

**Universidad de las Ciencias Informáticas**  
**Facultad 8**



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

**PROCESO DE PRUEBAS AL  
PRODUCTO MULTIMEDIA 'BARRIO  
ADENTRO AL ALCANCE DE TUS  
MANOS'**

**Autora:** Yanet Brito Riverol

**Tutor:** Ing. Liván Peña Prieto

**Ciudad de la Habana, Julio 2007**

**Año 49 de la Revolución**

## **DECLARACIÓN DE AUTORÍA**

Declaramos ser autores de la presente tesis y reconocemos 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 \_\_\_\_\_.

Yanet Brito Riverol

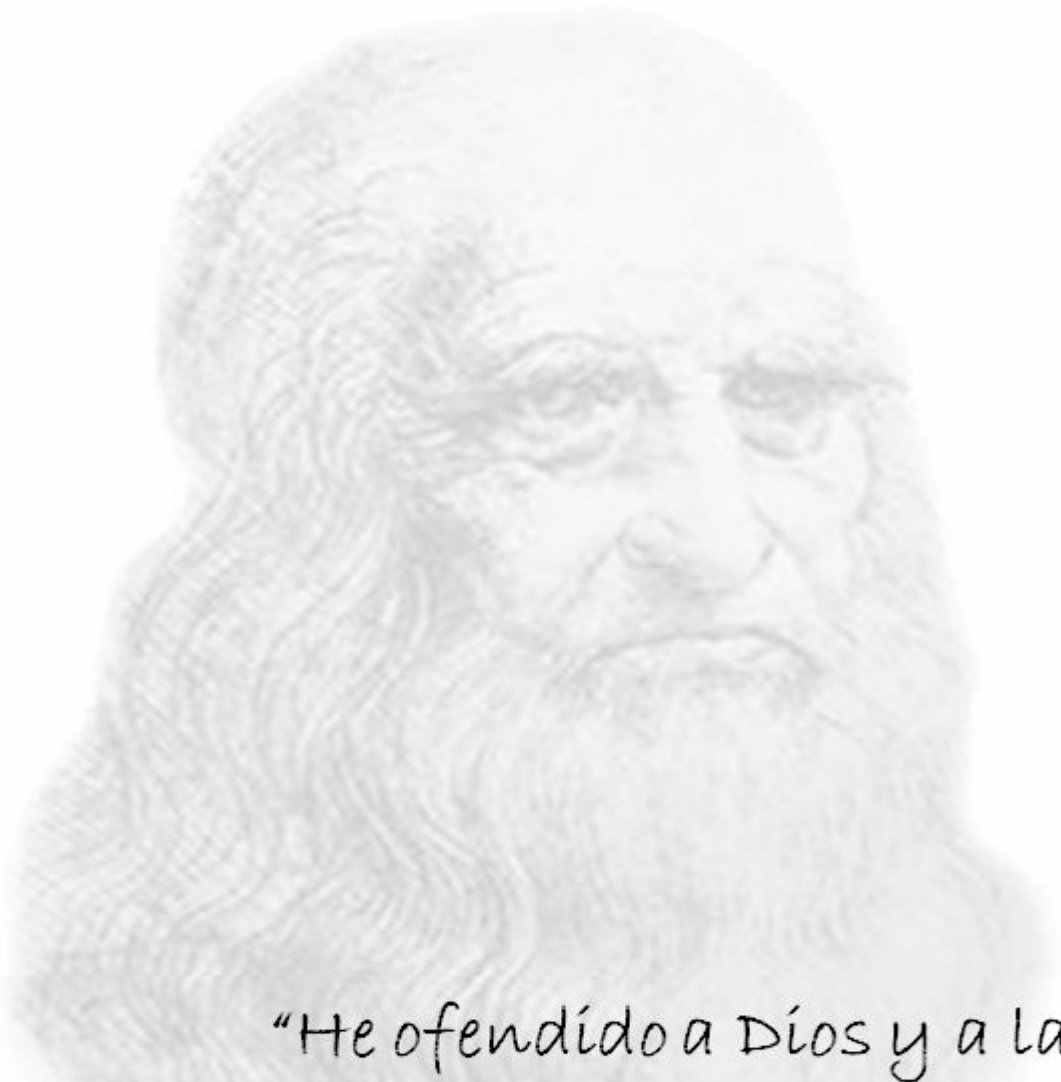
---

Firma del Autor

Ing. Liván Peña Prieto

---

Firma del Tutor



“He ofendido a Dios y a la  
humanidad porque mi trabajo no  
tuvo la calidad que debía haber  
tenido”

Leonardo Da Vinci

## AGRADECIMIENTOS

Al profesor *Ing. Liván Peña Prieto*, por sus consejos, su orientación, su ayuda y paciencia, como tutor de este trabajo de diploma.

A la *Ing. Marianne Licea Fonseca*, por sus valiosos aportes y consejos, gracias por no negarme su tiempo.

A los *profesores* que me brindaron su colaboración, por su tiempo, sugerencias y orientación en el transcurso de la investigación.

Al profesor *Francisco Javier Hdz Caos*, por saber ser guía y ejemplo además de profe, por recordar a sus estudiantes aunque el tiempo pase..

A mi *mami*, incansable crítica de mis actos, por sus desvelos y sugerencias en la revisión del manuscrito, por darme todo lo que hoy soy, por ser tan especial...

A mi *papá*, por refunfuñar tanto.. todo en la vida pasa por más sacrificio que exija...te quiero.

A *Luis y Milagro*, a ustedes también por buscarse una hija más de gratis... y bien cara...jejeje, gracias por todo....

A mis *tíos*, por tenerme siempre presente y no dejar de quererme en todo este tiempo que he estado tan lejos....los sigo queriendo igual de grande.

A mis *primos*, a ustedes porque si...como me extrañarás *gordi (Yeni)*.

A mis *otras almas (mamá Lila y abue Chico)* por no dejar escapar el más mínimo detalle en mi vida y permitirme seguir siendo *la niña*....aunque ya vaya quedando poco de esa etapa...

A *Yase (mi gordo por siempre)*.... por creerse el hermano mayor y correr con mis gustos.. ya casi sales de mi...

A *Nohelda, Zenaida y Gerónimo*, mi *otra gran familia*, quienes con su cariño sincero, llenándome de detalles y sacándome de un que otro susto, me hicieron sentir como en casa, gracias por todo....

A mis *viejas amigas (Ari, Yeney, Dayi)*... por ser sencillamente eso, *grandes amigas* y no aves de paso...

A mi *gente de allá (el clan de mi mami)* por haber hecho suyos con gestos de preocupación todos estos años de estudio.

A esas personas especiales que me dieron una amistad sincera y con cariño desde siempre....*Mendoza, Blanca, Yennia, Betty, Yudisbel, Osvaldo*.....gracias por cada plática y detalle, los quiero asíí... de grande..

También al *clan* que llegó luego.... *Lilian, Isis, Yini, Telma*... no por eso fueron menos, a todos gracias por existir y llenarme de alegrías..

A mi *grupo*, por su diversidad de caracteres y originalidades que lo hacen tan especial, sean siempre así.

A los *muchos de tesis de prueba (Lilian, Rey, Sadhit, Yennia, Blanca, Lida, Tony, Aramis)* por fucionar como un equipo de investigación que supo salir a flote,...a Telma y Adis también. Por fin terminamos.

A todas aquellas *instituciones* y *personas* que de una forma u otra colaboraron en el desarrollo del trabajo.

A la *universidad*, por ser un gran centro de estudios y moldearme como lo que hoy soy..

A esta eterna *Revolución* y a nuestro excepcional Comandante *Fidel Castro* por permitirme ser parte de un sueño hecho realidad.

*A todos una vez más gracias... son y serán parte de mí.*

**DEDICATORIA**

*A mis padres*

*por su esfuerzo incansable y paciencia infinita en darme lo mejor de ellos:  
educación, experiencias, regaños y sobre todo, un amor incondicional.*

*A mis abu*

*por sus consentimientos y mimos, por esa preocupación y desvelo  
impregnados de un cariño sin límites.*

*A mi tío*

*por quererme así de grande, por enseñarme a ser mejor y confiar siempre en  
mí, nunca te defraudaré.*

*A mi hermano,*

*por ser esa voz pequeña con alma de hermano mayor.*

---

## RESUMEN

La exigencia e interés creciente por parte del mercado de productos software de alta calidad, es un indicador de la percepción de la calidad como un elemento cada vez más necesario para su comercialización. Un requisito indispensable para ello, lo constituye la realización de pruebas, en el momento adecuado y del tipo adecuado. Estas figuran como un instrumento para evaluar la calidad al detectar los errores existentes. La presente investigación, versa sobre la aplicación de pruebas al producto multimedia “Barrio Adentro al alcance de tus manos”, en aras de obtener un producto con un mínimo de errores posibles. La misma describe el proceso de pruebas planteado, a partir de una estrategia que define dos niveles de evaluación: modular y sistema. Se diseñan casos de prueba según las pruebas a aplicar, basados en los requerimientos a evaluar (funcionales en su gran mayoría). Una vez ejecutadas y obtenidos resultados, estos se documentan para su análisis y posterior uso, en caso de un futuro proceso de depuración. Se determina la factibilidad del proceso obteniéndose conclusiones generales.

## PALABRAS CLAVE

Prueba de software, plan de prueba, estrategia de prueba, multimedia, caso de prueba.

ÍNDICE DE CONTENIDO

**INTRODUCCIÓN** ..... 1

**CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA** ..... 5

1.1 Introducción ..... 5

1.2 Calidad de Software ..... 5

    1.2.1 *Calidad y pruebas de Software* ..... 6

1.3 Importancia de probar ..... 7

1.4 Génesis de las pruebas ..... 8

1.5 Las pruebas de software ..... 9

    1.5.1 *Objetivos de las pruebas* ..... 9

    1.5.2 *Principios de las pruebas* ..... 10

    1.5.3 *Tareas a realizar para probar software* ..... 10

1.6 ¿Dónde radica el éxito de una prueba? ..... 11

1.7 Proceso de pruebas ..... 12

1.8 Estrategia de pruebas ..... 13

    1.8.1 *Características generales de la Estrategia de Prueba* ..... 13

1.9 Plan de prueba ..... 14

    1.9.1 *Posibles componentes de un plan de pruebas* ..... 14

1.10 Caso de prueba ..... 17

    1.10.1 *Diseño de Casos de prueba* ..... 17

1.11 Métodos de pruebas ..... 18

    1.11.1 *Prueba de Caja Blanca* ..... 18

        1.11.1.1 *Métodos de prueba de caja blanca* ..... 19

    1.11.2 *Prueba de Caja Negra* ..... 20

        1.11.2.1 *Métodos de prueba de caja negra* ..... 21

1.12 Niveles de Prueba ..... 23

1.13 Tipos de pruebas por niveles ..... 24

    1.13.1 *Pruebas a Nivel de unidad* ..... 25

    1.13.2 *Pruebas a Nivel de Integración* ..... 25

    1.13.3 *Pruebas a Nivel de sistema* ..... 26

    1.13.5 *Pruebas a nivel de aceptación* ..... 28

    1.13.6 *Otras* ..... 29

1.14 Multimedia ..... 30

    1.14.1 *Pruebas a Multimedia* ..... 30

    1.14.2 *Estrategias de pruebas a Multimedia* ..... 31

1.15 Conclusiones ..... 34

**CAPÍTULO II: DESCRIPCIÓN DEL PROCESO DE PRUEBA** ..... 35

2.1 Introducción ..... 35



2.2 Descripción del producto .....	35
2.3 Proceso de pruebas.....	36
2.3.1 Descripción del proceso de prueba a aplicar .....	37
2.3.1.1 Descripción de las etapas a desarrollar.....	38
2.4 Plan de prueba general .....	40
2.4.1 Introducción .....	41
2.4.2 Organización del equipo de pruebas .....	42
2.4.3 Especificación del software y hardware necesarios .....	42
2.4.4 Descripción de Requerimientos a probar .....	43
2.4.5 Estrategia de prueba.....	45
2.4.6 Diseño de las pruebas .....	45
2.4.7 Procedimiento de pruebas .....	46
2.4.7.1 Pruebas de unidad .....	46
2.4.7.2 Pruebas de sistema.....	47
2.4.7.3 Otras .....	49
2.4.8 Casos de Prueba .....	51
2.4.9 Criterios de validación.....	76
2.4.10 Definición de entregables .....	77
2.4.11 Recursos requeridos.....	77
2.4.12 Planificación .....	78
2.5 Conclusiones .....	80
<b>CAPÍTULO III: RESULTADOS DE LAS PRUEBAS .....</b>	<b>81</b>
3.1 Introducción .....	81
3.2 Documentación de los resultados.....	81
3.3 Resultados de las pruebas .....	81
3.3.1 Resultados específicos por pruebas .....	82
3.4 Análisis de los resultados.....	90
3.4.1 Análisis de resultados .....	92
3.5 Evaluación de las pruebas.....	93
3.6 Recomendaciones específicas al producto.....	95
3.7 Costos de las pruebas .....	96
3.7.1 Estimación de costos .....	96
3.8 Conclusiones .....	99
<b>CONCLUSIONES GENERALES .....</b>	<b>100</b>
<b>RECOMENDACIONES .....</b>	<b>102</b>
<b>BIBLIOGRAFÍA REFERENCIADA.....</b>	<b>103</b>
<b>BIBLIOGRAFÍA CONSULTADA.....</b>	<b>106</b>

## ÍNDICE DE FIGURAS

<u>Fig. 1 Representación de técnica de prueba de Caja blanca</u> .....	19
<u>Fig. 2 Representación de técnica de prueba de Caja negra</u> .....	21
<u>Fig. 3 Descripción de las actividades del proceso de pruebas</u> .....	40
<u>Fig. 4 Estrategia de prueba</u> .....	46
<u>Fig. 5 Planificación de etapas de pruebas según actividades</u> .....	79
<u>Fig.6 Planificación de las actividades (Diagrama de Gantt)</u> .....	80
<u>Fig.7 Tabla de defectos y no conformidades</u> .....	82
<u>Fig.8 Porcentaje de errores según su clasificación</u> .....	92
<u>Fig.9 Número de errores según pruebas realizadas</u> .....	93
<u>Fig. 10 Análisis de requerimientos del producto</u> .....	95

## ÍNDICE DE TABLAS

<u>Tabla 1 Procedimiento de prueba de interfaz de usuario</u> .....	47
<u>Tabla 2 Procedimiento de prueba funcional</u> .....	48
<u>Tabla 3 Procedimiento de prueba de instalación</u> .....	49
<u>Tabla 4 Procedimiento de prueba de interfaz de usuario</u> .....	49
<u>Tabla 5 Procedimiento de prueba de solidez</u> .....	50
<u>Tabla 6 Procedimiento de prueba a la documentación</u> .....	50
<u>Tabla 7 Procedimiento de prueba de interfaz gráfica de usuario</u> .....	51
<u>Tabla 8 CP: Interactuar escenario de presentación Módulo2.</u> .....	53
<u>Tabla 9 CP: Mostrar presentación del producto</u> .....	54
<u>Tabla 10 CP: Acceder a módulos 1 o 2.</u> .....	56
<u>Tabla 11 CP: Interactuar escenario ¿Qué es Barrio Adentro?</u> .....	58
<u>Tabla 12 CP: Interactuar escenario “Servicios que ofrece”, Nivel 2.</u> .....	60
<u>Tabla 13 CP: Interactuar juego Sopa de letras.</u> .....	62
<u>Tabla 14 CP: Interactuar ventana Ayuda.</u> .....	64
<u>Tabla 15 CP: Interactuar ventana de felicitación.</u> .....	65
<u>Tabla 16 CP: Interactuar escenario informativos módulo 2.</u> .....	66
<u>Tabla 17 CP: Interactuar opción Controlar audio de fondo.</u> .....	68
<u>Tabla 18 CP: Interactuar opción Inicio.</u> .....	69
<u>Tabla 19 CP: Interactuar vínculos de navegabilidad.</u> .....	70
<u>Tabla 20 CP: Instalación de la Multimedia en los diferentes sistemas de software.</u> .....	71
<u>Tabla 21 CP: Reconocer eventos de hardware</u> .....	72
<u>Tabla 22 CP: Correspondencia entre diseño e interfaz del sistema.</u> .....	73
<u>Tabla 23 CP: Comprobar integridad y coexistencia en la documentación.</u> .....	74
<u>Tabla 24 CP: Chequeo de estándares para elementos de interfaz gráfica de usuario (IGU)</u> .....	76
<u>Tabla 25 Descripción de Recursos humanos.</u> .....	78
<u>Tabla 26 Recursos de hardware.</u> .....	78

---

<u>Tabla 27 Resultados de pruebas funcionales</u> .....	84
<u>Tabla 28 Resultados de pruebas a interfaz gráfica</u> .....	86
<u>Tabla 29 Resultados de Prueba de solidez</u> .....	89
<u>Tabla 30 Resultados de prueba a la documentación</u> .....	90
<u>Tabla 31 Formato de registro de defectos – Inspecciones</u> .....	91
<u>Tabla 32 Gastos por concepto de consumo de tiempo</u> .....	97
<u>Tabla 33 Gastos por concepto de uso de recursos materiales</u> .....	97
<u>Tabla 34 Gastos por concepto de consumo energético (GE)</u> .....	98

## **INTRODUCCIÓN**

Propiciar la calidad en el Software es una actividad que ha surgido como consecuencia de la fuerte demanda y de la competencia, debido a la vorágine de ofertas en el mercado, convirtiéndose en una necesidad prioritaria para las organizaciones que lo desarrollan. El interés por la calidad crece de forma continua, a medida que los clientes se vuelven más selectivos y comienzan a rechazar los productos poco fiables o que realmente no dan respuesta a sus necesidades.

El aseguramiento de la calidad toma en cuenta todas aquellas acciones planificadas y sistemáticas necesarias para proporcionar la confianza de que un producto o servicio satisfaga los requisitos de calidad establecidos (Mendoza, 2003).

La Universidad de las Ciencias Informáticas (UCI), es un centro de estudios que contribuye a la formación de profesionales integrales, tanto en su desarrollo intelectual, como profesional. Fue creada en el año 2002, como miembro activo del Programa Nacional de Informatización de la Sociedad. Sus estudiantes tienen la particularidad de vincular directamente el estudio y la producción desde etapas tempranas en el centro, ayudando a dar solución a las múltiples solicitudes de automatización que llegan a la misma, ya sean de carácter nacional o internacional.

Para la universidad el argumento de la calidad es exhibido como factor diferenciador, como clave de sus procesos de negocio y como slogan de competitividad empresarial.

Hoy, intenta dar respuesta a la creciente demanda que llega a ella de productos de tipo multimedia utilizados para el aprendizaje, como alternativa a la barrera del alcance tecnológico a Internet. Conjuntamente con el apoyo de una de sus facultades dedicada a esta esfera de la producción (Facultad 8) la institución ha asumido la elaboración de este tipo de software como medio de enseñanza que arroja resultados positivos en su variante de educación a distancia por su integración de texto, gráficos, sonido digitalizado y video, bajo la premisa de satisfacer las necesidades planteadas por el cliente.

Esto ha acarreado una preocupación constante en los grupos de trabajo que llevan adelante la tarea, al intentar dar respuesta a una serie de interrogantes que con la práctica van surgiendo: ¿cómo saber si el producto construido se corresponde exactamente con lo que el cliente pidió o si va a funcionar

correctamente?, ¿cómo evitar entrega tardía del producto, reiteradas revisiones, errores de diseño y funcionalidad?, ¿cómo mantener una documentación íntegra, disponible y actualizada que describa exactamente lo que se quiere, finalidad del material, objetivos?

La solución estaría dada con la realización de un adecuado proceso de pruebas, actividad de la cual carece la universidad (UCI) por encontrarse aún en una fase de formación. Actualmente dicho proceso existe, pero no es el adecuado pues no se detectan en tiempo las deficiencias de la multimedia y por consiguiente no se evita el empleo de tiempo y recursos innecesarios.

*“Barrio Adentro al alcance de tus manos ”*, es una multimedia desarrollada por estudiantes de la UCI, específicamente de la facultad 8, que enfrenta también los problemas antes citados. Pertenece al programa “Atención integral de salud”, en el marco del convenio suscrito entre la República de Cuba y la República Bolivariana de Venezuela como parte del proyecto “Desarrollo de contenidos en tecnologías de la información” (2005).

A raíz de esta situación desencadenada en nuestra universidad, y en particular en la facultad que promueve el tema de esta investigación al comenzar a dar los primeros pasos en el desarrollo de este tipo de productos, cabe plantearse a modo de interrogante el siguiente **problema**: ¿Con un adecuado proceso de pruebas a la multimedia Barrio Adentro al alcance de tus manos, se evaluará eficientemente el estado de calidad de la misma?

Las pruebas de software son una parte importante del proceso de aseguramiento de la calidad. Según (Vegas, 2005) el objetivo de las pruebas no es asegurar la ausencia de defectos en un software, únicamente puede demostrar que existen. Las pruebas deben diseñarse para que saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo. Constituyen un elemento indispensable en el desarrollo de software, puesto que a partir de ellas se puede determinar la calidad de los productos implementados. A pesar de esto, no es difícil percibir como su importancia se ha subestimado y en ocasiones hasta ignorado.

El presente trabajo abarca un estudio de las pruebas a software y de manera particular a los de tipo multimedia, dando una panorámica de la importancia de su aplicación. De aquí que el **objeto de estudio** de la investigación se defina como el proceso de pruebas a productos Multimedia, siendo el **campo de**

**acción** el proceso de prueba para medir la calidad al producto multimedia “Barrio Adentro al alcance de tus manos”.

Como **ideas a defender** se plantea que una vez analizado el tema referente a pruebas de software de tipo multimedia y haber definido la estrategia específica a aplicar al producto señalado, se logrará detectar la mayor cantidad de deficiencias. Además, la documentación del proceso desarrollado puede constar como guía para futuras revisiones, garantizando la certificación de productos con un nivel de calidad óptimo para liberar.

El **objetivo general** de la investigación promueve la evaluación de la calidad a la multimedia “Barrio Adentro al alcance de tus manos” a través de un adecuado proceso de pruebas.

Como respuesta para dar cumplimiento al objetivo general, se trazaron una serie de **objetivos específicos**:

- ✓ Lograr que las pruebas a realizar al producto multimedia “Barrio Adentro al alcance de tus manos” sean las adecuadas.
- ✓ Realizar pruebas al producto multimedia “Barrio Adentro al alcance de tus manos”
- ✓ Analizar resultados obtenidos de las pruebas al producto multimedia “Barrio Adentro al alcance de tus manos”.

Las **tareas** a realizar en esta investigación que dan cumplimiento a los objetivos específicos incluyen:

- ✓ Estudiar estado del arte de las estrategias de prueba a software, haciendo especial énfasis en los productos de tipo Multimedia.
- ✓ Estudiar los diferentes niveles de pruebas existentes, así como las aplicables a productos multimedia.
- ✓ Determinar el proceso de pruebas a aplicar según las características de la multimedia “Barrio Adentro al alcance de tus manos”.
- ✓ Seleccionar las pruebas a aplicar en el producto.
- ✓ Definir la estrategia de pruebas a desarrollar para evaluar la calidad al producto multimedia.
- ✓ Diseñar las pruebas a aplicar a módulos y al sistema según niveles de pruebas.
- ✓ Ejecutar pruebas a partir de las necesidades y características del producto.

- ✓ Documentar el resultado de las pruebas.
- ✓ Evaluar proceso de pruebas realizado.
- ✓ Generar informe de costos sobre el desarrollo del proceso de pruebas al producto multimedia “Barrio Adentro al alcance de tus manos”.
- ✓ Realizar análisis sobre factibilidad del proceso de pruebas desarrollado.

La investigación consta de 3 capítulos que se extienden desde el diseño de las pruebas hasta el análisis de los resultados, posterior a su aplicación. El primero de estos titulado “Fundamentación teórica” comprende todo lo relacionado al objeto de estudio: conceptos importantes, relación directa entre las pruebas y la calidad del software, los objetivos y estrategias a seguir para su aplicación, así como los tipos de pruebas agrupadas por niveles y las técnicas para su diseño y en un último momento soluciones a situaciones existentes similares.

En un segundo capítulo “Descripción del proceso de pruebas” se abarca una panorámica del proyecto a evaluar unido al plan de pruebas que describe la ejecución del mismo. Comprende aspectos significativos como estrategia de pruebas, procedimiento y casos de prueba.

El tercer capítulo: “Resultados de las pruebas”, como un archivo de lo obtenido con su correspondiente análisis y valoración. Además, muestra un pequeño estudio de la factibilidad de la ejecución de la solución propuesta por razón de costos ocasionados.

Finalizando se exponen las conclusiones generales de la investigación, recomendaciones, bibliografía referenciada y bibliografía consultada.



## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción

Una de las grandes problemáticas que aún continúa vigente en el desarrollo de sistemas software es la evaluación de la calidad. Cuando se habla de calidad de software la referencia obligada es hacia las pruebas de software como vía óptima para la detección y prevención de errores durante todo el ciclo de desarrollo del mismo.

Las pruebas al software constituyen un proceso centrado en el objetivo de encontrar defectos a un software. Tratar de descubrir cada vez más errores, principalmente donde no han surgido hasta el momento, es el punto de partida de todo proceso de pruebas. Todos coinciden en que es mejor prevenir los errores que detectarlos y corregirlos, pues se ganaría en tiempo y recursos, sin embargo, lo cierto es que aún estamos muy lejos de alcanzar esa perfección en el trabajo.

Las pruebas no mejoran directamente la calidad del sistema pero sí dan una visión clara de las debilidades observadas y de los riesgos asociados, sobre los cuales se puede trabajar para perfeccionar.

En este capítulo se expone un estudio detallado sobre el estado del arte de este tema. Son analizados temas como: calidad y prueba a software, particularizando en la estrecha relación que los une; los objetivos y estrategias para la aplicación de las mismas, así como los niveles con los tipos de pruebas que lo integran y las características para su aplicación.

### 1.2 Calidad de Software

Desde los años 90 la calidad ha sido integrada en el proceso del desarrollo de software con mayor o menor intensidad según la criticidad del producto a obtener. Diversos han sido los criterios al respecto:

ParqueSoft ha identificado la Calidad como un factor diferenciador y estratégico para desarrollar y empoderar un mecanismo de apoyo continuo a la Gestión de Calidad tanto del Proceso Productivo como de los Productos de Software, estrategia orientada a definir un modelo de producción de software y

aseguramiento, basados en la adaptación de las mejores prácticas de la Industria y normas de gestión de calidad con reconocimiento internacional (Arena, 2006).

Según Pressman (2002), la calidad del software es la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente.

Para (Lovelley, 1999) la calidad es el conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas. La ausencia de defectos, la aptitud para el uso, la seguridad, la confiabilidad y la reunión de especificaciones son elementos que están involucrados en este concepto.

Teniendo en cuenta todo lo anterior, se puede concluir que la calidad del software está dada por el conjunto de características que le confieren su capacidad de satisfacer las expectativas del cliente, acorde al cumplimiento de una serie de requisitos establecidos previamente. Todo esto certificado bajo ciertas normas y métricas internacionales.

Al software como objeto inmaterial resulta muy difícil medir su calidad. Sin embargo, a la hora de dar un criterio sobre alguno se acude a términos como: el acercamiento a cero defectos, el cumplimiento de los requisitos intrínsecos y expresos, la satisfacción del cliente, etc.

La satisfacción del cliente es algo determinante aunque pueda parecer poco significativo. El cliente casi siempre tiene razón y es el que ubica el producto en los primeros puestos de la tabla de calidad por aclamación de los usuarios (Fernández, 2001).

La calidad del software debe implementarse a lo largo de todo el ciclo de vida, debe correr paralela desde la planificación del producto hasta la fase de producción del mismo como actividad de protección. Algunos grupos de desarrollo erróneamente creen que la calidad es algo en lo que deben preocuparse una vez que se ha generado el código.

### **1.2.1 Calidad y pruebas de Software**

La incorporación de las prácticas de calidad en el desarrollo de software tiene numerosos beneficios. En cuanto al cliente, se consigue una mejora de su satisfacción, de la fiabilidad del software, se reducen los errores en explotación y se logra el cumplimiento de los requisitos. Desde la perspectiva de una organización, se consigue verificar que se han implantado las características que indicó el usuario, asegurar que los procesos se aplican de la forma adecuada y que esos procesos se mejoran con el tiempo (Brualla, 2005).

Dada la importancia que el software ha adquirido en la actualidad ha dejado ya de ser una actividad artesanal, para convertirse en una actividad programada, planificada y con un ciclo de vida definido, otorgándole un carácter formal, sin embargo, uno de los aspectos más descuidados en este gran proceso de desarrollo es el de las pruebas, a pesar de ser el medio por el cual se puede asegurar la calidad del producto de software (Valdivia 2005).

Cuando se habla de mejorar la calidad del software la referencia obligada es hacia las pruebas. Aunque no mejoran directamente la calidad del sistema, dan una visión clara de las debilidades observadas y de los riesgos asociados, sobre los cuales se puede trabajar para perfeccionar, catalogándose así como un aspecto crucial dentro de la calidad. Este control de la calidad exige una serie de revisiones y pruebas utilizadas a lo largo del ciclo de desarrollo para asegurar que el producto cumple con los requisitos solicitados por el usuario.

### **1.3 Importancia de probar**

El desarrollo de software como actividad humana no es perfecto, aún cuando el progreso es sustancial. Es inherente a él toda una serie de deficiencias transparentes al programador, solo detectadas si probamos el producto. Las pruebas al software arrojan un panorama claro de las deficiencias del sistema detectadas y de los riesgos asociados a la organización. Esto, aunque directamente no mejore la calidad si permite a la administración tomar decisiones encaminadas a la asignación de recursos para elevar la misma.

A través de la prueba se ve la medida en que las funcionalidades del software se corresponden con las especificaciones establecidas y los datos que va arrojando constituyen un indicador de la fiabilidad del mismo.

## 1.4 Génesis de las pruebas

Cuando aparecieron los primeros grandes sistemas informáticos se incluyó a nivel metódico e imprescindible un, hasta entonces, nuevo proceso en la confección de los mismos: el proceso de prueba (D'Onofrio, 2002).

En sus inicios las pruebas de software se consideraban sólo responsabilidad de los programadores para encontrar fallas en los productos que hacían; hoy constituyen un proceso para evaluar la funcionalidad del software respecto a los requerimientos establecidos al inicio, donde participan: ingeniero del software, técnicos de prueba, especialistas de prueba y en algunos casos usuarios finales y responsables de la calidad, entre otros, en dependencia del tipo de prueba y la fase de desarrollo en que esté el software.

Si en un principio las pruebas se limitaban a detectar errores para corregirlos, actualmente la idea es prevenirlos, influyendo y controlando el diseño y desarrollo del software. Las pruebas deben ser empleadas como modelos de los requerimientos de la aplicación a construir, por tanto, deben existir especificaciones de pruebas dentro de las especificaciones de software. Esto deberá ser revisado como un conjunto por al menos un especialista de prueba, en caso de ser un grupo de probadores (D'Onofrio, 2002).

Es importante que el factor humano tenga conciencia acerca de la importancia de las pruebas, así evitará las reticencias inherentes a lo desconocido, conseguirá actitudes más positivas y facilitará la integración al proyecto, siendo responsables de su calidad y otorgándole resultados propios de un sistema de calidad.

En estos momentos se estima que la fase o proceso de pruebas representa más de la mitad del coste de un programa, pues necesita un tiempo similar al de la programación, lo cual conlleva a un alto costo económico, resultando esta etapa más cara que el propio desarrollo y diseño de los distintos programas que conforman el sistema. Esta, añade valor al producto que se maneja: todos los programas tienen errores que la fase de pruebas se encarga de descubrir, siendo este el valor añadido (Pons, 2006).

Las pruebas sólo deben arrojar un producto práctico con la calidad y funcionalidad requeridas, aún cuando este no sea el mejor en comparación con el desarrollo de otros similares.

## 1.5 Las pruebas de software

Cuando hablamos de pruebas de software nos referimos a la verificación dinámica del comportamiento de un sistema, basada en la observación de un conjunto seleccionado de ejecuciones controladas. Para los ingenieros de software las pruebas representan un gran reto pues intentan demoler lo que han hecho hasta ese momento.

Según la IEEE [IEEE, 1991] las pruebas constituyen “Una actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas, se observan o almacenan los resultados y se realiza una evaluación de algún aspecto del sistema o componente”.

Particularmente (Pressman, 2002) plantea que “Las pruebas del software son un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación.”

Las pruebas de software se ubican dentro del tema “Verificación y Validación (V&V)”. La verificación se refiere a las actividades que demuestran que el software funciona correctamente y la validación responde a las actividades que aseguran que el software se ajusta a los requisitos del cliente

Analizando todo lo expuesto anteriormente por diversos desarrolladores del tema se puede concluir que, las pruebas al software constituyen una actividad en la cual un sistema bajo ciertas condiciones específicas se ejecuta con el propósito de demostrar su incapacidad para ser liberado.

### 1.5.1 Objetivos de las pruebas

Probar, es el proceso de ejecutar un programa con el fin de encontrar errores o fallas. Comprende los objetivos siguientes (Quesada, 2005):

- ✓ La prueba es el proceso de ejecución de un programa con la intención de descubrir un error.
- ✓ Un buen caso de prueba es aquel que tiene una alta probabilidad de descubrir un error no encontrado hasta entonces.
- ✓ Una prueba tiene éxito si descubre un error no detectado hasta entonces.

Las pruebas no sólo se hacen al código, también a documentación y ayuda.

Los objetivos planteados anteriormente desacreditan la filosofía acerca de otorgarle éxito a una prueba cuando esta no descubre errores. Por el contrario, tiene éxito si encuentra un error no detectado hasta el momento (Pressman, 2002).

### 1.5.2 Principios de las pruebas

Las pruebas se rigen por una serie de principios. Una buena comprensión de estos facilitará el posterior uso de los métodos en un efectivo diseño de casos de prueba (Quesada, 2005).

- ✓ Hacer un seguimiento de las pruebas hasta los requisitos del cliente (trazabilidad).
- ✓ Deben ser planificadas antes de que empiecen.
- ✓ Plantear y diseñar las pruebas antes de generar ningún código.
- ✓ El principio de Parapeto es aplicable a la prueba del software (“donde hay un defecto, hay otros).
- ✓ Empezar las pruebas en módulos individuales y avanzar hasta probar el sistema entero, empezar por “lo pequeño” y progresar hacia “lo grande “.
- ✓ No son posibles las pruebas exhaustivas.
- ✓ Deben realizarse por un equipo independiente al equipo de desarrollo, para que sean más efectivas.
- ✓ Se deben evitar los casos de prueba no documentados ni diseñados.
- ✓ No deben realizarse planes de prueba suponiendo que prácticamente no hay defectos en los programas y, por tanto, dedicando pocos recursos a las pruebas.

Se puede decir que los principios constituyen para las pruebas, requerimientos a seguir para certificar su éxito: detectar la mayor cantidad de errores.

### 1.5.3 Tareas a realizar para probar software

La ejecución de las pruebas conlleva a la realización de una serie de actividades, descritas por (Vegas, 2005) como:

1. Diseño de las pruebas: Comprende la identificación de la técnica o técnicas de pruebas que se utilizarán para probar el software. Distintas técnicas de prueba ejercitan diferentes criterios como guía para realizar las pruebas.
2. Generación de los casos de prueba: Consiste en la confección de los distintos casos de prueba según la técnica o técnicas identificadas previamente. La generación de cada caso de prueba debe ir acompañada del resultado que ha de producir el software al ejecutar dicho caso para detectar un posible fallo en el programa. Los casos de prueba determinan un conjunto de entradas, condiciones de ejecución y resultados esperados para un objetivo particular. Cada técnica de pruebas proporciona unos criterios distintos para generar estos casos o datos de prueba.
3. Definición de los procedimientos de la prueba: Conlleva a una especificación de cómo se va a llevar a cabo el proceso, quién lo va a realizar y cuándo.
4. Ejecución de la prueba: Es el momento de aplicar los casos de prueba generados previamente e identificar los posibles fallos producidos al comparar los resultados esperados con los resultados obtenidos.

## 1.6 ¿Dónde radica el éxito de una prueba?

Las pruebas son una tarea tanto o más creativa que el desarrollo de software. Sin embargo, ha sido subvalorada por considerarse erróneamente como una tarea destructiva y rutinaria. No debe recurrirse a ella en una última etapa del desarrollo del software. El costo de detección de un error es mucho mayor entre más tarde sea detectado. Representa un gran incremento en el costo total del desarrollo. El imperativo en cada fase del ciclo es construir y probar enseguida. El éxito de la prueba está dado por varios aspectos (Patón, 2007):

- ✓ Cada caso de prueba debe definir el resultado de salida esperado que se comparará con el realmente obtenido.
- ✓ El programador debe evitar probar sus propios programas, pues desea (consciente o inconscientemente) demostrar que funcionan sin problemas.
- ✓ Además, es normal que las situaciones que olvidó considerar al crear el programa queden de nuevo olvidados al crear los casos de prueba.
- ✓ Se debe inspeccionar a conciencia el resultado de cada prueba, para así, poder descubrir posibles síntomas de defectos.

- ✓ Al generar casos de prueba, se deben incluir tanto datos de entrada válidos y esperados como no válidos e inesperados.
- ✓ Se deben evitar los casos desechables, es decir, los no documentados ni diseñados con cuidado.
- ✓ No deben hacerse planes de prueba suponiendo que, prácticamente, no hay defectos en los programas y por lo tanto, dedicando pocos recursos a las pruebas. Siempre hay defectos.
- ✓ La experiencia parece indicar que donde hay un defecto hay otros, es decir, la probabilidad de descubrir nuevos defectos en una parte del software es proporcional al número de defectos ya descubiertos.

La prueba no es una actividad sencilla, ni una etapa del proyecto en la cual se asegura la calidad. Es más bien, una acción que debe ocurrir durante todo el ciclo de vida para probar: funcionalidad de los primeros prototipos, estabilidad, cobertura y rendimiento de la arquitectura; probar el producto final, etc.

## 1.7 Proceso de pruebas

Según aumenta la complejidad de los sistemas software y la demanda de calidad, se hacen necesarios procesos y métodos que permitan obtener buenos conjuntos de pruebas del sistema. La importancia de los costos asociados a los errores, promueve la definición y aplicación de un proceso de pruebas minuciosas y bien planificadas (Reina, 2006).

Un proceso no es más que la sucesión de pasos y decisiones que se siguen para realizar una determinada actividad o tarea (Pons, 2006). Por tanto, puede definirse como proceso de pruebas de manera general, al grupo de actividades y entregables que se siguen para realizar pruebas a un software, garantizando un producto confiable, sin errores y con la mayor calidad posible. Se recomienda su integración dentro del propio desarrollo del producto y no de manera aislada.

Consta generalmente de cuatro fases: la fase de diseño de pruebas, la fase de codificación, la fase de ejecución y la fase de análisis de los resultados. Dicho proceso toma como punto de partida los requisitos y, a partir de ellos genera los resultados y construye las pruebas (Reina, 2006). Un proceso de prueba es enfocado sobre la lógica interna del software y las funciones externas.



## 1.8 Estrategia de pruebas

El desarrollo de pruebas requiere la definición de una estrategia que enmarque una serie de actividades para dar cumplimiento a los objetivos de la misma en el control de la calidad. Se considera necesaria debido al creciente aumento de la complejidad que han adquirido en los últimos años los productos software, dicha complejidad hace que estos sistemas requieran ser sometidos a pruebas con el propósito de garantizar su calidad.

Se agrupan en dos direcciones diferentes (Torres, 2003), (Reyes, 2003):

Las que analizan el comportamiento interno del sistema, son las llamadas pruebas de caja blanca o caja de cristal. Comprueban la lógica interna de los componentes de software a través de la inspección al código.

Aquellas que trabajan con la interfaz del sistema, denominadas pruebas de caja negra. Estas verifican los dominios de entrada y salida del programa o programas para descubrir errores de funcionalidad, comportamiento y rendimiento.

### 1.8.1 Características generales de la Estrategia de Prueba

Una estrategia de prueba del software debe ser elaborada flexiblemente para que pueda ser reutilizable en los diferentes proyectos, adaptándola a las características particulares del mismo. Sin embargo, todas se componen por una serie de pautas generales: [Pressman, 2000]

- ✓ Las pruebas comienzan a nivel de módulo y trabajan hacia fuera, 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 óptima depende de la aplicación final y de las necesidades del usuario para accesos de prueba, velocidad, costes y fiabilidad. Estas proporcionan a las organizaciones, una ventaja estratégica para mejorar el rendimiento y la calidad ofrecida (Verma, 2003).

## 1.9 Plan de prueba

El plan de prueba es el encargado de explicitar el alcance, enfoque, recursos requeridos, calendario, responsables y manejo de riesgos de un proceso de pruebas (Teruel, 2001). Está constituido por un conjunto de pruebas. Estas deben especificar: tipo de propiedades a evaluar (corrección, robustez, fiabilidad), cómo medir el resultado, ejecución detallada de las mismas y resultado esperado para valorar luego si es o no acorde a lo esperado.

El plan de prueba constituye la piedra angular para el éxito de la puesta en práctica de un proceso de pruebas que permita liberar un software de mejor nivel. A pesar de no existir un proceso de prueba estándar para evaluar cualquier producto, la mayoría de los proyectos informáticos, ya sean de nuevo desarrollo o de mantenimiento de aplicaciones, trabajan con modelos comunes para la realización de las pruebas.

### 1.9.1 Posibles componentes de un Plan de pruebas

Este marco consta de componentes como guía para la elaboración de un posible plan de pruebas según (González, 2002):

- I. Descripción de Aspectos Generales: Esta sección establece el alcance y el objetivo del Plan de pruebas. Es aquí donde se describen los aspectos fundamentales del esfuerzo que se hará para probar una aplicación computacional, independientemente de las características y tamaño que ésta pueda tener.
  - ✓ Objetivo: Describir por qué el Plan de pruebas fue desarrollado (objetivos). Puede incluir requerimientos de documentación, definición de estrategias de prueba, identificación de recursos, estimación de plazos y proyección de entregables.
  - ✓ Entorno o Marco: Explicar los eventos que dan origen al Plan de pruebas.

- ✓ Arquitectura Técnica: Diagramación de las partes que componen el sistema bajo prueba (almacenamiento de datos y conexiones para su transferencia). Se debe documentar tanto las capas, como la presentación / interfaz del usuario, la base de datos, los emisores de informes, etc.
- ✓ Especificaciones de software (SW) y hardware (HW): Lista individualizada de todo el SW y el HW que utiliza la aplicación, incluyendo proveedores y versiones.
- ✓ Alcance: Describe brevemente los recursos que el plan requiere, las áreas de responsabilidad, las etapas y los riesgos potenciales.
- ✓ Información del proyecto: Identifica toda la información que está disponible en relación con el proyecto. La documentación del usuario, el plan de proyecto, las especificaciones del producto, los materiales para entrenamiento y las revisiones ejecutivas, son algunos ejemplos de información del proyecto.

II. Descripción de Requerimientos: Lista de los requerimientos que serán probados. Cualquier requerimiento no incluido en esta lista estará fuera del alcance de las pruebas. Puede incluirse la lista completa en esta sección o hacer referencia a otro documento que contenga la información.

- ✓ Requerimientos Funcionales
- ✓ Requerimientos de Diseño
- ✓ Requerimientos de Integración
- ✓ Otros Requerimientos

III. Definición de la Estrategia de Pruebas: Describir como los objetivos de la prueba serán alcanzados para cada uno de los tipos de pruebas que hacen parte del plan:

- ✓ Unitarias.
- ✓ Funcionales.
- ✓ De integración.
- ✓ De sistema.
- ✓ De resistencia (stress).
- ✓ De rendimiento (performance)
- ✓ De configuración y/o de instalación.

Para cada subconjunto requerido o definido como necesario, debe detallarse lo siguiente:

- ✓ Objetivo: El objetivo global de esta estrategia debe alcanzarse.
- ✓ Técnica: Especifica cómo los casos de prueba serán desarrollados, el instrumento o herramienta usado para almacenarlos y donde pueden ser encontrados; cómo ellos serán ejecutados y los datos que serán usados.
- ✓ Consideraciones Especiales: Definir datos necesarios de la prueba, condiciones de ambiente u otros aspectos.
- ✓ Casos de prueba: Lista detallada o una referencia a los casos de prueba que serán ejecutados.
- ✓ Criterios de Término: Registrar los criterios que serán usados para determinar la aprobación o rechazo de pruebas y la acción que debe ser tomada con base en los resultados de la prueba.
- ✓ Supuestos: Describir cualquier otro proyecto externo o asuntos que pueden impactar en la eficacia o la oportunidad del esfuerzo de prueba.
- ✓ Herramientas: Documentar los instrumentos o herramientas a usar (proveedor, versión).

IV. Recursos Requeridos: Identificar los roles y las responsabilidades que serán requeridas para la ejecución del Plan de Pruebas.

V. Plan de proyecto: Desarrollar un plan que muestre las fases, tareas y recursos.

VI. Calendario y Plazos: Documentar el plazo en el cual la aplicación a probar estará disponible para pruebas y el tiempo estimado para ejecutar los casos de prueba.

VII. Definición de los Entregables: Listar cualquier entregable asociado con el esfuerzo de pruebas y localización de copias de los mismos (Plan de Pruebas en sí mismo, escenarios para prueba, casos de prueba y el plan de proyecto).

VIII. Seguimiento y Reporte de Defectos: Documentar el instrumento y el proceso usado para registrar y rastrear los defectos. Listar los informes generados. Describa una calificación, categoría o clasificación que se usará para identificar o priorizar los defectos.

IX. Aprobación del Plan: Debe ser revisado por todas las partes responsables de su ejecución y aprobado por el equipo de prueba, el jefe del proyecto y el gerente de desarrollo.

X. Documentación de los Resultados: Documentar los resultados y mediciones una vez terminado el esfuerzo de prueba. Identifique cualquier discrepancia entre el plan y la puesta en práctica real y como fueron manejadas.

## 1.10 Caso de prueba

Se define como caso de prueba al conjunto de entradas, condiciones de ejecución y resultados esperados, desarrollados para un objetivo particular como, por ejemplo, ejercitar un flujo de un programa o verificar el cumplimiento de un determinado requisito. También se puede “referir a la documentación en la que se describen las entradas, condiciones y salidas de un caso de prueba” [IEEE, 1990].

Cada caso de prueba está asociado a un escenario de un caso de uso en particular y extiende o amplía su descripción, permitiendo validar si el software desarrollado cumple con las funciones descritas en los casos de uso. Para su elaboración se necesita de los documentos de requisitos, de arquitectura o de cualquier otro que muestre lo desarrollado y lo susceptible a tener errores.

Consta de tres bloques de información, los cuales deben quedar bien documentados (Maña, 2000):

1. Propósito de la prueba.
2. Pasos de ejecución de la prueba.
3. Resultado que se espera.

Abarcar de manera explícita todos estos bloques puede parecer un poco laborioso y con frecuencia tedioso, lo que torna aburrida la fase de pruebas. A pesar de esto, es importante no incurrir en ello pues se caería en un error mayor, subestimar una tarea imprescindible durante el desarrollo de software: las pruebas.

### 1.10.1 Diseño de Casos de prueba

Resulta vital el desarrollo de casos de prueba completos y no los que parecen correctos, producto de una elaboración en la cual se le restó importancia a la prueba. El diseño de casos de prueba está totalmente mediatizado por la imposibilidad de probar exhaustivamente el software.

Durante su elaboración los diseñadores deben estar impregnados de un alto sentido de creatividad, de manera que puedan abarcar la máxima probabilidad de detección de errores con la menor cantidad de esfuerzo, recursos y tiempo disponible.

Ya que no se pueden probar todas las posibilidades de funcionamiento del software, la idea fundamental para el diseño de casos de prueba consiste en elegir algunas de ellas que, por sus características, se consideren representativas del resto. Así, se asume que, si no se detectan defectos en el software al ejecutar dichos casos, podemos tener un cierto nivel de confianza (que depende de la elección de los casos) en que el programa no tiene defectos. La dificultad de esta idea es saber elegir los casos que se deben ejecutar.

## 1.11 Métodos de pruebas

Los métodos de prueba del software tienen el objetivo de diseñar pruebas que descubran diferentes tipos de errores con menor tiempo y esfuerzo. El diseño de pruebas a software puede requerir tanto esfuerzo como el propio diseño del producto.

Generalmente los métodos de caja blanca y caja negra son los más utilizados a la hora de probar software (Pressman, 2002), para llegar a un método que valide la interfaz del software y asegure selectivamente que el funcionamiento interno del mismo es correcto.

### 1.11.1 Prueba de Caja Blanca

Conocida también como prueba de caja de cristal, las pruebas de caja blanca se detienen fundamentalmente en “cómo” está hecho el código. Son mucho más amplias, normalmente se denominan pruebas de cobertura o pruebas de caja transparente (ver Fig.1). Al total de pruebas de caja blanca se le llama cobertura. La cobertura es un número porcentual que indica cuanto código del programa se ha probado. Se basan en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles. Permite examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado.

Según (Pressman, 2002) permiten obtener casos de prueba que:

- ✓ Garanticen que se ejercitan por lo menos una vez todos los caminos independientes de cada módulo.
- ✓ Ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa.
- ✓ Ejecuten todos los bucles en sus límites y con sus límites operacionales.

- ✓ Ejercitan la estructura interna de datos para asegurar su validez.

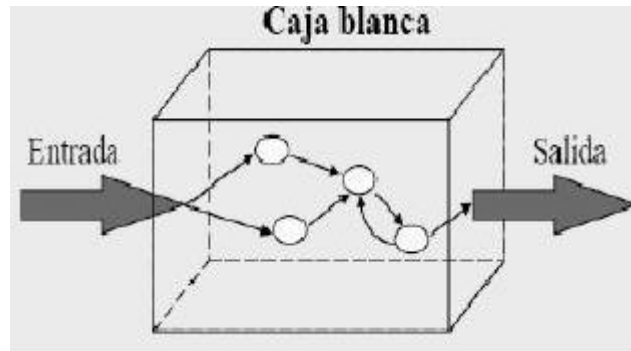


Fig. 1 Representación de técnica de prueba de Caja blanca (Vegas, 2005).

#### 1.11.1.1 Métodos de prueba de Caja blanca.

Con el objetivo de diseñar casos de prueba que saquen a la luz la mayor cantidad de errores, se listan a continuación los diferentes métodos de pruebas de Caja blanca existentes según (Pressman, 2002):

##### ✓ Prueba del camino básico

La prueba del camino básico es una técnica de prueba de Caja blanca introducida por Tom McCabe. Permite obtener una medida de la complejidad lógica de un diseño procedimental para su uso posterior como guía en la definición de un conjunto básico de caminos de ejecución que garanticen la ejecución de al menos una vez cada sentencia del programa durante la aplicación de la prueba.

Usa la notación de grafos de flujo para representar el flujo de control lógico. Compuesto por nodos, aristas y regiones. El grafo de flujo o grafo de programa es una notación para representar el flujo de control lógico. Está compuesto por tres componentes fundamentales:

- **Nodo:** Es representado por un círculo, se denomina nodo del Grafo de Flujo, el cual puede representar una o más sentencias de código fuente. Un solo nodo puede corresponder a una secuencia de procesos o a una sentencia de decisión. Pueden existir nodos que no se asocien, utilizados principalmente al inicio y final del grafo.

- **Aristas** (o enlaces): Es asociado a flechas en el grafo que representan el flujo de control, y análogas a las representadas en un diagrama de flujo. Una arista debe terminar en un nodo, aun cuando el nodo no represente ninguna sentencia procedimental.
- **Regiones:** Áreas delimitadas por nodos y aristas, se incluye también el área exterior del grafo. Las regiones se enumeran y es equivalente a la cantidad de caminos independientes del conjunto básico de un programa.

#### ✓ **Prueba de condición**

El método de prueba de condiciones se centra en la prueba de cada una de las condiciones del programa. Su propósito es detectar cualquier error en el programa, no solamente en las condiciones. Ejercita las condiciones lógicas contenidas en el módulo de un programa.

Trabaja con condición simple (variable lógica o una expresión relacional) y compuesta (formada por dos o más condiciones simples, operadores lógicos y paréntesis.)

#### ✓ **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. Se recomienda su uso para la selección de caminos de prueba de un programa que contenga sentencias if o de bucles anidados.

#### ✓ **Prueba de bucles**

Se centra exclusivamente en la validez de las construcciones de bucle dado su uso en la gran mayoría de los algoritmos de implementación de software. Se definen cuatro clases diferentes de bucles: bucles simples, bucles concatenados, bucles anidados y bucles no estructurados. Cada una tiene su propia estrategia y condiciones de aplicación dada su particularidad.

### **1.11.2 Prueba de Caja Negra**

Las pruebas de caja negra o pruebas de comportamiento, son aquellas que se enfocan directamente en el exterior del módulo, sin importar el código (ver Fig.2). Permiten al ingeniero de software obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Ignora intencionadamente la estructura de control y centra su atención en el campo de la



información. Este tipo de pruebas no es aplicable a los módulos que trabajan en forma transparente al usuario.

Esta prueba no constituye una alternativa a las técnicas de prueba de la Caja Blanca, sino una actividad complementaria que intenta descubrir diferentes tipos de errores a los encontrados en los métodos de la Caja Blanca. El diseño de estas pruebas tiene el propósito de detectar (Pressman, 2002):

- ✓ 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.

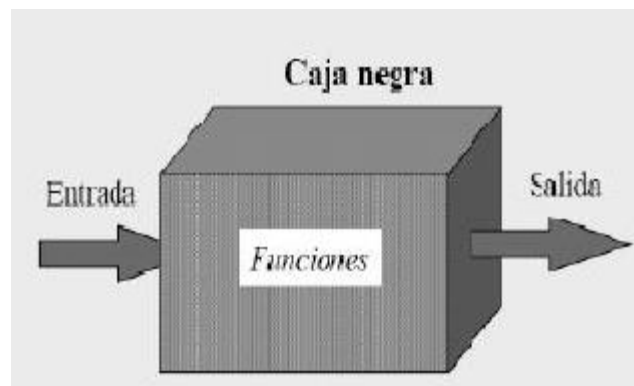


Fig. 2 Representación de técnica de prueba de Caja negra. (Vegas, 2005)

#### *1.11.2.1 Métodos de prueba de Caja negra.*

A continuación se exponen los diferentes métodos de prueba de Caja negra descritos por (Pressman, 2002) para validar la funcionalidad del sistema sin entrar a analizar su ejecución interna.

- ✓ **Métodos de prueba basados en grafos.**

El método de prueba basada en grafos analiza el software para definir objetos y relaciones que conecten a los objetos para su posterior uso en los casos de prueba. Pretende evaluar que todos los objetos tienen

entre ellos las relaciones esperadas. Para ello crea un grafo de objetos importantes con sus relaciones y a partir de él, diseñar pruebas que cubran el grafo ejercitando sus relaciones en la búsqueda de errores.

#### ✓ **Partición equivalente**

La partición equivalente es un método de Caja negra que divide el dominio de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Se basa en una evaluación de las clases de equivalencia para una condición de entrada.

Una clase de equivalencia representa un conjunto de estados válidos o no válidos para condiciones de entrada. Una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica (Pressman, 2002).

Las clases de equivalencias se pueden definir de acuerdo a 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.

Aplicando estas directrices para la obtención de clases de equivalencias, se pueden desarrollar y ejecutar casos de prueba para cada elemento de datos del campo de entrada. Los casos de prueba se seleccionan de forma que se ejerciten el mayor número de atributos de cada clase de equivalencia a la vez.

#### ✓ **Análisis de valores límites**

La técnica de valores límites analiza las fronteras de los campos de entrada. Para ello nos lleva a la elección de casos de prueba que ejerciten los valores límites. Esta técnica de diseño de casos de prueba complementa a la partición equivalente. En lugar de seleccionar cualquier elemento de una clase de equivalencia, toma los casos de prueba de los extremos de la clase. No se centra solo en las condiciones de entrada, también tiene casos de prueba para el campo de salida.

### ✓ **Tabla ortogonal**

El método de prueba de la tabla ortogonal tiene uso en problemas en que el dominio de entrada es relativamente pequeño pero demasiado grande para realizar pruebas exhaustivas. Es muy útil a la hora de encontrar errores asociados con fallos relacionados - una categoría de error asociada con defectos de la lógica dentro de un componente software.

## **1.12 Niveles de Prueba**

Existen diferentes niveles de pruebas, los cuales se realizan en determinados momentos del ciclo de vida del software. Estos organizan e integran el conjunto de técnicas y pruebas a realizar. Según (Mañas, 2002), (Molina, 2005) se distinguen los siguientes:

### ✓ **Pruebas unitarias**

Son pruebas de menor escala. Son las encargadas de probar el correcto funcionamiento de un módulo de código. Cuando estos módulos son extensos o complejos se dividen para probar objetivamente partes más pequeñas, este tipo de pruebas es la más común. Sirven para asegurar que cada uno de los módulos funcione correctamente por separado. Las pruebas unitarias no descubrirán todos los errores del código. Por definición, sólo prueban las unidades por sí solas.

### ✓ **Pruebas de integración:**

La prueba de integración se lleva a cabo durante la construcción del sistema, involucran a un número creciente de módulos y terminan probando el sistema como conjunto. Su realización es después de probar los componentes individuales del programa e integrarlos. Se desarrollan a partir de las especificaciones de los componentes y del sistema.

Una vez culminada la prueba de integración, el software está completamente ensamblado como un paquete, se han encontrado y corregido los errores de interfaz y puede comenzar una serie final de pruebas del software: la prueba de validación.

### ✓ **Prueba del Sistema:**

Es el proceso de integrar el sistema y el hardware para verificar que el sistema reúna los requerimientos especificados. Una prioridad de este sistema es que debe concentrarse más en la capacidad del sistema que en la capacidad de los componentes, es decir, en el uso de la interacción de las funciones, que en probar los detalles de implementación; otra prioridad es probar las situaciones críticas en lugar de probar casos especiales. La preocupación principal es la compatibilidad de los módulos individuales. La persona encargada de hacer las pruebas debe encontrar las áreas en donde los módulos hayan sido diseñados con especificaciones distintas para la longitud y tipo de datos y los nombres de los elementos de dato. Es la combinación sistemática con una ejecución de componentes de subsistemas.

#### ✓ **Pruebas de Aceptación**

Son pruebas realizadas por el cliente. Son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario. Se realizan después del desarrollo del producto, una vez que este ha sido terminado o integrado, pueden existir casos sobre una versión del producto o una iteración funcional aceptada previamente por el usuario.

Determina la conformidad del cliente antes de que el programa le sea entregado como una versión final. Estas pruebas son muy importantes, ya que definen el paso a nuevas fases del proyecto como: despliegue y mantenimiento. Tienen la particularidad de descubrir errores desapercibidos hasta el momento por los desarrolladores que llevaron a cabo todo el proceso de pruebas anterior.

#### ✓ **Pruebas de Regresión:**

Es un conjunto de pruebas que se ejecutan nuevamente. Su objetivo es verificar que cambios ocurridos en un componente del sistema no provocan errores adicionales en otros componentes del mismo que ya han sido probados. Asegura que los cambios no introducen un comportamiento no deseado o errores adicionales.

### **1.13 Tipos de pruebas por niveles**

Todas las pruebas en su conjunto, permiten verificar y validar un software, independientemente de las características y el entorno de desarrollo, de los recursos necesarios y disponibles para la evaluación y de factores vinculados al proceso de desarrollo del producto. A continuación se describen brevemente diferentes tipos de pruebas agrupadas según nivel correspondiente.

### **1.13.1 Pruebas a Nivel de unidad**

Estas pruebas centran el proceso de verificación en la parte individual de unidades (componente o módulo) separadas de un sistema de software.

#### **✓ Prueba de interfaz**

La prueba de interfaz se realiza una vez que han sido integrados los módulos o subsistemas para crear un sistema completo. Cada módulo o subsistema tiene una interfaz definida que es llamada por otros componentes del programa. Detectan los errores introducidos en el sistema por errores de las interfaces o suposiciones inválidas a cerca de las interfaces.

Estas pruebas analizan la interfaz del módulo para probar si la información fluye correctamente hacia y desde la unidad de programa que está siendo probada (Pressman, 2002).

#### **✓ Prueba de estructura de datos**

El análisis de la estructura de datos locales tiene el propósito de comprobar si los datos que se mantienen temporalmente conservan su integridad durante todos los pasos de ejecución del algoritmo (Pressman, 2002).

#### **✓ Condiciones límites**

Se prueban las condiciones límites para comprobar la correcta ejecución del módulo en los límites establecidos de condiciones como restricciones de procesamiento (Pressman, 2002).

#### **✓ Caminos independientes**

Una vez que se ejerciten todos los caminos independientes o básicos de la estructura de control del módulo se asegura probar con su ejecución, al menos una vez, todas las sentencias del mismo (Pressman, 2002).

### **1.13.2 Pruebas a Nivel de Integración**

Consiste en realizar pruebas para verificar que un gran conjunto de partes (módulos) del software funcionan juntas. Las partes se van incrementando a medida que se conforma el todo. La integración puede realizarse desde un extremo u otro y continuar hacia el extremo opuesto, según la prioridad.

#### **✓ Integración descendente**

La prueba de integración descendente parte del módulo de control principal (mayor nivel) para luego ir incorporando los módulos subordinados progresivamente. La selección del próximo módulo a incorporar es libre, solo exige que haya sido probado anteriormente. Existen principalmente dos métodos para la incorporación de módulos (Pressman, 2002):

- Primero en profundidad: Primero se mueve verticalmente en la estructura de módulos.
- Primero en anchura: Primero se mueve horizontalmente en la estructura de módulos.

#### ✓ **Integración ascendente**

En el caso de la prueba ascendente se integran y prueban los componentes de los niveles más bajos de la estructura del programa antes del desarrollo de los niveles altos. La integración de los módulos de abajo hacia arriba, posibilita la disponibilidad del proceso requerido de los módulos subordinados y la eliminación de necesidad de resguardos (Pressman, 2002).

#### ✓ **Prueba de regresión**

La prueba de regresión es realizada después de aplicada la prueba de integración, en la medida que se añade un nuevo módulo. Abarcan un conjunto de pruebas realizadas anteriormente para comprobar que los cambios introducidos por el módulo añadido no han provocado daños colaterales, no surgidos hasta el momento. Tienden a aumentar a medida que se progresa en la integración de los módulos, por lo que se sugiere sean diseñadas para incluir sólo aquellas pruebas que traten una o más clases de errores en cada una de las funcionalidades del programa (Pressman, 2002).

#### ✓ **Prueba de humo**

Se aplica en productos software desarrollados por paquetes, generalmente proyectos críticos por tiempo. Con su ejecución se prueba al software de principio a fin y aunque no sea exhaustiva descubre importantes errores. Permite minimizar los riesgos de integración, perfeccionar la calidad del producto final, simplificar el diagnóstico y la corrección de errores, además de seguir con la observación del progreso final (Pressman, 2002).

### **1.13.3 Pruebas a Nivel de sistema**

La realización de pruebas de sistema busca verificar la funcionalidad interna del mismo, así como, el cumplimiento de las especificaciones recogidas en los requerimientos.

### ✓ **Pruebas de funcionalidad**

Se centran en las funciones, entradas y salidas. Por lo impracticable de probar el software para todos los juegos de datos, es necesaria la elección de buenos casos de prueba.

El caso de prueba de funcionalidad elegido debe reducir en sí, el número de otros casos necesarios, de manera que se reduzcan el total de casos. Con esto el caso ejecutará el máximo número de posibilidades de entradas diferentes y quedarán reducidos el total de casos de manera que el trabajo pueda ser más preciso.

También debe abarcar un conjunto extenso de otros posibles casos, de manera que la ausencia o presencia de defectos, no se correspondan sólo al conjunto específico de entradas de prueba, sino que los resultados sean también de conjuntos similares.

### ✓ **Prueba de esfuerzo:**

Comprueban el desempeño y fiabilidad del sistema una vez integrado. Se basan en la respuesta del sistema ante el procesamiento de la carga para la cual fue diseñado. La prueba consiste en el incremento de la carga de forma constante hasta que el desempeño del sistema sea inaceptable.

### ✓ **Prueba de rendimiento:**

Determina el tiempo de respuesta, el espacio que ocupa el módulo en disco o en memoria, el flujo de datos que genera a través de un canal de comunicaciones, etc. En resume, prueba el rendimiento del software en tiempo de ejecución dentro de un sistema integrado (Pressman, 2002).

### ✓ **Prueba de requerimientos:**

El objetivo de esta prueba es buscar discrepancias entre los requerimientos y la ejecución del software. Se basa en el análisis de los requerimientos y una inspección en la cual se busca evaluar la consistencia, completitud y factibilidad de estos, tanto individualmente como juntos para facilitar el análisis de errores y la generación de casos de prueba. Adicionalmente los requerimientos deben ser revisados y validados por los distintos actores involucrados con el sistema (stakeholders). Se inician durante el proceso de elicitación de los requisitos, mediante una revisión al documento de especificación de los requerimientos, con la lista de chequeo general del documento y la lista de chequeo de requerimientos (Teruel, 2001).

### ✓ **Pruebas de carga**

Es la encargada de probar el funcionamiento del software bajo condiciones extremas. Estudia la especificación del software, las funciones que debe realizar, las entradas y las salidas analizando los Valores Límite (AVL) (Ruztort, 2004), (Jara, 2004).

✓ **Prueba de recuperación**

Esta prueba fuerza al software al fallo de diferentes maneras, verificando que la recuperación se realice apropiadamente. La recuperación puede ser automática o requerir la intervención humana. En caso de ser automática, el análisis recae sobre la corrección de la inicialización, sobre los mecanismos de recuperación del estado del sistema, de los datos y del proceso de rearranque. En el otro caso se analizan los tiempos medios de reparación, para determinar si están dentro de los límites aceptables (Pressman, 2002).

✓ **Prueba de seguridad**

Comprueba la respuesta de los mecanismos de protección incorporados al sistema ante acciones inapropiadas sobre este. La idea es probar al sistema directamente en sus invulnerabilidades (Pressman, 2002).

✓ **Prueba de resistencia (stress)**

Están diseñadas para enfrentar al programa a condiciones anormales. La prueba ejecuta un sistema de manera que demande recursos en cantidad, frecuencia o volúmenes anormales. Una variante de prueba de resistencia lo constituye una técnica denominada prueba de sensibilidad (Pressman, 2002).

### ***1.13.5 Pruebas a nivel de aceptación***

Las pruebas de aceptación buscan la validación de los requerimientos por parte del cliente. Estas son básicamente pruebas funcionales sobre el sistema completo. Se distinguen las siguientes:

✓ **Prueba alfa.**

Se lleva a cabo, por un cliente, en el lugar de desarrollo. Se usa el software de forma natural con el desarrollador como observador del usuario. Las pruebas alfa se llevan a cabo en un entorno controlado. Para que tengan validez, se debe primero crear un ambiente con las mismas condiciones que se



encontrarán en las instalaciones del cliente. Una vez logrado esto, se procede a realizar las pruebas y a documentar los resultados (Suárez, 1999).

✓ **Prueba beta.**

Se lleva a cabo por los usuarios finales del software en los lugares de trabajo de los clientes. A diferencia de la prueba alfa, el desarrollador no está presente normalmente. Esto hace a la prueba beta una aplicación “en vivo” del software, en un entorno que no puede ser controlado por el desarrollador. El cliente registra todos los problemas (reales o imaginarios) que encuentra durante la prueba beta e informa a intervalos regulares al desarrollador (Suárez, 1999).

Como resultado de los problemas informados durante la prueba beta, el desarrollador del software lleva a cabo modificaciones y así prepara una versión del producto de software para toda la clase de clientes.

### **1.13.6 Otras**

Existen pruebas que aunque no evalúan particularidades de los niveles de pruebas existentes, cubren aspectos que apuntan también hacia la obtención de un producto con un adecuado nivel de calidad.

✓ **Prueba de solidez**

Comprueban la correspondencia entre el tema tratado, el texto y el resto de los medios que aparecen en cada pantalla y el guión de contenidos o medias. Analiza además ortografía, calidad en los medios que se muestran, ya sea sonidos o imágenes. De manera general vela por el cumplimiento de las pautas trazadas durante el diseño (Sacha, 2006).

✓ **Prueba a la documentación**

Son pruebas necesarias dentro del plan de pruebas de cualquier software. Comprueban la integridad de la documentación. Se basa en la correspondencia entre el manual de usuarios y lo pautado en la documentación. Se puede enfocar en dos fases. Como primera la revisión e inspección a la documentación, para comprobar la claridad editorial. Como segundo momento, la prueba en vivo, que utiliza la documentación junto al uso del programa real. Esta última puede realizarse usando técnicas análogas a métodos de prueba de caja negra (Pressman, 2002).

✓ **Prueba de interfaz gráfica de usuario (IGU)**

La interfaz gráfica constituye el vínculo entre el usuario y el programa. Las actuales IGU se diseñan bajo la misma apariencia y filosofía. Esto ha permitido obtener una serie de pruebas estándar, para evaluarlas: las pruebas de IGU. Están asociadas a la evaluación de los atributos de la interfaz (pantalla), que hacen que sea más atractiva al usuario y fácil de usar. Dígase elementos como: botones, menús emergentes operantes con el mouse, imágenes, ventanas, etc., a los cuales pueden creárseles directrices (Feijoo, 2002).

## **1.14 Multimedia**

En la década de los 80, el mundo informático expuso a luz otro de sus extraordinarios sistemas aplicados, los software de tipo Multimedia, material que, con finalidad didáctica, lúdica o informativa, se está diseñando en la actualidad.

Según (Fernández, 2000), (Espinosa, 2000), (Sánchez, 2000), "la **multimedia** es la integración de diferentes tipos de medios en un sólo documento [...] texto, gráficos, sonido digitalizado, vídeo y otros tipos de información". Bajo esta descripción hablar de multimedia, ya sea educativa, informativa o interactiva es referirse a documentos informáticos, sea bajo soporte físico (CD-ROM, DVD,...) sea bajo soporte on-line.

### **1.14.1 Pruebas a Multimedia**

Las producciones de software multimedia están enfrentando un desarrollo velozmente incremental, por sus excelentes resultados en las más diversas aplicaciones. Como producto comercial que se inserta en un exigente mercado mundial, necesita para su liberación una calidad bien certificada. Basado en esto y entendiendo la evaluación como una valoración orientada a la toma de decisiones y a la mejora, es una actuación necesaria en dichas producciones.

La validación de las producciones requiere criterios de evaluación dependiendo de la finalidad del material y los objetivos que se persigue con él. Este proceso evaluativo es una de las etapas más significativas que requiere de estándares para su utilidad y viabilidad (Graells, 2005).

Las pruebas representan un momento imprescindible dentro del ciclo de desarrollo de los productos multimedia, como caso particular de software. Constituyen un medio de evaluación a dichos productos, a partir de la detección y prevención de fallas o errores en los mismos.

Como expusieron (Fernández, 2000), (Espinosa, 2000), (Sánchez, 2000), todo diseño de un material multimedia debe contener los objetivos claramente delimitados en función del género, la edad de los usuarios y de los clientes, el contexto, etc., destacando que nunca se pueden dar por supuesto estos objetivos por muy obvios que parezcan pues la simple elaboración del material en soporte electrónico o en línea constituye una de las principales decisiones a tomar y por tanto forma parte de la naturaleza intrínseca de estos materiales. En relación con esto, hay que tener en cuenta que la definición del objetivo del proyecto no debe presuponer su adecuación al mundo digital, sino más justificarla.

### ***1.14.2 Estrategias de pruebas a Multimedia***

Actualmente la garantía de calidad al software multimedia le llega de diversas vías. Cada centro productor evalúa sus resultados de una manera diferente, ajustada a las especificaciones que considera de manera particular más relevantes a la hora de poner la multimedia en manos del cliente. Las estrategias descritas a continuación resumen la diversidad de criterios definidos tanto nacional como internacionalmente respecto al tema.

#### **✓ Universidad de las Ciencias Informáticas (UCI)**

En la universidad de las Ciencias Informáticas, la experiencia en el desarrollo de productos multimedia, es pequeña, si se tiene en cuenta que el centro surgió en el año 2002. Por la creciente demanda que llega a ella de estos software, tanto nacional como internacionalmente, ha tenido que trazarse estrategia para alcanzar resultados satisfactorios en la entrega de los mismos. La institución cuenta con un grupo de estudiantes vinculados a un proyecto de calidad, responsable de certificar la misma en los productos a través de la realización de pruebas. Su particularidad radica en asignar a un grupo de miembros de este proyecto el producto multimedia a probar. Se solicita a los desarrolladores toda la documentación del producto, dígame: levantamiento de requisitos, especificación de casos de uso, guión técnico, etc. A partir de esta documentación se realizan las pruebas a la multimedia (pruebas de funcionalidad generalmente) donde se comprueba la correspondencia entre lo esperado y lo obtenido. El siguiente paso consiste en

entrevistar a especialistas del tema abordado en el producto para comprobar que la información es válida y está acorde a los usuarios finales.

Toda esta estrategia de prueba está basada en una metodología y la norma de calidad ISO 9126.

No obstante, esta solución no ha proporcionado la calidad requerida que espera darle a sus productos la institución.

### ✓ **CITMATEL**

CITMATEL, antiguo **CEDISAC** surgido al final de la década de los 80 es un hoy un consolidado proveedor de productos multimedia. En sus inicios el método para certificar sus productos consistía en la realización de pruebas empíricas, sólo se encargaban de probar que el producto no explotara y no tuviera errores garrafales. Esto no satisfacía la necesidad de obtener productos confiables, que se entregaran al cliente en el período de tiempo señalado, por lo que surge la necesidad de crear un grupo de calidad en la empresa. El trabajo desarrollado actualmente es a partir del uso de una metodología propia basada en RUP y la certificación de los productos que revisan bajo la norma de calidad ISO 9126.

Definen un plan de prueba beta que no es más que una aplicación “en vivo” del software en un entorno que no puede ser controlado por el desarrollador, pruebas al cliente y otras basadas fundamentalmente en el conocimiento y experiencia de los probadores (CITMATEL, 2007).

### ✓ **ParqueSoft**

**ParqueSoft** (Parque Tecnológico de Software), se inició en 1999 en Colombia, con la misión de servir como espacio de innovación y respaldo para emprendedores en el área de desarrollo de software. Define su trabajo de evaluación a productos multimedia en dos líneas fundamentales (Arena, 2006):

- **Programa Pruebas de Usabilidad Multimedia:** Desarrollado por un equipo multidisciplinario encargado de definir herramientas de trabajo, entre ellas se destaca: estándar de usabilidad por tipo de multimedia (educativas, informativas) que posteriormente servirá para hacer las pruebas manuales y semiautomáticas de las multimedia desarrolladas en ParqueSoft.

Este equipo está conformado por ingenieros de sistema, que trabajan concretamente en un generador estándar de framework para multimedia, diseñadores gráficos que son responsables del look and feel e ingenieros de pruebas que a partir del trabajo realizado en el laboratorio tienen la responsabilidad de abstraer una metodología para las pruebas de usabilidad.

- **Programa Pruebas de Usabilidad Multimedia para niños** :( Agosto/2005) Consiste en evaluar mediante la observación en uso, las multimedia desarrolladas por el cluster multimedia y probadas por el programa de pruebas funcionales. El laboratorio cuenta con un grupo de niños entre 6 y 14 años, en función de usuarios finales y que permite que el ingeniero de pruebas líder, evalúe un conjunto de criterios previamente definidos, logrando encontrar aspectos de mejoramiento en el software, asociados a la calidad en uso de los productos. Aún no tienen definida completamente la metodología utilizada pues muchas de las actividades, se derivan del conocimiento que genera el laboratorio mediante la práctica.

✓ “ **Imagenio: La nueva forma de ocio a la carta** “

Imagenio es un servicio multimedia destinado al entretenimiento del sector residencial. Utiliza la televisión como terminal multimedia para el acceso al producto.

Fue desarrollado bajo una metodología de calidad que propone como fases del ciclo de desarrollo del producto: viabilidad, definición de requisitos, análisis de sistema, diseño de arquitectura, diseño detallado y construcción, integración del sistema, **pruebas de sistema, pruebas de certificación** y soporte en campo (Linares, 2004).

Las pruebas a nivel de sistema se realizaron con el objetivo de verificar la estructura interna del programa antes de realizar su entrega al cliente, por un grupo de personas ajenas al de desarrollo, para evitar influencias de los desarrolladores. Se analizaron los requisitos de funcionalidad establecidos en las primeras fases de desarrollo. Los mismos fueron utilizados para el diseño de los casos de prueba que evaluarían dichas funcionalidades y estos se recopilaban en el documento Plan de pruebas de sistema.

Las pruebas fueron aplicadas en un entorno similar al de producción, con aproximaciones en las configuraciones de hardware o utilización de aplicaciones y herramientas para simular funcionalidades no disponibles en los sistemas de prueba.

Se realizaron las siguientes pruebas (Linares, 2004):

- Usabilidad a las interfaces de cliente, para medir que fueran amigables y fáciles de manejar tanto para el usuario como para los operadores del sistema.

- Rendimiento y prestaciones de los componentes hardware y software en condiciones límites y de sobrecarga, para comprobar la situación de carga que soporta la aplicación en un entorno de producción y detectar anomalías frente a la entrada masiva de peticiones.
- Instalación y desinstalación, donde se analizaran componentes software fácilmente instalables en un tiempo adecuado y la desinstalación de aquellos que no tienen un adecuado funcionamiento.
- Robustez de las aplicaciones gráficas con la inserción de valores correctos y valores sin sentido, ya sea que contengan diferentes tipos de caracteres u omisión de la información.
- Integración entre los elementos software y luego de regresión para ver si continúa existiendo un buen comportamiento de las diferentes aplicaciones una vez realizadas actualizaciones de software o incluidas nuevas funcionalidades.
- A la documentación, para verificar que contenga la información necesaria para el cliente u operador de manera clara y concisa. Anteriormente fue revisada en la fase de diseño detallado y construcción.

## 1.15 Conclusiones

Como principal conclusión está la estrecha dependencia que existe entre calidad de software y prueba a software, de ahí su importancia. La garantía de la calidad del software es determinante por la repercusión en la satisfacción del cliente y esta exige un producto libre de errores o al menos con un grado mínimo para su liberación.

Es vital que los desarrolladores hagan conciencia del significado de esta fase en el desarrollo del software y no lo continúen viendo como una actividad sencilla e intrascendente necesaria sólo al final de la elaboración del sistema.

## CAPÍTULO II: DESCRIPCIÓN DEL PROCESO DE PRUEBA

### 2.1 Introducción

Actualmente el aumento y la complejidad de los productos informáticos es considerable. Eso ha hecho que la ejecución de pruebas sea visto como una actividad inherente a todo proceso de desarrollo de software. El presente capítulo se compone por una breve descripción del producto a evaluar así como del proceso de pruebas aplicable a dicha multimedia. De este último se detalla el artefacto plan de pruebas con la especificación de recursos, alcance, responsables, estrategia, etc., a modo de maqueta de la actividad a realizar.

### 2.2 Descripción del producto

“Barrio Adentro al alcance de tus manos”, es un producto de tipo multimedia informativa, donde se utilizan varios medios en una presentación para, a través de sus actividades, mostrar contenidos que proporcionan información de un tema determinado. Fue desarrollada por estudiantes de la facultad 8 de la universidad de Ciencias Informáticas (UCI). Pertenece al programa “Atención integral de salud”, en el marco del convenio suscrito entre la República de Cuba y la República Bolivariana de Venezuela como parte del proyecto: “Desarrollo de contenidos en tecnologías de la información” (2005).

Es un producto sencillo, con una interfaz amena, que motiva al usuario a navegar. Proporciona información desde los entornos de las pantallas a través de imágenes, voces y textos referentes a la misión. Barrio Adentro es la prestación de servicios de Salud con calidad, dentro y fuera del Consultorio Popular, para y con la comunidad. Está vinculada al resto de los programas sociales que se brindan en el país actualmente.

En sus inicios, la misión fue concebida como una alternativa para satisfacer las necesidades de la población más humilde de Caracas. Más tarde, se convierte en un Plan Integral de Desarrollo Local que conjuga un conjunto de políticas sociales orientadas a mejorar la calidad de vida, promoviendo la organización y participación social de los sectores populares.

Tuvo sus orígenes en la actuación del personal médico cubano, durante la tragedia del Estado Vargas, en Diciembre de 1999. Motivados por tan grande y humana labor, la Alcaldía del municipio Libertador

suscribe un convenio con nuestro país para realizar una tarea similar en los barrio de Caracas, centro de las clases más desposeídas y necesitadas.

El objetivo de la misión es garantizar el acceso a los servicios de salud de la población excluida, que generaliza el número de habitantes de la región. Para ello trabajan en aras de implantar un modelo de gestión participativo que responda a las necesidades sociales de los grupos de población excluidos; además de fortalecer la red ambulatoria aumentando su capacidad mediante la implantación, consolidación y extensión de consultorios populares.

La misión está dirigida principalmente al 60 % de la población venezolana, compuesta por los sectores populares urbanos (gente de barrio), rural, pueblos fronterizos e indígenas. El sector más humilde y maltratado del país.

Este software informativo, está estructurado en dos módulos de contenido. En el primer módulo, representado por un consultorio popular, el usuario puede navegar y realizar un recorrido dentro de un consultorio a fin de conocer de manera general los servicios de asistencia médica integral que se ofrecen a la comunidad. En el segundo módulo, representado por la comunidad y guiado por una persona del comité de salud, puede navegar y conocer los trámites que debe realizar para solicitar la construcción de un consultorio en su comunidad y así mismo, conocerá donde están ubicados los consultorios, puntos de consulta y los otros servicios médicos que ofrece la Misión Barrio Adentro en todo el territorio nacional.

Mediante este producto los usuarios finales tendrán acceso a información necesaria para el desarrollo y éxito de dicha misión. El sector humilde, a quien va dirigida principalmente esta labor, desconoce los privilegios y opciones con que puede contar si accede a los servicios médicos cubanos, motivo suficiente para que se solicite la realización del producto multimedia “Barrio Adentro al alcance de tus manos” como un material informativo para el uso de toda la sociedad venezolana.

### **2.3 Proceso de pruebas**

Cuando se trazan un conjunto de actividades a realizar durante la evaluación de un software, se hace referencia al término proceso de pruebas. Dada la necesidad de obtener un producto libre de fallas, se recurrió a la elaboración y aplicación de un proceso de pruebas a la multimedia “Barrio Adentro al alcance de tus manos” que describiera y registrara el trabajo a desempeñar por el equipo de pruebas.



El mismo se conformó teniendo en cuenta una serie de pautas, que agilizando el trabajo, sin restarle importancia, garantizaran el éxito del mismo. El objetivo, es llevar un adecuado control de los elementos a probar, así como de los probados y resultados obtenidos.

### ***2.3.1 Descripción del proceso de prueba a aplicar***

El presente proceso describe los tipos de pruebas a ejecutar, a partir de la secuencia de acciones a realizar. Está basado en una propuesta de niveles de pruebas de donde se seleccionaron, dada las características del producto multimedia “Barrio Adentro al alcance de tus manos”, las pruebas que evaluarán la mayor cantidad de elementos incidentes en su calidad. Teniendo en cuenta que la multimedia se encuentra terminada y las particularidades de este tipo de software, que esencialmente se resumen en los requerimientos de “permitir y mostrar”, el peso de las pruebas recae básicamente sobre la funcionalidad de la multimedia.

Dicho proceso comienza con la generación de un plan de pruebas en base a la documentación sobre el software a probar. A partir de dicho plan, se entra en detalle diseñando pruebas específicas basándose en la documentación entregada. Una vez detalladas las pruebas (especificaciones de casos y de procedimientos), se toma la configuración del producto (revisada, para confirmar que se trata de la versión apropiada del programa) que se va a probar para ejecutar sobre ella los casos. Esto exige por parte de los desarrolladores de la entrega en tiempo, de toda la documentación existente: manual de usuario, especificación de casos de uso, en caso de no estar incluidos en la especificación de casos de uso, un documento que contenga los requerimientos y a que caso de uso corresponde cada uno de ellos, guión de contenido, guión final y la última versión de la multimedia.

El proceso integra un conjunto de actividades y entregables definidos para cada una de las etapas o fases propuestas. Comienza con la planificación de las pruebas, donde se define la estrategia, metas, objetivos, pruebas a aplicar y se selecciona la técnica de prueba a utilizar. Le sigue el diseño de las mismas, abarcando todo lo relacionado con los procedimientos de prueba, elaboración de los casos de prueba y ambiente de ejecución. Como fases finales del proceso está, probar y evaluar, para un posterior análisis de calidad de las mismas desde la efectividad obtenida (Ver fig. 3).

Teniendo en cuenta el análisis de los estados finales funcionales, examinados en la documentación, se trazarán las pautas a evaluar. Estas constituyen la línea de trabajo a la hora de generar los casos de prueba. Luego de aplicadas las pruebas se genera el documento de defectos y no conformidades, como documento de los resultados obtenidos. Contempla una descripción detallada de todos los defectos y errores detectados, así como, sugerencias realizadas al equipo de desarrollo. Los resultados reales (satisfactorios e insatisfactorios) quedarán archivados en el diseño de los casos de prueba. La intención de todo esto es, su posterior utilización en la depuración del producto y/o futuras iteraciones de prueba que se deseen realizar.

La estrategia seguida concibe probar el producto a nivel modular primeramente, para seguir el diseño y comportamiento de este como un componente que puede ejecutarse independientemente. En un segundo momento se comprobará su funcionalidad a nivel de sistema, en un entorno que simule lo más perfecto posible el entorno final. No se tienen en cuenta las pruebas a nivel de integración pues al producto estar finalizado se considera que una vez probado el sistema se da cumplimiento a los objetivos particulares de esta prueba.

La utilización de listas de chequeo que contienen aspectos de la interfaz gráfica, es una técnica muy específica que se aplica a este tipo de software, permitiendo la evaluación de los atributos de las interfaces y de calidad con los que debe contar la multimedia. Dado el alcance de la presentación de los contenidos en cada escenario, es necesario obtener un resultado cuantitativo de estos atributos y características, contenido que puede variar según las particularidades de la interfaz del producto que se esté probando.

Para llevar adelante este proceso se cuenta con un equipo de pruebas conformado por un miembro, desempeñando los diferentes roles definidos: administrador de pruebas, diseñador de pruebas y probador. En conjunto con miembros del equipo de desarrollo hará la formulación del plan de prueba, el programa y la estrategia con base en el alcance de la multimedia y los posibles escenarios identificados. El plan recoge todo lo que se describe en el proceso de pruebas para guiar el desarrollo de las mismas, por lo que puede estar sujeto a cambios.

### *2.3.1.1 Descripción de las etapas a desarrollar.*

A partir de la propuesta realizada sobre las pruebas a aplicar al producto señalado, se describen brevemente las etapas que conforman el proceso:

### ✓ **Planificar las pruebas**

Consiste en realizar una planificación detallada de las pruebas a aplicar. Como actividades a desarrollar comprende la definición de la estrategia de pruebas, así como la identificación de factores vinculados a las pruebas, dígame: meta, objetivos, alcance. La estrategia contempla la selección de pruebas por niveles, técnica de diseño a aplicar y requerimientos a probar. Durante esta etapa se genera el artefacto plan de pruebas, en una primera versión no finalizada.

### ✓ **Diseñar las pruebas**

El propósito general de esta actividad es definir y diseñar los casos de prueba que se ejecutarán durante el proceso de pruebas planificado, documentando la secuencia de pasos para ejecutar cada caso de prueba, los requisitos de datos para realizar las pruebas, los resultados esperados y los métodos para registrar los resultados.

Para el diseño de los casos de prueba se analizan los casos de uso, los requerimientos a probar y las técnicas de pruebas seleccionadas, para obtener pruebas que tengan la mayor probabilidad de descubrir los defectos existentes. Se especifican elementos de software y hardware a tener en cuenta. En esta fase se realiza el diseño de los casos de prueba, dando por culminado el artefacto plan de prueba.

### ✓ **Ejecutar las pruebas**

Abarca la implementación y ejecución de las pruebas anteriormente diseñadas, con el fin de evaluar las características y funcionalidades definidas a través de los requerimientos y determinar en que medida estos se han cumplido o no. Comprende además la documentación (generación de lista de defectos y no conformidades) de los errores que se van detectando para darle seguimiento, hasta garantizar la total corrección de los mismos. Se obtiene de modo general un resumen de los resultados de las pruebas.

### ✓ **Evaluar resultados de las pruebas**

En esta etapa se aplicará el criterio de terminación del proceso de pruebas para saber si se puede dar por concluido exitosamente o no. Tiene el propósito de mostrar a los involucrados en el proceso de pruebas si se dio cumplimiento a los objetivos y metas trazadas en el Plan de Pruebas, y decidir con esto si finaliza el ciclo de prueba o hay que recurrir a nuevas iteraciones o cambios. Se analizan los resultados obtenidos, a

partir de los diferentes aspectos que puedan haberlos ocasionados, así como la repercusión que tienen para la certificación de calidad del producto.

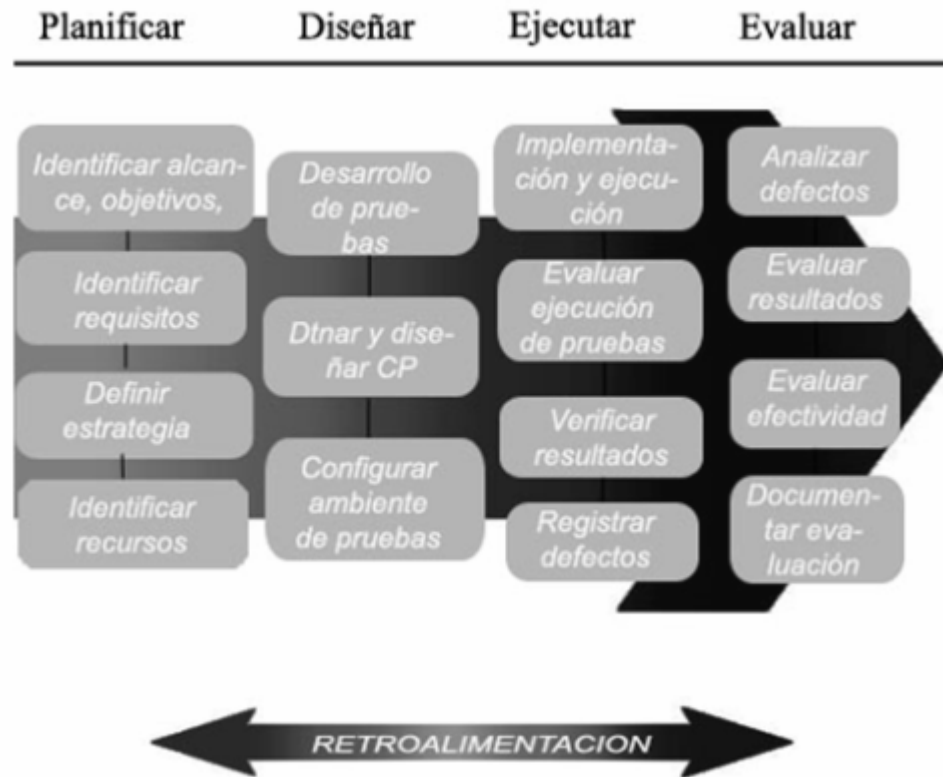


Fig. 3 Descripción de las actividades del proceso de pruebas.

## 2.4 Plan de prueba general

El plan de prueba es un documento generado durante el proceso de prueba. El presente, proporciona el marco dentro del cual el equipo de prueba desarrolla las pruebas al producto multimedia “Barrio Adentro al alcance de tus manos”, trabajando con los recursos y la planificación dada. Está sujeto a cambios durante la aplicación del proceso de pruebas, por lo que se considera meramente una guía en la comprensión del mismo.

### **2.4.1 Introducción**

La realización del siguiente plan de pruebas está enmarcado en la necesidad de certificar un producto con calidad, óptimo para entregar al cliente. Esto conlleva a establecer una estrategia sistemática que evite desaprovechar el tiempo, esfuerzo consumido innecesariamente y errores sin detectar.

Las pruebas tendrán dos etapas para su desarrollo. En ambas se utilizará el método de pruebas de caja negra, centrado en los requisitos fundamentales del software. Los datos de prueba se escogerán atendiendo a las especificaciones del problema sin importar los detalles internos del programa, a fin de verificar que el producto corra bien.

Se pretende evaluar el funcionamiento de la multimedia a partir de su ejecución bajo ciertas condiciones predefinidas, recogidas en los casos de prueba. Este equipo de pruebas intenta encontrar funciones incorrectas o ausentes, errores de interfaz, de rendimiento, ambigüedades e incoherencias en la documentación; de manera general, aspectos que atenten contra la idoneidad del producto.

#### **✓ Propósito**

Se busca encontrar el mayor número de defectos al producto, cometidos durante su desarrollo, lo cual conllevaría a un proceso de depuración, o sea, la corrección de los defectos en caso de encontrarlos. Esta última tarea no será desarrollada por no ser objetivo del presente trabajo pero si queda recomendada para posteriores procesos de prueba a productos multimedia y software de manera general.

Contiene la información necesaria para describir el plan a la hora de realizar las pruebas. Se especifican aspectos como: diseño, validación y generación de pruebas en una primera etapa, así como su ejecución, evaluación y reporte de los resultados obtenidos posteriormente.

#### **✓ Alcance**

Es necesario realizar una evaluación de la multimedia “Barrio Adentro al alcance de tus manos” que permita comprobar en que estado se encuentra para su liberación. Por tanto, mientras más temprano sea

la detección de defectos y su pronta solución, menor será el impacto en el tiempo y recursos necesarios. Por ello, el alcance está en determinar correctamente los aspectos relevantes que necesitan ser probados durante la etapa de pruebas, a partir de los requisitos definidos en la especificación de requisitos.

### **2.4.2 Organización del equipo de pruebas**

El producto a probar tiene la característica particular de ser muy sencillo y de fácil manejo. Por ello se decidió asignar un solo miembro al equipo de pruebas que desarrollara el proceso de manera íntegra. El mismo, asumió los diferentes roles que se consideraron necesarios para el éxito del mismo:

- ✓ **Administrador o jefe de pruebas:** Constituye la cabeza de la prueba, como responsable de las actividades fundamentales a realizar. Dirige el comportamiento de estas, definiendo los tipos de pruebas necesarias y manejando los resultados.
- ✓ **Diseñador de pruebas:** Encargado de definir el alcance de la prueba y asegurar su correcta implementación. Identifica las técnicas, herramientas y guías de trabajo apropiadas para la implementación de la prueba. Brinda la información de los recursos necesarios para ejecutar la prueba. Es conocido también como arquitecto de prueba, arquitecto de automatización de prueba o especialista automático de prueba.
- ✓ **Probador:** Responsable de las principales actividades y el mayor esfuerzo de las pruebas. Produce las principales pruebas a realizar y obtiene directamente los resultados.

### **2.4.3 Especificación del software y hardware necesarios**

Para la realización de las pruebas se necesita un ambiente bajo las siguientes características:

Una PC con:

- ✓ Requerimiento de software:
  - PC con sistema operativo Windows.
  - Macromedia Flash MX 2004
- ✓ Requerimiento de hardware:
  - Microprocesador: 486 a 66Mhz 42.

- Memoria RAM: 32Mb
- Disco Duro: 2Gb
- Lector de CD con tasa de transferencia de 24X (3.600KB/s) o superior.
- Tarjeta de Sonido de 16 bits.
- Periféricos: Teclado, ratón, audífonos o bocinas.

#### **2.4.4 Descripción de Requerimientos a probar**

Se listan en esta sección los elementos (requisitos funcionales y no funcionales) que constituyen objetivos de las pruebas, es decir, las características y funcionalidades que serán probadas.

##### ➤ **Requisitos Funcionales**

###### *Generales:*

- ✓ RF1\_ Mostrar presentación general del producto.
- ✓ RF2\_ Controlar audio de fondo.
- ✓ RF3\_ Salir de la aplicación.
- ✓ RF4\_ Acceder a pantalla de inicio.
- ✓ RF5\_ Acceder a módulos.
- ✓ RF6\_ Mostrar contenidos.

###### *Requisitos Módulo 1*

- ✓ RF6.1\_ Mostrar ¿Qué es Barrio Adentro?
- ✓ RF6.2\_ Mostrar los Antecedentes de Barrio Adentro.
- ✓ RF6.3\_ Mostrar Servicios que ofrece Barrio Adentro.
  - Mostrar los Servicios que ofrece Barrio Adentro en Nivel 1
  - Mostrar los Servicios que ofrece Barrio Adentro en Nivel 2.
  - Mostrar los Servicios que ofrece Barrio Adentro en Nivel 3.
- ✓ RF7\_ Jugar juego Sopa de Letras.

###### *Requisitos Módulo 2*

- ✓ RF6.4\_ Mostrar los Requisitos mínimos para la certificación de una edificación.

- ✓ RF6.5\_ Mostrar los Requisitos mínimos para la certificación de un terreno
- ✓ RF8\_ Jugar juego Rompecabezas.

### ➤ **Requisitos no Funcionales**

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

- ✓ RNF1\_ El diseño de la interfaz debe reconocer eventos de mouse y/o teclado para la navegación del sistema.
- ✓ RNF2\_ Mostrar interfaz lo más clara y atractiva posible.
- ✓ RNF3\_ El sistema debe correr sobre los sistemas operativos Windows 98, Windows 2000, Windows XP.
- ✓ RNF4\_ Permitir navegabilidad desde cualquier escenario.
- ✓ RNF5\_ Mostrar producto a pantalla completa.
- ✓ RNF6\_ Los elementos relacionados con la estética visual de la interfaz deben tener relación, correspondencia y calidad.

Se probarán elementos como:

- Correspondencia entre tema tratado, texto y demás medios que aparecen en cada pantalla.
- Relación entre la interfaz gráfica de la pantalla contra el guión de contenidos o medias.
- Revisión ortográfica de los textos de la multimedia.
- Calidad de los medios (sonido, imágenes, resolución de pantalla) que se muestran.
- ✓ RNF7\_ La documentación del producto multimedia, debe presentar claridad editorial, ortografía, coherencia de ideas.

Dígase elemento de documentación a:

- Guión técnico.
- Guión de contenido.
- Requerimientos.
- Manual de usuario.



### 2.4.5 Estrategia de prueba

La línea de trabajo a la hora de evaluar dinámicamente un software simula una trayectoria en espiral (Pressman, 2000). Debe comenzar probando el componente más simple y pequeño e ir avanzando progresivamente hasta probar todo el software en su conjunto. El enfoque a utilizar para probar la multimedia y la definición de la realización de las pruebas fue elaborada teniendo en cuenta esa premisa, Ver fig.4. Se hizo aplicable al agrupar las funcionalidades a evaluar por módulos y posteriormente al sistema, teniendo en cuenta siempre las zonas más vulnerables y propensas a generar errores. Se definieron las siguientes pruebas por niveles:

- ✓ **Nivel de Unidad:** Estas comienzan con la prueba de cada módulo.
  - Pruebas de interfaz de usuario.
- ✓ **Nivel de Sistema:** El software ensamblado totalmente con cualquier componente hardware que requiere se prueba para comprobar que se cumplen los requisitos funcionales.
  - Funcionales
  - Portabilidad
  - Compatibilidad
- ✓ **Otras:**
  - Solidez.
  - Interfaz gráfica de usuario (IGU).
  - Documentación.

La especificación de cada prueba se estipulará en un documento de diseño generado para cada una de ellas, el cual contendrá los aspectos específicos para su aplicación.

### 2.4.6 Diseño de las pruebas

El producto puede ser probado según las técnicas de pruebas existentes. Distintas técnicas de prueba ejercitan diferentes métodos como guía para realizar las pruebas.

- ✓ Técnicas de diseño de casos de prueba
  - **Prueba de la caja negra:** Consiste en estudiar la especificación de las funciones, la entrada y la salida para derivar los casos. La prueba consiste en probar todas las posibles entradas y salidas del programa sin considerar el código en lo absoluto.

✓ Método a utilizar:

- **Particiones o clases de equivalencia:** La partición de equivalencia divide el campo de entrada de un programa en *clases de datos* de los que se pueden derivar casos de prueba, en que en la especificación o condición del caso de uso indica: rango de valores, número de valores, tipo de dato. Divide el dominio de entrada de un programa en un número finito de *clases de equivalencia*.



Fig. 4 Estrategia de prueba.

### 2.4.7 Procedimiento de pruebas

El presente plan contiene una descripción detallada de los procedimientos de prueba correspondiente a los distintos casos de prueba. Las pruebas fueron agrupadas por niveles o grupos de prueba y estos a su vez contienen los respectivos casos de prueba.

Cada grupo de prueba contiene la siguiente descripción: (*propósito* del grupo de pruebas, los distintos *conjuntos de pruebas* que lo conforman, con su respectivo *propósito*, *alcance* y *criterio de aceptación*).

#### 2.4.7.1 Pruebas de unidad

Son las encargadas de comprobar la ejecución de cada módulo como unidad mínima funcional. Para ello los casos de prueba deben tener una alta probabilidad de encontrar errores transparentes hasta el momento. Finalmente los resultados dirán si es o no correcto el trabajo evaluado.

✓ **Interfaz de usuario**

Verificar la interacción del usuario con el producto multimedia, para comprobar si la interfaz de usuario permite al usuario acceder y navegar a través de toda la funcionalidad de la aplicación. Probar todas las posibles entradas y salidas del programa, a nivel de unidad en la interfaz de usuario.

<b>Tipo de prueba</b>	Interfaz de usuario
<b>Propósito</b>	Comprobar si la navegación por la aplicación responde a los requisitos pautados, a nivel de pantalla a pantalla a través de eventos del mouse y/o teclado.
<b>Técnica</b>	<ul style="list-style-type: none"> <li>- A partir de la selección de particiones equivalentes, comprobar en la interfaz de cada pantalla eventos como: onclip, mouse over en botones y vínculos (abrir, cerrar, atrás, adelante) para la navegabilidad e interacción con servicios del escenario.</li> <li>- Crear o modificar pruebas para cada ventana con el objetivo de verificar la correcta navegación y su estado.</li> <li>- Comparar el resultado esperado con el resultado obtenido.</li> </ul>
<b>Criterio de aceptación</b>	Cada pantalla se verifica con éxito y es consistente. La interfaz externa resulta atractiva y clara al usuario durante el acceso a los servicios solicitados de manera íntegra y siempre que lo desee.

Tabla 1 Procedimiento de prueba de interfaz de usuario

2.4.7.2 *Pruebas de sistema*

Buscan ejercitar profundamente el sistema comprobando su integración globalmente, verificando así el funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen, dando las entradas y obteniendo salidas del programa.

✓ **Pruebas Funcionales**

Fueron aplicadas a requisitos funcionales y no funcionales. Están enfocadas a asegurar que el software realiza correctamente todas las funciones que se han detallado en los requerimientos tanto funcionales como no funcionales dados por el usuario. Se pueden realizar pruebas para cada caso de uso.

<b>Tipo de prueba</b>	Funcional
<b>Propósito</b>	Comprobar concordancia entre requisitos dados por el usuario y funcionamiento del software. Verificar además requisitos de navegabilidad, rendimiento, configuración, etc. a través de pruebas minuciosas a todas las funciones de la multimedia.
<b>Técnica</b>	<ul style="list-style-type: none"> <li>- Probar minuciosamente la funcionalidad de la multimedia acorde al diseño en un entorno que simule de la mejor manera el real a partir de la técnica de partición equivalente.</li> <li>- Realizar iteraciones con el sistema, a través de los análisis de posibles entradas y salidas al mismo. En este caso clases válidas e inválidas, como términos para acciones correctas e incorrectas del usuario con la multimedia.</li> </ul>
<b>Criterio de aceptación</b>	Verificar correspondencia entre requerimientos y funcionalidad del producto.

Tabla 2 Procedimiento de prueba funcional.

✓ **Pruebas de instalación**

Verifica que la multimedia desde cualquier sistema operativo de la familia Windows permita al usuario acceder a los diferentes servicios que ella ofrece.

<b>Tipo de prueba</b>	Instalación.
<b>Propósito</b>	Probar el funcionamiento de la multimedia bajo las diferentes de plataformas de software preestablecidas para su desarrollo.
<b>Técnica</b>	<ul style="list-style-type: none"> <li>- Instalación de la multimedia sobre diferentes sistemas operativos y versiones de la herramienta de elaboración.</li> <li>- Comprobar que sea boteable y su posterior ejecución a partir de la interacción del usuario.</li> </ul>
<b>Criterio de aceptación</b>	Verificar que la multimedia se puede ejecutar desde el sistema operativo señalado.

Tabla 3 Procedimiento de prueba de Instalación.

✓ **Pruebas de portabilidad**

La multimedia debe permitir a través de los diferentes eventos de mouse y/o teclado interactuar con ella cabalmente.

<b>Tipo de prueba</b>	Portabilidad
<b>Propósito</b>	Probar el funcionamiento de la multimedia bajo las diferentes configuraciones de hardware.
<b>Técnica</b>	- Ejecución de la multimedia a través de los diferentes elementos de hardware (periféricos) requeridos para interactuar con los servicios que presta.
<b>Criterio de aceptación</b>	Permitir interactuar con el producto mediante el uso de los periféricos requeridos.

Tabla 4 Procedimiento de prueba de portabilidad.

2.4.7.3 Otras

✓ **Prueba de Solidez**

Se aplicó a la documentación del producto para comprobar solidez de la información y el funcionamiento adecuado en relación a lo preestablecido. La solidez comprende desde ortografía de los documentos hasta estética visual de la interfaz.

<b>Tipo de prueba</b>	Solidez.
<b>Propósito</b>	Comprobar la ejecución de lo pautado en el diseño, teniendo en cuenta la calidad que exigen los medios expuestos.
<b>Técnica</b>	Comprobar en los escenarios que componen el sistema elementos como: <ul style="list-style-type: none"> <li>▪ Correspondencia entre tema, texto, imágenes y demás medios de un escenario o conjunto de escenarios de un mismo módulo.</li> <li>▪ Ortografía y coherencia en los textos de los mensajes.</li> <li>▪ Apariencia en cuanto a calidad y formato estándar de las pantallas.</li> <li>▪ Correspondencia entre guión final e interfaz gráfica.</li> </ul>

<b>Criterio de aceptación</b>	Existe relación entre lo preestablecido y lo realizado. El producto tiene un nivel de solidez requerido para su uso.
-------------------------------	--

Tabla 5 Procedimiento de prueba de solidez.

✓ **Prueba a la documentación**

Las pruebas a la documentación aseguran su optimización para el uso de la misma dada su incidencia en todo el desarrollo del producto y por último en la comprensión de este por parte del usuario (Manual de usuario)

<b>Tipo de prueba</b>	Prueba a la documentación.
<b>Propósito</b>	Comprobar que la documentación necesaria para realizar el proceso de pruebas tiene el formato adecuado, claridad editorial y coherencia necesaria para su uso.
<b>Técnica</b>	Revisión de la documentación (guión técnico y final, requerimientos) en cuanto a: nivel de actualización y disponibilidad, transparencia y coherencia en las ideas, especificación y realización de lo que se quiere.
<b>Criterio de aceptación</b>	La documentación está acorde a lo que se necesita para hacer uso de la misma.

Tabla 6 Procedimiento de prueba a la documentación.

✓ **Pruebas de interfaz gráfica de usuario (IGU)**

Las pruebas de IGU se centran en la verificación del cumplimiento de estándares en las interfaces de usuario. Garantizan la evaluación de un conjunto de elementos gráficos como: ventanas, menús, botones, etc. que permiten la interacción entre el usuario y la aplicación informática (interfaz gráfica).

<b>Tipo de prueba</b>	Interfaz gráfica de usuario (IGU)
<b>Propósito</b>	Comprobar calidad, uniformidad y nivel de comunicación en el diseño de los elementos por pantallas para garantizar fluidez durante la interacción del usuario con el producto.
<b>Técnica</b>	Revisión de estándares en elementos gráficos que garanticen un entorno acorde al tema abordado y usuarios finales, a través del diseño de casos de prueba que toquen a fondo las particularidades en cada una de las pantallas de la multimedia.
<b>Criterio de aceptación</b>	Se verificó el cumplimiento de estándares en los elementos gráficos utilizados en las interfaces del producto (pantallas).

Tabla 7 Procedimiento de prueba de interfaz gráfica de usuario (IGU).

### 2.4.8 Casos de Prueba

Los casos de prueba (CP) fueron definidos con el fin de validar y verificar características específicas del producto que inciden en la calidad del mismo, dígase: funcionalidad, en cuanto a exactitud y seguridad; confiabilidad para mantener la integridad y recuperarse de cambios y anomalías; usabilidad; eficiencia en tiempo de respuesta y uso de recursos, solidez, etc. Se componen por: datos informativos generales(Caso de uso y requisito) y los específicos al caso de prueba (descripción de la funcionalidad, flujo central, prerrequisito, poscondiciones, observaciones y las iteraciones). Las iteraciones de entrada al caso de prueba agrupan las acciones a realizar según clase válida e inválida. Hay presencia de una clase válida cuando la condición probada es verdadera o se muestra como se refleja en el resultado esperado e inválida si lo probado no es correcto y el sistema responde ante la acción ejecutada erróneamente .

➤ *Casos de prueba de interfaz: Interactuar escenario de presentación Módulo2.*

Los casos de prueba agrupan el conjunto de pruebas realizadas sobre el escenario de presentación del módulo 2: ¿Cómo construir un Consultorio Popular? para comprobar su funcionalidad durante la interacción con el usuario.

**CP: Interactuar escenario de presentación Módulo2.**

<b>Datos iniciales</b>	
<b>Caso de Uso:</b> Acceder a módulos.	<b>Requisito:</b> RF5
<b>Caso de Prueba</b>	
Interactuar escenario de presentación Módulo 2.	<b>Código del CP:</b> CPRF_01
<b>Descripción de funcionalidad</b>	Permite comprobar la ejecución correcta del sistema frente a la interacción del usuario para acceder a escenarios del módulo 2.
<b>Prerrequisitos</b>	Acceder al módulo 2 ¿Cómo construir un Consultorio Popular? en el escenario de Inicio.

<b>Flujo Central</b>	<p>El caso de prueba se inicia cuando el usuario accede al módulo 2. El sistema muestra:</p> <ul style="list-style-type: none"> <li>▪ Imágenes representativas de viviendas y un hombre, a la izquierda y derecha del escenario respectivamente. Al fondo, imagen de una comunidad.</li> <li>▪ Nubes en movimiento hacia la izquierda.</li> <li>▪ Tool tips de color Rojo al lado de cada imagen representativa y en la barra inferior que aparecen según se cliquee sobre ellos.</li> <li>▪ Barra de opciones básicas (Audio, Inicio y Salir).</li> <li>▪ Barra inferior con las opciones de ocultarla y acceder a los módulos 1 y 2.</li> </ul>		
<b>Iteraciones de entrada al Caso de Prueba</b>			
<b>Clases Válidas</b>	<b>Clases Inválidas</b>	<b>Resultado</b>	
		<b>Esperado</b>	<b>Real</b>
<p>1. Hacer mouse over sobre imagen representativa del escenario (viviendas y/o hombre).</p> <p>2. Hacer clic sobre imagen de viviendas.</p> <p>3. Hacer clic sobre imagen de hombre.</p>		<p>1.1 La imagen cambia de color según se acerca y retira el mouse.</p> <p>1.2 Tool tips visible en color Rojo con texto referente a la imagen señalada.</p> <p>1.3 Se escucha un efecto de sonido.</p> <p>2.1 El sistema muestra pantalla referente al servicio solicitado.</p> <p>2.2 Se da inicio al Caso de Prueba: <i>Interactuar escenario Requisitos mínimos para la certificación de una edificación.</i></p> <p>3.1 El sistema muestra pantalla referente al servicio solicitado.</p> <p>3.2 Se da inicio al Caso de Prueba <i>Requisitos mínimos para la certificación de un terreno.</i></p>	<p>S</p> <p>S</p> <p>S</p> <p>S</p> <p>S</p> <p>S</p> <p>–</p>



**Poscondiciones:** El sistema muestra pantalla referente al servicio solicitado.

**Observaciones:** Tooltip referente al módulo 2 permanecerá visible, sin función de vínculo alguno.

Tabla 8 CP: Interactuar escenario de presentación Módulo2.

➤ *Casos de prueba de interfaz: Mostrar presentación del producto.*

Los casos de prueba agrupan el conjunto de pruebas realizadas para comprobar que el sistema muestre correctamente la presentación general del producto.

**CP: Mostrar presentación del producto.**

<b>Datos iniciales</b>	
<b>Caso de Uso:</b> Mostrar presentación.	<b>Requisito:</b> RF1
<b>Caso de Prueba</b>	
Mostrar presentación del producto.	<b>Código del CP:</b> CPRF_012
<b>Descripción de funcionalidad</b>	Permite comprobar que el producto muestre correctamente la presentación.
<b>Prerrequisitos</b>	Ejecutar la multimedia.
<b>Flujo Central</b>	<p>El caso de prueba tiene inicio con la ejecución de la aplicación por parte del usuario dando lugar a la presentación general del producto.</p> <ul style="list-style-type: none"> <li>▪ El sistema muestra una secuencia de escenarios con los organismos e instituciones vinculados a la elaboración del producto e imagen de un consultorio en unas manos con el título del producto, respectivamente.</li> <li>▪ Tema musical de fondo desde que se carga el producto, canción típica del país.</li> <li>▪ Al concluir la presentación se muestra la pantalla inicial del producto.</li> <li>▪ El usuario puede acceder directamente al escenario de inicio presionando la tecla Esc.</li> </ul>
<b>Iteraciones de entrada al Caso de Prueba</b>	
<b>Clases</b>	<b>Clases</b>
	<b>Resultado</b>

<i>Válidas</i>	<i>Inválidas</i>	<i>Esperado</i>	<i>Real</i>
1. Permitir cargar automáticamente la presentación del producto.		1.1 El sistema muestra la presentación general del producto (imágenes, tema musical).	S
		1.2 El cursor permanece invisible.	S
		1.3 Terminada la presentación, el sistema muestra la pantalla de inicio. Se da inicio al Caso de Prueba: <i>Interactuar escenario de inicio</i> .	S
2. Presionar tecla <i>Esc</i> , para anular la presentación.		2.1 El sistema aborta la presentación.	S
		2.2 Se muestra la pantalla de inicio. Se da inicio al Caso de Prueba: <i>Interactuar escenario de inicio</i> .	S
<b>Poscondiciones:</b> El sistema muestra la pantalla de inicio, primer escenario desde el cual el usuario puede acceder a los módulos de contenido.			
<b>Observaciones:</b>			

Tabla 9 CP: Mostrar presentación del producto.

➤ *Casos de prueba de interfaz: Interactuar escenario de presentación.*

Los casos de prueba agrupan el conjunto de pruebas realizadas para comprobar la correcta ejecución del sistema frente a las interacciones del usuario en el escenario de presentación de Barrio Adentro, con el fin de acceder a los módulos de contenido.

✓ *Elementos a evaluar:*

- Acceder al módulo 1 ¿Qué es Barrio Adentro?
- Acceder al módulo 2 ¿Cómo construir un Consultorio Popular?

**CP: Acceder a módulos 1 o 2.**

<b>Datos iniciales</b>			
<b>Caso de Uso:</b> Acceder a módulos.		<b>Requisito:</b> RF4	
<b>Caso de Prueba</b>			
Acceder al módulo 1: ¿Qué es Barrio Adentro?		<b>Código del CP:</b> CPRF_013	
<b>Descripción de funcionalidad</b>	Permitir al usuario acceder correctamente a Módulo 1: ¿Qué es Barrio Adentro? seleccionado por él.		
<b>Prerrequisitos:</b>	Seleccionar módulo al que desea acceder.		
<b>Flujo Central:</b>	<p>El caso de prueba se inicia una vez terminada la presentación de la multimedia. El sistema muestra:</p> <ul style="list-style-type: none"> <li>▪ Comunidad donde se encuentra el consultorio de Barrio Adentro.</li> <li>▪ Grupo de tres personas del Comité de Salud.</li> <li>▪ Doctor ubicado próximo a la puerta del consultorio Barrio Adentro.</li> <li>▪ Imágenes (consultorio y grupo del Comité de Salud) que cambian de coloración y muestran un tool tips, según se mueva el Mouse sobre ellas.</li> <li>▪ Locución desde el inicio de la pantalla referente a la presentación del producto.</li> <li>▪ Textos (nombre de los módulos) en extremos inferiores con acceso a los módulos.</li> <li>▪ Scroll formado por la interfaz del escenario y los textos inferiores según se mueva el Mouse por el escenario.</li> </ul> <p>El usuario accede al módulo que desea a través de la selección del objeto o del texto en la parte inferior de la pantalla referente a cada módulo.</p>		
<b>Iteraciones de entrada al Caso de Prueba</b>			
<b>Clase válida</b>	<b>Clase inválida</b>	<b>Resultado</b>	
		<b>Esperado</b>	<b>Real</b>

<p>1. Mover Mouse sobre el escenario Inicio que constituye un scroll.</p>		<p>1.1 El sistema desplaza el escenario en sentido contrario al movimiento del Mouse.</p>	<p>S</p>
<p>2. Mover Mouse sobre imagen (consultorio o grupo de 3 personas).</p>		<p>2.1 La imagen hace una animación (Cambia de coloración y muestra un tool tips).</p>	<p>S</p>
<p>3. Hacer clic sobre vínculos de acceso al módulo 1 (Imagen de consultorio, texto inferior izquierdo).</p>		<p>3.1 Se accede al módulo 1 ¿Qué es Barrio Adentro?</p>	<p>S</p>
<p>4. Hacer clic sobre vínculos de acceso al módulo 2 (Imagen de 3 personas del Comité de Salud, texto inferior izquierdo).</p>		<p>4.1 Se accede al módulo 2 ¿Cómo construir un Consultorio Popular?</p>	<p>S</p>
<p><b>Poscondiciones:</b> El sistema muestra la pantalla inicial ¿Qué es Barrio Adentro? correspondiente al módulo 1.</p>			
<p><b>Observaciones:</b></p>			

Tabla 10 CP: Acceder a módulos 1 o 2.

➤ *Casos de prueba funcionales: Interactuar escenarios mostrar contenidos.*

Los casos de prueba agrupan el conjunto de pruebas realizadas para comprobar la funcionalidad: mostrar contenidos, correspondiente a las pantallas del módulo 1: ¿Qué es barrio Adentro?

✓ *Escenarios a probar:*

- Interactuar con escenario ¿Qué es barrio Adentro?

- Interactuar con escenario Antecedentes (Ver anexo 1).
- Interactuar con escenario Servicios que ofrece.
- Interactuar con escenario Juegos.

**CP: Interactuar escenario ¿Qué es Barrio Adentro?**

<b>Datos iniciales</b>			
<b>Caso de Uso:</b> Mostrar contenidos ¿Qué es Barrio Adentro?		<b>Requisito:</b> R6.1	
<b>Caso de Prueba</b>			
Interactuar con escenario ¿Qué es Barrio Adentro?		<b>Código del CP:</b> CPRF_02	
<b>Descripción de funcionalidad</b>	Permitir interactuar correctamente con el escenario ¿Qué es Barrio Adentro? para mostrar la información referente a dicha pantalla.		
<b>Prerrequisitos</b>	Acceder al módulo 1 ¿Qué es Barrio Adentro?		
<b>Flujo Central</b>	<p>El caso de prueba se inicia cuando el usuario accede al módulo 1. El sistema muestra en el escenario ¿Qué es Barrio Adentro?:</p> <ul style="list-style-type: none"> <li>▪ Mural del consultorio titulado” Mural de Consultorio Popular” que incluye textos e imágenes animadas.</li> <li>▪ A la derecha, texto en forma de pergamino referente a la pantalla y a la izquierda imagen de un consultorio popular.</li> <li>▪ Locución íntegra del texto expuesto.</li> <li>▪ Barra de opciones básicas (Audio, Inicio y Salir).</li> <li>▪ Menú principal con acceso a los escenarios del módulo 1.</li> <li>▪ Barra inferior con las opciones de ocultarla y acceder a los módulos, 1(deshabilitado por encontrarse en él) y 2.</li> </ul> <p>El usuario interactúa con las opciones al mover el mouse sobre ellas y hacer clic.</p>		
<b>Iteraciones de entrada al Caso de Prueba</b>			
<b>Clases Válidas</b>	<b>Clases Inválidas</b>	<b>Resultado</b>	
		<b>Esperado</b>	<b>Real</b>

1. Hacer clic sobre vínculo del menú principal ¿Qué es Barrio Adentro?		1.1 El sistema muestra texto e imagen.	S
		1.2 Se escucha locución.	S
2. Hacer mouse over sobre imagen del consultorio popular.		2.1 La imagen hace un movimiento hacia la izquierda y luego a la derecha con un efecto, según se posiciona y retira el mouse respectivamente.	S
<b>Poscondiciones:</b>			
<b>Observaciones:</b>			

Tabla 11 CP: Interactuar escenario ¿Qué es Barrio Adentro?

➤ *Casos de prueba funcionales: Interactuar escenario Servicios que ofrece.*

Este caso de prueba agrupa el conjunto de pruebas realizadas dentro del escenario “*Servicios que ofrece*”, a los diferentes niveles de atención de salud que conforman las actividades de la misión Barrio Adentro.

✓ *Escenarios a evaluar:*

- Servicios que ofrece en Nivel 1 (Ver anexo 2).
- Servicios que ofrece en Nivel 2.
- Servicios que ofrece en Nivel 3 (Ver anexo 3).

**CP: Interactuar escenario “Servicios que ofrece”, Nivel 2.**

<b>Datos iniciales</b>	
<b>Caso de Uso:</b> Mostrar los Servicios que ofrece Barrio Adentro en Nivel 2.	<b>Requisito:</b> R6.3
<b>Caso de Prueba</b>	
Interactuar escenario “Servicios que ofrece” Nivel 2	<b>Código del CP:</b> CPRF_05
<b>Descripción de funcionalidad</b>	Permitir interactuar correctamente con el escenario <i>Servicios que ofrece</i> para mostrar la información referente a la pantalla <i>nivel 2</i> .

<b>Prerrequisitos</b>	Hacer clic sobre el vínculo <i>Nivel 2</i> , del escenario <i>Servicios que ofrece</i> .		
<b>Flujo Central</b>	<p>El Caso de Prueba se inicia cuando el usuario accede a la opción (haciendo clic) <i>Nivel 2</i>. El sistema muestra a través de una imagen de un mural titulado: Mural del Consultorio popular:</p> <ul style="list-style-type: none"> <li>▪ Scroll en forma horizontal con íconos (imágenes) que representan cada servicio del Nivel2.</li> <li>▪ Submenú con acceso a los niveles 1,2 y 3.</li> <li>▪ A la izquierda texto informativo en forma de pergamino y a la derecha imagen, referentes al Nivel 2.</li> <li>▪ Locución sobre texto expuesto que comienza desde el inicio de la pantalla.</li> </ul> <p>El usuario interactúa con las diferentes opciones del escenario a través de un clic.</p>		
<b>Iteraciones de entrada al Caso de Prueba</b>			
<b>Clases Válidas</b>	<b>Clases Inválidas</b>	<b>Resultado</b>	
		<b>Esperado</b>	<b>Real</b>
1. Hacer clic sobre vínculo Nivel 2 del submenú.		1.1 El sistema muestra texto, imagen y vínculos relacionados con el tema.	I
2. Hacer clic sobre vínculo del scroll, excepto Salas de Rehabilitación y CAT.		2.1 El sistema muestra escenario del servicio seleccionado:	S
		2.1.1 Texto e imagen.	S
		2.1.2 Se escucha una locución.	S
3. Hacer clic sobre vínculos del scroll: Salas de Rehabilitación y CAT		3.1 El sistema muestra una imagen referente al servicio.	S
		3.2 Un cuadro de texto en forma de pergamino con scroll	S
		3.3 Comienza Caso de Prueba: <i>Interactuar scroll</i> .	-
4. Hacer clic en un		4.1 Los vínculos del scroll se mueven a la derecha si interactúa con la esquina	S

extremo del scroll que contiene los servicios del nivel.		izquierda y viceversa.	
5. Hacer clic sobre vínculo <i>Nivel 1</i> del submenú.		5.1 El sistema muestra escenario correspondiente al <i>Nivel 1</i> .	S
		5.2 Comienza Caso de Prueba: <i>Interactuar escenario Servicios que ofrece, Nivel 1</i> .	–
6. Hacer clic sobre vínculo <i>Nivel 3</i> del submenú.		6.1 El sistema muestra el escenario correspondiente al <i>Nivel 3</i> .	S
		6.2 Se da inicio al Caso de Prueba: <i>Interactuar escenario Servicios que ofrece, Nivel 3</i> .	–
<b>Poscondiciones:</b>			
<b>Observaciones:</b>			

Tabla 12 CP: Interactuar escenario “Servicios que ofrece”, Nivel 2.

➤ 2.4.8.4 Casos de Prueba funcionales: Jugar juego Sopa de letras.

Los casos de prueba agrupan el conjunto de pruebas realizadas sobre el juego Sopa de letras, del módulo 1, para comprobar la interactividad del usuario con el juego en la búsqueda y selección de palabras. La programación del juego fue realizada para interactuar con el mismo a través de un solo clic. El juego contiene varias sopas de letras, ubicadas una por pantalla. Por tener todas igual funcionamiento se decidió diseñar un solo caso de prueba que evaluara:

- Interactuar juego sopa de letras 1.
- Interactuar juego sopa de letras 2.
- Interactuar juego sopa de letras 3.

**CP: Interactuar juego Sopa de letras.**

<b>Datos iniciales</b>	
<b>Caso de Uso:</b> Jugar juego sopa de letras.	<b>Requisito:</b> RF7



<b>Caso de Prueba</b>			
Interactuar juego Sopa de letras.		<b>Código del CP:</b> CPFR_07	
<b>Descripción de funcionalidad</b>	Permite comprobar el funcionamiento de los juegos una vez que el usuario va seleccionando las palabras encontradas.		
<b>Prerrequisitos</b>	Acceder al vínculo Juegos del módulo 1.		
<b>Flujo Central</b>	<p>El Caso de Prueba se inicia cuando el usuario accede a la opción (haciendo clic) <i>Juegos</i>, del menú principal. El sistema muestra en un mural titulado: Mural del Consultorio Popular:</p> <ul style="list-style-type: none"> <li>▪ En la parte superior izquierda, submenú con las opciones de Juego 1, 2, 3 y Ayuda.</li> <li>▪ A la izquierda, imagen de un tablero con letras.</li> <li>▪ A la derecha imagen, titulado “Palabras que debe buscar en el tablero”, con las palabras a encontrar.</li> <li>▪ Locución sobre cómo jugar.</li> </ul> <p>El usuario interactúa con el juego haciendo un clic.</p>		
<b>Iteraciones de entrada al Caso de Prueba</b>			
<b>Clases Válidas</b>	<b>Clases Inválidas</b>	<b>Resultado</b>	
		<b>Esperado</b>	<b>Real</b>
1. Hacer mouse over sobre imagen con palabras a encontrar.		1.1 La imagen hace un movimiento a la izquierda y luego a la derecha según se posesiona y retira el mouse respectivamente.	S
2. Hacer clic en la opción Ayuda del submenú.		2.1 Termina Caso de Prueba actual. Se da inicio al Caso de Prueba <i>Interactuar opción Ayuda</i> .	–
3. Hacer clic en la primera ó última letra de la palabra		3.1 El sistema sombrea la letra seleccionada en color azul.	I

encontrada.			
4. Hacer clic sobre letra que completa la palabra.		4.1 El sistema sombrea la palabra en color azul. 4.2 Se escucha un efecto de sonido aprobatorio. 4.3 La palabra desaparece de la lista de palabras del cuadro de texto.	S S S
	5. Hacer clic en una letra incorrecta al completar la palabra.	5.1 El sistema desmarca la letra seleccionada anteriormente. 5.2 Emite efecto de sonido desaprobatorio.	I I
6. Seleccionar letra que completa la última palabra del juego.		6.1 Animación correspondiente a palabra seleccionada 6.2 Muestra ventana de felicitación. Se da inicio al Caso de Prueba <i>Interactuar ventana Felicitación</i> .	S -
7. Hacer clic en vínculo a otro juego del submenú.		7.1 El sistema muestra el escenario del juego seleccionado. Se da inicio al caso de Prueba: <i>Interactuar juego Sopa de letras</i> .	-
<b>Poscondiciones:</b>			
<b>Observaciones:</b>			

Tabla 13 CP: Interactuar juego Sopa de letras.

➤ *Casos de prueba funcionales: Interactuar ventanas de aceptación.*

Los casos de prueba agrupan el conjunto de pruebas realizadas para comprobar la funcionalidad de pequeñas ventanas que se ejecutan ya sea por solicitud del usuario (Ayuda) o de manera automática como respuesta del sistema ante un evento (Ventana de felicitación).

✓ *Elementos a evaluar:*

- Interactuar ventana Ayuda.
- Interactuar ventana de felicitación.

**CP: Interactuar ventana Ayuda**

<b>Datos iniciales</b>			
<b>Caso de Uso:</b> Jugar juego Sopa de letras.			<b>Requisito:</b> RF 7
<b>Caso de Prueba</b>			
Interactuar ventana Ayuda.			<b>Código del CP:</b> CPRF_09
<b>Descripción de funcionalidad</b>	Permite comprobar la correcta funcionalidad de la ventana de Ayuda.		
<b>Prerrequisitos</b>	El usuario hace clic sobre la Ayuda del submenú del escenario Juegos.		
<b>Flujo Central</b>	<p>El Caso de Prueba se inicia cuando el usuario selecciona (haciendo clic) la opción <i>Ayuda</i> en el submenú del escenario <i>Juegos, del módulo 1</i>. El sistema muestra:</p> <ul style="list-style-type: none"> <li>▪ Pequeña ventana al centro del escenario del juego seleccionado con texto en forma de pergamino, titulado <i>¿Cómo jugar?</i></li> <li>▪ Opción de cerrar la ventana a través de un símbolo (X) en la esquina superior derecha.</li> </ul> <p>El usuario interactúa con la ventana haciendo clic, sobre esta.</p>		
<b>Iteraciones de entrada al Caso de Prueba</b>			
<b>Clases Válidas</b>	<b>Clases Inválidas</b>	<b>Resultado</b>	
		<b>Esperado</b>	<b>Real</b>
1. Hacer clic sobre vínculo Ayuda.		1.1 El sistema deshabilita las demás opciones del escenario. 1.2 Muestra ventana de Ayuda.	I S
2. Mover Mouse sobre el botón (X).		2. El botón hace una animación de movimiento constante de izquierda a derecha.	S

3 Hacer clic sobre el botón (X).		3.1 El sistema desaparece la ventana de Ayuda. 3.2 El escenario del juego queda completamente visible.	S S
<b>Poscondiciones:</b> El sistema retorna a la pantalla del juego desde la cual el usuario accedió a la Ayuda.			
<b>Observaciones:</b>			

Tabla 14 CP: Interactuar ventana Ayuda.

**CP: Interactuar ventana de felicitación.**

<b>Datos iniciales</b>			
<b>Caso de Uso:</b> Jugar juego Sopa de letras y Jugar juego rompecabezas.		<b>Requisito:</b> RF7 y RF10	
<b>Caso de Prueba</b>			
<b>Interactuar ventana de felicitación.</b>		<b>Código del CP:</b> CPRF_10	
<b>Descripción de funcionalidad</b>	Permite comprobar interacción del usuario con una ventana de felicitación.		
<b>Prerrequisitos</b>	El usuario debe haber seleccionado todas las palabras contenidas en la sopa de letras.		
<b>Flujo Central</b>	<p>El caso de Prueba comienza cuando el usuario selecciona la última palabra presente en la sopa de letra. El sistema muestra:</p> <ul style="list-style-type: none"> <li>▪ Pequeña ventana al centro del escenario del juego finalizado con texto de felicitación por terminar satisfactoriamente el juego.</li> <li>▪ Opción de cerrar la ventana a través de un símbolo (X) en la esquina superior derecha.</li> </ul> <p>El usuario interactúa con el sistema haciendo clic.</p>		
<b>Iteraciones de entrada al Caso de Prueba</b>			
<b>Clases Válidas</b>	<b>Clases Inválidas</b>	<b>Resultado</b>	
		<b>Esperado</b>	<b>Real</b>

1. Terminar juego.		1.1 Se deshabilitan todas las opciones de la pantalla.	I
		1.2 Se despliega ventana de felicitación.	S
2. Mover Mouse sobre el botón de cerrar (X).		2.1 El botón hace una animación de movimiento constante de izquierda a derecha.	S
3. Hacer clic sobre el botón cerrar (X).		3.1 El sistema desaparece la ventana de felicitación.	S
		3.2 El escenario del juego queda completamente visible.	S
<b>Poscondiciones:</b> El sistema retorna a la pantalla del juego recién finalizado.			
<b>Observaciones:</b>			

Tabla 15 CP: Interactuar ventana de felicitación.

➤ *Casos de prueba funcionales: Interactuar escenarios informativos módulo 2.*

Los casos de prueba agrupan el conjunto de pruebas realizadas para comprobar la funcionalidad de mostrar contenidos, de pantallas correspondientes al módulo 2: ¿Cómo construir un Consultorio popular? Fueron evaluados con este caso de prueba diferentes escenarios de igual comportamiento como:

- Requisitos mínimos para la certificación de una edificación.
- Requisitos mínimos para la certificación de un terreno.

**CP: Interactuar escenario informativo módulo 2.**

<b>Datos iniciales</b>	
<b>Caso de Uso:</b> Mostrar los Requisitos mínimos para la certificación de una edificación y Mostrar los Requisitos mínimos para la certificación de un terreno.	<b>Requisito:</b> RF6.4 y RF6.5
<b>Caso de Prueba</b>	
Interactuar escenario informativo módulo 2.	<b>Código del CP:</b> CPRF_ 011

<b>Descripción de funcionalidad</b>	Permitir interactuar correctamente con el escenario <i>Requisitos mínimos para la certificación de un terreno</i> o <i>Requisitos mínimos para la certificación de una edificación</i> , para mostrar la información referente a dicha pantalla.		
<b>Prerrequisitos</b>	Acceder al módulo 2: ¿Cómo construir un Consultorio popular?		
<b>Flujo Central</b>	<p>El Caso de Prueba tiene inicio cuando el usuario accede al vínculo <i>Requisitos mínimos para la certificación de un terreno</i> o <i>Requisitos mínimos para la certificación de una edificación</i> del menú principal. El sistema muestra:</p> <ul style="list-style-type: none"> <li>▪ Imagen de viviendas bordeando lado izquierdo inferior.</li> <li>▪ Submenú en posición vertical con requisitos a tener en cuenta (el 2<sup>do</sup> elemento no constituye vínculo) al lado izquierdo.</li> <li>▪ Texto que se despliega en una ventana y se pliega cuando el usuario hace clic en otro título del submenú.</li> <li>▪ Locución introductoria al tema presentado.</li> <li>▪ Menú principal con vínculos a otros enlaces del módulo</li> <li>▪ Barra de opciones básicas (Inicio, Audio, Salir).</li> <li>▪ Barra inferior con las opciones de ocultarla y acceder a los módulos 1 y 2 (deshabilitado por encontrarse en él).</li> </ul>		
<b>Iteraciones de entrada al Caso de Prueba</b>			
<b>Clases Válidas</b>	<b>Clases Inválidas</b>	<b>Resultado</b>	
		<b>Esperado</b>	<b>Real</b>
1. Hacer clic sobre vínculos del submenú.		1.1 El sistema muestra texto que se despliega en una ventana y se pliega cuando el usuario hace clic en otro título del submenú.	S
<b>Poscondiciones:</b>			
<b>Observaciones:</b>			

Tabla 16 CP: Interactuar escenario informativos módulo 2.

➤ *Casos de prueba funcionales: Interactuar barra de opciones básicas.*

Los casos de prueba agrupan el conjunto de pruebas realizadas para comprobar la funcionalidad de los botones que conforman la barra de opciones básicas (controlador de audio de fondo, inicio, salir) desde cualquier pantalla de la multimedia.

✓ *Elementos a evaluar:*

- Interactuar opción salir.
- Interactuar opción inicio.
- Interactuar opción controlador de audio.

**CP: Interactuar opción Controlar audio de fondo.**

<b>Datos iniciales</b>			
<b>Caso de Uso:</b> Controlar audio de fondo.		<b>Requisito:</b> RF2	
<b>Caso de Prueba</b>			
Interactuar opción controlar <i>audio de fondo</i> .		<b>Código del CP:</b> CPRF_ 015	
<b>Descripción de funcionalidad</b>	Permite comprobar la ejecución correcta del sistema frente a las interacciones del usuario del botón de Audio, desde cualquier escenario.		
<b>Prerrequisitos</b>	Estar ubicado en un escenario de la aplicación.		
<b>Flujo Central</b>	El Caso de Prueba se inicia cuando el usuario interactúa con el botón controlador de audio de fondo desde cualquier escenario. Este se activa o desactiva en dependencia del estado actual en que se encuentra. Termina así el Caso de Prueba.		
<b>Iteraciones de entrada al Caso de Prueba</b>			
<b>Clases Válidas</b>	<b>Clases Inválidas</b>	<b>Resultado</b>	
		<b>Esperado</b>	<b>Real</b>
1. Mover Mouse sobre el botón de audio.		1.1 La imagen del botón hace una animación de parpadeo.	S
2. Hacer clic sobre el botón <i>audio</i> .		2.1 El audio de fondo se deja de escuchar si estaba en modo de reproducción y viceversa.	S

<b>Poscondiciones:</b>
<b>Observaciones:</b>

Tabla 17 CP: Interactuar opción Controlar audio de fondo.

**CP: Interactuar opción Inicio**

<b>Datos iniciales</b>			
<b>Caso de Uso:</b> Acceder a Inicio		<b>Requisito:</b> RF4	
<b>Caso de Prueba</b>			
Interactuar opción <i>Inicio</i> .		<b>Código del CP:</b> CPRF_016	
<b>Descripción de funcionalidad</b>	Permite comprobar la ejecución correcta del sistema frente a las interacciones del usuario del botón Inicio, desde cualquier escenario.		
<b>Prerrequisitos</b>	Estar ubicado en una pantalla excepto en la de Inicio.		
<b>Flujo Central</b>	<p>El Caso de Prueba se inicia cuando el usuario interactúa con el botón <i>Inicio</i>.</p> <ul style="list-style-type: none"> <li>▪ Se cierra el escenario donde se encuentra el usuario.</li> <li>▪ Se muestra inmediatamente la pantalla de Inicio de presentación de la multimedia desde donde se accede a los módulos.</li> </ul>		
<b>Iteraciones de entrada al Caso de Prueba</b>			
<b>Clases Válidas</b>	<b>Clases Inválidas</b>	<b>Resultado</b>	
		<b>Esperado</b>	<b>Real</b>
1. Mover Mouse sobre el botón <i>Inicio</i> en la barra de opciones básicas.  2. Hacer clic sobre botón <i>Inicio</i> .		1.1 La imagen del botón toma coloración oscura y dentro unos puntos luminosos se iluminan de manera alterna.	S
		2.1 Se cierra el escenario donde se encuentra el usuario.	S
		2.2 Se muestra la pantalla de Inicio de acceso a los módulos.	S
		2.3 Se deshabilita el botón en la pantalla de inicio.	I



<b>Poscondiciones:</b> El sistema muestra la pantalla Inicial de la multimedia, correspondiente a la presentación del producto.
<b>Observaciones:</b>

Tabla 18 CP: Interactuar opción Inicio.

➤ *Casos de prueba funcionales: Permitir navegabilidad.*

Los casos de prueba agrupan el conjunto de pruebas funcionales realizadas sobre los diferentes vínculos utilizados en la multimedia para permitir la navegabilidad del usuario por la misma entre escenarios de un mismo módulo y entre módulos. Se evalúan vínculos en menú principal de cada uno de los módulos, submenú de escenarios particulares y como caso especial menú inferior de acceso a módulos, al que se le diseñó un caso de prueba particular pero respondiendo al mismo requisito: permitir navegabilidad.

**CP: Interactuar vínculos de navegabilidad.**

<b>Datos iniciales</b>			
<b>Caso de Uso:</b> Permitir navegabilidad		<b>Requisito:</b> RNF4	
<b>Caso de Prueba</b>			
Interactuar menú principal.		<b>Código del CP:</b> CPRNF_01	
<b>Descripción de funcionalidad</b>	Permite comprobar la navegabilidad por los diferentes escenarios de la multimedia desde los accesos que tenga el usuario en el escenario donde se encuentre.		
<b>Prerrequisitos</b>	Acceder (haciendo clic) a un vínculo.		
<b>Flujo Central</b>	El Caso de prueba tiene inicio cuando el usuario interactúa (haciendo clic) con un vínculo que constituya enlace a otro escenario. El sistema cierra el escenario actual mostrando el solicitado.		
<b>Iteraciones de entrada al Caso de Prueba</b>			
<b>Clases Válidas</b>	<b>Clases Inválidas</b>	<b>Resultado</b>	
		<b>Esperado</b>	<b>Real</b>

1. Hacer clic sobre un vínculo (x).		1.1 Se muestra escenario correspondiente al vínculo seleccionado.	S
<b>Poscondiciones:</b>			
<b>Observaciones:</b>			

Tabla 19 CP: Interactuar vínculos de navegabilidad.

- *Casos de prueba de instalación: Instalación de la multimedia en los diferentes sistemas de software.*

Los casos de prueba agrupan el conjunto de pruebas realizadas para comprobar la instalación del producto desde los diferentes sistemas operativos para los que fue desarrollado. (RNF2)

**CP: Instalación de la Multimedia en los diferentes sistemas de software**

<b>Datos iniciales</b>		
<b>Caso de Uso:</b> El sistema debe correr sobre los sistemas operativos Windows 98, Windows 2000, Windows XP.		<b>Requisito:</b> RNF3
<b>Caso de Prueba</b>		
Instalación de la Multimedia en los diferentes sistemas de software.		<b>Código del CP:</b> CPRNF_03
<b>Descripción de funcionalidad</b>	Permitir comprobar la correcta ejecución de la multimedia desde cualquiera de los sistemas operativos de la familia Windows para los que fue diseñada, especificada en los requisitos no funcionales.	
<b>Prerrequisitos</b>	Ejecutar la aplicación en cualquiera de los sistemas operativos Windows (Windows 98, Windows 2000, Windows XP).	
<b>Flujo Central</b>	El Caso de Prueba se inicia cuando el usuario instala la aplicación en máquinas con sistema operativo Windows (Windows 98, Windows 2000, Windows XP). El usuario ejecuta la aplicación en un sistema operativo especificado, a fin de comprobar su compatibilidad con este.	
<b>Iteraciones de entrada al Caso de Prueba</b>		
<b>Clases</b>	<b>Clases</b>	<b>Resultado</b>

<i>Válidas</i>	<i>Inválidas</i>	<i>Esperado</i>	<i>Real</i>
1. Ejecutar la multimedia en PC con sistema operativo Windows especificado.		1.1El sistema reconoce el producto.	S
2. Interactuar con la multimedia.		2.1El sistema permite la navegabilidad con un correcto funcionamiento del producto.	S
<b>Poscondiciones:</b>			
<b>Observaciones:</b>			

Tabla 20 CP: Instalación de la Multimedia en los diferentes sistemas de software.

➤ *Casos de prueba de portabilidad: Reconocer eventos de hardware.*

Los casos de prueba agrupan el conjunto de pruebas realizadas para comprobar si el sistema reconoce los eventos de hardware de Mouse para la navegación por el producto y de bocinas que permitan amplificar las locuciones acerca de la información expuesta. Son realizadas sobre los requisitos suplementarios.

**CP: Reconocer eventos de hardware**

<i>Datos iniciales</i>	
<b>Caso de Uso:</b> El diseño de la interfaz debe reconocer eventos de mouse y/o teclado para la navegación del sistema.	<b>Requisito:</b> RNF1
<i>Caso de Prueba</i>	
Reconocer eventos de hardware	<b>Código del CP:</b> CPRNF_04
<b>Descripción de funcionalidad</b>	Permitir comprobar si la multimedia reconoce elementos de hardware periféricos como: teclado y/o Mouse, tarjeta de sonido, audífonos y/o bocinas, necesario para la ejecución correcta del producto.
<b>Prerrequisitos</b>	Ejecución por parte del usuario de la multimedia a través de la navegabilidad por los escenarios de cada módulo con la utilización de los periféricos requeridos.

<b>Flujo Central</b>	El Caso de Prueba se inicia cuando el usuario decide interactuar con la aplicación. El usuario accede a los diferentes escenarios de los respectivos módulos desde donde se encuentre a través de las opciones de navegabilidad, beneficiándose con las diferentes opciones que le ofrece el producto mediante los eventos de hardware requeridos.		
<b>Iteraciones de entrada al Caso de Prueba</b>			
Clases Válidas	Clases Inválidas	Resultado	
		Esperado	Real
1. Acceder a escenarios y servicios del producto a través del uso de periféricos.		1.1El sistema permite el acceso solicitado una vez reconocido los eventos de hardware periféricos.	S
<b>Poscondiciones:</b>			
<b>Observaciones:</b>			

Tabla 21 CP: Reconocer eventos de hardware.

➤ **Casos de prueba de solidez: Correspondencia entre diseño e interfaz del sistema.**

Los casos de prueba agrupan el conjunto de pruebas realizadas para comprobar la apariencia del sistema. Comprende la correspondencia entre solicitud del cliente, diseño aplicado al producto y la salida de interfaz al usuario. Para ello se verificarán elementos como: ortografía a textos informativos, calidad de elementos tratados (imágenes, audio, textos), navegabilidad y la interrelación de todos estos en el objetivo de transmitir una misma idea.

**CP: Correspondencia entre diseño e interfaz del sistema**

<b>Datos iniciales</b>	
<b>Caso de Uso:</b> _____	<b>Requisito:</b> RNF6
<b>Caso de Prueba</b>	
Correspondencia entre diseño e interfaz del sistema.	<b>Código del CP:</b> CPRNF_05

<b>Descripción de funcionalidad</b>	Permitir comprobar la correspondencia entre el tema tratado y los medios para la transmisión de dicha información desde los diferentes escenarios de la multimedia.		
<b>Prerrequisitos</b>	Interacción del usuario con el producto desde los niveles mínimos (escenarios) hacia la integración de estos desde los diferentes módulos de manera que navegue por todo el sistema.		
<b>Flujo Central</b>	El Caso de Prueba tiene inicio cuando el usuario abre la multimedia y decide navegar desde los diferentes módulos de acceso a la información.  Se muestra escenarios coherentes en cuanto a ortografía, correspondencia entre imágenes, textos y locuciones de un mismo escenario y entre escenarios de un mismo módulo.		
<b>Iteraciones de entrada al Caso de Prueba</b>			
<b>Clases válidas</b>	<b>Clases inválidas</b>	<b>Resultado</b>	
		<b>Esperado</b>	<b>Real</b>
1. Acceder a un escenario.		1.1 Se muestra en escenario correspondencia entre texto imagen y locución.	S
		1.2 Correspondencia entre guión final y producto terminado.	I
<b>Poscondiciones:</b> En el sistema existe correspondencia entre los diferentes medios que se utilizan para transmitir la información y esta.			
<b>Observaciones:</b>			

Tabla 22 CP: Correspondencia entre diseño e interfaz del sistema.

➤ **Caso de prueba a la documentación: Integridad y disponibilidad de la documentación.**

Los casos de prueba agrupan el conjunto de pruebas realizadas a la documentación del producto (guión técnico, guión final, lista de requerimientos) para comprobar su integridad en cuanto al contenido y disponibilidad para su uso.

**CP: Comprobar integridad y coexistencia en la documentación.**



<b>Datos iniciales</b>			
<b>Caso de Uso:</b> _____		<b>Requisito:</b> RNF7	
<b>Caso de Prueba</b>			
Comprobar integridad y coexistencia en la documentación.		<b>Código del CP:</b> CPRS_06	
<b>Descripción de funcionalidad</b>	Permite comprobar el correcto estado de la documentación del producto en cuanto a claridad editorial y nivel de actualización, necesario para diseñar y ejecutar las pruebas.		
<b>Prerrequisitos</b>	Acceder a la documentación general del producto.		
<b>Flujo Central</b>	<p>El caso de prueba se inicia cuando se accede a la revisión de la documentación. Se evalúan aspectos como:</p> <ul style="list-style-type: none"> <li>- Nivel de actualización,</li> <li>- Claridad y coherencia en las ideas,</li> <li>- Integridad de la información,</li> <li>- Nivel de detalle en los requerimientos para poder evaluar el producto del mismo modo.</li> <li>- Documentación completa.</li> </ul>		
<b>Iteraciones de entrada al Caso de Prueba</b>			
<b>Clases válidas</b>	<b>Clases inválidas</b>	<b>Resultado</b>	
		<b>Esperado</b>	<b>Real</b>
<b>Poscondiciones:</b>			
<b>Observaciones:</b> No se hace uso de las iteraciones de entrada al caso de prueba por desarrollarse de manera general el objeto analizado.			

Tabla 23 CP: Comprobar integridad y coexistencia en la documentación.

➤ *Caso de prueba: Interfaz gráfica de elementos en escenario.*

El caso de prueba contiene una lista de chequeo de estándares para elementos y funciones de la interfaz gráfica de la multimedia. Permite comprobar el diseño gráfico de múltiples elementos en escenarios como:

vínculos, menús operantes con el mouse y/o teclado, subtítulos, tool tips, etc., con los que el usuario se relaciona directamente y facilitan la utilización del sistema. Su comportamiento se analiza de acuerdo a su agrupación en el contexto en que están ubicados:

- Textos de funciones de los menús (menú principal, módulo 1 y 2, menú inferior).
- Vínculos en tira de imágenes de scroll.

**CP: Chequeo de estándares para elementos de interfaz gráfica.**

<b>Datos iniciales</b>				
<b>Caso de Uso:</b> _____		<b>Requisito:</b> RNF2		
<b>Caso de Prueba</b>				
Chequeo de estándares para elementos de interfaz gráfica.		<b>Código del CP:</b> CPRS_07		
<b>Descripción de funcionalidad</b>	Permite comprobar si diseño gráfico de elementos en escenarios cumplen con el estándar preestablecido para el producto.			
<b>Prerrequisitos</b>	Interactuar con la interfaz gráfica de elementos en un escenario de la multimedia.			
<b>Flujo central</b>	_____			
<b>Lista de chequeo de estándares</b>				
<b>Elemento a revisar</b>			<b>Si</b>	<b>No</b>
<b>Scroll</b>				
¿Está el scroll vertical ubicado a la derecha?			X	
¿Están las barras de scroll vertical iguales a la altura de sus campos asociados?				X
¿Está el scroll horizontal ubicado en la parte superior?			X	
¿Están las barras de scroll horizontal iguales al ancho de sus campos asociados?				X
<b>Menús operantes con el mouse</b>				
¿Se muestra la barra de menú apropiadamente en el contexto apropiado?			X	
¿Están listadas adecuadamente todas las funciones del menú?			X	
¿Es correcto el tipo, tamaño y formato del texto de las funciones del menú?			X	

¿Están resaltadas las funciones del menú dependiendo del contexto de las operaciones actuales de la ventana?		X
¿Son suficientemente claros los nombres de la funciones del menú?	X	
¿Cambian adecuadamente el cursor, el indicador de procesamiento y puntero al invocar diferentes operaciones?	X	
¿Realizan la animación correspondiente ante un mouse over las funciones del menú?		X
<b>Tooltip</b>		
¿Tienen el formato adecuado?	X	
¿Están visibles cuando se requiere que reverencien un contenido de las acciones de la ventana?		X
<b>Botones</b>		
Si están ubicados horizontalmente, ¿están en la parte superior de la pantalla?	X	
¿Los botones requieren y tienen Tooltip?		X
¿Tienen la misma altura los botones organizados horizontalmente?	X	
¿Está colocada la opción más frecuente a la izquierda o en el tope, según corresponda?	X	

Tabla 24 CP: Chequeo de estándares para elementos de interfaz gráfica de usuario (IGU)

### 2.4.9 Criterios de validación

Para la certificación del éxito o no de una prueba se tuvo en cuenta el análisis de los resultados esperados contenidos en la descripción de los casos de prueba con los resultados reales. Se considera:

- ✓ Prueba Satisfactoria: Si el resultado esperado es idéntico al resultado obtenido.
- ✓ Prueba Insatisfactoria: Si el resultado esperado es diferente al resultado obtenido.

En caso de no coincidir, el equipo encargado de la prueba determinará si la discordancia supone un fallo en la validación del sistema y si debe continuarse con los restantes casos de prueba o bien dar por finalizada la validación del sistema.



### 2.4.10 Definición de entregables

Del proceso de pruebas se obtienen los siguientes documentos generados durante las fases propuestas:

- ✓ Plan de Prueba.
- ✓ Especificaciones de prueba.
- ✓ Especificaciones de casos de prueba.
- ✓ Reportes de las pruebas (registro de defectos y no conformidades).

### 2.4.11 Recursos requeridos

En esta sección se describen los recursos necesarios para realizar el proceso de pruebas, especificando las características necesarias y deseables del ambiente de pruebas. Se tuvo en cuenta tanto recursos humanos (hombres) como de trabajo (computadoras).

#### Recursos humanos:

<b>Multimedia: Barrio Adentro al alcance de tus manos</b>		
<b>Rol</b>	<b>Mínimos recursos recomendados</b>	<b>Responsabilidades específicas</b>
Jefe o administrador de prueba	Yanet Brito Riverol	Proporcionar una gestión adecuada. Responsable de: <ul style="list-style-type: none"> <li>✓ Proporcionar una dirección técnica (progreso y efectividad de las pruebas).</li> <li>✓ Adquirir los recursos apropiados.</li> <li>✓ Plan de pruebas.</li> <li>✓ Resumen de los resultados de las pruebas.</li> <li>✓ Negociar o acordar las pruebas.</li> </ul>
Diseñador de prueba.	Yanet Brito Riverol	Identificar, priorizar e implementar los casos de prueba. Responsable de: <ul style="list-style-type: none"> <li>✓ Definir configuración del ambiente de las pruebas.</li> <li>✓ Identificar las técnicas de pruebas.</li> <li>✓ Generar el Plan de pruebas.</li> <li>✓ Diseñar los Casos de prueba.</li> <li>✓ Evaluar el esfuerzo de prueba.</li> </ul>

Probador o Verificador	Yanet Brito Riverol	Ejecutar pruebas diseñadas. Responsable de: <ul style="list-style-type: none"> <li>✓ Ejecutar la suite de pruebas.</li> <li>✓ Documentar los errores.</li> <li>✓ Analizar fallos de prueba.</li> </ul>
------------------------	---------------------	--

Tabla 25 Descripción de Recursos humanos.

**Recursos de trabajo:**

<b>Recurso</b>	PC
<b>Cantidad</b>	1
<b>Nombre/Tipo</b>	Repositorio de pruebas (diseño y ejecución de pruebas).

Tabla 26 Recursos de hardware.

**2.4.12 Planificación**

Para la planificación de las actividades, se hizo una asignación de tiempo según las fases o etapas definidas, ver Fig. 6. Cada fase está integrada por tareas específicas de las cuales se detalla: nombre de la tarea a realizar, período de duración y fecha de comienzo y fin, ver Fig. 5. Como responsable de todas las actividades estuvo: Yanet Brito Riverol.

Nombre de tarea	Duración	Comienzo	Fin
<b>Planificar Pruebas</b>	<b>25 días?</b>	<b>lun 1/15/07</b>	<b>vie 2/16/07</b>
Identificar aspectos ge	5 días?	lun 1/15/07	vie 1/19/07
Identificar requisitos	4 días?	lun 1/22/07	jue 1/25/07
Desarrollar estrategia	11 días?	vie 1/26/07	vie 2/9/07
Identificar recursos	5 días?	lun 2/12/07	vie 2/16/07
<b>Diseñar pruebas</b>	<b>25 días?</b>	<b>mar 2/20/07</b>	<b>lun 3/26/07</b>
Desarrollo de pruebas	10 días?	mar 2/20/07	lun 3/5/07
Dtnar y diseñar casos	12 días?	mar 3/6/07	mié 3/21/07
Configurar ambiente de	3 días?	jue 3/22/07	lun 3/26/07
<b>Ejecutar pruebas</b>	<b>28 días?</b>	<b>jue 3/29/07</b>	<b>lun 5/7/07</b>
Implementación y ejec	13 días?	jue 3/29/07	lun 4/16/07
Evaluar ejecución de p	4 días?	mar 4/17/07	vie 4/20/07
Verificar resultados	6 días?	lun 4/23/07	lun 4/30/07
Registrar defectos	4 días?	mié 5/2/07	lun 5/7/07
<b>Evaluar Pruebas</b>	<b>16 días?</b>	<b>mié 5/9/07</b>	<b>mié 5/30/07</b>
Analizar defectos	6 días?	mié 5/9/07	mié 5/16/07
Evaluar resultados	3 días?	jue 5/17/07	lun 5/21/07
Evaluar efectividad	3 días?	mar 5/22/07	jue 5/24/07
Documentar evaluació	4 días?	vie 5/25/07	mié 5/30/07

Fig. 5 Planificación de etapas de pruebas según actividades.

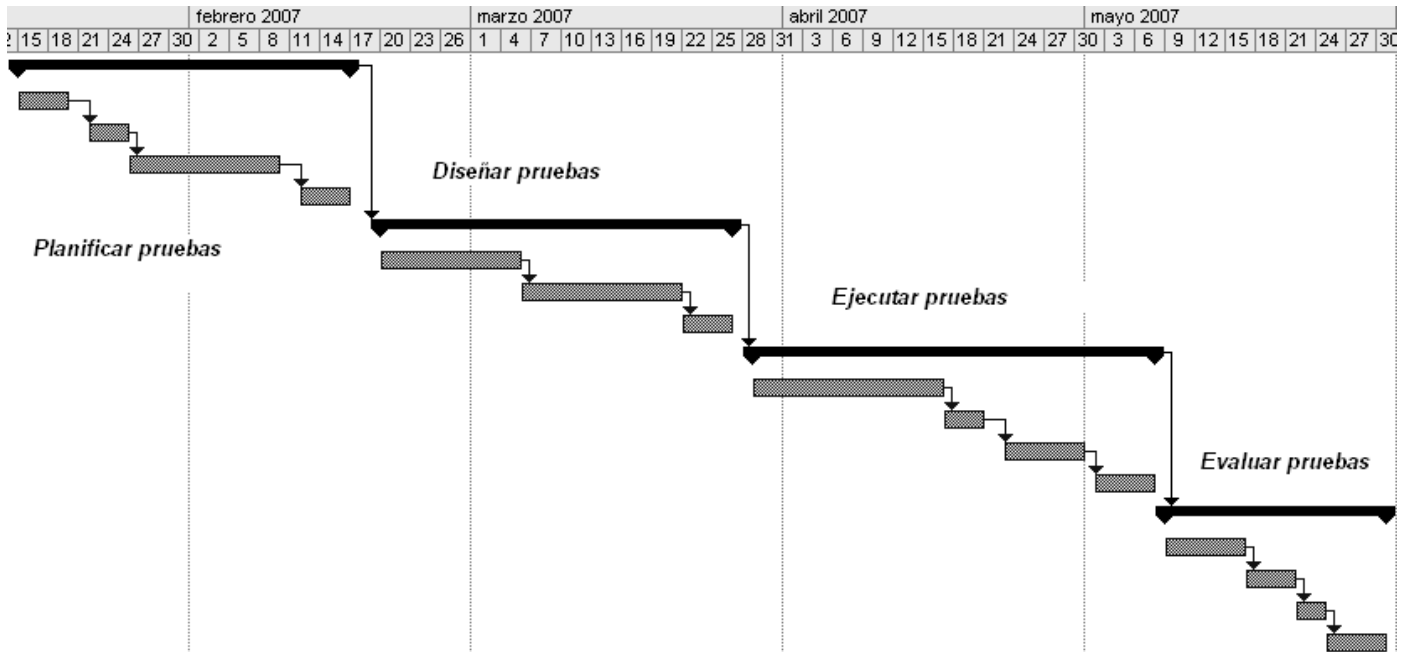


Fig.6 Planificación de las actividades (Diagrama de Gantt)

## 2.5 Conclusiones

El proceso de pruebas genera información muy valiosa acerca de la madurez de un producto software, que permite tomar decisiones al respecto. Su planificación es muy importante pues además de definir la estrategia y establecer los recursos necesarios, permite realizar estimaciones que establecen una base para el control y seguimiento durante la ejecución del proyecto, así como determinar si el nivel de calidad es el requerido. El proceso descrito en este capítulo, realizado a partir de la selección de los niveles de prueba a implementar, abarcó una descripción detallada de las actividades realizadas para detectar errores y no justificar su ausencia.

## **CAPÍTULO III: RESULTADOS DE LAS PRUEBAS**

### **3.1 Introducción**

De manera general se habla mucho de la documentación, sin embargo, es común enfrentar procesos de desarrollo de software donde no se hace, o no se le asigna presupuesto, o no se la mantiene. En el mejor de los casos la documentación existe, pero no está al día con el proyecto. Lo importante es la disponibilidad de la documentación, que se necesita en el momento en que se la necesita. El presente capítulo tiene como objetivo realizar un análisis de las pruebas ejecutadas a la multimedia para verificar en que medida se cumplen o no los requerimientos que se decidió evaluar y su repercusión en la calidad del producto final. Para ello se realiza un estudio de los defectos detectados a partir de varios aspectos que inciden en los mismos.

### **3.2 Documentación de los resultados**

En un proceso de pruebas la documentación consta como un requisito imprescindible a desarrollar. Debe escribirse sobre el análisis, el diseño y los resultados de la ejecución del programa a partir del conjunto de valores definidos previamente en el plan de pruebas. Esto ofrece un control de los casos que se van a aplicar en cada tipo de prueba así como de los resultados que se van obteniendo con su aplicación.

Estos son aspectos significativos que mediante la documentación llegan al desarrollador para su posterior trabajo de rectificación de errores de cada una de las partes del sistema. La documentación de las no conformidades debe ser detallada y tener un orden lógico, comprensible para su posterior uso.

### **3.3 Resultados de las pruebas**

En el caso particular de esta investigación los resultados se documentaron con la precisión y nivel de detalle requerido, ya que es muy difícil asimilar el trabajo realizado por otras personas cuando no se tiene una descripción del mismo para su comprensión. Se tienen en cuenta elementos ilustrativos como: un identificador del error, responsable de la ejecución, ambiente de ejecución (hardware y software de sistema y apoyo) tabla de registro de defectos y no conformidades (Ver Fig.6) con el propósito registrar,

entregar y administrar los defectos encontrados en la ejecución del plan de pruebas, además de un breve resumen estadístico sobre los resultados obtenidos de manera global y su incidencia en el producto. Los resultados se agruparon según los tipos de pruebas ejecutadas, siguiendo el orden de nivel modular y luego sistema.

Para el seguimiento y reporte de los defectos en cuanto a la priorización que se les dio se calificaron en las siguientes categorías (González, 2002)

- ✓ **Crítico:** Denota una función inutilizable que causa un término anormal o una falla general, o cuando un cambio en un área de la aplicación causa un problema en otra parte.
- ✓ **Severo:** Una función no actúa como fue requerido o diseñado, o un objeto de interfaz no trabaja como se muestra.
- ✓ **Advertencia:** La función trabaja, pero no tan rápidamente como se espera, o no se ajusta a las normas y convenciones.
- ✓ **Cosmético:** No es crítico para el funcionamiento del sistema: palabras con mala ortografía, formato incorrecto, mensajes de error vagos o confusos o advertencias.

<i>N°</i>	<i>No conformidad</i>	<i>Descripción de la no conformidad</i>	<i>Localización</i>
<i>Recomendaciones</i>			

Fig.7 Tabla de defectos y no conformidades.

### 3.3.1 Resultados específicos por pruebas

Las pruebas fueron aplicadas por un probador (T1: Yanet Brito Riverol). Se ejecutaron utilizando 1 PC con sistema operativo Windows XP, con 256 MByte de RAM y microprocesador Pentium 4. La multimedia se ejecutó desde un CD, dando cumplimiento al requisito de portabilidad y evaluándolo a la vez.

#### ➤ *Resultados prueba funcional*

<i>N°</i>	<i>No conformidad</i>	<i>Descripción de la no conformidad</i>	<i>Localización</i>

1	No visibles vínculos a Nivel 1 y Nivel 2.	Al hacer clic sobre el vínculo del submenú, Nivel 2, desde el propio escenario, desaparecen los demás vínculos del submenú (Nivel 1 y Nivel 3).	Pantalla: Servicios que ofrece/ Nivel 2, vínculo del submenú, Nivel2. Ver anexo 5
<b>Recomendaciones:</b> Los demás vínculos del submenú (Nivel 1 y Nivel 3) deberán permanecer visibles y habilitados, de manera que el usuario pueda acceder a ellos desde el propio escenario del Nivel 2 siempre que lo desee.			
2	No emitido efecto de sonido desaprobatario al seleccionar letra.	Al seleccionar la letra que completa la palabra si esta se ubica en un sentido diferente al horizontal o vertical el sistema no emite el efecto de sonido desaprobatario.	Pantalla: Juego Sopa de letras / Juego1, Juego 2, Juego 3.
<b>Recomendaciones</b>			
3 3.1	Habilitada la navegabilidad sin cerrar pequeña ventana desplegada.	Una vez desplegada la ventana de felicitación ó ayuda el usuario puede acceder a otro escenario seleccionado sin antes cerrar la misma. Esta permanece visible hasta tanto el usuario no decida cerrarla.	- Pantalla: Juegos correspondiente tanto al módulo 1 como al módulo 2.  - Pantalla: Juegos, Sopa de letras, vínculo del submenú Ayuda (Ver anexo 6).
<b>Recomendaciones:</b> Debe inhabilitarse el acceso a cualquier pantalla hasta tanto el usuario no cierre la ventana de felicitación.			
4	Botón de inicio no deshabilitado.	Al hacer clic en el botón Inicio desde la propia pantalla se invalida la ejecución de la multimedia mostrando el escenario sin contenido (solo botón de audio). Hay reiniciar la aplicación para	Pantalla de Inicio. Menú de opciones básicas. Clic en Botón Inicio.

		continuar interactuando con ella.	
<b>Recomendaciones:</b>			
5	Letra seleccionada no toma coloración azul.	Al seleccionar la última letra de la palabra que ya está marcada por pertenecer a otra palabra encontrada, esta cambia de color azul a verde.	Módulo1, Juegos, Juego1(Ver anexo 7).
<b>Recomendaciones:</b>			

Tabla 27 Resultados de pruebas funcionales.

➤ **Resultados prueba de interfaz gráfica de usuario (IGU)**

<b>N°</b>	<b>No conformidad</b>	<b>Descripción de la no conformidad</b>	<b>Localización</b>
6	Tool tips (Consulta integral) no visible.	Al realizar mouse over sobre algunas zonas del ícono (imagen de consulta médica) el sistema no muestra el tool tips "Consulta integral"	Módulo1.Pantalla: <i>Servicios que ofrece/ Nivel 1</i> , en la tira de imágenes superior, vínculo a "Consulta integral"
<b>Recomendaciones:</b> El tool tips deberá permanecer visible mientras el mouse haga contacto con el vínculo.			
7 7.1	No se observa en vínculo animación completa ante un evento de mouse over.	-Al realizar un evento de mouse over sobre un vínculo el sistema emite solo parcialmente el efecto de sonido.  -Al realizar un mouse over sobre un ícono de la tira de imágenes que contiene el scroll no se emite el correspondiente efecto de sonido.	-Módulo1.Pantalla: <i>Servicios que ofrece/ Nivel 1</i> , en la tira de imágenes superior, vínculo a "Consulta integral".  -Módulo 1: <i>¿Qué es Barrio Adentro? / Servicios que ofrece / Nivel2</i> . En la tira de imágenes que contiene el scroll todos los íconos, excepto Rayos X.



<b>Recomendaciones</b>			
8	Vínculo sin resaltar la función realizada.	-Al acceder a un escenario el vínculo correspondiente a este no muestra transformación en respuesta a la función realizada (borde rojo).  -Al realizar un mouse over sobre vínculo del escenario en que se encuentra navegando el usuario se escucha efecto de sonido.	-Módulo1.Pantalla: <i>Servicios que ofrece/ Nivel 2</i> , vínculos en scroll.  -Módulo1: <i>¿Qué es Barrio Adentro? / Servicios que ofrece / Nivel 1. Vínculo Nivel 1</i> del submenú.
8.1			
<b>Recomendaciones</b>			
9	Zona caliente de vínculos con límite indiscriminado.	Al hacer clic fuera del vínculo en un espacio considerable se accede al escenario correspondiente al mismo.	-Módulo2.Menú inferior, vínculo a módulo 1.  -Pantalla de presentación del módulo2, vínculo representado por imagen de hombre.
9.1			
<b>Recomendaciones</b>			
10	Interfaz deficiente para intercambio de información.	Las presentaciones son muy extensas, pueden aburrir al usuario posibilitando que no continúe con la navegabilidad o disminuir el nivel de aceptación del mismo.	Presentación general del producto. Créditos
<b>Recomendaciones</b>			
11	La interfaz requiere familiarización para navegar.	El usuario pierde tiempo identificando los recursos y servicios de cada escenario.	Producto multimedia en general

<b>Recomendaciones</b>			
12	Inconsistencia en la localización de funciones.	El usuario requiere de tiempo para reconocer los vínculos del escenario al carecer de identificación (Tool tips, imágenes representativas)	Vínculos a servicios y opciones de las diferentes pantallas.
<b>Recomendaciones</b>			

Tabla 28 Resultados de pruebas a interfaz gráfica.

➤ **Resultados prueba de solidez**

<b>N°</b>	<b>No conformidad</b>	<b>Descripción de la no conformidad</b>	<b>Localización</b>
13	Errores ortográficos, de puntuación, y de tecleo en el texto (Ver anexo 8).	<b>Palabras sin acentuar:</b>	- Módulo1. Pantalla: Antecedentes, Texto informativo, 5 <sup>ta</sup> línea.
13.1		-válida	- Módulo1. Pantalla: Antecedentes. Texto informativo 1 <sup>ra</sup> línea.
13.2		-médico	-Módulo1. Pantalla: Servicios que ofrece Nivel 3. Texto informativo, 6 <sup>ta</sup> línea.
13.3			- Módulo 2. Escenarios Requisitos mínimos para la certificación de un terreno y Requisitos mínimos para la certificación de una edificación. Texto del submenú, <i>Debe existir un médico cubano.</i>
13.4		Preposición escrita con letra mayúscula (am <b>A</b> 12:00 am).	-Módulo 1.Pantalla Servicios que ofrece, Nivel 1. Texto informativo 4 <sup>ta</sup> línea.
13.5		Sustantivo propio escrito con minúscula ( <b>barrio adentro</b> ).	Módulo1. Pantalla Servicios que ofrece, Nivel2. Texto informativo 1 <sup>ra</sup> línea.
13.6		Sustantivo común escrito con mayúscula	-Módulo 2. Pantalla Requisitos mínimos para la certificación de una edificación / Servicios

13.7	(: Agua )	básicos. Texto informativo 1 <sup>ra</sup> línea.
13.8	(:Como)	-Módulo 2. Pantalla Requisitos mínimos para la certificación de un terreno / Servicios básicos. Texto informativo 1 <sup>ra</sup> línea. -Módulo2. Pantalla Requisitos mínimos para la certificación de un terreno / Edificaciones. Texto informativo 2 <sup>da</sup> línea.
13.9	Uso de comillas diferentes al inicio y fin de un mismo texto (”).	Módulo1. Pantalla Antecedentes. Inicio y fin del texto informativo.
13.10	Empleo de coma en lugar de punto (, Si )	- Módulo 2. Requisitos mínimos para la certificación de una edificación / Título de propiedad. Texto informativo 3 <sup>ra</sup> línea.
13.11	Paréntesis sin cerrar (SUMA	-Módulo1. Pantalla Servicios que ofrece / Nivel 2. Icono del scroll: Suma. Texto informativo 1 <sup>ra</sup> línea.
13.12	Abreviatura de la unidad de medida metro, escrita incorrectamente (120 mts <sup>2</sup> )	- Módulo 2. Pantalla: Requisitos mínimos para la certificación de una edificación / Área mínima. Texto informativo 1 <sup>ra</sup> línea.
13.13	Abreviatura mal escrita	-Módulo 1.Pantalla Servicios que ofrece, Nivel 1. Texto informativo 4 <sup>ta</sup> línea.
13.14	Cambio de si no por <b>sino</b>	-Módulo 1. Pantalla Servicios que ofrece / Nivel1 / Vacunación. Texto informativo 3 <sup>ra</sup> línea.
<b>Recomendaciones</b>		

14	No existe correspondencia total entre guión final y producto terminado (imágenes, funciones y locuciones).	El contenido del texto y la locución no es el mismo. Cambio de activar por operativizar.	Módulo1. Pantalla Servicios que ofrece / Nivel 3.
14.1		<b>El Sistema no muestra:</b> Scroll luego de concluir el movimiento de una secuencia de imágenes.	-Módulo1. Pantalla Antecedentes, zona inferior, posición horizontal.
14.2		Imagen de comunidades en primer plano, que represente un barrio urbano, rural, e indígena; con un efecto que indique movimiento de esas personas al ambulatorio.	-Módulo1. Pantalla Población beneficiaria.
14.3		Imagen de una ingeniera con un efecto de animación que mueve la boca, después de aparecer todos los textos de la pantalla.	- Módulo 2. Pantalla: Requisitos mínimos para la certificación de un terreno (Ver anexo 9).
14.4		Imagen de un terreno en construcción.	- Pantalla de presentación del módulo 2: ¿Cómo construir un Consultorio popular? En la zona derecha.
14.5			- Pantalla de presentación del módulo2. Imagen izquierda (Casa) y derecha (Terreno).
14.6		Tool tips de color Azul subrayado siempre visible.	- Modulo 2, submenú con los requisitos a tener en cuenta tanto para la certificación de

14.7		Se despliega identificación el servicio en tool tips en la parte superior del ícono.	un terreno como para una edificación. Excepto el segundo elemento.  -Módulo1. Pantalla servicios que ofrece, Nivel 1 y Nivel 2. Tira de imágenes en la parte superior que constituyen iconos.
<b>Recomendaciones</b>			

Tabla 29 Resultados de Prueba de solidez.

➤ **Resultados prueba de documentación**

<b>N°</b>	<b>No conformidad</b>	<b>Descripción de la no conformidad</b>	<b>Localización</b>
15	Errores ortográficos	Palabras sin acentuar, escritas incorrectamente.	Guión final (Ver anexo 10).
<b>Recomendaciones:</b> Debe revisarse bien toda la documentación, pues muchos de los errores ortográficos de la multimedia coinciden con los del texto analizado.			
16	Descripción general de elementos en pantalla.	La descripción por pantalla no tiene el nivel de detalle requerido para comprender cabalmente lo que se desea obtener como producto.	Guión final.
<b>Recomendaciones:</b>			
17	Requerimientos generales sin nivel adecuado de especificación.	Los requisitos (funcionales y no funcionales) no son lo suficientemente detallados y explicativos. Al ser generales no detallan claramente las necesidades del producto.	Lista de requerimientos del producto.
<b>Recomendaciones:</b> Deben ser específicos para probar más detalladamente el producto.			
18	Ausencia de	La documentación no cuenta con manual	Documentación entregada

	manual de usuario.	de usuario como ayuda al usuario a la hora de interactuar con la multimedia.	
<b>Recomendaciones:</b>			

Tabla 30 Resultados de prueba a la documentación.

### 3.4 Análisis de los resultados

La clasificación realizada a los errores permite determinar su impacto y envergadura para el sistema. Una concepción clara del estado actual del producto evita la toma de decisiones inadecuadas en el momento de planificar futuras tareas a ejecutar. La siguiente estandarización de defectos se muestra como complemento a la documentación realizada sobre los resultados generados durante las pruebas con el fin de hacerlos más comprensibles.

<b>Formato de registro de defectos – Inspecciones</b>			
<b>Objeto:</b> Multimedia Barrio Adentro		<b>Versión:</b> 1.0	
<b>Fecha:</b>		<b>Inspección N<sub>o</sub> :</b> 1.0	
<b>Tabla de tipos estándar de defectos</b>			
<b>No</b>	<b>Código</b>	<b>Nombre</b>	<b>Calificación</b>
	<b>Etapas Prueba</b>		
1	CPRF_05	Visibilidad de elementos de submenú.	Crítico
2	CPRF_07	Animación (efecto de sonido) en juego.	Advertencia
3	CPRF_09	Navegabilidad (Navegación permitida con ventana desplegada).	Severo
3.1	CPRF_10	Navegabilidad (Navegación permitida con ventana desplegada).	Severo
4	CPRF_16	Funcionalidad (Estado funcional de un botón)	Crítico
5	CPRF_07	Animación (cambio de color en letra) en juego.	Advertencia

6	CPRNF_07	Visibilidad de Tooltip.	Severo
7	CPRNF_07	Funcionalidad (animación vínculo)	Severo
7.1	CPRNF_07	Funcionalidad (animación vínculo)	Severo
8	CPRNF_07	Funcionalidad (animación vínculo)	Severo
8.1	CPRNF_07	Funcionalidad (animación vínculo)	Severo
9	CPRNF_07	Límite de zona caliente de vínculos.	Severo
9.1	CPRNF_07	Límite de zona caliente de vínculos.	Severo
10	CPRNF_07	Diseño interfaz gráfica.	Advertencia
:	:	:	:
12	CPRNF_07	Diseño interfaz gráfica.	Advertencia
13	CPRNF_05	Sintaxis en textos (Ortografía).	Cosmético
:	:	:	:
13.5	CPRNF_05	Sintaxis en textos (Ortografía).	Cosmético
13.6	CPRNF_05	Sintaxis en textos (Puntuación).	Cosmético
:	:	:	:
13.8	CPRNF_05	Sintaxis en textos (Puntuación).	Cosmético
13.9	CPRNF_05	Sintaxis en textos (paréntesis, llaves, comillas).	Cosmético
:	:	:	:
13.11	CPRNF_05	Sintaxis en textos (paréntesis, llaves, comillas).	Cosmético
13.12	CPRNF_05	Sintaxis en textos ( abreviaturas)	Cosmético
13.13	CPRNF_05	Sintaxis en textos(abreviaturas)	Cosmético
13.14	CPRNF_05	Sintaxis en textos(escritura)	Cosmético
14	CPRNF_05	Integridad de locución.	Severo
14.1	CPRNF_05	Diseño del entorno de escenarios.	Advertencia
:	:	:	:
14.6	CPRNF_05	Diseño del entorno de escenarios.	Advertencia
15	CPRNF_06	Documentación (Ortografía)	Cosmético
16	CPRNF_06	Documentación (Claridad editorial)	Advertencia
17	CPRNF_06	Documentación (Requerimientos)	Advertencia
18	CPRNF_06	Documentación (Manual de usuario).	Advertencia

Tabla 31 Formato de registro de defectos – Inspecciones

### 3.4.1 Análisis de resultados

De las pruebas diseñadas se detectaron un total de 8 pruebas fallidas de 23 aplicadas a la multimedia, razón para decir que el proceso fue efectivo, con resultados satisfactorios en cuanto al cumplimiento del objetivo de la misión: encontrar errores.

Se registraron un total de 43 errores de las pruebas aplicadas, de los cuales sólo 2 se catalogaron como críticos, al denotar una función incorrecta que causó una falla general al sistema, provocando abortar la ejecución del mismo por no restaurarse los eventos necesarios para que continuase. Esto implica que ante la inestabilidad del producto sea necesario volver a ejecutarlo para continuar la interacción con él.

En cuanto a los demás errores prevalecen los de tipo cosmético (17), relacionados con la ortografía. Estos no constituyen un punto decisivo para las funcionalidades de la multimedia, sin embargo, determinan entre otros aspectos la calidad de los contenidos, a la hora de presentar la información al usuario.

En menor medida se detectaron los errores de advertencia (15), funciones que trabajaban pero sin ajustarse a las normas establecidas y los de tipo severo (9), término con que se catalogó a las funciones que se ejecutaban pero no de la forma que fue requerido (ver Fig. 8).

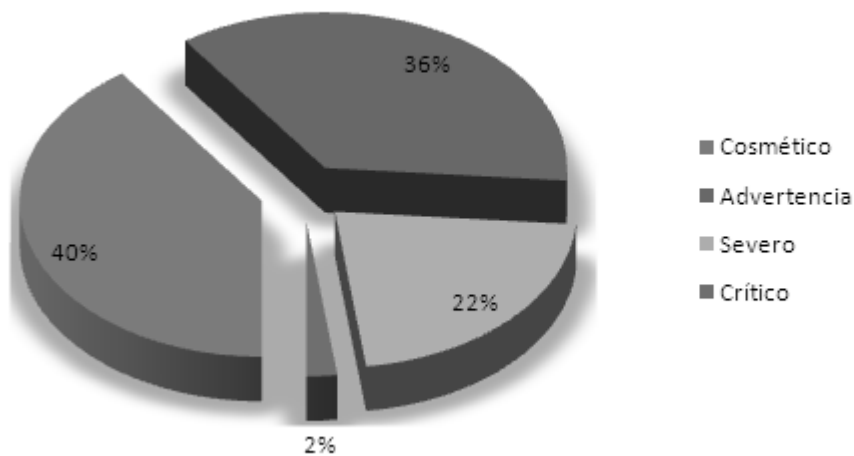


Fig.8 Porcentaje de errores según su clasificación.



Las pruebas seleccionadas y aplicadas a través de los casos de prueba diseñados, descubrieron errores en mayor o menor medida, dejando claro que el nivel de calidad del producto todavía no es el esperado y requerido. Esto no significa que el producto no desarrolle la mayoría de sus funcionalidades, según una comparación entre resultados esperados y obtenidos.

Un poco más de la mitad de los errores detectados (53.4 %) se obtuvieron mediante la prueba de solidez, que demostró el poco uso de la documentación y su inconsistencia. El resto del total se equilibró entre las restantes pruebas: interfaz gráfica, funcionalidad, documentación de manera decreciente respectivamente (ver Fig. 9).

En cuanto a las cualidades o características que la multimedia debía cumplir, se observó su desarrollo y funcionamiento óptimo.

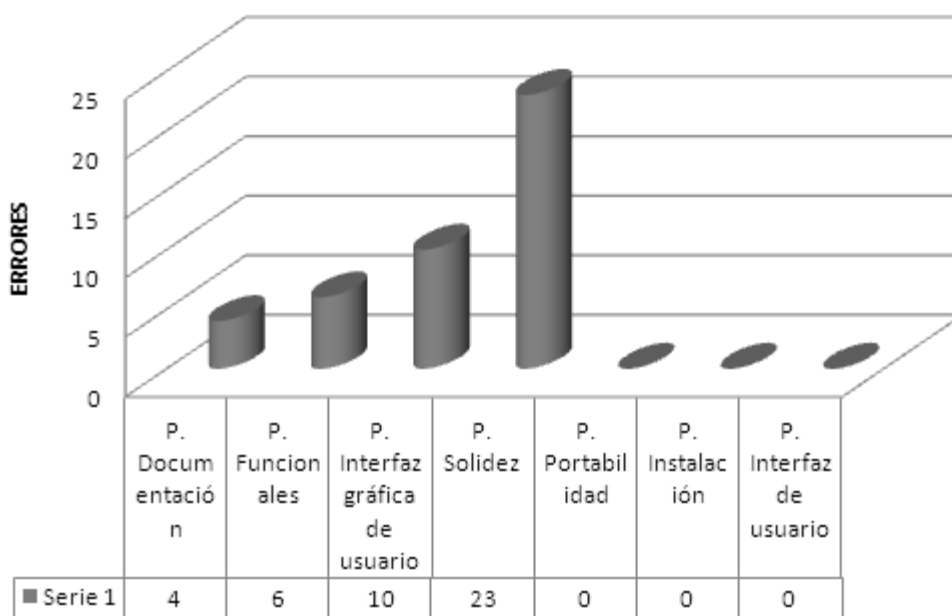


Fig.9 Número de errores según pruebas realizadas.

### 3.5 Evaluación de las pruebas

La ejecución de las pruebas estuvo centrada en detectar la mayor cantidad de errores previstos en etapas tempranas. Para ello se probó el producto no de manera completa, sino sobre los puntos más vulnerables

y propensos a generar errores según los desarrolladores. Eso no excluyó que durante el diseño de los casos de prueba no se tuvieran en cuenta otros aspectos también importantes para la certificación de la calidad del mismo como son: defectos conocidos en la práctica del desarrollo y suposiciones a partir de la naturaleza de los productos multimedia.

A partir de esto, se realizaron casos de prueba a una parte de los requisitos, tanto funcionales como no funcionales (ver Fig. 10), los que se consideraron de mayor relevancia para la ejecución del mismo. Se probó más de la séptima parte del producto, con el diseño de casos de prueba relacionados a 22 requerimiento, de los 28 funcionales y no funcionales implementados.

Cabe mencionar que en ocasiones para cada requerimiento basado en los casos de uso, aparecen varios casos de prueba, los cuales se realizaron para evaluar aspectos contenidos en el mismo pero no especificados por describir una funcionalidad específica (interfaz de vínculos, funcionalidad de scroll).

La funcionalidad del producto estuvo analizada por aproximadamente el 80% del total de requerimientos funcionales que en su mayoría tributan a las opciones de permitir y mostrar, tanto a nivel modular como de sistema. El criterio usado para la mayoría de ellos fue Clases de Equivalencia del método de Caja negra, que se definió en el plan de pruebas anteriormente.

Los requisitos no funcionales entregados fueron 10 pero se probaron solamente 7. Estos se analizaron teniendo en cuenta solamente que se cumpliera el requerimiento.

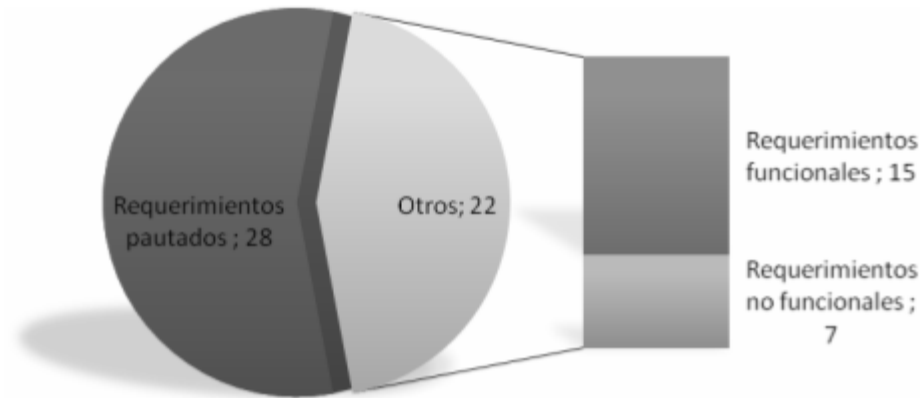


Fig. 10 Análisis de requerimientos del producto.

### 3.6 Recomendaciones específicas al producto

Durante el análisis del producto se encontraron aspectos que no constituyen errores pero que influyen en el nivel de aceptación del mismo por el usuario. Se describen a modo de recomendaciones o sugerencias con vistas al desarrollo de nuevas versiones:

- ✓ El vínculo correspondiente al escenario en que se encuentra navegando el usuario debe permanecer deshabilitado hasta tanto él no acceda a otro.
- ✓ Para la repetición de las locuciones debe existir un botón que tribute a dicha funcionalidad y a través de un nuevo clic al vínculo de dicho escenario.
- ✓ Mejorar el diseño de los botones de la barra de opciones básicas (botón de inicio y controlador de audio de fondo) o implementar tool tips que identifiquen su funcionalidad.
- ✓ En el menú inferior debe permanecer siempre visible tool tips referente a ambos módulos.
- ✓ En la pantalla de Servicios solicitados, una vez que el usuario accede a un vínculo de la tira de imágenes debe permanecer siempre visible el tool tips del mismo para que este conozca el escenario en que se encuentra.

- ✓ En el submenú de las pantallas de Requisitos mínimos para la certificación de un terreno y una edificación, el segundo elemento (Debe existir un médico cubano) no debe incluirse ahí pues no constituye vínculo a menos que se le agregue una breve descripción como a los demás elementos.
- ✓ Debe existir variedad en la interfaz gráfica de las pantallas.
- ✓ Mejorar el diseño visual de las pantallas, en aras de que sea más atractivo y ameno.
- ✓ Mejorar el diseño de los botones.
- ✓ Mostrar siempre visible tool tips correspondiente al recurso accedido.
- ✓ Mostrar diferencia entre un recurso utilizado y uno sin utilizar.

### **3.7 Costos de las pruebas**

El desarrollo de cualquier proyecto es realizable siempre que se cuente con todo el tiempo y recursos necesarios. Sin embargo, la realidad ha demostrado que la probabilidad de ocurrencia de este hecho es casi nula. La producción de software, está acarreada muchas veces, al poco tiempo de entrega del producto y a la limitación de recursos. Es necesario por tanto, el análisis de la factibilidad cuanto antes, en términos de costo.

El costo es un recurso que se sacrifica o al que se renuncia para alcanzar un objetivo específico. El costo de producción es el valor de un conjunto de bienes y esfuerzos en que se ha incurrido o se va a incurrir, que deben consumir los centros fabriles para obtener un producto terminado, que va a ser entregado (Ivnisky, 2007).

El coste de la calidad, por su parte, abarca todos los costes acarreados en la búsqueda de la calidad o en las actividades relacionadas en la obtención de la calidad. La calidad a la multimedia Barrio Adentro, fue analizada, entre otros aspectos, a través del recién finalizado proceso de pruebas. A este se le aplicó un análisis de los costos ocasionados, con el objetivo de propiciar información en cuanto a gastos y poder medir la utilidad del mismo a partir del estado de resultados generados para tomar decisiones en cuanto a su factibilidad o no.

#### **3.7.1 Estimación de costos**

Se analizaron los gastos ocasionados en términos de horas trabajadas, salarios pagados, uso de recursos materiales y consumo energético.

✓ **Gastos por concepto de consumo de tiempo:**

Se trabajó en un período de 8 meses, de lunes a viernes, a razón de 8 horas diarias. En el mes de mayo la semana laboral se extendió de lunes a sábado.

	<b>Nov.</b>	<b>Dic.</b>	<b>Enero</b>	<b>Febrero</b>	<b>Marzo</b>	<b>Abril</b>	<b>Mayo</b>	<b>Junio</b>	<b>Total</b>
<b>Días</b>	15	10	15	10	18	20	22	4	114(D)
<b>Horas</b>	8	8	8	8	8	8	8	8	8(H)
<b>Total Hm (Horas/mes)</b>	120	80	120	80	144	160	176	32	<b>912</b>
Mes = Días trabajados * Horas trabajadas (Ctd. de horas trabajadas en un mes)									
Total= Mes <sub>1</sub> + ... + Mes <sub>n</sub> (Total de horas trabajadas en el periodo analizado)									

Tabla 32 Gastos por concepto de consumo de tiempo.

✓ **Gastos por concepto de uso de recursos materiales (GR):**

Se utilizaron una serie de recursos materiales, que a continuación constan:

<b>N<sub>o</sub></b>	<b>Recurso</b>	<b>Unidades</b>	<b>Precio</b>	<b>Total</b>
1	Libreta	1	\$5.00	\$ 5.00
2	Hojas	50	\$1.00	\$ 50.00
3	Disco Compacto(CD/RW)	1	\$25.00	\$25.00
4	Lapicero	2	\$5.00	\$10.00
<b>Total</b>				<b>\$90.00</b>
Total= Unidad*Precio (Precio total por unidad analizada )				
Total = Monto <sub>1</sub> + ...+Monto <sub>n</sub> (Precio total por cuestión de recursos utilizados)				

Tabla 33 Gastos por concepto de uso de recursos materiales.

✓ **Gastos por concepto de consumo energético (GE):**

Como elemento con consumo energético, relacionado a la investigación se encuentra la computadora. La misma tiene un consumo si se trabaja en ella al 100% de 150 watts/hora. Teniendo en cuenta que sólo se empleó 1 unidad y que 1 watt equivale a \$ 0.00016, se tiene por concepto de gastos energéticos que:

	<i>Nov.</i>	<i>Dic.</i>	<i>Enero</i>	<i>Febrero</i>	<i>Marzo</i>	<i>Abril</i>	<i>Mayo</i>	<i>Junio</i>	<i>Total</i>
<b>Horas</b>	120	80	120	80	144	160	176	32	912
<b>Consumo (watt/hora * \$)</b>	0.024	0.024	0.024	0.024	0.024	0.024	0.024	0.024	0.024
<b>Total</b>	2.88	1.92	2.88	1.92	3.456	3.6	4.224	0.786	<b>\$21.89</b>
Consumo = Precio * Ctd. de watt									
GE= Horas * Consumo (Gasto energético (GE) a razón de un mes )									

Tabla 34 Gastos por concepto de consumo energético (GE).

✓ **Gasto salarial (GS):**

El equipo de pruebas trabajó con 1 miembro, que desempeñando los diferentes roles obtiene un salario mínimo de \$ 150.00 por concepto de estipendio.

GS= Ctd.P\*Ctd. M\* Salario    **Ctd.P** (Cantidad de personas)

GS = 1\*8\*150                    **Ctd.M** (Cantidad de meses)

GS= \$1200

✓ **Gastos totales:**

Costos= GR + GE+ GS

Costos= \$90.00+\$21.89+ \$1200.00

Costos= \$1311.89

Teniendo en cuenta los resultados objetivos de las pruebas, se demostró que el producto no cuenta con la calidad requerida para ser liberado y debe someterse a una nueva fase de depuración donde se de solución a las deficiencias detectadas. Esta conclusión es el producto resultante del recién proceso concluido, el cual generó en un período de 8 meses un costo total de \$1311.89 en moneda nacional (MN) equivalente a \$ 52.48 en moneda libremente convertible (CUC). Por tanto, haciendo un balance general de los gastos ocasionados por cuestiones materiales y el cumplimiento de los objetivos de las pruebas, existe un estado de satisfacción en cuanto a la eficiencia de las operaciones y reducción de gastos para la universidad durante la evaluación de productos similares, ya que fueron estudiantes los ejecutores de la

propuesta realizada, lo que evitó gastos por contratación de un personal que desarrolle la tarea. Por todo lo anterior se puede abalar el desarrollo de las pruebas al producto multimedia “Barrio Adentro al alcance de tus manos” como un proceso factible a desarrollar.

### **3.8 Conclusiones**

Las pruebas no deben terminar con su ejecución, la documentación de los resultados forma parte del proceso. Este capítulo como parte final de las actividades evaluativas desarrolladas expone cada una de sus particularidades incluyendo un breve análisis de los costos para el desarrollo de la misma.

Los errores detectados aunque en su mayoría no son críticos atentan contra la presentación del producto al cliente y el prestigio de la institución proveedora, por lo que se prevé que se libere una vez que sean solucionados. Sin embargo, la correcta y detallada recopilación de estos, unido a una serie de recomendaciones realizadas, facilitará un futuro proceso de depuración, que se expone como recomendación.

## CONCLUSIONES GENERALES

El mercado de software que se ha establecido es un mercado abierto, innovador y en continuo crecimiento, motivos suficientes para que los productos requieran de una calidad bien certificada para su inserción en él. La calidad, es un elemento que está dado por varios aspectos, pero uno de los más decisivos lo constituyen las pruebas, al permitir detectar y corregir errores. La aceptación de un producto por parte del cliente, requiere de este, por la confiabilidad y cumplimiento de expectativas que impregna al producto adquirido.

La aparición del software multimedia supone un salto cualitativamente superior respecto al software tradicional, razón para dedicarle toda la atención que merece, en aras de certificar productos con una calidad óptima para su aplicación.

La reciente investigación demostró que:

- ✓ En nuestra universidad aún las multimedias no se realizan con la precisión necesaria, bajo un esquema de trabajo adecuado, motivo por el cual, el producto final es regresado una vez que el equipo de desarrollo lo liberó.
- ✓ Las principales causas de los errores detectados viene dada por la documentación de la multimedia, enlace directo entre el cliente y los desarrolladores. Su ausencia, integridad y/o poca claridad editorial acarrea el desarrollo de un producto que no se corresponde con lo solicitado.
- ✓ Es vital durante el proceso de pruebas llevar una documentación íntegra y detallada, para obtener una simplificación de la realidad y comprenderla mejor. Debe reflejar las principales características de las pruebas (enfoque, recursos, actividades), elementos a probar (características), y resultados alcanzados para mantener informada a la organización respecto a los defectos observados relacionados.
- ✓ Se realizó una adecuada documentación de todo el proceso de prueba, con vista a la corrección de los errores encontrados y como guía de consulta a futuros procesos similares.
- ✓ Se diseñaron casos de prueba con un alto nivel de detalle, que permitieron encontrar la mayor cantidad de errores.

Como idea concluyente se puede decir que se dio cumplimiento al objetivo de la investigación, al realizar un proceso de pruebas que evaluara la calidad a la multimedia, demostrando la presencia de errores en la



misma, motivo por el cual no se recomienda la entrega del producto al cliente hasta tanto no se solucionen las deficiencias detectadas.

## RECOMENDACIONES

Como resumen de lo discutido hasta este momento y la presentación de lo realizado durante la aplicación de las pruebas, se listan a continuación una serie de recomendaciones para la modificación y mejora del proceso de pruebas a multimedia en la universidad y la facultad 8 dada la significación de la detección de errores:

- ✓ Apoyarse en la documentación del producto (guión de contenido, guión técnico) para el desarrollo de la multimedia.
- ✓ Evaluar el estado de calidad del producto desde sus primeras fases, mediante la aplicación de pruebas durante todo su ciclo de desarrollo y no una vez finalizado.
- ✓ Corregir errores detectados lo más pronto posible.
- ✓ Aplicar pruebas de regresión.
- ✓ Realizar pruebas de aceptación (pruebas alfa y beta) siempre que se pueda contactar con el cliente y/o el usuario.
- ✓ Integrar el presente documento como complemento a la bibliografía sobre calidad a productos multimedia, específicamente en la particularidad de pruebas de la facultad 8.
- ✓ Generar de forma clara y explícita la documentación que acompaña a la multimedia para el cliente final: manual de usuario.

**BIBLIOGRAFÍA REFERENCIADA**

- ARENA, L. G. *ESTRATEGIA DE CALIDAD - PARQUESOFT* Colombia: Fundación Parque Tecnológico del Software – ParqueSoft, [Consultado el: 16 de diciembre de 2006]. Disponible en: <http://www.greensqa.com/archivos/GSQA-Estrategia%20Calidad%20ParqueSoft.pdf>.
- BRUALLA, C. R. Curso de introducción a la Calidad y Mejora del Proceso Software. 2005, n° Disponible en: <http://www.calidaddelsoftware.com/modules.php?name=News&file=article&sid=97>.
- CALSADILLA, R. Realización de Pruebas a multimedia.
- CRESPO, R. M. C. L. Y. *Estudio de la Distribución Docente de Pruebas del Software y Refactoring para la Incorporación de Metodologías Ágiles* España: [Consultado el: 8 de febrero de 2007]. Disponible en: <http://www.giro.infor.uva.es/Publications/2004/MLC04/JENUI2004.pdf>.
- D'ONOFRIO, D. L. *Probando software y números de versión* España: [Consultado el: 10 de febrero de 2007]. Disponible en: <http://www.elguille.info/Clipper/probando.htm>.
- FEIJOO, M. G. D. D. *PROPUESTA DE UNA METODOLOGÍA DE DESARROLLO Y EVALUACIÓN DE SOFTWARE EDUCATIVO BAJO UN ENFOQUE DE CALIDAD SISTÉMICA*. UNIVERSIDAD SIMÓN BOLÍVAR, 2002.
- FERNÁNDEZ, F. M. S. M. P. P. E. M. B. A. G. L. A. P. T. R. C. I. M. S. *HERRAMIENTA DE EVALUACIÓN DE MULTIMEDIA DIDÁCTICO* Murcia: Departamento de Didáctica y Organización Escolar, [Consultado el: 12 de noviembre de 2006]. Disponible en: <http://www.sav.us.es/pixelbit/articulos/n18/n18art/art187.htm>.
- FERNÁNDEZ, L. *Calidad del Software* España: [Consultado el: 16 de diciembre de 2006]. Disponible en: <http://www.ati.es/gt/calidad-software/presentacion.htm>.
- GONZÁLEZ, C. Un Plan de Pruebas Exitoso. 10 de febrero 2002, n° Disponible en: [D:\Tesis\Salvas\10\\_2\América XXI - Un plan de Pruebas Exitoso.htm](D:\Tesis\Salvas\10_2\América XXI - Un plan de Pruebas Exitoso.htm).
- GRAELLS, D. P. M. *Los medios didácticos* España: Departamento de Pedagogía Aplicada, Facultad de Educación, UAB, [Consultado el: 12 de noviembre de 2006]. Disponible en: <http://dewey.uab.es/pmarques/medios.htm>.
- GUERRERO, G. M. Estrategia de pruebas a multimedia. En:
- IEEE. Standard Glossary of Software Engineering Terminology 1991.
- IVNISKY, M. *Introducción a la teoría de costos* [Consultado el: 31 de mayo de 2006]. Disponible en: <http://www.monografias.com/trabajos4/costos/costos.shtml>.

- LINARES, J. *Imagenio: La nueva forma de ocio a la carta* España: [Consultado el: 28 de noviembre de 2006]. Disponible en: <http://www.telefonica.es/sociedaddelainformacion/pdf/publicaciones/imagenio/capitulos/imagenioca p8.pdf> .
- LOVELLE, J. M. C. *Calidad del Software* España: Universidad Nacional de la Pampa, [Consultado el: 5 de diciembre de 2006]. Disponible en: [http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad\\_software.PDF](http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF).
- MAÑAS, J. A. *Prueba de Programas* España: [Consultado el: 11 de abril de 2007]. Disponible en: <http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm>.
- MENDOZA, A. C. G. M. P. L. E. *Estrategia de Pruebas para Software OO que garantiza Requerimientos No Funcionales* Venezuela: Universidad Simón Bolívar, [Consultado el: 25 de mayo de 2007]. Disponible en: [http://www.lisi.usb.ve/publicaciones/02%20calidad%20sistemica/calidad\\_09.pdf](http://www.lisi.usb.ve/publicaciones/02%20calidad%20sistemica/calidad_09.pdf).
- MOLINA, Y. F. P. R. L. C. D. C. L. C. A. M. G. J. D. A. PROCESO DE PRUEBAS DE ACEPTACIÓN PARA UN SOFTWARE DE GESTIÓN. 2005, nº
- PATÓN, D. E. F.-M. *Tema 9. Pruebas del software* España: [Consultado el: 23 de enero de 2007]. Disponible en: <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema09.pdf>.
- PONS, Y. F. *Pruebas de Aceptación del Cliente* Cuba: Universidad de Ciencias Informáticas, [Consultado el: 20 de febrero de 2007]. Disponible en: <http://www.monografias.com/trabajos36/pruebas-de-aceptacion/pruebas-de-aceptacion.shtml>.
- PRESSMAN, R. *Ingeniería de Software: Un enfoque Práctico*. Quinta ed. La Habana: Felix Varela, 2002. vol. 1,
- QUESADA, J. A. L. *Pruebas del software* España: [Consultado el: 18 de marzo de 2007]. Disponible en: [http://dis.um.es/~lopezquesada/documentos/Tema\\_6.ppt](http://dis.um.es/~lopezquesada/documentos/Tema_6.ppt)
- REINA, J. G. M. J. E. M. M. A. M. *MODELOS DE PRUEBAS PARA PRUEBAS DEL SISTEMA* España: [Consultado el: 11 de abril de 2006]. Disponible en: <http://www.lsi.us.es/~javierj/publications/MDA14.pdf>.
- SACHA. Universidad de Ciencias Informáticas (UCI), 2006.
- SEGURADO, M. T. G. D. C. *Un ejemplo de evaluación de software educativo multimedia* España: [Consultado el: 25 de noviembre de 2006]. Disponible en: [http://www.ieev.uma.es/edutec97/edu97\\_c3/2-3-03.htm](http://www.ieev.uma.es/edutec97/edu97_c3/2-3-03.htm).
- SUÁREZ, P. *Algoritmos y Programación III Pruebas* Argentina: [Consultado el: 5 de abril de 2007]. Disponible en: <http://www.fi.uba.ar/materias/7507F/Clases/PruebasGral.PDF>.

- TERUEL, A. *El Plan de Pruebas Venezuela*: [Consultado el: 10 de febrero de 2007]. Disponible en: <http://www ldc.usb.ve/~teruel/ci4713/clases2001/planPruebas.html>.
- TORRES, I. B. R. I. N. *Las pruebas del software, su aplicación al Config. Case*. Facultad de ingeniería industrial. INSTITUTO SUPERIOR POLITÉCNICO JOSÉ ANTONIO ECHEVERRÍA, 2003.
- VALDIVIA, D. R. V. E. E. G. E. *Estándares de calidad para pruebas de software* Sistemas e Informática. Universidad Nacional Mayor de San Marcos, 2005.
- VEGAS, N. J. A. M. M. S. *TÉCNICAS DE EVALUACIÓN DE SOFTWARE* España: [Consultado el: 11 de abril de 2007]. Disponible en: <http://is.ls.fi.upm.es/udis/docencia/erdsi/Documentacion-Evaluacion-6.pdf>.
- VERMA, A. *Optimización de estrategias de prueba durante el diseño* [Consultado el: 11 de abril de 2007]. Disponible en: [http://athena.teradyne.com/atd/resource/docs/spectrum/Verma\\_electrónica\\_Feb2003.pdf](http://athena.teradyne.com/atd/resource/docs/spectrum/Verma_electrónica_Feb2003.pdf).

## BIBLIOGRAFÍA CONSULTADA

1. CELENTANO, O. G. A. A Laboratory for Prototyping and Testing Multimedia Presentations 2006, nº p. 28.
2. CORTÉS, O. H. G. Diseño de Pruebas Disponible en: [http://www.willydev.net/descargas/oguzmandiseño\\_pruebas.pdf](http://www.willydev.net/descargas/oguzmandiseño_pruebas.pdf).
3. CRESPO, R. M. C. L. Y. *Estudio de la Distribución Docente de Pruebas del Software y Refactoring para la Incorporación de Metodologías Ágiles* España: [Consultado el: 9 de abril de 2007]. Disponible en: <http://bioinfo.uib.es/~joemiro/semDOC/Jen2004/ponencias/ponencia28.pdf>.
4. DÍAZ, P. *El proceso de desarrollo de sistemas Hipermedia* Madrid: [Consultado el: 29 de marzo de 2007]. Disponible en: [http://dei.inf.uc3m.es/docencia/p\\_s\\_ciclo/dsh/teoria/t3.pdf](http://dei.inf.uc3m.es/docencia/p_s_ciclo/dsh/teoria/t3.pdf).
5. DORREGO, E. *MODELO PARA LA PRODUCCIÓN Y EVALUACION FORMATIVA DE MEDIOS INSTRUCCIONALES, APLICADO AL VIDEO Y AL SOFTWARE* Venezuela: Universidad de Venezuela, [Consultado el: 3 de abril de 2007]. Disponible en: <http://lsm.dei.uc.pt/ribie/docfiles/txt200341733415MODELO%20PARA%20LA%20PRODUCCI%C3%93N.pdf>.
6. ESCUDERO, D. *Aplicaciones Multimedia* España: [Consultado el: 9 de abril de 2007]. Disponible en: <http://www.infor.uva.es/~descuder/docencia/multim/aplicMultim/aplicMultim.ppt#265,7,GUION>.
7. FIDALGO, R. L. *TÉCNICAS DE VERIFICACIÓN Y PRUEBAS* España: [Consultado el: 16 de abril de 2007]. Disponible en: <http://trevinca.ei.uvigo.es/~rlaza/teoria/Verificacion.pdf>.
8. LAGE, F. A. G. Z. C. F. J. *La revalorización del hipermedia en las agendas didácticas universitarias: criterios y consideraciones para su construcción y evaluación* Argentina: [Consultado el: 28 de marzo de 2007]. Disponible en: <http://lsm.dei.uc.pt/ribie/docfiles/txt2003729138La%20revalorizaci%C3%B3n%20del%20hipermedia.pdf>.
9. LONDOÑO, F. W. *Metodología De Desarrollo De Producciones Educativas Hipermediales Personalizantes[1]* Colombia: [Consultado el: 11 de abril de 2007]. Disponible en: [http://www.colombiaaprende.edu.co/html/mediateca/1607/articles-75593\\_archivo.pdf](http://www.colombiaaprende.edu.co/html/mediateca/1607/articles-75593_archivo.pdf).
10. MAÑAS, J. A. *Prueba de Programas* [Consultado el: 5 de abril de 2007]. Disponible en: <http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm>.
11. MÉNDEZ, P. *Evaluación de multimedios aplicados a la educación* Venezuela: Universidad Central de Venezuela facultad de humanidades y educación, [Consultado el: 25 de enero de 2007]. Disponible en: <http://www.revele.com.ve/programas/indice/ria.php?id=12151&rev=agenda>.

12. MORENO, G. A. [http://lsi.ugr.es/~arroyo/inndoc/doc/pruebas/pruebas\\_pd.php](http://lsi.ugr.es/~arroyo/inndoc/doc/pruebas/pruebas_pd.php) España: [Consultado el: 11 de abril de 2007].
13. QUESADA, J. A. L. *Ingeniería Técnica de Informática de Gestión* España: [Consultado el: 9 de abril de 2007]. Disponible en: [http://dis.um.es/~jnicolas/09BK\\_FIS.html](http://dis.um.es/~jnicolas/09BK_FIS.html).
14. RUZTORT, H. V. J. C. H. *Software de control remoto para aplicaciones del escritorio GNOME. (GBTcr)*. Sistemas de Información. Universidad del Bío-Bío, 2004.
15. SUÁREZ, P. *ALGORITMOS Y PROGRAMACION III* Argentina: [Consultado el: 5 de abril de 2007]. Disponible en: <http://www.fi.uba.ar/materias/7507F/Clases/PruebasGral.PDF>.
16. ---. *Documentación y pruebas* Argentina: [Consultado el: 5 de abril de 2007]. Disponible en: [http://www.fi.uba.ar/materias/7507F/Clases/Doc\\_y\\_pruebas.PDF](http://www.fi.uba.ar/materias/7507F/Clases/Doc_y_pruebas.PDF).
17. TORRES, J. J. G. M. J. E. A. H. T. M. M. J. *HACIA UNA PROPUESTA DE PRUEBAS TEMPRANAS DEL SISTEMA* Sevilla: Departamento de Lenguajes y Sistemas Informáticos, Escuela Técnica Superior de Ingeniería Informática, [Consultado el: 16 de abril de 2007]. Disponible en: <http://in2test.lsi.uniovi.es/pris2006/PRIS2006-GutierrezEscalonaMejiasTorres.pdf>.
18. VALENCIA, G. A. *Subcontratación del servicio de pruebas de software* Colombia: [Consultado el: 2 de abril de 2007]. Disponible en: <http://www.greensqa.com/archivos/Servicio%20Pruebas%20de%20Software.pdf>.
19. VELTHUIS, M. P. U. M. P. *AUTOMATIZACIÓN DEL PROCESO DE PRUEBAS UNITARIAS* España: [Consultado el: 9 de abril de 2007]. Disponible en: <http://in2test.lsi.uniovi.es/pris2006/PRIS2006-PoloPiattini.pdf>.