

Universidad de las Ciencias Informáticas



# **Estrategia de prueba para el producto Heredia**

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

## **AUTOR**

Jorge Antonio Gómez Colombat

## **TUTORA**

Lic. Liana Isabel Araujo Pérez

Ciudad de La Habana  
Julio 2007  
“Año 49 del triunfo de la Revolución”

*“Algunas enfermedades, como dicen los médicos, en su comienzo son fáciles de curar pero difíciles de reconocer... pero con el paso del tiempo, cuando no se han reconocido, ni tratado al principio, se vuelven fáciles de reconocer pero difíciles de curar.”*

*Niccolo Machiavelli*

## DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Firma del Autor

Jorge Antonio Gómez Colombat

---

Firma de la Tutora

Lic. Liana Isabel Araujo Pérez

## DATOS DE CONTACTO

### Tutora:

Nombre: Liana Isabel Araujo Pérez

Grado: Licenciada en Ciencias Informáticas

Correo Electrónico: [lianaisabel@uci.cu](mailto:lianaisabel@uci.cu)

Ubicación: Universidad de la Ciencias Informáticas.

## **AGRADECIMIENTOS**

Agradezco inicialmente a la Revolución Cubana y en especial al comandante, por haberme dado la posibilidad de formarme como lo que soy, y de realizar mis sueños.

A mi mamá, por haberme dado siempre el mejor ejemplo durante tantos años, por estar ahí siempre para mí.

A mi abuela porque ha sido como una segunda madre para mí.

Al resto de mi familia, por apoyarme y darme ánimo siempre que lo necesite, en especial a mi papá, a Gilberto, a Andrés, a Rogelio, a Heredia y todas mis tías y primos en general.

A mis hermanas, por quererme.

A mi tutora y a los colaboradores de mi tesis, por su paciencia y los conocimientos que me han transmitido.

A Heidy Valenzuela, por su cariño y apoyo incondicional en todo momento y situación.

A Carmen por siempre darme fuerzas para seguir adelante.

A Roberkis Terrero, Hugo Vargas, Luis Lamela, Alicia Fernández, Isel Batista, Dailiany Soler y otros que me ayudaron en gran manera.

A todas las personas que he conocido, pues de una manera u otra han hecho de mí lo que soy, muchísimas gracias.

## DEDICATORIA

*...A mi mamá*

## RESUMEN

Este trabajo surge con motivo de probar eficaz y eficientemente un libro electrónico dedicado al poeta José María Heredia. Para ello se realiza un estudio del estado del arte de las pruebas del software, se presentan conceptos, se ejemplifican herramientas, normas y procesos relacionados con las pruebas y la tecnología multimedia, y se describe brevemente la aplicación a probar. Luego de un análisis de los elementos mencionados, se define una estrategia de prueba a seguir para organizar y documentar las pruebas aplicadas al producto “Heredia” y los resultados arrojados por las mismas, facilitando la realización de un resumen, análisis y evaluación del costo del proceso desarrollado. En consecuencia se obtuvo un plan de pruebas y una colección de los resultados de la ejecución de los casos de prueba de dicho plan, permitiendo arribar a la conclusión de que el producto “Heredia” no está apto para ser entregado al cliente, debido a la falta de calidad del mismo.

Para la visualización y estudio de la aplicación a probar se utilizó la herramienta Flash 8 y Flash Player 9.

# ÍNDICE

<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>FUNDAMENTACIÓN TEÓRICA.....</b>	<b>4</b>
1.1 Introducción.....	4
1.2 Pruebas. Conceptos. Costo .....	4
1.2.1 Casos de prueba .....	6
1.2.2 Plan de Pruebas .....	6
1.2.3 Estrategia de pruebas.....	8
1.2.4 Costos del proceso de prueba .....	9
1.3 Niveles y tipos de prueba .....	11
1.3.1 Prueba de Caja Blanca.....	12
1.3.1.1 Prueba de camino básico.....	12
1.3.1.2 Prueba de condición .....	12
1.3.1.3 Prueba de flujo de datos .....	13
1.3.1.4 Prueba de bucles .....	13
1.3.2 Prueba de Caja Negra .....	13
1.3.2.1 Pruebas basadas en grafos .....	14
1.3.2.2 Partición equivalente.....	14
1.3.2.3 Análisis de valores límite.....	14
1.3.3 Pruebas de Interfaces Gráficas de Usuario (IGUs).....	15
1.3.4 Pruebas de la documentación y facilidades de ayuda .....	16
1.3.5 Pruebas unitarias.....	17
1.3.6 Pruebas de integración.....	18
1.3.7 Pruebas de validación .....	18
1.3.8 Pruebas de usabilidad .....	18
1.3.9 Pruebas de aceptación .....	19
1.3.10 Pruebas del sistema .....	19
1.3.10.1 Pruebas de seguridad .....	19
1.3.10.2 Pruebas de volumen .....	20
1.3.10.3 Pruebas de integridad .....	20
1.3.10.4 Pruebas de recuperación .....	20
1.3.10.5 Pruebas de stress .....	20
1.3.10.6 Pruebas de Benchmark.....	20
1.3.10.7 Pruebas de contención.....	20
1.3.10.8 Pruebas de carga.....	21
1.3.10.9 Pruebas de perfil de desempeño .....	21
1.3.10.10 Pruebas de configuración.....	21
1.3.10.11 Pruebas de instalación .....	21
1.3.11 Pruebas de regresión.....	21



1.4	Procesos de pruebas estandarizados .....	21
1.5	Propuestas de generación de casos de prueba .....	24
1.6	Herramientas de apoyo a las pruebas.....	25
1.7	Procesos de desarrollo de aplicaciones hipermedia .....	27
1.7.1	Método de Desarrollo de Aplicaciones Educativas Hipermedia .....	27
1.7.2	Elaboración de Materiales Formativos Multimedia. Criterios de Calidad .....	27
1.8	Producto Heredia .....	28
1.9	Conclusiones.....	29
<b>PLAN DE PRUEBAS.....</b>		<b>30</b>
2.1	Introducción.....	30
2.2	Plan de Pruebas Global del producto Heredia .....	31
2.2.1	Información del plan .....	31
2.2.3	Ítems a probar.....	31
2.2.4	Estrategia .....	31
2.2.5	Categorización de la configuración.....	32
2.2.6	Tangibles.....	32
2.2.7	Procedimientos especiales .....	33
2.2.8	Recursos .....	33
2.2.9	Calendario .....	34
2.2.10	Responsable.....	34
2.3	Plan de Pruebas a la Documentación Técnica.....	34
2.3.1	Prueba de Revisión e Inspección a la Documentación 1.01 .....	35
2.3.1.1	Caso de prueba RID 01 .....	35
2.3.1.2	Caso de prueba RID 02 .....	35
2.3.1.3	Caso de prueba RID 03 .....	36
2.3.1.4	Caso de prueba RID 04 .....	36
2.3.1.5	Caso de prueba RID 05 .....	37
2.3.1.6	Caso de prueba RID 06 .....	37
2.4	Plan de Pruebas de Validación.....	38
2.4.1	Prueba de validación 1.01 .....	38
2.4.1.1	Caso de prueba ATPD 01 .....	38
2.4.2	Prueba de validación 2.01 .....	39
2.4.2.1	Caso de prueba ANM 01 .....	39
2.4.2.2	Caso de prueba AC 01 .....	40
2.4.2.3	Caso de prueba ANM 02.....	40
2.4.2.4	Caso de prueba ANM 03.....	41
2.4.2.5	Caso de prueba ANM 04.....	41
2.4.2.6	Caso de prueba ANM 05.....	42
2.4.2.7	Caso de prueba AC 02.....	42
2.4.2.8	Caso de prueba AC 03.....	43
2.4.2.9	Caso de prueba AC 04.....	44

2.4.2.10	Caso de prueba ANM 06.....	44
2.4.2.11	Caso de prueba ANM 07.....	45
2.4.2.12	Caso de prueba ANM 08.....	45
2.4.2.13	Caso de prueba ANM 09.....	46
2.4.2.14	Caso de prueba AC 05.....	46
2.4.2.15	Caso de prueba AC 06.....	47
2.4.2.16	Caso de prueba AC 07.....	48
2.4.2.17	Caso de prueba ANM 10.....	48
2.4.2.18	Caso de prueba ANM 11.....	49
2.4.2.19	Caso de prueba AC 08.....	49
2.4.2.20	Caso de prueba AC 09.....	50
2.4.2.21	Caso de prueba AC 10.....	51
2.4.2.22	Caso de prueba ANM 12.....	51
2.4.2.23	Caso de prueba ANM 13.....	52
2.4.2.24	Caso de prueba AC 11.....	52
2.4.2.25	Caso de prueba AC 12.....	53
2.4.2.26	Caso de prueba AC 13.....	54
2.4.2.27	Caso de prueba ANM 14.....	54
2.4.2.28	Caso de prueba ANM 15.....	55
2.4.2.29	Caso de prueba AC 14.....	55
2.4.2.30	Caso de prueba AC 15.....	56
2.4.2.31	Caso de prueba AC 16.....	56
2.4.2.33	Caso de prueba AC 18.....	57
2.4.2.34	Caso de prueba AC 19.....	58
2.4.2.35	Caso de prueba AC 20.....	58
2.4.3	Prueba de validación 3.01 .....	59
2.4.3.1	Caso de prueba PS 01 .....	59
2.4.3.2	Caso de prueba PS 02.....	60
2.4.3.3	Caso de prueba PS-03.....	60
2.4.3.4	Caso de prueba PS 04.....	61
2.4.3.5	Caso de prueba PS-05.....	61
2.4.3.6	Caso de prueba PS 06.....	62
2.4.3.7	Caso de prueba PS-07.....	62
2.4.3.8	Caso de prueba PS 08.....	63
2.4.3.9	Caso de prueba PS-09.....	63
2.4.3.10	Caso de prueba PS-10.....	64
2.4.3.11	Caso de prueba PS-11.....	64
2.4.3.12	Caso de prueba PS 12.....	65
2.4.3.13	Caso de prueba PS-13.....	65
2.4.3.14	Caso de prueba PS 14.....	66
2.4.3.15	Caso de prueba PS-15.....	66

2.4.3.16 Caso de prueba PS-16.....	67
2.4.3.17 Caso de prueba PS-17.....	67
2.4.3.18 Caso de prueba PS-18.....	68
2.5 Plan de Pruebas de Usabilidad.....	68
2.5.1 Prueba de Usabilidad 1.01.....	69
2.5.1.1 Caso de prueba CIU 01.....	69
2.5.2 Prueba de Usabilidad 2.01.....	70
2.5.2.1 Caso de prueba CC- 01.....	70
2.5.2.2 Caso de prueba CC- 02.....	70
2.5.2.3 Caso de prueba CC- 03.....	71
2.5.2.4 Caso de prueba CC- 04.....	71
2.5.2.5 Caso de prueba CC- 05.....	72
2.5.2.6 Caso de prueba CC- 06.....	72
2.5.2.7 Caso de prueba CC- 07.....	73
2.5.2.8 Caso de prueba CC- 08.....	73
2.5.2.9 Caso de prueba CC- 09.....	73
2.5.2.10 Caso de prueba CC- 10.....	74
2.5.2.11 Caso de prueba CC- 11.....	74
2.5.2.12 Caso de prueba CC- 12.....	75
2.5.2.13 Caso de prueba CC- 13.....	75
2.5.2.14 Caso de prueba CC- 14.....	76
2.5.2.15 Caso de prueba CC- 15.....	76
2.5.3 Prueba de Usabilidad 3.01.....	76
2.5.3.1 Caso de prueba IGU- 01.....	77
2.5.3.2 Caso de prueba IGU- 02.....	77
2.6 Conclusiones.....	78
<b>RESULTADOS DE LAS PRUEBAS.....</b>	<b>79</b>
3.1 Introducción.....	79
3.2 Resultados de las pruebas.....	79
3.3 Resumen de las pruebas.....	89
3.4 Costo del trabajo.....	98
3.5 Conclusiones.....	100
<b>CONCLUSIONES GENERALES.....</b>	<b>101</b>
<b>RECOMENDACIONES.....</b>	<b>102</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>103</b>
<b>BIBLIOGRAFÍA.....</b>	<b>105</b>
<b>GLOSARIO DE TÉRMINOS.....</b>	<b>107</b>

## INTRODUCCIÓN

### **Actualidad y necesidad del trabajo**

El desarrollo de productos hipermedia con fines educativos e informativos ha crecido notablemente en la actualidad debido a las ventajas que brindan en el proceso de formación. Estos productos requieren de varios recursos, tanto materiales como humanos para su correcta elaboración. En el caso especial de los libros electrónicos, son escasos los trabajos que se pueden hallar en la red, que describan como realizar de modo estándar este tipo de aplicaciones. La Organización Internacional de Estándares (ISO) produjo la Norma ISO 9000-3, que provee, tanto a los desarrolladores como a los clientes, de un conjunto de guías para el aseguramiento de la calidad de los procesos de desarrollo del software de modo general.

Varios intelectuales y grupos de trabajo han estado cooperando en el desarrollo de metodologías y procedimientos para construir productos hipermedia con una determinada calidad, sin embargo todavía no se ha logrado una estandarización de los mismos. Existen algunas metodologías como RUP, MSF y XP que guían el proceso de desarrollo de software de modo general, siendo muy abarcadoras y extensas para libros electrónicos como el caso del producto “Heredia”.

Como no se ha logrado un estándar y los libros electrónicos no presentan un negocio tan complejo como otros tipos de software, se ha ignorado un poco el hecho de apoyarse en un conjunto de procedimientos, para lograr la calidad requerida en aplicaciones de este corte; lo que trae consigo resultados indeseados tanto para los desarrolladores, como para los clientes y usuarios finales, y por consiguiente la reiterada aplicación de pruebas al sistema.

### **Situación Problémica:**

La Universidad de Ciencias Informáticas (UCI) ha desarrollado y continúa desarrollando algunos productos con tecnología multimedia, fundamentalmente de corte educativo e informativo. En la elaboración de estos sistemas, no se ponen en práctica o no se cumplen en su totalidad procedimientos probados que organicen, documenten y faciliten el proceso de desarrollo, lo que provoca reiteradas revisiones, sin lograr los resultados deseados ya que se siguen liberando

productos sin la calidad requerida.

### **Problema científico:**

¿Cómo organizar y desarrollar eficientemente el proceso de prueba para el producto “Heredia”?

### **Objeto de estudio:**

Pruebas al software

### **Campo de acción:**

Pruebas al producto “Heredia”

### **Objetivo general:**

Desarrollar una estrategia de prueba para el producto “Heredia”.

### **Objetivos específicos:**

- Definir qué tipos de pruebas se le realizarán al producto “Heredia”.
- Realizar las pruebas al producto “Heredia”.
- Analizar los resultados arrojados por las pruebas al producto “Heredia”.

### **Tareas de investigación:**

Para lograr el primer objetivo, se deben cumplir las siguientes tareas:

- Estudiar sobre las pruebas del software
- Estudiar algunos procesos de desarrollo de software hipermedial
- Profundizar sobre las pruebas que se realizan con mayor frecuencia a las aplicaciones hipermedia
- Realizar un estudio del estado deseado y actual del producto “Heredia”

Para alcanzar el segundo objetivo, se deben concretar las siguientes acciones:

- Analizar la documentación del producto “Heredia”
- Definir una estrategia de prueba
- Planificar las pruebas

- Diseñar los casos de prueba
- Ejecutar los casos de prueba

Para alcanzar el tercer objetivo, se deben realizar las siguientes tareas:

- Investigar sobre la forma de recolección y organización de los resultados de prueba
- Recolectar los resultados de las pruebas y las inconformidades

### **Aportes prácticos esperados del trabajo**

El trabajo constituye una propuesta de cómo organizar y desarrollar el proceso de prueba de un libro electrónico. La cual puede servir de referencia y estudio a probadores, estudiosos e interesados en el tema de las pruebas.

### **Estructura de los contenidos**

La información de este trabajo está distribuida en tres capítulos.

**Capítulo 1 Fundamentación teórica:** Contiene conceptos relacionados con las pruebas, como caso, estrategia, plan, costo, tipos y niveles de prueba. Se hace mención de algunos procesos de pruebas estandarizados, herramientas para la automatización de las pruebas, propuesta de generación de casos de prueba. Se analiza brevemente el estado del arte de las pruebas, se presentan conceptos relacionados con la aplicación, métodos de elaboración de productos hipermedia, y una descripción del sistema.

**Capítulo 2 Plan de pruebas:** En este se detalla la solución al problema, que sería la utilización de un plan de pruebas, en el que se define una estrategia de prueba a seguir y está compuesto por tres subplanes definidos por los tipos de prueba albergan.

**Capítulo 3 Resultado de las pruebas:** Este capítulo contiene una tabla con los resultados e inconformidades de los casos de pruebas ejecutados, un análisis de dichos resultados, y una evaluación del proceso de prueba.

Capítulo  
1

## FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción

El aseguramiento de la calidad de software se ha convertido en una necesidad prioritaria para las organizaciones que desarrollan este tipo de producto, ya sea para uso interno o para implementaciones externas en clientes, porque cada vez más los errores en el software repercuten directa o indirectamente en consecuencias indeseables para la organización. Un error que desde el punto de vista de codificación puede ser relativamente simple de corregir puede resultar muy difícil y costoso de detectar y llegar a tener graves efectos en la organización. Todo ello, sin mencionar los costos políticos para las gerencias de sistemas, normalmente responsables de la dirección de los proyectos de software.

Generalmente el equipo de desarrollo del proyecto se encuentra presionado por la necesidad de cumplir con las fechas establecidas en el cronograma y el proceso de pruebas no se cumple o se ejecuta de una manera desorganizada, sin método y sin considerar los tiempos establecidos para esta fase. El resultado es un software sin las pruebas mínimas requeridas y sin el nivel de calidad esperado.

### 1.2 Pruebas. Conceptos. Costo

Una prueba es una actividad en la cual un sistema o uno de sus componentes se ejecutan en circunstancias previamente especificadas, los resultados se observan y registran, y se realiza una evaluación de algún aspecto. Sencillamente, es el proceso de ejecutar un programa con el fin de encontrar errores. (Patón, 2006)

Algunas características comunes a todas las pruebas de software se listan a continuación.  
(Gutiérrez, 2005)

- Para probar necesitamos código que se pueda ejecutar.
- Para probar necesitamos saber cual es el resultado esperado.
- Una prueba es mejor que otra cuando encuentra más errores.
- Es imposible probar una aplicación al 100%.
- Las pruebas no deben dejarse para el final.
- Deben ser finitas.
- Deben ser mínimas (\*).
- Deben ser fácilmente repetibles (\*\*).

(\*) – El mínimo conjunto de pruebas que permita alcanzar el nivel de prueba establecido.

(\*\*) – Rara vez un conjunto de pruebas se ejecuta una única vez. Si un conjunto de pruebas descubre errores, es normal ejecutarlas una segunda vez para verificar que dichos errores han sido corregidos.

Las pruebas constituyen el último bastión desde el que se puede evaluar la calidad y, de forma más pragmática descubrir errores. Estas no deben ser vistas como una red de seguridad. Como se suele decir: <<No se puede probar la calidad. Si no está ahí antes de comenzar la prueba, no estará cuando se termine>>. La calidad se incorpora en el software durante el proceso de ingeniería del mismo. La aplicación adecuada de los métodos y de las herramientas, las revisiones técnicas formales efectivas y una sólida gestión y medición, conducen a la calidad, que se confirma durante las pruebas. (Pressman, 2005)

Según William Howden la prueba es una parte inevitable de cualquier esfuerzo responsable para desarrollar un sistema de software.

En general las pruebas de software son una parte del proceso de aseguramiento de calidad; realizar pruebas a un sistema de información no significa necesariamente que el proceso de



desarrollo esté asegurado y tampoco que de manera directa esté mejorando, pero nos permite encontrar fallas o características no deseadas que pueden socavar la calidad de un producto.

### 1.2.1 Casos de prueba

Un caso de prueba es un conjunto específico de entradas de prueba, condiciones de ejecución, y resultados esperados, que se identifican con el propósito de hacer una evaluación de un aspecto particular de un ítem de prueba determinado. (GCSUA-HSH, 2006)

Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces. Existe una sola regla para el diseño de estos: cubrir todas las posibilidades, con la menor cantidad de casos. (Pressman, 2005)

### 1.2.2 Plan de Pruebas

El propósito de un plan de pruebas es explicitar el alcance, enfoque, recursos requeridos, calendario, responsables y manejo de riesgos de un proceso de pruebas. (Teruel, 2001)

Un plan de pruebas incluye:

- **Identificador del plan:** Preferiblemente de alguna forma mnemónica que permita relacionarlo con su alcance, por ej. TP-Global (plan global del proceso de pruebas), TP-Req-Seguridad1 (plan de verificación del requerimiento 1 de seguridad), TP-Contr-X (plan de verificación del contrato asociado al evento de sistema X), TP-Unit-Despachador.iniciar (plan de prueba unitario para el método iniciar de la clase Despachador). Como todo artefacto del desarrollo, está sujeto a control de configuración, por lo que debe distinguirse adicionalmente la versión y fecha del plan.
- **Alcance:** Indica el tipo de prueba y las propiedades/elementos del software a ser probado.
- **Ítems a probar:** Indica la configuración a probar y las condiciones mínimas que debe cumplir para comenzar a aplicarle el plan. Por un lado, es difícil y riesgoso probar una configuración que aún reporta fallas; por otro, si esperamos a que todos los módulos estén perfectos, puede que detectemos fallas graves demasiado tarde.

- **Estrategia:** Describe la técnica, patrón y/o herramientas a utilizarse en el diseño de los casos de prueba. Por ejemplo, en el caso de pruebas unitarias de un procedimiento, esta sección podría indicar: "Se aplicará la estrategia caja-negra de fronteras de la precondition" o "Ejercicio de los caminos ciclomáticos válidos". En lo posible la estrategia debe precisar el número mínimo de casos de prueba a diseñar, por ej. 100% de las fronteras, 60% de los caminos ciclomáticos... La estrategia también explicita el grado de automatización que se exigirá, tanto para la generación de casos de prueba como para su ejecución.
- **Categorización de la configuración:** Explicita las condiciones bajo las cuales, el plan debe ser suspendido, repetido y culminado. En algunas circunstancias (las cuales deben ser explicitadas) el proceso de prueba debe suspenderse en vista de los defectos o fallas que se han detectado. Al corregirse los defectos, el proceso de prueba previsto por el plan puede continuar, pero debe explicitarse a partir de qué punto, pues es probable que sea necesario repetir algunas pruebas. Los criterios de culminación pueden ser tan simples como aprobar el número mínimo de casos de prueba diseñados o tan complejos como tomar en cuenta no sólo el número mínimo, sino también el tiempo previsto para las pruebas y la tasa de detección de fallas.
- **Tangibles:** Explicita los documentos a entregarse al culminar el proceso previsto por el plan p. ej. subplanes, especificación de pruebas, casos de prueba, resumen gerencial del proceso y bitácora de pruebas.
- **Procedimientos especiales:** Identifica el grafo de las tareas necesarias para preparar y ejecutar las pruebas, así como cualquier habilidad especial que se requiere.
- **Recursos:** Especifica las propiedades necesarias y deseables del ambiente de prueba, incluyendo las características del hardware, el software de sistemas (p. ej. el sistema de operación), cualquier otro software necesario para llevar a cabo las pruebas, así como la colocación específica del software a probar (p. ej. qué módulos se colocan en qué máquinas de una red local) y la configuración del software de apoyo.

- **Manejo de riesgos:** Explicita los riesgos del plan, las acciones mitigantes y de contingencia.
- **Calendario:** Esta sección describe los hitos del proceso de prueba y el grafo de dependencia en el tiempo de las tareas a realizar.
- **Responsables:** Especifica quién es el responsable de cada una de las tareas previstas en el plan.

Un plan de pruebas traza la clase de prueba que se han de llevar a cabo, y un procedimiento de prueba define los casos de prueba específicos en un intento por descubrir errores de acuerdo con los requisitos. Tanto el plan como el procedimiento estarán diseñados para asegurar que se satisfacen todos los requisitos funcionales, que se alcanzan todos los requisitos de rendimiento, que la documentación es correcta e inteligible y que se alcanzan otros requisitos (por ejemplo, portabilidad, compatibilidad, recuperación de errores, facilidad de mantenimiento). (Pressman, 2005)

### 1.2.3 Estrategia de pruebas

Algunos autores afirman que el proceso de ejecución de pruebas debe ser considerado durante todo el ciclo de vida de un proyecto, para así obtener un producto de alta calidad. Su éxito dependerá del seguimiento de una estrategia de prueba adecuada.

Una estrategia de prueba de software integra un conjunto de actividades que describen los pasos que hay que llevar a cabo en un proceso de prueba: la planificación, el diseño de casos de prueba, la ejecución y los resultados, tomando en consideración cuánto esfuerzo y recursos se van a requerir, con el fin de obtener como resultado una correcta construcción del software. (Pressman, 2005)

Se han propuesto varias estrategias de prueba de software en distintos libros. Todas proporcionan al ingeniero del software una plantilla para la prueba y tienen las siguientes características generales.

- Las pruebas comienzan a nivel de módulo y trabajan <<hacia afuera>>, hacia la integración de todo el sistema basado en computadora.
- Según el momento, son apropiadas diferentes técnicas de prueba.
- La prueba la lleva a cabo el responsable del desarrollo del software y (para grandes proyectos) un grupo independiente de pruebas.
- La prueba y la depuración son actividades diferentes, pero la depuración se debe incluir en cualquier estrategia de prueba.

Una estrategia de prueba 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. Una estrategia debe proporcionar una guía al profesional y un conjunto de hitos para el jefe de proyecto. 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. (Pressman, 2005)

#### **1.2.4 Costos del proceso de prueba**

Un estudio de 1027 proyectos de sistemas de información revela que solo el 12,7% concluyó satisfactoriamente. La baja calidad de los requisitos fue considerado el principal motivo de fracaso en el 76% de los proyectos. Los problemas más comunes en los requisitos son: ambigüedad, imprecisión, inexactitud e inconsistencia. Asegurar la calidad de los requisitos es una tarea casi imprescindible para culminar un proyecto de desarrollo de software con éxito. (Gutiérrez, 2005)

Existen varias técnicas ampliamente aceptadas y documentadas para mejorar la calidad de los requisitos, como pueden ser revisiones formales, inspecciones, listas de verificación, ontologías, auditorías, matrices de rastreabilidad, métricas de calidad, etc.

Desde mediados de los años setenta se estima que el coste de corregir un error en un sistema de software aumenta a medida que avanza el desarrollo del sistema. El coste de corregir un error en

las últimas etapas se estima entre 50 y 100 veces por encima del coste de corregir en mismo error en las primeras etapas. (Gutiérrez, 2005)

Otros estudios más recientes, como (Mogyorodi, 2001), siguen confirmando esta tendencia. Las estimaciones de (Mogyorodi, 2001) se recogen en la tabla 1. Actualmente, estas estimaciones siguen teniendo plena vigencia y valores similares.

**Tabla 1 Coste de corrección de errores (2001)**

Fase	Porcentaje
Requisitos	1
Diseño	3-6
Codificación	10
Pruebas unitarias / integración	15-40
Pruebas de sistema / aceptación	30-70
Producción	40-100

Los costes de corregir errores en la fase de prueba son mucho mayores que los costes de corregir dichos errores en fases tempranas, como la fase de requisitos o de análisis. Adelantar la fase de pruebas a la fase de requisitos o de análisis permite reducir el coste de corrección de errores. Esto debe complementarse con estrategias que permitan, además, disminuir el número de errores potenciales.

Sin embargo, como se ha visto en la sección 1.2, la realización de una prueba implica la ejecución del código. Para probar es necesario disponer del código del sistema. Por este motivo no es posible adelantar toda la fase de pruebas a etapas tempranas como la fase de requisitos o análisis.

Sin embargo, sí es posible adelantar la generación de casos de prueba a etapas tempranas. La generación de pruebas en la etapa de requisitos o análisis va a permitir encontrar fallos, inconsistencias, omisiones y ambigüedades en las especificaciones del sistema cuando es más económico corregirlas.

Además, adelantando la generación de pruebas se evita dejar todo el proceso de prueba para final de la construcción, momento donde el tiempo y los recursos disponibles suelen ser insuficientes para realizar este proceso adecuadamente.

### 1.3 Niveles y tipos de prueba

Son numerosas y variables las definiciones de los diferentes tipos de prueba que se pueden encontrar. La metodología Métrica 3 (CISMAP, 2000) enumera los niveles de prueba que se muestran en la siguiente tabla.

**Tabla 2 Descripción de los niveles de prueba del software**

Niveles de pruebas	Fase de realización	Descripción
Pruebas Unitarias	Durante la construcción del sistema	Prueban el diseño y el comportamiento de cada uno de los componentes del sistema una vez construidos.
Pruebas de Integración	Durante la construcción del sistema	Comprueban la correcta unión de los componentes del sistema entre sí a través de sus interfaces, y si cumplen con la funcionalidad establecida
Pruebas de Sistema	Después de la construcción del sistema	Prueban a fondo el sistema, comprobando su funcionalidad e integridad globalmente, en un entorno lo más parecido posible al entorno final de producción.
Pruebas de Implantación	Durante la implantación en el entorno de producción	Comprueba el correcto funcionamiento del sistema dentro del entorno real de producción
Pruebas de Aceptación	Después de la implantación en el entorno de producción	Verifican que el sistema cumple con todos los requisitos indicados y permite que los usuarios del sistema den el visto bueno definitivo
Pruebas de Regresión	Después de realizar modificaciones al sistema	El objetivo es comprobar que los cambios sobre un componente del sistema, no generan errores adicionales en otros componentes no modificados

Es probable hallar documentos que definan estos niveles como tipos de prueba, el término nivel viene dado por el momento específico del desarrollo del software en que son realizadas estas pruebas. Estos niveles aglutinan tipos de pruebas, técnicas y métodos definidos a continuación.

### 1.3.1 Prueba de Caja Blanca

La prueba de caja blanca, denominada a veces prueba de caja de cristal es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de la caja blanca el ingeniero del software puede obtener casos de prueba que:

- Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites y con sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

Es por ello que se considera a la prueba de Caja Blanca como uno de los tipos de pruebas más importantes que se le aplican a los software, logrando como resultado que disminuya en un gran por ciento el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad. (Pressman, 2005)

#### 1.3.1.1 Prueba de camino básico

La prueba del camino básico es una técnica de prueba de caja blanca propuesta inicialmente por Tom McCabe. Esta técnica permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa. (Pressman, 2005)

#### 1.3.1.2 Prueba de condición

La prueba de condición es un método de diseño de casos prueba que ejercita las condiciones lógicas contenidas en el módulo de un programa. Una condición simple es una variable lógica o una expresión relacional. Una condición compuesta está formada por dos o más condiciones simples, operadores lógicos y paréntesis.

Por tanto, los tipos posibles de componentes en una condición pueden ser: un operador lógico, una variable lógica, un par de paréntesis lógicos, un operador relacional o una expresión aritmética.

Existen además estrategias de pruebas de condiciones como: prueba de ramificaciones, prueba del dominio, BRO (prueba del operador relacional y de ramificación). (Pressman, 2005)

#### **1.3.1.3 Prueba de flujo de datos**

El método de prueba de flujo de datos selecciona caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa. (Pressman, 2005)

#### **1.3.1.4 Prueba de bucles**

La prueba de bucles se centra exclusivamente en validez de las construcciones de bucles. Se pueden definir cuatro clases diferentes de bucles: simples, concatenados, anidados y no estructurados.

### **1.3.2 Prueba de Caja Negra**

Las pruebas de caja negra se centran en los requisitos funcionales del software, permitiendo al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. No suponen una alternativa a las pruebas de caja blanca, sino que permiten descubrir otra clase de errores:

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

La prueba de caja negra intencionadamente ignora la estructura de control y se centra en el campo de la información. (Pressman, 2005)



### **1.3.2.1 Pruebas basadas en grafos**

El primer paso en la prueba de caja negra es entender los objetos que se modelan en el software y las relaciones que conectan a estos objetos. Una vez que se ha llevado a cabo esto, el siguiente paso es definir una serie de pruebas que verifiquen que todos los objetos tienen entre ellos las relaciones esperadas. Dicho de otra manera, la prueba del software empieza creando un grafo de objetos importantes y sus relaciones, y después diseñando una serie de pruebas que cubran el grafo de manera que se ejerciten todos los objetos y sus relaciones para descubrir los errores. (Pressman, 2005)

### **1.3.2.2 Partición equivalente**

La partición equivalente según (Pressman, 2005) es un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Esta se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. El diseño de casos de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada. Las clases de equivalencia se pueden definir según las siguientes directrices:

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

### **1.3.2.3 Análisis de valores límite**

El análisis de valores límite (AVL) es una técnica de diseño de casos de prueba que

complementa la partición equivalente. En lugar de seleccionar cualquier elemento de una clase de equivalencia, el AVL lleva la elección de casos de prueba en los extremos de la clase. En lugar de centrarse solamente en las condiciones de entrada, el AVL obtiene casos de prueba también para el campo de salida. (Pressman, 2005)

### 1.3.3 Pruebas de Interfaces Gráficas de Usuario (IGUs)

Debido a los componentes reutilizables provistos como parte de los entornos de desarrollo de las IGUs, la creación de la interfaz de usuario se ha convertido en menos costosa en tiempo y más exacta. Al mismo tiempo, la complejidad de las IGUs ha aumentado, originando más dificultad en el diseño y ejecución de los casos de prueba. Dado que las IGUs modernas tienen la misma apariencia y filosofía, se pueden obtener una serie de pruebas estándar. Los métodos basados en grafos pueden ser utilizados para realizar pruebas que vayan dirigidas sobre datos específicos y programas objeto que sean relevantes para las IGUs. (Pressman, 2005)

Las siguientes preguntas pueden utilizarse para crear una serie de pruebas genéricas.

#### Ventanas

- ¿Se abren apropiadamente basadas en comandos digitalizados o de menú?
- ¿Se puede modificar su tamaño, mover y scroll?
- ¿Todos los contenidos dentro de la ventana son apropiadamente direccionables con un mouse, teclas funcionales, flechas de dirección y teclado?
- ¿Se regenera apropiadamente cuando es sobrescrita y después rellamada?
- ¿Están todas las funciones disponibles cuando se necesitan?
- ¿Están todas las funciones operacionales?
- ¿Están disponibles y se despliegan apropiadamente los menús, barras, diálogos, botones, iconos y otros controles?
- ¿Cuando se despliegan múltiples ventanas, se representan apropiadamente sus nombres?
- ¿Se destaca apropiadamente la ventana activa?
- ¿En caso de multitareas, se actualizan todas las ventanas en momentos apropiados?

- ¿Clic del mouse múltiples o incorrectos dentro de la ventana causan efectos laterales no esperados?
- ¿Cierran apropiadamente?

#### Menús y operaciones del mouse

- ¿Se despliega la barra de menú apropiada y en el contexto apropiado?
- ¿Funcionan apropiadamente los menús descomponibles, paletas y barras de herramientas?
- ¿Están apropiadamente listadas las funciones y subfunciones desplegadas de los menús?
- ¿Son todas las funciones de menú direccionables apropiadamente por el mouse?
- ¿Es correcto el tipo, tamaño y formato del texto?
- ¿Es posible invocar cada función de menú usando su comando de texto alternativo?
- ¿Se desempeña cada función del menú según se anuncia?
- ¿Son los nombres de los menús autoexplicativos?
- ¿Existe ayuda disponible para cada ítem de menú?
- ¿Se reconocen las operaciones del mouse en forma apropiada a través del contexto interactivo?
- ¿Si se requieren múltiples clic se reconocen apropiadamente en el contexto?
- ¿Si el mouse tiene múltiples botones se reconocen apropiadamente en el contexto?
- ¿Cambian apropiadamente el cursor, indicador de procesamiento y puntero según se invocan diferentes operaciones?

#### Entrada de datos

- ¿La entrada alfanumérica se despliega e ingresa al sistema apropiadamente?
- ¿Funcionan adecuadamente los modos gráficos de entrada de datos?
- ¿Se reconocen apropiadamente los datos inválidos?
- ¿Son entendibles los mensajes de entrada de datos?

#### 1.3.4 Pruebas de la documentación y facilidades de ayuda

La prueba de la documentación puede enfocarse en dos fases. La primera fase, revisión e inspección, examina el documento para comprobar la claridad editorial. La segunda fase, la prueba en vivo, utiliza la documentación junto al uso del programa real. Esta prueba se puede enfocar usando técnicas análogas a muchos de los métodos de prueba de caja negra. (Pressman, 2005)

Las siguientes preguntas pueden utilizarse para crear una serie de casos de prueba.

- ¿Describe con precisión la documentación cómo cumplir cada modo de uso?
- ¿Es precisa la descripción de cada secuencia de iteración?
- ¿Son precisos los ejemplos?
- ¿Son consistentes la terminología, descripciones de menú, y respuestas del sistema con el programa real?
- ¿Se pueden resolver problemas fácilmente con la documentación?
- ¿Son precisos la tabla de contenido o índice de la documentación?
- ¿Es el diseño de la documentación conducente a la comprensión y rápida asimilación de información?
- ¿Todos los mensajes de error desplegados para el usuario están descritos en mayor detalle en el documento?

### **1.3.5 Pruebas unitarias**

Es una forma de probar el correcto funcionamiento de un módulo de código y son diseñadas por los programadores. Permiten asegurar que cada uno de los módulos funcione correctamente por separado. Proveen un mejor modo de manejar la integración de las unidades en componentes mayores, buscan que el código funcione de acuerdo con las especificaciones y que el módulo lógico implementado sea válido. Para realizar este tipo de pruebas se deben particionar los módulos en unidades lógicas fáciles de probar, definiendo para cada unidad los casos de prueba a realizar. Estos casos de prueba deben diseñarse de tal forma que recorran todos los caminos lógicos posibles dentro del código bajo prueba considerando rutinas de excepción, rutinas de error, manejo de parámetros, validaciones, valores válidos, valores límite, rangos, mensajes

posibles, etc.

En el caso de la metodología XP estas pruebas se desarrollan antes de crear el módulo a probar y evolucionan junto con el código. En (Pressman, 2005) se define como prueba de unidad aquella que se centra en el proceso de verificación en la menor unidad del diseño del software, definiendo que en la unidad se debe probar: la interfaz, estructuras de datos locales, condiciones límite, caminos independientes, caminos de manejo de errores.

Es importante darse cuenta de que las pruebas unitarias no descubrirán todos los errores del código. 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.

### **1.3.6 Pruebas de integración**

Es una técnica sistemática para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción. El objetivo es coger los módulos probados mediante la prueba de unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño.

Este tipo de prueba cuenta con varias estrategias de integración incremental que pueden ser ampliadas en (Pressman, 2005) como son: la integración descendente y la ascendente. Además se puede profundizar sobre la prueba de humo que es un método de prueba de integración.

### **1.3.7 Pruebas de validación**

Es donde se validan los requisitos establecidos como parte del análisis de requisitos del software, comparándolos con el sistema que ha sido construido. La prueba de validación proporciona una seguridad final de que el software satisface todos los requisitos funcionales, de comportamiento y de rendimiento. Durante la validación se usan exclusivamente técnicas de caja negra. (Pressman, 2005)

### **1.3.8 Pruebas de usabilidad**

Las pruebas de usabilidad son una forma de medir qué tan bien puede una persona usar un

objeto hecho por el hombre, en este caso un sistema computacional. Existen algunas definiciones que alegan que las pruebas de usabilidad es tomar un grupo de usuarios para los cuales fue desarrollada una aplicación y en función de la reacción de dichos usuarios al interactuar con el sistema, definir el nivel de usabilidad de la aplicación y los posibles elementos a modificar en el sistema. Esto indica que estas pruebas forman parte de las pruebas de aceptación, sin embargo sería beneficioso pensar en qué se puede hacer para lograr una buena aceptación del sistema por los usuarios desde el inicio de la construcción de la aplicación. En este caso existen una serie de aspectos a tener en cuenta, como son: estética, consistencia, documentación para el usuario, mapas de navegación, buen sistema de navegación, ayudas en línea, etc.

### **1.3.9 Pruebas de aceptación**

La prueba de aceptación es realizada por un grupo de usuarios finales o los clientes para asegurarse que el sistema desarrollado cumple los requisitos deseados. Estas pruebas pueden realizarse al terminar la implementación de cada módulo y sirven para detectar nuevas funcionalidades. Las pruebas de aceptación son más importantes que el resto dado que significan la satisfacción del cliente con el producto desarrollado y el final de una iteración y el comienzo de la siguiente. Las pruebas alfa y beta son de aceptación dado que son realizadas por clientes y/o usuarios finales en entornos reales o semejantes al de explotación, diferenciándose la alfa de la beta en que el usuario es supervisado por el desarrollador, en el lugar de desarrollo.

### **1.3.10 Pruebas del sistema**

En (Pressman, 2005) se define sistema como la integración de software, hardware e información, fundamentando así que la prueba de sistema está constituida por una serie de pruebas diferentes cuyo propósito primordial es ejercitar profundamente el sistema basado en computadora. Aunque cada prueba tiene un propósito diferente, todas trabajan para verificar que se han integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas.

#### **1.3.10.1 Pruebas de seguridad**

Pruebas enfocadas en asegurar que los datos o el sistema pueden ser accedidos solamente

por aquellos actores autorizados.

#### **1.3.10.2 Pruebas de volumen**

Dirigidas a la verificación de la habilidad de manejar grandes cantidades de datos, bien sea como entrada, salida o residente en una base de datos.

#### **1.3.10.3 Pruebas de integridad**

Diseñadas para probar la robustez (resistencia a fallas) y el uso adecuado del lenguaje, sintaxis y uso de recursos. Este tipo de prueba puede aplicarse tanto a unidades como a integración de unidades.

#### **1.3.10.4 Pruebas de recuperación**

Esta prueba fuerza el fallo del software de muchas formas y verifica que la recuperación se lleva a cabo apropiadamente. Si la recuperación es automática hay que evaluar la corrección de la inicialización, de los mecanismos de recuperación del estado del sistema, de la recuperación de datos y del proceso de re arranque. Si la recuperación requiere de la intervención humana, hay que evaluar los tiempos medios de reparación para determinar si están dentro de unos límites aceptables. (Pressman, 2005)

#### **1.3.10.5 Pruebas de stress**

Esta se refiere a extrema carga, memoria insuficiente, no disponibilidad de servicios y hardware o recursos compartidos limitados. Este tipo de prueba permite comprender mejor cómo y qué áreas del sistema colapsarán, de este modo es posible planificar contingencias, actualizar el mantenimiento, planear y asignar recursos de antemano.

#### **1.3.10.6 Pruebas de Benchmark**

Compara el desempeño del elemento objetivo de la prueba con un sistema conocido y una carga de trabajo definida.

#### **1.3.10.7 Pruebas de contención**

Validar que el elemento que se prueba maneja adecuadamente cuando muchos actores

solicitan el mismo recurso.

#### **1.3.10.8 Pruebas de carga**

Validar y evaluar la aceptabilidad de un elemento de un sistema sobre diferentes cargas de trabajo mientras el sistema permanece constante. Generalmente se incluye simulación de cargas de trabajo promedio y pico que puedan ocurrir dentro de la tolerancia operacional normal.

#### **1.3.10.9 Pruebas de perfil de desempeño**

Monitorea el perfil en el tiempo incluyendo flujo de ejecución, acceso a datos, llamadas a funciones para identificar cuellos de botella y procesos ineficientes.

#### **1.3.10.10 Pruebas de configuración**

Se enfocan en evaluar aquellos elementos configurados para diferente hardware y/o configuraciones de software. Pueden implementarse como pruebas de rendimiento del sistema.

#### **1.3.10.11 Pruebas de instalación**

Se enfoca en evaluar que el elemento a probar se instala como se indica, en diferentes hardware y/o configuraciones de sistemas de software y bajo diferentes condiciones (tales como espacio insuficiente en disco, interrupción de electricidad). Este tipo de prueba se aplica y ejecuta sobre aplicaciones y sistemas.

### **1.3.11 Pruebas de regresión**

Las pruebas de regresión consisten en verificar que el software sigue cumpliendo las especificaciones y requisitos después de cada versión. Son realizadas por desarrolladores y analistas de la aplicación. Pueden ser ejecutadas en el nivel de integración, para minimizar costos.

## **1.4 Procesos de pruebas estandarizados**

Son conocidos los beneficios de estandarizar procesos y productos, la ingeniería de software



como proceso no queda fuera de esto, ni el proceso de pruebas software como subproceso de la ingeniería. Existen varias organizaciones dedicadas al desarrollo de estándares y normas como el caso de la IEEE (Institute of Electrical and Electronics Engineers) y la ISO (Organización Internacional de Estándares) que han desarrollado algunas normas referentes al proceso de desarrollo de software; por citar algunas:

- IEEE/ANSI Standard 829-1983, Software Test Documentation
- IEEE/ANSI Standard 1008-1987, Software Unit Testing
- IEEE/ANSI Standard 1012-1986, Software Verification and Validation Plans
- IEEE/ANSI Standard 1028-1988, Software Reviews and Audits
- IEEE/ANSI Standard 730-1989, Software Quality Assurance Plans
- IEEE/ANSI Standard 610.12-1990, Glossary of Software Engineering Terminology
- ISO 9000 Sistemas de Gestión de la Calidad - Fundamentos y vocabulario
- ISO 9000-3 Establece directrices para la aplicación de la norma ISO 9001 dedicada al desarrollo, suministro y mantenimiento de software.
- ISO 9001 Sistemas de Gestión de la Calidad - Requisitos
- ISO/IEC 9126 Calidad del Software
- ISO 9241 Guías sobre usabilidad. (Arango, 2005)
- ISO 9004 Sistemas de Gestión de la Calidad - Directrices para la mejora del desempeño

La Integración del Modelo de Capacidad de Madurez es otro ejemplo de estándar del proceso ingenieril del software. Se desarrolló para facilitar y simplificar la adopción de varios modelos de forma simultánea, y su contenido integra y da relevo a la evolución de sus predecesores:

- CMM-SW (CMM for Software)
- SE-CMM (Systems Engineering Capability Maturity Model)
- IPD-CMM (Integrated Product Development)

El cuerpo de conocimiento disponible en CMMI incluye:

- Systems engineering (SE)
- Software engineering (SW)

- Integrated product and process development (IPPD)
- Supplier sourcing (SS)

Además de estos modelos y normas se han creado metodologías para el desarrollo del software, como RUP, XP, MSF y otras muchas.

A pesar de estas metodologías, normas y modelos ser universalmente usados, deberían existir otros que se ajusten más a las características del producto a desarrollar, como es el caso de los libros electrónicos y otros tipos de software que tienen características especiales.

En Cuba el software generalmente es desarrollado por empresas. Estas se dedican a un tipo de software o sus trabajos van dirigidos a un campo específico. Está el caso de SOFTEL que desarrolla software para la salud, DESOFT para el gobierno y CITMATEL a desarrollar mayormente aplicaciones educativas hipermedia. En SOFTEL utilizan la metodología RUP para realizar pruebas adaptadas a sus productos. En el caso de algunos productos multimedia que han realizado se determinó realizar pruebas de funcionalidad e integración.

En el caso de CITMATEL utilizan la metodología RUP para guiar sus proyectos realizando adaptaciones a esta metodología. Para evaluar la calidad de sus productos se apoyan en la Norma ISO/IEC 9126 la cual plantea un marco de referencia para evaluar la calidad del software, al caso específico del software educativo. Esta norma define un conjunto de características, sub-características y atributos de calidad del software a tener en cuenta para la prueba de aceptación del producto.

En la UCI se utiliza la metodología RUP para guiar y evaluar el desarrollo de algunos productos. En el caso de las aplicaciones hipermedia son revisadas generalmente desde el punto de vista funcional, con una pobre base en estándares, metodologías, métodos, etc. Se han desarrollado algunos estudios sobre el proceso de desarrollo de este tipo de producto y algunos proyectos pretenden adoptar como metodología de desarrollo a OMMA-L. Además fue presentado por la Msc. Yadenis Piñero como trabajo para optar por la maestría una metodología para desarrollar productos hipermedia, la cual debe ser probada en un ambiente productivo para una futura

evaluación.

## 1.5 Propuestas de generación de casos de prueba

En la Universidad de Sevilla, año 2005, fue presentado por Javier Jesús Gutiérrez un resumen comparativo de varias propuestas de generación de pruebas centradas en la funcionalidad (Gutiérrez, 2005), realizadas hasta aquel momento. Resumiendo en dicho trabajo la incompletitud, la falta de herramientas automatizadas y el no uso de estándares, en dichas propuestas, destacando además la importancia que tienen estas pruebas y la creación de herramientas automatizadas para lograr mayor fiabilidad y rapidez en el proceso de realización y ejecución de estas pruebas. Luego de este trabajo ha publicado otros relacionados con el tema, en conjunto con otros autores, los cuales han sido presentados en simposios, talleres y conferencias nacionales e internacionales. Entre estos trabajos destacan los siguientes:

- Mejorando la calidad de los requisitos mediante la generación de pruebas. J. J. Gutiérrez, M. J. Escalona, M. Mejías, J. Torres, Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla. 2005
- Pruebas del Sistema en Programación Extrema, J. J. Gutiérrez, M. J. Escalona, M. Mejías, J. Torres, Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla. 2005
- Modelos de Pruebas para Pruebas del Sistema, Javier J. Gutiérrez, María J. Escalona, Manuel Mejías y Antonia M. Reina, Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, XV Jornadas de Ingeniería del Software y Bases de Datos 2006.
- Hacia una Propuesta de Pruebas Tempranas del Sistema, Javier J. Gutiérrez, María J. Escalona, Arturo H. Torres, Manuel Mejías y Jesús Torres, Departamento de Lenguajes y Sistemas Informáticos, Escuela Técnica Superior de Ingeniería Informática, Universidad de Sevilla, Taller sobre Pruebas en Ingeniería del Software, PRIS 2006.

En estos trabajos se entiende por pruebas del sistema las denominadas en (Pressman, 2005) como pruebas de validación. Los mismos se pueden encontrar en la siguiente dirección electrónica:

<http://www.lsi.us.es/~javierj/publications.html>.

## 1.6 Herramientas de apoyo a las pruebas

Además de los tipos de pruebas vistos en el epígrafe 1.3 se pueden definir otros como las pruebas manuales y las automáticas; estas últimas son realizadas por un programa o herramienta que prueba el sistema sin necesidad de la interacción de una persona. La herramienta suministra una serie de valores, o acciones de prueba al sistema y verifica los resultados devueltos por el mismo con los resultados esperados. Al final del proceso de prueba se emite un informe con los resultados de las mismas. Cuando se desea repetir el proceso de prueba, generalmente después de una modificación en el mismo, solo es necesario volver a ejecutar la herramienta y emitir un nuevo informe.

Las pruebas manuales, en cambio, son aquellas pruebas realizadas por una o más personas que interactúan directamente con el sistema. Estas personas verifican si los resultados obtenidos son válidos o no. Cuando se desea repetir el proceso es necesario que la persona o grupo repita las interacciones y vuelvan a verificar todos los resultados obtenidos.

Las pruebas automáticas suelen ser más deseables que las manuales ya que evitan el factor humano.

Como ejemplos de herramientas utilizadas para la generación automática de casos de prueba están los siguientes:

- JUnit: Entorno de pruebas para Java creado por Erich Gamma y Kent Beck. Fue el primero y actualmente es el más usado.
- TestNG: Creado para suplir algunas deficiencias en JUnit.
- JTiger: Basado en anotaciones, como TestNG.
- SimpleTest: Entorno de pruebas para aplicaciones realizadas en PHP.
- CCPUnit: Migración del entorno JUnit para lenguajes C/C++.
- NUnit: Migración del entorno JUnit para lenguajes de la plataforma .NET.
- JCap: Versión de JUnit desarrollada para practicas docentes.

- DevPartner de Compuware: Entorno para pruebas unitarias, presentado por Abits Colombia.
- ObjectGen: Es una herramienta para generar diagramas de comportamiento a partir de casos de uso.
- TestComplete de AutomatedQA: Es un Entorno de pruebas automáticas sin necesidad de Scripting. Permite la definición y ejecución de pruebas unitarias, funcionales, distribuidas, de carga, de una forma sencilla mediante la definición de estrategias de pruebas muy intuitivas. Esto permite alcanzar un nivel de garantía de calidad de su producto desde el inicio de la fase de codificación. Las características técnicas y funcionales de TestComplete permiten que sea portable y ejecutable en diferentes entornos tecnológicos, además de integrable con otras herramientas de desarrollo (VS .NET, Visual Basic, Visual C++, Visual FoxPro, Delphi, C++ Builder, Java, HTML, etc.
- IBM Rational Robot permite a los equipos de pruebas (probadores) automatizar las pruebas de configuración, regresión y funcionales de aplicaciones .NET, Java, Web y otras aplicaciones basadas en GUI. Proporciona casos de prueba para objetos comunes, como menús, listas y mapas de bits, y casos de prueba especializados para los objetos específicos del entorno de desarrollo. Incluye un componente de gestión de pruebas y se integra con las herramientas de IBM Rational Team Unifying Platform para realizar el seguimiento de defectos, gestión de cambios y rastreo de requisitos.
- Rational TestManager es la consola central para la administración de la actividad, ejecución y reportes de prueba. Construida para ser extensible, soporta desde pruebas manuales puras a variados paradigmas de automatización incluyendo las pruebas unitarias, funcionales, y de regresión. Rational TestManager esta concebido para ser accedido por todos los miembros del equipo de proyecto, asegurando una alta visibilidad de la información de cobertura de la prueba, tendencia de defectos y grado de preparación de la aplicación.

## **1.7 Procesos de desarrollo de aplicaciones hipermedia**

La hipermedia surge como resultado de la fusión de dos tecnologías, el hipertexto y la multimedia. El hipertexto es la organización de una determinada información en diferentes nodos, conectados entre sí a través de enlaces. Los nodos pueden contener sub-elementos con entidad propia. Un hiperdocumento estaría formado por un conjunto de nodos conectados y relacionados temática y estructuralmente. (Hassan, 2002)

La tecnología multimedia es la que permite integrar diferentes medios (sonido, imágenes, secuencias, etc.) en una misma presentación.

La hipermedia, por tanto, es la tecnología que nos permite estructurar la información de una manera no-secuencial, a través de nodos interconectados por enlaces. La información presentada en estos nodos podrá integrar diferentes medios (texto, sonido, gráficos...).

### **1.7.1 Método de Desarrollo de Aplicaciones Educativas Hipermedia**

En el año 1997 el Grupo de Investigación y Desarrollo de Software Educativo (GIDSE) del Departamento de Ciencias de la Computación de la Universidad del Valle, Cali, Colombia presentó un trabajo titulado "Un Método de Desarrollo de Aplicaciones Educativas Hipermedia" en el Taller Internacional de Software Educativo, el cual se apoya en las guías de la Norma ISO 9000-3 y divide en 6 fases el proceso obtención de un producto multimedia con calidad:

1. Descripción del problema
2. Definición de Requisitos y Términos del Contrato
3. Diseño Educativo
4. Diseño Computacional
5. Producción
6. Evaluación del Producto.

### **1.7.2 Elaboración de Materiales Formativos Multimedia. Criterios de Calidad**

Otro método para la elaboración de software multimedia de corte educativo fue presentado por el Departamento de Pedagogía Aplicada de la Universidad Autónoma de Barcelona en el doceavo

Congreso Nacional y primero Iberoamericano de Pedagogía, Madrid, España, en septiembre del 2000, titulado "Elaboración de Materiales Formativos Multimedia. Criterios de Calidad". Este proceso de desarrollo describe tres fases:

1. Diseño Funcional
2. Desarrollo
3. Post-Producción

Además una serie de criterios de calidad agrupados en tres familias:

1. Aspectos Funcionales
2. Aspectos Técnicos y Estéticos
3. Aspectos Pedagógicos

## **1.8 Producto Heredia**

El producto Heredia es un libro electrónico dedicado al poeta, profesor, historiador y crítico José María Heredia, en el bicentenario de su natalicio, la propuesta fue realizada por la Biblioteca Nacional José Martí (BNJM). Este libro electrónico contiene una numerosa cantidad de documentos de los 254 con que cuenta la BNJM, fruto del talento de los investigadores y especialistas del Departamento de Manuscritos y de los especialistas y técnicos del Laboratorio Digital, además de algunos textos suministrados por la Dra. Ana Cairo, la especialista mexicana Alicia Ponce y Tomás Fernández Robaina. Esta aplicación hipertexto, fue concebida para el estudio, apreciación y valoración, de estudiosos de la obra del poeta, especialistas interesados y público en general. El título de la aplicación es "José María Heredia en el bicentenario de su natalicio", su autor es Gabriel George y según la aplicación está clasificada como un ensayo.

La aplicación está dividida en 4 secciones o áreas:

Área de Título del programa y documento actual

Área de navegación entre módulos

Área de contenidos

Panel de servicios

La aplicación fue realizada con la herramienta Flash MX 2004 y debe correr sin dificultad en el

sistema operativo Windows 95, Pentium 133 Mhz, 64 MB de RAM.

## **1.9 Conclusiones**

Según lo expuesto en este capítulo una propuesta de solución al problema en cuestión sería la definición de una estrategia de prueba la cual estaría conformada por un plan de pruebas y una estructura, puede ser una o varias tablas, para archivar los resultados de las pruebas. Teniendo en cuenta las características y el estado de la aplicación a probar, los tipos de prueba que podrían utilizarse serían las de validación, usabilidad, instalación, perfil de desempeño y a la documentación. En el siguiente capítulo se define el plan de pruebas que guiará el proceso.



A square graphic with a light gray background. At the top, the word "Capítulo" is written in a light gray sans-serif font. Below it, a large, bold, dark gray number "2" is centered.

## PLAN DE PRUEBAS

### 2.1 Introducción

En este capítulo se describe y confecciona el plan de pruebas que se define como parte de la solución al problema descrito en la Introducción de este documento, apoyándonos para esto en la definición de plan de pruebas expuesto en el capítulo anterior y la guía propuesta en (GCSUA-HSH, 2006). Según esta guía debería crearse un plan por cada componente del sistema, pero como el producto “Heredia” no es el resultado de la integración de módulos de funcionalidad, los subplanes serán organizados de acuerdo a la información que se tiene de la aplicación y los tipos de prueba a realizar. Estos subplanes son parte de los entregables de un plan global que determina el alcance del proceso de pruebas, y otras cuestiones de orden general.

Antes de comenzar con el diseño de los casos de prueba se deben analizar los modelos de prueba, los cuales son aquellos documentos generados en el desarrollo de la aplicación, que tienen relación con las características principales del software a probar. Estos documentos son la base para un correcto diseño de los casos de prueba de cualquier sistema de información. En el artículo titulado “Modelos de pruebas para pruebas del sistema”, referenciado en el epígrafe 1.5 del capítulo anterior, se definen algunos modelos y la relación entre ellos. La aplicación “Heredia” no cuenta con documentos de este tipo, por lo que para la realización de los casos de prueba se utilizará la bibliografía confeccionada en el desarrollo de este producto.

El plan de pruebas debe ser revisado y aprobado, antes de ser ejecutado.

## 2.2 Plan de Pruebas Global del producto Heredia

### 2.2.1 Información del plan

Información del plan de pruebas	
Proyecto que lo generó	Producto Heredia
Identificador	PHPGlobal
Fecha de creación	Del 16 al 17 de abril del 2007
Autor(es)	Jorge Antonio Gómez Colombat

Cuadro de control del procedimiento				
Versión	Aprobado por	Fecha de Aprobación	Revisado por	Fecha de Revisión
1.0	Jorge A. Gómez	19/05/07	Jorge A. Gómez	18/05/07

### 2.2.2 Alcance

- Tipos de prueba: a la documentación, validación, usabilidad, instalación y perfil de desempeño. Algunas pruebas se realizarán a nivel de unidad y otras a nivel de sistema.
- Propiedades/Elementos del software a ser probado: interfaces gráficas, documentación, funcionalidad, usabilidad y rendimiento.

### 2.2.3 Ítems a probar

- Configuración a probar: Aplicación y Documentación
- Condiciones mínimas: Toda la documentación referente a la aplicación, la aplicación y la instalación del Flash Player.

### 2.2.4 Estrategia

Los primeros casos de prueba a ser diseñados son los de prueba a la documentación, dado que lo primero que se debe analizar, para realizar cualquier tipo de prueba, es la especificación de las pautas y requisitos del sistema, que en el caso de aplicaciones hipermedia generalmente es el guión. Estas pruebas serán las de revisión e inspección, pues el enfoque de la prueba en vivo es muy parecido a las pruebas de validación.

Las pruebas de validación se realizarán en dos niveles, en las unidades y en el sistema. Por concepto estas pruebas son un conjunto de pruebas de caja negra, por lo que nos apoyaremos

en la definición de métodos basados en grafos, el cual plantea la definición de nodos y enlaces, donde los nodos son los objetos de datos a nivel de interfaz y los enlaces las relaciones que existen entre dichos objetos, luego de crear el grafo, se verifica que todos los objetos estén presentes en la aplicación, contengan todas sus características y estén relacionados correctamente. De modo general para desarrollar los casos de prueba se utilizará la técnica de Partición Equivalente.

Otros casos de pruebas a diseñar son los relacionados con la usabilidad, tomando como principales características y elementos a evaluar la estética, la consistencia, el sistema de navegación y orientación, la documentación para el usuario, entre otros aspectos contenidos en (Marqués, 2000) definidos como criterios de calidad y algunos mencionados en la definición de pruebas de interfaces gráficas de usuario, como la verificación de que los nombres de los menús son autoexplicativos y la disponibilidad de ayuda para cada ítem de menú. (Marqués, 2000)

La prueba de instalación consistirá en la verificación de la correcta ejecución de la aplicación. Por último la prueba de perfil de desempeño consistirán en evaluar el tiempo de respuesta de las peticiones; para esto se tomará como patrón alguna especificación al respecto, y en caso de ausencia, la evaluación será a consideración del probador. Estos dos últimos tipos de prueba solo serán realizados en caso de que estén presentes las condiciones de software y de hardware especificadas por el cliente.

Todas estas pruebas serán realizadas y ejecutadas de forma manual, debido a la escasez de herramientas automáticas para probar aplicaciones de este tipo.

### **2.2.5 Categorización de la configuración**

Este plan culminará al ejecutar todos los casos de prueba diseñados, o exceda el 25 de mayo del 2007.

### **2.2.6 Tangibles**

Los documentos a entregarse al culminar el proceso previsto por el plan serán:

- Subplanes

- Especificación de pruebas
- Casos de prueba

### 2.2.7 Procedimientos especiales

Este grafo identifica las tareas necesarias para crear el plan de pruebas.

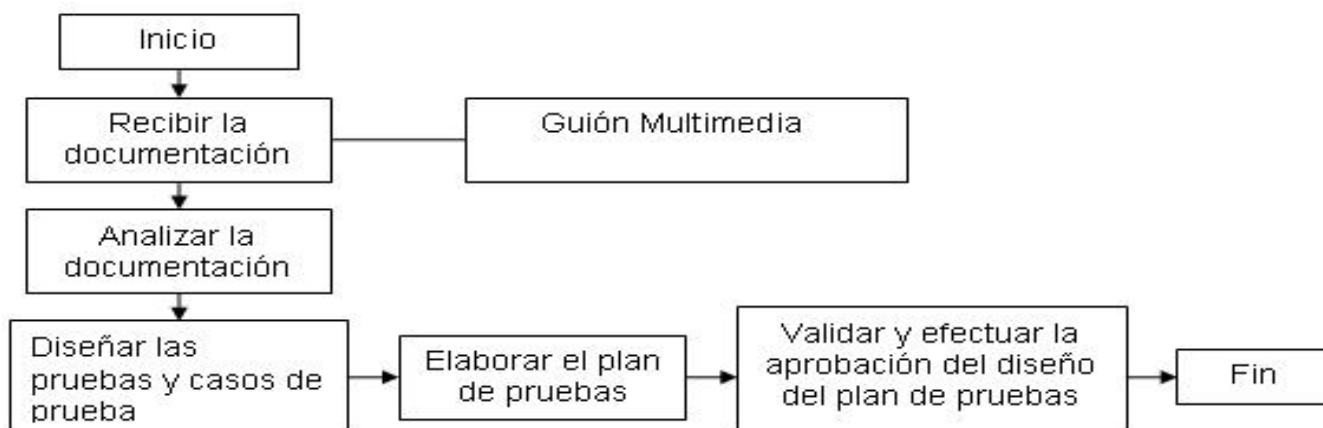


Ilustración 2. 1 Tareas para crear el plan de pruebas

### 2.2.8 Recursos

Para la realización de las pruebas se necesita un ordenador con las siguientes características de hardware:

- Lector de CD.
- Espacio en disco de 500 MB
- 64 MB o más de RAM.
- Parlantes (Bocinas, Audífonos, etc.)
- Mouse (Ratón)
- Tarjeta Gráfica
- Monitor VGA o LCD
- Microprocesador P1 o más potente.

Características de software:

- Sistema Operativo Windows 95 o más reciente.

- Flash Player MX 2004 o más reciente

Otros:

- Una persona
- 8 horas al día de trabajo.

### 2.2.9 Calendario



Ilustración 2. 2 Calendario de actividades relacionadas con las pruebas

### 2.2.10 Responsable

Responsable de las tareas del plan de pruebas:

- Jorge Antonio Gómez Colombat

### 2.3 Plan de Pruebas a la Documentación Técnica

- Tema: PHPD-01
- Descripción: Este plan de pruebas define todas las pruebas y los casos de pruebas relacionados con las pruebas a la documentación de la primera fase, aquellas relacionadas con la revisión e inspección.
- Usuario Responsable: Jorge Antonio Gómez Colombat
- Estado: Aprobado
- Fecha de Creación: 24/04/07, 6.05 p.m.

- Usuario de Creación: Jorge Antonio Gómez Colombat
- Tiempo Total de Ejecución: 385 minutos.

### 2.3.1 Prueba de Revisión e Inspección a la Documentación 1.01

- Nombre de la Prueba: Revisión e Inspección a la Documentación
- Fecha y Usuario de creación: 24/04/07:6.25 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Duración de la Prueba: 375 minutos
- Descripción de la prueba: Este tipo de prueba va dirigido a la claridad editorial de los documentos confeccionados en el desarrollo de la aplicación, o sea, el uso de estándares, la homogeneidad, la completitud, la organización, el nivel de especificación y de detalle, consistencia con respecto a la aplicación, entre otros aspectos.

#### 2.3.1.1 Caso de prueba RID 01

- No. del Caso: RID-01
- Fecha y Usuario de creación: 24/04/07:6.35 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Uso de Estándares en la creación de los documentos
- Duración del Caso: 30 minutos
- Prioridad: 1

No. CP	Entrada	Clase Válida	Clase Inválida
RID-01	Leer todos los documentos relacionados con las especificaciones generales	Se usó algún estándar en la confección de la documentación.	No se usó ningún estándar en la confección de la documentación.

#### 2.3.1.2 Caso de prueba RID 02

- No. del Caso: RID-02
- Fecha y Usuario de creación: 24/04/07:10.20 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:

- Descripción del Caso de Prueba: Homogeneidad, este caso de prueba verifica si los documentos fueron creados con la misma tipografía.
- Duración del Caso: 45 minutos
- Prioridad: 6

No. CP	Entrada	Clase Válida	Clase Inválida
RID-02	Comparar la estructura de todos los documentos que guarden relación con el producto	Se crearon todos los documentos con la misma estructura	No se crearon todos los documentos con la misma estructura

### 2.3.1.3 Caso de prueba RID 03

- No. del Caso: RID-03
- Fecha y Usuario de creación: 24/04/07:10.35 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Completitud, este caso de prueba analiza si la información recogida en la documentación es suficiente para lograr un software con calidad.
- Duración del Caso: 150 minutos
- Prioridad: 2

No. CP	Entrada	Clase Válida	Clase Inválida
RID-03	Analizar la completitud de la información	La documentación es suficiente para lograr un software con calidad.	La documentación no es suficiente para lograr un software con calidad.

### 2.3.1.4 Caso de prueba RID 04

- No. del Caso: RID-04
- Fecha y Usuario de creación: 24/04/07:10.45 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Organización, este caso de prueba analiza la organización de la información
- Duración del Caso: 30 minutos
- Prioridad: 4

No. CP	Entrada	Clase Válida	Clase Inválida
RID-04	Analizar el nivel organizacional de la documentación	La documentación esta correctamente organizada	La documentación no esta correctamente organizada

### 2.3.1.5 Caso de prueba RID 05

- No. del Caso: RID-05
- Fecha y Usuario de creación: 24/04/07:10.59 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Este caso de prueba analiza la consistencia de la descripción general del sistema, con la aplicación.
- Duración del Caso: 60 minutos
- Prioridad: 3

No. CP	Entrada	Clase Válida	Clase Inválida
RID-05	Analizar todas las especificaciones del sistema recogidas en la documentación.	La documentación es consistente en relación con la aplicación.	La documentación no es consistente en relación con la aplicación.

### 2.3.1.6 Caso de prueba RID 06

- No. del Caso: RID-06
- Fecha y Usuario de creación: 24/04/07:11.10 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Este caso de prueba analiza la precisión de la tabla de contenidos o índice de la documentación.
- Duración del Caso: 60 minutos
- Prioridad: 5

No. CP	Entrada	Clase Válida	Clase Inválida
RID-06	Analizar la tabla de contenidos o índice de la documentación y la ubicación de sus elementos.	La tabla de contenidos o índice de la documentación es precisa.	La tabla de contenidos o índice de la documentación no es precisa.



## 2.4 Plan de Pruebas de Validación

- Tema: PHPV-01
- Descripción: Este plan de pruebas define todas las pruebas y los casos de pruebas necesarios para validar la completitud de los requisitos del sistema, tanto funcionales, como no funcionales y de información, apoyándose en el concepto de pruebas basadas en grafos. Para comenzar con estas pruebas debe estar creado un documento con todos los objetos y sus características, y las relaciones entre ellos, ver Anexo1.
- Usuario Responsable: Jorge Antonio Gómez Colombat
- Estado: Aprobado
- Fecha de Creación: 09/05/07:12.15 a.m.
- Usuario de Creación: Jorge Antonio Gómez Colombat
- Tiempo Total de Ejecución: 500 minutos.

### 2.4.1 Prueba de validación 1.01

- Nombre de la Prueba: Pruebas al área de título del programa y documento actual.
- Fecha y Usuario de creación: 09/05/07:12.20 a.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Duración de la Prueba: 10 minutos.
- Descripción de la prueba: Este tipo de prueba va dirigido a la validación de los objetos, sus características y relaciones.

#### 2.4.1.1 Caso de prueba ATPD 01

- No. del Caso: ATPD-01
- Fecha y Usuario de creación: 09/05/07:12.25 a.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la presencia y estructura del área de título del programa y documento actual, en todas las interfaces de la aplicación.
- Flujo Central: Ejecutar la aplicación y esperar que cargue la pantalla principal

- Duración del Caso: 10 minutos
- Prioridad: 1

No. CP	Entrada	Clase Válida	Clase Inválida1	Clase Inválida2
ATPD-01	Recorrer todas las vistas de la aplicación	Aparece el área de título del programa y documento actual en todas las interfaces y tiene la estructura especificada en el Anexo1	No aparece el área de título del programa y documento actual	Aparece el área de título del programa y documento actual en todas las interfaces pero no tiene la estructura especificada en el Anexo1

### 2.4.2 Prueba de validación 2.01

- Nombre de la Prueba: Pruebas al área de navegación entre módulos y al área de contenidos.
- Fecha y Usuario de creación: 09/05/07:2.45 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Duración de la Prueba: 330 minutos.
- Descripción de la prueba: Este tipo de prueba va dirigido a la validación de los objetos, sus características y relaciones.

#### 2.4.2.1 Caso de prueba ANM 01

- No. del Caso: ANM-01
- Fecha y Usuario de creación: 09/05/07:2.50 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la presencia y estructura del área de navegación entre módulos en todas las interfaces de la aplicación.
- Flujo Central: Ejecutar la aplicación y esperar que cargue la pantalla principal
- Duración del Caso: 10 minutos
- Prioridad: 1

No. CP	Entrada	Clase Válida	Clase Inválida1	Clase Inválida2
ANM-01	Recorrer todas las vistas de la aplicación	Aparece el área de navegación entre módulos en todas las interfaces y tiene la estructura especificada en el Anexo1	No aparece el área de navegación entre módulos	Aparece el área de navegación entre módulos en todas las interfaces pero no tiene la estructura especificada en el Anexo1

### 2.4.2.2 Caso de prueba AC 01

- No. del Caso: AC-01
- Fecha y Usuario de creación: 09/05/07:3.15 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la presencia y estructura del área de contenidos en todas las interfaces de la aplicación.
- Flujo Central: Ejecutar la aplicación y esperar que cargue la pantalla principal
- Duración del Caso: 10 minutos
- Prioridad: 2

No. CP	Entrada	Clase Válida	Clase Inválida1	Clase Inválida2
AC-01	Recorrer todas las vistas de la aplicación	Aparece el área de contenidos en todas las interfaces y tiene la estructura especificada en el Anexo1	No aparece el área de contenidos	Aparece el área de contenidos en todas las interfaces pero no tiene la estructura especificada en el Anexo1

### 2.4.2.3 Caso de prueba ANM 02

- No. del Caso: ANM-02
- Fecha y Usuario de creación: 09/05/07:3.25 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación: 10/05/07:4.35 p.m. Jorge Antonio Gómez
- Descripción del Caso de Prueba: Verificar la presencia y la estructura del Botón 1
- Flujo Central: Al cargar la pantalla principal, observar en el área de navegación el primer botón de izquierda a derecha.
- Duración del Caso: 10 minutos
- Prioridad: 3

No. CP	Entrada	Clase Válida	Clase Inválida1	Clase Inválida2
ANM-02	Mostrar la pantalla principal	Aparece el Botón 1 en el área de navegación y tiene la estructura especificada en el Anexo1	No aparece el Botón 1 en el área de navegación	Aparece el Botón 1 en el área de navegación pero no tiene la estructura especificada en el Anexo1

#### 2.4.2.4 Caso de prueba ANM 03

- No. del Caso: ANM-03
- Fecha y Usuario de creación: 09/05/07:4.00 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad del Botón 1
- Flujo Central: Al cargar la pantalla principal, en el área de navegación, el primer botón de izquierda a derecha, ponga el cursor del mouse sobre dicho botón.
- Duración del Caso: 5 minutos
- Prioridad: 4

No. CP	Entrada	Clase Válida	Clase Inválida
ANM-03	Hacer MouseOver sobre el Botón 1	El Botón 1 cuenta con la funcionalidad especificada en el Anexo1	El Botón 1 no cuenta con la funcionalidad especificada en el Anexo1

#### 2.4.2.5 Caso de prueba ANM 04

- No. del Caso: ANM-04
- Fecha y Usuario de creación: 09/05/07:4.15 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación: 10/05/07:4.45 p.m. Jorge Antonio Gómez
- Descripción del Caso de Prueba: Verificar la presencia y estructura del Botón 1.1
- Flujo Central: En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú observar el primer botón de arriba hacia abajo.
- Duración del Caso: 5 minutos
- Prioridad: 5

No. CP	Entrada	Clase Válida	Clase Inválida1	Clase Inválida2
ANM-04	Hacer MouseOver sobre el Botón 1 y luego sobre el Botón 1.1	Aparece el Botón 1.1 y tiene la estructura especificada en el Anexo1	No aparece el Botón 1.1	Aparece el Botón 1.1 pero no tiene la estructura especificada en el Anexo1

### 2.4.2.6 Caso de prueba ANM 05

- No. del Caso: ANM-05
- Fecha y Usuario de creación: 09/05/07:4.00 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad del Botón 1.1
- Flujo Central: En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú hacer clic izquierdo sobre el primer botón de arriba hacia abajo.
- Duración del Caso: 5 minutos
- Prioridad: 6

No. CP	Entrada	Clase Válida	Clase Inválida
ANM-05	Hacer OnClick sobre el Botón 1.1	El Botón 1.1 cuenta con la funcionalidad especificada en el Anexo1	El Botón 1.1 no cuenta con la funcionalidad especificada en el Anexo1

### 2.4.2.7 Caso de prueba AC 02

- No. del Caso: AC-02
- Fecha y Usuario de creación: 10/05/07:5.20 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Validar la calidad del documento mostrado en el área de contenidos.
- Flujo Central: En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú hacer clic izquierdo sobre el primer botón de arriba hacia abajo.
- Duración del Caso: 15 minutos
- Prioridad: 7

No. CP	Entrada	Clase Válida	Clase Inválida
AC-02	Hacer OnClick sobre el Botón 1.1	El documento tiene una calidad aceptable.	El documento no tiene una calidad aceptable.

Nota: La evaluación de la calidad de las imágenes, gráficos y textos se hará en correspondencia, a lo especificado en (Marqués, 2000) sobre este tema.

En caso de ser necesario el uso de objetos que faciliten la visualización y navegación sobre el texto, tener en cuenta los siguientes casos de prueba.

#### 2.4.2.8 Caso de prueba AC 03

- No. del Caso: AC-03
- Fecha y Usuario de creación: 10/05/07:5.35 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la presencia y estructura de los objetos auxiliares que se deben mostrar en el área de contenidos.
- Flujo Central: En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú hacer clic izquierdo sobre el primer botón de arriba hacia abajo.
- Duración del Caso: 5 minutos
- Prioridad: 8

No. CP	Entrada	Clase Válida	Clase Inválida1	Clase Inválida2
AC-03	Hacer OnClick sobre el Botón 1.1 y todas las acciones relacionadas con la estructura de los objetos auxiliares	Aparecen los objetos auxiliares y tienen la estructura especificada en el Anexo1	No aparecen los objetos auxiliares	Aparecen los objetos auxiliares pero no tienen la estructura especificada en el Anexo1

Nota: Este caso de prueba una vez ejecutado sobre los objetos auxiliares relacionados con una determinada media, cambiará pues no será necesario verificar la estructura de los objetos relacionados con otras medias del mismo tipo, pues es el mismo objeto referenciado en diferentes pantallas, es decir solo se verificaría su presencia. En el caso de mostrar un documento, si es necesario que aparezca un objeto deslizador, se verificará de dicho objeto su presencia y estructura, ya cuando deba ocurrir lo mismo con otro documento, solo se verificará la presencia de dicho objeto, en este tipo de caso de prueba.

### 2.4.2.9 Caso de prueba AC 04

- No. del Caso: AC-04
- Fecha y Usuario de creación: 10/05/07:5.55 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad de los objetos auxiliares que se deben mostrar en el área de contenidos.
- Flujo Central: En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú hacer clic izquierdo sobre el primer botón de arriba hacia abajo.
- Duración del Caso: 10 minutos
- Prioridad: 9

No. CP	Entrada	Clase Válida	Clase Inválida
AC-04	Realizar las acciones especificadas en el Anexo1 sobre los objetos auxiliares para la ejecución de su funcionalidad	Los objetos auxiliares cuentan con la funcionalidad especificada en el Anexo1	Los objetos auxiliares no cuentan con la funcionalidad especificada en el Anexo1

### 2.4.2.10 Caso de prueba ANM 06

- No. del Caso: ANM-06
- Fecha y Usuario de creación: 10/05/07:6.05 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la presencia y estructura del Botón 1.2
- Flujo Central: En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú observar el segundo botón de arriba hacia abajo.
- Duración del Caso: 5 minutos
- Prioridad: 10

No. CP	Entrada	Clase Válida	Clase Inválida1	Clase Inválida2
ANM-06	Hacer MouseOver sobre el Botón 1 y luego sobre el Botón 1.2	Aparece el Botón 1.2 y tiene la estructura especificada en el Anexo1	No aparece el Botón 1.2	Aparece el Botón 1.2 pero no tiene la estructura especificada en el Anexo1

#### 2.4.2.11 Caso de prueba ANM 07

- No. del Caso: ANM-07
- Fecha y Usuario de creación: 10/05/07:6.14 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad del Botón 1.2
- Flujo Central: En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú hacer MouseOver sobre el segundo botón de arriba hacia abajo.
- Duración del Caso: 5 minutos
- Prioridad: 11

No. CP	Entrada	Clase Válida	Clase Inválida
ANM-07	Hacer MouseOver sobre el Botón 1.2	El Botón 1.2 cuenta con la funcionalidad especificada en el Anexo1	El Botón 1.2 no cuenta con la funcionalidad especificada en el Anexo1

#### 2.4.2.12 Caso de prueba ANM 08

- No. del Caso: ANM-08
- Fecha y Usuario de creación: 10/05/07:6.18 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la presencia y estructura del Botón 1.2.1
- Flujo Central: En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú hacer MouseOver sobre el segundo botón de arriba hacia abajo, al desplegarse el segundo menú observar el primer botón de arriba hacia abajo.
- Duración del Caso: 5 minutos



- Prioridad: 12

No. CP	Entrada	Clase Válida	Clase Inválida1	Clase Inválida2
ANM-08	Hacer MouseOver sobre el Botón 1.2 y luego sobre el 1.2.1	Aparece el Botón 1.2.1 y tiene la estructura especificada en el Anexo1	No aparece el Botón 1.2.1	Aparece el Botón 1.2.1 pero no tiene la estructura especificada en el Anexo1

#### 2.4.2.13 Caso de prueba ANM 09

- No. del Caso: ANM-09
- Fecha y Usuario de creación: 10/05/07:6.20 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad del Botón 1.2.1
- Flujo Central: En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú, hacer MouseOver sobre el segundo botón de arriba hacia abajo y luego OnClick sobre el Botón 1.2.1, es decir el primero de arriba hacia abajo del menú que comienza al lado del Botón 1.2.
- Duración del Caso: 5 minutos
- Prioridad: 13

No. CP	Entrada	Clase Válida	Clase Inválida
ANM-09	Hacer OnClick sobre el Botón 1.2.1	El Botón 1.2.1 cuenta con la funcionalidad especificada en el Anexo1	El Botón 1.2.1 no cuenta con la funcionalidad especificada en el Anexo1

#### 2.4.2.14 Caso de prueba AC 05

- No. del Caso: AC-05
- Fecha y Usuario de creación: 10/05/07:9.50 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Validar la calidad del documento mostrado en el área de contenidos.

- Flujo Central: En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú hacer MouseOver sobre el segundo botón de arriba hacia abajo y luego dar clic izquierdo sobre el primer botón de arriba hacia abajo que aparece en el último menú desplegado.
- Duración del Caso: 15 minutos
- Prioridad: 14

No. CP	Entrada	Clase Válida	Clase Inválida
AC-05	Hacer OnClick sobre el Botón 1.2.1	El documento tiene una calidad aceptable.	El documento no tiene una calidad aceptable.

En caso de ser necesario el uso de objetos que faciliten la visualización y navegación sobre el texto, tener en cuenta los siguientes casos de prueba.

#### 2.4.2.15 Caso de prueba AC 06

- No. del Caso: AC-06
- Fecha y Usuario de creación: 10/05/07:9.55 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la presencia de los objetos auxiliares que se deben mostrar en el área de contenidos.
- Flujo Central: En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú hacer MouseOver sobre el segundo botón de arriba hacia abajo y luego dar clic izquierdo sobre el primer botón de arriba hacia abajo que aparece en el último menú desplegado.
- Duración del Caso: 5 minutos
- Prioridad: 15

No. CP	Entrada	Clase Válida	Clase Inválida
AC-06	Hacer OnClick sobre el Botón 1.2.1	Aparecen los objetos auxiliares	No aparecen los objetos auxiliares

#### 2.4.2.16 Caso de prueba AC 07

- No. del Caso: AC-07
- Fecha y Usuario de creación: 10/05/07:10.00 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad de los objetos auxiliares que se deben mostrar en el área de contenidos.
- Flujo Central En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú hacer MouseOver sobre el segundo botón de arriba hacia abajo y luego dar clic izquierdo sobre el primer botón de arriba hacia abajo que aparece en el último menú desplegado.
- Duración del Caso: 10 minutos
- Prioridad: 16

No. CP	Entrada	Clase Válida	Clase Inválida
AC-07	Realizar las acciones especificadas en el Anexo1 sobre los objetos auxiliares para la ejecución de su funcionalidad	Los objetos auxiliares cuentan con la funcionalidad especificada en el Anexo1	Los objetos auxiliares no cuentan con la funcionalidad especificada en el Anexo1

#### 2.4.2.17 Caso de prueba ANM 10

- No. del Caso: ANM-10
- Fecha y Usuario de creación: 10/05/07:10.08 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la presencia y estructura del Botón 1.2.2
- Flujo Central: En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú hacer MouseOver sobre el segundo botón de arriba hacia abajo, al desplegarse el segundo menú observar el segundo botón de arriba hacia abajo.
- Duración del Caso: 5 minutos
- Prioridad: 17

No. CP	Entrada	Clase Válida	Clase Inválida1	Clase Inválida2
ANM-10	Hacer MouseOver sobre el Botón 1.2 y luego sobre el 1.2.2	Aparece el Botón 1.2.2 y tiene la estructura especificada en el Anexo1	No aparece el Botón 1.2.2	Aparece el Botón 1.2.2 pero no tiene la estructura especificada en el Anexo1

#### 2.4.2.18 Caso de prueba ANM 11

- No. del Caso: ANM-11
- Fecha y Usuario de creación: 10/05/07:10.12 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad del Botón 1.2.2
- Flujo Central: En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú, hacer MouseOver sobre el segundo botón de arriba hacia abajo y luego OnClick sobre el Botón 1.2.2, es decir el segundo de arriba hacia abajo del menú que comienza al lado del Botón 1.2.
- Duración del Caso: 5 minutos
- Prioridad: 18

No. CP	Entrada	Clase Válida	Clase Inválida
ANM-11	Hacer OnClick sobre el Botón 1.2.2	El Botón 1.2.2 cuenta con la funcionalidad especificada en el Anexo1	El Botón 1.2.2 no cuenta con la funcionalidad especificada en el Anexo1

#### 2.4.2.19 Caso de prueba AC 08

- No. del Caso: AC-08
- Fecha y Usuario de creación: 10/05/07:10.15 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Validar la calidad de la imagen mostrada en el área de contenidos.
- Flujo Central: En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú hacer MouseOver sobre el segundo botón de

arriba hacia abajo y luego dar clic izquierdo sobre el segundo botón de arriba hacia abajo que aparece en el último menú desplegado.

- Duración del Caso: 15 minutos
- Prioridad: 19

No. CP	Entrada	Clase Válida	Clase Inválida
AC-08	Hacer OnClick sobre el Botón 1.2.2	La imagen tiene una calidad aceptable.	La imagen no tiene una calidad aceptable.

En caso de ser necesario el uso de objetos que faciliten la visualización y navegación sobre la imagen, tener en cuenta los siguientes casos de prueba.

#### 2.4.2.20 Caso de prueba AC 09

- No. del Caso: AC-09
- Fecha y Usuario de creación: 10/05/07:10.18 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la presencia y la estructura de los objetos auxiliares que se deben mostrar en el área de contenidos.
- Flujo Central: En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú hacer MouseOver sobre el segundo botón de arriba hacia abajo y luego dar clic izquierdo sobre el segundo botón de arriba hacia abajo que aparece en el último menú desplegado.
- Duración del Caso: 10 minutos
- Prioridad: 20

No. CP	Entrada	Clase Válida	Clase Inválida1	Clase Inválida2
AC-09	Hacer OnClick sobre el Botón 1.2.2 y todas las acciones relacionadas con la estructura de los objetos auxiliares	Aparecen los objetos auxiliares y tienen la estructura especificada en el Anexo1	No aparecen los objetos auxiliares	Aparecen los objetos auxiliares pero no tienen la estructura especificada en el Anexo1

Nota: Este caso de prueba una vez ejecutado sobre los objetos auxiliares relacionados con una determinada media, cambiará pues no será necesario verificar la estructura de los objetos relacionados con otras medias del mismo tipo, pues es el mismo objeto referenciado en

diferentes pantallas, es decir solo se verificaría su presencia. En el caso de mostrar una imagen, si es necesario que aparezca un objeto tope hacia adelante, se verificará de dicho objeto su presencia y estructura, ya cuando deba ocurrir lo mismo con otra imagen, solo se verificará la presencia de dicho objeto, en este tipo de caso de prueba.

#### 2.4.2.21 Caso de prueba AC 10

- No. del Caso: AC-10
- Fecha y Usuario de creación: 10/05/07:10.20 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad de los objetos auxiliares que se deben mostrar en el área de contenidos.
- Flujo Central En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú hacer MouseOver sobre el segundo botón de arriba hacia abajo y luego dar clic izquierdo sobre el segundo botón de arriba hacia abajo que aparece en el último menú desplegado.
- Duración del Caso: 10 minutos
- Prioridad: 21

No. CP	Entrada	Clase Válida	Clase Inválida
AC-10	Realizar las acciones especificadas en el Anexo1 sobre los objetos auxiliares para la ejecución de su funcionalidad	Los objetos auxiliares cuentan con la funcionalidad especificada en el Anexo1	Los objetos auxiliares no cuentan con la funcionalidad especificada en el Anexo1

#### 2.4.2.22 Caso de prueba ANM 12

- No. del Caso: ANM-12
- Fecha y Usuario de creación: 10/05/07:10.33 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la presencia y estructura del Botón 1.2.3

- Flujo Central: En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú hacer MouseOver sobre el segundo botón de arriba hacia abajo, al desplegarse el segundo menú observar el tercer botón de arriba hacia abajo.
- Duración del Caso: 5 minutos
- Prioridad: 22

No. CP	Entrada	Clase Válida	Clase Inválida1	Clase Inválida2
ANM-12	Hacer MouseOver sobre el Botón 1.2 y luego sobre el 1.2.3	Aparece el Botón 1.2.3 y tiene la estructura especificada en el Anexo1	No aparece el Botón 1.2.3	Aparece el Botón 1.2.3 pero no tiene la estructura especificada en el Anexo1

#### 2.4.2.23 Caso de prueba ANM 13

- No. del Caso: ANM-13
- Fecha y Usuario de creación: 10/05/07:10.36 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad del Botón 1.2.3
- Flujo Central: En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú, hacer MouseOver sobre el segundo botón de arriba hacia abajo y luego OnClick sobre el Botón 1.2.3, es decir el tercero de arriba hacia abajo del menú que comienza al lado del Botón 1.2.
- Duración del Caso: 5 minutos
- Prioridad: 23

No. CP	Entrada	Clase Válida	Clase Inválida
ANM-13	Hacer OnClick sobre el Botón 1.2.3	El Botón 1.2.3 cuenta con la funcionalidad especificada en el Anexo1	El Botón 1.2.3 no cuenta con la funcionalidad especificada en el Anexo1

#### 2.4.2.24 Caso de prueba AC 11

- No. del Caso: AC-11

- Fecha y Usuario de creación: 10/05/07:10.39 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Evaluar la calidad de la imagen mostrada en el área de contenidos.
- Flujo Central: En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú hacer MouseOver sobre el segundo botón de arriba hacia abajo y luego dar clic izquierdo sobre el tercer botón de arriba hacia abajo que aparece en el último menú desplegado.
- Duración del Caso: 15 minutos
- Prioridad: 24

No. CP	Entrada	Clase Válida	Clase Inválida
AC-11	Hacer OnClick sobre el Botón 1.2.3	La imagen tiene una calidad aceptable.	La imagen no tiene una calidad aceptable.

En caso de ser necesario el uso de objetos que faciliten la visualización y navegación sobre la imagen, tener en cuenta los siguientes casos de prueba.

#### 2.4.2.25 Caso de prueba AC 12

- No. del Caso: AC-12
- Fecha y Usuario de creación: 10/05/07:10.40 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la presencia de los objetos auxiliares que se deben mostrar en el área de contenidos.
- Flujo Central: En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú hacer MouseOver sobre el segundo botón de arriba hacia abajo y luego dar clic izquierdo sobre el tercer botón de arriba hacia abajo que aparece en el último menú desplegado.
- Duración del Caso: 5 minutos
- Prioridad: 25



No. CP	Entrada	Clase Válida	Clase Inválida
AC-12	Hacer OnClick sobre el Botón 1.2.3	Aparecen los objetos auxiliares	No aparecen los objetos auxiliares

#### 2.4.2.26 Caso de prueba AC 13

- No. del Caso: AC-13
- Fecha y Usuario de creación: 10/05/07:10.42 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad de los objetos auxiliares que se deben mostrar en el área de contenidos.
- Flujo Central: En el área de navegación, el primer botón de izquierda a derecha, hacer MouseOver sobre él, al desplegarse el menú hacer MouseOver sobre el segundo botón de arriba hacia abajo y luego dar clic izquierdo sobre el tercer botón de arriba hacia abajo que aparece en el último menú desplegado.
- Duración del Caso: 10 minutos
- Prioridad: 26

No. CP	Entrada	Clase Válida	Clase Inválida
AC-13	Realizar las acciones especificadas en el Anexo1 sobre los objetos auxiliares para la ejecución de su funcionalidad	Los objetos auxiliares cuentan con la funcionalidad especificada en el Anexo1	Los objetos auxiliares no cuentan con la funcionalidad especificada en el Anexo1

#### 2.4.2.27 Caso de prueba ANM 14

- No. del Caso: ANM-14
- Fecha y Usuario de creación: 11/05/07:12.05 a.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la presencia y estructura del Botón 5

- Flujo Central: Al cargar la pantalla principal, observar en el área de navegación el quinto botón de izquierda a derecha.
- Duración del Caso: 5 minutos
- Prioridad: 27

No. CP	Entrada	Clase Válida	Clase Inválida1	Clase Inválida2
ANM-14	Mostrar la pantalla principal	Aparece el Botón 5 en el área de navegación y tiene la estructura especificada en el Anexo1	No aparece el Botón 5 en el área de navegación	Aparece el Botón 5 en el área de navegación pero no tiene la estructura especificada en el Anexo1

#### 2.4.2.28 Caso de prueba ANM 15

- No. del Caso: ANM-15
- Fecha y Usuario de creación: 11/05/07:12.11 a.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad del Botón 5
- Flujo Central: Al cargar la pantalla principal, en el área de navegación, hacer clic izquierdo sobre el quinto botón de izquierda a derecha.
- Duración del Caso: 5 minutos
- Prioridad: 28

No. CP	Entrada	Clase Válida	Clase Inválida
ANM-15	Hacer OnClick sobre el Boton5	El Botón 5 cuenta con la funcionalidad especificada en el Anexo1	El Botón 5 no cuenta con la funcionalidad especificada en el Anexo1

#### 2.4.2.29 Caso de prueba AC 14

- No. del Caso: AC-14
- Fecha y Usuario de creación: 10/05/07:12.25 a.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad de los objetos auxiliares que se deben mostrar en el área de contenidos.

- Flujo Central: Al cargar la pantalla principal, en el área de navegación, hacer clic izquierdo sobre el quinto botón de izquierda a derecha.
- Duración del Caso: 10 minutos
- Prioridad: 29

No. CP	Entrada	Clase Válida	Clase Inválida
AC-14	Realizar las acciones especificadas en el Anexo1 sobre los objetos auxiliares para la ejecución de su funcionalidad	Los objetos auxiliares cuentan con la funcionalidad especificada en el Anexo1	Los objetos auxiliares no cuentan con la funcionalidad especificada en el Anexo1

#### 2.4.2.30 Caso de prueba AC 15

- No. del Caso: AC-15
- Fecha y Usuario de creación: 11/05/07:4.50 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la estructura de las imágenes en miniatura mostradas en el área de contenidos.
- Flujo Central: Al cargar la pantalla principal, en el área de navegación, hacer clic izquierdo sobre el quinto botón de izquierda a derecha.
- Duración del Caso: 20 minutos
- Prioridad: 31

No. CP	Entrada	Clase Válida	Clase Inválida1
AC-15	Hacer OnClick sobre el Botón 5	Las 36 imágenes en miniatura tienen la estructura especificada en el Anexo1	Las imágenes no tienen la estructura especificada en el Anexo1

#### 2.4.2.31 Caso de prueba AC 16

- No. del Caso: AC-16
- Fecha y Usuario de creación: 11/05/07:4.54 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:

- Descripción del Caso de Prueba: Evaluar la calidad de las imágenes en miniatura mostradas en el área de contenidos.
- Flujo Central: Al cargar la pantalla principal, en el área de navegación, hacer clic izquierdo sobre el quinto botón de izquierda a derecha.
- Duración del Caso: 20 minutos
- Prioridad: 31

No. CP	Entrada	Clase Válida	Clase Inválida
AC-16	Hacer OnClick sobre el Botón 5	Las imágenes tienen una calidad aceptable.	Las imágenes no tienen una calidad aceptable.

#### 2.4.2.32 Caso de prueba AC 17

- No. del Caso: AC-17
- Fecha y Usuario de creación: 11/05/07:5.03 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad de las imágenes en miniatura que se deben mostrar en el área de contenidos.
- Flujo Central: Al cargar la pantalla principal, en el área de navegación, hacer clic izquierdo sobre el quinto botón de izquierda a derecha.
- Duración del Caso: 20 minutos
- Prioridad: 32

No. CP	Entrada	Clase Válida	Clase Inválida
AC-17	Hacer OnClick sobre cada imagen mostrada en el área de contenido	Las imágenes cuentan con la funcionalidad especificada en el Anexo1	Las imágenes no cuentan con la funcionalidad especificada en el Anexo1

#### 2.4.2.33 Caso de prueba AC 18

- No. del Caso: AC-18
- Fecha y Usuario de creación: 11/05/07:5.06 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:

- Descripción del Caso de Prueba: Evaluar la calidad de la imagen que se debe mostrar en el área de contenidos al hacer OnClick sobre la primera imagen en miniatura.
- Flujo Central: Al cargar la pantalla principal, en el área de navegación, hacer clic izquierdo sobre el quinto botón de izquierda a derecha y luego sobre la primera imagen en miniatura mostrada de arriba hacia abajo y de izquierda a derecha.
- Duración del Caso: 15 minutos
- Prioridad: 33

No. CP	Entrada	Clase Válida	Clase Inválida
AC-18	Hacer OnClick sobre la primera imagen	La imagen tiene una calidad aceptable.	La imagen no tiene una calidad aceptable.

#### 2.4.2.34 Caso de prueba AC 19

- No. del Caso: AC-19
- Fecha y Usuario de creación: 11/05/07:5.11 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la presencia de los objetos auxiliares que se deben mostrar en el área de contenidos.
- Flujo Central: Al cargar la pantalla principal, en el área de navegación, hacer clic izquierdo sobre el quinto botón de izquierda a derecha y luego sobre la primera imagen en miniatura mostrada de arriba hacia abajo y de izquierda a derecha.
- Duración del Caso: 5 minutos
- Prioridad: 34

No. CP	Entrada	Clase Válida	Clase Inválida
AC-19	Hacer OnClick sobre la primera imagen	Aparecen los objetos auxiliares	No aparecen los objetos auxiliares

#### 2.4.2.35 Caso de prueba AC 20

- No. del Caso: AC-20
- Fecha y Usuario de creación: 11/05/07:5.14 p.m. Jorge Antonio Gómez

- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad de los objetos auxiliares que se deben mostrar en el área de contenidos.
- Flujo Central: Al cargar la pantalla principal, en el área de navegación, hacer clic izquierdo sobre el quinto botón de izquierda a derecha y luego sobre la primera imagen en miniatura mostrada de arriba hacia abajo y de izquierda a derecha.
- Duración del Caso: 10 minutos
- Prioridad: 35

No. CP	Entrada	Clase Válida	Clase Inválida
AC-20	Realizar las acciones especificadas en el Anexo1 sobre los objetos auxiliares para la ejecución de su funcionalidad	Los objetos auxiliares cuentan con la funcionalidad especificada en el Anexo1	Los objetos auxiliares no cuentan con la funcionalidad especificada en el Anexo1

### 2.4.3 Prueba de validación 3.01

- Nombre de la Prueba: Pruebas al Panel de Servicios.
- Fecha y Usuario de creación: 11/05/07:6.15 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Duración de la Prueba: 120 minutos.
- Descripción de la prueba: Este tipo de prueba va dirigido a la validación de los objetos, sus características y relaciones.

#### 2.4.3.1 Caso de prueba PS 01

- No. del Caso: PS-01
- Fecha y Usuario de creación: 11/05/07:6.17 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la presencia y estructura del Panel de Servicios.
- Flujo Central: Ejecutar la aplicación y esperar que cargue la pantalla principal
- Duración del Caso: 10 minutos

- Prioridad: 1

No. CP	Entrada	Clase Válida	Clase Inválida1	Clase Inválida2
PS-01	Mostrar la pantalla principal	Aparece el panel de servicios y tiene la estructura especificada en el Anexo1	No aparece el panel de servicios	Aparece el panel de servicios pero no tiene la estructura especificada en el Anexo1

#### 2.4.3.2 Caso de prueba PS 02

- No. del Caso: PS-02
- Fecha y Usuario de creación: 12/05/07:2.41 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la presencia y estructura del Botón Ayuda
- Flujo Central: Ejecutar la aplicación y esperar que cargue la pantalla principal
- Duración del Caso: 5 minutos
- Prioridad: 2

No. CP	Entrada	Clase Válida	Clase Inválida1	Clase Inválida2
PS-02	Mostrar la pantalla principal y hacer MouseOver sobre el Botón Ayuda	Aparece el Botón Ayuda y tiene la estructura especificada en el Anexo1	No aparece el Botón Ayuda	Aparece el Botón Ayuda pero no tiene la estructura especificada en el Anexo1

#### 2.4.3.3 Caso de prueba PS-03

- No. del Caso: PS-03
- Fecha y Usuario de creación: 12/05/07:2.54 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad del Botón Ayuda.
- Flujo Central: Al cargar la pantalla principal, en el panel de contenidos, hacer clic izquierdo sobre el Botón Ayuda.
- Duración del Caso: 5 minutos
- Prioridad: 3

No. CP	Entrada	Clase Válida	Clase Inválida
PS-03	Hacer clic izquierdo sobre el Botón Ayuda	El Botón Ayuda cuenta con la funcionalidad especificada en el Anexo1	El Botón Ayuda no cuenta con la funcionalidad especificada en el Anexo1

#### 2.4.3.4 Caso de prueba PS 04

- No. del Caso: PS-04
- Fecha y Usuario de creación: 12/05/07:2.59 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la presencia y estructura del Botón Imprimir.
- Flujo Central: Ejecutar la aplicación y esperar que cargue la pantalla principal
- Duración del Caso: 5 minutos
- Prioridad: 4

No. CP	Entrada	Clase Válida	Clase Inválida1	Clase Inválida2
PS-04	Mostrar la pantalla principal y hacer MouseOver sobre el Botón Imprimir	Aparece el Botón Imprimir y tiene la estructura especificada en el Anexo1	No aparece el Botón Imprimir	Aparece el Botón Imprimir pero no tiene la estructura especificada en el Anexo1

#### 2.4.3.5 Caso de prueba PS-05

- No. del Caso: PS-05
- Fecha y Usuario de creación: 12/05/07:3.08 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad del Botón Imprimir.
- Flujo Central: Al cargar la pantalla principal, en el panel de contenidos, hacer clic izquierdo sobre el Botón Imprimir.
- Duración del Caso: 10 minutos
- Prioridad: 5



No. CP	Entrada	Clase Válida	Clase Inválida
PS-05	Hacer clic izquierdo sobre el Botón Imprimir	El Botón Imprimir cuenta con la funcionalidad especificada en el Anexo1	El Botón Imprimir no cuenta con la funcionalidad especificada en el Anexo1

#### 2.4.3.6 Caso de prueba PS 06

- No. del Caso: PS-06
- Fecha y Usuario de creación: 12/05/07:3.11 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la presencia y estructura del Botón Inicio.
- Flujo Central: Ejecutar la aplicación y esperar que cargue la pantalla principal
- Duración del Caso: 5 minutos
- Prioridad: 6

No. CP	Entrada	Clase Válida	Clase Inválida1	Clase Inválida2
PS-06	Mostrar la pantalla principal y hacer MouseOver sobre el Botón Inicio	Aparece el Botón Inicio y tiene la estructura especificada en el Anexo1	No aparece el Botón Inicio	Aparece el Botón Inicio pero no tiene la estructura especificada en el Anexo1

#### 2.4.3.7 Caso de prueba PS-07

- No. del Caso: PS-07
- Fecha y Usuario de creación: 12/05/07:3.17 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad del Botón Inicio.
- Flujo Central: Al cargar la pantalla principal, en el panel de contenidos, hacer clic izquierdo sobre el Botón Inicio.
- Duración del Caso: 5 minutos
- Prioridad: 7

No. CP	Entrada	Clase Válida	Clase Inválida
PS-07	Hacer clic izquierdo sobre el Botón Inicio	El Botón Inicio cuenta con la funcionalidad especificada en el Anexo1	El Botón Imprimir no cuenta con la funcionalidad especificada en el Anexo1

#### 2.4.3.8 Caso de prueba PS 08

- No. del Caso: PS-08
- Fecha y Usuario de creación: 12/05/07:3.34 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la presencia y estructura del Botón1 (Música).
- Flujo Central: Ejecutar la aplicación y esperar que cargue la pantalla principal
- Duración del Caso: 5 minutos
- Prioridad: 8

No. CP	Entrada	Clase Válida	Clase Inválida1	Clase Inválida2
PS-08	Mostrar la pantalla principal y hacer MouseOver sobre el Botón1 (Música)	Aparece el Botón1 (Música) y tiene la estructura especificada en el Anexo1	No aparece el Botón1 (Música)	Aparece el Botón1 (Música) pero no tiene la estructura especificada en el Anexo1

#### 2.4.3.9 Caso de prueba PS-09

- No. del Caso: PS-09
- Fecha y Usuario de creación: 12/05/07:3.37 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad del Botón1 (Música).
- Flujo Central: Al cargar la pantalla principal, en el panel de contenidos, hacer clic izquierdo sobre el Botón1 (Música).
- Duración del Caso: 5 minutos
- Prioridad: 9

No. CP	Entrada	Clase Válida	Clase Inválida
PS-09	Hacer clic izquierdo sobre el Botón1 (Música)	El Botón1 (Música) cuenta con la funcionalidad especificada en el Anexo1	El Botón1 (Música) no cuenta con la funcionalidad especificada en el Anexo1

#### 2.4.3.10 Caso de prueba PS-10

- No. del Caso: PS-10
- Fecha y Usuario de creación: 12/05/07:3.39 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la presencia y estructura del Botón2 (Música).
- Flujo Central: Ejecutar la aplicación, esperar que cargue la pantalla principal y hacer OnClick sobre el Botón2 (Música).
- Duración del Caso: 5 minutos
- Prioridad: 10

No. CP	Entrada	Clase Válida	Clase Inválida1	Clase Inválida2
PS-10	Hacer OnClick sobre el Botón1 (Música) y luego MouseOver sobre el Botón2 (Música)	Aparece el Botón2 (Música) y tiene la estructura especificada en el Anexo1	No aparece el Botón2 (Música)	Aparece el Botón2 (Música) pero no tiene la estructura especificada en el Anexo1

#### 2.4.3.11 Caso de prueba PS-11

- No. del Caso: PS-11
- Fecha y Usuario de creación: 12/05/07:3.47 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad del Botón2 (Música).
- Flujo Central: Al cargar la pantalla principal, en el panel de contenidos, hacer clic izquierdo sobre el Botón1 (Música) y luego sobre el Botón2 (Música).
- Duración del Caso: 5 minutos
- Prioridad: 11

No. CP	Entrada	Clase Válida	Clase Inválida
PS-11	Hacer clic izquierdo sobre el Botón2 (Música)	El Botón2 (Música) cuenta con la funcionalidad especificada en el Anexo1	El Botón2 (Música) no cuenta con la funcionalidad especificada en el Anexo1

#### 2.4.3.12 Caso de prueba PS 12

- No. del Caso: PS-12
- Fecha y Usuario de creación: 12/05/07:3.52 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la presencia y estructura del Botón Minimizar.
- Flujo Central: Ejecutar la aplicación y esperar que cargue la pantalla principal
- Duración del Caso: 5 minutos
- Prioridad: 12

No. CP	Entrada	Clase Válida	Clase Inválida1	Clase Inválida2
PS-12	Mostrar la pantalla principal y hacer MouseOver sobre el Botón Minimizar	Aparece el Botón Minimizar y tiene la estructura especificada en el Anexo1	No aparece el Botón Minimizar	Aparece el Botón Minimizar pero no tiene la estructura especificada en el Anexo1

#### 2.4.3.13 Caso de prueba PS-13

- No. del Caso: PS-13
- Fecha y Usuario de creación: 12/05/07:3.58 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad del Botón Minimizar.
- Flujo Central: Al cargar la pantalla principal, en el panel de contenidos, hacer clic izquierdo sobre el Botón Minimizar.
- Duración del Caso: 5 minutos
- Prioridad: 13

No. CP	Entrada	Clase Válida	Clase Inválida
PS-13	Hacer clic izquierdo sobre el Botón Minimizar	El Botón Minimizar cuenta con la funcionalidad especificada en el Anexo1	El Botón Minimizar no cuenta con la funcionalidad especificada en el Anexo1

#### 2.4.3.14 Caso de prueba PS 14

- No. del Caso: PS-14
- Fecha y Usuario de creación: 12/05/07:4.05 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la presencia y estructura del Botón Salir.
- Flujo Central: Ejecutar la aplicación y esperar que cargue la pantalla principal
- Duración del Caso: 5 minutos
- Prioridad: 14

No. CP	Entrada	Clase Válida	Clase Inválida1	Clase Inválida2
PS-14	Mostrar la pantalla principal y hacer MouseOver sobre el Botón Salir	Aparece el Botón Salir y tiene la estructura especificada en el Anexo1	No aparece el Botón Salir	Aparece el Botón Salir pero no tiene la estructura especificada en el Anexo1

#### 2.4.3.15 Caso de prueba PS-15

- No. del Caso: PS-15
- Fecha y Usuario de creación: 12/05/07:4.10 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad del Botón Salir.
- Flujo Central: Al cargar la pantalla principal, en el panel de contenidos, hacer clic izquierdo sobre el Botón Salir.
- Duración del Caso: 5 minutos
- Prioridad: 15

No. CP	Entrada	Clase Válida	Clase Inválida
PS-15	Hacer clic izquierdo sobre el Botón Salir	El Botón Salir cuenta con la funcionalidad especificada en el Anexo1	El Botón Salir no cuenta con la funcionalidad especificada en el Anexo1

#### 2.4.3.16 Caso de prueba PS-16

- No. del Caso: PS-16
- Fecha y Usuario de creación: 12/05/07:4.15 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la estructura de la ventana de Verificación de Salida.
- Flujo Central: Al cargar la pantalla principal, en el panel de contenidos, hacer clic izquierdo sobre el Botón Salir.
- Duración del Caso: 10 minutos
- Prioridad: 16

No. CP	Entrada	Clase Válida	Clase Inválida
PS-16	Hacer clic izquierdo sobre el Botón Salir	La ventana de verificación tiene la estructura especificada en el Anexo1	La ventana de verificación no tiene la estructura especificada en el Anexo1

#### 2.4.3.17 Caso de prueba PS-17

- No. del Caso: PS-17
- Fecha y Usuario de creación: 12/05/07:4.18 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la estructura de los botones de la ventana de Verificación de Salida.
- Flujo Central: Al cargar la pantalla principal, en el panel de contenidos, hacer clic izquierdo sobre el Botón Salir.
- Duración del Caso: 5 minutos
- Prioridad: 17

No. CP	Entrada	Clase Válida	Clase Inválida
PS-17	Hacer clic izquierdo sobre el Botón Salir	Los botones de la ventana de verificación tienen la estructura especificada en el Anexo1	Los botones no tienen la estructura especificada en el Anexo1

#### 2.4.3.18 Caso de prueba PS-18

- No. del Caso: PS-18
- Fecha y Usuario de creación: 12/05/07:4.23 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la funcionalidad de los botones de la ventana de Verificación de Salida.
- Flujo Central: Al cargar la pantalla principal, en el panel de contenidos, hacer clic izquierdo sobre el Botón Salir.
- Duración del Caso: 10 minutos
- Prioridad: 18

No. CP	Entrada	Clase Válida	Clase Inválida1	Clase Inválida2
PS-18	Hacer clic izquierdo sobre cada uno de los botones	Los botones cuentan con la funcionalidad especificada en el Anexo1	El Botón Si, no cuenta con la funcionalidad especificada en el Anexo1	El Botón No, no cuenta con la funcionalidad especificada en el Anexo1

## 2.5 Plan de Pruebas de Usabilidad

- Tema: PHPU-01
- Descripción: Este plan de pruebas define todas las pruebas y los casos de pruebas relacionados con la usabilidad, específicamente las siguientes características y elementos: estética, consistencia, sistema de navegación y orientación, documentación para el usuario, entre otros aspectos contenidos en [Marqués00] definidos como criterios de calidad y algunos mencionados en la definición de pruebas de interfaces gráficas de usuario, como la verificación de que los nombres de los menús son autoexplicativos y la disponibilidad de ayuda para cada ítem de menú.

- Usuario Responsable: Jorge Antonio Gómez Colombat
- Estado: En diseño
- Fecha de Creación: 25/04/07, 5.35 p.m.
- Usuario de Creación: Jorge Antonio Gómez Colombat
- Tiempo Total de Ejecución: 425 minutos.

### 2.5.1 Prueba de Usabilidad 1.01

- Nombre de la Prueba: Consistencia de la Interfaz de Usuario
- Fecha y Usuario de creación: 25/04/07:5.42 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Duración de la Prueba: 45 minutos
- Descripción de la prueba: Este tipo de prueba va dirigido a evaluar la consistencia de la interfaz de usuario.

#### 2.5.1.1 Caso de prueba CIU 01

- No. del Caso: CIU-01
- Fecha y Usuario de creación: 25/04/07:5.45 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Recorrer todas las áreas de la aplicación y ejecutar todas las acciones posibles en busca de inconsistencia con otros programas o acciones del mismo tipo.
- Flujo Central: Ejecutar la aplicación y esperar que cargue la pantalla principal
- Duración del Caso: 45 minutos
- Prioridad: 1

No. CP	Entrada	Clase Válida	Clase Inválida
CIU-01	Recorrer todas las áreas de la aplicación y ejecutar todas las acciones posibles	La aplicación de cara al usuario es consistente	La aplicación presenta inconsistencias de cara al usuario



### 2.5.2 Prueba de Usabilidad 2.01

- Nombre de la Prueba: Criterios de Calidad
- Fecha y Usuario de creación: 13/05/07:2.20 a.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Duración de la Prueba: 320 minutos
- Descripción de la prueba: Este tipo de prueba va dirigido a verificar la presencia, en el producto, de algunos de los criterios de calidad abordados en [Marqués00].

#### 2.5.2.1 Caso de prueba CC- 01

- No. del Caso: CC-01
- Fecha y Usuario de creación: 13/05/07:2.25 a.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Evaluar la eficacia de la aplicación, teniendo en cuenta que la eficacia es el nivel en que se cumplen los objetivos trazados.
- Flujo Central: Ejecutar la aplicación y esperar que cargue la pantalla principal
- Duración del Caso: 45 minutos
- Prioridad: 1

No. CP	Entrada	Clase Válida	Clase Inválida
CC-01	Recorrer todas las áreas de la aplicación	La aplicación es eficaz totalmente	La aplicación no es del todo eficaz

#### 2.5.2.2 Caso de prueba CC- 02

- No. del Caso: CC-02
- Fecha y Usuario de creación: 13/05/07:6.04 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Evaluar la facilidad de instalación de la aplicación.
- Flujo Central: Ejecutar la aplicación
- Duración del Caso: 10 minutos

- Prioridad: 2

No. CP	Entrada	Clase Válida	Clase Inválida
CC-02	Ejecutar la aplicación	La aplicación es fácil de instalar	La aplicación no es fácil de instalar

### 2.5.2.3 Caso de prueba CC- 03

- No. del Caso: CC-03
- Fecha y Usuario de creación: 13/05/07:6.14 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Evaluar la facilidad de manejo de la aplicación.
- Flujo Central: Ejecutar la aplicación y esperar que cargue la pantalla principal.
- Duración del Caso: 30 minutos
- Prioridad: 3

No. CP	Entrada	Clase Válida	Clase Inválida
CC-03	Recorrer todas las áreas de la aplicación	La aplicación es fácil de manejar	La aplicación no es fácil de manejar

### 2.5.2.4 Caso de prueba CC- 04

- No. del Caso: CC-04
- Fecha y Usuario de creación: 13/05/07:6.19 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Evaluar la parametrización en la aplicación.
- Flujo Central: Ejecutar la aplicación, esperar que cargue la pantalla principal y hacer clic izquierdo sobre el Botón Con Música y Sin Música que se encuentran en el Panel de Servicios.
- Duración del Caso: 5 minutos
- Prioridad: 4

No. CP	Entrada	Clase Válida	Clase Inválida
CC-04	Hacer clic izquierdo sobre el Botón1 (Música) y Botón2 Música.	El efecto sonoro es controlado satisfactoriamente	El efecto sonoro no es controlado del modo esperado.

#### 2.5.2.5 Caso de prueba CC- 05

- No. del Caso: CC-05
- Fecha y Usuario de creación: 13/05/07:9.49 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Evaluar la funcionalidad de la documentación.
- Flujo Central: Ejecutar la aplicación, esperar que cargue la pantalla principal y hacer clic izquierdo sobre el Botón Ayuda que se encuentran en el Panel de Servicios.
- Duración del Caso: 30 minutos
- Prioridad: 5

No. CP	Entrada	Clase Válida	Clase Inválida
CC-05	Hacer clic izquierdo sobre el Botón Ayuda y analizar la documentación.	La documentación es bastante funcional	La documentación no es lo suficiente funcional

#### 2.5.2.6 Caso de prueba CC- 06

- No. del Caso: CC-06
- Fecha y Usuario de creación: 13/05/07:9.57 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Evaluar la calidad del entorno.
- Flujo Central: Ejecutar la aplicación, esperar que cargue la pantalla principal.
- Duración del Caso: 20 minutos
- Prioridad: 6

No. CP	Entrada	Clase Válida	Clase Inválida
CC-06	Recorrer todas las áreas de la aplicación	El entorno de la aplicación tiene una calidad aceptable	El entorno de la aplicación no tiene una calidad aceptable

### 2.5.2.7 Caso de prueba CC- 07

- No. del Caso: CC-07
- Fecha y Usuario de creación: 13/05/07:10.03 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Evaluar la potencia audiovisual de la aplicación.
- Flujo Central: Ejecutar la aplicación, esperar que cargue la pantalla principal.
- Duración del Caso: 20 minutos
- Prioridad: 7

No. CP	Entrada	Clase Válida	Clase Inválida
CC-07	Recorrer todas las áreas de la aplicación	La aplicación cuenta con una buena potencia audiovisual	La aplicación no cuenta con una buena potencia audiovisual

### 2.5.2.8 Caso de prueba CC- 08

- No. del Caso: CC-08
- Fecha y Usuario de creación: 13/05/07:10.06 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Evaluar la organización de los contenidos.
- Flujo Central: Ejecutar la aplicación, esperar que cargue la pantalla principal.
- Duración del Caso: 20 minutos
- Prioridad: 8

No. CP	Entrada	Clase Válida	Clase Inválida
CC-08	Recorrer todas las áreas de la aplicación	Los contenidos están bien organizados	Los contenidos no están organizados correctamente

### 2.5.2.9 Caso de prueba CC- 09

- No. del Caso: CC-09
- Fecha y Usuario de creación: 13/05/07:10.10 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Evaluar el sistema de navegación.

- Flujo Central: Ejecutar la aplicación, esperar que cargue la pantalla principal.
- Duración del Caso: 20 minutos
- Prioridad: 9

No. CP	Entrada	Clase Válida	Clase Inválida
CC-09	Recorrer todas las áreas de la aplicación	El sistema de navegación es bueno.	El sistema de navegación está incompleto

#### 2.5.2.10 Caso de prueba CC- 10

- No. del Caso: CC-10
- Fecha y Usuario de creación: 13/05/07:10.16 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Evaluar la estructura del mapa de navegación.
- Flujo Central: Ir a la dirección: "Documentación\Guión Multimedia\mapas".
- Duración del Caso: 20 minutos
- Prioridad: 10

No. CP	Entrada	Clase Válida	Clase Inválida
CC-10	Analizar los mapas de navegación presentes en la dirección especificada	Los mapas están bien estructurados	Los mapas no están correctamente estructurados

#### 2.5.2.11 Caso de prueba CC- 11

- No. del Caso: CC-11
- Fecha y Usuario de creación: 13/05/07:10.28 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Evaluar facilidad de la entrada de mensajes.
- Flujo Central: Ejecutar la aplicación, esperar que cargue la pantalla principal.
- Duración del Caso: 20 minutos
- Prioridad: 11

No. CP	Entrada	Clase Válida	Clase Inválida
CC-11	Recorrer todas las áreas de la aplicación, introduciendo todos los mensajes posibles	Los mensajes son introducidos fácilmente en el sistema	Los mensajes no son introducidos fácilmente en el sistema

#### 2.5.2.12 Caso de prueba CC- 12

- No. del Caso: CC-12
- Fecha y Usuario de creación: 13/05/07:10.40 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Evaluar la fiabilidad de la aplicación.
- Flujo Central: Ejecutar la aplicación, esperar que cargue la pantalla principal.
- Duración del Caso: 5 minutos
- Prioridad: 12

No. CP	Entrada	Clase Válida	Clase Inválida
CC-12	Ejecutar la aplicación	La ejecución de la aplicación es fiable	La ejecución de la aplicación no es fiable

#### 2.5.2.13 Caso de prueba CC- 13

- No. del Caso: CC-13
- Fecha y Usuario de creación: 13/05/07:10.43 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Evaluar la originalidad de la aplicación.
- Flujo Central: Ejecutar la aplicación, esperar que cargue la pantalla principal.
- Duración del Caso: 10 minutos
- Prioridad: 13

No. CP	Entrada	Clase Válida	Clase Inválida
CC-13	Ejecutar la aplicación y recorrer todas sus áreas	La aplicación es original	La aplicación carece de originalidad

#### 2.5.2.14 Caso de prueba CC- 14

- No. del Caso: CC-14
- Fecha y Usuario de creación: 13/05/07:10.47 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Evaluar la estética de la aplicación.
- Flujo Central: Ejecutar la aplicación, esperar que cargue la pantalla principal.
- Duración del Caso: 40 minutos
- Prioridad: 14

No. CP	Entrada	Clase Válida	Clase Inválida
CC-14	Ejecutar la aplicación y recorrer todas sus áreas	La aplicación tiene una buena estética	La aplicación no tiene una buena estética.

#### 2.5.2.15 Caso de prueba CC- 15

- No. del Caso: CC-15
- Fecha y Usuario de creación: 13/05/07:10.54 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Evaluar la calidad técnica de la documentación.
- Flujo Central: Ejecutar la aplicación, esperar que cargue la pantalla principal.
- Duración del Caso: 20 minutos
- Prioridad: 15

No. CP	Entrada	Clase Válida	Clase Inválida
CC-15	Hacer clic izquierdo sobre el Botón Ayuda y analizar la documentación	La documentación tiene una calidad técnica aceptable	La documentación no tiene una calidad técnica aceptable.

#### 2.5.3 Prueba de Usabilidad 3.01

- Nombre de la Prueba: Interfaz Gráfica de Usuario
- Fecha y Usuario de creación: 17/05/07:4.00 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:

- Duración de la Prueba: 45 minutos
- Descripción de la prueba: Este tipo de prueba va dirigido a analizar algunas cuestiones presentes en la definición de pruebas de IGUs, relacionadas con la usabilidad que no hayan sido referenciadas en los casos de pruebas anteriores.

### 2.5.3.1 Caso de prueba IGU- 01

- No. del Caso: IGU-01
- Fecha y Usuario de creación: 17/05/07:4.05 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Analizar si los nombres de los ítems de menú son autoexplicativos.
- Flujo Central: Ejecutar la aplicación y esperar que cargue la pantalla principal
- Duración del Caso: 30 minutos
- Prioridad: 1

No. CP	Entrada	Clase Válida	Clase Inválida
IGU-01	Analizar todas las etiquetas de los ítems del área de navegación entre módulos	Los nombres de los ítems son autoexplicativos.	Los nombres de los ítems no son autoexplicativos.

### 2.5.3.2 Caso de prueba IGU- 02

- No. del Caso: IGU-02
- Fecha y Usuario de creación: 17/05/07:4.15 p.m. Jorge Antonio Gómez
- Fecha y Usuario de última Modificación:
- Descripción del Caso de Prueba: Verificar la disponibilidad de ayuda por cada ítem de menú.
- Flujo Central: Ejecutar la aplicación y esperar que cargue la pantalla principal.
- Duración del Caso: 10 minutos
- Prioridad: 2



No. CP	Entrada	Clase Válida	Clase Inválida
IGU-02	Hacer clic izquierdo sobre el Botón Ayuda y analizar la documentación o seleccionar un ítem y presionar F1	El documento de ayuda aborda cada ítem de menú o aparece información al presionar F1	El documento de ayuda no aborda cada ítem de menú, ni se muestra información al presionar F1

## 2.6 Conclusiones

En este capítulo se diseñó el plan de pruebas propuesto como parte de la solución al problema planteado en la Introducción de este documento. Este plan está compuesto por tres subplanes, los cuales albergan pruebas y casos de pruebas diseñados para cubrir todas las especificaciones del sistema. Algunos casos de prueba pertenecientes a las Pruebas de Validación 2.01, no están presentes en este capítulo debido a la similitud que tienen con los expuestos en dicho epígrafe. Los resultados arrojados por los casos de prueba al ser ejecutados se muestran en el siguiente capítulo, además de un resumen y análisis del proceso de prueba.

Capítulo  
3

## RESULTADOS DE LAS PRUEBAS

### 3.1 Introducción

En el capítulo anterior se diseñaron más de 70 casos de pruebas, los cuales deben ser ejecutados y sus resultados archivados en el presente capítulo. Para realizar dicha ejecución debe prepararse el ambiente de pruebas, recibir la documentación, la aplicación y verificar la correcta instalación de la misma.

Son diversos los diseños de las estructuras realizadas para archivar los resultados e inconformidades de las pruebas, sin embargo no difieren mucho los elementos que las constituyen.

En el próximo epígrafe se propone una tabla con los datos que se consideraron más significativos de los resultados de las pruebas. En el tercero y cuarto epígrafe un resumen y evaluación del costo del proceso pruebas, respectivamente; y en el último las conclusiones del capítulo.

### 3.2 Resultados de las pruebas

La ejecución de un caso de prueba muestra un determinado resultado que se espera que sea la acción especificada en la documentación; no siempre es así y es por eso que se definen clases válidas e inválidas.

No es necesario detallar un resultado válido o esperado, pues se supone que este sea la validación de un determinado requisito o pauta del sistema. Es probable que surja la necesidad de realizar alguna observación sobre dicho resultado, la cual puede ser archivada en el campo de recomendación, aunque lo correcto sería antes de comenzar a implementar efectuar un correcto

análisis y validación de los requisitos del sistema.

Las respuestas inválidas si deben ser descritas, informar de su localización y recomendar alguna acción para erradicar el fallo. Es recomendable archivar los tiempos de ejecución de cada caso de prueba, pues son fundamentales para realizar un correcto análisis del proceso y futuras planificaciones de pruebas.

La estructura propuesta para almacenar los resultados e inconformidades consta de 8 elementos, 3 de los cuales serán utilizados en caso de aparición de algún defecto en la aplicación tras la ejecución de algún caso de prueba.

Leyenda:

**No.** : Número de la secuencia de casos de prueba.

**SP:** Tema del Subplan de prueba

**CP:** Número del Caso de Prueba

**Resultados:** Según lo observado después de la ejecución del caso de prueba, referenciar la clase de equivalencia que se manifestó.

**t(min):** Tiempo de ejecución en minutos.

**Descripción:** Descripción del defecto o inconformidad.

**Ubicación:** El lugar exacto de la aplicación en que se observó dicho defecto.

**Recomendaciones:** Que acciones se pueden realizar para la erradicación del defecto o inconformidad.

**Tabla 3 Resultados e Inconformidades.**

No.	SP	CP	Resultados	t(min)	Inconformidades		
					Descripción	Ubicación	Recomendación
01	PHPD-01	RID-01	C. Inválida	25	No aparece dentro de la documentación ninguna especificación sobre el uso de algún estándar de documentación, ni la implementación de alguno.	D:\Heredia\ Guión Multimedia	Puede utilizarse algún estándar definido por la universidad, nacional o internacional (ISO/IEC 15910).
02	PHPD-	RID-02	C. Inválida	22	No hay homogeneidad	D:\Heredia\ 	Puede definirse una

	01				en la tipografía de los documentos, incluso aquellos que guardan relación entre sí, ya sea porque están en la misma raíz o por que realizan funciones parecidas.	Guión Multimedia	tipografía de letra y otros aspectos de la documentación y aplicarlo a todos los documentos referentes a la aplicación.
03	PHPD-01	RID-03	C. Inválida	110	Es incompleta la información contenida en la documentación pues faltan elementos por definir acerca del software y no están suficientemente detallados algunos elementos.	D:\Heredia\Guión Multimedia	Utilizar una guía de desarrollo de hipertexto para guiar el desarrollo, puede ser (Marqués, 2000) o (Valencia, 1997).
04	PHPD-01	RID-04	C. Inválida	10	La documentación no esta lo suficiente organizada pues es muy difícil encontrar un determinado documento o elemento, pues el documento no es lo suficientemente preciso.	D:\Heredia\Guión Multimedia	Definir tipos de información y hacer una categorización exacta, crear un documento de ayuda en caso de ser necesario, y un índice bien preciso.
05	PHPD-01	RID-05	C. Inválida	18	La documentación no es consistente con relación a la aplicación, debido a que existen diferencias entre los mapas de navegación y la barra de navegación del sistema.	D:\Heredia\Guión Multimedia\mapas	Establecer criterios para establecer el orden de aparición de la información (cronológico, alfabético).
06	PHPD-01	RID-06	C. Inválida	35	El documento LEEME funciona como relación de contenidos del Guión Multimedia, este no es totalmente preciso pues no menciona la carpeta de mapas, ni el documento Pantalla 2.1 Introduccion presente en la carpeta pantallas	D:\Heredia\Guión Multimedia\LEEME	Incorporar la información sobre la carpeta mapas y su contenido y el documento Pantalla 2.1 Introduccion presente en la carpeta pantallas \DocPersonales

					\DocPersonales		
07	PHPV-01	ATPD-01	C. Inválida	1	20	La aplicación no muestra en ninguna de sus interfaces información referente al título del programa ni al tópico, documento o imagen mostrado en el área de contenidos.	Esta área debe mostrarse en todas las vistas de la aplicación  Modificar la estructura de la aplicación y crear un área para mostrar el título del programa, encima del área de navegación y otra para el tópico e ítem actual, entre el área de navegación y la de contenidos, o un lado del área de contenidos
08	PHPV-01	ANM-01	C. Válida		35		
09	PHPV-01	AC-01	C. Válida		15		
10	PHPV-01	ANM-02	C. Válida		2		
11	PHPV-01	ANM-03	C. Válida		1		
12	PHPV-01	ANM-04	C. Válida		1		
13	PHPV-01	ANM-05	C. Válida		5		
14	PHPV-01	AC-02	C. Inválida		10	Tiene una falta ortográfica, la palabra usted no debe de escribirse con letra inicial mayúscula	En la penúltima línea del último párrafo  Corregir la falta ortográfica
15	PHPV-01	AC-03	C. Válida		3		
16	PHPV-01	AC-04	C. Válida		3		
17	PHPV-01	ANM-06	C. Válida		3		
18	PHPV-01	ANM-07	C. Válida		2		
19	PHPV-01	ANM-08	C. Válida		2		
20	PHPV-01	ANM-09	C. Inválida		12	El documento no se carga completamente.	Ultima oración

21	PHPV-01	AC-05	C. Inválida	25	El texto esta muy mal distribuido diversas medidas de espaciados y datos en forma de tabla completamente desorganizados.	Texto integro	
22	PHPV-01	AC-06	C. Válida	1			
23	PHPV-01	AC-07	C. Válida	1			
24	PHPV-01	ANM-10	C. Válida	1			
25	PHPV-01	ANM-11	C. Válida	3			
26	PHPV-01	AC-08	C. Inválida	10	El texto no puede ser bien leído en ninguna de las imágenes de la secuencia.	Todas las imágenes de la secuencia	Llevar el texto del gráfico a un *.rtf y convertirlo a *.xml
27	PHPV-01	AC-09	C. Inválida	2	No se muestran los botones tope hacia adelante y hacia atrás.	Parte inferior de la imagen	
28	PHPV-01	AC-10	C. Válida	4			
29	PHPV-01	ANM-12	C. Válida	1			
30	PHPV-01	ANM-13	C. Válida	1			
31	PHPV-01	AC-11	C. Inválida	4	El texto no puede ser bien leído en ninguna de las imágenes de la secuencia.	Todas las imágenes de la secuencia	Llevar el texto del gráfico a un *.rtf y convertirlo a *.xml
32	PHPV-01	AC-12	C. Inválida	1	No se muestran los botones tope hacia adelante y hacia atrás.	Parte inferior de la imagen	
33	PHPV-01	AC-13	C. Válida	1			
34	PHPV-01	ANM-14	C. Válida	1			
35	PHPV-01	ANM-15	C. Válida	3			
36	PHPV-01	AC-14	C. Válida	10			
37	PHPV-	AC-15	C. Inválida	20			

	01						
38	PHPV-01	AC-16	C. Válida	8			
39	PHPV-01	AC-17	C. Inválida	7	La imagen 18 (Ambrosio Lamadriz y Heredia) no se muestra	En la sexta fila, la tercera	
40	PHPV-01	AC-18	C. Inválida	3	La imagen esta algo dañada y no se pueden distinguir algunos rasgos	Parte inferior y rostro de la persona	
41	PHPV-01	AC-19	C. Inválida	2	No se muestran la lupa ni el botón tope hacia atrás.	Parte inferior de la imagen	
42	PHPV-01	AC-20	C. Inválida	1	El hacer OnClick sobre el botón tope hacia delante se cargan en el área de contenidos las imágenes en miniatura, en vez de mostrar la última imagen de la galería.	Parte inferior de la imagen, ultimo icono de izquierda a derecha.	
43	PHPV-01	PS-01	C. Inválida	20	Solo tiene tres iconos de cinco	Parte inferior de la interfaz, a la derecha	Incluir los dos botones que faltan pues de gran importancia.
44	PHPV-01	PS-02	C. Inválida1	2	No se muestra el botón Ayuda	Panel de Servicios	
45	PHPV-01	PS-03	No aplicado				
46	PHPV-01	PS-04	C. Inválida1	1	No se muestra el botón Imprimir	Panel de Servicios	
47	PHPV-01	PS-05	No aplicado				
48	PHPV-01	PS-06	C. Inválida2	3	Se mantiene activo en la pagina inicial	Panel de Servicios	
49	PHPV-01	PS-07	C. Válida	1			
50	PHPV-01	PS-08	C. Inválida2	2	El texto mostrado no es el especificado	Panel de Servicios	El texto podría ser Detener o Pausar
51	PHPV-01	PS-09	C. Válida	1			
52	PHPV-01	PS-10	C. Inválida2	2	El texto mostrado no es el especificado	Panel de Servicios	El texto podría ser Reproducir

53	PHPV-01	PS-11	C. Válida	1			
54	PHPV-01	PS-12	C. Inválida	1	No se muestra el botón Minimizar	Panel de Servicios	
55	PHPV-01	PS-13	No aplicado				
56	PHPV-01	PS-14	C. Inválida	1	El texto mostrado no es el especificado	Panel de Servicios	
57	PHPV-01	PS-15	C. Válida	2			
58	PHPV-01	PS-16	C. Válida	2			
59	PHPV-01	PS-17	C. Válida	1			
60	PHPV-01	PS-18	C. Válida	6			
61	PHPU-01	CIU-01	C. Inválida	35	En Galería al hacer OnClick sobre cualquier imagen en miniatura, se muestra una imagen y algunos iconos, uno de ellos es el de tope hacia delante, utilizado como vínculo a la interfaz inicial de Galería. El botón Inicio se encuentra activo en la misma pantalla inicial, lo mismo sucede con el Botón5 o Galería que se mantiene activo en la pantalla inicial de esta misma sección. Sucede lo mismo con todos los botones que al ser seleccionados se muestra una determinada media en el área de contenidos, permanecen activos.	Botón Inicio. Botón 5 o Galería. Botón tope hacia delante de la Sección Galería. Todos los botones del área de navegación que al ser seleccionados se muestra una determinada media en el área de contenidos	En el caso del Botón tope hacia delante de la Sección Galería, modificar la función por la de cargar la última media de la secuencia. En el resto de los casos deshabilitarlos mientras esté presente el resultado de la ejecución de cada uno de ellos.
62	PHPU-01	CC-01	C. Inválida	10	No son correctamente percibidos muchos documentos por el	D:\Heredia\Guión Multimedia\	



					usuario, ni la estructura de la aplicación tiene todos los elementos especificados.	Pautas Generales	
63	PHPU-01	CC-02	C. Inválida	10	El proceso de instalación y desinstalación no está automatizado. No se contemplan y solucionan posibles problemas de instalación, ni incluye todo el software necesario para la instalación.		
64	PHPU-01	CC-03	C. Inválida	15	El entorno audiovisual es claro, sencillo e intuitivo, pero no existen instrucciones para la utilización de la aplicación.		Crear un manual para el usuario.
65	PHPU-01	CC-04	C. Válida	2			
66	PHPU-01	CC-05	No aplicado				
67	PHPU-01	CC-06	C. Válida	8			
68	PHPU-01	CC-07	C. Válida	5			
69	PHPU-01	CC-08	C. Inválida	15	No se sigue un orden lógico (cronológico, alfabético, etc.) de los eventos.	En los tópicos de: Documentos Personales, Creación Literaria y Ediciones.	Organizar los contenidos por un orden lógico.
70	PHPU-01	CC-09	C. Inválida	8	La aplicación no cuenta con instrumentos para facilitar la orientación del usuario y el acceso a las prestaciones del programa como:		Incluir algunos de estos instrumentos en la aplicación.

					índices temáticos, índices históricos, mapas de posición, información de la media que se esté mostrando, etc.		
71	PHPU-01	CC-10	C. Inválida	15	Omisión y desorganización de los ítems y contenido, ausencia de algunos mapas.	D:\Heredia\ Guión Multimedia\ mapas Omisión del ítem Introducción , en Documentos , (general-mapa) y Relación Ediciones, en Ediciones (ediciones-mapa). Desorganización en todos los mapas.	Reestructurar los mapas y construir los que faltan.
72	PHPU-01	CC-11	C. Válida	20			
73	PHPU-01	CC-12	C. Inválida	5	No muestra la animación de presentación.	Ejecutar el archivo D:\Heredia\ Multimedia\ presentación.swf	Incluir dicha animación.
74	PHPU-01	CC-13	C. Válida	10			
75	PHPU-01	CC-14	C. Inválida	15	Las imágenes y textos gráficos son cargados en diferentes contenedores (carteros y marcos).	Seleccionar cada ítem que muestre imágenes.	Definir un solo tipo de contenedor de imágenes.
76	PHPU-01	CC-15	No aplicado				
77	PHPU-	IGU-	C. Válida	15			

	01	01					
78	PHPU-01	IGU-02	C. Inválida	5	No se muestra el botón Ayuda, ni ninguna información al presionar F1 en cualquier área de la aplicación.	Panel de Servicios	Mostrar el botón Ayuda y que sea funcional.

Los defectos encontrados deben ser clasificados en dependencia de que tanto afecta a la funcionalidad o usabilidad de la aplicación. Los valores que pueden tomar son:

Valor	Significado
Crítico	El defecto causa un fallo completo del sistema, subsistema, o de un módulo dentro del sistema.
Importante	El defecto no causa un fallo, pero causa que el sistema se comporte de forma incorrecta, incompleta o con resultados inconsistentes, o comprometa la usabilidad del sistema.
Mejora	El defecto es estético, es una extensión, o es el resultado de una no-conformidad con algún estándar.

**Tabla 4 Clasificación de los defectos encontrados**

No.	SP	CP	Valor	No.	SP	CP	Valor
01	PHPD-01	RID-01	Mejora	19	PHPV-01	PS-01	Crítico
02	PHPD-01	RID-02	Mejora	20	PHPV-01	PS-02	Importante
03	PHPD-01	RID-03	Crítico	21	PHPV-01	PS-04	Importante
04	PHPD-01	RID-04	Importante	22	PHPV-01	PS-06	Mejora
05	PHPD-01	RID-05	Importante	23	PHPV-01	PS-08	Importante
06	PHPD-01	RID-06	Importante	24	PHPV-01	PS-10	Mejora
07	PHPV-01	ATPD-01	Crítico	25	PHPV-01	PS-12	Importante
08	PHPV-01	AC-02	Mejora	26	PHPV-01	PS-14	Mejora
09	PHPV-01	ANM-09	Importante	27	PHPU-01	CIU-01	Importante
10	PHPV-01	AC-05	Importante	28	PHPU-01	CC-01	Crítico
11	PHPV-01	AC-08	Importante	29	PHPU-01	CC-02	Importante
12	PHPV-01	AC-09	Importante	30	PHPU-01	CC-03	Importante
13	PHPV-01	AC-11	Importante	31	PHPU-01	CC-08	Mejora

14	PHPV-01	AC-12	Importante	32	PHPU-01	CC-09	Importante
15	PHPV-01	AC-17	Importante	33	PHPU-01	CC-10	Importante
16	PHPV-01	AC-18	Mejora	34	PHPU-01	CC-12	Crítico
17	PHPV-01	AC-19	Importante	35	PHPU-01	CC-14	Mejora
18	PHPV-01	AC-20	Importante	36	PHPU-01	IGU-02	Importante

La información presentada en esta tabla se muestra a modo de resumen en la Ilustración 3.1.

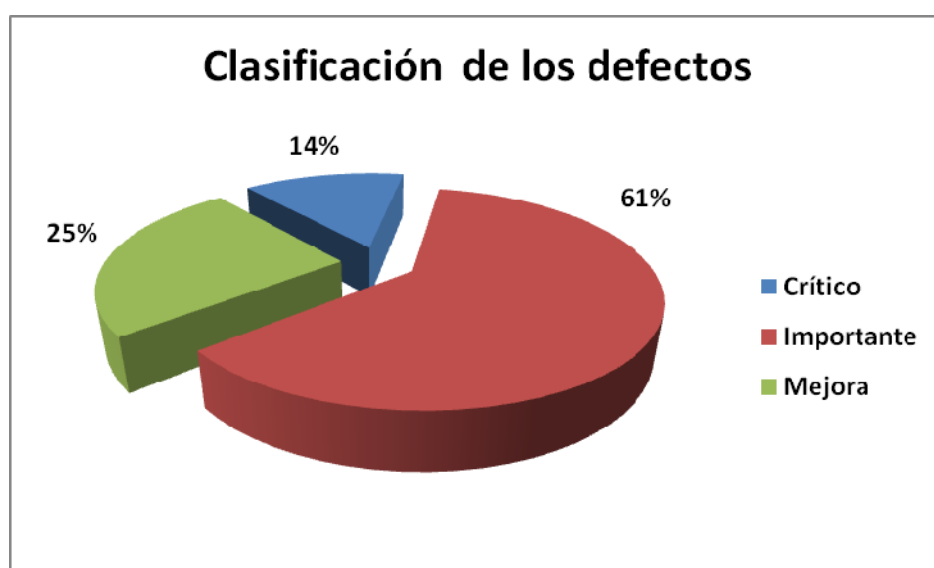


Ilustración 3.1 Clasificación de los defectos

### 3.3 Resumen de las pruebas

Como se ha mostrado el proceso de pruebas fue guiado por un plan global, compuesto por tres subplanes aglutinadores de pruebas y casos de prueba, a los que se les asignó un tiempo estimado de ejecución. Para la estimación del tiempo de las pruebas se realizó un estudio de dos propuestas de planificación del proceso de pruebas, una del ingeniero José Ignacio Cao del Instituto Tecnológico de Buenos Aires y la otra de María José Roca de GreenSQA-ParqueSoft, ambas publicadas en el año 2006. Luego en la tabla de resultados e inconformidades se registro el tiempo real de ejecución de cada uno de los casos de prueba, lo que permite realizar una

comparación entre el tiempo estimado y real por subplanes de prueba, ver Ilustración 3.2 e Ilustración 3.3.

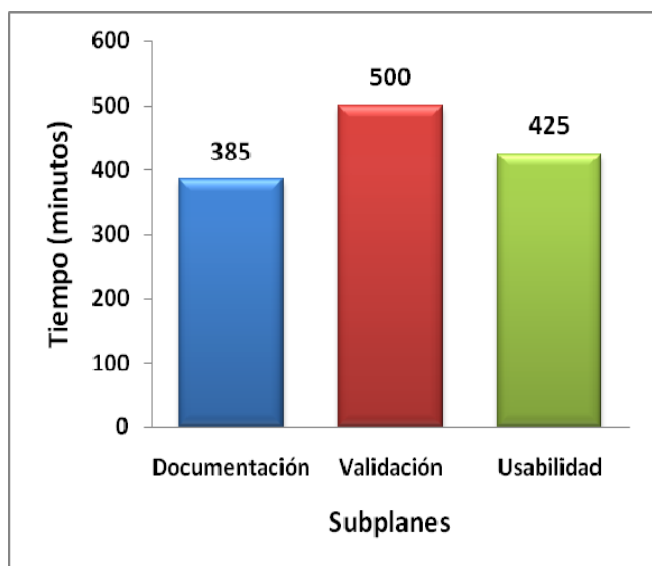


Ilustración 3. 2 Estimación del tiempo de ejecución

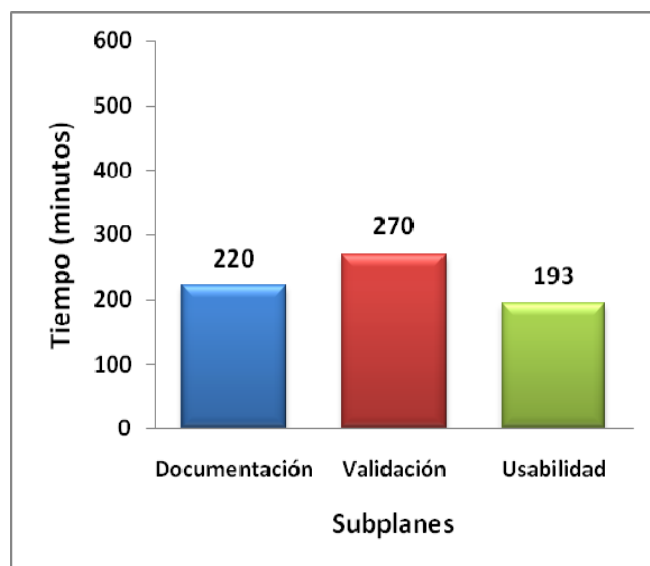


Ilustración 3. 3 Tiempo real de ejecución

De forma general se estimó la ejecución del plan en 22 horas, y el tiempo real de ejecución duró alrededor de 12. Esto fue motivado por el extenso proceso de análisis de la documentación técnica y la aplicación, antes de diseñar las pruebas, que a pesar de reducir el tiempo de ejecución en 10 horas, prolongó el de diseño a dos semanas después de lo planeado.

En el subplan de pruebas a la documentación se diseñaron casos de prueba para validar ciertas características que debe presentar la documentación técnica de cualquier software, para ello hubo que revisar todos los documentos que forman parte de dicha documentación. La siguiente tabla enumera todos los documentos analizados, sus versiones, veces que fueron revisados y errores encontrados en ellos. Desafortunadamente todos los documentos contienen errores, además de la imprecisión e incompletitud que caracteriza a esta documentación de modo general.

**Tabla 5 Relación de Documentos y Errores**

No.	Nombre del documento	Versión	Veces que fue revisado	Cantidad de errores
01	Listado de Hotwords	1	2	1. Incompleto 2. Falta ortográfica
02	listadoImagenes	1	2	1. Inconsistente (Referencia no hallada) 2. Incompleto
03	Listado de pantallas	1	2	1. Incompleto (Faltan las descripciones de las pantallas y algunas pantallas) 2. Mala organización.
04	general-mapa	1	2	1. Incompleto 2. Mala organización de los ítems
05	documentos-personales-mapa	1	2	1. Mala organización de los ítems 2. Nombre de ítem incorrecto (Relación de documentos personales)
06	creacion-literaria-mapa	1	2	1. Mala organización de los ítems
07	ediciones-mapa	1	2	1. Incompleto 2. Mala organización de los ítems
08	Pantalla 1.1 Presentacion	1	2	1. Mala especificación (Se debe mostrar una imagen, no un texto)
09	MóduloBibliografiaDescGen	1	2	1. Falta ortográfica 2. Estructura no uniforme en la numeración del subtítulo Objetos
10	Pantalla 2.6.1 Bibliografia	1	2	1. Falta ortográfica 2. Incompleto 3. Inclusión incorrecta ( Relacion de CREACIÓN LITERARIA.rtf)
11	Módulo Creación LiterariaDescGen	1	2	1. Falta ortográfica 2. Estructura no uniforme en la numeración del subtítulo Objetos
12	Pantalla 2.2 Creacion literaria	1	2	1. Falta ortográfica 2. La estructura de la información no está correctamente definida
13	Pantalla 2.2.1 Creacion literaria hist_narrativacartas_mitologia	1	2	1. Falta ortográfica 2. La estructura de la información no está correctamente definida 3. Nombre de pantalla incorrecto
14	Pantalla 2.2.2 Creacion literaria hist_narrativaerupcion_jorullo	1	2	1. Falta ortográfica 2. La estructura de la información no está correctamente definida 3. Nombre de pantalla incorrecto
15	Pantalla 2.2.3 Creacion literaria hist_	1	2	1. Falta ortográfica 2. La estructura de la información no

	narrativaespiritu_hist			está correctamente definida 3. Nombre de pantalla incorrecto
16	Pantalla 2.2.4 Creacion literaria lecciones_historiajpg	1	2	1. Falta ortográfica 2. La estructura de la información no está correctamente definida 3. Nombre de pantalla incorrecto
17	Pantalla 2.2.5 Creacion literaria miscelaneasMis1T1	1	2	1. Falta ortográfica 2. La estructura de la información no está correctamente definida 3. Nombre de pantalla incorrecto
18	Pantalla 2.2.6 Creacion literaria miscelaneasMis1T2	1	2	1. Falta ortográfica 2. La estructura de la información no está correctamente definida 3. Nombre de pantalla incorrecto
19	Pantalla 2.2.7 Creacion literaria miscelaneasMis2T1	1	2	1. Falta ortográfica 2. La estructura de la información no está correctamente definida 3. Nombre de pantalla incorrecto
20	Pantalla 2.2.8 Creacion literaria miscelaneasMis2T2	1	2	1. Falta ortográfica 2. La estructura de la información no está correctamente definida 3. Nombre de pantalla incorrecto
21	Pantalla 2.2.9 Creacion literaria otrosnoticias_inquisicion	1	2	1. Falta ortográfica 2. La estructura de la información no está correctamente definida 3. Nombre de pantalla incorrecto
22	Pantalla 2.2.10 Creacion literaria poesiadesconfianza	1	2	1. Falta ortográfica 2. La estructura de la información no está correctamente definida 3. Nombre de pantalla incorrecto
23	Pantalla 2.2.11 Creacion literaria poesiapiramide_cholula	1	2	1. Falta ortográfica 2. La estructura de la información no está correctamente definida 3. Nombre de pantalla incorrecto
24	MóduloCronologiaDescGen	1	2	1. Falta ortográfica 2. Estructura no uniforme en la numeración del subtítulo Objetos
25	Pantalla 2.5.2 textos Cronologia	1	2	1. Falta ortográfica 2. La estructura de la información no está correctamente definida 3. Nombre de pantalla incorrecto
26	Módulo DocPersonales DescGen	1	2	1. Falta ortográfica 2. Estructura no uniforme en la numeración del subtítulo Objetos
27	Pantalla 2.1 Introduccion	1	2	1. La estructura de la información no

				está correctamente definida 2. Nombre de pantalla incorrecto
28	Pantalla 2.1.2 DocPersonalesMuestra	1	2	1. La estructura de la información no está correctamente definida
29	Pantalla 2.1.2.1 DocPersonalesMuestra cartasgeorge_bourne 15-11-1834	1	2	1. La estructura de la información no está correctamente definida 2. Nombre de pantalla incorrecto
30	Pantalla 2.1.2.2 DocPersonalesMuestra cartasm 12-1829	1	2	1. La estructura de la información no está correctamente definida
31	Pantalla 2.1.2.3 DocPersonalesMuestra cartasmadre20-03-1828	1	2	1. La estructura de la información no está correctamente definida 2. Nombre de pantalla incorrecto
32	Pantalla 2.1.2.4 DocPersonalesMuestra cartasmadre21-09-1828	1	2	1. La estructura de la información no está correctamente definida 2. Nombre de pantalla incorrecto
33	Pantalla 2.1.2.5 DocPersonalesMuestra cartasmadre23-10-1828	1	2	1. La estructura de la información no está correctamente definida 2. Nombre de pantalla incorrecto
34	Pantalla 2.1.2.6 DocPersonalesMuestra cartassilvestre20-05-1827	1	2	1. La estructura de la información no está correctamente definida 2. Nombre de pantalla incorrecto
35	Pantalla 2.1.2.7 DocPersonalesMuestra cartassilvestre22-12-1826	1	2	1. La estructura de la información no está correctamente definida 2. Nombre de pantalla incorrecto 3. Falta ortográfica
36	Pantalla 2.1.2.8 DocPersonalesMuestra cartastio_ignacio04-11-1825	1	2	1. La estructura de la información no está correctamente definida 2. Nombre de pantalla incorrecto 3. Falta ortográfica
37	Pantalla 2.1.2.9 DocPersonalesMuestra cartastio_ignacio06.03.1826	1	2	1. La estructura de la información no está correctamente definida 2. Nombre de pantalla incorrecto 3. Falta ortográfica
38	Pantalla 2.1.2.10 DocPersonalesMuestra cartastio_ignacio08-10-1824	1	2	1. La estructura de la información no está correctamente definida 2. Nombre de pantalla incorrecto 3. Falta ortográfica
39	Pantalla 2.1.2.11 DocPersonalesMuestra cartastio_ignacio15-04-1826	1	2	1. La estructura de la información no está correctamente definida 2. Nombre de pantalla incorrecto 3. Falta ortográfica
40	Pantalla 2.1.2.12	1	2	1. La estructura de la información no



	DocPersonalesMuestra cartatio_ignacio22-02-1826			está correctamente definida 2. Nombre de pantalla incorrecto 3. Falta ortográfica
41	Pantalla 2.1.2.13 DocPersonalesMuestra cartasto 03-08-1828	1	2	1. La estructura de la información no está correctamente definida 2. Nombre de pantalla incorrecto 3. Falta ortográfica
42	Pantalla 2.1.2.14 DocPersonalesMuestra cartasviaje08-06-1824	1	2	1. La estructura de la información no está correctamente definida 2. Nombre de pantalla incorrecto 3. Falta ortográfica
43	Pantalla 2.1.2.15 DocPersonalesMuestra certificacion_nacimiento	1	2	1. La estructura de la información no está correctamente definida 2. Falta ortográfica
44	Pantalla 2.1.2.16 DocPersonalesMuestra certificado_defuncion	1	2	1. La estructura de la información no está correctamente definida 2. Falta ortográfica
45	Pantalla 2.1.2.16 DocPersonalesMuestra pasaporte	1	2	1. La estructura de la información no está correctamente definida 2. Falta ortográfica
46	Pantalla 2.1.2.17 DocPersonalesMuestra epitafio	1	2	1. La estructura de la información no está correctamente definida 2. Falta ortográfica
47	Módulo EdicionesDescGen	1	2	1. Falta ortográfica 2. Estructura no uniforme en la numeración del subtítulo Objetos
48	Pantalla 2.3 Ediciones introduccion	1	2	1. La estructura de la información no está correctamente definida 2. Falta ortográfica
49	Pantalla 2.3.1 Ediciones poesiaedicion1825	1	2	1. La estructura de la información no está correctamente definida 2. Nombre de pantalla incorrecto 3. Falta ortográfica
50	Pantalla 2.3.2 Ediciones poesiatomo1	1	2	1. La estructura de la información no está correctamente definida 2. Nombre de pantalla incorrecto 3. Falta ortográfica
51	Pantalla 2.3.3 Ediciones poesiatomo2	1	2	1. La estructura de la información no está correctamente definida 2. Nombre de pantalla incorrecto 3. Falta ortográfica
52	MóduloSobre HerediaDescGen	1	2	1. Falta ortográfica 2. Estructura no uniforme en la numeración del subtítulo Objetos

53	Pantalla 2.4.1 Sobre Heredia sobre_herediabachiller	1	2	1. La estructura de la información no está correctamente definida. 2. Nombre de pantalla incorrecto 3. La tipografía del texto es heterogénea
54	Pantalla 2.4.2 Sobre Heredia sobre_herediaenrique _pinneiro	1	2	1. La estructura de la información no está correctamente definida. 2. Nombre de pantalla incorrecto 3. La tipografía del texto es heterogénea
55	Pantalla 2.4.4 Sobre Heredia sobre_herediagertrudis _gomez	1	2	1. La estructura de la información no está correctamente definida. 2. Nombre de pantalla incorrecto 3. La tipografía del texto es heterogénea
56	Pantalla 2.4.5 Sobre Heredia sobre_herediajose_marti	1	2	1. La estructura de la información no está correctamente definida. 2. Nombre de pantalla incorrecto 3. La tipografía del texto es heterogénea
57	Pantalla 2.4.6 Sobre Heredia sobre_heredialezama_lima	1	2	1. La estructura de la información no está correctamente definida. 2. Nombre de pantalla incorrecto 3. La tipografía del texto es heterogénea
58	Pantalla 2.4.7 Sobre Heredia sobre_herediamanuel _sanguily	1	2	1. La estructura de la información no está correctamente definida. 2. Nombre de pantalla incorrecto 3. La tipografía del texto es heterogénea
59	Pantalla 2.4.8 Sobre Heredia sobre_herediaplacido	1	2	1. La estructura de la información no está correctamente definida. 2. Nombre de pantalla incorrecto 3. La tipografía del texto es heterogénea
60	Menu POp up Bibliografía	1	2	1. Incompleto 2. Falta ortográfica
61	Menu POp up Documentos	1	2	1. Falta ortográfica 2. La tipografía del texto es heterogénea
62	Menu POp upPrincipal	1	2	1. Incompleto 2. Falta ortográfica
63	LEEME	1	2	1. Incompleto 2. Falta ortográfica
64	Pautas generales	1	2	1. Falta ortográfica

65	Pautas generales funcionales	1	2	1. Falta de coordinación gramatical en el punto 2 del subtítulo Nota 2. Falta ortográfica
66	Proyecto Multimedia	1	2	1. Incompleto 2. Falta ortográfica

En los subplanes de pruebas a la documentación, de validación y usabilidad se diseñaron algunas pruebas en función del elemento o características del software que prueban y otras según su misma definición. En la siguiente tabla se muestra por cada una de las pruebas diseñadas en los subplanes antes mencionados, la cantidad de casos de prueba, cantidad de casos de prueba exitosos, cantidad de verificadores y tiempo de ejecución.

Leyenda:

**No.** : Número de la secuencia de las prueba.

**SP:** Tema del Sub-Plan de prueba

**NP:** Nombre de la Prueba

**CP:** Cantidad de casos de prueba

**CE:** Cantidad de casos de prueba exitosos

**CV:** Cantidad de verificadores

**t(min):** Tiempo de ejecución en minutos.

**Tabla 6 Relación de pruebas y errores**

No.	SP	NP	CP	CE	CV	t(min)
1	PHPD-01	Revisión e Inspección a la documentación	6	6	1	220
2	PHPV-01	Pruebas al área de título del programa y documento actual	1	1	1	20
3		Pruebas al área de navegación entre módulos y al área de contenidos	35	11	1	204
4		Pruebas al Panel de Servicios	18	8	1	46
5	PHPU-01	Consistencia de la Interfaz de Usuario	1	1	1	35
6		Criterios de Calidad	15	8	1	65
7		Interfaz Gráfica de Usuario	2	1	1	20

La anterior información muestra que de un total de 78 casos de prueba diseñados, 5 no fueron aplicados y 36 descubrieron errores, este resultado de manera gráfica se muestra en la Ilustración 3.4.

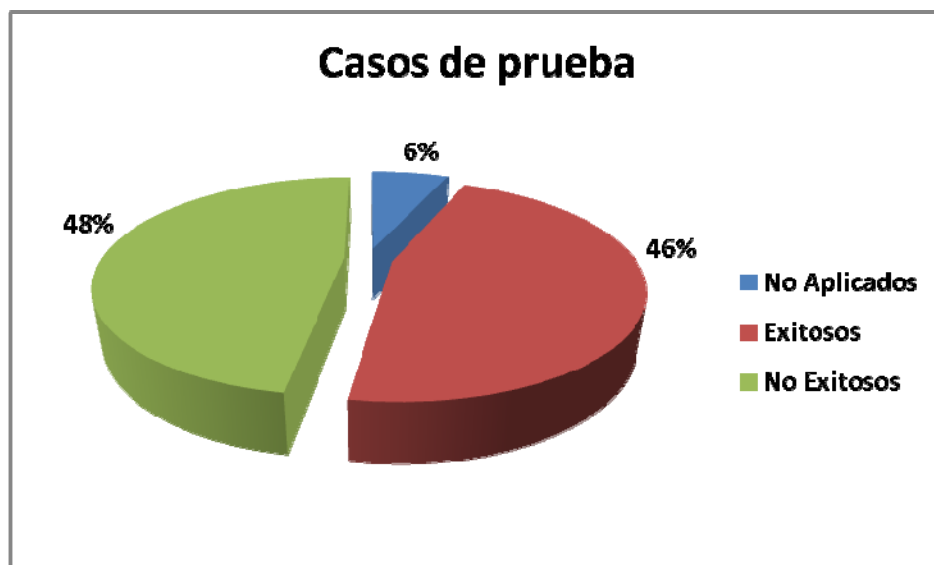


Ilustración 3. 4 Resumen de la ejecución de los casos de prueba

De modo general el proceso de prueba fue exitoso, pues se logró cubrir casi el 100 % de las especificaciones del sistema y el tiempo real de ejecución de los casos de prueba fue bastante inferior con respecto al estimado, aunque por inexperiencia se prolongó bastante el tiempo de diseño del plan de pruebas.

Hubieron algunos factores que dificultaron un tanto las pruebas como, la desorganización e imprecisión de la documentación técnica, la no actualización del guión en función de los cambios propuestos por los desarrolladores, la no utilización de una guía probada o propuesta de desarrollo de software hipermedia y la ausencia de recursos especificados por el cliente, con el fin de simular un ambiente parecido al de explotación, y por consiguiente la imposibilidad de diseño y ejecución de algunos casos de prueba.

### 3.4 Costo del trabajo

Para definir el costo del proceso de desarrollo de esta tesis, se decidió calcular algunos aspectos generales como:

**ESF** - Esfuerzo

**CH** - Cantidad de hombres

**CMT** - Cantidad meses trabajados

**CDMT** - Cantidad de días por mes trabajados

**TDT** – Total de días trabajados.

**CHDT** - Cantidad de horas por día trabajadas

**SM** - Salario mensual

**CTR** – Costo total de los recursos empleados

**GE** - Gasto energético

**CPDT** - Costo del proceso de desarrollo de la tesis

El costo del proceso de desarrollo de la tesis estará expresado de forma monetaria en pesos cubanos. Para calcular el mismo se deben calcular tres factores, el esfuerzo, el gasto energético y el costo total de los recursos empleados. El esfuerzo se expresará a través de lo invertido en hombres y tiempo, para el gasto energético se calculará cuanta energía consumió la computadora personal a lo largo del proceso y cuanto esto representa en efectivo para la UCI. El costo total de los recursos será la sumatoria de lo invertido en cada recurso.

El trabajo fue desarrollado en 7 meses por una sola persona, con salario mensual de 100 pesos. Por día como promedio se trabajaron 6 horas, y la cantidad de días por mes se especifica a continuación.

	Diciembre	Enero	Febrero	Marzo	Abril	Mayo	Junio	TDT
<b>CDMT</b>	15	17	25	27	23	28	5	140

**CMT = 7      CH = 1      SM = 100      CHDT = 6**

**ESF = CMT \* CH \* SM**

$$ESF = 7 * 1 * 100$$

$$ESF = 700.00$$

Para computar el gasto energético se debe calcular el total de horas trabajadas, infiriéndose que todo el trabajo se desarrolló utilizando la computadora, conociendo que la misma consume 150 watts por hora (150 W/h) y que a la UCI le cuesta 1.6 pesos el kilowatts (Kw). La ecuación del gasto energético quedaría expresada de la siguiente forma.

$$GE = THT * 150 * 1.6 / 1000$$

Donde

$$THT = CHDT * TDT$$

$$THT = 6 * 140$$

$$THT = 840$$

$$GE = 840 * 150 * 1.6 / 1000$$

$$GE = 201.60$$

Recursos	Costo
Parlantes	20.00
2 Diskettes	10.00
Un lapicero	5.00
Una libreta	5.00
120 hojas de papel	48.00
<b>CTR</b>	<b>88.00</b>

$$CPDT = ESF + GE + CTR$$

$$CPDT = 700.00 + 201.60 + 88.00$$

$$CPDT = 989.60$$

El costo del proceso de desarrollo de la tesis se estima en 989.60 pesos.

### **3.5 Conclusiones**

En este capítulo se organizaron y documentaron los resultados arrojados por la ejecución de los casos de prueba, diseñados en el capítulo anterior. Luego se realizó un resumen, análisis y costo del proceso de prueba, permitiendo concluir que el software probado no cumple con las especificaciones establecidas en la documentación técnica, la cual a su vez contiene diversos errores, poniendo en evidencia la falta de calidad de la aplicación y la no aptitud de la misma para ser entregada al cliente.

## **CONCLUSIONES GENERALES**

En este trabajo se definieron un conjunto de pruebas a realizar sobre la aplicación Heredia, no antes de hacer un análisis y comparación de conceptos y definiciones relacionados con el tema, además de un estudio del estado del arte de las pruebas del software.

A continuación se realizó el diseño y ejecución de un plan de pruebas, compuesto por subplanes, pruebas y casos de prueba, con el objetivo de evaluar la completitud funcional y la usabilidad de la aplicación y su documentación.

Los resultados de la ejecución de los casos de prueba y la descripción de las inconformidades fueron registrados en una tabla, con otros datos de interés, lo que propició la realización de un resumen, análisis y costo del proceso de prueba.

Quedó demostrado que la aplicación y su documentación presentan muchos defectos, principalmente por el mal proceso de captura y organización de las especificaciones y la no actualización de los cambios propuestos por el equipo de desarrollo; además de la no utilización de documentos que pudiesen guiar acertadamente la construcción del software.

Se validó que los métodos de prueba de caja negra basados en grafos, cubren casi el 100% de la estructura y funcionalidad de un sistema.



## RECOMENDACIONES

- ✓ Diseñar y ejecutar casos de prueba pertenecientes a las pruebas de validación del área de navegación entre módulos y el área de contenidos, que no fueron incluidos en este documento por la similitud con los expuestos, con el fin de encontrar más errores y erradicarlos.
- ✓ Redefinir la documentación técnica, en un intento de facilitar la comprensión del sistema y sus requisitos.
- ✓ Implementar los elementos ausentes y que todos los presentes cumplan con las características y funcionalidad especificada, además de tener en cuenta los aspectos que se definen en el subplan de usabilidad.
- ✓ Aplicar el plan de pruebas diseñado en este trabajo a otros libros electrónicos, incorporando o disminuyendo casos de prueba, en función de la aplicación a probar, propiciando esto una variedad de pruebas que pueden tenerse en cuenta, a la hora de preparar una guía para realizar el proceso de prueba a este tipo de software.
- ✓ Desarrollar un sistema capaz de almacenar, organizar, modificar, mostrar y eliminar la información del plan de pruebas y los resultados.
- ✓ Apoyarse en una guía de desarrollo de aplicaciones hipermedia para realizar un producto de este tipo.
- ✓ Antes de empezar a implementar cualquier sistema, se deberían validar los requisitos del mismo.

## REFERENCIAS BIBLIOGRÁFICAS

**CISMAP. 2000.** Metrica 3, Metodología de Planificación, Desarrollo y Mantenimiento de sistemas de información. [En línea] 2000. [Citado el: 12 de Marzo de 2007.]

<http://www.csi.map.es/csi/metrica3/index.html>.

**Gutiérrez, Javier Jesus. 2005.** Generación de Pruebas a partir de la especificación funcional. [En línea] 2005. [Citado el: 7 de Febrero de 2007.]

[http://www.lsi.us.es/~javierj/investigacion\\_ficheros/NDEAv25.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/NDEAv25.pdf) .

**Hassan, Yusef. 2002.** Diseño Hipermedia centrado en el usuario. . [En línea] 2002. [Citado el: 16 de Mayo de 2007.] <http://www.nosolousabilidad.com/articulos/hipermedia.htm>.

**GCSUA-HSH. 2006.** Diseño de Plan de Prueba Funcionales. [En línea] 2006. [http://chie.uniandes.edu.co/~gsd/index.php?option=com\\_content&task=view&id=131&Itemid=183](http://chie.uniandes.edu.co/~gsd/index.php?option=com_content&task=view&id=131&Itemid=183).

**Marqués, Graells. 2000.** Elaboración de Materiales Formativos Multimedia. Criteios de Calidad. [En línea] 2000. [Citado el: 15 de Febrero de 2007.]

[http://dewey.uab.es/paplicada/htm/papers/Paper\\_materialesmultimedia\\_marques.htm](http://dewey.uab.es/paplicada/htm/papers/Paper_materialesmultimedia_marques.htm) .

**Mogyorodi, Gary. 2001.** *Requirements-Based Testing: An Overview*. 2001.

**Patón, Eduardo Fernández-Medina. 2006.** Curso de Ingeniería del Software, Tema 9. [En línea] 2006. [Citado el: 5 de Abril de 2007.] <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema09.pdf>.

**Pressman, Roger S. 2005.** *Ingeniería del software. Un enfoque práctico. Parte I*. La Habana, Cuba : Felix Varela, 2005.

**Soriano, Amelia. 2005.** Tipos de Prueba. [En línea] 2005. [Citado el: 18 de Abril de 2007.] <http://carolina.terna.net/ingsw3/links.htm> .

**Teruel, Alejandro. 2001.** El Plan de Pruebas. [En línea] 2001. [Citado el: 2 de Abril de 2007.] <http://www ldc.usb.ve/~teruel/ci4713/clases2001/planPruebas.html>.

**Valencia, Eugenia. 1997.** Un Método de Desarrollo de Aplicaciones Educativas Hipermedia. [En línea] 1997. [Citado el: 15 de Diciembre de 2006.]  
<http://www.tise.cl/archivos/tise97/trabajos/trabajo2/index.htm>.

**Zuleta Arango, René. 2005.** Estándares ISO y Usabilidad en desarrollo de software. [En línea] 2005. [Citado el: 3 de Abril de 2007.] <http://www.metodus.com/articulos/iso/index.jsp>.

## BIBLIOGRAFÍA

1. **Cao, José Ignacio. 2006.** Principios para un método de estimación de proyectos de software basado en los escenarios principales. [En línea] 2006. [Consultado el: 15 de Abril de 2007] <http://www.itba.edu.ar/capis/epg-tesis-y-tf/cao-trabajofinaldeespecialidad.pdf>.
2. **Fernández Carrasco, Oscar M., García León, Delba y Beltrán Benavides, Alfa. 1995.** Un enfoque actual sobre la calidad del software. [En línea] 1995. [Consultado el: 3 de Abril de 2007.]. [http://bvs.sld.cu/revistas/aci/vol3\\_3\\_95/aci05395.htm](http://bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm).
3. **Fernández Escribano, Gerardo. 2002.** Introducción a Extreme Programming. [En línea] 2002. [Consultado el: 7 de Febrero de 2007.]. <http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion-XP.pdf>.
4. **GCSUA-HSH. 2006.** Diseño de Plan de Prueba Funcionales. [En línea] 2006. [Consultado el: 23 de Marzo de 2007.]. [http://chie.uniandes.edu.co/~gsd/index.php?option=com\\_content&task=view&id=131&Itemid=183](http://chie.uniandes.edu.co/~gsd/index.php?option=com_content&task=view&id=131&Itemid=183).
5. **Joel Spolsky. 2000.** Diseño de Interfaz de Usuario para Programadores, Capítulo 5: Consistencia y Otros Duendes. [En línea] 2000. [Consultado el: 19 de Abril de 2007.]. <http://spanish.joelonsoftware.com/uibook/chapters/5.html>.
6. **López, Hozcar Andrés. 2006.** Metodología de trabajo, Cognox Ltda. [En línea] 2001. [Consultado el: 12 de Febrero de 2007] [http://www.cognox.com/index.php?option=com\\_content&task=blogcategory&id=73&Itemid=58&lang=es](http://www.cognox.com/index.php?option=com_content&task=blogcategory&id=73&Itemid=58&lang=es).
7. **Mendoza Sánchez, María. 2004.** Metodologías De Desarrollo De Software. [En línea] 2004. [Consultado el: 2 de Febrero de 2007.]. [http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004)

.html.

8. **Menéndez-Barzanallana Asensio, Rafael. 2005.** Metodologías de desarrollo de software. [En línea] 2005. [Consultado el: 3 de Abril de 2007.].  
[http://www.wikilearning.com/metodologia\\_de\\_desarrollo\\_de\\_software-wkccp-3617-1.htm](http://www.wikilearning.com/metodologia_de_desarrollo_de_software-wkccp-3617-1.htm).
9. **Pérez, M., Rojas, T., Mendoza, L. & A. Grimán. 2001.** Modelo de calidad de software educativo: Aplicación del estándar internacional ISO/IEC 9126. [En línea] 2001. [Consultado el: 12 de Febrero de 2007]  
[http://www.lisi.usb.ve/publicaciones/02%20calidad%20sistemica/calidad\\_29.pdf](http://www.lisi.usb.ve/publicaciones/02%20calidad%20sistemica/calidad_29.pdf).
10. **Pressman, Roger S. 2005.** *Ingeniería del software. Un enfoque práctico. Parte I.* La Habana, Cuba : Felix Varela, 2005.
11. **Roca, María José. 2006.** Planeación de pruebas funcionales: Aspectos relevantes. [En línea] 2006. [Consultado el: 15 de Abril de 2007].  
<http://www.greensqa.com/archivos/Art02-PlaneacionPruebasFuncionales.pdf>.
12. **Teruel, Alejandro. 2001.** Las Pruebas de Verificación de Requerimientos. 2001. [En línea] 2001. [Consultado el: 12 de Febrero 2007]  
<http://www ldc.usb.ve/~teruel/ci4713/clases2001/testReqs.html>.

## GLOSARIO DE TÉRMINOS

**ATPD:** Área de título del programa y documento actual

**ANM:** Área de navegación entre módulos

**AC:** Área de contenidos

**CIU:** Consistencia en la Interfaz del usuario

**CC:** Criterios de calidad

**CISMAP:** Centro de Informática Superior. Ministerio de Administración Pública.

**Defecto:** Problema de calidad que es descubierto de entregar el software a los usuarios finales (o a otra actividad del proceso del software)

**Error:** Problema de calidad que es descubierto por los ingenieros del software (u otros) antes de entregar el software a los usuarios finales (o a otra actividad del proceso del software)

**GCSUA-HSH:** Grupo de Construcción de Software de la Universidad de los Andes-Heinsohn Software House

**IGU:** Interfaz gráfica de usuario

**PS:** Panel de Servicios

**RID:** Revisión e Inspección de la Documentación