista de despliegue, Vista de Implementación y Vista de Datos.

La Lista de Chequeo que se aplicará en el Documento de Arquitectura tiene como objetivo lograr que el documento expuesto se haya realizado con los elementos correspondientes. Dicha Lista exhibe diferentes especificaciones que no son recomendables obviar al realizar el Documento de Arquitectura. Ver Anexo 7.

2.3.6 Listas de Chequeo para la revisión de la documentación del Flujo de Trabajo Implementación.

Para la revisión del documento creado en el Flujo de Trabajo Implementación se propone la Lista de Chequeo para el Modelo de Despliegue, ya que muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos). Son los complementos de los diagramas de componentes que unidos proveen la vista de implementación del sistema. Ver Anexo 10.

2.3.7 Listas de Chequeo para la documentación de las Pruebas de Funcionalidad.

Las Listas de Chequeo creadas para la revisión de las funcionalidades en los disímiles módulos que serán probados, se toma como propuesta Lista de Chequeo Funcional, Lista de Chequeo de Seguridad y Lista de Chequeo de Volumen.

➤ Pruebas Funcionales.

Se propone la Lista de Chequeo Funcional para la revisión de las Pruebas de Funcionalidad, ya que esta verifica detalladamente cada función del sistema, permite comprobar la instalación del mismo y el tiempo de respuesta de cada acción que se realice. Las funciones son probadas ingresando las entradas y examinando las salidas. Esta Lista de Chequeo valida si el comportamiento observado del software cumple o no con sus especificaciones. Ver Anexo 11.

➤ Pruebas de Seguridad.

Se propone la Lista de Chequeo de Seguridad, para evidenciar si el sistema que se esta probando es totalmente confiable. A través de esta se verifica si cada usuario que debe interactuar con el sistema presenta el permiso correspondiente, permite verificar además si los componentes del sistema se encuentran disponibles cuando los usuarios lo deseen, entre otros aspectos que son de vital importancia para que el favorecido este satisfecho con el trabajo de los desarrolladores y de los probadores. Ver Anexo 12.

➤ Pruebas de Volumen.

Se propone la Lista de Chequeo de Volumen, ya que permite comprobar si el sistema que se esta probando puede funcionar adecuadamente bajo diferentes escenario de volúmenes, como son el número máximo de clientes conectados al sistema realizando la misma función y múltiples consultas de la Base de Datos ejecutadas simultáneamente. Ver Anexo 13.

2.3.8 Pruebas para verificar la documentación de la Interfaz de Usuario y Atributos de Calidad

➤ **Lista de Chequeo para Interfaz de Usuario**

Se propone una Lista de Chequeo para la revisión de la Interfaz de Usuario, ya que esta se ha convertido en una de las partes más importantes de cualquier sistema; es el vínculo entre la computadora y el usuario. Permite determinar también que tan fácil es que el programa realice lo que el usuario pretende hacer. Ver Anexo 14.

➤ **Lista de Chequeo para Atributos de Calidad**

Se propone una Lista de Chequeo de Atributos de Calidad con el objetivo de determinar en el sistema que se esta probando, la usabilidad, la seguridad que este presenta, la confiabilidad, la eficacia y la portabilidad del mismo. Ver Anexo 15.

2.4 Diseño de Casos de Prueba

El Diseño de Casos de Prueba propuesto en este capítulo, nos permite especificar aquellas pruebas que tienen la probabilidad de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y tiempo. Incluyen una descripción de la funcionalidad que se probará, la cuál es tomada ya sea de los requisitos o de los casos de uso, y la preparación requerida para asegurarse de que la prueba pueda ser dirigida. Ver Anexo 16.

El Diseño de Casos de Prueba verifica:

1. Si el producto satisface los requisitos del usuario, tal y como se describe en las especificación de los requisitos.
2. Si el producto se comporta como se desea, tal y como se describe en las especificaciones funcionales del diseño.
3. Verifica la funcionalidad de cada caso de uso del sistema que se esta probando.

2.5 Criterio de Evaluación

Se realiza un Criterio de Evaluación para cada Lista de Chequeo que se le aplican a cada una de las plantillas del Expediente de Proyecto, son utilizados como una referencia frente a la cual se determina la conformidad para cada plantilla establecida. Estos se encuentran documentados desde el Anexo 17- Anexo 31.

Ejemplo de Criterio de Evaluación

UCID

Criterio de Evaluación. Modelo de Casos de Uso del Negocio

Criterio de Evaluación

Evaluación	Evaluación Adecuada
Si se realizó el modelo del negocio	++
Si no se especificó un artefacto del modelo del negocio	-
Si no se especificó mas de un artefacto del modelo del negocio	--
Si se realizó su descripción	++
Si se realizó las descripción con poca claridad	+
Si no se realizó las descripción del negocio	--
Si se especificaron correctamente los actores del negocio	++
Si no se especificaron correctamente los actores del negocio	-
Se representó correctamente el rol del actor del negocio	++
Si se especificaron correctamente los trabajadores del negocio	++
Si no se especificaron correctamente los trabajadores del negocio	-
Si se realizó el diagrama de Casos de Uso del Negocio	++
Si no se realizó el diagrama de Casos de Uso del Negocio.	--
Si se especificaron correctamente los artefactos del diagrama de Casos de Uso del Negocio.	++
Si no se especificó correctamente un artefacto del diagrama de Casos de Uso del Negocio.	-
Si no se especificó correctamente más de un artefacto del diagrama de Casos de Uso del Negocio.	--
Si se especificó en la tabla de descripción los detalles del caso de uso en fusión de acción del actor y respuesta del sistema.	++
Si no se especificó con claridad en la tabla de descripción los detalles	+

del caso de uso en fusión de acción del actor y respuesta del sistema.	
Si no se especificó en la tabla de descripción los detalles del caso de uso.	--

Estructura de un Diagrama de Actividades

Evaluación	Evaluación Adecuada
En el caso de que el Diagrama de Actividad no presente un estado inicial.	--
En el caso de que el Diagrama de Actividad tenga un sólo estado inicial.	++
En el caso de que el Diagrama de Actividad presente más de un estado inicial.	--
El prototipo del elemento que representa las actividades en el Diagrama de Actividades es el correcto.	++
Si menos de un prototipo del elemento que representa las actividades en el Diagrama de Actividades es incorrecto.	-
Todos los prototipos del elemento que representa las actividades en el Diagrama de Actividades es el correcto.	++
Todos los prototipos del elemento que representa las actividades en el Diagrama de Actividades no es el correcto.	--
Si las actividades representadas en el Diagrama no representan al usuario la información necesaria.	--
Si al menos una actividad representada no brinda la información necesaria.	-
Si no existe transitividad de una actividad a otra actividad o a una decisión, o a una barra de sincronización o a un estado final.	--
Si no existe una actividad que esta precedida por otra actividad, una decisión o una barra de sincronización o un estado inicial.	--
En el caso que en el Diagrama no quede claro el orden en que ocurren las actividades.	--
En el caso que en el Diagrama se represente el orden en que ocurren las actividades.	++
Si las barras de sincronización representan caminos alternativos.	++
Si las barras de sincronización representan caminos concurrentes.	--

Si las calles solo se corresponden con Unidades Organizativas, Trabajadores de Negocio o Actores de Negocio.	++
Si las calles no se corresponden con Unidades Organizativas, Trabajadores de Negocio o Actores de Negocio.	--
Si es fácil y claro de comprender el flujo de las actividades del Diagrama.	++
Si no existe fácil comprensión en el flujo de cada actividad del Diagrama.	-
Si no existe claridad en el flujo de cada actividad del Diagrama.	-
Si son resaltadas aquellas actividades que se pretenden automatizar.	++
Si no se encuentran resaltadas las actividades que se pretenden automatizar.	-

Estructura del Diagrama de Clases del Modelo de Objeto de Negocio

Evaluación	Evaluación Adecuada
Si incluyen las entidades del negocio, la relación entre ellas y los trabajadores del negocio.	++
Si no se incluye una entidad del negocio, o una relación entre ellas o un trabajador del negocio.	-

Balance entre artefactos

Evaluación	Evaluación Adecuada
Si cada Caso de Uso tiene asociado al menos un Diagrama de Actividad que lo describa.	++
Si existe un Caso de Uso que no tenga asociado un Diagrama de Actividad que lo describa.	-
Si existe más de un Caso de Uso que no presente un Diagrama de Actividad que lo describa.	--
Si todas las entidades del negocio que aparecen en los Diagramas de Actividades aparecen en el Diagrama de Clases del Modelo de Objeto de Negocio.	++
Si existe una entidad del negocio que aparece en el Diagrama de Actividades pero no en el Diagrama de de Clases del Modelo de Objeto	-

de Negocio.	
Si existe mas de una entidad del negocio que aparece en el Diagrama de Actividades pero no en el Diagrama de de Clases del Modelo de Objeto de Negocio.	--
Si todos los Trabajadores del Negocio que aparecen en los Diagramas de Actividades manipulando alguna Entidad de Negocio aparecen en el Diagrama de Clases del Modelo Objeto del Negocio.	++
Si existe un Trabajador del Negocio que aparecen en los Diagramas de Actividades manipulando alguna Entidad de Negocio no aparecen en el Diagrama de Clases del Modelo Objeto del Negocio.	-
Si existe mas de un Trabajador del Negocio que aparecen en los Diagramas de Actividades manipulando alguna Entidad de Negocio no aparecen en el Diagrama de Clases del Modelo Objeto del Negocio.	--

Notas:

<Insertar nota si es necesario>

Las evaluaciones serán:

- - **Malo:** propiedad parcialmente disponible. (0)
- **Satisfactorio:** propiedad parcialmente disponible. (2)
- + **Bien:** propiedad parcialmente disponible. (4)
- ++ **Muy Bien:** propiedad muy bien implementada. (5)

2.6 Documento de No Conformidad

En el Documento de No Conformidad se recogen todas las no conformidades y recomendaciones al aplicar las Listas de Chequeo, las cuales verifican la documentación creada en cada Flujo de Trabajo en la Primera y Segunda Fase del proceso de revisión. Esta plantilla se encuentra ubicada en el Expediente de Proyecto de la UCID.

El Documento de No Conformidad debe estar estructurado de la siguiente forma:

1. Nombre de quién realiza la revisión (Nombre, Fecha, Hora).
2. Nombre del artefacto al que se le realiza el chequeo.
3. Elementos probados
4. Elementos no probados y causas

5. Etapa de detección
6. Si es significativa
7. No Significativa
8. Las observaciones que se realizan (si son errores que se encuentran o si son recomendaciones que se realizan al sistema)
9. Estado de la No Conformidad
10. Respuesta del equipo de desarrollo.

Este documento se especifica en detalle en el Anexo 14.

Esta plantilla tomara las no conformidades en la revisión de:

- Revisión de los Manuales entregables al Usuario.
- Revisión de la documentación en el Procesos del Negocio.
- Revisión de la documentación del Flujo de Trabajo Requisitos.
- Revisión de la documentación del Flujo de Trabajo Análisis y Diseño.
- Revisión de la documentación del Flujo de Trabajo Implementación.
- Pruebas Funcionalidad y Diseño de Casos de Prueba.
- Pruebas para verificar Interfaz de Usuario y Atributos de Calidad

2.7 Plan de Pruebas

Un Plan de Pruebas está constituido por un conjunto de pruebas. Cada prueba debe dejar claro qué tipo de propiedades se quieren probar (corrección, robustez, fiabilidad, amigabilidad), cómo se mide el resultado. Un Plan de Prueba debe especificar en qué consiste la prueba (hasta el último detalle de cómo se ejecuta), y debe definir cual es el resultado que se espera (identificación, tolerancia, etc.)

Un Plan de Pruebas contiene información detallada sobre los diferentes casos de prueba (planificados) para su posterior ejecución en el sistema:

- Tipos de pruebas planificadas.
- Entorno operativo de ejecución de las pruebas.
- Elementos auxiliares (ficheros, tablas, etc.) necesarios para la ejecución de los casos de prueba.
- Responsables de ejecución. Ver Anexo 2.

2.7.1 Estructura del Plan de Pruebas de Módulo

La construcción de un buen Plan de Pruebas es la piedra angular, y en consecuencia el principal factor de éxito para la puesta en práctica de un proceso de pruebas que permita entregar un software de mejor nivel. No obstante que cada esfuerzo o proceso de pruebas puede ser diferente y específico, la mayor parte de los proyectos informáticos, sean de nuevos desarrollos o de mantenimiento de aplicaciones, tienen un marco común para la realización de las pruebas.

En el Plan de Pruebas definido para la revisión de Módulo, se identifican los elementos que serán probados, los recursos necesarios para hacer las pruebas, así como la estrategia de pruebas que se llevará a cabo para lograr un buen diseño de casos de pruebas que permitan encontrar la mayor cantidad posible de defectos al software en cuestión. Se describen además, las pruebas de unidad que se aplicarán al sistema de software desarrollado, con el objetivo de probar todos los requisitos definidos en la Especificación de Casos de uso y validar y verificar las funcionalidades de la aplicación. El Plan de Pruebas se precisa el equipo de pruebas, que serán los encargados de planificar, diseñar y evaluar las pruebas, quienes además realizarán las pruebas de unidad al software. Se incluyen las especificaciones de Software y Hardware, teniendo en cuenta las herramientas y los dispositivos que se necesitan para el desarrollo de la aplicación, detallando las versiones utilizadas de cada uno de ellos y los proveedores de los mismos.

2.7.2 Estrategia de prueba.

Una estrategia de prueba integra las técnicas del Diseño de Casos de Prueba, en una serie de pasos bien planificados, que llevan a una construcción correcta del software, debe incluir pruebas de bajo nivel que verifiquen que todos los pequeños segmentos de código fuente se han implementado correctamente, así como pruebas de alto nivel que validen las principales funciones del sistema frente a los requisitos del cliente.

Debido a que los pasos de la estrategia de prueba se dan a la vez cuando aumenta la presión de los plazos fijados, se debe poder medir el progreso y los problemas deben aparecer lo antes posible.

La estrategia de pruebas desarrollada para el proceso de revisión, tiene los siguientes objetivos:

- Identificar los tipos de prueba a utilizar.
- Definir las técnicas de pruebas que se van a utilizar a lo largo del proceso.
- Definir el entorno en que se van a desarrollar las pruebas.
- Definir los casos de pruebas que se deriven de la aplicación de las técnicas de pruebas.

2.7.3 Métodos de prueba.

El método de prueba que se decidió aplicar en el Plan de Prueba, es el método de caja negra, con el fin de estudiar la especificación de las funciones, la entrada y la salida para poder derivar los casos de prueba, definiendo como algo fundamental probar todas las posibles entradas y salidas del sistema que se está construyendo.

Este método permitirá realizarle pruebas a la interfaz del software y examinar aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del programa, se centrará en el estudio de la especificación del software y las funciones que debe realizar, sus entradas y sus salidas.

2.7.4 Técnicas de pruebas de Caja Negra.

➤ Técnicas de Caja Negra. Análisis de Valores Límite (AVL).

Mediante la experiencia se ha podido constatar que los casos de prueba que exploran las condiciones límite de un programa producen un mejor resultado para la detección de defectos, es decir, es más probable que los defectos del software se acumulen en estas condiciones. Podemos definir las condiciones límite como las situaciones que se hallan directamente arriba, abajo y en los márgenes de las clases de equivalencia.

El análisis de valores límite es un técnica de diseño de casos que complementa a la de particiones de equivalencia. Las diferencias entre ambas son las siguientes:

1. Más que elegir cualquier elemento como representativo de una clase de equivalencia, se requiere la selección de uno o más elementos tal que los márgenes se sometan a prueba.
2. Más que concentrarse únicamente en el dominio de entrada (condiciones de entrada), los casos de prueba se generan considerando también el espacio de salida.

El proceso de selección de casos es también heurístico, aunque existen ciertas reglas orientativas. Aunque parezca que el AVL es simple de usar (a la vista de las reglas), su aplicación tiene múltiples matices que requieren un gran cuidado a la hora de diseñar las pruebas.

➤ Conjetura de errores.

La idea básica de esta técnica consiste en enumerar una lista de equivocaciones que pueden cometer los desarrolladores y de las situaciones propensas a ciertos errores. Después se generan casos de prueba en base a dicha lista (se suelen corresponder con defectos que aparecen comúnmente y no

con aspectos funcionales). Esta técnica también se ha denominado generación de casos (o valores) especiales, ya que no se obtienen en base a otros métodos sino mediante la intuición o la experiencia. No existen directrices eficaces que puedan ayudar a generar este tipo de casos, ya que lo único que se puede hacer es presentar algunos ejemplos típicos que reflejan esta técnica. Algunos valores a tener en cuenta para los casos especiales son los siguientes:

- El valor cero es una situación propensa a error tanto en la salida como en la entrada.
- En situaciones en las que se introduce un número variable de valores (por ejemplo, una lista), conviene centrarse en el caso de no introducir ningún valor y en el de un solo valor. También puede ser interesante una lista que tiene todos los valores iguales.
- Es recomendable imaginar que el programador pudiera haber interpretado algo mal en la especificación.
- También interesa imaginar lo que el usuario puede introducir como entrada a un programa. Se dice que se debe prever toda clase de acciones de un usuario como si fuera “completamente tonto” o, incluso, como si quisiera sabotear el programa.

2.7.5 Configuración del entorno de prueba.

La configuración del entorno donde se vayan a ejecutar las diferentes pruebas que se realizan a un software es un aspecto muy importante dentro del proceso de pruebas, pues si no se analizan bien los recursos de software y hardware que necesita el producto que se está construyendo, a la hora de probar dicho producto se prescindirá de los elementos necesarios para la ejecución de un proceso de pruebas exitoso. Se tuvo en cuenta a la hora de llevar a cabo el proceso de pruebas a los distintos módulos de la UCID requisitos de hardware y de software que hicieron posible un mejor desarrollo de las pruebas, en aras de que las mismas logran minimizar los errores en la aplicación en desarrollo.

2.8 Procedimientos de prueba.

Un procedimiento de prueba especifica como realizar uno o varios casos de prueba o parte de estos, un procedimiento de prueba puede ser una instrucción para un individuo sobre como ha de realizar un caso de prueba manualmente, o puede ser una especificación de cómo interactuar manualmente con una herramienta de automatización de pruebas para crear componentes ejecutables.

Cada caso de prueba precisará varios procedimientos de prueba, quizás uno por cada subsistema de servicio probado en el caso de pruebas. Relacionando de esta forma los procedimientos de prueba con los subsistemas de servicios que serian más fáciles de mantener.

2.9 Elementos a revisar del Modelo de Negocio.

Los elementos del Modelo de Negocio serán probados en el módulo Centro de Balance del proyecto Sistema para la Planificación Material y Financiera.

- Presentación
 - <Nombre del Proyecto>
 - <Módulo>
 - Modelo del Negocio
 - <Versión>
- Revisiones históricas
- Introducción
 - <Propósito>
 - <Alcance>
 - <Definiciones, Acrónimos y Abreviaturas>
 - <Referencias>
- Breve Descripción del Negocio
- Actores del Negocio
- Trabajadores del Negocio
- Diagrama de Casos de Uso del Negocio
- Especificación de los Casos de Uso
- Diagrama de Actividad para cada caso de uso del Negocio
- Modelo de Objetos del Negocio
- Reglas Generales del Negocio

2.10 Elementos a revisar en Requisitos

Los elementos de Requisito serán probados al módulo Centro de Balance del proyecto Sistema para la Planificación Material y Financiera.

2.10.1 Modelo de Casos de Uso del Sistema

- Presentación
 - <Nombre del Proyecto>
 - <Módulo>

Modelo de casos de uso del Sistema

<Versión>

- Control de Versiones
- Introducción
 - <Propósito>
 - <Alcance>
 - <Definiciones, Acrónimos y Abreviaturas>
 - <Referencias>
- Breve Descripción del Sistema
- Actores del Sistema
- Diagrama de Casos de Uso del Sistema
- Especificación de los Casos de Uso

2.10.2 Especificación de Requisitos

- Presentación
 - <Nombre del Proyecto>
 - <Módulo>
 - Especificación de Requisitos
 - <Versión>
- Revisiones históricas
- Introducción
 - <Propósito>
 - <Alcance>
 - <Definiciones, Acrónimos y Abreviaturas>
 - <Referencias>
- Descripción General
- Requisitos Funcionales
- Requisitos no Funcionales
 - <Usabilidad>
 - <Confiabilidad>
 - <Rendimiento>
 - <Soporte>

- <Restricciones de diseño>
- <Requisitos de ayuda y documentación>
- <Adquisición de componentes>
- <Interfaz>
- < Requisitos de licencias y patentes>
- <Aplicación de estándares >

2.10.3 Plan de Gestión de Requisitos

- Presentación
 - <Nombre del Proyecto>
 - <Módulo>
 - Plan de Gestión de Requisitos
 - <Versión>
- Control de versiones
- Introducción
 - <Propósito>
 - <Alcance>
 - <Definiciones, Acrónimos y Abreviaturas>
 - <Referencias>
- Gestión de Requisitos
- Programa de Gestión de Requisitos
 - <Identificación de Requisitos>
 - <Seguimiento>
 - <Atributos>
- Entregables del Proyecto
- Gestión de cambio a los requisitos

2.11 Elementos a revisar en Análisis y Diseño.

Los elementos del Análisis y Diseño serán probados al módulo Recepción del proyecto Contabilidad Material.

2.11.1 Modelo de Análisis

- Presentación
 - <Nombre del Proyecto>
 - <Módulo>
 - Modelo de Análisis
 - <Versión>
- Control de versiones
- Introducción
 - <Propósito>
 - <Alcance>
 - <Definiciones, Acrónimos y Abreviaturas>
 - <Referencias>
- Realización de los CU del análisis.
 - <Primer Caso de Uso>
 - <Diagrama de Clases <Caso de Uso>>
 - <Diagrama de interacción <Caso de Uso>>
 - <Descripción del flujo de sucesos <Caso de Uso>>

2.11.2 Modelo de Diseño

- Presentación
 - <Nombre del Proyecto>
 - <Módulo>
 - Modelo de Diseño
 - <Versión>
- Control de versiones
- Introducción
 - <Propósito>
 - <Alcance>
 - <Definiciones, Acrónimos y Abreviaturas>
 - <Referencias>
- Diagrama de Paquetes
- Diagrama de Clases
- Descripción de Clases

<Capa << Nombre de la Capa >>>

- <<No. del Módulo>>.<<No. de la Clase>> Clase <<Nombre de la Clase>>
- <Propósito>
- <Descripción>
- <Observaciones>

2.11.3 Documento de Arquitectura de Software

- Presentación
 - <Nombre del Proyecto>
 - <Módulo>
 - Documento de Arquitectura de Software
 - <Versión>
- Control de versiones
- Introducción
 - <Propósito>
 - <Alcance>
 - <Definiciones, Acrónimos y Abreviaturas>
 - <Referencias>
- Representación Arquitectónica
- Objetivos y Restricciones Arquitectónicas
- Tamaño y Rendimiento
- Vista de casos de uso
 - <Casos de uso arquitectónicamente significantes>
 - <Nombre de caso de uso: Breve descripción. >
 - <Realización del caso de uso: Documentación de la realización del caso de uso, <Diagrama de interacción.>
- Vista Lógica
 - <Elementos del modelo arquitectónicamente significantes>
 - <Visión general de la arquitectura – Alineamiento de paquetes, subsistemas y capaz >
- Vista de procesos
 - <Diagrama de vista de procesos que muestra la composición de los procesos e hilos, y la distribución de clase en estos procesos e hilos.>

- Diagrama de despliegue
 - <Nombre de dispositivo: descripción de la capacidad que el dispositivo provee al sistema.>
 - <Nombre del procesador: descripción de la funcionalidad y capacidad del nodo.>
- Descripción de elementos e interfaces de comunicación
 - <<<Nombre tipo de conexión>>: Características físicas de la conexión>
- Vista de Implementación
- Vista de Datos
- Calidad

2.12 Elementos a revisar en Implementación

Los elementos de la Implementación serán probados al módulo Reclutamiento del proyecto Comités Militar.

2.12.1 Modelo de Despliegue

- Presentación
 - <Nombre del Proyecto>
 - <Módulo>
 - Modelo de Despliegue
 - <Versión>
- Control de versiones
- Introducción
 - <Propósito>
 - <Alcance>
 - <Definiciones, Acrónimos y Abreviaturas>
 - <Referencias>
- Diagrama de Despliegue
- Descripción de Nodos
 - < Requisitos de Hardware>
 - <Subsistemas de Implementación>

2.13 Manuales a revisar del sistema

Los manuales entregables al usuario (Manual de Usuario, Manual de Instalación, Manual de Tecnología) fueron chequeados en el módulo Reclutamiento del proyecto Comités Militar.

2.13.1 Manual de Usuario

Los elementos a probar del mismo son:

- Estructura de la documentación para el usuario del software.
- Componentes de software usados en la documentación.
- Tabla de contenido.
- Lista de Ilustraciones, de Tablas o de Figuras.
- Introducción.
- Información para el uso de la documentación.
- Procedimientos.
- Información en comandos de software o instrucciones informáticas.
- Mensajes de error y resolución de problemas.
- Glosario.
- Fuentes relacionadas con la información.
- Características de navegabilidad.
- Índice.
- Capacidad de búsqueda.
- Estructura global de la documentación.
- Ubicación de la información crítica.
- Contenido de la información.
- Integridad de la información.
- Veracidad de la información.
- El formato de la documentación informática del usuario
- La consistencia de terminología y los formatos.
- Uso de impresora o formato electrónico.
- La legibilidad.
- Formatos para advertencias, precauciones, y notas.
- Formato para instrucciones.
- Formatos para representar elementos de interfaces de usuario.

- Formatos para las características de la documentación para acceder a la información.
- Formatos para ilustraciones.

2.13.2 Manual de Instalación

Los elementos a probar del mismo son:

- Pasos de la Instalación.
- Tratamiento de errores de la Instalación.

2.13.3 Manual de Tecnología

Los elementos a probar del mismo son:

- Estado actual de los dispositivos externos.
- Estado actual de los dispositivos internos.

2.14 Elementos a revisar mediante las Pruebas de Funcionalidad

Las pruebas de funcionalidad fueron chequeados en el módulo Reserva del proyecto Comités Militar.

2.14.1 Pruebas Funcionales

En las Listas de Chequeos para las Pruebas Funcionales se verificaran elementos como:

- Instalación del sistema.
- Función de la aplicación con el hardware y el software propuesto.
- Funcionamiento generalizado de la aplicación.

2.14.2 Prueba de Seguridad

En las Listas de Chequeos para las Pruebas de Seguridad se verificaran elementos como:

- Autenticidad del Sistema.
- Permisos de cada usuario.

2.14.3 Prueba de Volumen

En las Listas de Chequeos para las Pruebas de Volumen se verificaran elementos como:

- Soporte del Sistema.
- Soporte de la Base de Datos.

2.15 Elementos a revisar en la Interfaz de Usuario y en los Atributos de Calidad que este presenta.

Las pruebas de Interfaz de Usuario y Atributos de Calidad fueron realizadas en el módulo Reserva del proyecto Comités Militar.

2.15.1 Interfaz de Usuario

En las Listas de Chequeos para verificar la Interfaz de Usuario se comprobaran elementos como:

- Características específicas de la Interfaz del sistema.
 - <Estilos>
 - <Mensajes>
- Elementos de diseño.

2.15.2 Atributos de Calidad

En las Listas de Chequeos para verificar los Atributos de Calidad se comprobaran elementos como:

- Usabilidad
- Seguridad.
- Confiabilidad
- Eficiencia
- Portabilidad

2.16 Elementos del Plan de Prueba

2.16.1 Plan de Prueba

- Presentación
 - <Nombre del Proyecto>
 - <Módulo>
- Introducción
 - <Propósito>
 - < Alcance >
- Organización del Equipo de Prueba Arquitectura técnica
- Especificaciones del Software y Hardware.
- Descripción del Plan de Pruebas.
 - <Descripción de los requisitos.>
- Estrategia de Prueba

<Objetivo>

<Método de prueba>

<Técnicas de prueba>

<Entorno de Prueba (Descripción de los casos de Usos)>

<Casos de Prueba.>

<Criterios de Término>

< Herramientas>

- Documentación de los Resultados

2.17 Plantilla de Diseño de Casos de Prueba.

El Diseño de Casos de Prueba fue aplicado como muestra a unos de los casos de uso del módulo Reserva del proyecto Comités Militar.

2.17.1 Diseño de Casos de Prueba

- Control de Versiones.
 - <Descripción General>
 - <Condiciones de Ejecución>
- Secciones a probar en el Caso de Uso.
- Registro de defectos y dificultades detectados.
- Anexos

2.18 Conclusiones

En este capítulo se modelan las Listas de Chequeo para cada Flujo de Trabajo, incluyendo los Manuales entregables al Usuario, teniendo en cuenta además las Pruebas de Funcionalidad, Interfaz del Usuario y para los Atributos de Calidad del mismo. Se definió la estructura del Diseño de Casos de Prueba, así como el Documento de No Conformidad donde quedarán plasmadas las no conformidades encontradas en las pruebas realizadas posteriormente.

Resultados al aplicar las Pruebas

3.1 Introducción.

Este capítulo hará referencia al análisis de los resultados obtenidos luego de diseñar y aplicar el proceso de pruebas y revisión a las plantillas del Expediente de Proyecto en diferentes etapas de desarrollo en los módulos Centro de Balance, Recepción, Reclutamiento y Reserva. Para poner en práctica dicho análisis se hará una pequeña valoración de las No Conformidades encontradas en cada una de las pruebas aplicadas en las diferentes fases (Primera y Segunda), por último se creará una clasificación de los errores mayormente encontrados y se explicará el proceso que se llevo a cabo en Plan de Pruebas de Módulo que fue diseñado.

3.2 Elementos a probar, análisis de los resultados de las pruebas realizadas del Flujo de Trabajo de Negocio.

Elementos a Probar:

- Modelo de Casos de Uso del Negocio

Análisis de los resultados obtenidos en la revisión de las plantillas del Expediente de Proyecto del módulo Centro de Balance del proyecto Sistema para la Planificación Material y Financiera:

Las pruebas a la documentación del Flujo de Trabajo del Negocio fueron diseñadas y realizadas por 2 probadoras. En la Primera Fase se verificó el documento Modelo de Casos de Uso del Negocio mediante la Lista de Chequeo realizada, se evaluó el trabajo efectuado y los resultados obtenidos mediante el Criterio de Evaluación, el cual proporciona una evaluación final. Posteriormente los errores encontrados en este documento fueron especificados en el Documento de No Conformidad con el fin de pasar a la Segunda Fase donde se evalúan los errores ya informados para establecer una última valoración del trabajo realizado al módulo en cuanto a la Ingeniería de Software que se especifica en cada una de las plantillas del Expediente de Proyecto.

Ejemplo de Lista de Chequeo probada en la Primera Fase.

UCID

Modelo de Casos de Uso del Negocio

Proyecto	Módulo	Etapas	Versión	Fecha
Sistema para la Planificación Material y Financiera.	Centro de Balance	Fase de Inicio, Flujo de trabajo Negocio.	<1.0>	6/05/08

Control de Versiones:

Fecha	Versión	Descripción	Autor
6/05/08	<1.0>	Revisión de la Plantilla de Modelo de Casos de Uso del Negocio ubicada en el Expediente de Proyecto.	Annalie Pacheco, Yudisleidis Báez

No.	Evaluación	Eval.	NP.	Comentario
1	¿Se realizó el Modelo de Negocio, así como su breve descripción?	4		Se realizó el modelo de Negocio, pero algunos de los artefactos tienen señalamientos.
2	¿Se han especificado los actores del negocio?	5		
3	¿El nombre que presenta el actor del negocio, es el rol que este desempeña en el mismo?	5		
4	¿Se especificaron correctamente los trabajadores del negocio?	5		
5	¿Se realizó correctamente el diagrama de CU de Negocio?	2		En el diagrama el estereotipo de los CU (Casos de Usos) no son del Negocio. Los Actores representan el estereotipo de Negocio y los CU (Casos de

				Usos) no.
6	¿Se especificaron correctamente los casos de usos, así como su descripción?	4		Los CU (Casos de Usos) muy brevemente descritos.
7	¿Se describe en la tabla los detalles del caso de uso en fusión de acción del actor y respuesta del sistema?	2		Las acciones deben de ser explicadas un poco mas detallada, además faltan datos por especificar.

Diagrama de Actividad

No.	Evaluación	Eval.	NP.	Comentario
8	¿Se realizó el diagrama de Actividades?	5		
9	¿Los Diagramas de Actividad tienen bien representados su único estado inicial y al menos un estado final?	5		
10	¿Las actividades transitan siempre a: otra actividad, una decisión, una barra de sincronización o un estado final?	5		
11	¿Las actividades están precedida por: otra actividad, una decisión, una barra de sincronización o un estado inicial?	5		
12	¿En el diagrama quedan claro el orden en que ocurren las actividades y además quedan claro cuando el orden de las actividades no es significativo (pueden ejecutarse en paralelo) utilizando para ello, barras de sincronización?	5		
13	¿Se utilizan barra de sincronización para representar caminos alternativos?	5		
14	¿Todas las actividades que están descritas dentro de un Diagrama de Actividad son posibles?	5		
15	¿Las calles sólo se corresponden con	5		

	Unidades Organizativas Trabajadores de Negocio o Actores de Negocio?			
16	Una misma Entidad de Negocio puede ser utilizada como entrada (con un mismo estado) en más de una actividad.		X	
17	Siempre que una actividad del diagrama transforme una Entidad de Negocio debe señalarse ésta como entrada y salida pero con dos estados diferentes.		X	
18	Varias actividades no pueden hacer igual transformación a la misma Entidad de Negocio.	5		
19	¿El flujo de actividades del diagrama es claro y fácil de comprender?	2		En la plantilla no se define el flujo de actividades es decir la comprensión del texto.
20	¿Para indicar en el Diagrama de Actividad la ejecución de un Caso de Uso extendido o incluido se colocaron las actividades con el mismo nombre del Caso de Uso? Se sugiere resaltar dicha actividad		X	
21	¿Están resaltadas dentro del diagrama aquellas actividades que se pretenden automatizar?	5		

Diagrama de Clase del Modelo de Objeto del Negocio

No.	Evaluación	Eval.	NP.	Comentario
22	¿Se realizó el Modelo de Objeto?	5		
23	¿Se incluyeron: las Entidades de Negocio, la relación que existe entre ellas y los Trabajadores de Negocio que interactúan con dichas entidades?	5		

Balance entre artefactos

No.	Evaluación	Eval.	NP.	Comentario
24	¿Cada Caso de Uso tiene asociado al menos un Diagrama de Actividad que lo describa?	5		
25	¿Las Entidades de Negocio que aparecen en los Diagramas de Actividades aparecen en el Diagrama de Clases del Modelo Objeto de Negocio?	5		
26	¿Los Trabajadores de Negocio que aparecen en los Diagramas de Actividades manipulando alguna Entidad de Negocio aparecen en el Diagrama de Clases del Modelo Objeto de Negocio?	5		

Notas:

<Insertar nota si es necesario>

Cant 5	Cant 4	Cant 2	Cant 0
18	2	3	0

$$\text{Real} = \text{Cant } 5 * 5 + \text{Cant } 4 * 4 + \text{Cant } 2 * 2 + \text{Cant } 0 * 0$$

$$\text{Real} = 18 * 5 + 2 * 4 + 3 * 2 + 0 * 0 = 104$$

$$\text{Ideal} = (\text{Cant } 5 + \text{Cant } 4 + \text{Cant } 2 + \text{Cant } 0) * 5$$

$$\text{Ideal} = 23 * 5 = 115$$

Real-----X

Ideal----100

$$X = (\text{Real} * 100) / \text{Ideal}$$

$$X = 104 * 100 / 115$$

$$X = 90\%$$

Leyenda

Eval. : Evaluación que se le otorga al aspecto a evaluar: [0, 2, 4,5,].

NP. : No procede (se marca con una **X** la pregunta que no será necesaria realizarla en el sistema que se le esta aplicando la lista de chequeo).

Comentario: Es obligatorio en las evaluaciones distintas de 5 puntos.

Las evaluaciones serán:

- - **Malo:** propiedad parcialmente disponible. (0)
- **Satisfactorio:** propiedad parcialmente disponible. (2)
- + **Bien:** propiedad parcialmente disponible. (4)
- ++ **Excelente:** propiedad muy bien implementada. (5)

Criterio de evaluación

- () Sin modificaciones.
- (**X**) Pequeñas modificaciones.
- () Grandes modificaciones.
- () Nueva elaboración.

Conclusión

	Aceptado
X	Diferido
	No Aceptado

- **Plantilla de No Conformidad**

Los errores encontrados en la Primera Fase al aplicar las Listas de Chequeos, quedaran plasmados en el Documento de No Conformidad para que los desarrolladores puedan validar las fallas encontradas en las plantillas del Expediente de Proyecto. Siguiendo el mismo ejemplo de Modelo de Casos de Uso del Negocio el Documento de No Conformidad quedaría de la siguiente manera:



No Conformidades Detectadas

Proyecto	Módulo	Versión
Sistema para la Planificación Material y Financiera.	Centro de Balance.	1.0

Control de Versiones:

Fecha	Versión	Descripción	Autor
6/05/08	1.0	Revisión del Modelo de Casos de Uso del Negocio aplicando la técnica de lista de chequeo para evaluar la calidad del mismo.	Annalie Pacheco, Yudisleidis Báez.

1. Aspectos Generales

Para llevar a cabo la revisión del Modelo de Casos de Uso del Negocio se aplicó una Lista de Chequeo que verifica la correcta realización de este flujo de trabajo. Para desarrollar de manera eficiente este proceso se tuvo en cuenta aspectos como: Conformidad con la estructura de la Plantilla del Expediente de Proyecto, formato del documento, concordancia, redacción, conocimientos de Ingeniería de Software, conceptos del negocio y la correcta realización de cada uno de los artefactos.

1.1 Elementos Probados.

Se revisaron los artefactos y todas las informaciones recogidas en los acápites pertenecientes al flujo de trabajo de Negocio especificados en la Plantilla correspondiente ubicada en el Expediente de Proyecto.

1.2 Elementos no Probados y causas.

Diagrama de Actividades: No se distingue el flujo de actividades. (Contenido)

2. Tabla de No Conformidades Detectadas

Elemento	No conformidad	Aspecto correspondiente	Etapas de detección	Significativa	No Significativa	Recomendación	Estado NC	Respuesta del Equipo Desarrollo	
Modelo de CUN.	1.	Se encierran el nombre del proyecto, modulo y la versión entre símbolos de comparación .	Portada	Prueba		X	Los signos que encierran el nombre del proyecto, modulo y la versión deben eliminarse porque forman parte de la plantilla no del documento oficial	Pendiente.	
	2.	Se encierra la versión entre símbolos de comparación .	Revisión Historias, Versiones	Prueba		X	Los signos que encierran la versión deben eliminarse porque forman parte de la plantilla no del documento oficial.	Pendiente.	
	3.	Se escribe el nombre del flujo de trabajo con minúscula (modelo del negocio).	Propósito.	Prueba		X	Especificar el nombre con mayúscula.	Pendiente	

	4.	El Diagrama de CUN tiene diferentes estereotipos.	Diagrama CUN.	Prueba	X		Realizar el diagrama con los estereotipos del negocio que corresponden a la metodología seleccionada.	Pendiente.	
	5.	La descripción de los casos de uso no esta completa faltan detalles por precisar. (Pos condiciones y Mejoras.)	Especificación de los Casos de Uso.	Prueba	X		Hacer un poco más entendible la descripción y especificar los detalles que faltan. En el caso que esos detalles no existan no se eliminan de la tabla.	Pendiente	
	6.	No es fácil de comprender el flujo de actividades que se representa en el Diagrama, porque el mismo es muy pequeño.	Diagrama de actividades.	Prueba		X	Representar de manera que el diagrama se pueda ver sin utilización del zoom.	Pendiente	

	7.	En el diagrama de CU del Negocio el nombre del Actor no es igual al de la descripción.	Descripción de los actores del negocio. (Órganos Consumidores)	Prueba	X		El nombre del rol del Actor debe ser igual al de su descripción.	Pendiente.	
--	----	----------------------------------------------------------------------------------------	----------------------------------------------------------------	--------	---	--	------------------------------------------------------------------	------------	--

3.3 Elementos a probar, análisis de los resultados de las pruebas realizadas del Flujo de Trabajo de Requisitos.

Elementos a Probar:

- Plantilla Modelo de Casos de Uso del Sistema
- Plantilla Plan de Gestión de Requisitos
- Plantilla Especificación de Requisitos

Análisis de los resultados obtenidos en la revisión de las plantillas del Expediente de Proyecto del módulo Centro de Balance del proyecto Sistema para la Planificación Material y Financiera:

Seguidamente se realizó las pruebas del Flujo de Trabajo de Requisitos, donde se encontraron determinados errores, los cuales fueron especificados en el Documento de No conformidad. En su Primera Fase se efectuó una prueba completa a la plantilla mediante las Listas de Chequeo realizadas para cumplir este fin, posteriormente se evaluó el trabajo efectuado y los resultados obtenidos por el Criterio de Evaluación, donde se proporcionó su evaluación final con el fin de pasar a la Segunda Fase donde se evalúan los errores ya definidos en el Documento de No Conformidad para establecer una última valoración del trabajo realizado al módulo en cuanto a la Ingeniería de Software que se especifica en cada una de las plantillas del Expediente de Proyecto.

3.4 Elementos a probar, análisis de los resultados de las pruebas realizadas del Flujo de Trabajo de Análisis y Diseño.

Elementos a Probar:

- Plantilla del Modelo de Análisis
- Plantilla del Modelo de Diseño
- Documento de Arquitectura de Software

Análisis de los resultados obtenidos en la revisión de las plantillas del Expediente de Proyecto del Módulo de Recepción del Proyecto Contabilidad Material:

Al aplicarle las pruebas al Flujo de Trabajo de Análisis y Diseño se hallaron algunos errores los cuales fueron detallados en el Documento de No Conformidad. En su Primera Fase se efectuó una prueba completa a la plantilla mediante las Listas de Chequeo realizadas para cumplir este fin, posteriormente se evaluó el trabajo efectuado y los resultados obtenidos por el Criterio de Evaluación, donde se proporcionó su evaluación final con el fin de pasar a la Segunda Fase donde se evalúan los errores ya informados para establecer una última valoración del trabajo realizado al módulo en cuanto a la Ingeniería de Software que se especifica en cada una de las plantillas del Expediente de Proyecto.

3.5 Elementos a probar, análisis de los resultados de las pruebas realizadas al Flujo de Trabajo de Implementación.

Elementos a Probar:

- Plantilla del Modelo de Despliegue

Análisis de los resultados obtenidos en la revisión de las plantillas del Expediente de Proyecto del Módulo de Reclutamiento del Proyecto Comités Militar:

Al aplicarle la pruebas al Flujo de Trabajo de Implementación se encontraron diferentes errores los cuales fueron especificados en el Documento de No Conformidades, al igual que en los flujos anteriores en su Primera Fase se efectuó una prueba completa a la plantilla mediante las Listas de Chequeo realizadas para cumplir este fin, posteriormente se evaluó el trabajo efectuado y los resultados obtenidos por el Criterio de Evaluación, donde se proporcionó su evaluación final con el fin de pasar a la Segunda Fase donde se evalúan los errores ya informados para establecer una última valoración del trabajo realizado al módulo en cuanto a la Ingeniería de Software que se especifica en cada una de las plantillas del Expediente de Proyecto.

3.6 Elementos a probar, análisis de los resultados de las pruebas realizadas en los Manuales.

Elementos a Probar:

- Manual de Usuario
- Manual de Instalación

Análisis de los resultados obtenidos en la revisión de las estructuras de los manuales expuestos en el Expediente de Proyecto del Módulo de Reclutamiento del Proyecto Comités Militar:

En la Primera Fase se realizó una prueba completa al Manual de Usuario donde se evaluó el trabajo realizado por parte del equipo de desarrollo y los resultados obtenidos de acuerdo al Criterio de Evaluación. En correspondencia de los resultados obtenidos se establece una evaluación final. Una vez concluida esta fase el equipo de calidad y prueba entrega los resultados en el Documento de No Conformidad al equipo de desarrollo, donde el mismo procede a la corrección de errores detectados.

Se realizó una prueba completa al Manual Tecnología y al Manual de Instalación los cuales están establecidos en este caso como un todo en el Manual de Instalación, donde se evaluó el trabajo realizado por parte del equipo de desarrollo y los resultados obtenidos de acuerdo al Criterio de Evaluación. En correspondencia de los resultados obtenidos se establece una evaluación final. Una vez concluida esta fase el equipo de calidad y prueba entrega los resultados en el Documento de No Conformidad al equipo de desarrollo, donde el mismo procede a la corrección de errores detectados.

3.7 Elementos a probar y análisis de los resultados de las Pruebas al Sistema.

Elementos a Probar.

- Pruebas de Funcionalidad.
- Interfaz de Usuario.
- Atributos de Calidad.

Pruebas de Funcionalidad

- Pruebas de Seguridad
- Pruebas Funcionales
- Pruebas de Volumen

Análisis de los resultados obtenidos en la revisión las Pruebas de Funcionalidad del Módulo de Reserva del Proyecto Comités Militar:

Al aplicar las pruebas de funcionalidad al sistema se comprobó cada una de las especificaciones de confiabilidad, integridad y eficiencia del software en desarrollo además se verificar los campos de entrada de datos, almacenamiento, instalación del sistema, especificaciones de hardware y de software y navegabilidad.

Análisis de los resultados obtenidos en la revisión las Pruebas Interfaz de Usuario del Módulo Reserva del Proyecto Comités Militar:

Para la revisión de la Interfaz de Usuario se tuvieron en cuenta aspectos como la correspondencia con los términos y experiencia del usuario, que la información que se brinda este libre de errores, la existencia de menú así como su función, comandos y opciones apropiadas, funciones de los botones mensajes que brinda el sistema, color, entre otras funciones que se miden en la interfaz de un sistema.

Análisis de los resultados obtenidos en la revisión de los Atributos de Calidad del Módulo Reserva del Proyecto Comités Militar:

Para la revisión de los Atributos de Calidad se tuvieron en cuenta aspectos como idiomas para las instrucciones, ausencia de términos en idiomas diferentes o mezclados, tiempo suficiente para realizar las entradas por teclado, opciones que brinda el producto, asistencia para los usuarios que hacen uso del sistema por primera vez, soporte apropiado a los usuarios más novatos, entre otras operaciones que se miden a partir de los atributos de robustez, eficiencia, corrección, etc.

3.8 Segunda Fase. Respuesta a las No Conformidades.

Al ser entregados a los desarrolladores todos los Documentos de No Conformidad de las revisiones realizadas en la Primera Fase por el equipo de prueba, estos deben corregir las no conformidades expuestas en el mismo, una vez concluida las revisiones y correcciones por el equipo de desarrollo se pasa a la Segunda Fase donde se iteran las revisiones con el objetivo verificar que las no conformidades hayan sido solucionadas y localizar posibles errores introducidos.

Ejemplo de respuesta del proyecto Planificación Material y Financiera del módulo Centro de Balance del Flujo de Trabajo de Negocio.

UCID

No Conformidades Detectadas

Proyecto	Módulo	Versión
Sistema para la Planificación Material y Financiera.	Centro de Balance.	1.0

Control de Versiones:

Fecha	Versión	Descripción	Autor
6/05/08	1.0	Revisión del Modelo de Casos de Uso del Negocio aplicando la técnica de lista de chequeo para evaluar la calidad del mismo.	Annalie Pacheco, Yudisleidis Baez.

1. Aspectos Generales

Para llevar a cabo la revisión del Modelo de Casos de Uso del Negocio se aplicó una Lista de Chequeo que verifica la correcta realización de este flujo de trabajo. Para desarrollar de manera eficiente este proceso se tuvo en cuenta aspectos como: Conformidad con la estructura de la Plantilla del Expediente de Proyecto, formato del documento, concordancia, redacción, conocimientos de Ingeniería de Software, conceptos del negocio y la correcta realización de cada uno de los Artefactos.

1.1 Elementos Probados.

Se revisaron los artefactos y todas las informaciones recogidas en los acápites pertenecientes al flujo de trabajo de Negocio especificados en la Plantilla correspondiente ubicada en el Expediente de Proyecto.

1.2 Elementos no Probados y causas.

Elemento	No	No conformidad	Aspecto correspondiente	Etapas de detección	Significativa	No Significativa	Recomendación	Estado NC	Respuesta del Equipo de Desarrollo
Modelo de CUN.	1.	Se encierran el nombre del proyecto, modulo y la versión entre símbolos de comparación.	Portada	Prueba		X	Los signos que encierran el nombre del proyecto, modulo y la versión deben eliminarse porque forman parte de la plantilla no del documento oficial.	Pendiente.	Resuelto por el equipo de desarrollo
	2.	Se encierra la versión entre símbolos de comparación.	Revisiones Históricas Versión.	Prueba		X	Los signos que encierran la versión deben eliminarse porque forman parte de la plantilla no del documento oficial.	Pendiente.	Resuelto por el equipo de desarrollo
	3.	Se escribe el nombre del flujo de trabajo con minúscula	Propósito	Prueba		X	Especificar el nombre con mayúscula.	Pendiente	Resuelto por el equipo de desarrollo

		(modelo del negocio).						
4.	El Diagrama de CUN tiene diferentes estereotipos.	Diagrama CUN.	Prueba	X		Realizar el diagrama con los estereotipos del negocio que corresponden a la metodología seleccionada.	Pendiente.	Resuelto por el equipo de desarrollo
5.	La descripción de los casos de uso no esta completa faltan detalles por precisar. (Pos condiciones y Mejoras.)	Especificación de los Casos de Uso.	Prueba	X		Hacer un poco más entendible la descripción y especificar los detalles que faltan. En el caso que esos detalles no existan no se eliminan de la tabla.	Pendiente	Resuelto por el equipo de desarrollo
6.	No es fácil de comprender el flujo de actividades que se representa en el Diagrama,	Diagrama de actividades.	Prueba		X	Representar de manera que el diagrama se pueda ver sin utilización del zoom.	Pendiente	Resuelto por el equipo de desarrollo

		porque el mismo es muy pequeño.							
	7	En el diagrama de CU del Negocio el nombre del Actor no es igual al de la descripción.	Descripción de los actores del negocio. (Órganos Consumidores)	Prueba	X		El nombre del rol del Actor debe ser igual al de su descripción.	Pendiente.	Resuelto por el equipo de desarrollo

3.9 Resultados Obtenidos

Al aplicar las Listas de Chequeo anteriormente expuestas, se puede llegar a una estimación real dada en por ciento de la efectividad del artefacto evaluado. Esto es mostrado en la Tabla 2.

Artefactos	Flujos de Trabajo						
	Negocio	Requisito	Análisis y Diseño	Implementación	Funcionalidad	Atributos de Calidad	Interfaz de Usuario
Modelo CUN	90%						
Especificación Requisitos		94%					
Plan de Gestión		0					
Modelo CUS		95%					
Modelo Análisis			100%				

Modelo de Diseño			93%				
Documento de Arquitectura de Software			93%				
Despliegue				95%			
Funcional					99%		
Volumen					100%		
Seguridad					98%		
Atributo de Calidad						98%	
Interfaz							98%

Tabla 2 : Por ciento Real de la Lista de Chequeo a través del Criterio de Evaluación.

La figura que se muestra a continuación corresponde a la tabla anterior.

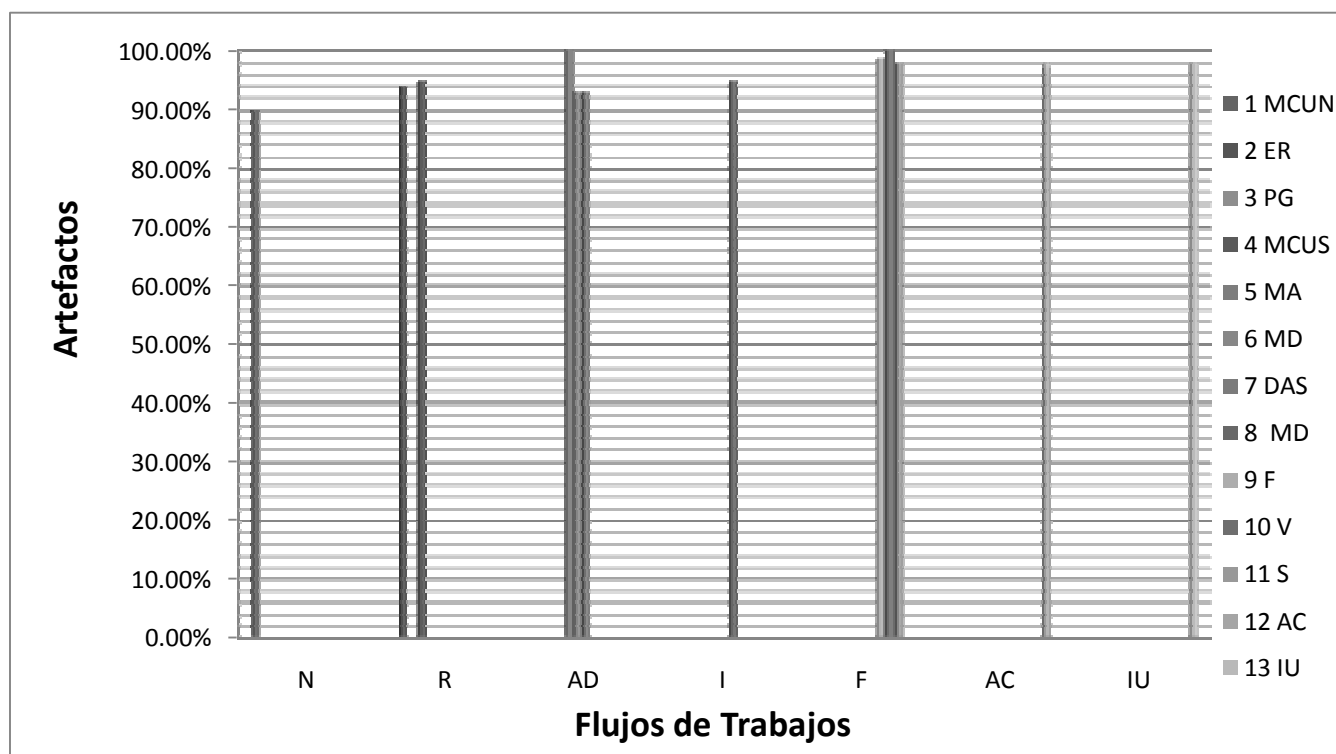


Figura 1: Por ciento reales de cada artefacto evaluado

3.10 Conclusiones

En el período de prueba que fueron sometidos los diferentes módulos se pudieron detectar disímiles errores, los cuales se expusieron en los Documentos de No Conformidades, haciendo una valoración de los resultados al aplicar cada una de las pruebas propuestas en la Primera y Segunda Fase, posibilitando que los desarrolladores de cada módulo conocieran cada falta cometida en su labor. Se definió un nivel de clasificación de acuerdo a la magnitud de los errores detectados. Para dar cumplimiento al objetivo general se realizó un Plan de Pruebas de Módulo, el cual es una guía que facilita el trabajo de cada uno de los probadores obteniendo un mejor resultado en cada prueba realizada.

CONCLUSIONES GENERALES

La presente tesis demostró la necesidad que tienen cada uno de los probadores de poseer una guía, una estrategia a seguir cuando se desea analizar, probar algún sistema. A raíz de esta se realizaron dos fases de pruebas. En la Primera Fase se recogen cada una de las pruebas que serán aplicadas a un software en el Laboratorio de Calidad y en la Segunda Fase se realizan diversas revisiones tanto para validar los errores existidos en la Primera Fase, como para erradicar aquel nuevo que se introdujo y no ha sido detectado. También se realizaron Listas de Chequeo para cada documento expuesto en el Expediente de Proyecto de la UCID, con el fin de que cada probador realice su trabajo con mayor facilidad y mejor organización.

Para cada una de estas Listas de Chequeo se realizó un Criterio de Evaluación con el fin de evaluar cada aspecto de los documentos realizados por los desarrolladores, posteriormente se especificaron las no conformidades encontradas en el Documento de No Conformidades con el fin de que estas lleguen a manos de los desarrolladores de forma organizada y con claridad en lo que se plantea por parte de los probadores.

Como resultado final se obtiene un Plan de Pruebas de Módulo para la UCID que proporciona un mejor resultado y desempeño en el trabajo realizado por los probadores en el Laboratorio de Calidad, permitiendo que cada usuario quede satisfecho con el producto obtenido.

RECOMENDACIONES

Se recomienda capacitar a los grupos de calidad de cada proyecto del Centro con el objetivo de que puedan aplicar correctamente el Plan de Pruebas de Módulo definido.

Mejorar el Plan de Pruebas de acuerdo con las políticas que posteriormente el Laboratorio de Calidad defina para de esta manera lograr que los productos que aquí se realizan salgan con una buena calidad.

REFERENCIAS BIBLIOGRÁFICAS

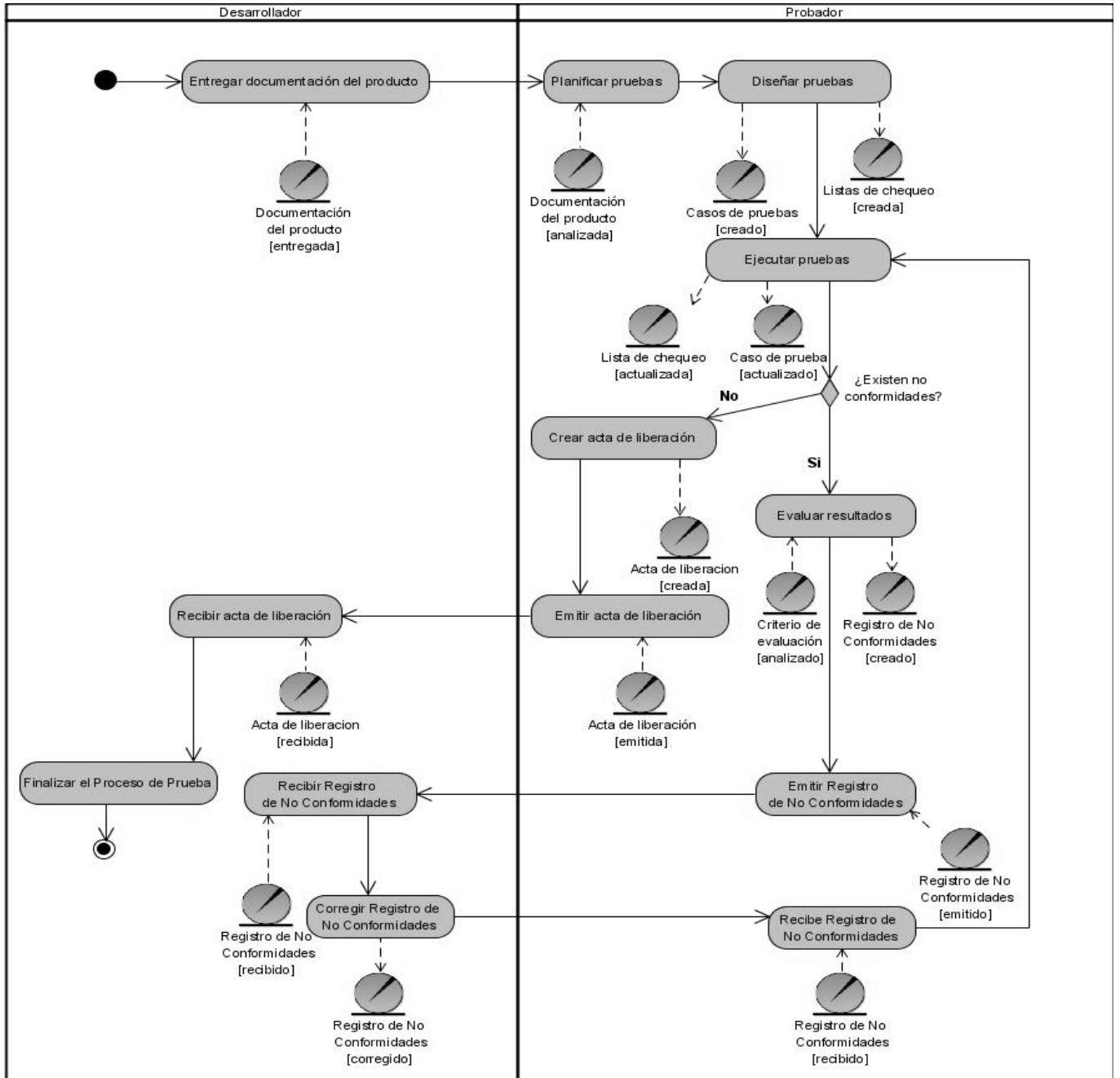
1. Pressman, *Ingeniería de Software. Un Enfoque Práctico*. 1992.
2. Pressman, *Ingeniería de Software. Un Enfoque Práctico*. 1998.
3. IEEE (1990) *Calidad de Software*. Volume,
4. 9000, I. *Calidad*. Volume,
5. Hernández, M.A.C. *Modelo para Pruebas de Software y auditoria en Entorno Microsoft.Net.* . Volume,
6. 1990, I. *Diseño de la interfaz de usuario. Pruebas del software*. Volume,
7. *Prueba de software*. Volume,
8. *Pruebas de Software*. Volume,
9. Mañas, J.A., *Prueba de Programas*. 16 de marzo de 1994.
10. Rubio, F.Ó.G. *Pruebas del Software*. Volume,
11. *VALIDACIÓN*.

BIBLIOGRAFÍAS

1. Pressman, *Ingeniería de Software. Un Enfoque Práctico*. 1992.
2. Pressman, *Ingeniería de Software. Un Enfoque Práctico*. 1998.
3. IEEE (1990) *Calidad de Software*. Volume,
4. 9000, I. *Calidad*. Volume,
5. Hernández, M.A.C. *Modelo para Pruebas de Software y auditoria en Entorno Microsoft.Net.* . Volume,
6. 1990, I. *Diseño de la interfaz de usuario. Pruebas del software*. Volume,
7. González, *Un plan de Pruebas Exitoso*. 2002.
8. *El Plan de Prueba*.
9. Mañas, J.A., *Prueba de Programas*. 16 de marzo de 1994.
10. Rubio, F.Ó.G. *Pruebas del Software*. Volume,
11. *Proceso de Prueba* Volume,
12. *Pruebas de Software* Volume,
13. *Diseño de la interfaz de usuario. Pruebas del software*. Volume,
14. *Pruebas unitarias*. Volume,
15. *Listas de Chequeo*. Volume,
16. (14 de noviembre 2006) *Diseño de Casos de Prueba*. Volume,
17. Bravo, F.O.G.y.C. *Pruebas de Sistemas Orientados a Objetos*. Volume,
18. Mañas (2003) *Calidad de las Pruebas*. Volume,
19. Moreno (2003) *Pruebas*. Volume,
20. Pressman, *Ingeniería de Software. Un enfoque Práctico*. 2002.
21. Roberto (Abril 8, 2007) *Los 5 tipos de prueba de un software*. Volume,
22. Rojas, J.C.O. *Plan de Pruebas*. Volume,

ANEXOS

Anexo#1: Proceso de Prueba.



Anexo # 2: Plan de Prueba de Módulo.

Plan de Prueba de Módulo

Elementos a Revisar: [Documentación y/o Aplicación]

Versión: 1.0

Revisiones históricas

Proyecto	Módulo	Etapas	Versión	Fecha
<ul style="list-style-type: none">• Planificación Material y Financiera.• Comités Militar.• Contabilidad Material.	<ul style="list-style-type: none">• Centro de Balance.• Reclutamiento.• Recepción.• Reserva.	Prueba	1.0	11/05/08

Plan de Pruebas

1. Introducción

Este documento se confecciona con el objetivo de definir el Plan de las Pruebas de Módulo para diferentes sistemas, teniendo en cuenta la documentación del mismo con respecto a las plantillas expuestas en el Expediente de Proyecto de la UCID. No existe una estrategia de las diferentes pruebas a aplicar a un software

Los elementos a probar en este Plan son la documentación de cada uno de los Flujos de Trabajo, como también los Manuales entregable a los Usuarios, la Interfaz del Usuario, los Atributos de Calidad, la Seguridad, la Funcionalidad y el Volumen del sistema a probar.

Se le realizan estas pruebas a diferentes módulos con el fin de definir si la técnica expuesta en el Plan de Prueba es la correcta y así validar si el comportamiento observado del software cumple o no con sus especificaciones. Estas son aplicadas también al sistema Reserva del proyecto Comités Militar.

- **Propósito**

Este Plan de Prueba tiene como propósito identificar los elementos que serán probados, los recursos necesarios para hacer las pruebas y establece los objetivos y las estrategias para los deliverables de evaluación del proyecto en parte, con el fin de lograr un buen diseño de casos de prueba que permitan encontrar la mayor cantidad de defectos al software en cuestión.

- **Alcance**

Este Plan de Prueba describe las pruebas que se aplicarán a los diferentes módulos (Centro de Balance, Reclutamiento, Recepción, Reserva), con el objetivo de obtener resultados satisfactorios durante su aplicación, así como la estrategia de pruebas que se llevará a cabo para lograr un buen diseño de casos de pruebas que permitan encontrar la mayor cantidad posible de defectos al software en cuestión.

2. Organización del Equipo de Prueba

Probadores: Annalie Pacheco Carrio

Yudisleidis Baez Ortiz

3. Arquitectura técnica.

El Documento de Arquitectura proporciona una descripción entendible de la arquitectura del sistema software y sirve como medio de comunicación entre el arquitecto de software y otros miembros de equipo del proyecto con respecto a las decisiones arquitectónicamente significativas que se han tomado en el proyecto. Contiene varias vistas que muestran aspectos distintos del sistema como son: Vista de Casos de Uso, Vista Implementación, Vista de Procesos, Vista Lógica, Vista y Vista de Despliegue. Para mayor información consultar el Documento de Arquitectura del Sistema Reserva del proyecto Comités Militar.

4. Especificaciones del Software y Hardware.

- Tarjeta de red
- Procesador Pentium III o superior, capacidad de memoria RAM 256 MB o superior
- Impresora
- Procesador Pentium 4 dual core, capacidad de memoria RAM 2 GB, 200 GB de capacidad de disco duro.

5. Descripción del Plan de Pruebas

a. Flujo de Trabajo de Negocio

En la Revisión del Flujo de Trabajo de Negocio se encierran todos los artefactos que se obtienen como resultado del Modelamiento del Negocio así como lo referente a los Modelos Casos de Uso del Negocio y Modelo del Dominio.

b. Flujo de Trabajo de Requisitos

En la Revisión del Flujo de Trabajo Requisitos se tiene en cuenta la revisión de documentos como Especificación de Requisitos, Plan de Gestión de Requisitos y Modelo de Casos de Uso del Sistema.

c. Flujo de Trabajo de Análisis y Diseño

En la Revisión del Flujo de Trabajo Análisis y Diseño se tendrá en cuenta el Modelo de Análisis, Modelo de Diseño y el Documento de Arquitectura de Software.

d. Flujo de Trabajo de Implementación

En la Revisión del Flujo de Trabajo de Implementación se verifica que los elementos del diseño se implementen en término de componentes y que se encuentren agrupados de acuerdo a los nodos específicos en el Modelo de Despliegue.

5.1 Descripción de los requerimientos.

Requerimientos Funcionales

- Permitir actualizar los datos de la tarjeta de registro de la persona
- Permitir certificar los datos actualizados de la tarjeta de registro
- Actualizar datos de teléfono de la persona
 - Insertar teléfonos
 - Modificar teléfonos
 - Eliminar teléfonos
- Actualizar datos de idioma
- Actualizar datos de nacimiento
- Actualizar datos de tallaje
- Actualizar dirección particular
- Actualizar datos de estudiantes
- Actualizar datos del trabajador
- Actualizar datos de cuenta propista
- Actualizar situación en el registro
- Actualizar ascenso en grado militar
 - Insertar grado militar
 - Modificar grado militar
 - Eliminar grado militar
- Actualizar preparación militar
 - Insertar preparación militar
 - Modificar preparación militar
 - Eliminar preparación militar
- Actualizar datos de chequeo médico
 - Insertar datos de chequeo médico

Requerimientos No Funcionales

- Tarjeta de red
- Procesador Pentium III o superior, capacidad de memoria RAM 256 MB o superior
- Impresora
- Procesador Pentium 4 dual core, capacidad de memoria RAM 2 GB, 200 GB de capacidad de disco duro.

5.1.1 Requerimientos Funcionales

- Permitir actualizar los datos de la tarjeta de registro de la persona
- Permitir certificar los datos actualizados de la tarjeta de registro

5.1.2 Requerimientos No Funcionales.

Especificaciones de software:

En el sistema a probar se utiliza el servidor Web Apache para el trabajo con PHP, se utiliza un sistema gestor de BD que soporta grandes volúmenes de datos, debido a la gran cantidad de información que se maneja, por lo que se utiliza PostgreSQL. El sistema es multiplataforma debe ser compatible con los sistemas operativos, Windows, Linux.

Especificaciones de hardware:

Para el cliente:

- Tarjeta de red
- Procesador Pentium III o superior, capacidad de memoria RAM 256 MB o superior
- Impresora

Para el servidor:

Procesador Pentium 4 dual core, capacidad de memoria RAM 2 GB, 200 GB de capacidad de disco duro.

6 Estrategia de Prueba

Las pruebas que se aplicarán en este plan son las Unitarias.

6.1 Objetivo.

La Estrategia de Prueba utilizada en este plan (Unitaria) es aplicada con el objetivo de asegurarse que cada uno de los módulos funcione correctamente por separado, además de aislar cada parte del programa y mostrar que estas individualmente son correctas.

Las pruebas que serán definidas en esta estrategia son las siguientes:

- Revisión de los Manuales entregables al Usuario

- Revisión de los Procesos del Negocio
- Revisión del Flujo de Trabajo de Requisitos
- Revisión del Flujo de Trabajo Análisis y Diseño
- Revisión del Flujo de Trabajo Implementación
- Pruebas de Funcionalidad y Diseño de Casos de Prueba
- Pruebas de Funcionalidad
- Pruebas para verificar Interfaz de Usuario y Atributos de Calidad.

6.2 Método de prueba.

El método de prueba que se decidió utilizar en el proceso de prueba fue el de Caja Negra, al proyecto Comités Milita específicamente el módulo Reserva.

6.3 Técnicas de prueba.

Se le harán revisión a los documentos expuestos en el Expediente de Proyecto de la UCID.

- Modelo de Casos de Uso del Negocio
- Modelo del Dominio
- Especificación de Requisitos
- Modelo de Casos de Uso del Sistema
- Plan de Gestión de Requisitos
- Modelo de Análisis
- Modelo de Diseño
- Documento de Arquitectura de Software
- Modelo de Despliegue
- Manual de Usuario
- Manual de Instalación
- Manual de Tecnología

Listas de Chequeo	Proyectos	Módulos
Listas de Chequeo del Modelo de Casos de Uso del Negocio.	Planificación Material y Financiera.	Centro de Balance
Listas de Chequeo del Modelo del Sistema.	Planificación Material y Financiera.	Centro de Balance
Listas de Chequeo de Especificación de Requisitos.	Planificación Material y Financiera.	Centro de Balance
Listas de Chequeo de Gestión de Requisitos.	Planificación Material y Financiera.	Centro de Balance
Listas de Chequeo del Modelos de Análisis.	Contabilidad Material	Recepción
Listas de Chequeo del Modelo de Diseño.	Contabilidad Material	Recepción
Listas de Chequeo del Documento de Arquitectura de Software.	Contabilidad Material	Recepción
Listas de Chequeo del Modelo de Despliegue.	Comités Militar	Reclutamiento
Listas de Chequeo del Manual de Usuario.	Comités Militar	Reclutamiento
Listas de Chequeo del Manual de Instalación.	Comités Militar	Reclutamiento
Listas de Chequeo del Manual de Tecnología.	Comités Militar	Reclutamiento
Lista de Chequeo de Atributos de Calidad.	Comités Militar	Reserva

Lista de Chequeo de Interfaz de Usuario.	Comités Militar	Reserva
------------------------------------------	-----------------	---------

La técnica utilizada en este proceso es caja negra basada en las interfaces y las especificaciones de los módulos. Esta permite detectar el funcionamiento incorrecto e incompleto, errores de interfaces, problemas de rendimiento y errores de inicio y terminación.

6.4 Casos de Prueba.

Permitir actualizar los datos de la tarjeta de registro de la persona

Permitir certificar los datos actualizados de la tarjeta de registro

6.5 Criterios de Término

Negocio

- En la plantilla Modelo de Casos de Uso del Negocio del proyecto Sistema para la Planificación Material y Financiera, presenta en su diagrama CU Negocio estereotipos que no corresponden al Negocio, y no se define correctamente en el diagrama de actividades el flujo de dichas actividades.

Requisito

- La plantilla de Gestión de Requisitos fue entregada incompleta, no se definieron sus acápites y especificaciones.
- En la plantilla de Especificación de Requisito su descripción general es muy breve, no se describen todos los requisitos funcionales, no se especifican los recursos tecnológicos en el requisito no funcional de Rendimiento, no se especifican los protocolos de interfaz, no se especifican las entradas y salidas del sistema, los requisitos funcionales no detallan claramente como deben comportarse el sistema, y los requisitos no especifican lo que el sistema no debe hacer.
- En la plantilla de Modelo de Casos de Uso del Sistema en el acápite especificación de los casos de uso del sistema no se especifican las pos condiciones y no se define la prioridad.

Análisis y Diseño

- En la plantilla del Modelo de Análisis no se especifican los diagramas de interacción y las plantillas se encuentran divididas por casos de uso.
- En la plantilla de Modelo de Diseño los diagramas no se muestran de manera legible.

- En el Documento de Arquitectura de Software no se describen las características de dimensiones y no se especifican correctamente los casos de usos arquitectónicamente significativos.

Implementación

- En la plantilla del Modelo de Despliegue no se encuentran especificados los subsistemas y los componentes de implementación.

Manuales

- En la plantilla del Manual de Usuario no se especifica la versión del documento, no posee la versión del producto, la caratula no incluye la versión de hardware, el sistema operativo que presenta, no incluye todas las porciones que presenta el documento, esta debe tratar de englobar todo el contenido, no incluye las instrucciones de como desinstalar el software, no se incluyen instrucciones de logueo, en este documento no se explica cómo interrumpir y deshacer operación durante la ejecución de instrucciones y cómo volverla a reiniciar, no presenta un listado de palabras reservadas, la documentación no se ocupa de todos los problemas conocidos en usar el software, no presenta glosario de término, no presentan precauciones o advertencias de la aplicación, no presentan precauciones o advertencias de la aplicación y en el documento no existe advertencias, precauciones y nota.
- En la plantilla del Manual de Instalación existen palabras con difícil comprensión para usuarios que no presenten conocimiento de informática.

Funcionalidad

- En las pruebas funcionales aplicadas al sistema se comprobó que no se especifican los campos obligatorios y la ayuda se muestra en ventanas diferente a la ventana padre.

Interfaz de Usuario y Atributos de Calidad

- En la plantilla Atributos de Calidad se pudo determinar que el sistema no es adaptable a otros navegadores que no sean los que se definieron para la ejecución del sistema, existen algunos botones que no se describe la funcionalidad.
- En la plantilla Interfaz de Usuario se pudo determinar que se muestra algunas abreviaturas que no define su significado y la ayuda abre una nueva ventana pudiendo ser mostrada en la ventana padre.

Cantidad de Conformidades	No	Grado de Influencia de la Calidad
0 - 2		Alto
3 - 6		Medio
> 6		Bajo

	Alto	Medio	Bajo
Modelo de Casos de Uso del Negocio			X
Modelo de Casos de Uso del Sistema	X		
Plantilla de Especificación de Requisitos			X
Plantilla del Modelo de Análisis		X	
Plantilla del Modelo de Diseño	X		
Documento de Arquitectura	X		
Plantilla del Modelo de Despliegue	X		
Manual de Usuario			X
Manual de Instalación	X		
Plantilla de Interfaz de Usuario		X	
Plantilla de Atributos de Calidad	X		

	Alto	Medio	Bajo
Modelo de Casos de Uso del Negocio			X
Modelo de Casos de Uso del Sistema	X		
Plantilla de Especificación de Requisitos			X
Plantilla del Modelo de Análisis		X	
Plantilla del Modelo de Diseño	X		
Documento de Arquitectura	X		
Plantilla del Modelo de Despliegue	X		
Manual de Usuario			X
Manual de Instalación	X		
Plantilla de Interfaz de Usuario		X	
Plantilla de Atributos de Calidad	X		

Alto: El documento presente un grado alto de calidad.

Medio: El documento presente un grado medio de calidad.

Bajo: El documento presente un grado bajo de calidad.

6.6 Herramientas

Al aplicar las pruebas al sistema Reserva no se utiliza ningún instrumento o herramienta ya que solamente le aplicamos las pruebas de Caja Negra (Interfaz de Usuario y Atributos de Calidad). También le aplicamos las pruebas a los documentos del sistema expuestos en el Expediente de Proyecto, estas se realizan a través de Listas de Chequeos.

7. Documentación de los Resultados.

Caso de Prueba	Auditoría
Fecha	16/04/08
Resultado	
<ul style="list-style-type: none"> • Los campos obligatorios no están referenciados • No se envían mensajes de error cuando se deja algún campo sin llenar 	
Observaciones	
Al probar en el sistema este Caso de Uso se detectaron errores Simples	
Responsables de las Pruebas	Yudisleidis Baez Ortiz Annalie Pacheco Carrio

Al aplicar las Listas de Chequeo a cada documento de los módulos correspondientes, se pudo llegar a la conclusión de que el trabajo desempeñado por los probadores se ha realizado correctamente, ya que se encontraron disímiles no conformidades en cada aspecto examinado.

Anexo # 3: Plan de Prueba de Módulo.

UCID

Especificación de Requisitos

Proyecto	Módulo	Etapa	Versión	Fecha
<nombre del proyecto>	<nombre del módulo>	<etapa en que se realiza>	<1.0>	<00/00/00>

Control de Versiones:

Fecha	Versión	Descripción	Autor
<00/00/00>	<1.0>	<>	<nombre del que aplicó la lista>

Claridad

No.	Evaluación	Eval.	NP.	Comentario
1	¿Están descritos los requisitos funcionales en lenguaje natural?			
2	¿La estructura y formato del documento facilitan la comprensión?			
3	¿Cada requerimiento admite tan solo una única interpretación?			
4	¿Cada característica del producto final está descrita utilizando una terminología única?			
5	¿Hay un glosario en el que se define el significado específico de cada término?			
6	¿Los requerimientos podrían ser comprendidos, implementados y verificados por un grupo independiente?			
7	¿Se especifican solo una vez cada uno de los requisitos?			
8	¿No se definen similares características en diferentes requisitos?			
9	¿Se describen correctamente cada una de las especificidades de descripción de los requisitos: Precedencia, Seguimiento, estado (propuesto, aprobado), prioridad, nivel de riesgo y atributos del requerimiento?			
10	¿Se describen correctamente los requisitos No Funcionales?			
11	¿Se describen los requerimientos de Usabilidad?			
12	¿Se especifican las acciones necesarias para que los			

	usuarios hagan operaciones particulares?			
13	¿Se especifican los requerimientos de conformidad con normas y estándares internacionales?			
14	¿Se describen correctamente los requerimientos de usabilidad?			
15	¿Se especifican los requerimientos de Confiabilidad ?			
16	¿Se establecen las características que debe cumplir el sistema para brindar una mayor confiabilidad al usuario?			
17	¿Se describe (Correctamente) algún requisito de Usabilidad?			
18	¿Se especifican los requerimientos de Rendimiento ?			
19	¿Se especifican las características de rendimiento?			
20	¿Se especifican los Recursos tecnológicos requeridos?			
21	¿Se especifica el Tiempo de respuesta en una transacción o traspaso de información (promedio y máximo)?			
22	¿Se especifica la capacidad de conexión (por ejemplo cantidad usuarios permitidos)?			
23	¿Se especifican los requerimientos de Soporte ?			
24	¿Se especifican los requerimientos que soportan y mantienen el proyecto? (Se incluyen estándares, convenciones, normas, etc.)			
25	¿Se describe (Correctamente) algún requisito de Soporte?			
26	¿Se especifican los requerimientos de Restricciones de diseño ?			
27	¿Se especifican todas las restricciones de diseño para el desarrollo del proyecto?			
28	¿Se describe (Correctamente) algún requisito de Diseño?			
29	¿Se describen todos los Requisitos de Ayuda y Documentación que requiera el proyecto?			

30	¿Se especifican los requerimientos de Adquisición de Componentes ?			
31	¿Se incluyen todos los componentes que son necesarios adquirir para la ejecución del proyecto?			
32	¿Se especifican los requerimientos de Interfaz ?			
33	¿Se especifican los protocolos de comunicación y tipos de interfaces?			

Complejidad Externa

No.	Evaluación	Eval.	NP.	Comentario
34	Están incluidos todos los requerimientos relacionados con:			
35	➤ Funcionalidad			
35	➤ Contexto de Uso			
37	➤ Software a utilizar			
38	➤ Comunicación			
39	➤ Hardware			
40	➤ Entradas			
41	➤ Salidas			
42	➤ Informes			
44	➤ Seguridad			
45	➤ Portabilidad			
46	➤ Permanencia de los datos			
47	➤ Mantenibilidad			
48	➤ Instalación			
49	➤ Disponibilidad			
37	➤ Recuperación			
38	➤ Seguridad			
39	➤ Legales			
40	➤ Aplicación de estándares			
41	➤ Integridad.			
42	➤ Fiabilidad			

43	¿Están especificados los cambios posibles a los requerimientos?			
44	¿La probabilidad de cambios está especificada para cada requerimiento?			

Consistencia Interna

No.	Evaluación	Eval.	NP.	Comentario
45	No hay dos requerimientos en conflicto debido a que describen:			
46	➤ El mismo objeto con distinta terminología			
47	➤ El mismo objeto como si tuviera distintas características			
48	➤ Dos o más acciones incompatibles lógicamente			
49	➤ Dos o más acciones incompatibles en el tiempo			

Consistencia Externa

No.	Evaluación	Eval.	NP.	Comentario
50	Los requerimientos son consistentes con:			
51	➤ El contexto de negocio y las justificaciones del proyecto			
52	➤ Supuestos y dependencias para el sistema			

Trazabilidad

No.	Evaluación	Eval.	NP.	Comentario
53	Cada requerimiento:			
54	➤ Obedece a una necesidad específica de usuario			
56	➤ Tiene su origen en una fuente (documento o persona) específica			
57	➤ Se puede rastrear hacia delante su incorporación al diseño			
58	➤ Se puede rastrear hacia delante su incorporación en determinados módulos			

Verificabilidad

No.	Evaluación	Eval.	NP.	Comentario
59	¿Cada requerimiento es implementable?			
60	¿Para cada requerimiento existe un procedimiento que, ejecutado por una persona o máquina, permite verificar si se cumple?			
61	¿Existe algún requerimiento que se va a expresar en términos verificables más adelante?			
62	¿Cada requerimiento es relevante para el problema y su solución?			

Modificabilidad

No.	Evaluación	Eval.	NP.	Comentario
63	¿El documento de requerimientos está organizado de forma clara y lógica?			
64	¿La estructura adhiere a algún estándar?			
65	¿La redundancia es mínima y sólo se debe a distintos niveles de abstracción o detalle?			
66	¿Están especificados los cambios posibles a los requerimientos?			

Priorización

No.	Evaluación	Eval.	NP.	Comentario
67	¿Cada requerimiento tiene establecido un nivel de prioridad asignado por el Cliente/Usuario para guiar las negociaciones/compromisos?			

Contenido General

No.	Evaluación	Eval.	NP.	Comentario
68	¿Cada requerimiento es relevante para el problema y su solución?			
69	Ninguno de los requerimientos realmente corresponde a:			
70	➤ detalles de diseño			

71	➤ detalles de verificación			
72	➤ detalles de gestión del proyecto			
73	Hay una sección de:			
74	➤ Introducción” que describe el documento y sus destinatarios			
75	➤ Alcance” que se describe lo que cubre el documento			
76	➤ Definiciones, acrónimos y abreviaturas			
77	➤ Propósito del producto.			
78	➤ Referencias			
79	➤ Descripción general			
	➤ Requerimientos específicos			
80	Están presentes:			
81	➤ Todos los anexos necesarios			
82	➤ Todas las figuras, tablas y diagramas necesarias			

Requerimientos específicos

No.	Evaluación	Eval.	NP.	Comentario
83	En todas las funciones está especificada:			
84	➤ Todos los aspectos del procesamiento no exitoso			
85	➤ Todas las entradas			
86	➤ Todas las salidas			
87	➤ Todos los requerimientos de desempeño			
88	¿Para cada una de las interfaces externas están especificados los detalles de la comunicación?			
89	¿Se especificaron los requisitos funcionales como capacidades o condiciones que el sistema debe cumplir?			
90	¿Los requisitos funcionales especifican la manera en que el sistema debe reaccionar a determinadas entradas?			

91	¿Los requisitos funcionales especifican cómo debe comportarse el sistema en situaciones particulares?			
92	¿Los requisitos funcionales declaran explícitamente lo que el sistema no debe hacer?			
93	¿Los requerimientos funcionales escogen la descripción de todos los servicios esperados por el usuario?			
94	¿Se especificaron los requisitos no funcionales como propiedades o cualidades que el producto debe tener?			
95	¿Los requisitos no funcionales son aplicables al sistema en su totalidad?			
96	¿Se definen las características que debe tener la interfaz del sistema?			
97	¿Se especifican las características que debe cumplir el sistema para ser fácil de utilizar por el cliente/usuario? (requisitos de usabilidad)			
98	¿Se establecen las propiedades que debe tener el sistema para que sea compatible con varios sistemas operativos? (requisitos de portabilidad)			
99	¿Se declaran las características que debe poseer el sistema para que esté disponible a cada uno de los usuarios que lo necesiten? (requisitos de disponibilidad)			
100	¿Se exponen las características que debe tener el sistema para permitir la integridad de la información almacenada? (requisitos de integridad)			
101	¿Se definen las características que debe tener el sistema para lograr su correcto funcionamiento? (requisitos de funcionalidad)			
102	¿Se establecen las restricciones del diseño a tener en cuenta?			
103	¿Se establecen las restricciones de la implementación (estándares requeridos, normas de codificación, lenguajes de programación, políticas para la integridad			

	de la base de datos, limitaciones de recurso y entornos operativos)?			
104	¿Los requisitos funcionales definen propiedades y restricciones del sistema: tiempos de respuesta, requisitos de almacenamiento?			
105	¿Todos los requisitos están declarados en forma de acción (deben terminar en ar, er, ir)?			

Notas:

<Insertar nota si es necesario>

Leyenda:

Eval. : Evaluación que se le otorga al aspecto a evaluar: [0, 2, 4,5,].

NP. : No procede (se marca con una **X** la pregunta que no será necesaria realizarla en el sistema que se le esta aplicando la lista de chequeo).

Comentario: Es obligatorio en las evaluaciones distintas de 5 puntos.

Las evaluaciones serán:

- - **Malo:** propiedad parcialmente disponible. (0)
- **Satisfactorio:** propiedad parcialmente disponible. (2)
- + **Bien:** propiedad parcialmente disponible. (4)
- ++ **Excelente:** propiedad muy bien implementada. (5)

Criterio de evaluación:

- () Sin modificaciones.
- () Pequeñas modificaciones.
- () Grandes modificaciones.
- () Nueva elaboración.

Conclusión

	Aceptado
	Diferido
	No Aceptado

Anexo # 4: Diseño de Casos de Prueba.

UCID

Diseño de Casos de Prueba		
Proyecto	Módulo	Versión
Comités Militar	Reserva	1.0
Caso de Uso	Versión CP	
Auditoria	1.0	

Control de Versiones:

Fecha	Versión	Descripción	Autor
16/04/08	1.0	Aplicación del Caso de Prueba al Proyecto Comités Militar, Módulo Reserva.	Yudisleidis Baez Ortiz Annalie Pacheco Carrió.

Condiciones de Ejecución:

Los campos usuario y sucesos se deben estar llenos para que se genere la tabla de Auditoria del Sistema.

1. Secciones a probar en el Caso de Uso

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: Generar.	EC 1.1: Generar	Dado nombre y suceso genera tabla de Auditoria.	<ol style="list-style-type: none"> Se selecciona el menú Administrador y el submenú Auditoria. Se llenan los campos (nombre, sucesos,

			<p>sucesos ocurridos y configuración de impresión).</p> <p>3. Se da clic en el botón generar y se muestra la tabla con los datos de Auditoria.</p>
	EC 1.2: Terminar	Se termina el proceso.	<p>1. Se selecciona el menú Administrador y el submenú Auditoria.</p> <p>2. Se llenan los campos (nombre, sucesos, sucesos ocurridos y configuración de impresión).</p> <p>3. Se decide terminar el proceso dando clic en el botón terminar.</p>

SC 1: <Sección #1 a revisar>

Id del escenario	Escenario	Nombre	Suceso	Sucesos Ocurridos (Desde)	Sucesos Ocurridos (Hasta)	Conf. Imp. (Registro de hojas)	Respuesta del Sistema	Resultado de la Prueba
EC 1	Generar	Yoel Alfonso.	Todos	03/06/08	04/06/08	20	Que se muestren las tablas correso	Se abre una nueva ventana con las tablas
		Yoel Alfonso.	Salida del Sistema	03/03/08	04/06/08	18		

		Yoel Alfonso.	Entrada al Sistema	03/03/08	06/04/08	1	ndientes de los campos llenados.	correspondientes a las auditorias realizadas con respecto a los campos llenados.
EC 1	Generar	Yoel Alfonso	Todos	-	04/06/08	20	Que el sistema envíe un mensaje pidiendo los campos que son obligatorios para que se muestre un resultado satisfactorio.	Se abre una ventana que muestra un mensaje "No se encontraron sucesos para JEFE POTENCIAL HUMANO SM ALQUIZAR Yoel(jpotsm) Alfonso Crespo."
		Yoel Alfonso	Salida del Sistema	03/03/08	-	18		
		Yoel Alfonso	Entrada al Sistema	-	-	1		
		Yoel Alfonso.	Todos	03/06/08	04/06/08	-		Cuando el campo de Registro de hojas se deja vacio toma por defecto (1).

								No muestra error.
EC 1	Generar	Yoel Alfonso.	Todos	03/06/08	04/06/08	20	El sistema muestra un mensaje informativo.	Se escribe el resultado que se obtiene al realizar la prueba.
		Yoel Alfonso	Baja de un Registro	03/03/08	04/06/08	18		
		Yoel Alfonso	Todos	01/07/08	04/06/08	20		
		Yoel Alfonso	Todos	03/06/08	04/03/08	20		
EC 2	Terminar	Yoel Alfonso.	Todos	03/06/08	04/06/08	20	Muestra un mensaje de si se esta seguro de abandonar y las opciones de Aceptar y Cancelar	Si se escoge el botón aceptar de inmediato se abandona y muestra la interfaz inicial. Si se escoge el botón cancelar se cancela la opción Terminar.
		Yoel Alfonso.	Salida del Sistema	03/03/08	04/06/08	18		
		Yoel Alfonso.	Entrada al Sistema	03/03/08	06/04/08	1		
EC 2	Terminar	Yoel Alfonso	Todos	-	04/06/08	20		
		Yoel Alfonso	Salida del Sistema	03/03/08	-	18		
		Yoel Alfonso	Entrada al Sistema	-	-	1		
		Yoel Alfonso.	Todos	03/06/08	04/06/08	-		
EC 2	Terminar	Yoel Alfonso.	Todos	03/06/08	04/06/08	20		
		Yoel	Baja de	03/03/08	04/06/08	18		

		Alfonso	un Registro						
		Yoel Alfonso	Todos	01/07/08	04/06/08	20			
		Yoel Alfonso	Todos	03/06/08	04/03/08	20			

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
Caso de Uso Auditoria del Módulo Reserva Proyecto	1.	Los campos obligatorios no están referenciados	Administración/Auditoria.	Prueba	X		Señalar los campos obligatorios.	Pendiente.	
Comités Militares	2.	No se envían mensajes de error cuando se deja algún campo sin llenar	Administración/Auditoria.	Prueba	X			Pendiente.	

GLOSARIO DE TÉRMINOS

Términos:

- ¹ **Plan de Pruebas:** Es una guía que describe las estrategias, los recursos y la planificación de las pruebas, es el principal factor del éxito para la puesta en práctica de un proceso de pruebas que permita entregar un software de mejor nivel.
- ² **Estrategia:** Arte de dirigir operaciones, habilidad para dirigir.
- ³ **Procedimiento:** Modo de obrar, modo de proceder.
- ⁴ **Proceso:** Acción de ir hacia adelante, conjunto de las fases sucesivas de un fenómeno natural o de una operación artificial.
- ⁵ **Tendencia:** Propensión o inclinación en las cosas hacia determinados fines.
- ⁶ **Estándares:** Tipos, modelos, patrones, niveles o referencias.
- ⁷ **Acarrea:** Acción de acarrear. **Acarrear:** Ocasionar, producir, traer consigo daños o desgracias.
- ⁸ **Exhaustivo:** Que agota o apura por completo.
- ⁹ **Idónea:** Adecuado y apropiado para algo.
- ¹⁰ **Depuración:** Acción y efecto de depurar. **Depurar:** limpiar, purificar.
- ¹¹ **Iteración:** Acción y efecto de iterar. **Iterar:** repetir.
- ¹² **Validación:** Acción y efecto de validar. **Validar:** dar fuerza o firmeza a algo, hacerlo válido.
- ¹³ **Automatizable: Automatizar:** convertir ciertos movimientos corporales en movimientos automáticos o indeliberados. Aplicar la automática a un proceso, a un dispositivo, etc.
- ¹⁴ **Estructura:** Distribución y orden con que está compuesta una obra.
- ¹⁵ **Anomalías:** Irregularidad, irregular, extraño.
- ¹⁶ **Vulnerabilidad:** Cualidad de vulnerable. **Vulnerable:** Que puede ser herido o recibir lesión, física o moralmente.