

**Universidad de las Ciencias Informáticas**

**FACULTAD 6**



**Título:** Diseño y aplicación de pruebas a los módulos PRIMICIA Y WebTV del Sistema de Gestión y Transmisión de Contenido Audiovisual.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

**Autor:** Yandys Menéndez Delgado.

**Tutor:** Ing. Eriellis Reyes González.

*La Habana, junio 2012*

*“Año 54 de la Revolución”*

### DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor del trabajo “**Diseño y aplicación de pruebas al Sistema de Gestión y transmisión de contenido audiovisual (SIAV). Módulos PRIMICIA y WebTV.**” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

\_\_\_\_\_  
**Autor:** Yandys Menéndez Delgado.

\_\_\_\_\_  
**Tutor:** Ing. Erielis Reyes González.

### DATOS DE CONTACTO

**Tutor:** Ing. Eriellis Reyes González.

**Formación académica:**

- Ingeniero en Ciencias Informáticas, Universidad de Ciencias Informáticas, 2010.

**Centro Laboral:**

- Centro de Desarrollo de Geoinformática y Señales Digitales. Facultad 6.

**Correo electrónico:** ereyesg@uci.cu

# Agradecimientos

---

*Agradezco...*

*A mi abuela Zoíla...por ser mi todo, por ser madre, abuela al mismo tiempo, por todo su amor y cariño...Gracias Mamazoíla por estar en cada momento para mí.*

*A mi mamá...por darme la vida...por ser madre...por todo su amor...Gracias mamá.*

*A mi tía Reina... por ser mi otra mamá, por ser mi ejemplo a seguir, por su sacrificio y entrega incondicional, por todo el cariño que me ha dado...Gracias por existir.*

*A mi tía Mabel...por apoyarme en los momentos difíciles, por sus buenos consejos, por transmitirme la confianza y seguridad que necesito para enfrentarme a los retos que me impone la vida.*

*A Cristina...por ser la hermana que nunca tuve, la gran amiga y confidente, por todos los buenos momentos que pasamos juntos en la universidad...a ti un beso grande.*

*A todas las personas maravillosas que he conocido durante estos 5 años, sin ellos no hubiera sido posible disfrutar al máximo las fiestas a las que fuimos.*

*A mi tutora Eriélis por su dedicación y ayuda.*

*A la UCI por darme la oportunidad de conocer y experimentar cosas nuevas e inolvidables...Muchas Gracias.*

*Dedicado...*

*A mi abuela Zoíla...por haberme dado todo el amor del mundo, por su dedicación y entrega...*

*A mis tías Reina y Mabel...por apoyarme y confiar siempre en mí, e impulsarme siempre a llegar más lejos...*

*A mi mamá Clarita... por su cariño incondicional...*

*A los que ya no están.*

### **RESUMEN**

El presente trabajo, titulado “Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del Sistema de Gestión y Transmisión de Contenido Audiovisual” se realizó con el objetivo fundamental de garantizar la calidad de este producto. Para dar cumplimiento a este objetivo se llevó a cabo todo un proceso de pruebas, con el fin de detectar la mayor cantidad de defectos; y una vez identificados estos, corregirlos en el menor tiempo posible. Para ello fue necesario realizar una fundamentación teórica en la que se abordan los diferentes conceptos emitidos por varios autores relacionados con la calidad de software, las pruebas de software y el aseguramiento de la calidad en el proceso de desarrollo. Se examinó todo lo referente al flujo de pruebas que define RUP como metodología de desarrollo de software. Se analizaron los tipos de pruebas y los niveles a los cuales se aplican los mismos. Luego se seleccionaron las pruebas que se le aplicarían: pruebas de unidad específicamente la prueba de la ruta básica como técnica de caja blanca, pruebas de sistema concretamente la prueba de partición equivalente como técnica de caja negra y pruebas de rendimiento particularmente de carga y estrés. Posteriormente se elaboraron los diseños de casos de prueba para cada prueba seleccionada. Por último se le aplicaron las pruebas a los módulos con el objetivo de verificar y validar que los requerimientos planteados fueron cumplidos; los resultados finales se documentaron y se emitió una evaluación final sobre el estado de los módulos.

### **PALABRAS CLAVES**

Calidad de software, Caso de prueba, Estrategia de prueba, Plan de prueba, Prueba de software.

## TABLAS

Tabla 1: Factores de calidad de McCall. (10).....	8
Tabla 2: Métricas para pruebas. (11).....	9
Tabla 3: Escenarios de pruebas.....	26
Tabla 4: Casos de Usos de WebTV. ....	27
Tabla 5: Casos de Usos de PRIMICIA. ....	28
Tabla 6: CU-Gestionar tipología de archivo multimedia.....	41
Tabla 7: CU-Buscar archivo multimedia.....	41
Tabla 8: CU-Crear publicación de archivo multimedia. ....	42
Tabla 9: CU-Gestionar señal en vivo.....	43
Tabla 10: CU-Publicar Noticia. ....	44
Tabla 11: CU-Gestionar Sección. ....	46
Tabla 12: Gestionar Infocintas.....	48
Tabla 13: CU-Transmitir Canal.....	49
Tabla 14: SC 1 Adicionar Tipología. ....	52
Tabla 15: SC 2 Gestionar Campos.....	54
Tabla 16: SC 3 Gestionar Relaciones.....	56
Tabla 17: SC 4 Editar Datos de Tipología.....	57
Tabla 18: SC 5 Crear Tipología.....	58
Tabla 19: SC 6 Eliminar Tipología.....	58
Tabla 20: SC 1 Buscar Archivo Multimedia.....	59
Tabla 21: SC 1 Publicar Noticia.....	59
Tabla 22: SC 2 Mostrar Noticia.....	59
Tabla 23: SC 3 Archivar Noticia.....	60
Tabla 24: SC 4 Eliminar Noticia.....	60
Tabla 25: SC 1 Transmitir Canal.....	60
Tabla 26: SC 2 Visualizar Canal.....	61
Tabla 27: SC 3 Transmitir Fuera de Servicio.....	61
Tabla 28: DCP Carga y Estrés: WebTV.....	64
Tabla 29: DCP Carga y Estrés: PRIMICIA.....	65

## ILUSTRACIONES

Ilustración 2: Los trabajadores y artefactos involucrados en las pruebas. (13).....	11
Ilustración 3: Prueba de Caja Blanca.....	19
Ilustración 4: Prueba de Caja Negra.....	21
Ilustración 5: Notación de gráfica de flujo. (41).....	29
Ilustración 6: Grafo de flujo asociado al código fuente WebTV.....	35
Ilustración 7: Grafo de flujo asociado al código fuente PRIMICIA.....	36

**ÍNDICE GENERAL**

INTRODUCCIÓN..... 1

CAPÍTULO 1: “Fundamentación Teórica” ..... 5

1.1 Introducción ..... 5

1.2 Calidad de software. Conceptos y definiciones ..... 5

1.3 Factores que miden la calidad del software. Métricas ..... 7

1.4 Metodología de desarrollo de software..... 9

    1.4.1 Proceso Unificado de Desarrollo (RUP)..... 10

    1.4.2 Proceso de Pruebas de RUP ..... 11

    1.4.3 Artefactos del flujo de trabajo de prueba ..... 12

        1.4.3.1 Documentos del proceso de pruebas utilizados en la UCI ..... 12

1.5 Estrategia de prueba de software..... 13

    1.5.1 Aspectos estratégicos ..... 14

    1.5.2 Niveles de prueba..... 15

        1.5.2.1 Prueba de unidad ..... 15

        1.5.2.2 Prueba de integración..... 15

        1.5.2.3 Prueba de validación..... 16

        1.5.2.4 Prueba del sistema..... 17

1.6 Técnica de diseño de prueba ..... 18

    1.6.1 Técnica de prueba Caja blanca..... 19

    1.6.2 Técnica de prueba Caja negra ..... 21

1.7 Conclusiones parciales ..... 22

CAPÍTULO 2: “Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV”  
..... 23

2.1 Introducción ..... 23

2.2 Sistema de Gestión y Transmisión de Contenido Audiovisual (SIAV). ..... 23

2.3 Plan de Prueba ..... 24

2.4 Pasos a seguir en las pruebas o Estrategia ..... 25

2.5 Escenarios de pruebas ..... 26

2.6 Herramienta utilizada ..... 27

2.7 Casos de usos seleccionados para el proceso de pruebas ..... 27

2.8 Tipos de pruebas a realizar..... 28



2.9 Diseño de los Casos de Prueba de Caja Blanca .....	34
2.10 Diseño de los Casos de Prueba de Caja Negra. ....	37
2.11 Conclusiones parciales .....	49
CAPÍTULO 3: “Resultados de las pruebas” .....	50
3.1 Introducción .....	50
3.2 Resultados obtenidos de aplicar pruebas de Caja Blanca.....	50
3.2.1 WebTV-Caso de Prueba: Buscar Archivo Multimedia. ....	50
3.2.2 PRIMICIA-Caso de Prueba: Gestionar Sección. ....	50
3.3 Resultados obtenidos de aplicar pruebas de Caja Negra .....	51
3.3.1 Módulo WebTV- Subsistema Transmisión. ....	51
3.3.1.1 Caso de Prueba: Gestionar tipología de archivo multimedia. ....	51
3.3.2 Módulo WebTV- Subsistema Gestión y Presentación de Contenido .....	58
3.3.2.1 Caso de Prueba: Buscar archivo multimedia. ....	58
3.3.2.2 Caso de Prueba: Crear publicación de archivo multimedia. <i>Ver Anexo 5</i> .....	59
3.3.2.3 Caso de Prueba: Gestionar señal en vivo. <i>Ver Anexo 6</i> .....	59
3.3.3 Módulo PRIMICIA- Subsistema Administración.....	59
3.3.3.1 Caso de Prueba: Publicar Noticia. ....	59
3.3.3.2 Caso de Prueba: Gestionar Sección. <i>Ver Anexo 7</i> .....	60
3.3.3.3 Caso de Prueba: Gestionar Infocintas. <i>Ver Anexo 8</i> . ....	60
3.3.4 Módulo PRIMICIA- Subsistema Transmisión .....	60
3.3.4.1 Caso de Prueba: Transmitir Canal. ....	60
3.4 Resultados obtenidos de aplicar pruebas de Carga y Estrés .....	63
3.5 Conclusiones parciales .....	66
CONCLUSIONES .....	67
RECOMENDACIONES .....	68
BIBLIOGRAFÍA REFERENCIADA .....	69

## INTRODUCCIÓN

Los medios de comunicación centrados en la televisión (TV) han devenido de disímiles avances tecnológicos que el hombre en un breve período de tiempo ha sido capaz de instaurar, usar y renovar con fines propios. Este creciente desarrollo experimentado por la humanidad ha provocado que se produzcan cambios en las tecnologías analógicas de transmisión televisiva, introduciendo de forma paulatina mejoras y servicios adicionales en el proceso comunicativo. Estos avances permitieron la transición a la tecnología digital y con ello la creación de sistemas de software capaces de procesar y transmitir contenidos audiovisuales.

Actualmente existen diversas soluciones de software a nivel mundial que se caracterizan por brindar una automatización sin fallos de señales que emiten eficientemente contenido audiovisual, entre las que se destacan los sistemas YouTube (video en la web), VSNMatic (TV), BigBlueButton (video conferencia). En estos sistemas es imprescindible que la realización de los mismos sea de buena calidad y con un elevado nivel de responsabilidad e importancia, por lo que se hace necesaria la presencia de la calidad de software, que no es más que la *“concordancia con los requisitos funcionales y de desempeño explícitamente establecidos, estándares de desarrollo explícitamente documentados y características implícitas que se esperan de cualquier software desarrollado profesionalmente”*. (1)

En vistas de las exigencias de las industrias de software que existen actualmente en el mundo y atendiendo a la necesidad real de desarrollar soluciones informáticas con un alto compromiso, el estado cubano creó la Universidad de Ciencias Informáticas (UCI) constituyendo así el elemento fundamental para el desarrollo y puesta en práctica de sistemas audiovisuales. Para dar cumplimiento a este objetivo la UCI cuenta dentro de la Facultad 6 con el departamento de Señales Digitales del centro de desarrollo de Geoinformática y Señales Digitales (GEYSED), el cual ha desarrollado varios tipos de sistemas como son: PRIMICIA, el Sistema de Transmisión de Canales Virtuales y WebTV. Los mismos presentan funcionalidades comunes de un sistema a otro, pero también particularidades propias que los diferencian.

PRIMICIA es una solución integral capaz de proveer un canal de televisión para la transmisión automática y constante de información. El Sistema de Transmisión de Canales Virtuales constituye el medio que posibilita la distribución de contenidos audiovisuales haciendo uso de sistemas de

transmisión de señales. La plataforma WebTV permite la visualización de TV online utilizando servidores de streaming<sup>1</sup> para satisfacer las necesidades de video bajo demanda de los usuarios.

El departamento Señales Digitales se ha propuesto la creación de un Sistema de Gestión y Trasmisión de Contenido Audiovisual (SIAV), permitiendo que los resultados y experiencias obtenidas en estos proyectos anteriormente mencionados puedan ser utilizados en una finalidad común, facilitando así un producto de mejor calidad. Teniendo en cuenta lo antes planteado se hace imprescindible que en todo el proceso de desarrollo del software se tomen las medidas pertinentes para que el producto deseado tenga la calidad requerida.

Por tal motivo, en el proyecto SIAV existe la necesidad primordial de verificar y validar que el mismo tenga una correcta implementación, minimizando así las posibles consecuencias negativas que se pudieran presentar durante el proceso de desarrollo de software, para ello se concibe desarrollar un proceso de pruebas de software que permita mitigar las situaciones de discrepancias entre lo que debía ser y lo que se realizó realmente antes de entregar la versión final del producto a los clientes.

Luego del análisis se plantea como **problema a resolver**: ¿Cómo reconocer la mayor cantidad de errores posibles en los módulos PRIMICIA y WebTV del Sistema de Gestión y Trasmisión de Contenido Audiovisual que permita la verificación y validación de los requisitos planteados por el cliente?

Para el desarrollo de la investigación y a partir de lo antes expuesto se define como **objeto de estudio**: el proceso de pruebas como parte del desarrollo del software y como **campo de acción**: proceso de pruebas de calidad aplicado a los módulos PRIMICIA y WebTV del Sistema de Gestión y Trasmisión de Contenidos Audiovisual.

Se propone como **objetivo general**: Diseñar y aplicar pruebas de software a los módulos PRIMICIA y WebTV del Sistema de Gestión y Trasmisión de Contenido Audiovisual que permita certificar la calidad del producto. Como guía de observación se plantea la siguiente **idea a defender**: si se realiza adecuadamente el diseño de casos de pruebas y se aplican de forma efectiva a los módulos PRIMICIA

---

<sup>1</sup> *Consiste en una tecnología utilizada para permitir la visualización y la audición de un archivo mientras se está descargando, a través de la construcción de un buffer por parte del cliente, una vez que este se ha conectado al servidor el buffer del cliente se va llenando de la información descargada y se va reproduciendo en el ordenador. El sistema se encuentra sincronizado, tal que, una vez terminada la reproducción del contenido del archivo, finaliza la descarga siempre y cuando no existan interrupciones en el envío del archivo. (Tomado de [www.mastermagazine.info](http://www.mastermagazine.info), consultado en octubre de 2011).*

y WebTV del Sistema de Gestión y Transmisión de Contenido Audiovisual, se logrará un producto con la calidad requerida que cumpla con los requisitos del cliente.

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

1. Caracterización de los distintos tipos de pruebas para el diseño teórico y metodológico de la investigación.
2. Selección de las pruebas que se le van a realizar al sistema durante todo el proceso de desarrollo.
3. Documentación de los procedimientos para cada una de las pruebas seleccionadas.
4. Confección de un cronograma con todas las pruebas que se van a aplicar.
5. Diseño de los casos de prueba para los casos de usos críticos del sistema.
6. Aplicación de las pruebas a los módulos.
7. Caracterización del estado de los módulos una vez finalizado el proceso de pruebas.

Para obtener los conocimientos necesarios que hagan posible el cumplimiento del objetivo trazado, en la investigación se pondrán en práctica diferentes **métodos científicos** existentes, tanto teóricos como empíricos:

## **Métodos Teóricos**

**Método histórico-lógico:** se utilizó el mismo para investigar la información que se tiene hasta el momento del estado del arte, sobre las estrategias de pruebas utilizadas en proyectos informáticos; y los principales conceptos y herramientas relacionadas con la calidad de software.

**Método analítico sintético:** se aplicó con el objetivo de realizar un estudio detallado de los casos de usos del producto Sistema de Gestión y Transmisión de Contenido Audiovisual con el fin de seleccionar a los que se le van a realizar las pruebas.

**Método de modelación:** se usó para diseñar casos de pruebas a cada uno de los casos de usos del producto SIAV. Pues se convierte en una necesidad que se ejecute esta modelación a la hora de realizar el diseño a cada uno de los que se escogieron para aplicar las pruebas correspondientes.

## **Métodos Empíricos**

**Método de Observación:** se utilizó en toda la investigación, pues se recoge información de los aspectos tratados en la tesis desde el punto de vista de los autores así como sus definiciones y resultados para permitir realizar un análisis del arte en este campo.

**Método de Entrevista:** se utilizó con el objetivo fundamental de seleccionar aquellos casos de usos que iban a ser sometidos a prueba.

El contenido de este trabajo se encuentra estructurado en tres capítulos, los que se definen de la siguiente manera:

**Capítulo 1:** En este capítulo se realizará un análisis de todos los aspectos fundamentales que están relacionados con las pruebas de software. Se hará referencia a la metodología de desarrollo por la cual se guiará todo el proceso durante la fase de prueba. Además se tratarán los principales aspectos relacionados con los niveles de prueba, así como las técnicas de pruebas más utilizadas. En este capítulo se demostrará la importancia que tiene el proceso de pruebas durante el proceso de desarrollo de un producto informático.

**Capítulo 2:** En este capítulo se confeccionará el plan de pruebas donde quedará plasmada toda la estrategia a seguir, para que las pruebas sean llevadas a cabo eficientemente. Se realizará todo el diseño de casos de pruebas a los casos de usos críticos seleccionados para ser objetos de prueba del producto SIAV. Además se hará referencia a los procedimientos utilizados para su desarrollo, que servirán de base para el diseño y la ejecución de los casos de pruebas.

**Capítulo 3:** En este capítulo se documentarán los resultados de las pruebas aplicadas en el capítulo anterior. Además se realizará un análisis de los resultados que se obtendrán durante todo el proceso de pruebas que se llevará a cabo sobre el producto SIAV. Se redactará un resumen donde quedarán plasmadas las principales deficiencias encontradas durante todo el proceso y las principales recomendaciones a tener en cuenta para posteriores trabajos.

## CAPÍTULO 1: “Fundamentación Teórica”

### 1.1 Introducción

Durante el proceso de desarrollo de software la etapa de prueba constituye una de las disciplinas más importantes del ciclo de vida, pues con esta se logra depurar un software desde sus inicios, llegando este a manos de los clientes lo más libre posible de defectos y cumpliendo con todas las funcionalidades requeridas. El proceso de prueba cuenta con actividades bien definidas, comienza con la planificación de las pruebas, luego la ejecución, el control y por último la evaluación de las mismas. Todas estas actividades son llevadas a cabo por un proceso de retroalimentación, lo que permite que los errores encontrados se resuelvan de manera efectiva en iteraciones posteriores de forma tal que no afecten el tiempo de duración del proyecto y el costo del mismo.

En este capítulo se especifican los principales aspectos que conforman la fundamentación teórica de la investigación, donde se definen los conceptos relacionados con el dominio del problema y el objeto de estudio, haciendo énfasis en el proceso de pruebas fundamentalmente en algunos aspectos relacionados con la calidad de software, tales como: los factores que miden la calidad de estos, así como las métricas asociadas; estrategias de prueba de software; técnicas de diseño de prueba; destacando algunos criterios generales, de los cuales se presentará su definición. Además se hará referencia a la fase de prueba que propone el Proceso Unificado de Desarrollo (*RUP*<sup>2</sup>) como metodología de desarrollo para este proceso.

### 1.2 Calidad de software. Conceptos y definiciones

Cuando se usa el término de calidad de software, muchas personas desconocen el verdadero significado que este encierra. Pudiéndose decir que es usado frecuentemente por muchas personas sin saber cuál es la magnitud de esta palabra. Para definir esta, primeramente se expone un concepto de que se entiende por calidad, de forma general.

**Calidad:** El American Heritage Dictionary define calidad como “una característica o atributo de algo”. Como un atributo de algo de un elemento, la calidad se refiere a características mensurables, es decir:

---

<sup>2</sup> (*RUP*): Por sus siglas en inglés *Rational Unified Process*, algunos autores se refieren como el *Proceso Unificado de Desarrollo* y otros al *Proceso Unificado de Rational*.

cosas que se pueden comparar para conocer estándares. Sin embargo, el software, principalmente una entidad intelectual, es más difícil de caracterizar que los objetos físicos. (2)

La Real Academia Española define la calidad en su primera acepción como la “propiedad o conjunto de propiedades inherentes a una cosa, que permiten apreciarla como igual, mejor o peor que las restantes de su especie”, y en la segunda, “en sentido absoluto calidad, superioridad o excelencia”. (3)

**Calidad de software:** se define como “Concordancia con los requisitos funcionales y de desempeño explícitamente establecidos, estándares de desarrollo explícitamente documentados y características implícitas que se esperan de cualquier software desarrollado profesionalmente”. (1)

“El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas”. (4)

“Grado con el cual el cliente o usuario percibe que el software satisface sus expectativas”. (5)

En cierta medida todos los conceptos expuestos hacen referencia a lo que es la calidad de software por lo que ¿Cómo se define la calidad? en el sentido más amplio, se puede decir que el término calidad siempre será entendido de diferentes maneras por cada persona, ya que para unos la calidad residirá en un producto y para otros en el servicio posventa del mismo. Lo cierto es que se puede afirmar que es un aspecto fundamental con máxima prioridad para un equipo de desarrollo a la hora de construir un producto de software, por consiguiente la calidad de software reunirá las características y cualidades principales que necesita un producto para hacer válida su utilidad. Después de haber analizado el concepto de calidad, centrándose fundamentalmente en el de calidad de software como tal, es necesario analizar otros conceptos muy importantes como es el caso de:

**Prueba de Software:** En inglés testing o beta testing, las pruebas de 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 aplicación. (6) El objetivo de la prueba del software es descubrir errores; se cumple planeando y ejecutando una serie de pasos (pruebas de unidad, integración, validación y sistema). (7)

Otro aspecto que hay que tener en cuenta cuando se va a analizar la calidad de un producto son las tareas que se van a llevar a cabo para garantizar esta, comúnmente llamado **aseguramiento de la calidad (SQA)**, aunque otros autores lo denominan **garantía de la calidad**.

**SQA:** es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto (software) satisfará los requisitos dados de calidad. (4)

El *estándar ISO 9000*<sup>3</sup> describe un sistema de garantía de calidad en términos genéricos que se aplican a cualquier negocio sin importar los productos o servicios ofrecidos.

La *norma NMX-CC-001:199*<sup>4</sup> define al aseguramiento de la calidad como el "conjunto de actividades planeadas y sistemáticas implantadas dentro del sistema de calidad, y demostradas según se requiera para proporcionar confianza adecuada de que un elemento cumplirá los requisitos para la calidad". (8)

La **Garantía de la calidad** consiste en un conjunto de funciones de auditoría e información que evalúan la efectividad y qué tan completas son las actividades de control de calidad. La meta del aseguramiento de la calidad es brindarle al gestor los datos necesarios para que esté informado acerca de la calidad del producto, y por consiguiente que comprenda y confíe en que la calidad del producto está satisfaciendo sus metas. Desde luego, si los datos que ofrece el aseguramiento de la calidad identifican problemas, es responsabilidad del gestor abordarlos y aplicar los recursos necesarios para resolver los conflictos de calidad. (9)

No hay duda de que estas definiciones pueden modificarse o extenderse. De hecho, la definición de calidad de software podría debatirse interminablemente. Por ello cuando se hable de calidad de software es necesario comprender que alcanzar la calidad perfecta en el desarrollo de un producto es un gran objetivo, una meta y un logro importante dentro de un proyecto; la calidad de software de por sí implica un conjunto de pruebas, procedimientos que permiten medir la efectividad, rendimiento, seguridad y la satisfacción de los usuarios que interactúen con el producto final. Luego de haber abordado los conceptos principales es necesario tener en cuenta una serie de factores de calidad de software para poder justificar la calidad del producto.

### 1.3 Factores que miden la calidad del software. Métricas

Un aspecto importante son los factores por los cuales se puede medir la calidad de un producto, estos se pueden dividir en tres aspectos:

#### FACTORES DE CALIDAD DE McCall

---

<sup>3</sup>ISO 9000: Organización Internacional de Normalización (Internacional Organization for Standarding) Organismo internacional más conocido en materia de normatividad (ISO).

<sup>4</sup>NMX-CC-001:1995 (ISO): Norma Mexicana de la ISO 9000.



Aspecto que trata	Factor Calidad
<b>Operación del Producto</b>	<ul style="list-style-type: none"> <li>▪ <b>Corrección:</b> El grado en que el programa cumple con su especificación y satisface los objetivos que propuso el cliente.</li> <li>▪ <b>Confiabilidad:</b> El grado en que se esperaría que un programa desempeñe su función con la precisión requerida.</li> <li>▪ <b>Eficiencia:</b> La cantidad de código y de recursos de cómputo necesarios para que un programa realice su función.</li> <li>▪ <b>Integridad:</b> El grado de control sobre acceso al software o los datos por parte de las personas no autorizadas.</li> <li>▪ <b>Facilidad de uso:</b> El esfuerzo necesario para aprender, operar y preparar los datos de entrada de un programa e interpretar la salida.</li> </ul>
<b>Revisión del producto</b>	<ul style="list-style-type: none"> <li>▪ <b>Facilidad de Mantenimiento:</b> El esfuerzo necesario para localizar y corregir un error en un programa.</li> <li>▪ <b>Flexibilidad:</b> El esfuerzo necesario para modificar un programa en operación.</li> <li>▪ <b>Facilidad de prueba:</b> El esfuerzo que demanda probar un programa con el fin de asegurar que se realiza su función.</li> </ul>
<b>Transición del producto</b>	<ul style="list-style-type: none"> <li>▪ <b>Portabilidad:</b> El esfuerzo necesario para transferir el programa de un entorno de hardware o software a otro.</li> <li>▪ <b>Facilidad de Reutilización:</b> El grado en que un programa (o partes de él) puede reutilizarse en otras aplicaciones (en relación con el empaquetamiento y el alcance de las funciones que realiza el programa).</li> <li>▪ <b>Interoperabilidad:</b> El esfuerzo necesario para acoplar un sistema con otro.</li> </ul>

Tabla 1: Factores de calidad de McCall. (10)

**Métricas para pruebas.** Estas métricas ayudan a diseñar casos de pruebas efectivos y a evaluar la eficacia de las pruebas:

MÉTRICAS PARA PRUEBAS	
<b>Métricas de cobertura de instrucciones y ramas</b>	Lleva al diseño de casos de prueba que proporcionan cobertura del programa.
<b>Métricas relacionadas con los defectos</b>	Se concentran en encontrar defectos y no en las propias pruebas.
<b>Efectividad de la prueba</b>	Proporcionan un indicio en el tiempo real de la efectividad de las pruebas aplicadas.

<b>Métricas en el proceso</b>	Métricas relacionadas con el proceso que se determina a medida que se aplican las pruebas.
-------------------------------	--

**Tabla 2: Métricas para pruebas. (11)**

Los ingenieros de software usan las métricas del producto como apoyo para construir software de mayor calidad.

Las métricas del software proporcionan una manera cuantitativa de evaluar la calidad de los atributos internos de un producto, lo que permite que un ingeniero de software evalúe la calidad antes de construirlo. Las métricas proporcionan los conocimientos necesarios para crear modelos efectivos de análisis y diseño, un código sólido y pruebas exhaustivas. Para que resulte útil en la realidad, una métrica del software debe ser simple y calculable, persuasiva, consistente y objetiva. Debe ser independiente del lenguaje de programación y proporcionar retroalimentación efectiva al ingeniero del software. (12)

Aunque las métricas del producto para el software de computadora no suelen ser absolutas, proporcionan una manera sistemática de evaluar la calidad a partir de un conjunto de reglas definidas con claridad. También proporcionan al ingeniero de software información inmediata y en la aplicación; no posterior al hecho. Esto permite al ingeniero descubrir y corregir problemas potenciales antes de que se conviertan en defectos catastróficos. Luego de haber abordado los factores que miden la calidad del software es necesario ver cuál metodología de desarrollo se utilizará durante todo el proceso de desarrollo del software. A continuación se detallan los elementos principales de la metodología seleccionada.

## **1.4 Metodología de desarrollo de software**

Las metodologías de desarrollo de software definen un conjunto de criterios que guían la forma en que se aplica la ingeniería del software en un proyecto productivo; en otras palabras, es la base para la construcción de un proyecto de software, la etapa fundamental para lograr los objetivos buscados con dicho proyecto. No es que haya una metodología superior a las demás, todas las metodologías son, en esencia, buenas. Obviamente, las más modernas responden a problemas y necesidades actuales. La ausencia de metodología en el desarrollo de un proyecto de software garantiza con seguridad también la poca o nula calidad del mismo.

Con el surgimiento de nuevos modelos de producción, han ido apareciendo nuevas metodologías de proceso de desarrollo, estos procesos tienen características marcadas, lo que ha conllevado a

agruparlos en dos grupos, los llamados "métodos robustos" y los "métodos ágiles". La diferencia más notable entre estos es que mientras los métodos robustos intentan obtener los resultados ayudándose principalmente de la documentación ordenada, los métodos ágiles tienen como base de sus resultados a la comunicación e interacción directa con todos los usuarios involucrados en el proceso.

Los arquitectos de software decidieron que la metodología de desarrollo que se utilizará durante todo el proceso de desarrollo de construcción del software es la metodología de RUP. Ahora bien, ¿Por qué aplicar RUP en el proyecto SIAV y no otra metodología?:

Se selecciona RUP porque sus características son factibles para la realización del proyecto, debido a que proporciona una documentación actualizada en correspondencia con el desarrollo del sistema permitiendo al mismo tiempo que si algún miembro del equipo no puede seguir con el trabajo otro que ocupe su lugar, tenga por donde guiarse y conozca qué fue lo que se hizo. Ahora se hace viable analizar qué es lo que plantea esta metodología para su aplicación y sus principales elementos.

### 1.4.1 Proceso Unificado de Desarrollo (RUP)

En primer lugar, el Proceso Unificado es un proceso de desarrollo de software. Un *proceso de desarrollo de software* es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto. (13)

El *Proceso Unificado de Rational* es un proceso de ingeniería del software. Proporciona un acercamiento disciplinado a la asignación de tareas y responsabilidades en una organización de desarrollo. Su propósito es asegurar la producción de software de alta calidad que se ajuste a las necesidades de sus usuarios finales con unos costos y calendario predecibles. (14)

RUP es uno de los procesos más generales de los existentes actualmente, ya que en la realidad está pensado para adaptarse a cualquier proyecto, y no tan solo de software. RUP se basa en casos de uso para describir lo que se espera del software y está muy orientado a la arquitectura del sistema, documentándose lo mejor posible, basándose en *UML*<sup>5</sup> como lenguaje de modelado. (15)

---

<sup>5</sup> (UML): Por sus siglas en inglés *Unified Modeling Language*, es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. UML capta la información sobre la estructura estática y

El diagrama de los flujos de trabajo que define RUP para cada una de las fases se encuentra en: **Ver Anexos.**

RUP como metodología de desarrollo al transitar por cada fase define una serie de disciplinas dentro de las cuales se encuentra la de prueba, esta disciplina en particular tiene el mayor peso en la fase de construcción del software garantizando así un producto con calidad, de ahí que se hace necesario analizar cómo funciona este proceso de prueba.

## 1.4.2 Proceso de Pruebas de RUP

Las pruebas se llevan a cabo sobre todo cuando una construcción (como resultado de implementación) es sometida a *pruebas de integración y de sistema*. Más concretamente, los objetivos de la prueba son: (13)

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



**Ilustración 1: Los trabajadores y artefactos involucrados en las pruebas. (13)**

el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo.

El resultado principal de la prueba es el modelo de prueba, el cual describe cómo ha sido probado el sistema. El modelo de prueba incluye: (13)

- Casos de prueba, que especifican qué probar en el sistema.
- Procedimientos de prueba, que especifican cómo realizar los casos de prueba.
- Componentes de prueba, que automatizan los procedimientos de prueba.

La prueba da también como resultado un plan de prueba, evaluaciones de las pruebas realizadas y los defectos que pueden ser pasados como entrada a flujos de trabajo anteriores, como el diseño y la implementación. (13)

### 1.4.3 Artefactos del flujo de trabajo de prueba

El Proceso Unificado de Desarrollo de software plantea siete artefactos para el flujo de trabajo de prueba los cuales deben ser generados dentro de esta fase. Los cuales se muestran a continuación: (13)

1. Modelo de pruebas.
2. Caso de prueba.
3. Procedimiento de prueba.
4. Componente de prueba.
5. Plan de prueba.
6. Defecto.
7. Evaluación de prueba.

#### 1.4.3.1 Documentos del proceso de pruebas utilizados en la UCI

En el proceso de prueba a realizar al producto SIAV quedan definidos los mismos entregables que define el Centro de Calidad para Soluciones Informáticas (Calisoft). (13)

1. Plan de prueba.
2. Casos de prueba.
3. Documento de No Conformidades (Resultado de la prueba).

Dentro de estos artefactos que define Calisoft el *plan de pruebas* describe las estrategias, recursos y planificación de la prueba. La estrategia de prueba incluye la definición del tipo de pruebas a realizar para cada iteración y sus objetivos, el nivel de cobertura de prueba y de código necesario y el

porcentaje de pruebas que deberían ejecutarse con un resultado específico. Todo ello con el fin de obtener un producto que tenga una buena aceptación por parte del usuario final. A continuación es necesario analizar detalladamente la estrategia de prueba de software garantizando así una mayor visión de lo que se tendría en cuenta a la hora de realizar las pruebas.

### 1.5 Estrategia de prueba de software

Una estrategia de prueba del software integra los métodos de diseño de casos de pruebas del software en una serie bien planeada de pasos que desembocará en la eficaz construcción del software. La estrategia proporciona un mapa que describe los pasos que se darán como parte de la prueba, indica cuándo se planean y cuándo se dan estos pasos, además de cuánto esfuerzo, tiempo y recursos consumirán. Por tanto, cualquier estrategia de prueba debe incorporar la planeación de pruebas, el diseño de casos de pruebas, la ejecución de pruebas, la recolección y evaluación de los datos resultantes. (17)

Se han propuesto varias estrategias de prueba del software en distintos libros; todas proporcionan al desarrollador del software una plantilla para pruebas y todas tienen las siguientes características: (18)

- Para realizar pruebas efectivas en equipo de software deben efectuar revisiones técnicas formales y efectivas. Esto eliminará muchos errores antes de que empiece la prueba.
- La prueba comienza al nivel de componentes y trabaja “hacia afuera”, hacia la integración de todo el sistema de cómputo.
- Diferentes técnicas de prueba son apropiadas en diferentes momentos.
- La prueba la dirige el desarrollador del software.
- La prueba y la depuración son actividades diferentes, pero la segunda debe incluirse en cualquier estrategia de prueba.

Una estrategia para la prueba del software debe incluir pruebas de bajo nivel (necesarias para confirmar la correcta implantación de un pequeño segmento de código fuente) y de alto nivel (que validen las principales funciones del sistema a partir de los requisitos del cliente). Una estrategia debe servir como guía para el profesional y fijar un conjunto de guías para el jefe de proyecto. Debido a que los pasos de la estrategia de prueba son simultáneos, cuando empieza a aumentar la presión por las fechas límites debe tenerse la opción de medir los avances y buscar que los problemas aparezcan lo antes posible. (19)

Sin duda alguna cada una de estas buenas prácticas a la hora de realizar el proceso de pruebas ayudará a que el mismo se lleve con la calidad que requiere. Cuando se van a realizar las pruebas es necesario analizar cuáles son los elementos que hay que tener en cuenta para realizar una prueba, ya sean aspectos estratégicos y niveles de prueba a realizar en cada uno de estos. A continuación se mostrarán dichos elementos.

### 1.5.1 Aspectos estratégicos

Si se desea implementar con éxito una estrategia de prueba de software hay que tener en cuenta los siguientes aspectos estratégicos: (20)

- *Especificar los requisitos del producto de manera cuantificable mucho antes de que empiecen las pruebas.* Aunque el objetivo primordial de la prueba es encontrar errores, una buena estrategia de prueba también evalúa otras características de la calidad.
- *Establecer los objetivos de la prueba.* Los objetivos específicos de la prueba se deben establecer en término cuantificables.
- *Comprender cuáles son los usuarios del software y desarrollar un perfil para cada categoría de usuario.* Los casos de uso que describan el escenario de interacción para cada clase de usuario reducen el esfuerzo general de prueba, ya que concentra la prueba en la utilización real del producto.
- *Desarrollar un plan de prueba que destaque la “prueba de ciclo rápido”.* La retroalimentación que generan estas pruebas de ciclo rápido se utiliza para controlar los grados de calidad y las correspondientes estrategias de prueba.
- *Construir un software “robusto” diseñado para probarse a sí mismo.* El software debe tener la capacidad de diagnosticar ciertas clases de errores. Además debe incluir pruebas automatizadas y de regresión.
- *Usar revisiones técnicas formales y efectivas como filtro previo a la prueba.* Las revisiones reducen la cantidad de esfuerzo de prueba que se requiere para producir software de alta calidad.
- *Realizar revisiones técnicas formales para evaluar la estrategia de prueba y los propios casos de prueba.* Las revisiones técnicas formales posibilitan descubrir inconsistencias, omisiones y errores evidentes en el enfoque de la prueba. Esto ahorra tiempo y también mejora la calidad del producto.

- *Desarrollar un enfoque de mejora continua para el proceso de prueba.* Es necesario medir la estrategia de prueba. Las medidas reunidas durante la prueba deben usarse como parte de un enfoque estadístico de control del proceso para la prueba del software.

### 1.5.2 Niveles de prueba

En la prueba del software es posible aplicar muchas estrategias. En un extremo, un equipo de software podría esperar hasta que el sistema esté totalmente construido y luego aplicar pruebas al sistema general esperando encontrar errores. Este enfoque, aunque atractivo, simplemente no funciona. Arrojará un software plagado de errores, molesto para el cliente y usuario final. En el otro extremo, un ingeniero de software podría aplicar pruebas diariamente, sin importar la parte del sistema que construya. Este enfoque, aunque menos atractivo para muchos, es muy efectivo. Por desgracia, la mayoría de los desarrolladores de software dudan en usarlo. (21)

La estrategia de prueba que elige la mayor parte de los equipos de software se ubica entre dos extremos. Toma un enfoque incremental de las pruebas; inicia con la prueba de unidades individuales del programa, pasa a pruebas diseñadas para facilitar la integración de las unidades, y culmina con pruebas que realizan sobre el sistema construido. (21) En las siguientes secciones se expone cada una de estas clases de prueba.

#### 1.5.2.1 Prueba de unidad

La prueba de unidad se concentra en el esfuerzo de verificación de la unidad más pequeña del diseño del software: el componente o módulo de software. Tomando como guía la descripción del diseño al nivel de componentes, se prueban importantes caminos de control para descubrir errores dentro de los límites del módulo. El alcance restringido que se ha determinado para las pruebas de unidad limita la relativa complejidad de las pruebas y los errores que éstas descubren. Las pruebas de unidad se concentran en la lógica del procesamiento interno y en las estructuras de datos dentro de los límites de un componente. Este tipo de prueba se puede aplicar en paralelo a varios componentes. (22)

#### 1.5.2.2 Prueba de integración

La prueba de integración es una técnica sistemática para construir la arquitectura del software, mientras, al mismo tiempo, se aplican las pruebas para descubrir errores asociados con la interfaz. El objetivo es tomar componentes a los que se aplicó una prueba de unidad y construir una estructura de programas que determine el diseño. (23)



En los siguientes párrafos se expondrán varias estrategias diferentes de integración incremental:

**Integración descendente:** La prueba de integración descendente es un enfoque incremental para la construcción de la arquitectura del software. El proceso de integración se realiza en una serie de cinco pasos:

1. El módulo de control principal se utiliza como controlador de prueba, y se sustituyen los resguardos en todos los componentes directamente subordinados al módulo de control principal.
2. Dependiendo del enfoque de integración seleccionado (es decir, primero-en-profundidad o primero en anchura) se van reemplazando los resguardos subordinados, uno por uno, con los componentes reales.
3. Se aplican pruebas cada vez que se integra un nuevo módulo.
4. Al contemplar cada conjunto de pruebas, se reemplaza otro resguardo con el módulo real.
5. Se aplica la prueba de regresión para asegurarse de que no se han introducido nuevos errores.

El proceso continúa a partir del paso 2 hasta la construcción total de la estructura del programa.

**Integración ascendente:** La prueba de integración ascendente empieza la construcción y la prueba con componentes de los niveles más bajos de la estructura del programa. Una estrategia de integración ascendente se implementa mediante los siguientes pasos:

1. Se combinan los módulos de bajo nivel e grupos (también llamados construcciones) que realicen sub-función específica del software.
2. Se escribe un controlador (un programa de control para pruebas) con el fin de coordinar la entrada y la salida de los casos de prueba.
3. Se prueba el grupo.
4. Se eliminan los controladores y se combinan los grupos ascendiendo por la estructura del programa.

### 1.5.2.3 Prueba de validación

Las pruebas de validación empiezan tras la culminación de la prueba de integración, cuando se han ejercitado los componentes individuales, se ha terminado de ensamblar el software como paquete y se han descubierto y corregido los errores de interfaz. En el nivel de validación o sistema desaparece la distinción entre software convencional y orientado a objetos. La prueba se concentra en las acciones visibles para el usuario y en la salida que este puede reconocer. (24)

### 1.5.2.4 Prueba del sistema

La prueba del sistema abarca una serie de pruebas diferentes cuyo propósito principal es ejercitar profundamente el sistema de cómputo. Aunque cada prueba tiene un propósito diferente, todas trabajan para verificar que se hayan integrado adecuadamente todos los elementos del sistema y que realicen las funciones apropiadas. (25)

**Prueba de recuperación:** Muchos sistemas de cómputo deben recuperarse de fallas y reanudar el procesamiento en un tiempo determinado. En algunos casos, un sistema debe ser tolerante con las fallas; es decir, las fallas de procesamiento no deben llevar a la caída del sistema en general, en otros casos, una falla del sistema debe corregirse dentro de un período específico o se sufrirá un fuerte daño económico. (25)

La prueba de recuperación es una prueba del sistema que obliga al software a fallar de varias maneras y a verificar que la recuperación se realice apropiadamente. Si la recuperación es automática (la realiza el propio sistema) debe evaluarse que sean correctos la reiniciación, los mecanismos de respaldo del sistema, la recuperación de datos y el nuevo arranque. (25)

**Prueba de seguridad:** cualquier sistema de cómputo que maneje información confidencial o que desencadene acciones que dañen (o beneficien) inapropiadamente a los individuos es un blanco para irrupciones impropias o ilegales. La prueba de seguridad comprueba que los mecanismos de protección integrados en el sistema realmente lo protejan de irrupciones inapropiadas. (25)

**Prueba de resistencia:** ejecuta un sistema de tal manera que se requiera una cantidad, una frecuencia o un volumen anormal de recursos. En esencia, la persona que aplica la prueba tratará de sobrecargar el programa. (26)

Una variante de la prueba de resistencia es una técnica denominada prueba de sensibilidad. En algunas situaciones (la más común de ellas ocurre con los algoritmos matemáticos), un rango muy pequeño de datos contenidos dentro de los límites de los datos válidos para un programa puede causar procesamiento extremo incluso erróneo, o una fuerte degradación del desempeño. Las pruebas de sensibilidad tratan de descubrir combinaciones de datos dentro de las clases de entrada válidas que causen inestabilidad o procesamiento inapropiado. (26)

**Prueba de desempeño:** se aplican para descubrir problemas de desempeño que se presentan debido a falta de recursos en el lado del servidor, ancho de banda de red inapropiado, capacidades inadecuadas de base de datos, defectuosas o débiles capacidades del sistema operativo y otros

conflictos que pueden conducir a un pobre desempeño cliente-servidor. La finalidad es doble: 1) comprender cómo responde el sistema a la carga (es decir, número de usuarios, número de transacciones o volumen de datos global), 2) recolectar métricas que conducirán a modificaciones de diseño para mejorar el desempeño. (27)

Pruebas de Carga: Una prueba de carga se realiza generalmente para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperada. Esta carga puede ser el número esperado de usuarios concurrentes utilizando la aplicación y que realizan un número específico de transacciones durante el tiempo que dura la carga.

Pruebas de Estrés o Tensión: El "Stress Test" es una prueba que mide el comportamiento del sistema bajo una cierta demanda concurrente de conexiones. Ésta es una de las pruebas clave que se deben realizar durante el ciclo de vida del software para garantizar que nuestro sistema va a cumplir con las expectativas previstas cuando sea implementado en producción. (28)

Esta prueba se utiliza normalmente para romper la aplicación. Se va doblando el número de usuarios que se agregan a la aplicación y se ejecuta una prueba de carga hasta que se rompe. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema y ayuda a los administradores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

Tanto los niveles de prueba como los tipos de prueba que se mencionan con anterioridad para ser llevados a cabo de una forma eficiente, necesitan de procedimientos o técnicas que lo guíen en su ejecución, por lo que será una necesidad analizar cada una de estas técnicas.

## 1.6 Técnica de diseño de prueba

Hay dos maneras de probar cualquier producto construido:1) si se conoce la función específica para la que se diseñó el producto, se aplican pruebas, que demuestren que cada función es plenamente operacional, mientras se buscan los errores de cada función;2) si se conoce el funcionamiento interno del producto, se aplican pruebas para asegurarse de que "todas las piezas encajan"; es decir, que las operaciones internas se realizan de acuerdo con las especificaciones y que se han probado todos los componentes internos de manera adecuada. Al primer enfoque de prueba se le denomina **prueba de caja negra (ilustración 4)**; al segundo, **prueba de caja blanca (ilustración 3)**. (29)

Las pruebas de caja negra son las que se aplican a la interfaz del software. Una prueba de este tipo examina algún aspecto funcional de un sistema que tiene poca relación con la estructura lógica interna del software. La prueba de caja blanca del software se basa en un examen cercano al detalle procedimental. (29)

A continuación se describe con detalle los métodos de caja blanca y caja negra:

### 1.6.1 Técnica de prueba Caja blanca

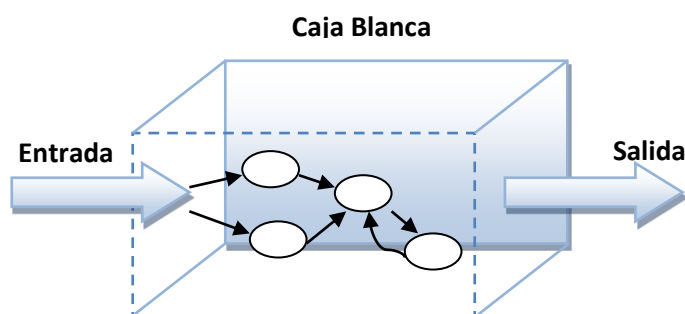


Ilustración 2: Prueba de Caja Blanca.

La *prueba de caja blanca*, en ocasiones llamada *prueba de cristal*, es un método de diseño que usa la estructura de control descrita como parte del diseño al nivel de componentes para derivar los casos de prueba. Al emplear los métodos de prueba de caja blanca, el ingeniero del software podrá derivar casos de prueba que: (30)

- 1) Garanticen que todas las rutas independientes dentro del módulo se han ejercitado por lo menos una vez.
- 2) Ejerciten los lados verdadero y falso de todas las decisiones lógicas.
- 3) Ejecuten todos los bucles en sus límites y dentro de sus límites operacionales.
- 4) Ejerciten estructuras de datos internos para asegurar su validez.

Algunas técnicas empleadas en las pruebas de caja blanca son las siguientes:

**Prueba de la ruta básica:** el método de la ruta básica permite que el diseñador de casos de prueba obtenga una medida de complejidad lógica de un diseño procedimental y que use esta medida como guía para definir un conjunto básico de rutas de ejecución. Los casos de prueba derivados para ejercitar el conjunto básico deben garantizar que se ejecuta cada instrucción del programa por lo menos una vez durante la prueba. (30)

**Pruebas de la estructura de control:** la técnica de la ruta básica descrita anteriormente es una de las varias técnicas para la prueba de estructuras de control. Aunque la prueba de la ruta básica es simple y efectiva, no es suficiente por sí misma. Por lo que se analizarán brevemente variaciones sobre la prueba de estructuras de control. Éstas ensanchan la cobertura de las pruebas y mejoran la calidad de la prueba de caja blanca. (31)

Prueba de condición: La *prueba de condición* es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en un módulo del programa. Algunos conceptos empleados alrededor de esta prueba son los siguientes: (32)

- *Condición simple:* es una variable lógica o una expresión relacional ( $E1 < \text{operador} - \text{relacional} > E2$ ) donde E1 y E2 son expresiones aritméticas.
- *Condición compuesta:* está formada por dos o más condiciones simples, operadores lógicos y paréntesis.

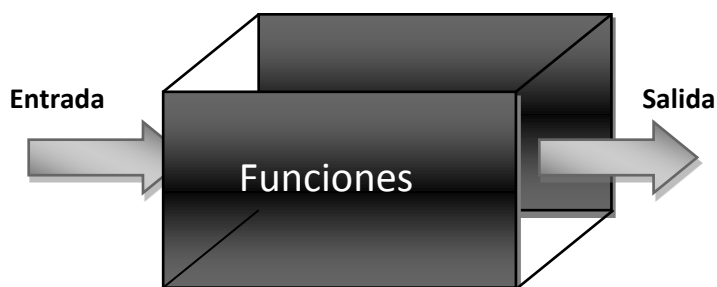
En general los tipos de errores que se buscan en una prueba de condición, son los siguientes: Error en operador lógico (existencia de operadores lógicos incorrectos, desaparecidos, sobrantes), error en variable lógica, error en paréntesis lógico, error en operador relacional, error en expresión aritmética.

Prueba del flujo de datos: El método de *prueba del flujo de datos* selecciona las rutas de prueba en un programa de acuerdo con las ubicaciones de las definiciones y los usos de las variables en el programa. El enfoque de prueba de flujo de datos se ilustra suponiendo que a cada instrucción de un programa se le asigna un número de instrucción, y que ninguna función modifica sus parámetros o variables globales. (32)

Prueba de bucles: La *prueba de bucles* es una técnica de prueba de caja blanca que se concentra exclusivamente en la validez de la construcción de bucles. Es posible definir diferentes clases de bucles: bucles simples, concatenados, anidados y no estructurados. (33)

En resumen, las pruebas de caja blanca se centran en la estructura de control del programa. Los casos de prueba se derivan para asegurar que todas las instrucciones del programa se ejecuten por lo menos una vez durante la prueba, y que todas las condiciones lógicas se ejerciten. La prueba de ruta básica, una técnica de caja blanca, aprovecha las gráficas del programa para derivar un conjunto de pruebas linealmente independientes que aseguren una cobertura. La prueba de condición y de flujo de datos ejercita aún más la lógica del programa, y la prueba de bucle complementa otras técnicas de caja blanca al proporcionar un procedimiento para ejercitar bucles con grados diversos de complejidad.

## 1.6.2 Técnica de prueba Caja negra



**Ilustración 3: Prueba de Caja Negra.**

Las *pruebas de caja negra*, también denominadas, *pruebas de comportamiento*, se concentran en los requisitos funcionales del software. Es decir, permiten al ingeniero de software derivar conjuntos de condiciones de entrada que ejercitarán por completo todos los requisitos funcionales de un programa. La prueba de caja negra no es una opción frente a las técnicas de caja blanca. Es, en cambio, un enfoque complementario que tiene probabilidades de descubrir una clase diferente de errores de los que descubrirían con los métodos de caja blanca. Este método de prueba es realizado a nivel de sistema es decir no se tiene ninguna relación con el código del producto. (34)

La prueba de caja negra, intentan encontrar errores de las siguientes categorías: (34)

- Funciones incorrectas o faltantes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de comportamiento o desempeño.
- Errores de inicialización y término.

Método de gráficos de prueba: El primer paso en la prueba de caja negra es comprender los *objetos*<sup>6</sup> modelados en el software y la relación entre ellos. La prueba empieza al crear una gráfica de objetos importantes y sus relaciones y luego idea una serie de pruebas que cubran la gráfica de tal manera que se ejercite cada objeto y relación y que se descubran errores. (35)

Partición equivalente: La *partición equivalente* es un método de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos a partir de las cuales pueden derivarse casos de prueba. La partición equivalente se esfuerza por definir un caso de prueba que descubra ciertas clases de errores, reduciendo así el número total de casos de prueba que deben desarrollarse. (36)

<sup>6</sup> En este caso el término "objeto" se considera componentes o módulos tradicionales y elementos orientados a objetos del software de cómputo.

Análisis de valores límite (AVL): El *análisis de valores límite* es una técnica de diseño de casos de prueba que complementa la partición equivalente. El AVL lleva a una selección de casos que prueba los valores al límite. En lugar de concentrarse exclusivamente en las condiciones de entrada, el AVL también deriva casos de prueba del dominio de salida. (37)

Prueba de tabla ortogonal: La *prueba de tabla ortogonal* se aplica en problemas en los cuales el dominio de entrada es relativamente pequeño, pero demasiado grande para una prueba exhaustiva. El método de prueba de tabla ortogonal resulta útil sobre todo encontrar errores asociados con las *fallas de región* (una categoría de error asociada con los defectos de la lógica en un componente de software). (38)

En resumen, las pruebas de caja negra están diseñadas para validar requisitos funcionales sin importar el funcionamiento interno de un programa. Estas técnicas de pruebas se concentran en el dominio de la información del software, derivando casos de prueba mediante partición de los dominios de entrada y salida de un programa en forma tal que proporcione cobertura completa. La partición equivalente el dominio de entrada en clases de datos que probablemente ejercitarán una función específica del software. El análisis de valores límite prueba la capacidad del programa para manejar datos en los límites de aceptabilidad. La prueba de tabla ortogonal proporciona un método eficiente y sistemático de probar con números pequeños de parámetros de entrada.

### 1.7 Conclusiones parciales

El análisis realizado en este capítulo ha demostrado de manera explícita la importancia que tiene la calidad del software, a lo que contribuyó el abarcamiento de los temas tratados. Se analizó todo lo referente al proceso de pruebas en un software y los factores por los que se ve afectada la calidad de un producto de software, garantizando con esto tener una mejor visión ante posibles fallos.

Se estudiaron los niveles de prueba, así como de los métodos o técnicas que se emplean para realizar las pruebas a cada uno de estos niveles. Se enfatizó en la metodología de desarrollo que el proyecto seleccionó con anterioridad (RUP), dando a conocer cuáles eran las ventajas que esta ofrecía para el desarrollo de las pruebas al producto. También se mostró cuáles son los principales artefactos que están estrechamente relacionados con el flujo de pruebas.

## CAPÍTULO 2: “Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV”

### 2.1 Introducción

En el capítulo anterior se realizó un análisis exhaustivo relacionado con los principales conceptos relacionados a las pruebas de software. Todo lo anteriormente planteado servirá de guía para el diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV. En este capítulo se concebirá un plan de pruebas como base para todo el proceso que se llevará a cabo. Es aquí donde quedará plasmado el diseño de casos de pruebas de todos casos de usos que serán sometidos a prueba.

### 2.2 Sistema de Gestión y Transmisión de Contenido Audiovisual (SIAV).

El Sistema de Gestión y Transmisión de Contenido Audiovisual es un producto que se está desarrollando en el centro de producción de la facultad 6. Este se identifica fundamentalmente por la integración de módulos tales como PRIMICIA, WebTV y el Sistema de Transmisión de Canales Virtuales. Para el desarrollo de la tesis solo se tendrán en cuenta los módulos PRIMICIA y WebTV.

**PRIMICIA** es un sistema informático multiplataforma que ha sido desarrollado en software libre. Se encuentra estructurado en dos subsistemas que se relacionan entre sí y actúan como un todo para brindar un resultado final eficiente y acorde a las necesidades de los usuarios. (39)

El subsistema de Administración realiza la administración del canal y toda la gestión de las informaciones y recursos multimedia que se transmiten. Está compuesto por varios módulos que encapsulan todas las funcionalidades del subsistema. (39)

El subsistema de Transmisión se encarga de la visualización de las noticias y materiales publicados en la plataforma. Está compuesto por un solo módulo que establece el flujo de trabajo fijo que se repite en el canal durante las 24 horas del día. (39)

**WebTV** enfocado principalmente a entornos de software libre, se encuentra estructurado en los subsistemas siguientes: subsistema de Administración, subsistema de Gestión y Presentación de Contenidos. (40)



## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

El subsistema de Administración tendrá las siguientes prestaciones generales: (40)

- a. Funcionalidades para la gestión de los usuarios del sistema que permitan registrar y eliminar usuarios, así como establecer o modificar los permisos de acceso en el sistema.

El subsistema de Gestión y Presentación de Contenidos tendrá las siguientes prestaciones generales: (40)

- a. Funcionalidades para la gestión de los contenidos audiovisuales que se publicarán en la Plataforma, permitiendo ingresar materiales audiovisuales al sistema para su posterior publicación.
- b. Funcionalidades para la catalogación básica de los materiales que permitan su posterior recuperación para la publicación de los mismos en la Plataforma.
- c. Funcionalidades para la recuperación de materiales que se publicarán en la Plataforma.
- d. Funcionalidades para la gestión de la publicación de contenidos audiovisuales que permitan crear, modificar, eliminar categorías de publicación de contenidos.
- e. Definir tiempos de publicación de materiales en las distintas categorías.
- f. La reproducción en un cliente de los contenidos publicados a través de un reproductor embebido en las páginas de presentación de los diferentes contenidos.
- g. La descarga del reproductor en caso de no tenerlo en la PC cliente desde donde se acceda al sistema.

Luego de haber visto cuales son las principales funcionalidades de cada subsistema es sumamente importante realizar el plan de pruebas debido a que va a ser la principal guía a tener en cuenta para realizar las pruebas. En dicho plan se van a definir las actividades, responsables y el calendario para las pruebas, por ello se convierte en una necesidad la confección del mismo. En el siguiente epígrafe se mostrarán los principales elementos del mismo.

### 2.3 Plan de Prueba

Cuando se va a realizar un proyecto sea cual sea su magnitud, la calidad es un factor indispensable en el mismo, para lograr este objetivo es importante la realización de un plan de prueba. El *plan de prueba* es un documento que contiene toda la información sobre las pruebas, es decir, en el mismo se describe la estrategia que se utilizará, los recursos que serán empleados y la planificación de estas.

## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

Además, debe reflejarse el propósito de cada una de las pruebas, sus objetivos, y lo que se va a probar con la misma.

En el plan de prueba definido para el sistema SIAV, se identifican los elementos que serán probados, los recursos necesarios para hacer las pruebas, así como la estrategia de pruebas que se llevará a cabo para lograr un buen diseño de casos de pruebas que permitan encontrar la mayor cantidad de posibles fallos del software en cuestión.

Plan de prueba realizado para el proyecto SIAV ver en: **ANEXOS/PLAN DE PRUEBAS/Plan\_de\_Pruebas\_SIAV.doc**

El objetivo fundamental que tiene este plan de pruebas para el proyecto SIAV es el establecimiento del flujo de trabajo que se va a realizar para la puesta en marcha de pruebas al producto que se propone. En el plan de pruebas definido para el sistema en cuestión, se identifican los elementos que serán probados, los recursos necesarios para hacer las pruebas, así como la estrategia de pruebas que se llevará a cabo para lograr que se encuentren con este proceso, la mayor cantidad de defectos al software y a la documentación generada. Todo este trabajo debe ser documentado y aprobado por los participantes en las pruebas que se realicen.

Para dar cumplimiento a estos objetivos trazados es necesario que se lleve a cabo una estrategia. En esta estrategia de prueba deben quedar bien definidos cuáles serán los recursos necesarios para que las mismas sean llevadas a cabo con toda la eficacia posible. Ahora en el siguiente epígrafe quedará plasmada de una manera clara toda la estrategia que se llevará durante el proceso de pruebas enmarcado en este trabajo.

### 2.4 Pasos a seguir en las pruebas o Estrategia

Teniendo en cuenta las características del producto SIAV se definen los siguientes escenarios para la aplicación de las pruebas:

Inicialmente las pruebas que se le aplicarán al producto son las de Caja Blanca. Estas pruebas se basan en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software para determinar si el estado real del producto coincide con lo esperado o afirmado. Es aquí donde se le realizarán el diseño de casos de prueba a los casos de usos (CU) seleccionados, utilizando la técnica del camino básico o prueba de la ruta básica. Esta técnica no se les aplicará a

## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

todos los CU solamente a los CU críticos con los que cuenta el proyecto SIAV, pues esta práctica de pruebas requiere mucho tiempo para su ejecución.

Posteriormente se aplicarán las pruebas de Caja Negra donde se comprueban las funcionalidades del sistema en general, para el desarrollo de las mismas se aplica específicamente la técnica de partición equivalente la cual resulta efectiva a la hora de comprobar la validez de cada entrada en los diferentes módulos. Para finalizar una vez que el software se ha integrado se deben aplicar las pruebas de Carga y Estrés a nivel de sistema midiendo el comportamiento del sistema bajo una cierta demanda concurrente de conexiones, permitiendo así que el producto final no colapse y cumpla con las expectativas previstas.

Luego de mencionar toda la estrategia a seguir para la ejecución de las pruebas, es necesario tener en cuenta cuales son los recursos que hay que tener para que estas pruebas se lleven a cabo satisfactoriamente. A continuación se mostrarán dichos elementos.

### 2.5 Escenarios de pruebas

Para la realización de las pruebas se hace necesario contar con algunos recursos tangibles dentro de los cuales están como mínimo:

#### Recursos de software:

Especificaciones de Software					
<b>Sistema Operativo:</b>		GNU/Linux Ubuntu 10.10			
		Windows 7			
Especificaciones de Hardware					
CPU	Procesador	Memoria RAM	Disco Duro	Tarjeta de Red	Tarjeta de Video
2	Intel (R) Core (TM) 2 Duo CPU E4500 @ 2.20 GHz	1 giga <sup>7</sup> byte (GB)	150 Gb	Intel (R) 82566DC gigabit Network Connection	Intel (R) G965 Express Chipset Family

Tabla 3: Escenarios de pruebas

Luego de mencionar estos recursos es necesario mencionar las herramientas a utilizar para la realización de las pruebas correspondientes.

<sup>7</sup>Unidad de medida informática cuyo símbolo es el GB.

## 2.6 Herramienta utilizada

Para realizar las pruebas de sistema particularmente las pruebas de Carga y Estrés se utilizará JMeter una herramienta Java<sup>8</sup> que permite realizar pruebas de rendimiento y pruebas funcionales sobre aplicaciones web. Utilizar JMeter en aplicaciones web para la comprobación de los recursos del sistema, supone una mayor efectividad en el proceso y en la fiabilidad de los resultados. JMeter como herramienta de prueba dispone de varios componentes que facilitan la elaboración de los escenarios de prueba con la ventaja de simular para cada uno de esos escenarios miles de usuarios. El funcionamiento de la herramienta JMeter no será documentado, ya que queda fuera del alcance de esta Tesis.

Luego de mencionar la herramienta a utilizar es necesario que se lleve a cabo un proceso de selección de los CU a los cuales se les va a realizar las pruebas. Esta selección se llevará a cabo utilizando un criterio de evaluación, con el objetivo fundamental de que esta elección de CU sea la que verdaderamente defina la calidad del producto. En el siguiente epígrafe se mostrará cómo es que se llevó a cabo todo este proceso.

## 2.7 Casos de usos seleccionados para el proceso de pruebas

El Sistema de Gestión y Transmisión de Contenido Audiovisual cuenta con diferentes CU repartidos en los módulos antes mencionados (sección 2.2). A continuación se muestra una tabla donde están recogidos estos casos de uso, separados por los subsistemas a los cuales pertenecen:

Casos de Usos del proyecto SIAV. Módulo WebTV.				
Subsistemas	CU	Prioridad	CB	CN
Administración	Gestionar tipología de archivo multimedia.	Crítico		X
Gestión y presentación de contenido	Buscar archivo multimedia.	Crítico	X	X
	Crear publicación de archivo multimedia.	Crítico		X
	Gestionar señal en vivo.	Crítico		X
<b>Web TV Total CU: 28 Críticos: 13</b>				

Tabla 4: Casos de Usos de WebTV.

<sup>8</sup>Lenguaje de programación.

## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

Casos de Usos del proyecto SIAV particularmente PRIMICIA				
Subsistemas	CU	Prioridad	CB	CN
Administración	Publicar Noticia.	Crítico		X
	Gestionar Sección.	Crítico		X
	Gestionar Infocintas.	Crítico		X
Transmisión	Transmitir canal.	Crítico	X	X
<b>PRIMICIA Total CU: 21 Críticos: 18</b>				

**Tabla 5: Casos de Usos de PRIMICIA.**

Para realizar esta selección de CU se tuvo en cuenta, la importancia que cada uno de ellos les ofrece al subsistema al cual pertenecen. Para ello se aplicó una entrevista a cada jefe de módulo del proyecto SIAV. La misma se hizo con el objetivo fundamental de que la selección de los CU fuera realizada por las personas que en realidad tienen el conocimiento óptimo acerca de todas las funcionalidades del proyecto SIAV. Con lo que se pudo determinar aquellos CU que tienen un carácter crítico dentro de un módulo, que en realidad definen la línea base de la arquitectura y que son los que verdaderamente le van a dar cumplimiento a las funcionalidades primordiales que el cliente espera recibir. Al mismo tiempo se evaluaron las características que debían cumplir los CU de carácter crítico dentro de un módulo específico y se escogieron dentro de estos los que realmente son necesarios probar en esta etapa en la que se encuentra la construcción del producto.

Resultados de las entrevistas ver en: **ANEXOS/ENTREVISTAS/Entrevista\_Módulo\_X.doc**

### 2.8 Tipos de pruebas a realizar

#### Prueba de Unidad

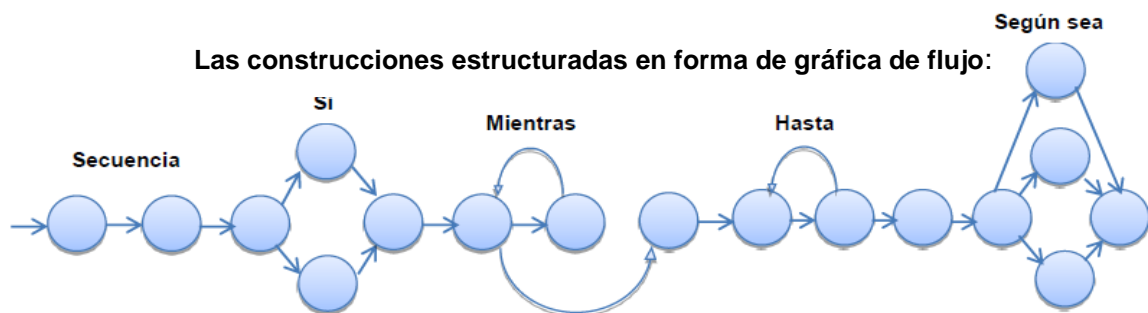
- De Caja Blanca
  - ✓ Prueba de la ruta básica

## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

El método de prueba de la ruta básica se aplica a un diseño procedimental o al código fuente. Antes de tratar el método de ruta básica, debe presentarse una notación simple para la representación del flujo de control, llamado *gráfica de flujo* (o *gráfica del programa*)<sup>9</sup>.

### Notación de gráfica de flujo

La gráfica de flujo describe un flujo de control lógico empleando la notación ilustrada en la siguiente figura.



**Ilustración 4: Notación de gráfica de flujo. (41)**

Tomando como referencia la **ilustración 5**, cada círculo, llamado *nodo de gráfica de flujo*, representa una o más instrucciones procedimentales. Las flechas en la gráfica de flujo, llamadas *aristas* o *enlaces*, representan el flujo de control. Una arista debe terminar en un nodo. Las áreas que limitan aristas y nodos se denominan *regiones*. Cuando se cuentan las regiones se incluyen las áreas ubicadas fuera de la gráfica.

### Rutas independientes del programa

Una *ruta independiente* es cualquier ruta del programa que ingresa por lo menos un nuevo conjunto de instrucciones de procesamiento o una nueva condición. Cuando se explica desde el punto de vista de una gráfica de flujo, una ruta independiente debe recorrer por lo menos una arista que no se haya recorrido antes. (42)

¿Cómo se sabe cuántas rutas buscar? El cálculo de la *complejidad ciclomática* proporciona la respuesta. La complejidad ciclomática es una métrica de software que proporciona una medida

<sup>9</sup> En realidad, el método de la ruta básica se aplica sin el uso de las gráficas de flujo. Sin embargo, sirven como notación útil para comprender el flujo de control e ilustrar el enfoque.

## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

cuantitativa de la complejidad lógica de un programa. Cuando se emplea en el contexto de método de la prueba de ruta básica, el valor calculado mediante la complejidad ciclomática define el número de rutas independientes en el conjunto básico de un programa, y proporciona un límite superior para el número de pruebas que se deben aplicar para asegurar que todas las instrucciones se hayan ejecutado por lo menos una vez. (43)

La complejidad ciclomática se basa en la teoría gráfica y se calcula de una de tres maneras: (43)

1. El número de regiones corresponde a la complejidad ciclomática.
2. La complejidad ciclomática,  $V(G)$ , de una gráfica de flujo,  $G$ , define como  $V(G) = E - N + 2$  donde  $E$  es el número de aristas, y  $N$ , el número de nodos de la gráfica de flujo.
3. La complejidad ciclomática,  $V(G)$ , de una gráfica de flujo,  $G$ , también se define como  $V(G) = P + 1$  donde  $P$  es el número de nodos predicado incluidos en la gráfica de flujo  $G$ .

### Procedimiento para la técnica de la prueba de ruta básica:

1. A partir del diseño o del código fuente, dibujar el grafo de flujo asociado.
2. Calcular la complejidad ciclomática del grafo de flujo.
3. Determinar el conjunto básico de caminos independientes.
4. Preparar los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

Se selecciona la técnica de *prueba de la ruta básica* como prueba de caja blanca debido a que es simple y efectiva, permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Además, los casos de prueba obtenidos del conjunto básico garantizarán que durante la prueba se ejecute por lo menos una vez cada sentencia del programa.

### **Prueba de Sistema**

- **De Caja Negra**
  - ✓ **Partición equivalente**

El diseño de casos de prueba para partición equivalente 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 y no válidos para las condiciones de entrada. Por lo general, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una

## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

condición booleana. Las clases de equivalencia se definen de acuerdo con las siguientes directrices:  
(36)

1. Si una condición de entrada especifica un rango, se definen una clase de equivalencia válida y dos no válidas.
2. Si una condición de entrada requiere un valor específico, se definen una clase equivalencia válida y dos no válidas.
3. Si una condición de entrada especifica un miembro de un conjunto, se definen una clase de equivalencia válida y otra no válida.
4. Si una condición de entrada es booleana, se definen una clase de equivalencia válida y otra no válida.

Al aplicar estas directrices para la derivación de clases de equivalencia, se desarrollarán y ejecutarán los casos de prueba para cada objeto de los datos del dominio de entrada. Los casos de prueba se seleccionan de modo que el mayor número de atributos de clase de equivalencia se ejercita una vez.

### Procedimiento para la técnica de partición equivalente:

1. Se identifican clases de equivalencia válida (CEV) y no válidas (CENV).
2. Asignar un número único a cada clase de equivalencia.
3. Escribir casos de prueba hasta que sean cubiertas todas las CEV, intentando cubrir en cada caso tantas CEV como sea posible.
4. Para cada CENV, escribir un caso de prueba, cubriendo en cada caso una CENV.

Se selecciona la técnica de *partición equivalente* como prueba de caja negra debido a que esta técnica se esfuerza por definir un caso de prueba que descubra ciertas clases de errores, reduciendo así el número total de casos de prueba que deben desarrollarse. Además, es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico.

- **Prueba de Desempeño**
  - ✓ **Pruebas de Carga y Estrés**

La prueba de desempeño abarca una serie de pruebas diseñadas para valorar 1) cómo afecta el aumento del tráfico de usuarios la respuesta en tiempo y confiabilidad de la Web, 2) cuáles componentes de la Aplicación Web (WebApp) son responsables de la degradación del desempeño y



## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

qué características de uso provocan que ocurra la degradación, y 3) cómo la degradación del desempeño impacta los objetivos y requisitos globales de la WebApp.

Las pruebas del desempeño se diseñan con el fin de simular situaciones de carga del mundo real. Conforme crece el número de usuarios simultáneos de la WebApp, o aumenta el número de transacciones en línea, o se incrementa la cantidad de datos (descargados o cargados), las pruebas de desempeño ayudarán a responder las siguientes preguntas:

- ¿El tiempo de respuesta del servidor se reduce hasta un punto donde es apreciable e inaceptable?
- ¿En qué punto (en términos de usuarios, transacciones o carga de datos) el desempeño se vuelve inaceptable?
- ¿Qué componentes del sistema son responsables de la reducción del desempeño?
- ¿Cuál es el tiempo de respuesta promedio para los usuarios en una variedad de condiciones de carga?
- ¿La reducción del desempeño tiene impacto sobre la seguridad del sistema?
- ¿La confiabilidad o la precisión de la WebApp se afectan conforme crece la carga del sistema?
- ¿Qué ocurre cuando se aplican cargas que rebasan la capacidad máxima del servidor?

Para desarrollar respuestas a estas preguntas, se dirigen dos pruebas de desempeño diferentes: Prueba de Carga y Prueba de Estrés.

### Prueba de Carga

El objetivo de las pruebas de carga es determinar cómo la WebApp y su ambiente del lado del servidor responderán a varias condiciones de carga. Conforme proceden las pruebas, las permutaciones a las siguientes variables definen un conjunto de condiciones de prueba:

$N$ , el número de usuarios concurrentes.

$T$ , el número de transacciones en línea por unidad de tiempo.

$D$ , la carga de datos procesada por el servidor por transacción.

En cada caso, estas variables se definen dentro de los límites operativos normales del sistema. Conforme se corre cada condición de prueba, se recopilan una o más medidas: la respuesta de usuario promedio, el tiempo promedio para descargar una unidad de datos estandarizada o el tiempo promedio para procesar una transacción. El equipo de ingeniería Web examina estas medidas para

## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

determinar si una disminución precipitada en el desempeño se puede rastrear hasta una combinación específica de  $N$ ,  $T$  y  $D$ .

La prueba de carga también se aplica para valorar la velocidad de conexión recomendada para los usuarios de la WebApp. La cantidad de información global procesada en una unidad de tiempo,  $P$ , se calcula de la forma siguiente:

$$P=N \times T \times D$$

Como ejemplo, considérese el popular sitio de noticias deportivas. En un momento dado, 20 000 usuarios concurrentes realizan una solicitud (una transacción,  $T$ ) una vez cada dos minutos promedio. Cada transacción requiere que la WebApp descargue un nuevo artículo que promedia 3 Kbyte de longitud. En consecuencia, la cantidad de información procesada en una unidad de tiempo se puede calcular como:

$$P= [20\ 000 \times 0.5 \times 3\ \text{Kb}] / 60 = 500\ \text{Kbyte} / \text{seg} = 4\ \text{megabits por segundo}$$

Por lo tanto, la conexión de red para el servidor tendría que soportar esta tasa de datos y se debería probar para garantizar que la tiene.

### Prueba de Estrés

La prueba de estrés o tensión es una continuación de la prueba de carga, pero en esta instancia las variables  $N$ ,  $T$  y  $D$  se fuerzan para alcanzar y luego superar los límites operativos. La finalidad de estas pruebas es responder cada una de las preguntas siguientes:

- ¿El sistema se degrada “gentilmente” o el servidor se desconecta cuando se rebase su capacidad?
- ¿El software del servidor genera mensajes de “servidor no disponible”? De manera general: ¿a los usuarios se les advierte que no pueden alcanzar el servidor?
- ¿El servidor pone en cola las solicitudes de recursos y vacía la cola una vez que la capacidad demanda disminución?
- ¿Las transacciones se pierden conforme se rebasa la capacidad?
- ¿La integridad de los datos se afecta cuando se rebasa la capacidad?
- Si el sistema falla, ¿Cuánto tardará en estar en línea de nuevo?

### Procedimiento:

## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

---

1. Se determina el número de usuarios que se desea intervengan a la vez en un/os camino/s crítico/s.
2. Simular con la herramienta JMeter las incorporaciones de usuarios que ejecutan el mismo camino, todos a la vez.
3. Incorporar a la herramienta JMeter más usuarios que los necesarios hasta conseguir que el sistema colapse.

La meta de probar la WebApp es ejercitar una de las muchas dimensiones de la calidad WebApp con la finalidad de encontrar o descubrir conflictos que pudieran conducir a fallas en la calidad. En este sentido se realiza este tipo de pruebas, asegurando que en condiciones particulares de trabajo no exista un punto de ruptura del sistema debido a un aumento del tráfico de usuarios, permitiendo así confiabilidad de la Web.

### 2.9 Diseño de los Casos de Prueba de Caja Blanca

Las pruebas de caja blanca ejercitan que se cumplan todos los caminos lógicos del programa. Con la aplicación de la técnica de la ruta básica se realiza un grafo de flujo del código fuente permitiendo obtener una visión clara de todos los posibles caminos independientes a probar del programa.

A continuación se muestran casos de pruebas utilizando esta técnica:

**WebTV-Caso de Prueba:** Buscar Archivo Multimedia.

**Código Fuente:** Buscar Archivo Multimedia. *Ver Anexo 1.*

**Paso 1:** Grafo de flujo asociado al código fuente.

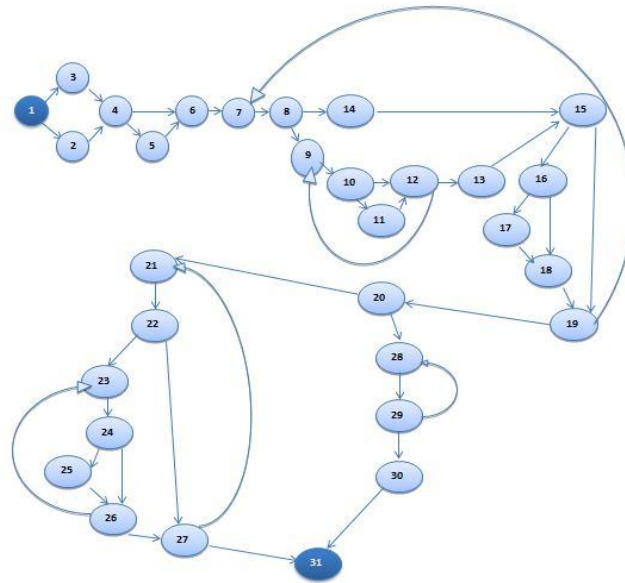


Ilustración 5: Grafo de flujo asociado al código fuente WebTV.

**Paso 2:** Cálculo de la complejidad ciclomática.

$V(G) = E - N + 2$ ; donde E es el número de aristas, y N, el número de nodos de la gráfica de flujo.

$$V(G) = 44 - 31 + 2$$

$$V(G) = 15$$

**Paso 3:** Conjunto básico de caminos independientes.

- ✓ CB 1: 1 2 4 5 6 7 8 9 10 11 12 13 15 16 17 18 19 20 21 22 23 24 25 26 27 31
- ✓ CB 2: 1 3 4 5 6 7 8 9 10 11 12 13 15 16 17 18 19 20 28 29 30 31
- ✓ CB 3: 1 2 4 6 7 8 9 10 11 12 13 15 16 17 18 19 20 21 22 23 24 25 26 27 31
- ✓ CB 4: 1 3 4 6 7 8 9 10 11 12 13 15 16 17 18 19 20 21 22 23 24 25 26 27 31
- ✓ CB 5: 1 2 4 5 6 7 8 14 15 16 17 18 19 20 21 22 23 24 25 26 27 31
- ✓ CB 6: 1 3 4 5 6 7 8 14 15 16 17 18 19 20 21 22 23 24 25 26 27 31
- ✓ CB 7: 1 2 4 6 7 8 14 15 16 17 18 19 20 21 22 23 24 25 26 27 31
- ✓ CB 8: 1 3 4 6 7 8 14 15 16 17 18 19 20 21 22 23 24 25 26 27 31
- ✓ CB 9: 1 2 4 5 6 7 8 9 10 12 13 15 16 17 18 19 20 21 22 23 24 25 26 27 31
- ✓ CB 10: 1 3 4 5 6 7 8 9 10 12 13 15 16 17 18 19 20 21 22 23 24 25 26 27 31
- ✓ CB 11: 1 2 4 6 7 8 9 10 12 13 15 16 17 18 19 20 21 22 23 24 25 26 27 31
- ✓ CB 12: 1 3 4 6 7 8 9 10 12 13 15 16 17 18 19 20 28 29 30 31
- ✓ CB 13: 1 2 4 5 6 7 8 9 10 11 12 13 15 16 18 19 20 28 29 30 31
- ✓ CB 14: 1 3 4 5 6 7 8 9 10 11 12 13 15 16 18 19 20 28 29 30 31
- ✓ CB 15: 1 2 4 6 7 8 9 10 11 12 13 15 16 18 19 20 28 29 30 31

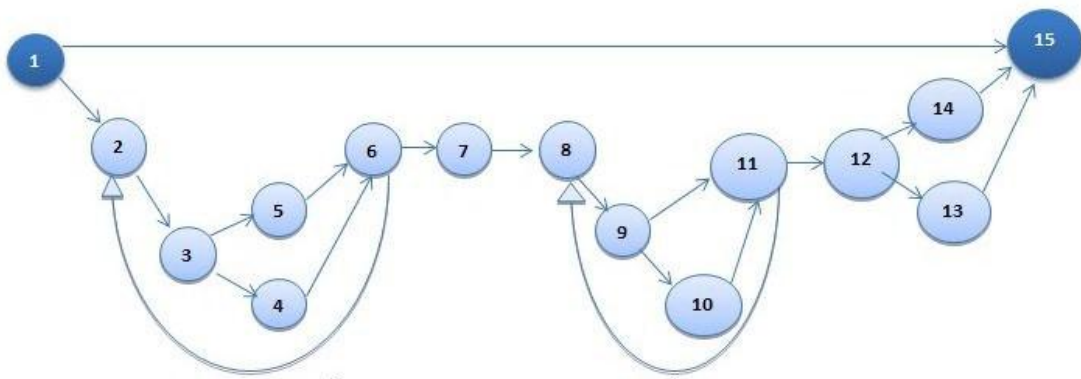
**Paso 4:** Caso de prueba para el camino básico.

- ✓ CB 1: 1 2 4 5 6 7 8 9 10 11 12 13 15 16 17 18 19 20 21 22 23 24 25 26 27 31

**PRIMICIA-Caso de Prueba:** Gestionar Sección.

**Código Fuente:** Eliminar múltiples secciones con éxito. **Ver Anexo 2.**

**Paso 1:** Grafo de flujo asociado al código fuente.



**Ilustración 6:** Grafo de flujo asociado al código fuente PRIMICIA.

**Paso 2:** Cálculo de la complejidad ciclomática.

$V(G) = E - N + 2$ ; donde E es el número de aristas, y N, el número de nodos de la gráfica de flujo.

$$V(G) = 20 - 15 + 2$$

$$V(G) = 7$$

**Paso 3:** Conjunto básico de caminos independientes.

- ✓ CB 1: 1 2 3 4 6 7 8 9 10 11 12 13 15
- ✓ CB 2: 1 2 3 4 6 7 8 9 10 11 12 14 15
- ✓ CB 3: 1 2 3 4 6 7 8 9 11 12 14 15
- ✓ CB 4: 1 2 3 5 6 7 8 9 10 11 12 13 15
- ✓ CB 5: 1 2 3 5 6 7 8 9 10 11 12 14 15
- ✓ CB 6: 1 2 3 5 6 7 8 9 11 12 13 15
- ✓ CB 7: 1 15

**Paso 4:** Caso de prueba para el camino básico.

- ✓ CB 1: 1 2 3 4 6 7 8 9 10 11 12 13 15

## 2.10 Diseño de los Casos de Prueba de Caja Negra.

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

A continuación se muestran los casos de pruebas de caja negra para los módulos de WebTV y PRIMICIA.

### Módulo WebTV

**Nombre del Caso de Uso:** Gestionar tipología de archivo multimedia.

**Descripción General:** El caso de uso se inicia cuando el Administrador desde la opción de Gestionar Tipología, en el menú de trabajo, desea añadir una nueva tipología de archivo multimedia, modificar los datos de una existente o eliminarla. Termina con la creación, modificación o eliminación de una tipología de archivo multimedia o con la cancelación de cualquiera de las operaciones anteriores por parte del Administrador.

**Condiciones de Ejecución:** Que el Administrador se haya autenticado.

**Secciones a probar en el Caso de Uso:**

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC1: Adicionar tipología.	EC1.1: Adicionar tipología exitosamente.	El sistema muestra un formulario para adicionar la tipología con los campos de la tipología (nombre con que se va a mostrar la tipología, el nombre de la tabla en la base de datos, una breve descripción, así como un cuadro seleccionable (checkbox) para escoger si la tipología creada será utilizada para categorizar archivos o es una tabla auxiliar. El Administrador entra los datos y presiona el botón "Guardar". El sistema almacena los datos de la tipología de archivo multimedia, muestra un mensaje indicando que la operación fue realizada satisfactoriamente.	Inicio WebTV/Formulario de autenticación/Interfaz de Administrar/ Gestionar tipologías de archivo multimedia/ Botón Adicionar Tipología
	EC1.2: Adicionar tipología con campos vacío.	El sistema muestra un mensaje indicando que los campos son obligatorios.	

## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

	EC1.3: Adicionar tipología con fallo.	El sistema muestra un mensaje indicando que el nombre de la tabla solo puede contener letras en minúsculas, números y guiones bajos.	
	EC1.4: Adicionar tipología repetida.	El sistema muestra un mensaje indicando que ya existe una tabla con ese nombre.	
	EC1.5: Cancelar tipología.	El Administrador presiona el botón "Cancelar".	
SC2: Sección Gestionar campos.	EC2.1: Adicionar campo exitosamente.	El sistema muestra un formulario para entrar los datos del nuevo campo (nombre del campo, nombre del campo en la base de datos, tipo del campo y descripción). El Administrador llena los datos y presiona el botón "Guardar". El sistema adiciona el nuevo campo a la tipología, muestra un mensaje indicando que la operación fue realizada satisfactoriamente.	Inicio WebTV/Formulario de autenticación/Interfaz de Administrar/ Gestionar tipologías de archivo multimedia/ Vínculo Gestionar en la columna Campos de la tipología X/ Botón Adicionar Campo.
	EC2.2: Adicionar campo con campos vacío.	El sistema muestra un mensaje indicando que los campos son obligatorios.	
	EC2.3: Adicionar campo con fallo.	El sistema muestra un mensaje indicando que el nombre de la tabla solo puede contener letras en minúsculas, números y guiones bajos.	
	EC2.4: Adicionar campo repetido.	El sistema muestra un mensaje indicando que ya existe una tabla con ese nombre.	
	EC2.5: Cancelar campo.	El Administrador presiona el botón "Cancelar".	
	EC2.6: Editar campo exitosamente.	El sistema busca los datos del campo seleccionado (nombre del campo, nombre del campo en la base de datos, tipo del campo y descripción) y los muestra en un formulario. El Administrador modifica los datos deseados y presiona el botón "Guardar". El sistema actualiza los datos del campo, muestra un mensaje indicando que la operación fue realizada satisfactoriamente.	Inicio WebTV/Formulario de autenticación/Interfaz de Administrar/ Gestionar tipologías de archivo multimedia/ Vínculo Gestionar en la columna Campos de la tipología X/ Botón Editar Campo.
	EC2.7: Editar campo con campos vacío.	El sistema muestra un mensaje indicando que los campos son obligatorios.	
	EC2.8: Editar campo con fallo.	El sistema muestra un mensaje indicando que el nombre del campo solo puede contener letras en minúsculas, números y guiones bajos.	
	EC2.9: Editar campo repetido.	El sistema muestra un mensaje indicando que ya existe un campo con ese nombre.	

## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

	EC2.10: Cancelar Edición.	El Administrador presiona el botón “Cancelar”.	
	EC2.11: Eliminar campo exitosamente.	El sistema muestra un mensaje de confirmación con las opciones “Aceptar” y “Cancelar”. El Administrador selecciona la opción “Aceptar”. El sistema elimina el campo y muestra un mensaje indicando que la operación fue realizada satisfactoriamente.	Inicio WebTV/Formulario de autenticación/Interfaz de Administrar/ Gestionar tipologías de archivo multimedia/ Vínculo Gestionar en la columna Campos de la tipología X/ Botón Eliminar Campo.
	EC2.12: Cancelar Eliminación del campo.	El Administrador selecciona la opción “Cancelar”.	
SC3: Gestionar relaciones.	EC3.1: Adicionar relación exitosamente.	El sistema muestra un formulario para entrar los datos de la nueva relación (nombre de la relación, la tabla con la que se va a relacionar, el campo por el cual se van a relacionar, el tipo de relación así como una breve descripción). El Administrador llena los datos y presiona el botón “Guardar”. El sistema adiciona la nueva relación a la tipología, muestra un mensaje indicando que la operación fue realizada satisfactoriamente.	Inicio WebTV/Formulario de autenticación/Interfaz de Administrar/ Gestionar tipologías de archivo multimedia/ Vínculo Gestionar en la columna Relaciones de la tipología X/ Botón Adicionar relación.
	EC3.2: Adicionar relación con campos vacío.	El sistema muestra un mensaje indicando que los campos son obligatorios.	
	EC3.3: Adicionar relación repetida.	El sistema muestra un mensaje indicando que ya existe una relación con esos datos.	
	EC3.4: Cancelar adición de relación.	El Administrador presiona el botón “Cancelar”.	
	EC3.5: Editar relación exitosamente.	El sistema busca los datos de la relación seleccionada (nombre de la relación, la tabla con la que se va a relacionar, el campo por el cual se van a relacionar, el tipo de relación así como una breve descripción) y los muestra en un formulario. El Administrador modifica los datos deseados y presiona el botón “Guardar”. El sistema actualiza los datos de la relación, muestra un mensaje indicando que la operación fue realizada satisfactoriamente.	Inicio WebTV/Formulario de autenticación/Interfaz de Administrar/ Gestionar tipologías de archivo multimedia/ Vínculo Gestionar en la columna Relaciones de la tipología X/ Botón Editar relación.
	EC3.6: Editar relación con campos vacío.	El sistema muestra un mensaje indicando que los campos son obligatorios.	
	EC3.7: Cancelar edición de relación.	El Administrador presiona el botón “Cancelar”.	



## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

	EC3.8: Eliminar relación exitosamente.	El sistema muestra un mensaje de confirmación con las opciones "Aceptar" y "Cancelar". El Administrador selecciona la opción "Aceptar". El sistema elimina la relación y muestra un mensaje indicando que la operación fue realizada satisfactoriamente.	Inicio WebTV/Formulario de autenticación/Interfaz de Administrar/ Gestionar tipologías de archivo multimedia/ Vínculo Gestionar en la columna Relaciones de la tipología X/ Botón Eliminar relación.
	EC3.9: Cancelar eliminación de relación.	El Administrador selecciona la opción "Cancelar".	
SC4: Editar datos de tipología.	EC4.1: Editar datos de tipología exitosamente.	El sistema busca los datos de la tipología seleccionada (nombre con que se va a mostrar la tipología, el nombre de la tabla en la base de datos, una breve descripción, así como un cuadro seleccionable (checkbox) y las muestras en un formulario. El Administrador modifica los datos deseados y presiona el botón "Guardar". El sistema actualiza los datos de la tipología de archivo multimedia, muestra un mensaje indicando que la operación fue realizada satisfactoriamente.	Inicio WebTV/Formulario de autenticación/Interfaz de Administrar/ Gestionar tipologías de archivo multimedia/ Vínculo Editar en la columna Editar de la tipología X
	EC4.2: Editar datos de tipología con campos vacío.	El sistema muestra un mensaje indicando que los campos son obligatorios.	
	EC4.3: Editar datos de tipología con fallo.	El sistema muestra un mensaje indicando que el nombre de la tabla solo puede contener letras en minúsculas, números y guiones bajos.	
	EC4.4: Editar datos de tipología repetida.	El sistema muestra un mensaje indicando que ya existe una tabla con ese nombre.	
	EC4.5: Cancelar Edición de datos de tipología.	El Administrador selecciona la opción "Cancelar".	
SC5: Crear tipología.	EC5.1: Crear tipología exitosamente.	El sistema muestra un mensaje de confirmación con las opciones "Aceptar" y "Cancelar".  El Administrador selecciona la opción "Aceptar". El sistema crea físicamente la tabla de la tipología en la base de datos, la tipología queda disponible para ser utilizada a la hora de adicionar archivos multimedia.	Inicio WebTV/Formulario de autenticación/Interfaz de Administrar/ Gestionar tipologías de archivo multimedia/ Vínculo Crear en la columna Operaciones de la tipología X
	EC5.2: Cancelar creación de tipología.	El Administrador selecciona la opción "Cancelar".	

## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

SC6: Eliminar tipología.	EC6.1: Eliminar tipología exitosamente.	El sistema muestra un mensaje de confirmación con las opciones "Aceptar" y "Cancelar".  El Administrador selecciona la opción "Aceptar". El sistema elimina la tipología, en caso de estar creada borra físicamente la tabla de la base de datos, muestra un mensaje indicando que se eliminó la tipología satisfactoriamente.	Inicio WebTV/Formulario de autenticación/Interfaz de Administrar/ Gestionar tipologías de archivo multimedia/ Vínculo Eliminar en la columna Eliminar de la tipología X
	EC6.2: Cancelar eliminación de tipología.	El Administrador selecciona la opción "Cancelar".	

**Tabla 6: CU-Gestionar tipología de archivo multimedia.**

**Nombre del Caso de Uso:** Buscar archivo multimedia.

**Descripción General:** Este caso de uso tiene lugar cuando el Editor o Revisor acceden a la funcionalidad Gestionar archivo multimedia y buscan un archivo multimedia para modificarle los datos o eliminarlo. Culmina con la devolución del resultado de la búsqueda realizada por el sistema.

**Condiciones de Ejecución:** Que el Editor o el Revisor se hayan autenticado.

**Secciones a probar en el Caso de Uso:**

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: Buscar archivos multimedia	EC 1.1: Buscar	El sistema muestra un formulario con una serie de opciones de filtrado. El sistema busca en los metadatos de los archivos multimedia almacenados en la base datos y muestra el resultado en la lista de archivos multimedia.	Inicio WebTV/Interfaz de autenticación/Gestionar/Gestionar archivo multimedia.

**Tabla 7: CU-Buscar archivo multimedia.**

**Nombre del Caso de Uso:** Crear publicación de archivo multimedia.

**Descripción General:** Este caso de uso se inicia cuando el Autor desde la opción Gestionar publicación de archivo multimedia, en el Menú de trabajo, crea una publicación de archivo multimedia a partir de un archivo multimedia previamente almacenado en el servidor de medias y le define la sección y la categoría a la que pertenece. El caso de uso termina cuando el sistema valida los datos de la publicación de archivo multimedia, muestra un mensaje de notificación al Autor de operación realizada correctamente, o termina cuando el Autor cancela la creación de la publicación del archivo multimedia.

## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

**Condiciones de Ejecución:** Que el Autor se haya autenticado, que existan archivos multimedia previamente adicionados, que existan puntos de publicación de streaming disponible.

**Secciones a probar en el Caso de Uso:**

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC1: Crear publicación de archivo multimedia.	EC1.1: Crear publicación de archivo multimedia exitosamente.	Se presiona el botón “Crear publicación de archivo multimedia”. El sistema muestra un formulario que permite nombrar la publicación, escoger las secciones y categorías a las cuales va a pertenecer, escoger el punto de publicación en el servidor de streaming, seleccionar o buscar el archivo a publicar así como introducirle una descripción a la publicación. Se entran los datos y acciona el botón “Crear publicación”. El sistema valida los datos entrados, en caso de ser correctos almacena los datos de la publicación en la base de datos y muestra un mensaje indicando que la publicación se ha creado correctamente.	Inicio WebTV/Formulario de autenticación/Interfaz de Gestionar/Gestionar publicaciones de archivo multimedia/Crear Publicación de archivo multimedia/Botón Crear Publicación
	EC1.2: Crear publicación de archivo multimedia con fallo.	Si se dejó algún campo requerido en blanco el sistema muestra un mensaje de error indicando que se deben llenar los campos obligatorios.	
	EC1.3: Cancelar la Creación de publicación de archivo multimedia.	El Autor presiona el botón “Cancelar”. El sistema regresa al listado de publicaciones.	Inicio WebTV/Formulario de autenticación/Interfaz de Gestionar/Gestionar publicaciones de archivo multimedia/Crear Publicación de archivo multimedia/Botón Cancelar

Tabla 8: CU-Crear publicación de archivo multimedia.

**Nombre del Caso de Uso:** Gestionar señal en vivo.

**Descripción General:** Este caso de uso consiste en la gestión, por parte del revisor, de las señales de audio o video en vivo, se inicia cuando el revisor se autentica y navega hasta el formulario de gestión de señales en vivo.

**Condiciones de Ejecución:** Que el revisor se haya autenticado. Que exista la señal de audio o video en vivo que se quiere transmitir.

## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

### Secciones a probar en el Caso de Uso:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC1: Crear señal en vivo.	EC 1.1: Crear señal en vivo satisfactoriamente.	El sistema muestra un formulario con los campos a llenar de la señal en vivo. El sistema valida y almacena los datos de la señal en vivo y regresa al formulario de gestión de señales en vivo.	Inicio WebTV/Formulario de autenticación/Interfaz de Gestionar/Gestionar señal en vivo/Crear señal en vivo
	EC 1.2: Crear señal en vivo. Falla.	Si se dejaron campos obligatorios en blanco o algún dato entrado es inválido el sistema muestra un mensaje de error.	
	EC 1.3 Crear señal en vivo. Cancelar.	El sistema cancela la operación y regresa al formulario de gestión de señales en vivo.	Inicio WebTV/Formulario de autenticación/Interfaz de Gestionar/Gestionar señal en vivo/Crear señal en vivo/Cancelar
SC2: Editar señal en vivo.	EC 2.1: Editar señal en vivo satisfactoriamente.	El sistema muestra un formulario con los campos a modificar de la señal en vivo. El sistema valida y actualiza los datos modificados de la señal en vivo.	Inicio WebTV/Formulario de autenticación/Interfaz de Gestionar/Gestionar señal en vivo/Editar
	EC 2.1: Editar señal en vivo falla.	El sistema muestra un mensaje indicando que los datos entrados no son válidos.	
	EC 2.1: Editar señal en vivo. Cancelar.	El sistema cancela la operación y regresa al formulario de gestión de señales en vivo.	Inicio WebTV/Formulario de autenticación/Interfaz de Gestionar/Gestionar señal en vivo/Editar/Cancelar
SC 3: Eliminar señal en vivo.	EC 3.1: Eliminar señal en vivo. Aceptar.	El sistema elimina la señal en vivo seleccionada y regresa al formulario de gestión de señales en vivo.	Inicio WebTV/Formulario de autenticación/Interfaz de Gestionar/Gestionar señal en vivo/Eliminar/Aceptar
	EC 3.1: Eliminar señal en vivo. Cancelar.	El sistema cancela la operación y regresa al formulario de gestión de señales en vivo.	Inicio WebTV/Formulario de autenticación/Interfaz de Gestionar/Gestionar señal en vivo/Eliminar/Cancelar

Tabla 9: CU-Gestionar señal en vivo.

### Módulo PRIMICIA

**Nombre del Caso de Uso:** Publicar Noticia.

**Descripción General:** El caso de uso inicia cuando el actor accede a la opción Publicación en el menú Noticias. El sistema muestra una lista con las noticias corregidas y le brinda la posibilidad de publicar,

## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

mostrar, editar, archivar, eliminar o filtrar las existentes. El caso de uso termina cuando el usuario realiza alguna de las acciones permitidas por el sistema.

**Condiciones de Ejecución:** El actor se ha autenticado y tiene los permisos necesarios para acceder al menú Editorial.

### Secciones a probar en el Caso de Uso:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: Publicar Noticia.	EC 1.1: Publicar noticia con éxito.	El usuario desea publicar una noticia, accede al menú Publicación del módulo Editorial y el sistema muestra un formulario con los datos (fecha y hora) de la noticia. Luego se guardan los datos y la noticia pasaría a ser publicada y se elimina del listado de noticias corregidas.	Subsistema de Administración/Módulo Editorial/Opción Publicar
	EC 1.2: Selección de la hora de fin menor que la de inicio.	El usuario desea publicar una noticia pero selecciona una hora de fin que es menor que la hora de inicio seleccionada.	Subsistema de Administración/Módulo de Editorial/Opción Publicar/botón Guardar
	EC 1.3: Se dejan campos vacíos.	El usuario desea publicar una noticia pero deja campos en blanco.	
SC 2: Mostrar noticia corregida.	EC 2.1: Mostar noticia con éxito.	El usuario selecciona la opción Mostrar de determinada noticia y el sistema muestra los datos de la noticia. El usuario puede retornar a la lista de noticias haciendo clic en el botón Volver a la lista.	Subsistema de Administración/Módulo de Editorial/Opción Publicar/botón Mostar
SC 3: Archivar noticia.	EC 3.1: Archivar noticia con éxito.	El usuario selecciona la opción Archivar de determinada noticia, el sistema pone el estado de la noticia como archivada y la elimina del listado de noticias corregidas.	Subsistema de Administración/Módulo de Editorial/Opción Publicar/botón Archivar
SC4: Eliminar noticia redactada.	EC 4.1: Eliminar noticia con éxito.	El usuario selecciona la opción Eliminar de una noticia y el sistema la elimina después de que el usuario haya confirmado la acción.	Subsistema de Administración/Módulo de Editorial/Opción Publicar/botón Eliminar/botón Aceptar
	EC 4.2: Se cancela la eliminación de la noticia.	El usuario cancela la ventana de confirmación y no elimina la noticia, se muestra el listado de noticias corregidas.	Subsistema de Administración/Módulo de Editorial/Opción Publicar/botón Eliminar/botón Cancelar

**Tabla 10: CU-Publicar Noticia.**

**Nombre del Caso de Uso:** Gestionar Sección.

## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

**Descripción General:** El caso de uso inicia cuando el actor accede a la función de Gestionar secciones, donde puede establecer el orden de las secciones, insertar nuevas, modificar y eliminar las existentes. Al finalizar el caso de uso queda definida la visualización de las secciones.

**Condiciones de Ejecución:** El actor se ha autenticado y tiene los permisos necesarios para realizar la Gestión de Secciones.

### Secciones a probar en el Caso de Uso:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC 1: Adicionar Sección.	EC 1.1: Adicionar sección con éxito.	El usuario desea adicionar una sección e introduce los datos necesarios., el sistema verifica que los datos sean correctos y guarda los cambios en la base de datos.	Subsistema Administración/ Módulo Secciones/opción Listado de Secciones/botón "Adicionar"
	EC 1.2: Se dejan campos vacíos.	El usuario desea adicionar una sección pero deja campos en blanco.	
	EC 1.3: Selección de la hora de fin menor que la de inicio.	El usuario desea adicionar una sección pero selecciona una hora de fin que es menor que la hora de inicio seleccionada.	
	EC 1.4: Selección de la opción "Volver a la lista".	El usuario desea volver a la lista de secciones mediante la opción correspondiente. El sistema muestra la lista de secciones.	Subsistema Administración/ Módulo Secciones/opción Listado de Secciones/botón "Adicionar"/botón "Volver a la lista"
	EC 1.5: Selecciona en el campo logo un archivo que no sea jpg o png.	El usuario selecciona en la ventana emergente de búsqueda del logo un archivo que no esté en el formato JPG o PNG. El sistema muestra un mensaje de error "Tipo de archivo no válido (sólo jpg y png)."	Subsistema Administración/ Módulo Secciones/opción Listado de Secciones/botón "Adicionar"
SC 2: Editar Sección.	EC 2.1: Editar sección con éxito.	Se muestran los campos que se pueden modificar y el usuario modifica los datos correspondientes a la sección seleccionada.	Subsistema Administración/ Módulo Secciones/opción Listado de Secciones/botón "Editar"
	EC 2.2: Se dejan campos vacíos.	El usuario desea modificar una sección pero deja campos en blanco.	
	EC 2.3: Selección de la hora de fin menor que la de inicio.	El usuario desea modificar una sección pero selecciona una hora de fin que es menor que la hora de inicio seleccionada.	
	EC 2.4: Se presiona el botón "Eliminar".	Ver SC 3 (EC 3.1, EC 3.2 y EC 3.3).	Subsistema Administración/ Módulo Secciones/opción Listado de Infocintas/botón "Editar"/botón "Eliminar"

## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

	EC 2.5: Selección de la opción "Volver a la lista".	El usuario desea volver a la lista de secciones mediante la opción correspondiente. El sistema muestra la lista de secciones.	Subsistema Administración/ Módulo Secciones/opción Listado de Secciones/botón "Editar"/botón "Volver a la lista"
	EC 2.6: Selecciona en el campo logo un archivo que no sea jpg o png.	El usuario selecciona en la ventana emergente de búsqueda del logo un archivo que no esté en el formato JPG o PNG. El sistema muestra un mensaje de error "Tipo de archivo no válido (sólo jpg y png)."	Subsistema Administración/ Módulo Secciones/opción Listado de Secciones/botón "Adicionar"
SC 3: Eliminar Sección.	EC 3.1: Eliminar sección con éxito.	El sistema elimina la sección después de que el usuario haya confirmado su eliminación.	Subsistema Administración/ Módulo Secciones/opción Listado de Secciones/botón "Eliminar"
	EC 3.2: Se cancela la eliminación de la sección.	El sistema cierra la ventana de confirmación y no elimina la sección.	Subsistema Administración/ Módulo Secciones/opción Listado de Secciones/botón "Eliminar"/botón "Cancelar"
	EC 3.3: La sección a eliminar tiene asignadas noticias o infocintas.	El usuario desea eliminar una sección que tiene asignada alguna noticia o infocinta. El sistema envía un mensaje indicando que no se puede eliminar.	Subsistema Administración/ Módulo Secciones/opción Listado de Secciones/botón "Eliminar"
SC 4: Eliminar Múltiples.	EC 4.1: Eliminar múltiples secciones con éxito.	El usuario selecciona varias secciones. El sistema las elimina luego de que el usuario haya confirmado su eliminación.	
	EC 4.2: No se selecciona ninguna sección.	El usuario no selecciona ninguna sección y presiona el botón "Eliminar".	
SC 5: Ordenar Sección.	EC 5.1: Ordenar sección con éxito.	El usuario mueve la sección que desea reubicar. Se guardan los cambios en la base de datos una vez que el usuario termina la reestructuración del orden de las secciones.	Subsistema Administración/ Módulo Secciones/opción Listado de Secciones/botón "Ordenar"
SC 6: Filtrar Sección.	EC 6.1: Filtrar sección con éxito.	El usuario llena los campos necesarios para realizar la búsqueda.	Subsistema Administración/ Módulo Secciones/opción Listado de Secciones/botón "Filtrar"
	EC 6.2: Selección de la opción "Restablecer".	El usuario desea restablecer el listado de secciones. El sistema muestra todas las secciones.	Subsistema Administración/ Módulo Secciones/opción Listado de Secciones/botón "Restablecer"
SC 7: Listar Secciones.	EC 7.1: Listar secciones con éxito.	El usuario accede al menú Secciones y el sistema muestra el listado de todas las acciones existente.	Subsistema Administración/ Módulo Secciones/opción "Listado de Secciones"

**Tabla 11: CU-Gestionar Sección.**

**Nombre del Caso de Uso:** Gestionar Infocintas.

## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

**Descripción General:** El caso de uso comienza cuando el actor decide redactar, modificar, eliminar, filtrar o establecer el orden en que se mostrarán las infocintas en sus secciones correspondientes. El actor realiza alguna de las acciones que brinda el sistema y termina así el caso de uso.

**Condiciones de Ejecución:** El actor se ha autenticado y tiene los permisos necesarios para gestionar las infocintas.

### Secciones a probar en el Caso de Uso:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC 1: Adicionar Infocinta.	EC 1.1: Adicionar infocinta con éxito.	El usuario desea adicionar una infocinta e introduce los datos necesarios, el sistema verifica que los datos sean correctos y guarda los cambios en la base de datos.	Subsistema Administración/ Módulo Infocinta/opción Listado de Infocintas/botón "Adicionar"
	EC 1.2: Se dejan campos vacíos.	El usuario desea adicionar una infocinta pero deja campos en blanco.	
	EC 1.3: Selección de la hora de fin menor que la de inicio.	El usuario desea adicionar una infocinta pero selecciona una hora de inicio que es mayor a la hora de fin que seleccionó.	
	EC 1.4: Selección de la opción "Volver a la lista".	El usuario desea volver a la lista de infocintas mediante la opción correspondiente. El sistema muestra la lista de infocintas.	
SC 2: Ordenar Infocinta.	EC 2.1: Ordenar infocinta con éxito.	El usuario mueve la infocinta que desea reubicar. Se guardan los cambios en la base de datos una vez que el usuario termina la reestructuración del orden de las infocintas.	Subsistema Administración/ Módulo Infocinta/opción Listado de Infocintas/"Ordenar"
SC 3: Editar Infocinta.	EC 3.1: Editar infocinta con éxito.	Se muestran los campos que se pueden modificar y el usuario modifica los datos correspondientes a la infocinta seleccionada.	Subsistema Administración/ Módulo Infocinta/opción Listado de Infocintas/botón "Editar"
	EC 3.2: Se dejan campos vacíos.	El usuario desea modificar una infocinta pero deja campos en blanco.	
	EC 3.3: Selección de la hora de fin menor que la de inicio.	El usuario desea modificar una infocinta pero selecciona una hora de inicio que es mayor a la hora de fin que seleccionó.	



## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

	EC 3.4: Se presiona el botón "Eliminar".	Ver SC 4 (EC 4.1 y EC 4.2).	Subsistema Administración/ Módulo Infocinta/opción Listado de Infocintas/botón "Editar"/ botón "Eliminar"
	EC 3.5: Selección de la opción "Volver a la lista".	El usuario desea volver a la lista de infocintas mediante la opción correspondiente. El sistema muestra la lista de infocintas.	Subsistema Administración/ Módulo Infocinta/opción Listado de Infocintas, botón "Editar"/ botón "Volver a la lista"
SC 4: Eliminar Infocinta.	EC 4.1: Eliminar Infocinta con éxito.	El sistema elimina la infocinta después de que el usuario haya confirmado su eliminación.	Subsistema Administración/ Módulo Infocinta/opción Listado de Infocintas/botón "Eliminar"
	EC 4.2: Se cancela la eliminación de la infocinta.	El sistema cierra la ventana de confirmación y no elimina la infocinta.	Subsistema Administración/ Módulo Infocinta/opción Listado de Infocintas/botón "Eliminar"/ botón "Cancelar"
SC 5: Eliminar Múltiples.	EC 5.1: Eliminar múltiples infocintas con éxito.	El usuario selecciona varias infocintas. El sistema las elimina luego de que el usuario haya confirmado su eliminación.	Subsistema Administración/ Módulo Infocinta/opción Listado de Infocintas/botón "Eliminar"
	EC 5.2: No se selecciona ninguna infocinta.	El usuario no selecciona ninguna infocinta y presiona el botón "Eliminar"	
SC 6: Filtrar Infocinta.	EC 6.1: Filtrar infocinta con éxito.	El usuario llena los campos necesarios para realizar la búsqueda.	Subsistema Administración/ Módulo Infocintas/opción Listado de Infocintas/botón "Filtrar"
	EC 6.2: Selección de la opción "Restablecer".	El usuario desea restablecer el listado de infocintas. El sistema muestra todas las infocintas.	Subsistema Administración/ Módulo Infocintas/opción Listado de Infocintas/botón "Restablecer"
SC 7: Listar infocintas.	EC 7.1: Listar infocintas con éxito.	El usuario accede al menú infocintas y el sistema muestra el listado de las infocintas existentes.	Subsistema Administración/ Módulo Infocintas/opción Listado de Infocintas

**Tabla 12: Gestionar Infocintas.**

**Nombre del Caso de Uso:** Transmitir Canal.

**Descripción General:** El caso de uso se inicializa cuando el actor Reloj detecta que existe alguna señal activada para transmitir. Se muestra inicialmente el video de presentación del canal. Si la señal activa es la propia del canal, se muestra la cartelera generada para el ciclo noticioso correspondiente. Al terminar de mostrar la cartelera se mostrarán las noticias según el orden que se estableció. Las noticias estarán acompañadas de un fondo musical que se reproduce durante su visualización, excepto en las pantallas de tipo video. Se mostrarán además las infocintas publicadas que estén relacionadas a la sección temática a la que pertenece la noticia, estas infocintas no se mostrarán en las pantallas de tipo imagen o video. El caso de uso termina cuando se saca el canal del aire.

**Condiciones de Ejecución:** Existe una señal de las posibles a transmitir activa.

## Capítulo 2: Diseño y aplicación de pruebas a los módulos PRIMICIA y WebTV del producto SIAV

### Secciones a probar en el Caso de Uso:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: Transmitir canal.	EC 1.1: Transmitir canal exitosamente.	El caso de uso se inicializa cuando el actor <b>reloj del sistema</b> detecta alguna señal activa para transmisión. El sistema comienza la transmisión del canal con las informaciones publicadas en la base de datos.	Subsistema de Transmisión/ Transmisión canal.
SC 2: Visualizar noticias.	EC 2.1: Visualizar noticias exitosamente.	Si la señal activa es la señal del canal se crea la cartelera con todas las noticias publicadas que su intervalo de publicación coincide con la fecha y hora actual, ordena las noticias descendientemente según la fecha de publicación y en la cartelera se mostrará número del orden de aparición de la noticia, sección temática a la que pertenece y su título. También estará acompañada de una música de fondo. Luego se muestran las noticias con las pantallas que las conforman, el audio y las infocintas asociadas. El ciclo noticioso se repite hasta que la señal cambie.	
SC 3: Fuera de servicio.	EC 3.1: Transmitir fuera de servicio.	Si la señal activa del canal es fuera de servicio, el sistema muestra el patrón del canal.	

Tabla 13: CU-Transmitir Canal.

### 2.11 Conclusiones parciales

En este capítulo se describió de forma clara las características de los diferentes módulos de los proyectos WebTV y PRIMICIA vinculados de manera directa al proyecto SIAV con el objetivo fundamental de conocer las principales funcionalidades que fueron sometidas a prueba. Fue aquí donde se definió la estrategia de prueba, los recursos necesarios y plan de pruebas que se llevarían a cabo, garantizando una mayor claridad, destreza y rapidez durante toda la ejecución de las pruebas. También se llevó a cabo un proceso de selección de los casos de uso con los cuales se desarrollaron las pruebas, favoreciendo así la realización de los diseños de pruebas correspondientes, lo que permitió dar cumplimiento a los objetivos de este trabajo.

### CAPÍTULO 3: “Resultados de las pruebas”

#### 3.1 Introducción

Luego de haber llevado a cabo todo el proceso de pruebas correspondiente al producto SIAV. En este capítulo se hace indispensable documentar todos los resultados obtenidos, con el objetivo de realizar un análisis de los errores encontrados durante todo el proceso de prueba. Posibilitando así tener un registro de estos y corregirlos en el menor tiempo posible, garantizando que el producto SIAV sea entregado al usuario final con la menor cantidad de defectos posibles.

En el presente capítulo se expondrán algunos de los resultados obtenidos al aplicar las pruebas los a módulos WebTV y PRIMICIA.

#### 3.2 Resultados obtenidos de aplicar pruebas de Caja Blanca

##### 3.2.1 WebTV-Caso de Prueba: Buscar Archivo Multimedia.

**Caso de Prueba para el camino básico:** 1 2 4 5 6 7 8 9 10 11 12 13 15 16 17 18 19 20 21 22 23 24  
25 26 27 31

**Entrada:** Publicado (un booleano), Tipologías (Una cadena de texto), es\_tipología (un booleano), restricciones (Una cadena de texto).

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

**Resultado de la Prueba:** Satisfactorio.

Para ver los resultados de los demás caminos básicos remitirse al **Anexo 3**.

##### 3.2.2 PRIMICIA-Caso de Prueba: Gestionar Sección.

**Caso de Prueba para el camino básico:** 1 2 3 4 6 7 8 9 10 11 12 13 15

**Entrada:** Seleccionar las secciones a eliminar (checkbox).

**Resultado esperado:** Se eliminan múltiples secciones.

**Resultado de la Prueba:** Satisfactorio.

Para ver los resultados de los demás caminos básicos remitirse al **Anexo 4**.

Analizados estos resultados se llega a la conclusión que los módulos PRIMICIA y WebTV no presentan ningún problema en las principales funciones implementadas, por lo que el código fuente desarrollado en ambos módulos está libre de defectos. Una vez documentado las pruebas de Caja Blanca se pasará analizar los resultados obtenidos tras aplicar las pruebas de Caja Negra.

### 3.3 Resultados obtenidos de aplicar pruebas de Caja Negra

#### 3.3.1 Módulo WebTV- Subsistema Transmisión.

3.3.1.1 Caso de Prueba: Gestionar tipología de archivo multimedia.

#### SC 1 Adicionar Tipología

ID del Escenario	Nombre	Nombre de la Tabla	Descripción	Es Tipología	Respuesta del Sistema	Resultado de la Prueba
EC1.1: Adicionar tipología exitosamente.	V/ "Deportivos"	V/"deportivo"	V/ "tipología para clasificar materiales deportivos"	N/A	El sistema muestra un mensaje de confirmación de que la operación fue realizada con éxito y regresa a la interfaz de gestión de tipologías.	Satisfactorio
EC1.2: Adicionar tipología con campos vacío.	I/ Dejar el campo en blanco	V/"deportivo"	V/ "tipología para clasificar materiales deportivos"	N/A	El sistema muestra un mensaje indicando que los campos son obligatorios.	Satisfactorio
EC1.3: Adicionar tipología con fallo.	V/ "Deportivos"	V/"Deportivo**"	V/ "tipología para clasificar materiales deportivos"	N/A	El sistema muestra un mensaje indicando que el nombre de la tabla solo puede contener letras en minúsculas, números y guiones bajos.	Satisfactorio

ID del Escenario	Nombre	Nombre de la Tabla	Descripción	Es Tipología	Respuesta del Sistema	Resultado de la Prueba
EC1.4: Adicionar tipología repetida.	V/"Series"	V/"series"	V/ <i>Clasificación para materiales seriados.</i>	N/A	El sistema muestra un mensaje indicando que ya existe una tabla con ese nombre.	Satisfactorio
EC1.5: Cancelar Adicionar tipología.	N/A	N/A	N/A	N/A	El sistema regresa a la interfaz de gestión de tipologías.	Satisfactorio

**Tabla 14: SC 1 Adicionar Tipología.**

### SC 2 Gestionar Campos

ID del Escenario	Nombre del campo	Nombre de la Tabla BD	Tipo de campo	Descripción	Es Tipología	Respuesta del Sistema	Resultado de la Prueba
EC2.1: Adicionar campo exitosamente.	V/ "CampoPrueba"	V/"campo_prueba"	V/ "varchar"	V/ "Campo de prueba para el CU"	N/A	El sistema muestra un mensaje de que la operación se realizó satisfactoriamente y regresa a la interfaz de gestión de campos de la tipología.	Satisfactorio
EC2.2: Adicionar campo con campos vacío.	V/ Vacío	V/"campo_prueba"	V/ "varchar"	V/ "Campo de prueba para el CU"	N/A	El sistema muestra un mensaje indicando si se dejó algún campo obligatorio vacío.	Satisfactorio

## Capítulo 3: Resultado de las pruebas

ID del Escenario	Nombre del campo	Nombre de la Tabla BD	Tipo de campo	Descripción	Es Tipología	Respuesta del Sistema	Resultado de la Prueba
EC2.3: Adicionar campo con fallo.	V/ "CampoPrueba"	V/"CampoPrueba"	V/ "varchar"	V/ "Campo de prueba para el CU"	N/A	El sistema muestra un mensaje indicando que el nombre de la tabla solo puede contener letras en minúsculas, números y guiones bajos. Regresa a la interfaz de adicionar campo.	Satisfactorio
EC2.4: Adicionar campo repetido.	V(para la tipología película) "Director"	V/ "director"	V/ "varchar"	V/ "Director de la película"	N/A	El sistema muestra un mensaje indicando que ya existe un campo con ese nombre dentro de la tipología.	Satisfactorio
EC2.5: Cancelar campo.	N/A	N/A	N/A	N/A	N/A	El sistema regresa a la interfaz de gestión de campos de la tipología.	Satisfactorio
EC2.6: Editar campo exitosamente.	V/ "Campo de Prueba"	V/"campo_prueba"	V/ "varchar"	V/ "Campo de prueba editado para el CU"	N/A	El sistema actualiza los datos del campo, muestra un mensaje indicando que la operación fue realizada satisfactoriamente y regresa a la interfaz de gestión de campos.	Satisfactorio
EC2.7: Editar campo con campos vacío.	V/ Vacío	V/"campo_prueba"	Tipo de campo	V/ "Campo de prueba editado para el CU"	N/A	El sistema muestra un mensaje indicando que los campos son obligatorios.	Satisfactorio

ID del Escenario	Nombre del campo	Nombre de la Tabla BD	Tipo de campo	Descripción	Es Tipología	Respuesta del Sistema	Resultado de la Prueba
EC2.8: Editar campo con fallo.	V/ "Campo de Prueba"	V/"Campo de Prueba"	V/ "varchar"	V/ "Campo de prueba editado para el CU"	N/A	El sistema muestra un mensaje indicando que el nombre del campo solo puede contener letras en minúsculas, números y guiones bajos.	Satisfactorio
EC2.9: Editar campo repetido.	V(para la tipología película) "Director"	V/ "director"	V/ "varchar"	V/ "Director de la película"	N/A	El sistema muestra un mensaje indicando que ya existe un campo con ese nombre.	Satisfactorio
EC2.10: Cancelar Edición.	N/A	N/A	N/A	N/A	N/A	El sistema regresa a la interfaz de gestión de campos.	Satisfactorio
EC2.11: Eliminar campo exitosamente.	N/A	N/A	N/A	N/A	N/A	El sistema elimina el campo y muestra un mensaje indicando que la operación fue realizada satisfactoriamente y regresa a la interfaz de gestión de campos.	Satisfactorio
EC2.12: Cancelar Eliminación del campo.	N/A	N/A	N/A	N/A	N/A	El sistema regresa a la interfaz de gestión de campos.	Satisfactorio

**Tabla 15: SC 2 Gestionar Campos**

### SC 3 Gestionar Relaciones

ID del Escenario	Nombre de la relación	Tabla de relación	Campo de relación	Tipo de relación	Requerido	Descripción	Respuesta del Sistema	Resultado de la Prueba
EC2.1: Adicionar relación exitosamente.	V/"Relación Género"	V/"Género archivo multimedia"	V/"id"	N/A	N/A	N/A	El sistema adiciona la nueva relación a la tipología, muestra un mensaje indicando que la operación fue realizada satisfactoriamente. El sistema regresa a la interfaz de gestión de relaciones.	Satisfactorio
EC2.2: Adicionar relación con campos vacío.	V/Dejar el campo vacío	V/"Género archivo multimedia"	V/"id"	V/"1.*"	N/A	N/A	El sistema muestra un mensaje indicando que los campos son obligatorios.	Satisfactorio
EC2.3: Adicionar relación repetida.	V/(para la tipología Audio)	V"Autor"	V/"id"	V/"1.*"	N/A	N/A	El sistema muestra un mensaje indicando que ya existe una relación con esos datos.	Satisfactorio
EC2.4: Cancelar adición de relación.	"Autor"	N/A	N/A	N/A	N/A	N/A	El Administrador presiona el botón "Cancelar".	Satisfactorio



## Capítulo 3: Resultado de las pruebas

ID del Escenario	Nombre de la relación	Tabla de relación	Campo de relación	Tipo de relación	Requerido	Descripción	Respuesta del Sistema	Resultado de la Prueba
EC2.5: Editar relación exitosamente.	N/A	V/"Autor"	V/"id"	N/A	N/A	N/A	El sistema actualiza los datos de la relación, muestra un mensaje indicando que la operación fue realizada satisfactoriamente.	Satisfactorio
EC2.6: Editar relación con campos vacío.	V/(para la tipología Audio)	V/"Autor"	V/"id"	N/A	N/A	N/A	El sistema muestra un mensaje indicando que los campos son obligatorios.	Satisfactorio
EC2.7: Cancelar edición de relación.	"Autor"	N/A	N/A	N/A	N/A	N/A	El sistema regresa a la interfaz de gestión de relaciones.	Satisfactorio
EC2.8: Eliminar relación exitosamente.	V/(para la tipología Audio)	N/A	N/A	V/"1.*"	N/A	N/A	El sistema elimina la relación y muestra un mensaje indicando que la operación fue realizada satisfactoriamente.	Satisfactorio
EC2.9: Cancelar eliminación de relación.	Dejar el campo vacío	N/A	N/A	V/"1.*"	N/A	N/A	El sistema regresa a la interfaz de gestión de relaciones.	Satisfactorio

Tabla 16: SC 3 Gestionar Relaciones

### SC4 Editar Datos de Tipología

ID del Escenario	Nombre	Nombre de la Tabla	Descripción	Es Tipología	Respuesta del Sistema	Resultado de la Prueba
EC4.1: Editar datos de tipología exitosamente.	V/ (para la tipología Películas) "Película"	V/"películas"	V/ "Largometraje"	V/ marcado	El sistema actualiza los datos de la tipología de archivo multimedia, muestra un mensaje indicando que la operación fue realizada satisfactoriamente y regresa a la interfaz de gestión de tipologías.	Satisfactorio
EC4.2: Editar datos de tipología con campos vacío.	I/ (para la tipología Películas) Vacío	V/"películas"	V/ "Largometraje"	V/ marcado	El sistema muestra un mensaje indicando que los campos son obligatorios.	Satisfactorio
EC4.3: Editar datos de tipología con fallo.	V/ (para la tipología Películas) "Película"	V/"Pelis"	V/ "Largometraje"	V/ marcado	El sistema muestra un mensaje indicando que el nombre de la tabla solo puede contener letras en minúsculas, números y guiones bajos.	Satisfactorio
EC4.4: Editar datos de tipología repetida.	V/ (para la tipología Películas) "Series"	V/"películas"	V/ "Largometraje"	V/ marcado	El sistema muestra un mensaje indicando que ya existe una tabla con ese nombre.	Satisfactorio
EC4.5: Cancelar Edición de datos de tipología.	N/A	N/A	N/A	N/A	El sistema regresa a la interfaz de gestión de tipologías.	Satisfactorio

Tabla 17: SC 4 Editar Datos de Tipología

### SC 5 Crear Tipología

ID del Escenario	Respuesta del Sistema	Resultado de la Prueba
EC5.1: Crear tipología exitosamente.	El sistema crea físicamente la tabla de la tipología en la base de datos, la tipología queda disponible para ser utilizada a la hora de adicionar archivos multimedia. El sistema regresa a la interfaz de gestión de tipologías.	Satisfactorio
EC5.2: Cancelar creación de tipología.	El sistema regresa a la interfaz de gestión de tipologías.	Satisfactorio

Tabla 18: SC 5 Crear Tipología

### SC 6 Eliminar Tipología

ID del Escenario	Respuesta del Sistema	Resultado de la Prueba
EC5.1: Crear tipología exitosamente.	El sistema crea físicamente la tabla de la tipología en la base de datos, la tipología queda disponible para ser utilizada a la hora de adicionar archivos multimedia. El sistema regresa a la interfaz de gestión de tipologías.	Satisfactorio
EC5.2: Cancelar creación de	El sistema regresa a la interfaz de gestión de tipologías.	Satisfactorio

Tabla 19: SC 6 Eliminar Tipología

### 3.3.2 Módulo WebTV- Subsistema Gestión y Presentación de Contenido

3.3.2.1 Caso de Prueba: Buscar archivo multimedia.

#### SC 1 Buscar Archivo Multimedia

ID del Escenario	Buscar	Tipologías	Respuesta del Sistema	Resultado de la Prueba
EC 1.1: Buscar.	V/ "Buena Fe"	N/A	El sistema muestra un listado con los archivos multimedia que coincidan con el criterio de búsqueda escrito.	Satisfactorio

**Tabla 20: SC 1 Buscar Archivo Multimedia**

**3.3.2.2 Caso de Prueba:** Crear publicación de archivo multimedia. *Ver Anexo 5.*

**3.3.2.3 Caso de Prueba:** Gestionar señal en vivo. *Ver Anexo 6.*

### 3.3.3 Módulo PRIMICIA- Subsistema Administración

**3.3.3.1 Caso de Prueba:** Publicar Noticia.

#### SC 1 Publicar Noticia

ID del Escenario	Comienza	Termina	Respuesta del Sistema	Resultado de la Prueba
1.1: Publicar noticia con éxito.	V/ 09/25/2011 00:00	V/ 10/25/2011 10:00	El sistema verifica que los datos sean correctos y guarda los cambios en la base de datos.	Satisfactorio
EC 1.2: Se dejan campos vacíos.	V/ 09/20/2011 9:40	I/ ( )	El sistema envía un mensaje indicando que los campos son requeridos.	Satisfactorio
	V/ 09/25/2011 10:00	I/ --/25/2011 10:00	El sistema envía un mensaje indicando que el campo es obligatorio.	Satisfactorio
EC 1.3: Selección de la hora de fin menor que la de inicio.	V/ 09/25/2011 10:00	I/ 09/25/2010 10:00	El sistema envía un mensaje indicando que la fecha de inicio debe ser menor que la fecha de fin.	Satisfactorio

**Tabla 21: SC 1 Publicar Noticia**

#### SC 2 Mostrar Noticia

ID del Escenario	Respuesta del Sistema	Resultado de la Prueba
EC 2.1: Mostrar noticia con éxito.	Muestra los datos de la noticia: Título, Sección, Audio, Total de Pantallas, Mostrándose, Tipo de Pantalla, Pantallas; además de una imagen que muestra como se visualizará en el canal la pantalla seleccionada.	Satisfactorio

**Tabla 22: SC 2 Mostrar Noticia**

### SC 3 Archivar Noticia

ID del Escenario	Respuesta del Sistema	Resultado de la Prueba
EC 3.1: Archivar noticia con éxito.	El sistema pone el estado de la noticia como archivada y la elimina del listado de noticias corregidas.	Satisfactorio

Tabla 23: SC 3 Archivar Noticia

### SC 4 Eliminar Noticia

ID del Escenario	Respuesta del Sistema	Resultado de la Prueba
EC 4.1: Eliminar noticia con éxito.	El sistema elimina la noticia seleccionada después de que el usuario confirma la acción.	Satisfactorio
EC 4.2: Se cancela la eliminación de la noticia.	No elimina la noticia y muestra el listado de noticias corregidas.	Satisfactorio

Tabla 24: SC 4 Eliminar Noticia

3.3.3.2 Caso de Prueba: Gestionar Sección. *Ver Anexo 7.*

3.3.3.3 Caso de Prueba: Gestionar Infocintas. *Ver Anexo 8.*

### 3.3.4 Módulo PRIMICIA- Subsistema Transmisión

3.3.4.1 Caso de Prueba: Transmitir Canal.

### SC 1 Transmitir Canal

ID del Escenario	Respuesta del Sistema	Resultado de la Prueba
EC 1.1: Transmitir canal exitosamente.	El caso de uso se inicializa cuando el actor <b>reloj del sistema</b> detecta alguna señal activa para transmisión. El sistema comienza la transmisión del canal con las informaciones publicadas en la base de datos.	Satisfactorio.

Tabla 25: SC 1 Transmitir Canal

### SC 2 Visualizar Canal

ID del Escenario	Respuesta del Sistema	Resultado de la Prueba
EC 2.1: Visualizar noticias exitosamente.	Crea la cartelera con las noticias publicadas, ordena las noticias descendientemente Se muestran las noticias con las pantallas que las conforman, el audio y las infocintas asociadas. El ciclo noticioso se repite hasta que la señal cambie.	Satisfactorio.

**Tabla 26: SC 2 Visualizar Canal**

### SC 3 Transmitir Fuera de Servicio

Escenario	Respuesta del Sistema	Resultado de la Prueba
EC 3.1: Transmitir fuera de servicio.	Si la señal activa del canal es fuera de servicio, el sistema muestra el patrón del canal.	Satisfactorio.

**Tabla 27: SC 3 Transmitir Fuera de Servicio**

Haciendo una breve comparación con los errores que se esperaban del sistema y los que se obtuvieron, se puede decir que el sistema realiza la mayor parte de las funcionalidades críticas que fueron definidas en la etapa inicial del producto. A continuación se mencionan los principales defectos encontrados durante todo el proceso de pruebas de Caja Negra:

Inicialmente en la primera iteración de SIAV se detectaron 40 no conformidades, los principales defectos encontrados estuvieron relacionados con errores ortográficos en los documentos oficiales, diseños de casos de pruebas mal confeccionados que no se correspondían con la versión del documento rector o diseños de casos de pruebas en los que en la matriz de datos se tomaban los botones como variables e incluso les hacían tablas aparte de las secciones.

Secciones	Resultado Esperado	Resultado Obtenido	Clasificación
<b>CP Gestionar señal en vivo en la Sección Crear señal en vivo.</b>	Se dice en el informe que el sistema muestra un formulario con los campos a llenar de la señal en vivo.	En la aplicación no hay una opción para poder crear una señal en vivo.	Significativa.

## Capítulo 3: Resultado de las pruebas

<b>CP Gestionar señal en vivo en la Sección Editar señal en vivo.</b>	Se dice en el informe que el sistema muestra un formulario con los campos a modificar de la señal en vivo.	En la aplicación, no se encuentran registradas las señales mencionadas, por lo que no se podrá modificar ninguna, pues no existen.	Significativa.
<b>CP Gestionar señal en vivo en la Sección eliminar señal en vivo.</b>	Se dice en el informe que el sistema muestra un formulario con el campo eliminar señal en vivo.	En la aplicación, no se encuentran registradas las señales mencionadas, por lo que no se podrá eliminar ninguna, pues no existen.	Significativa.
<b>CP Gestionar Sección en la Sección Editar Sección.</b>	En el EC 2.6: <i>Selecciona en el campo logo un archivo que no sea jpg o png</i> , el flujo central se especifica, que el botón con el que se trabaja es "Adicionar" y en la aplicación el nombre de este botón es " Editar ".	Imprecisiones en los nombres de los botones. Debe tenerse en cuenta el nombre de dicho botón, que tanto en el documento como en la aplicación coincidan para un mayor entendimiento del probador.	Significativa.
<b>CP Buscar archivo multimedia.</b>	A partir de una cadena de texto mostrar los resultados que coincidan totalmente con la cadena de texto.	No muestra el resultado de la búsqueda cuando solo parte de la cadena coincide con la media buscada. No es óptimo.	No Significativa.

Luego de haber visto los principales defectos encontrados y los resultados obtenidos durante la ejecución de las pruebas funcionales se hace indispensable que todos estos defectos encontrados se resuelvan en el menor tiempo posible por el equipo de desarrollo. Una vez corregidas estas no conformidades se decide pasar a la segunda iteración obteniendo como resultado un total de 10 no conformidades relacionadas con validaciones de datos de entrada, inmediatamente son reportadas al equipo de desarrollo el cual las soluciona en un breve período de tiempo, por lo que se llega a concluir que las pruebas aplicadas son satisfactorias ya que los módulos realizan las funcionalidades previstas. Una vez documentado las pruebas de Caja Negra se pasará a analizar los resultados obtenidos tras aplicar las pruebas de rendimiento.

### 3.4 Resultados obtenidos de aplicar pruebas de Carga y Estrés

#### 3.4.1 DCP Carga y Estrés: WebTV.

**Descripción General:** La pantalla referente a WebTV, permite acceder a otros vínculos como son: TV y Radio, Videos Musicales, Películas, Series, Docentes, Documentales, Avisos.

#### Condiciones de Ejecución:

- Para poder acceder a esta pantalla debe estar corriendo el servidor de aplicación.
- Las características del servidor deben ser de: Memoria RAM: 1 Gb DDR2 y Disco Duro 160 GB.
- Las características de la red: 100.0 Mps.

#### Secciones a probar en el Caso de Uso:

Sección	Escenarios de la Sección	Carga de Trabajo y Estrés	Descripción	Resultado esperado	Resultado de la prueba
SC1: Acceder a la Interfaz inicial del módulo WebTV.	EC 1.1: Acceder a la Interfaz inicial del módulo WebTV.  http://10.34.15.11	10/20/50	Esta interfaz es el inicio por tanto es la primera página por la que accederán todos los usuarios del sistema. Es el puente hacia el resto de las páginas que componen el sitio.	La aplicación ha respondido a una velocidad menor de 20 segundos.	10.2 seg  / 31.4 seg / error
SC2: Videos Musicales.	EC 2.1: Mostrar Videos.  Para ello ejecutar clic en la pestaña "Nacionales" perteneciente a "Videos Musicales"	10/20/50	Esta interfaz muestra todos los videos nacionales que están publicados en WebTV.	La aplicación ha respondido a una velocidad menor de 20 segundos.	10.2 seg  / 31.4 seg / error



Sección	Escenarios de la Sección	Carga de Trabajo y Estrés	Descripción	Resultado esperado	Resultado de la prueba
	<p>EC 2.2: Reproducir Videos Musicales</p> <p>Para ello ejecutar clic en el vínculo "Abrir en un reproductor embebido en el navegador" situado en la parte central del video musical.</p>	10/20/50	Esta interfaz se muestra aparte de la propia página WebTV, reproduciendo el video musical.	La aplicación ha respondido a una velocidad menor de 20 segundos.	10.2 seg / 31.4 seg / error

**Tabla 28: DCP Carga y Estrés: WebTV.**

### 3.4.2 DCP Carga y Estrés: PRIMICIA.

**Descripción General:** La pantalla referente a PRIMICIA, permite que después que el usuario se autentique pueda acceder a otros vínculos como son: Transmisión, Noticias, Medias, Seguridad.

#### Condiciones de Ejecución:

- Para poder acceder a esta pantalla debe estar corriendo el servidor de aplicación.
- Las características del servidor deben ser de: Memoria RAM: 1 Gb DDR2 y Disco Duro 160 GB.
- Las características de la red: 100.0 Mps.

#### Secciones a probar en el Caso de Uso:

Sección	Escenarios de la Sección	Carga de Trabajo y Estrés	Descripción	Resultado esperado	Resultado de la prueba
SC1: Acceder a la Interfaz inicial del módulo PRIMICIA.	<p>EC 1.1: Acceder a la Interfaz inicial del módulo PRIMICIA.</p> <p><a href="http://10.34.15.11:5801/">http://10.34.15.11:5801/</a></p>	10/15	Esta interfaz es el inicio por tanto es la primera página por la que accederán todos los usuarios del sistema. Primeramente se deberán autentificar todos los usuarios permitiendo el puente hacia el resto de las páginas que componen el sitio.	La aplicación ha respondido a una velocidad menor de 20 segundos.	14.2 seg / 23.9 seg

Sección	Escenarios de la Sección	Carga de Trabajo y Estrés	Descripción	Resultado esperado	Resultado de la prueba
SC2: Publicar Noticia.	EC: 2.1 Mostrar Publicaciones  Para ello ejecutar clic en la pestaña "Publicación" perteneciente a "Noticias"	10/15	Esta interfaz muestra el listado de noticias a publicar en PRIMICIA.	La aplicación ha respondido a una velocidad menor de 20 segundos.	14.2 seg / 23.9 seg
	EC: Publicar Noticia  Para ellos ejecutar clic en la pestaña "Publicar".	10/15	Esta interfaz muestra la publicación seleccionada para publicarla en una fecha determinada.	La aplicación ha respondido a una velocidad menor de 20 segundos.	14.2 seg / 23.9 seg

**Tabla 29: DCP Carga y Estrés: PRIMICIA.**

Durante la realización de las pruebas de carga y estrés referente a los módulos de WebTV y PRIMICIA se pudo recopilar cierta información vinculada al funcionamiento del sistema bajo condiciones extremas:

Para el caso de prueba WebTV inicialmente estaba previsto una carga de trabajo de 50 hilos pero tras aplicar la prueba se nota que el sistema se demoraba más de lo establecido con esta cantidad de usuarios conectados simultáneamente, por lo que se decide un decremento de la cantidad de hilos teniendo cuenta que el servidor donde está montada la aplicación no tiene buenas condiciones técnicas, luego de observar los resultados obtenidos para el nuevo número de hilos se llega a la conclusión de que la prueba es satisfactorio debido a que el sistema respondió de acuerdo a lo esperado.

Para el caso de prueba PRIMICIA luego de observar los resultados obtenidos con los hilos previstos se llega a la conclusión que la prueba es satisfactorio debido a que el sistema respondió de acuerdo a lo esperado.

Para mayor información de las pruebas realizadas con la herramienta JMeter: **Ver Anexo 9.**

### 3.5 Conclusiones parciales

Al finalizar este capítulo, se puede decir que el proceso de pruebas realizado a los módulos del producto SIAV se llevó a cabo de forma satisfactoria, pues mediante los casos de prueba confeccionados se obtuvieron muchos de los errores con los que contaba este sistema. En este capítulo quedaron documentados todos los defectos encontrados, luego de haber realizado las pruebas. Se realizó una comparación entre los resultados de las funcionalidades que se esperaban obtener del producto SIAV y los resultados que se obtuvieron realmente, teniendo con esto una visión más clara acerca de la calidad que presenta el sistema SIAV en la actualidad. En este capítulo se cumplió el objetivo fundamental del mismo, que era dar a conocer todos los defectos con los que contaba el producto luego de aplicar las pruebas correspondientes.

### CONCLUSIONES

En la presente investigación se le realizaron las pruebas a los módulos PRIMICIA y WebTV. De esta forma se le dio cumplimiento a los objetivos de la investigación, llegando a las siguientes conclusiones:

- La confección del plan de pruebas para los módulos PRIMICIA y WebTV permitió describir la estrategia, recursos, planificación de las pruebas y el cronograma de pruebas para el producto SIAV garantizando con esto que todo el proceso de pruebas fuera de una forma más clara y tangible.
- Se verificó la correcta implementación de los requerimientos planteados por los clientes de SIAV en un inicio, mediante las pruebas de caja blanca utilizando la técnica de la ruta básica, las pruebas de caja negra particularmente la técnica de partición equivalente y las pruebas de rendimiento específicamente pruebas de carga y estrés.
- Mediante la aplicación de las pruebas utilizando los diseños de casos de prueba se detectaron errores que son significativos y que no fueron detectados por los desarrolladores.

De forma general se puede decir que el flujo de prueba se ejecutó exitosamente partiendo de que el éxito de las pruebas se encuentra en la detección de errores. Todos los errores encontrados se documentaron e informaron al equipo de desarrollo. Se lograron los objetivos propuestos a inicios de este trabajo, logrando que se llevara a cabo un proceso de prueba al producto SIAV con el fin de garantizarle la calidad.

### RECOMENDACIONES

Una vez concluido este trabajo y dada la importancia que tiene el proceso de pruebas para cualquier sistema informático, se recomienda:

- Que se le apliquen las pruebas a los caso de usos que no fueron sometidos a prueba completando en su totalidad los CU de proyecto SIAV y a los que fueron sometidos a prueba que se le corrijan todos los errores encontrados.
- Mejorar las especificaciones de hardware donde está montada la aplicación con el objetivo de reducir el tiempo de respuesta del servidor, contribuyendo de esta manera que la confiabilidad del producto SIAV no sea afectada.

### BIBLIOGRAFÍA REFERENCIADA

1. **Pressman, Roger S.** *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 772.
2. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 769.
3. Diccionario de la Real Academia de la Lengua Española. 22.a edición, formato electrónico. [En línea] [Citado el: 27 de noviembre de 2011.] <http://buscon.rae.es/diccionario/drae.htm>.
4. **Lovelle, Juan Manuel Cueva.** Calidad del software. [En línea] 21 de octubre de 1999. [Citado el: 27 de noviembre de 2011.] [http://www.calidaddelsoftware.com/index.php?option=com\\_content&view=frontpageplus&Itemid=63](http://www.calidaddelsoftware.com/index.php?option=com_content&view=frontpageplus&Itemid=63).
5. slideshare. [En línea] [Citado el: 27 de noviembre de 2011.] <http://www.slideshare.net/lcahuich/calidad-del-software-presentation>.
6. **Pressman, Roger S.** *Ingeniería del software, Un enfoque Práctico*. Quinta Edición. s.l. : McGraw-Hill, 2002.
7. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 415.
8. Camova. *GERENCIA Y ASEGURAMIENTO DE LA CALIDAD*. [En línea] [Citado el: 27 de noviembre de 2011.] <http://camova.lacocelera.net/post/2010/>.
9. **Pressman, Roger S.** *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 770.
10. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. págs. 464-466.
11. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 473.
12. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 497.
13. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. [ed.] Andrés Otero. [trad.] Salvador Sánchez, y otros. Madrid : Pearson Edacación, S.A., 2000.
14. **Martínez, Alejandro y Raúl.** [En línea] [Citado el: 2 de diciembre de 2011.] <http://www.dsi.uclm.es/asignaturas/42551>.

15. **Molpeceres, Alberto.** willydev. [En línea] Revisión 1.0 (15.2.2003), 15 de diciembre de 2002. [Citado el: 2 de diciembre de 2011.] <http://www.willydev.net/descargas/Articu>.
16. Análisis1DAID. [En línea] [Citado el: 1 de diciembre de 2011.] [http:// analisis1daid.wikispaces.com/Metodologia\\_RUP\\_UML](http:// analisis1daid.wikispaces.com/Metodologia_RUP_UML).
17. **Pressman, Roger S.** *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 382.
18. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 383.
19. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. págs. 383-384.
20. —. *Ingeniería del Software, Un enfoque Práctico*. . Sexta Edición. s.l. : McGraw-Hill, 2007. págs. 390-391.
21. —. *Ingeniería del Software, Un enfoque Práctico*. . Sexta Edición. s.l. : McGraw-Hill , 2007 . pág. 391.
22. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 392.
23. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. págs. 394-395.
24. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 404.
25. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 407.
26. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 408.
27. —. *Ingeniería del Software, Un enfoque Práctico*. . Sexta Edición. McGraw-Hill : s.n., 2007. págs. 407-408.
28. El Blog de Leonardo Horikian. [En línea] 25 de junio de 2009. [Citado el: 17 de febrero de 2012.] <http://lhorikian.blogspot.com/2009/06/stress-test.html>.
29. **Pressman, Roger S.** *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 422.
30. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill , 2007 . pág. 423.
31. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. págs. 430-431.
32. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 431.

33. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 432.
34. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 433.
35. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 434.
36. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : 2007, McGraw-Hill. pág. 436.
37. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 437.
38. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 438.
39. **Hidalgo, Maralys Domenech**. Documento Visión 2.0. 1.5 2011.
40. **Delgado, Yunier Albrecht**. Documento Vision 2.0. 2009.
41. **Pressman, Roger S**. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. págs. 423-424.
42. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 425.
43. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 426.
44. —. *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición. s.l. : McGraw-Hill, 2007. pág. 767.
45. —. *Ingeniería del Software, Un enfoque Práctico*. [ed.] McGraw-Hill. Sexta Edición. 2007. pág. 463.
46. ISO. [En línea] [Citado el: 3 de diciembre de 2011.] [http://www.iso.org/iso/iso\\_9000\\_essentials](http://www.iso.org/iso/iso_9000_essentials).
47. Software Engineering Institute. [En línea] [Citado el: 3 de diciembre de 2011.] <http://www.sei.cmu.edu/cmml/>.
48. **Mc-Graw-Hill**. *Pressman, R. Ingeniería del software: un enfoque práctico*. 2002.
49. **McGraw-Hill**. *R.S. Pressman. Ingeniería de Software. Un enfoque Práctico*. s.l. : 4ta Edición, 1998.
50. **calisoft**. Curso calidad. *Introducción a la Pruebas de software*. C. Habana : s.n., 2009.
51. **Collazo, Manuel**. *Técnicas de prueba del software. Estrategias de prueba del software*. s.l. : IEEE Standard Glossary of Software Engineering Terminology 1990, 2003.