



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
Facultad 9

Diseño y aplicación de pruebas al producto PRIMICIA, Plataforma de Televisión Informativa.

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN
INFORMÁTICA**

AUTOR

Yisel Tornés López

TUTOR

Ing. Jorge Daniel Olivares Tamayo

Ciudad de La Habana, enero de 2009

“Año del 50 Aniversario del triunfo de la Revolución”



“Las cosas buenas se deben hacer sin llamar al Universo para que lo vea a uno pasar”

José Martí

A mi mamá...

A mi abuelita....

A toda mi familia....

A los amigos de toda la vida....

Agradecimientos

A mi mamá y mi abuela por darme este tamaño llenándome de amor y apoyándome en mis decisiones, confiando en mí.

A mis titis Sandra y Ignacia por su apoyo y cariño.

A mis titos Aleixi, Julio, Alexander que siempre me han querido como una hija y me han ayudado para salir adelante.

A mi hermana de quien aspiro ser quía, mis primos que quiero como hermanos, mi tía Trina que quiero y respeto mucho, en general a toda mi familia, por estar ahí siempre para apoyarme.

A mi padre, que aunque no esté cerca me apoya y se preocupa por mí.

A las chicas que conocí cuando empezaba mis estudios y más tarde consideré amigas, porque aprendí a confiar en ellas y de las que recibí apoyo y cariño: Adre, Yady, Mailin son muy buenas amigas al aguantar mis pesadeces.

A los amigos que conocí en la Universidad y que siempre voy a recordar con cariño los muchos momentos que compartimos: .

A todos mis amigos de la secundaria, del PRE y de la Universidad en general.

A mis vecinos y amigos por quererme tanto y preocuparse por mí.

A todos los profesores que me ayudaron en mi preparación profesional

A mi tutor Jorge Daniel (El Chino) por quiarme durante el desarrollo de la tesis.

A los compañeros del proyecto de quienes aprendí mucho.

Declaración de autoría

Declaro que soy la única autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas para que haga el uso que estime pertinente con el mismo.

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

Autora: Yisel Tornés López

Tutor: Jorge D. Olivares

Opiniones y avales

Resumen

La calidad del software es un aspecto que en la actualidad se tiene muy en cuenta en la producción de los mismos. Se pretende que los productos lleguen a mano de los usuarios finales con la mínima probabilidad de tener errores, aunque el software nunca es perfecto. El producto PRIMICIA, Plataforma de Televisión Informativa, provee un canal de teletexto a una red de televisión que tiene como objetivo la transmisión de noticias realizando una labor informativa-formativa, en el mismo se enfoca el presente trabajo para diseñar una estrategia de pruebas, como proceso de vital importancia para asegurar y garantizar en gran medida el buen desempeño de este software.

Para ello se diseñó un plan de pruebas, casos de pruebas y se documentaron los resultados de la aplicación de los mismos, atendiendo las características del sistema desarrollado, lo que permitió la detección de los errores de los casos de uso y de la lógica del programa de la aplicación.

Palabras claves: Calidad, Estrategia, Pruebas de software.

Contenido

Capítulo 1:.....	5
Introducción.....	5
1.1 Calidad del software. Conceptos y definiciones.....	5
1.1.1 Factores que miden la calidad del software	6
1.1.2 Aseguramiento de la calidad y Pruebas de software.	7
1.2 Metodologías de desarrollo de software	7
1.2.1 Proceso Unificado de Modelado (RUP)	9
1.2.2 Extreme Programing (XP).	9
1.2.3 Microsoft Solution Framework (MSF)	10
1.2.4 SCRUM.....	10
1.3 Proceso de pruebas de RUP.....	11
1.3.1 Artefactos del flujo de trabajo de prueba.	11
1.3.2 Actividades fundamentales del flujo de trabajo de prueba.	12
1.4 Plan de pruebas	12
1.5 Modelos de Calidad	13
1.6 Las pruebas en el desarrollo de un software.	15
1.7 Características típicas de las pruebas.....	16
1.8 ¿Cómo aseguro una buena prueba de software?.....	16
1.9 Elementos fundamentales del proceso de prueba.....	17
1.9.1 Niveles de prueba	17
1.9.2 Técnicas de diseño de pruebas.....	19
1.9.2.1 Método de prueba de caja blanca	20
1.9.2.2 Método de prueba de caja negra.....	22
1.10 Herramientas para pruebas automatizadas.....	23
1.11 Estado actual de las pruebas de software.....	24
Conclusiones	25
Capítulo 2	26
2.1 Primicia, Plataforma de Televisión Informativa.	26
2.2 Características a probar.	27
2.3 Plan de prueba general.	27
2.3.1 Objetivos	28
2.3.2 Estrategia de pruebas.	28
2.3.2.1 Recursos para las pruebas.....	28
2.3.2.2 Requerimientos generales.....	29
2.3.2.3 Pasos a seguir en las pruebas o estrategia.	30
2.3.3 Diseño de casos de pruebas de Caja Negra.	31
2.3.4 Diseño de casos de pruebas de Caja Blanca.	48
2.3.4.1 Utilizando Técnica del Camino Básico.....	48
2.3.4.1.1 Caso de prueba CB Redactar Noticia	49
2.3.4.1.2 Caso de prueba CB Publicar Noticia	50
Conclusiones	53
Capítulo 3	54
Introducción	54
3.1 Resultados obtenidos de aplicar las pruebas al módulo Seguridad.....	54
3.2 Resultados obtenidos de aplicar las pruebas al módulo Editorial.....	54

3.3 Resultados obtenidos de aplicar las pruebas al módulo Reporte.....	56
3.4 Resultados obtenidos de aplicar las pruebas al módulo Medias.....	57
3.5 Resultados obtenidos de aplicar las pruebas al módulo Redacción.....	58
3.6 Resultados de las pruebas de funcionalidad y de acceso a datos.....	60
3.7 Resultados Generales.....	61
Conclusiones.....	61
Conclusiones generales.....	62
Recomendaciones.....	62
Bibliografía.....	63

Índice de Tablas

Tabla 1: Requisitos mínimos de hardware.....	29
Tabla 2: Diseño de CN del CU: Autenticar usuario.....	32
Tabla 3: Diseño de CN del CU: Gestionar Usuario.....	34
Tabla 4: Diseño de CN del CU: Gestionar noticia publicada.....	36
Tabla 5: Diseño de CN del CU: Gestionar sección.....	36
Tabla 6: Diseño de CN del CU: Publicar Noticia.....	38
Tabla 7: Diseño de CN del CU: Gestionar Infocinta.....	39
Tabla 8: Diseño de CN del CU: Administrar archivo de noticias.....	40
Tabla 9: Diseño de CN del CU: Generar reporte.....	42
Tabla 10: Diseño de CN del CU: Exportar Reporte.....	42
Tabla 11: Diseño de CN del CU: Administrar archivos multimedias.....	45
Tabla 12: Diseño de CN del CU: Redactar noticia.....	47
Tabla 13: Diseño de CN del CU: Gestionar Noticia Redactadas.....	48
Tabla 14 Resultado de la sección Autenticar usuario.....	54
Tabla 15 Resultado de Gestionar sección.....	55
Tabla 16 Resultado de Gestionar infocinta.....	56
Tabla 17 Resultado de Generar reporte.....	57
Tabla 18 Resultado de Exportar reporte.....	57
Tabla 19 Resultado de Administrar archivos de recursos multimedia.....	58
Tabla 20 Resultado de Redactar noticia.....	60
Tabla 21 Resultado de Gestionar noticias redactadas.....	60

Índice de figuras

Figura 1: Prueba de caja blanca.....	20
Figura 2: Notación de grafos de flujo para las instrucciones: Secuenciales, If, While.....	21
Figura 3: Prueba de caja negra.....	22
Figura 4: Flujo de grafo para el método Redactar noticia.....	49
Figura 5: Flujo de grafo para el método Publicar noticia.....	52

Introducción

Hoy día, la calidad, tanto en la industria como en el comercio y en las organizaciones de servicios y nuevas tecnologías constituye un claro factor diferencial competitivo; dentro de las cuales, la industria del software no es una excepción.

El software se ha convertido en un tema crítico en la sociedad moderna mundial. Todos parecen necesitar mejores productos en el menor tiempo posible y a menor costo. Los métodos intuitivos de desarrollo de software que se usan actualmente son, básicamente, aquellos que los propios individuos “artesanalmente” siguen, los cuales sólo servirán mientras la sociedad pueda tolerar la falta de predicción que ellos acarrearán, es por ello que se hace cada vez más imponente el tema de la calidad de los productos de software que se producen en todo el mundo.

Las empresas dedicadas al desarrollo de software intentan aumentar la calidad de sus productos para evitar los problemas inherentes a sus procesos: plazos y presupuestos incumplidos, insatisfacción del usuario, escasa productividad y la baja calidad obtenida. En vista de las exigencias evidenciadas, ha sido necesario y de suma importancia hacer software de una buena calidad para poder entrar en el mercado. La coyuntura económica de Cuba favorece en este momento el desarrollo de sistemas, no solamente para consumo interno de una organización sino también para su comercialización a terceros e incluso para la exportación. La evolución permanente de la tecnología informática, y consecuentemente, de los principios y técnicas de desarrollo de sistemas, impacta constantemente la formación de los profesionales en informática, entre ellos, quienes construyen software. (2)

La Universidad de la Ciencias Informáticas juega un papel de gran importancia, pues su objetivo principal es producir software para Cuba y para el resto del mundo. Para el cumplimiento de este objetivo la universidad tiene proyectos productivos de diferentes temas; y para lograr un nivel óptimo de calidad en cada uno de ellos, dentro del proceso de aseguramiento de calidad, se hace necesario llevar a cabo una estrategia de pruebas que asegure el cumplimiento de los requisitos de calidad.

Uno de los productos con que cuenta la universidad es: PRIMICIA, Plataforma de Televisión Informativa. La realización de este sistema ha implicado un gran compromiso por parte de cada uno de los miembros del equipo de desarrollo con el cliente, para lograr la satisfacción del mismo en todos los sentidos. Es por tal motivo que se convierte en una necesidad, el desarrollar un proceso de pruebas para mitigar las situaciones de desavenencias entre lo que debía ser y lo que se automatiza antes de poner en uso el producto por el cliente. Por lo que el **problema a resolver** consiste en: ¿Cómo certificar la calidad del producto PRIMICIA, Plataforma de Televisión Informativa con la utilización de

procesos, técnicas y métricas de calidad de avanzada que permitan la posterior utilización del mismo por el cliente?

Para validar la calidad de este producto se propone realizar un adecuado diseño de pruebas, que una vez aplicado permita conocer los errores sobre los cuales trabajar para lograr un sistema más estable y eficiente. Por tanto el **objeto de estudio** de la investigación es: el flujo de trabajo de prueba, como parte del proceso de desarrollo del software; enmarcando el **campo de acción**: proceso de pruebas de calidad del software aplicado al producto PRIMICIA, Plataforma de Televisión Informativa.

Basado en lo anteriormente mencionado el **objetivo general** de este trabajo es:

Diseñar una estrategia de certificación de la calidad para el producto PRIMICIA, que garantice los niveles de calidad del mismo y que correspondan con los requisitos del cliente.

Del cual se derivan los siguientes **objetivos específicos**:

- Analizar los aspectos fundamentales relacionados con la calidad del software.
- Configurar las condiciones existentes para aplicar las pruebas.
- Establecer una estrategia idónea para aplicar pruebas de acuerdo al producto.
- Estructurar las pruebas a aplicar.
- Aplicar las pruebas al producto PRIMICIA.
- Documentar resultados de los errores obtenidos.

Por todo esto se tiene como **hipótesis** que si se aplica un proceso de pruebas bien definido en el producto PRIMICIA se podrá obtener un producto de aceptada calidad.

Para el cumplimiento de los objetivos especificados se trazaron diversas **tareas de investigación**, las cuales son:

1. Analizar las metodologías y tendencias existentes que permitan establecer el estado del arte en el desarrollo de pruebas de software, y los principales conceptos y herramientas relacionados con la calidad del software.
2. Definir el enfoque de prueba y las configuraciones del entorno de prueba.
3. Seleccionar la estrategia más idónea de acuerdo a las características del proceso productivo de la UCI y a las características del producto.
4. Definir los casos de uso de prueba.
5. Estructurar la implementación de las pruebas.
6. Aplicación de las pruebas al producto PRIMICIA.
7. Documentar los resultados obtenidos de la aplicación de las pruebas.

Una vez concluida la investigación se espera obtener los siguientes **resultados**:

- Una estrategia y un plan que organice la estructura de la realización de las pruebas al software PRIMICIA.
- Modelo de prueba.
- Certificación de la calidad del producto PRIMICIA.

Con el propósito de darle solución a los objetivos trazados durante la investigación, se utilizaron varios métodos científicos:

Métodos teóricos: Estos métodos permiten estudiar las características del flujo de trabajo de prueba de un software que no son observadas directamente.

- **Método histórico-lógico:** para investigar la información que se tiene hasta el momento sobre las pruebas y resumir los aspectos fundamentales necesarios para la investigación en curso.
- **Método analítico sintético:** para la división de los aspectos obtenidos sobre el desarrollo de pruebas en partes para un mejor estudio y poder analizar las características de las mismas dentro del flujo de desarrollo del software.
- **Método de modelación:** Este método se utiliza para modelar los casos de prueba, pues es una necesidad que se ejecute esta modelación a la hora de aplicar una prueba.

Métodos empíricos: Estos métodos describen y explican las características fenomenológicas del objeto, representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional.

- **Observación bibliográfica:** es el método utilizado en toda la investigación pues se recoge información de los aspectos tratados en la tesis desde el punto de vista de otros autores así como sus definiciones y resultados para permitir realizar un análisis del arte en ese aspecto.

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

Capítulo 1. Fundamentación teórica: en este capítulo se analizan los aspectos fundamentales que están relacionados con las pruebas de software y se hace referencia a las metodologías de desarrollo y a las pruebas que se realizan para validar la calidad de un software.

Capítulo 2. Diseño y aplicación de las pruebas: se ha de enmarcar en los resultados que se deben obtener del proyecto en específico, en el mismo se hará referencia a la aplicación en general y se definirá la estrategia a seguir, la configuración del entorno en que serán realizadas las pruebas, además se hará referencia a los procedimientos de estas, que servirán de base para el diseño y la ejecución de los casos de pruebas.

Capítulo 3. Resultados: se hará un breve análisis de los resultados obtenidos a raíz de la aplicación de las pruebas al software y se hará un resumen de los principales errores que se encontraron durante las pruebas.

Capítulo 1: Fundamentación Teórica.

Introducción

La etapa de pruebas constituye una de las más importantes durante el ciclo de desarrollo de un producto, pues con ésta se logra depurar un software desde sus inicios, para que el mismo llegue a manos de los clientes cumpliendo todas sus funcionalidades y libre de defectos.

Un proceso de pruebas 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 de forma tal que no afecten el tiempo de duración del proyecto y el costo.

En este capítulo se abordaran los temas fundamentales referidos a la ingeniería de pruebas, conceptos y definiciones expuestos por varios autores, aspectos relacionado con la calidad, los niveles, métodos, técnicas de diseño, procedimientos y plan de pruebas, destacando algunos criterios generales, de los cuales se presentará su definición. Además se hará referencia a las definiciones que propone RUP como metodología de desarrollo para este proceso.

1.1 Calidad del software. Conceptos y definiciones

La calidad del software es una preocupación a la que se dedican muchos esfuerzos. Sin embargo, el software casi nunca es perfecto. Todo proyecto tiene como objetivo producir software de la mejor calidad posible, que cumpla, y si puede supere las expectativas de los usuarios.

¿Qué es la calidad?

La capacidad de un conjunto de características inherentes de un producto, de los componentes del producto o de un proceso para satisfacer los requerimientos impuestos por los clientes (18)

Todos dicen dominar la calidad del software pero la realidad es que cada cual da una definición de calidad desde su punto de vista. No existe entonces una definición estándar ni universal del término, pero algunas instituciones y organismos como ISO, IEEE, SEI, brindan definiciones aceptables, aunque éstas no son homogéneas, dando como resultado que cada profesional o institución utilice su propia versión de calidad, sin embargo se exponen algunas definiciones que se han popularizado por todo el mundo:

“La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”. (4)

“Concordancia del software producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario” (1).

“El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas”. (17)

La calidad del software está dada por la calidad de los procesos usados para desarrollarlo y mantenerlo. (1)

Por tanto; la calidad de software no es más que el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia.

1.1.1 Factores que miden la calidad del software

Un aspecto importante de conocimiento del tema de la calidad del software, son los factores que determinan la misma.

Estos se centran en 3 aspectos importantes de un producto de software:

- Características operativas:
 - **Corrección:** hasta dónde satisface un programa su especificación y logra los objetivos del cliente.
 - **Fiabilidad:** hasta dónde se puede esperar que un programa lleve a cabo la función con la que fue concebido, con la exactitud requerida.
 - **Eficiencia:** la cantidad de recursos informáticos y de código necesarios para que un programa funcione.
 - **Seguridad:** hasta dónde se puede controlar el acceso al software o a los datos por personas no autorizadas.
 - **Facilidad de uso:** aprender a operar los datos de entrada e interpretar las salidas de un programa.
- Capacidad de soportar los cambios:
 - **Facilidad de mantenimiento:** la capacidad para localizar y arreglar un error en un programa.
 - **Flexibilidad:** el esfuerzo necesario para modificar un programa operativo.
 - **Facilidad de prueba:** el esfuerzo necesario para probar un programa para asegurarse de que realiza su función pretendida.
- Adaptabilidad a nuevos entornos:

- **Portabilidad:** la facilidad para transferir el programa de un entorno de sistema hardware y/o software a otro entorno diferente.
- **Reusabilidad (Capacidad de reutilización):** hasta dónde se puede volver a emplear un programa (o partes de un programa) en otras aplicaciones.
- **Interoperabilidad:** el esfuerzo para acoplar un sistema con otro.

1.1.2 Aseguramiento de la calidad y Pruebas de software.

Cuando se habla de calidad de software es necesario tratar 2 conceptos importantes: pruebas de software (Software Testing) y aseguramiento de la calidad (QA), siglas de las palabras en inglés: Quality Assurance, aunque las pruebas de software son una parte del proceso de aseguramiento de calidad.

Las pruebas de software son una fase importante dentro de casi todos los modelos conocidos de ciclo de vida del software y QA se referirá entonces, a asegurar la calidad de los resultados de cada una de las fases del ciclo de vida del software (incluida la de pruebas) y con esto, asegurar la calidad del producto final.

Una vez conocidas estas definiciones es factible analizar algunas diferencias y relaciones entre ambos:

- Las Pruebas de Software utilizarán casos de pruebas para ser ejecutados; en cambio QA de software utilizará los estándares y procedimientos establecidos para cada una de las fases del ciclo de vida del software.
- Ambas permitirán verificar y afirmar la calidad del producto final, el software.
- Ambas definen un conjunto de actividades a realizar dentro del ciclo de vida del software para mejorar y asegurar la calidad del mismo.

1.2 Metodologías de desarrollo de software

Los estándares o metodologías definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software; en palabras sencillas es la base para la edificación de un proyecto de software, la etapa fundamental para lograr los objetivos buscados con dicho proyecto. Si no se sigue ninguna metodología siempre habrá falta de calidad.

La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software. Su objetivo es controlar de manera transparente todo el proceso de desarrollo,

CAPÍTULO 1: Fundamentación teórica

fundamentalmente nos permite producir lo esperado en el tiempo esperado y con el costo esperado, esto es básico pues muchas veces se planea hacer algo que finalmente toma más tiempo de lo planeado, por tanto produce más costos por los que no se logra completar el proyecto según lo esperado.

Las metodologías están organizadas en generaciones las cuales son:

- Desarrollo Convencional (Sin Metodología) se caracteriza por:
 - Los resultados finales son impredecibles
 - No hay forma de controlar lo que está sucediendo en el Proyecto
 - Los cambios organizativos afectan negativamente al proceso de desarrollo
- Desarrollo Estructurado se caracteriza:
 - Programación estructurada
 - Diseño estructurado
 - Análisis estructurado
 - Especificaciones funcionales:
 - ✓ Gráficas
 - ✓ Particionadas
 - ✓ Mínimamente redundantes

Algunos ejemplos de metodologías estructuradas gubernamentales: MERISE (Francia) que surge en el año 1978, SSADM (Reino Unido), entre otras.

- Desarrollo Orientado a Objetos (OO).

La esencia del desarrollo orientado a objetos es la identificación y organización de conceptos del dominio de la aplicación y no tanto de su representación final en un lenguaje de programación. Se eliminan fronteras entre fases debido a la naturaleza iterativa del desarrollo orientado al objeto. Aparece una nueva forma de concebir los lenguajes de programación y su uso al incorporarse bibliotecas de clases y otros componentes reutilizables. Hay un alto grado de iteración y solapamiento, lo que lleva a una forma de trabajo muy dinámica. Es donde aparece la notación OO más popular en la actualidad UML (Unified Modeling Language) y RUP (Rational Unified Process) una metodología OO también y basada en UML.

Actualmente con el creciente desarrollo tecnológico y la aparición de nuevos modelos de producción, han ido apareciendo nuevas metodologías de proceso de desarrollo, estos procesos han ido tomando características marcadas, lo que ha conllevado agruparlos en dos grupos, los llamados "métodos robustos" y los "métodos ágiles", la diferencia más saltante 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.

Ya en estos tiempos se cuenta con una buena cantidad de propuestas que definen este marco de trabajo, escuchamos hablar de metodologías tradicionales o robustas como Rational Unified Process (RUP), Microsoft Solution Framework (MSF) y Métrica surgida en el año 1989 o metodologías ágiles como Extreme Programming, Scrum, Cristal Methods y Feature Driven Development.

1.2.1 Proceso Unificado de Modelado (RUP)

Dentro de las metodologías fuertes la que más se destaca es el Proceso Unificado de Modelado (RUP). RUP sugieren su uso para proyectos nuevos o actualizaciones de sistemas existentes, y recomiendan adoptarlo en forma gradual. En él, se asegura que aunque no hay evidencia amplia que documente sus éxitos, las grandes consultoras suelen recomendarlo incluso para delicados proyectos de misión crítica.

Este es uno de los procesos más generales que existe, está enfocado a cualquier tipo de proyecto así no sea de software, se basa en la documentación generada en cada uno de sus cuatro fases: Intercepción (puesta en marcha), 2. Elaboración (definición, análisis y diseño), 3. Construcción (implementación) y 4. Transición (fin del proyecto y puesta en producción) en las cuales se ejecutarán varias iteraciones (según el tamaño del proyecto).

RUP se basa en Use Case (casos de uso) para describir lo que se tiene y lo que se espera del software, está muy orientado a la arquitectura del sistema a implementarse, documentándose de la mejor manera, basándose en UML (Unified Modeling Language - Lenguaje de Modelado Unificado). Para poder usar RUP antes hay que adaptarlo a las características de la empresa, y medir de manera exacta el tiempo, costos y todos los demás recursos involucrados en el proceso.

1.2.2 Extreme Programing (XP).

XP, se basa en el trabajo orientado directamente al objetivo, basándose para esto en las relaciones interpersonales y en la velocidad de reacción para la implementación y para los cambios que puedan surgir durante el desarrollo del proceso.

Esto se logra, minimizando el riesgo de fallo del proceso manteniendo dentro del equipo a un representante "competente" del cliente, este representante es quién responderá a todas las preguntas y dudas que surjan por parte del equipo de desarrollo durante el proceso, de forma que no se retrase la toma de decisiones.

XP se basa en Usos tories (historias de uso), estas historias las escribe el cliente o su representante dentro de equipo y describen los escenarios claves del funcionamiento del software, a partir de estas se generan los releases (entregas) entre el equipo y el cliente. Estos releases son los que permiten definir las iteraciones necesarias para cumplir con los objetivos, de manera que cada resultado de la iteración sea un programa aprobado por el cliente de quien depende la definición de las siguientes iteraciones.

Una característica saltante de XP, es que el código siempre se produce en parejas, parejas que van cambiando constantemente para lograr así que todo el equipo sepa y pueda modificar según necesidades el código generado, esto logra en el equipo que los integrantes aprendan entre sí y compartan todo el código.

1.2.3 Microsoft Solution Framework (MSF)

La metodología Microsoft Solution Framework (MSF) es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de procesos y de equipos, dejando en un segundo plano las elecciones tecnológicas. MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación. Es una metodología adaptable, escalable, ya que organiza equipos pequeños de entre 3 o 4 personas, así como proyectos que requieren 50 personas o más; flexible, utilizada en el ambiente de desarrollo de cualquier cliente; tecnología agnóstica, porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

1.2.4 SCRUM

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas *sprint*, con una duración de 30 días. El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

1.3 Proceso de pruebas de RUP.

RUP describe cómo planear y ejecutar las pruebas teniendo en cuenta un grupo de artefactos, actividades y trabajadores. Los trabajadores definidos en RUP son:

- 1. Diseñador de pruebas:** Planifica las pruebas. Es responsable de la integridad del modelo de pruebas asegurando que el modelo cumple con su propósito. Del modelo de pruebas deben definir y describir los casos de prueba y los procedimientos de prueba. Además son los responsables de la evaluación de las pruebas de integración y de sistema cuando éstas se ejecuten.
- 2. Ingeniero de componentes:** Son responsables de los componentes de pruebas que automatizan algunos de los procedimientos de pruebas.
- 3. Ingeniero de pruebas de integración:** Son los responsables de realizar las pruebas de integración, las cuales se realizan para verificar que los componentes integrados en una construcción funcionan correctamente juntos, y se derivan a menudo de los casos de pruebas que especifican cómo probar realizaciones de caso de uso de diseño. Debe documentar los defectos en las pruebas de integración.
- 4. Ingeniero de pruebas de sistema:** Son los responsables de realizar las pruebas de sistema sobre los ejecutables. Estas pruebas se llevan a cabo fundamentalmente para verificar las interacciones entre los actores y el sistema. Estas pruebas se derivan a menudo de los casos de pruebas que especifican cómo probar los casos de uso, también se usan para probar el sistema como un todo. Debe documentar los defectos en las pruebas de sistema. [20]

1.3.1 Artefactos del flujo de trabajo de prueba.

El Proceso Unificado de Desarrollo de software plantea 7 artefactos fundamentales para el flujo de trabajo de prueba los cuales deben ser generados dentro de esta fase, pero específicamente en el presente trabajo se generan los artefactos de caso de prueba, plan de prueba y evaluación de las pruebas.

Artefacto modelo de prueba.

Describe fundamentalmente como se prueban los componentes en el modelo de implementación ejecutando pruebas de integración y de sistemas, este artefacto puede describir también como han de ser probados aspectos específicos del sistema, por ejemplo si la interfaz de usuario es utilizable y consistente o si el manual de usuario del sistema cumple con su cometido. (14)

Artefacto caso de prueba.

Especifica una forma de probar el sistema, incluyendo la entrada o resultado con la que se ha de probar y las condiciones bajo las que ha de probarse. (14)

Artefacto componente de prueba.

Automatizan uno o varios procedimientos de pruebas o partes de ellos, estos componentes pueden ser desarrollados utilizando un lenguaje de guiones o un lenguaje de programación, o pueden ser grabados con una herramienta de automatización de pruebas, se utilizan además para probar los componentes en el modelo de implementación, proporcionando entradas de pruebas, controlando y monitorizando la ejecución de los componentes a probar. (14)

Artefacto procedimiento de prueba.

Especifica cómo realizar uno o varios casos de prueba o partes de estos, por ejemplo un procedimiento de prueba puede ser una instrucción para un individuo sobre cómo ha de realizar un caso de prueba manualmente o puede ser una especificación de cómo interaccionar manualmente con una herramienta de automatización de pruebas para crear componentes ejecutables de prueba. (14)

Artefacto defecto.

Es una anomalía del sistema, como por ejemplo un síntoma de un fallo de software o un problema descubierto en una revisión. Un defecto puede ser utilizado para localizar cualquier cosa que los desarrolladores necesitan registrar como síntoma de un problema en el sistema que ellos necesitan controlar y resolver. (14)

Artefacto evaluación de prueba.

Es una evaluación de los resultados de los esfuerzos de prueba, tales como la cobertura del caso de prueba, la cobertura del código y el estado de los defectos. (14)

1.3.2 Actividades fundamentales del flujo de trabajo de prueba.

Las actividades fundamentales que se desarrollan en el flujo de trabajo de pruebas son las siguientes:

- Planificar las pruebas.
- Diseñar las pruebas: Dentro de esta actividad se realizan pruebas de integración y de sistema.
- Implementar prueba.
- Realizar pruebas de integración.
- Realizar pruebas de sistema.
- Evaluar pruebas.

1.4 Plan de pruebas

Un plan de pruebas está constituido por un conjunto de pruebas. Cada prueba debe dejar claro qué tipo de propiedades se quieren probar (corrección, robustez, fiabilidad, amigabilidad,...), cómo se mide

el resultado. Además, un plan de prueba debe especificar en qué consiste la misma (hasta el último detalle de cómo se ejecuta), y debe definir cuál es el resultado que se espera (identificación, tolerancia,...).

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

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

Estas mismas ideas suelen agruparse, especificando qué debe contener un caso de prueba:

- Identificación y objetivo.
- Descripción detallada de las entradas.
- Modo de ejecución.
- Descripción detallada de las salidas esperadas.

Cada uno de esos puntos debe quedar perfectamente documentado.

1.5 Modelos de Calidad

La premisa que indica: “La Calidad de un producto está determinada por la calidad del proceso que se utiliza para desarrollarlo y mantenerlo”, es hoy la que marca la diferencia. Las empresas orientadas a la calidad en sus productos, trabajan fuertemente en la mejora de sus prácticas, definiendo e implementando procesos disciplinados que facilitan el cumplimiento de los objetivos del proyecto, estableciendo un marco de trabajo. (5)

La tendencia de buscar sistemas de calidad con el objetivo de imponerlos no es ajena a las empresas de nuestra industria de software siendo lo más usual los sistemas de calidad relacionados con el estándar de ISO 9000. Para la industria del software los estándares relevantes son:

- *ISO 9001*: este es un estándar que describe el sistema de calidad utilizado para mantener el desarrollo de un producto que implique diseño.
- *ISO 9000-3*: este es un documento específico que interpreta el ISO 9001 para el desarrollador de software.
- *ISO 9004-2*: este documento proporciona las directrices para el servicio de facilidades del software como soporte de usuarios.

CAPÍTULO 1: Fundamentación teórica

Sin embargo, la opinión que se tiene de ellos y el uso que se les ha dado ha generado una extensa polémica específicamente en la industria de software. Por estas razones han surgido otros esquemas de certificación como el Tick IT (Esquema Británico de Certificación) y otros modelos de calidad y mejora de procesos como el ISO/SPICE (Software Process Improvement and Capability Determination) y CMM (Modelo de la capacidad de madurez de Estados Unidos). [6]

La definición misma de “modelo” indica que es una “idealización de la realidad utilizada para plantear un problema”. A partir de allí, nunca podría ser bien implementado si el mismo no es interpretado y adaptado inteligentemente a las necesidades de la organización. En particular, el modelo CMM (Capacity Maturity Model), seleccionado como guía para la realización de este producto, describe un marco de referencia para el desarrollo y mantenimiento de software en las organizaciones así como la contratación de software. CMM constituye un modelo en el que el mejoramiento de los procesos de software es implementado incrementalmente. Este modelo puede ser adaptado y moldeado a una particular organización pero requiere que se impongan los procedimientos a realizar.

CMMI actualmente se usa en la mayoría de las empresas y organizaciones productoras de software en el mundo y se han identificado varios modelos CMMI, en función de las áreas que integran.

Este modelo organiza los pasos para evolucionar los procesos de software en cinco niveles. Estos son:
Inicial: Las organizaciones se caracterizan por tener procesos caóticos, sus procesos son impredecibles y poco controlados, de manera general se dice que todas las empresas se encuentran en este nivel.

Repetible: La gestión de la ingeniería de software es utilizada para establecer procesos básicos de gestión de proyectos, para controlar costo, planificación y calidad del producto.

Definido: Existen procesos de software estándar y documentados para la gestión y la ingeniería de software. Todos los proyectos usan un proceso estándar de la organización para desarrollar y mantener software.

Gestionado: Los productos y los procesos son medidos y manejados cuantitativamente.

Optimizado: Se caracteriza por la mejoría cuantificable y continua de los procesos. Lo anterior es logrado por la retroalimentación de los procesos y por el pilotaje de nuevas ideas y tecnologías.

CMM se basa en el concepto de madurez de un proceso. La madurez de un proceso depende de cuán bien definido y completo estén los procesos y si ellos pueden, realísticamente ser usados para manejar un proyecto de software. La madurez también depende de cuán bien los procesos permitan a los gerentes de software definir las salidas del software y ajustar el plan, de manera que el proyecto sea controlado.

El sistema de calidad basado en CMM es más detallado y suministra un procedimiento para mejorar los procesos de software de forma incremental. Mientras que ISO 9000 centra la atención en la relación

cliente-suministrador para reducir el riesgo del cliente en seleccionar un suministrador CMM centra la atención en el suministrador para mejorar sus procesos de software internos. ISO9000 es un documento más abstracto que CMM, este último describe los procesos de software en detalle. Existe una vinculación entre el modelo CMM e ISO de manera que basta estar en el nivel 3 de CMM para alcanzar la certificación de las Normas ISO.

1.6 Las pruebas en el desarrollo de un software.

Las pruebas de software o Software testing como también se les conoce, pueden ser definidas como:

«Una actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas, se observan o almacenan los resultados y se realiza una evaluación de algún aspecto del sistema o componente.» (14)

El objetivo de estas es asegurar que el software cumpla con las especificaciones requeridas y eliminar los posibles defectos que este pudiera tener, la experiencia parece indicar que donde hay un defecto hay otros, es decir, la probabilidad de descubrir nuevos defectos en una parte del software es proporcional al número de defectos ya descubierto.

Un proceso de prueba será exitoso cuando encuentre errores, los que no son siempre fruto de la negligencia del programador; por lo que de forma general las pruebas deben centrarse en 2 objetivos:

- Probar si el software no hace lo que debe hacer.
- Probar si el software hace lo que no debe hacer, es decir, si provoca efectos secundarios adversos.

A pesar de que las pruebas no pueden asegurar la ausencia de defectos, ya que sólo se puede demostrar que existen defectos en el software, pues este es hecho por humanos, son parte fundamental antes de entregar el software final.

Probar exhaustivamente un software es imposible, pues es imposible evaluar todas las posibilidades. Es por esto que el testing debe apoyarse en metodologías generales que revisan los aspectos más fundamentales que debe considerar todo proceso de pruebas. Debido a esta complejidad actualmente se cuentan con una gran cantidad de software diseñado exclusivamente para la etapa de pruebas, incluyendo la gestión del proceso de software testing, la administración y seguimiento de errores, la administración de los casos de prueba, automatización de pruebas etc.(8)

No obstante, las pruebas de software son elementos críticos para determinar la calidad del software. La importancia de los costos asociados a los errores promueve la definición y aplicación de un proceso de pruebas minuciosas y bien planificadas. Las pruebas permiten validar y verificar el software, entendiendo como validación del software el proceso que determina si el software satisface los

requisitos, y verificación como el proceso que determina si los productos de una fase satisfacen las condiciones de dicha fase.

1.7 Características típicas de las pruebas.

- La realización de controles periódicos, normalmente coincidiendo con los hitos de los proyectos o la terminación de documentos. Estos controles pretenden una evaluación de la calidad de los productos generados (especificación de requisitos, documentos de diseño, etc.) para poder detectar posibles defectos cuanto antes. Sin embargo, todo sistema o aplicación, independientemente de estas revisiones, debe ser probado mediante su ejecución controlada antes de ser entregado al cliente. Estas ejecuciones o ensayos de funcionamiento, posteriores a la terminación del código del software, se denominan habitualmente pruebas.
- Las pruebas se deben realizar, en el entorno en el que se utilizará el sistema, lo que incluye el personal que lo maneja.
- La etapa de pruebas no debe ser posterior a la confección de un programa, tiene que ser paralela a la programación.
- Las pruebas no comienzan formalmente hasta que un número mínimo predeterminado ha instalado el software. En particular, usuarios que tardan en comenzar, generalmente nunca lo hacen y ponen en peligro todo el proceso.

1.8 ¿Cómo aseguro una buena prueba de software?

Al realizar una prueba se debe tratar de desarrollar lo mejor posible; pues de esto depende poder corregir todos los posibles errores del software por lo que se exponen las características para considerar una buena prueba de software:

- Una buena prueba tiene una alta probabilidad de encontrar un error.
- Una buena prueba no debe ser redundante. Uno de los objetivos de las pruebas es «*encontrar el mayor número de errores con la menor cantidad de tiempo y esfuerzo posibles*», por lo cual no se deben diseñar casos de prueba que tengan el mismo propósito que otros sino que se debe buscar diseñar el menor número de casos de prueba que permitan probar adecuadamente el software y que permitan optimizar los recursos.
- *Una buena prueba debería ser la mejor de la cosecha.* La limitación en tiempo y recursos puede impedir que se ejecuten todos los casos de prueba de un grupo de pruebas similares por lo cual en estos casos se debería seleccionar la prueba que tenga la mayor probabilidad de descubrir errores.

- *Una buena prueba no debería ser ni demasiado sencilla ni demasiado compleja.*

1.9 Elementos fundamentales del proceso de prueba

1.9.1 Niveles de prueba

La prueba es aplicada con diferentes objetivos y en disímiles escenarios o niveles de prueba, los que deben fluir de forma orgánica y ascendente, se considera que deben realizarse todos para garantizar la calidad de forma continua e incremental.

Se distinguen los siguientes niveles de pruebas:

- **Prueba de Unidad:** Se centra en el proceso de verificación en la menor unidad del diseño del software: el componente o módulo. (Pressman, 2005).

De modo que una prueba de unidad o una prueba unitaria, es una forma de probar el correcto funcionamiento de un módulo de código. Esta sirve para asegurar que cada uno de ellos funcione correctamente por separado.

- **Prueba del sistema:** Las pruebas de sistema buscan discrepancias entre el programa y sus objetivos o requerimientos, enfocándose en los errores hechos durante la transición del proceso al diseñar la especificación funcional. Esto hace a las pruebas de sistema un proceso vital, ya que en términos del producto, número de errores hechos, y severidad de esos errores, es un paso en el ciclo de desarrollo generalmente propenso a la mayoría de los errores. Las pruebas de sistema tienen como objetivo ejercitar profundamente el sistema comprobando la integración de la información globalmente, verificando el funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen y con el resto de sistemas de información con los que se comunica.

Son pruebas de integración del sistema de información completo, y permiten probar el sistema en su conjunto y con otros sistemas con los que se relaciona para verificar que las especificaciones funcionales y técnicas se cumplen. Dan una visión muy similar a su comportamiento en el entorno de producción.

En este nivel son aplicados de acuerdo a una especificación tipos de pruebas como:

- **Prueba de recuperación:** es una prueba del sistema que fuerza el fallo del software de muchas formas y verifica que la recuperación se lleva a cabo apropiadamente.
- **Prueba de seguridad:** Consisten en verificar los mecanismos de control de acceso al sistema para evitar alteraciones indebidas en los datos.[]

CAPÍTULO 1: Fundamentación teórica

- **Prueba de resistencia:** Consisten en comprobar el funcionamiento del sistema en el umbral límite de los recursos, sometiéndole a cargas masivas. El objetivo es establecer los puntos extremos en los cuales el sistema empieza a operar por debajo de los requisitos establecidos.
- **Prueba de rendimiento:** Consisten en determinar que los tiempos de respuesta están dentro de los intervalos establecidos en las especificaciones del sistema.
- **Prueba funcionalidad:** Pruebas fijando su atención en la validación de las funciones, métodos, servicios, caso de uso.
- **Prueba de Usabilidad:** Prueba enfocada a factores humanos, estéticos, consistencia en la interfaz de usuario, ayuda sensitiva al contexto y en línea, asistente documentación de usuarios y materiales de entrenamiento.
- **Pruebas de soportabilidad:** prueba enfocada a asegurar la instalación y funcionamiento en diferentes configuraciones de hardware y software.
- **Prueba de stress:** se proponen encontrar errores debidos a recursos bajos o completitud de recursos. El objetivo de esta prueba es investigar el comportamiento del sistema bajo condiciones que sobrecargan sus recursos.
- **Prueba de validación:** Son las pruebas realizadas sobre un software completamente integrado para evaluar el cumplimiento con los requisitos especificados.
- **Prueba de integración:** El objetivo de las pruebas de integración es verificar el correcto ensamblaje entre los distintos componentes una vez que han sido probados unitariamente con el fin de comprobar que interactúan correctamente a través de sus interfaces, tanto internas como externas, cubren la funcionalidad establecida y se ajustan a los requisitos no funcionales especificados en las verificaciones correspondientes.

Los tipos fundamentales de integración son los siguientes:

- **Integración incremental:** se combina el siguiente componente que se debe probar con el conjunto de componentes que ya están probados y se va incrementando progresivamente el número de componentes a probar.
- **Integración no incremental:** se prueba cada componente por separado y posteriormente se integran todos de una vez realizando las pruebas pertinentes.
- **Prueba de aceptación:** El objetivo de las pruebas de aceptación es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento.

Las pruebas de aceptación son definidas por el usuario del sistema y preparadas por el equipo de desarrollo, aunque la ejecución y aprobación final corresponden al usuario.

La validación del sistema se consigue mediante la realización de pruebas de caja negra que demuestran la conformidad con los requisitos y que se recogen en el plan de pruebas, el cual define las verificaciones a realizar y los casos de prueba asociados. Dicho plan está diseñado para asegurar que se satisfacen todos los requisitos funcionales especificados por el usuario teniendo en cuenta también los requisitos no funcionales relacionados con el rendimiento, seguridad de acceso al sistema, a los datos y procesos, así como a los distintos recursos del sistema.

- **Prueba del desarrollador:** Prueba diseñada e implementada por el equipo de desarrollo. Estas pruebas pueden realizarse cruzando los programadores, analistas, u otro rol, de forma tal que no solo sea la persona la que revise su propio trabajo, pues generalmente una persona ajena es la que más errores puede detectar. Tradicionalmente estas pruebas han sido consideradas solo para la prueba de unidad, aunque en la actualidad en algunos casos pueden ejecutar pruebas de integración. Se recomienda que estas pruebas cubran más que las pruebas de unidad.

1.9.2 Técnicas de diseño de pruebas

Las estrategias de pruebas del software integran las técnicas de diseño de casos de prueba en una serie de pasos planificados que dan como resultado una correcta construcción del software, con tal grado de confianza que se detectarán los errores existentes en él.

Un aspecto primordial de esta parte es la definición de los casos de prueba.

Pressman define los **casos de pruebas** como “un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo en particular. (5)

Para este diseño de casos de prueba se identifican 3 enfoques de pruebas:

- El enfoque funcional o de caja blanca consiste en centrarse en la estructura interna (implementación) del programa para elegir los casos de prueba.
- El enfoque estructural o de caja negra consiste en estudiar la especificación de las funciones, la entrada y la salida para derivar los casos.
- El enfoque aleatorio, consistente en utilizar modelos (en muchas ocasiones estadísticos) que representen las posibles entradas al programa para crear a partir de ellos los casos de prueba.

1.9.2.1 Método de prueba de caja blanca

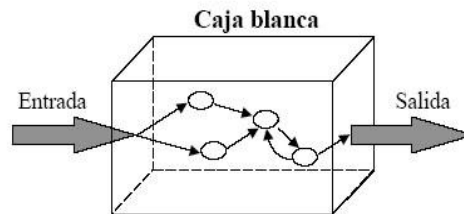


Figura 1: Prueba de caja blanca

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

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

Algunos métodos empleados en las pruebas de caja blanca son los siguientes:

- **Prueba del camino básico:** es una técnica que permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un *conjunto básico* de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizarán que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa. Aspectos claves de este tema:
 - *Grafo de flujo o grafo del programa:* representa el flujo de control lógico de un programa y se utiliza para trazar más fácilmente los caminos de éste. (Cada nodo representa una o más sentencias procedimentales y cada arista representa el flujo de control).

Para construir el grafo se debe tener en cuenta la notación para las instrucciones.

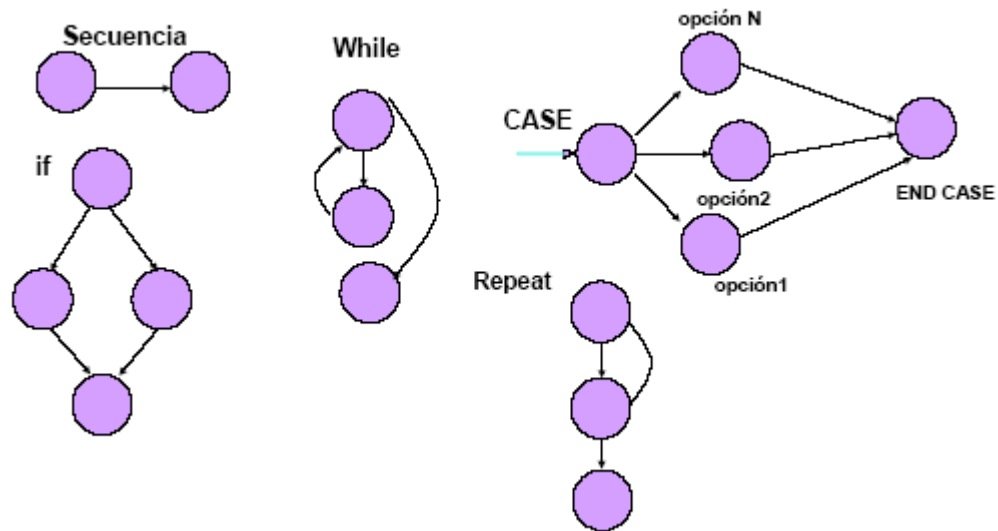


Figura 2: Notación de grafos de flujo para las instrucciones: Secuenciales, If, While

- *Complejidad ciclomática:* es una métrica de software que proporciona una medición cuantitativa de la complejidad lógica de un programa. Cuando se usa en el contexto de las pruebas, el cálculo de la complejidad ciclomática representa el número de caminos independientes del conjunto básico de un programa. Esta medida ofrece al probador de software un límite superior para el número de pruebas que debe realizar para garantizar que se ejecutan por lo menos una vez cada sentencia.

Se calcula:

1ra vía $V(G) = \text{Número de Regiones}$.

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

3ra vía $V(G) = \text{Número de Nodos Predicados} + 1$

- *Camino independiente:* cualquier camino del programa que introduce, por lo menos, un nuevo conjunto de sentencias de proceso o una nueva condición.
- **Prueba de la estructura de control:** dentro de éste tipo de prueba se contempla el método del camino básico mencionado anteriormente pero además existen otras pruebas asociadas que permiten ampliar la cobertura de la prueba y mejorar su calidad. Estas son:
 - **Prueba de condición:** es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en el módulo de un programa. Algunos conceptos empleados alrededor de esta prueba son los siguientes:

- **Condición simple:** es una variable lógica o una expresión relacional ($E_1 < operador - relacional > E_2$).
 - **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:** selecciona caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa.
- **Prueba de bucles:** es una técnica que se centra exclusivamente en la validez de las construcciones de bucles (bucles simples, anidados, concatenados y no estructurados).

1.9.2.2 Método de prueba de caja negra

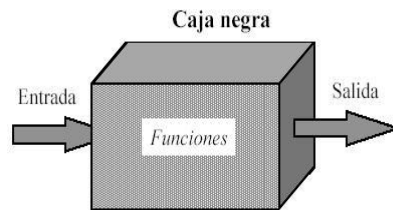


Figura 3: Prueba de caja negra

Las pruebas de caja negra se centran en los requisitos funcionales del software. Es decir, la prueba de caja negra permite obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Se trata de un enfoque que intenta descubrir diferentes tipos de errores que no se encuentran con los métodos de caja blanca.

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

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

- **Métodos de prueba basados en grafos:** en este método se debe entender los objetos (objetos de datos, objetos de programa tales como módulos o colecciones de sentencias del lenguaje de programación) que se modelan en el software y las relaciones que conectan a estos objetos. Una vez que se ha llevado a cabo esto, el siguiente paso es definir una serie de pruebas que verifiquen que todos los objetos tienen entre ellos las relaciones esperadas.
- **Partición equivalente:** método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.
- **Análisis de valores límite:** los errores tienden a darse más en los límites del campo de entrada que en el centro. Por ello, se ha desarrollado el análisis de valores límites (AVL) como técnica de prueba. El análisis de valores límite lleva a una elección de casos de prueba que ejerciten los valores límite.
- **Prueba de la tabla ortogonal:** hay aplicaciones donde el número de parámetros de entrada es pequeño y los valores de cada uno de los parámetros está claramente delimitado. Cuando estos números son muy pequeños (por ejemplo, 3 parámetros de entrada tomando 3 valores diferentes), es posible considerar cada permutación de entrada y comprobar exhaustivamente el proceso del dominio de entrada. En cualquier caso, cuando el número de valores de entrada crece y el número de valores diferentes para cada elemento de dato se incrementa, la prueba exhaustiva se hace impracticable.
- **Adivinando el error:** dado un programa particular, se conjetura, por la intuición y la experiencia, ciertos tipos probables de errores y entonces se escriben casos de prueba para exponer esos errores. Es difícil dar un procedimiento para esta técnica puesto que es en gran parte un proceso.

1.10 Herramientas para pruebas automatizadas.

Existen muchas herramientas utilizadas para desarrollar las pruebas de forma automática con diferentes objetivos y en diferentes escenarios o sistemas operativos, algunas de las que se relacionan a continuación:

Junit: Herramienta especialmente diseñada para implementar y automatizar la realización de pruebas de unidad en Java.

Selenium es una herramienta de Software Libre para pruebas de aplicaciones Web. Las pruebas de Selenium se ejecutan directamente en un navegador y facilitan las pruebas de compatibilidad en navegadores, también como pruebas funcionales de aceptación de aplicaciones Web. Esta herramienta posee un ambiente de desarrollo llamado Selenium IDE, este facilita el registro de pruebas de aceptación y su depuración.

PHPUnit es una herramienta para la ejecución de pruebas al desarrollo de programas/sistemas efectuados específicamente en PHP

Una usada por el equipo de aplicación de pruebas en la Universidad es:

JMeter: Herramienta desarrollada en java. Permite realizar Pruebas de Rendimiento y Pruebas Funcionales sobre Aplicaciones Web. El JMeter puede generar un caso de prueba a través de una navegación de usuario. Es destacada por su versatilidad, estabilidad y por ser de uso gratuito.

1.11 Estado actual de las pruebas de software.

La calidad no ha sido controlada siempre de la forma en que se suele hacer hoy, ya que es un proceso en constante cambio y perfeccionamiento. Es bien sabido que aplicaciones que fallan generan riesgos al negocio, muchas empresas no prueban lo suficiente para garantizar sus aplicaciones. El 90% de estas no realizan pruebas dado la baja disponibilidad de tiempo y recursos. En la actualidad, en nuestro país se está prestando gran interés debido a las exigencias de la Industria del software y a la necesidad de entrar al mercado con productos de buena calidad. Siempre tratando de evitar problemas que frecuentemente se presentan en una organización inmadura como:

- Procesos software normalmente improvisados.
- Si se han especificado, no se siguen rigurosamente.
- Planes y presupuestos excedidos sistemáticamente, al no estar basados en estimaciones realistas.
- No existen bases objetivas para juzgar la calidad del producto.
- Cuando los proyectos están fuera de plan, las revisiones o pruebas se recortan o eliminan.

Estas son algunas cifras significativas que se tienen:

- El 90% de los proyectos no alcanzan los objetivos.
- El 40% fracasan por completo.
- El 50% no se entregan nunca.
- Gastos de adaptación tecnológica al año.
- Coste de demandas y litigios legales añadidos.

En el mundo informático, los cambios se producen a máxima velocidad, por lo que se requiere de un ritmo acelerado y una calidad probada para ganar espacios en un terreno muy competitivo, en lo que influye de manera positiva y en gran medida las pruebas de software, que realizadas de manera eficiente se puede desarrollar un producto con calidad, evitando así, algunas de las implicaciones que traería no cumplir con los requisitos esperados y es precisamente la insatisfacción del cliente y un producto no reconocido por la falta de la calidad.

Conclusiones

El análisis realizado en este capítulo ha mostrado de manera clara, la importancia de la calidad del software, a lo que contribuyó el abarcamiento de los temas tratados, como lo relacionado al proceso de pruebas en un software, de los niveles en que estas se dividen, así como de los métodos o técnicas que se emplean para realizarlas. Además, de las metodologías y modelos de calidad que rigen un adecuado desarrollo de software, estos aspectos van asegurar un producto con calidad.

Capítulo 2

Diseño y aplicación de las pruebas

Introducción

Anteriormente se analizaron los conceptos fundamentales relacionados a las pruebas de software. Lo planteado constituye una guía para lo expuesto en este capítulo: el diseño y aplicación de pruebas al producto PRIMICIA. En el mismo se precisará un plan de pruebas como base para todo el proceso que se llevará a cabo, así como la estrategia de pruebas a seguir y la configuración del entorno de pruebas, especificando cada uno de los elementos de las pruebas definidas.

2.1 Primicia, Plataforma de Televisión Informativa.

El producto PRIMICIA, Plataforma de televisión informativa provee un canal de teletexto para una red de televisión que tiene como objetivo la transmisión de noticias realizando una labor informativa-formativa en cualquier entidad donde este tipo de sistema sea una necesidad. El sistema se encuentra estructurado en dos subsistemas que se relacionan entre sí y actúan como un todo para brindar un resultado final a través de una red de televisión. El Subsistema de Administración es el responsable de que, a través de él se realice la administración del canal y toda la gestión de las noticias y recursos multimedia. El Subsistema de Transmisión es el encargado de visualizar las noticias y materiales publicados.

A este sistema, con el objetivo de probar su eficiencia se le aplicarán pruebas de caja blanca y pruebas de caja negra, para comprobar la funcionalidad de los módulos:

- Subsistema de Administración
 - Módulo de Seguridad
 - Módulo de Redacción
 - Módulo Editorial
 - Módulo de Administración de Recursos Multimedia
 - Módulo de Gestión del Canal
 - Módulo de Reportes
- Subsistema de Transmisión

El buen funcionamiento de estos módulos es lo que hace posible la transmisión del canal.

2.2 Características a probar.

Las características a evaluar en un software son en dependencia del tipo de producto que sea y de lo que se pretenda probar con la prueba a realizar.

Hay algunas características que siempre se deben tener en cuenta, como por ejemplo, si la aplicación cumple con las funcionalidades requeridas desde el comienzo, si no tiene errores a la hora de ejecutarlo, si es comprensible para el usuario, si se puede usar correctamente con facilidad y si cuenta con una documentación detallada y que corresponda con la aplicación.

También se pueden medir aspectos generales que tienen que ver con el código del programa y permiten conocer si dicho código está bien estructurado y es reutilizable. Por ello es importante comprobar que la aplicación sea fácil de mantener para garantizar que se mantenga activa por mucho tiempo.

2.3 Plan de prueba general.

Un plan de pruebas tiene como objetivo el desarrollo de una hoja de ruta para la realización de las pruebas de sistema. Este debe ser documentado y aprobado por los participantes del mismo. En el plan de pruebas definido para el sistema PRIMICIA, se identifican los elementos que serán probados, los recursos necesarios para hacer las pruebas, así como la estrategia de pruebas que se llevará a cabo para lograr un buen diseño de casos de pruebas que permitan encontrar la mayor cantidad posible de defectos al software en cuestión.

Estas mismas ideas se suelen agrupar diciendo que un caso de prueba consta de 3 bloques de información:

1. El propósito de la prueba
2. Los pasos de ejecución de la prueba
3. El resultado que se espera

Respecto al orden de pruebas, una práctica frecuente es la siguiente:

1. Pasar pruebas de caja negra analizando valores límite, donde se ven las condiciones límite de entrada y de salida.
2. Identificar clases de equivalencia de datos (entrada y salida) y añadir más pruebas de caja negra para contemplar valores normales (en las clases de equivalencia en que estos sean diferentes de los valores límite; es decir, en rangos amplios de valores).

CAPÍTULO 2: Diseño y aplicación de las pruebas

3. Añadir pruebas basadas en "presunción de error". A partir de la experiencia y el sentido común, se aventuran situaciones que parecen proclives a padecer defectos, y se buscan errores en esos puntos.

4. Medir la cobertura de caja blanca que se ha logrado con las fases previas y añadir más pruebas de caja blanca hasta lograr la cobertura deseada. Normalmente se busca una buena cobertura de ramas.

2.3.1 Objetivos

El objetivo principal de aplicar pruebas al software PRIMICIA es ejecutar el programa teniendo en cuenta un conjunto de casos de pruebas que son definidos con anterioridad, con la primordial intención de detectar fallas o errores que puedan existir en el software y corregirlos, de esta manera darle más consistencia y una mayor confiabilidad.

Para la elaboración de los casos de pruebas y su posterior aplicación hay que tener claro que, un buen caso de prueba es aquel que tiene altísima probabilidad de mostrar un error que hasta entonces no se había descubierto y que por tanto los desarrolladores puedan corregirlo.

2.3.2 Estrategia de pruebas.

Una estrategia de prueba del software integra las técnicas de diseño de casos de prueba en una serie de pasos bien planificados que dan como resultado una correcta construcción del software. Describe el enfoque y los objetivos generales de las actividades de prueba, define: niveles, tipos, técnicas (manual o automática) de prueba a utilizar. Además de, ¿qué criterios de éxito y culminación de la prueba serán usados?, y alguna que otras consideraciones especiales afectadas por requerimientos de recursos o que tengan implicaciones en la planificación.

2.3.2.1 Recursos para las 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:

- Sistema Operativo: GNU/Linux NOVA.
- Servidor Web: Apache.
- Servidor de Base de Datos: PostgreSQL.
- Servidor Streaming: VLC.

CAPÍTULO 2: Diseño y aplicación de las pruebas

- Servidor FTP: pro FTP.

Los recursos mínimos del equipamiento tecnológico de la plataforma:

Especificaciones de Hardware

Servidor	Procesador	Memoria RAM	Disco Duro	Tarjeta de Red	Tarjeta de Video
Administración - Transmisión	Pentium IV 2.8 GHz	1 Gb	120 Gb	Ethernet 10/100 Mbs	Capturadora Hauppauge WinTV PVR 350
					Exportadora ATI Radeon X300

Tabla 1: Requisitos mínimos de hardware

2.3.2.2 Requerimientos generales.

En este proceso de pruebas se tienen en cuenta los requerimientos, que dan al traste con la funcionalidad del sistema, pues constituyen condiciones o capacidades que éste debe cumplir y forman parte de la documentación del proyecto.

A continuación se especifican una serie de requisitos que enmarcan la funcionalidad del software, son a modo general, extraídos de cada módulo, para dar una idea de la funcionalidad del sistema y constituyen los elementos a probar:

Módulo de Seguridad

Caso de Uso: Autenticar usuario: se encarga de a los usuarios que interactúan con los procesos del sistema administrativo permitirle el acceso a las actividades desarrolladas por el rol que desempeñan.

Caso de Uso: Gestionar Usuario: su función radica en el registro, modificación y eliminación de los usuarios que van a interactuar directamente con el sistema.

Módulo Editorial

Caso de Uso: Gestionar Noticia Publicada: el actor decide gestionar alguna noticia que ya ha sido publicada. Puede modificarla, retirarla de la publicación, eliminarla o archivarla.

Caso de Uso: Gestionar Sección: se tiene un control de las secciones temáticas que se van a transmitir y el orden de visualización de las mismas.

Caso de Uso: Publicar Noticias: se decide publicar las noticias que han sido redactadas, escoge la sección y selecciona la noticia que desea publicar, revisa la noticia pudiendo efectuar algunas modificaciones como: eliminar pantallas, modificar los textos si el tipo de pantalla es de texto o texto

CAPÍTULO 2: Diseño y aplicación de las pruebas

imagen, cambiar recurso multimedia si la pantalla es de texto-imagen, imagen o video y cambiar el fondo musical, luego le asigna el período de tiempo que se estará mostrando en el canal

Caso de Uso: Gestionar Infocinta: utilizar infocintas para promocionar eventos de última hora o acontecimientos de gran importancia.

Caso de Uso: Administrar archivo de noticia: se encarga de archivar los títulos de las noticias que se han publicado por secciones.

Módulo Reporte

Caso de Uso: Generar reporte: su función radica en generar reportes acerca de la actividad de los redactores del sistema. Además se generan reportes temáticos sobre las noticias publicadas en el canal atendiendo a diferentes criterios.

Caso de Uso: Exportar Reporte: es parte del CU Generar reporte cuando el actor decide exportar la información a un formato digital, se crea un documento PDF en el que aparece el reporte generado por el sistema y se le da la posibilidad al usuario de guardar físicamente este archivo.

Módulo Medias

Caso de Uso: Administrar archivo de recursos multimedia: se encarga de realizar tres procesos fundamentales, puede adicionar, eliminar o modificar recursos, además muestra listados de los recursos multimedia del sistema.

Módulo Redacción

Caso de Uso: Redactar noticia: se función consiste básicamente en redactar una noticia con el título, la sección y el fondo musical deseado.

Caso de Uso: Gestionar Noticia Redactadas: el actor decide gestionar alguna de las noticias que ha redactado y aun no han sido publicadas. Puede modificarla o eliminarlas.

2.3.2.3 Pasos a seguir en las pruebas o estrategia.

Teniendo en cuenta las características del producto PRIMICIA, un sistema para el desarrollo televisivo, que realizará el despliegue en este país, se definió las pruebas a realizar en los escenarios siguientes: Inicialmente, la prueba se centra en cada módulo individualmente, asegurando que funcionan adecuadamente como una unidad, aquí se hace un uso intensivo de las técnicas de prueba de caja blanca, se verá la técnica del camino básico a los casos de prueba diseñados para los casos de uso críticos de los módulos: Editorial, que permite la gestión de las secciones temáticas, las infocintas promocionales y las noticias que ya han sido publicadas, además de permitir que se administre el archivo de noticias del canal y el módulo Redacción, que permite la redacción de las noticias y la

CAPÍTULO 2: Diseño y aplicación de las pruebas

gestión de las que no estén publicadas, estos son los fundamentales para el correcto funcionamiento del sistema.

A continuación, se deben ensamblar o integrar los módulos para formar el paquete de software completo, en este caso específicamente la integración ascendente, dado que los módulos se integran de abajo hacia arriba. Durante la integración, las técnicas que más prevalecen son las de diseño de caso de prueba de caja negra (partición equivalente), probando cada una de las entradas con las salidas esperadas.

Después de que el software se ha integrado, se deben comprobar los criterios de validación, la prueba de validación proporciona una seguridad final de que el software satisface todos los requisitos funcionales, de comportamiento y de rendimiento. Durante la validación se usan exclusivamente técnicas de prueba de caja negra (partición equivalente).

Se realizarán por último pruebas del sistema para verificar que cada elemento encaja de forma adecuada y que se alcanza la funcionalidad y el rendimiento total del sistema, en este caso se realizarán pruebas de seguridad, de control de acceso a datos y funcionalidad.

Se enfocan los diferentes tipos de pruebas en dependencia del número de iteraciones, el tamaño de la iteración y el tipo de proyecto que se está probando.

2.3.3 Diseño de casos de pruebas de Caja Negra.

Con las pruebas de CN se comprueba la funcionalidad del sistema, para el desarrollo de las mismas, se aplica la técnica de partición de equivalencia que resulta efectiva a la hora de comprobar la validez de cada entrada en los diferentes módulos.

Modulo Seguridad

Nombre del CU: Autenticar Usuario

Descripción General: El caso de uso inicia cuando una persona desea acceder al sistema, dando su usuario y contraseña, el caso de uso verifica la existencia de ese usuario, de no ser correcto, envía un mensaje informando que el usuario no es válido, en caso contrario lo lleva a la página de inicio según los privilegios de dicho usuario finalizando así este caso de uso.

Condiciones de Ejecución:

Precondiciones: El actor se encuentra en la página de autenticación del sistema, para ingresar en él.

Secciones a probar en el Caso de Uso:

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

CAPÍTULO 2: Diseño y aplicación de las pruebas

SC "Autenticar usuario"	1: EC 1.1: Usuario correctamente autenticado	El usuario se autentica en el sistema y tiene acceso a las diferentes funcionalidades de este según sus permisos.	El usuario desde la página de autenticación intenta acceder al sistema introduciendo su usuario y contraseña. El sistema verifica que estos datos sean correctos, en caso afirmativo le da acceso al usuario según sus permisos a las funcionalidades correspondientes.
	EC 1.2: Usuario incorrectamente autenticado.	El sistema muestra un mensaje de error si los datos de autenticación son inválidos.	El usuario desde la página de autenticación intenta acceder al sistema introduciendo su usuario y contraseña. El sistema verifica que estos datos sean correctos, en caso negativo, se muestra un mensaje de error informando esta falla.

Tabla 2: Diseño de CN del CU: Autenticar usuario

Nombre del CU: Gestionar Usuario

Descripción General: El caso de uso se inicia cuando el actor necesita realizar operaciones con los usuarios del sistema: adicionar uno nuevo, modificar datos de uno existente, o eliminar alguno; quedando actualizados los usuarios del sistema finalizando el caso de uso.

Condiciones de Ejecución:

El actor se ha autenticado y tiene los permisos necesarios para Gestionar usuario.

Secciones a probar en el Caso de Uso:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
[SC 1: Gestionar Usuario]	EC 1.1: Gestionar Usuario	Por medio de este CU se puede gestionar, por parte de los usuarios autorizados, toda la información y acciones referente a la administración de usuarios en el sistema	El usuario accede a la sección Gestionar Usuario y puede seleccionar las opciones Registrar Usuario y Listado de Usuarios.

CAPÍTULO 2: Diseño y aplicación de las pruebas

[SC 2:Registrar Usuario]	EC Usuario registrado satisfactoriam ente.	2.1:	A través de esta funcionalidad es posible registrar los usuarios del sistema.	Una vez dentro de la sección Registrar Usuario, se introduce en un campo el usuario en cuestión que se desea registrar. El sistema verifica su existencia en el dominio, en caso afirmativo, muestra su nombre completo con los roles que les pueden ser asignados. Le son asignados los roles al usuario, se pulsa un botón para aceptar la acción y se muestra un mensaje de confirmación de éxito de la operación.
	EC Usuario no registrado satisfactoriam ente.	2.2:	A través de esta funcionalidad es posible registrar los usuarios del sistema.	Una vez dentro de la sección registrar Usuario, se introduce en un campo el usuario en cuestión que se desea registrar. El sistema verifica su existencia en el dominio, en caso negativo muestra un mensaje de error informando este hecho.
[SC 3:Listado de usuarios]	EC Listado de usuarios filtrar.	3.1:	Por medio de esta funcionalidad es posible obtener un listado de todos los usuarios del sistema.	Una vez seleccionada la función de Listado de Usuarios dentro del menú de Gestionar Usuario aparece el listado de todos los usuarios presentes en el sistema con el rol que estos desempeñan y la posibilidad para cada uno de ellos de Modificar la información que existe de ellos o Eliminarlos.
	EC Listado de usuarios atendiendo a algún filtro.	3.2:	Por medio de esta funcionalidad es posible obtener un listado de los usuarios del sistema que cumplan con determinada condición de búsqueda.	Una vez seleccionada la función de Listado de Usuarios dentro del menú de Gestionar Usuario aparece el listado de todos los usuarios presentes en el sistema con el rol que estos desempeñan y la posibilidad para cada uno de ellos de Modificarlos o Eliminarlos. Es posible mediante un campo introducir algún filtro que el administrador desee para tener en cuenta en la realización de esta búsqueda, de esta manera la misma se actualiza atendiendo a este.
[SC 4:Modificar Usuario]	EC Modificar los datos del usuario	4.1:	Por medio de esta funcionalidad es posible modificar la información del usuario existente en el sistema.	Una vez seleccionada la función Modificar Usuario dentro del Listado de Usuarios aparece una nueva ventana con los roles que posee el usuario en cuestión y que brinda la posibilidad de adicionarle otros y/o quitarle otros. Se muestra luego de presionar el botón "aceptar" una ventana de confirmación de esta acción y es aceptada también.

CAPÍTULO 2: Diseño y aplicación de las pruebas

	<i>EC 4.2: No modificar los datos del usuario</i>	<i>Por medio de esta funcionalidad es posible cancelar la acción de modificación de alguna información al usuario en cuestión.</i>	<i>Una vez seleccionada la función Modificar Usuario dentro del Listado de Usuarios aparece una nueva ventana con los roles que posee el usuario en cuestión y que brinda la posibilidad de adicionarle otros y/o quitarle otros. Se muestra luego de presionar el botón "aceptar" una ventana de confirmación de esta acción y no es aceptada.</i>
<i>[SC 5: Eliminar Usuario]</i>	<i>EC 5.1: Eliminar un usuario completamente.</i>	<i>Por medio de esta funcionalidad es posible eliminar la información existente del usuario en cuestión del sistema.</i>	<i>Una vez seleccionada la función Eliminar Usuario dentro del Listado de Usuarios aparece una ventana de confirmación de esta acción y es confirmada.</i>
	<i>EC 5.2: No eliminar el usuario.</i>	<i>Por medio de esta funcionalidad es posible cancelar la acción de eliminación de un usuario.</i>	<i>Una vez seleccionada la función Eliminar Usuario dentro del Listado de Usuarios aparece una ventana de confirmación de esta acción y no es confirmada.</i>

Tabla 3: Diseño de CN del CU: Gestionar Usuario

Modulo Editorial

Nombre del CU: Gestionar Noticias Publicadas.

Descripción General: El caso de uso se inicia cuando el actor decide gestionar alguna noticia que ya ha sido publicada. Puede modificarla, retirarla de la publicación, eliminarla o archivarla. El caso de uso termina cuando el actor ha realizado alguna de las operaciones que se mencionaron anteriormente.

Condiciones de Ejecución:

El actor se ha autenticado, tiene los permisos necesarios para Gestionar noticia y hay noticias publicadas.

Secciones a probar en el Caso de Uso: Gestionar noticia publicada

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

CAPÍTULO 2: Diseño y aplicación de las pruebas

SC 1: Gestionar noticia publicada.	EC 1.1: Gestionar noticia publicada.	Muestra una ventana emergente con los campos de la noticia para que el actor haga las modificaciones convenientes.	El flujo se inicia cuando el actor accede a las noticias publicadas donde el sistema le brinda la posibilidad de escoger el estado de la noticia que desea gestionar, luego que el actor escoge el estado el sistema muestra un listado con las noticias publicadas y el actor escoge selecciona una de las acciones a realizar sobre la noticia (Modificar, Sacar del aire, archivar, Eliminar y Visualizar modificaciones de la noticia).
	EC 1.2: Sacar del aire.	Muestra un mensaje para confirmar si se desea sacar del aire o no la noticia.	Luego que el actor selecciona uno de los estados en que se puede encontrar una noticia y la sección temática el sistema muestra el listado de las noticias publicadas y el actor saca del aire la noticia que desee, pasando la misma automáticamente al listado de noticias vencidas.
	EC 1.3: Eliminar noticia publicada.	Muestra un mensaje de confirmación antes de eliminar determinada noticia publicada del sistema.	Luego que el actor selecciona uno de los estados en que se puede encontrar una noticia y la sección temática el sistema muestra el listado de las noticias publicadas del cual el actor selecciona la noticia que desea eliminar, siendo eliminada la misma del sistema.
	EC 1.4: Archivar noticia publicada.	Muestra un mensaje para confirmar si se desea archivar o no la noticia publicada.	Luego que el actor selecciona uno de los estados en que se puede encontrar una noticia y la sección temática el sistema muestra el listado de las noticias publicadas selecciona la noticia que desea archivar, siendo archivada la misma por el sistema.
	EC 1.5: Cancela sacar del aire la noticia.	Permite cancelar la acción de sacar del aire las noticias.	Luego que el actor selecciona uno de los estados en que se puede encontrar una noticia y la sección temática el sistema muestra el listado de las noticias publicadas y el actor selecciona la noticia que desea sacar del aire y cancela la acción.

CAPÍTULO 2: Diseño y aplicación de las pruebas

	<i>EC 1.6: Cancela eliminar noticia.</i>	<i>Permite cancelar la acción de eliminar las noticias.</i>	<i>Luego que el actor selecciona uno de los estados en que se puede encontrar una noticia y la sección temática el sistema muestra el listado de las noticias publicadas del cual el actor selecciona la noticia que desea eliminar y cancela la acción.</i>
	<i>EC 1.7: Cancela archivar noticia.</i>	<i>Permite cancelar la acción de archivar las noticias.</i>	<i>Luego que el actor selecciona uno de los estados en que se puede encontrar una noticia y la sección temática el sistema muestra el listado de las noticias publicadas el actor selecciona la noticia que desea archivar y cancela la acción.</i>

Tabla 4: Diseño de CN del CU: Gestionar noticia publicada

Nombre del CU: Gestión de Secciones

Descripción General: El caso de uso se inicia cuando el actor decide gestionar alguna noticia que ya ha sido publicada. Puede modificarla, retirarla de la publicación, eliminarla o archivarla. El caso de uso termina cuando el actor ha realizado alguna de las operaciones que se mencionaron anteriormente.

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: Gestionar sección

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
<i>SC 1: Gestionar sección.</i>	<i>EC 1.1: Gestionar sección.</i>	<i>Muestra la ventana con todas las secciones de la aplicación.</i>	<i>Luego que el actor accede a la funcionalidad gestionar sección el sistema muestra un listado con las secciones existentes donde el actor podrá ordenar las transmisiones, cambiar el horario en que se mostrarán, habilitar la sección o no y al final presionar el botón de guardar para que el sistema guarde los cambios realizados.</i>

Tabla 5: Diseño de CN del CU: Gestionar sección

Nombre del CU: Publicar noticia.

Descripción General: El caso de uso inicia cuando el actor decide publicar las noticias que han sido redactadas, escoge la sección y selecciona la noticia que desea publicar, revisa la noticia pudiendo efectuar algunas modificaciones como: eliminar pantallas, modificar los textos si el tipo de pantalla es de texto o texto imagen, cambiar recurso multimedia si la pantalla es de texto-imagen, imagen o video

CAPÍTULO 2: Diseño y aplicación de las pruebas

y cambiar el fondo musical, luego le asigna el período de tiempo que se estará mostrando en el canal, terminando de esta forma el caso de uso.

Condiciones de Ejecución:

El actor se ha autenticado y tiene los permisos necesarios para Publicar noticia.

Secciones a probar en el Caso de Uso: Publicar Noticia

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: Publicar Noticia	EC 1.1: Publicar Noticia	Muestra la ventana con todas las secciones de la aplicación para publicar una noticia.	Este flujo comienza cuando el actor accede a la función de Publicar noticias. Es por ello que el sistema muestra un listado de las secciones del canal. El actor selecciona una sección y se muestra un listado con los títulos de las noticias que no han sido publicadas de esa sección y un grupo de opciones que el actor puede realizar: Mostrar, Modificar, Publicar, Archivar o Eliminar y Visualizar modificaciones de la noticia. El actor selecciona una opción.
	EC 1.2: Publicar	Muestra una ventana emergente para entrar fechas de inicio y fin para publicar una noticia.	Este flujo se inicia cuando se muestra una ventana emergente con un formulario para establecer la fecha de inicio y fin de publicación. El usuario entra la fecha de inicio y fin de publicación y presiona el botón Publicar. El sistema publica la noticia quedando lista para ser transmitida en el canal.
	EC 1.3: Archivar	Muestra un mensaje para confirmar si se desea archivar o no la noticia.	Este flujo se inicia cuando Muestra un mensaje de confirmación antes de archivar la noticia. El usuario confirma que desea archivar la noticia y el sistema pasa la noticia al archivo de noticias del sistema para que esta pueda ser reutilizada en otro momento.
	EC 1.4: Eliminar	Muestra un mensaje de confirmación antes de eliminar determinada noticia del sistema.	El flujo comienza cuando el sistema muestra un mensaje de confirmación antes de eliminar la noticia del sistema. Entonces el usuario confirma que desea eliminar la noticia y se elimina la noticia del sistema.
	EC 1.5: Cancela acción archivar	Permite cancelar la acción de archivar la noticia	Luego que el actor selecciona uno de los estados en que se puede encontrar una noticia y la sección temática el sistema muestra el listado de las noticias publicadas el actor decide cancelar la acción de archivar la noticia.

CAPÍTULO 2: Diseño y aplicación de las pruebas

	<i>EC 1.6 Cancela acción Eliminar</i>	<i>Permite cancelar la acción de eliminar</i>	<i>Luego que el actor selecciona uno de los estados en que se puede encontrar una noticia y la sección temática el sistema muestra el listado de las noticias publicadas del cual el actor selecciona la noticia que desea eliminar y cancela la acción.</i>
--	---	---	--

Tabla 6: Diseño de CN del CU: Publicar Noticia

Nombre del CU: Gestionar Infocinta

Descripción General: El caso de uso comienza cuando el actor decide agregar nuevas infocintas, modificarlas, eliminarlas o establecer el orden en que se mostraran en sus secciones correspondientes. El caso de uso termina cuando el actor ha realizado alguna de las actividades mencionadas anteriormente.

Condiciones de Ejecución:

El actor se ha autenticado y tiene los permisos necesarios para Gestionar infocintas

Secciones a probar en el Caso de Uso: Gestionar Infocinta.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
<i>SC 1: Gestionar Infocinta.</i>	<i>EC 1.1: Administrar Infocinta.</i>	<i>Muestra una ventana con todas las infocintas de la aplicación.</i>	<i>Este flujo comienza cuando el sistema le muestra al actor un campo desplegable con todas las secciones temáticas de las infocintas y el actor escoge una de esas secciones, mostrando el sistema el listado de las infocintas con fecha de inicio y fin de publicación y el orden de transmisión y las opciones ordenar, modificar y eliminar Infocinta.</i>
	<i>EC 1.2: Crear Infocinta.</i>	<i>Muestra un formulario para relacionar la infocinta a una sección.</i>	<i>Este flujo comienza cuando el actor accede a la función de Crear Infocinta. Muestra un formulario para relacionar la infocinta a una sección y da la posibilidad de entrar el texto de la infocinta, la fecha de inicio y fin de la publicación y si estará activada o no. El actor llena los campos y presiona el botón guardar quedando registrada la nueva infocinta en el sistema.</i>

CAPÍTULO 2: Diseño y aplicación de las pruebas

<i>EC Ordenar Infocinta.</i>	<i>1.3: Se ordenan las infocintas a conveniencia del actor.</i>	<i>Este flujo se inicia cuando el sistema muestra el listado de las infocintas donde el actor va a ordenar las infocintas a su conveniencia y a guardar los cambios en la base de datos.</i>
<i>EC Modificar Infocinta.</i>	<i>1.4: Se muestra una ventana emergente con la información de la infocinta</i>	<i>El flujo comienza cuando el sistema muestra una ventana emergente con la información de la infocinta: texto, sección temática, Fecha de inicio, fin de publicación y estado, el actor realiza los cambios necesarios y oprime el botón guardar, quedando registrados los cambios en la aplicación.</i>
<i>EC Eliminar Infocinta.</i>	<i>1.5: Muestra un mensaje de confirmación antes de eliminar determinada infocinta del sistema.</i>	<i>El flujo comienza cuando el sistema le muestra al actor una ventana emergente para que confirme que desea eliminar la infocinta, el actor acepta y la infocinta es eliminada de la aplicación.</i>
<i>EC Cancelar Modificar Infocinta.</i>	<i>1.6: Se muestra una ventana emergente con la información de la infocinta</i>	<i>El actor cancela la modificación de la infocinta y el sistema cierra la ventana emergente.</i>
<i>EC Cancelar Eliminar Infocinta.</i>	<i>1.7: Muestra un mensaje de confirmación antes de eliminar determinada infocinta del sistema.</i>	<i>El actor cancela la eliminación de la infocinta.</i>

Tabla 7: Diseño de CN del CU: Gestionar Infocinta.

Nombre del CU: Administrar archivo de noticias.

Descripción General: El caso de uso se inicia cuando el actor accede al Archivo de noticias por la necesidad de reutilizar una noticia que está en el archivo noticioso o desea eliminar noticias archivadas.

Condiciones de Ejecución:

El actor se ha autenticado y tiene los permisos necesarios para Administrar el archivo de noticias.

Secciones a probar en el Caso de Uso: Administrar archivo de noticias

CAPÍTULO 2: Diseño y aplicación de las pruebas

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: Administrar archivo de noticias	EC 1.1: Administrar archivo	Muestra la ventana con todas las noticias archivadas.	Este flujo comienza cuando el actor accede a la función de Administrar Archivo. El sistema muestra el listado de secciones del canal. El actor selecciona una sección y el sistema muestra un listado con el titular de las noticias archivadas en esa sección con las opciones de Mostrar, Publicar y Eliminar, para estas noticias. El usuario elige una acción sobre la noticia.
	EC 1.2: Publicar	Muestra una ventana emergente para entrar fechas de inicio y fin para publicar una	Este flujo se inicia cuando se muestra una ventana emergente con un formulario para establecer la fecha de inicio y fin de publicación. El usuario entra la fecha de inicio y fin de publicación y presiona el botón Publicar. El sistema publica la noticia quedando lista para ser transmitida en el canal.
	EC 1.3: Eliminar	Muestra un mensaje de confirmación antes de eliminar	El flujo comienza cuando el sistema muestra un mensaje de confirmación antes de eliminar la noticia del sistema. Entonces el usuario confirma que desea eliminar la noticia y se elimina la noticia del sistema
	EC 1.4: Cancela la acción Eliminar	Permite cancelar la acción de eliminar	Luego que el actor selecciona uno de los estados en que se puede encontrar una noticia y la sección temática el sistema muestra el listado de las noticias publicadas del cual el actor selecciona la noticia que desea eliminar y cancela la acción.

Tabla 8: Diseño de CN del CU: Administrar archivo de noticias.

Modulo Reporte

Nombre del CU: Generar Reporte

Descripción General: El caso de uso inicia cuando el actor desea conocer el desempeño de sus trabajadores o realizar una búsqueda de noticias atendiendo a diferentes criterios. El sistema le muestra la información que desea según el tipo de búsqueda realizada y le da la posibilidad de exportar e imprimir el reporte generado quedando de esta forma finalizado el caso de uso.

CAPÍTULO 2: Diseño y aplicación de las pruebas

Condiciones de Ejecución:

El actor se ha autenticado y tiene los permisos necesarios para Generar reporte.

Secciones a probar en el Caso de Uso:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
[SC 1: Generar Reporte]	EC 1.1: Generar Reporte	A través de esta funcionalidad pueden ser generados los reportes de redactores o por sección temática con el objetivo de comprobar el trabajo de los trabajadores del canal.	El usuario puede acceder dentro de la sección de Gestión de Reportes a las opciones de Reporte de Redactores o Reporte Temático.
[SC 2: Reporte de Redactores]	EC 2.1: Reporte completo de Redactores.	A través de esta funcionalidad es posible obtener un reporte de todos los redactores del canal.	Una vez dentro de la sección Reporte de Redactores el administrador puede acceder a una lista de los redactores del canal, pudiendo seleccionar algunos de ellos o todos si lo desea y generar un reporte pulsando el botón "Generar reporte" que contendrá todas las noticias publicadas por los redactores seleccionados previamente.
	EC 2.2: Reporte filtrado de de redactores	A través de esta funcionalidad es posible obtener un reporte de los redactores del canal que cumplan con el filtro introducido por el administrador.	Una vez dentro de la sección Reporte de Redactores el administrador puede acceder a una lista de los redactores del canal, que cumplan con los filtros establecidos, pudiendo seleccionar algunos de ellos o todos si lo desea y generar un reporte pulsando el botón "Generar reporte" que contendrá todas las noticias publicadas por los redactores seleccionados previamente.

CAPÍTULO 2: Diseño y aplicación de las pruebas

[SC 3:Reporte Temático]	EC 3.1: Reporte de Noticias atendiendo a los filtros establecidos.	Permite generar un reporte con todas las noticias que cumplan con los filtros que se establecen.	Una vez dentro de la sección Reporte temático, es posible establecer algunos parámetros tales como: sección temática, palabras clave, título de la noticia, fecha de inicio, fecha de fin del periodo sobre el que se desea generar el reporte. Si se encuentran noticias que cumplan con estos filtros estas son mostradas.
	EC 3.2: No se encuentran datos para el reporte.	Permite generar un reporte con todas las noticias que cumplan con los filtros que se establecen.	Una vez dentro de la sección Reporte temático, es posible establecer algunos parámetros tales como: sección temática, palabras clave, título de la noticia, fecha de inicio, fecha de fin del periodo sobre el que se desea generar el reporte. Si no se encuentran noticias que cumplan con estos filtros se muestra un mensaje avisando de este hecho.

Tabla 9: Diseño de CN del CU: Generar reporte.

Nombre del CU: Exportar Reporte

Descripción General: Se inicia como parte del CU Generar reporte cuando el actor decide exportar la información a un formato digital, se crea un documento PDF en el que aparece el reporte generado por el sistema y se le da la posibilidad al usuario de guardar físicamente este archivo.

Condiciones de Ejecución:

Se ha generado el reporte que se desea exportar.

Secciones a probar en el Caso de Uso:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
[SC 1: Exportar Información]	EC 1.1: Se exporta la información a otro formato.	Por medio de este CU es posible que la información obtenida en los reportes sea exportada a formato PDF.	Una vez que el usuario ha generado el reporte que desea consultar, pulsando el botón "Exportar a PDF", puede obtener un documento con encabezado y pie de página además de la información reportada. Se ofrece de esta manera la posibilidad de guardar dicha información digitalmente en el ordenador.

Tabla 10: Diseño de CN del CU: Exportar Reporte.

Modulo Medias

Nombre del CU: Administrar archivos multimedia.

CAPÍTULO 2: Diseño y aplicación de las pruebas

Descripción General: El caso de uso comienza cuando el actor accede a la función de Medias. Aquí se manejan todos los recursos multimedia del sistema, permitiendo, Agregar, eliminar y modificar la información. La música estará organizada por país, género y título y los videos e imágenes por sección temática, palabras claves, fecha y nombre. Al terminar de realizar las operaciones que se deseen en el archivo finaliza el caso de uso.

Condiciones de Ejecución:

El actor se ha autenticado y tiene los permisos necesarios para Gestionar las medias. Existan medias en uso y medias que no estén en uso.

Secciones a probar en el Caso de Uso:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: "Gestión de medias"	EC 1.1: Agregar música con éxito	Agregar recursos media al sistema.	El actor accede a la función de Medias y selecciona la opción Gestión de medias. Dentro de la cual selecciona la opción Agregar media. Posteriormente el sistema muestra un formulario con los tipos de recursos que puede agregar. Después de seleccionado el recurso Música el sistema muestra un formulario con las distintas informaciones que son necesarias para agregarlo. Después de llenados todos los campos del formulario el sistema guarda exitosamente el recurso.
	EC 1.2: Agregar imágenes y video con éxito	Agregar recursos media al sistema.	El actor accede a la función de Medias y selecciona la opción Gestión de medias. Dentro de la cual selecciona la opción Agregar media. Posteriormente el sistema muestra un formulario con los tipos de recursos que puede agregar. Después de seleccionado el recurso el sistema muestra un formulario con las distintas informaciones que son necesarias para agregarlo. Después de llenados todos los campos del formulario el sistema guarda exitosamente el recurso.
	EC 1.3: País no disponible	Introducir un nuevo país al sistema.	El actor accede a la función de Medias y selecciona la opción Gestión de medias. Posteriormente el sistema muestra un formulario con los tipos de recursos que puede agregar. Después de seleccionado el recurso el sistema muestra un formulario con las distintas informaciones que son necesarias para agregarlo. El actor selecciona el botón "Agregar país" para agregar un país no disponible y se muestra una ventana emergente para que sea introducido y se guarda el nombre del mismo. Después de llenados los restantes campos del formulario el sistema guarda exitosamente el recurso multimedia.
	EC 1.4: Género no	Introducir un nuevo	El actor accede a la función de Medias y selecciona la opción Gestión de medias. Posteriormente el sistema

CAPÍTULO 2: Diseño y aplicación de las pruebas

	<i>disponible</i>	<i>género al sistema.</i>	<i>muestra un formulario con los tipos de recursos que puede agregar. Después de seleccionado el recurso el sistema muestra un formulario con las distintas informaciones que son necesarias para agregarlo. El actor selecciona "Agregar género" para introducir un nuevo género no disponible hasta ese instante y se muestra una ventana emergente para introducir el nombre del mismo y para que sea guardado con éxito el nuevo género. Después de llenados los restantes campos del formulario el sistema guarda exitosamente el recurso multimedia.</i>
<i>SC 2: "Listado de medias"</i>	<i>EC 2.1: Listar música exitosamente</i>	<i>Listar los recursos media del sistema.</i>	<i>El actor accede a la función de Medias y selecciona la opción Listar medias. El actor selecciona el recurso Música y se muestra entonces un formulario con criterios de búsquedas específicos al recurso. El actor introduce la información necesaria y presiona el botón Buscar. Se muestran entonces los resultados obtenidos dando la posibilidad de: Modificar, Eliminar o Reproducir. El actor selecciona la opción que desee realizar.</i>
	<i>EC 2.2: Listar imagen/video exitosamente</i>	<i>Listar los recursos media del sistema.</i>	<i>El actor accede a la función de Medias y selecciona la opción Listar medias. El actor selecciona el tipo de recurso (Imagen/Video) y se muestra entonces un formulario con criterios de búsquedas específicos a dicho recurso. El actor introduce la información necesaria y presiona el botón Buscar. Se muestran entonces los resultados obtenidos dando la posibilidad de: Modificar, Eliminar o Reproducir. El actor selecciona la opción que desee realizar.</i>
<i>SC 3: "Modificar la información"</i>	<i>EC 3.1: Modificar la información de Música exitosamente</i>	<i>Modificar la información relacionada con los recursos media existente en el sistema.</i>	<i>El actor selecciona la opción Modificar. El sistema muestra una ventana emergente con un formulario para que sea modificada la información referente a la música. El actor realiza los cambios que desee, presiona el botón Guardar y son guardados todos los cambios realizados.</i>
	<i>EC 3.2: Modificar la información de Imagen exitosamente.</i>	<i>Modificar la información relacionada con los recursos media existente en el sistema.</i>	<i>El actor selecciona la opción Modificar. El sistema muestra una ventana emergente con un formulario para que sea modificada la información referente a la imagen. El actor realiza los cambios que desee, presiona el botón Guardar y son guardados todos los cambios realizados.</i>

CAPÍTULO 2: Diseño y aplicación de las pruebas

SC 4: "Eliminar"	EC 4.1: Eliminar medias sin usar	Eliminar un recurso media del sistema.	El actor selecciona la opción Eliminar. El sistema muestra una confirmación sobre la eliminación del recurso, el usuario confirma su intención de eliminar el recurso y este es eliminado del sistema.
	EC 4.2: Eliminar medias en uso	Eliminar un recurso media del sistema.	El actor selecciona la opción Eliminar. El sistema muestra un mensaje informando que está en uso y manda a cancelar la acción, el usuario cancela la eliminación.
	EC 4.3: Cancelar eliminación	Cancelar la eliminación del recurso media ante un posible error.	El actor selecciona la opción Eliminar. El sistema muestra una confirmación sobre la eliminación del recurso. El usuario cancela dicha eliminación y es cancelada la acción.

Tabla 11: Diseño de CN del CU: Administrar archivos multimedia.

Modulo Redacción

Nombre del CU: Redactar noticia

Descripción General: El caso de uso inicia cuando el actor accede a la función Redactar noticia, el sistema le mostrará un formulario para que entre el título, seleccione la sección y el fondo musical de la noticia y después entrará el cuerpo de la noticia dividido en pantallas que pueden ser de varios tipos (texto, imagen, texto-imagen, video). Concluyendo guarda la noticia finalizando así el caso de uso.

Condiciones de Ejecución:

El actor se ha autenticado y tiene los permisos necesarios para acceder a esta funcionalidad.

Secciones a probar en el Caso de Uso:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: Redactar noticia	EC 1.1: Redactar noticia con éxito	Redactar una noticia.	El actor accede a la función de Redactar noticia, el sistema muestra un formulario para editar la noticia. El redactor selecciona la sección a la que pertenece la noticia y para editar las pantallas, selecciona la opción Editar Pantalla. A continuación, aparecerá un campo para seleccionar una de las opciones: Pantalla texto, Pantalla texto-imagen, Pantalla imagen y Pantalla video. Luego debe presionar el botón Continuar.

CAPÍTULO 2: Diseño y aplicación de las pruebas

	<i>EC 1.2: Redactar noticia seleccionand o música de fondo</i>	<i>Redactar una noticia seleccionando para ella un fondo musical.</i>	<i>El actor accede a la función de Redactar noticia, el sistema muestra un formulario para editar la noticia. El redactor selecciona la sección a la que pertenece la noticia, le agrega un título y posteriormente presiona el botón Buscar, para agregarle un fondo musical. El sistema busca los recursos de multimedia, una vez seleccionado uno, el redactor prosigue a editar las pantallas, presionando el botón Editar Pantalla, el cual mostrará un formulario. A continuación presiona el botón Continuar, aparecerán las opciones: Pantalla texto, Pantalla texto-imagen, Pantalla imagen y Pantalla video, de ellas debe seleccionar una.</i>
	<i>EC 1.3: Redactar noticia sin éxito</i>	<i>Intentar redactar una noticia dejando campos vacíos.</i>	<i>El actor accede a la función de Redactar noticia, el sistema muestra un formulario para editar la noticia. El actor deja todos los campos vacíos y presiona el botón Editar Pantallas. El sistema mostrará mensajes de error, señalando que debe llenar los campos vacíos para poder continuar.</i>
SC 2: Pantalla Texto	<i>EC 2.1: Escribir el texto de la pantalla con éxito.</i>	<i>Escribir el texto de la pantalla, y a continuación finalizar la redacción o continuar la redacción.</i>	<i>El actor selecciona la opción Pantalla texto. El sistema permite entrar el texto y muestra el botón Guardar. El actor escribe el texto y presiona Guardar. El sistema guarda la pantalla y muestra las opciones de Continuar redacción o Finalizar redacción. El actor selecciona una opción. Si selecciona la primera el sistema mostrará un formulario, si selecciona la segunda guardará la noticia.</i>
	<i>EC 2.2: Escribir el texto de la pantalla sin éxito.</i>	<i>Intentar escribir un texto en la pantalla dejando campos vacíos.</i>	<i>El actor selecciona la opción Pantalla texto. El sistema permite entrar el texto y muestra el botón Guardar. El actor deja vacío el campo Texto y presiona el botón Guardar. El sistema mostrará mensajes de error, señalando que debe llenar los campos vacíos para poder continuar.</i>

CAPÍTULO 2: Diseño y aplicación de las pruebas

<p>SC 3: Pantalla texto-imagen</p>	<p>EC 3.1 Seleccionar la imagen y el texto de la pantalla.</p>	<p>Seleccionar la imagen y el texto de la pantalla, y a continuación finalizar la redacción o continuar la redacción.</p>	<p>El actor selecciona la opción Pantalla texto-imagen. El sistema muestra un botón para seleccionar la imagen y varios campos para introducir datos de la misma. El actor presiona el botón Buscar, para seleccionar la imagen, el sistema va a buscar los recursos de multimedia, una vez que seleccione la imagen el actor le escribe un comentario, y debe presionar el botón Guardar. El sistema guarda la pantalla y muestra dos opciones: Continuar redacción o Finalizar redacción. El actor selecciona una opción. Si selecciona la primera el sistema mostrará un formulario, si selecciona la segunda guardará la noticia.</p>
<p>SC 4: Pantalla video</p>	<p>EC 4.1: Seleccionar el video de la pantalla.</p>	<p>Seleccionar el video de la pantalla, y a continuación finalizar la redacción o continuar la redacción.</p>	<p>El actor selecciona la opción Pantalla video. El sistema muestra un campo para el nombre del video y un botón para Buscar el video. El actor presiona el botón Buscar para seleccionar el video. El sistema busca los recursos de multimedia, una vez que el actor seleccione el video, presiona el botón Guardar. El sistema guarda el video y muestra dos opciones: Continuar redacción o Finalizar redacción. El actor selecciona una opción. Si selecciona la primera el sistema mostrará un formulario, si selecciona la segunda guardará la noticia.</p>
<p>SC 5: Pantalla Imagen</p>	<p>EC 5.1: Seleccionar la imagen de la pantalla.</p> <p>EC 5.2 No poder seleccionar la imagen de la pantalla</p>	<p>Seleccionar la imagen de la pantalla, y a continuación finalizar la redacción o continuar la redacción.</p> <p>Intentar seleccionar una imagen para la pantalla dejando campos vacíos.</p>	<p>El actor selecciona la opción Pantalla Imagen. El sistema muestra un formulario para llenar los datos de la imagen, entre ellos un botón para buscarla. El sistema busca en los recursos de multimedia la imagen seleccionada. El actor presiona el botón Guardar. El sistema guarda la imagen y muestra dos opciones: Continuar redacción o Finalizar redacción. El actor selecciona una opción. Si selecciona la primera el sistema mostrará un formulario, si selecciona la segunda guardará la noticia.</p> <p>El actor selecciona la opción Pantalla Imagen. El sistema muestra un formulario para llenar los datos de la imagen, entre ellos un botón para buscarla. El actor deja campos vacíos y presiona el botón Guardar. El sistema mostrará mensajes de error, señalando que debe llenar los campos vacíos para poder continuar.</p>

Tabla 12: Diseño de CN del CU: Redactar noticia.

Nombre del CU: Gestionar Noticia Redactadas

CAPÍTULO 2: Diseño y aplicación de las pruebas

Descripción General: El caso de uso se inicia cuando el actor decide gestionar alguna de las noticias que ha redactado y aun no han sido publicadas. Puede modificarla o eliminarla. El caso de uso termina cuando el actor ha realizado alguna de las operaciones que se mencionaron anteriormente.

Condiciones de Ejecución:

El actor se ha autenticado, tiene los permisos necesarios para Gestionar noticia redactada.

Secciones a probar en el Caso de Uso:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: Gestionar noticia redactada	EC 1.1: Gestionar noticia redactada	Permite modificar, eliminar y visualizar modificaciones de las noticias antes redactadas.	El actor accede a la función de Noticias redactadas. El sistema muestra un listado de las noticias que ya han sido redactadas, con las posibles opciones a escoger. El actor selecciona una de las 3 opciones: modificar, eliminar y visualizar modificaciones.
SC 2: Eliminar	EC 2.1: Eliminar noticia	Permite eliminar una noticia ya redactada.	El actor selecciona la opción Eliminar. El sistema muestra un mensaje de confirmación para eliminar la noticia, dando la posibilidad de Aceptar o Cancelar la operación. El actor confirma la eliminación. El sistema elimina la noticia.
	EC 2.2: Cancelar la eliminación de una noticia	Permite cancelar la acción.	El actor selecciona la opción Eliminar. El sistema muestra un mensaje de confirmación para eliminar la noticia, dando la posibilidad de Aceptar o Cancelar la operación. El actor cancela la eliminación. El sistema cancela la acción.

Tabla 13: Diseño de CN del CU: Gestionar Noticia Redactadas

2.3.4 Diseño de casos de pruebas de Caja Blanca.

2.3.4.1 Utilizando Técnica del Camino Básico.

En el diseño de los caso de prueba de caja blanca para los CU críticos de los módulos de Editorial y Redacción, se siguió la técnica de diseño del camino básico, esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico. La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control.

Para obtener dicho conjunto de caminos independientes se construye el grafo de flujo asociado y se calcula su complejidad ciclomática. Los pasos que se siguen para aplicar esta técnica son:

1. A partir del diseño o del código fuente, se dibuja el grafo de flujo asociado.
2. Se calcula la complejidad ciclomática del grafo.
3. Se determina un conjunto básico de caminos independientes.
4. Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

2.3.4.1.1 Caso de prueba CB Redactar Noticia

Paso 1: Dibujar el grafo de Flujo asociado a partir del código fuente:

```
public function executeRedactarNoticia ()
{
1  $this->getUser () ->setAttribute ('cantidadPantallas', 0);
1  $criterioS =new Criteria ();
1  $criterioS->addAscendingOrderByColumn (SeccionPeer::ORDEN);
1  $secciones=SeccionPeer::doSelect ($criterioS);
1  $arreglo_n = array ();
1  $arreglo_m = array ();
1  $arreglo_n [0]='[Seleccione Sección³n]';
2  foreach ($secciones as $seccion)
    {
        3  $arreglo_n [$seccion->getIdSeccion ()]=$seccion->getNombre ();
    }
4  $this->arreglo_nombre=$arreglo_n;
4  $this->getUser () ->getAttributeHolder () ->remove ('noticia');
4  $this->getUser () ->getAttributeHolder () ->remove ('musica');
4  $this->getUser () ->getAttributeHolder () ->remove ('seccion');
4  $this->getUser () ->getAttributeHolder () ->remove ('titulo');

    4}
}
```



Figura 4: Flujo de grafo para el método Redactar noticia

Paso 2: Cálculo de complejidad ciclomática de Redactar Noticia.

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

Paso 3: Caminos básicos:

CB1: 1-2-3-2-4.

CB2: 1-2-4 Caso en que no existen secciones.

Paso 4: Caso de prueba para el camino básico.

CB1: 1-2-3-2-4

Caso de prueba: Redactar una noticia.

Entrada: Seccion=Laboral, Titulo =prueba, fondo musical=el padrino, tipo de pantalla= texto

Resultado Esperado: Se redacta una noticia de texto

Condiciones: No existe

2.3.4.1.2 Caso de prueba CB Publicar Noticia

```
public function executePublicarNoticia ()
{
    1 $this->getUser () ->setCulture ('es');
    2 if ($this->getRequestParameter ('tipo') == 'noticia')
    {
    3 $this->setlayout ('layoutventana');
    3 $this->getUser () ->setAttribute ('titulo', 'Publicar Noticia');

    3 $this->id=$this->getRequestParameter ('id');
    3 $this->objNot = NoticiaPeer::retrieveByPK ($this->id);

    4 if ($this->getRequest ()->getMethod() == sfRequest::POST)
    {
    5 $objNotPub= new NoticiaPublicada ();
    5 $objNotPub->setFechaPublicacion (date ("d-m-Y"));
    5 $objNotPub->setFechaInicio (str_replace ('/', '-', $this-
>getRequestParameter ('fechaini')));
    5 $objNotPub->setFechaFin (str_replace ('/', '-', $this-
>getRequestParameter ('fechafin')));
    5 $objNotPub->setUsuarioId ($this->getUser () ->getAttribute
('usuario_login'));
    5 $objNotPub->setNoticiaId ($this->id);
    5 $objNotPub->save ();
    6}
    7}

    8 if ($this->getRequestParameter ('tipo') ==
'noticiaArchivada')/*publicar noticia archivada*/
    {

    9 $this->id = $this->getRequestParameter ('id');
    9 $this->setlayout ('layoutventana');
    9 $this->getUser () ->setAttribute ('titulo', 'Publicar Noticia
Archivada');

    9 $this->objNot = NoticiaArchivadaPeer::retrieveByPK ($this->id);

    10 if ($this->getRequest ()->getMethod() == sfRequest::POST)
    {
    11 echo 'OK';
```

CAPÍTULO 2: Diseño y aplicación de las pruebas

```
12     $noticia_desarchivada = new Noticia ();
12     $noticia_desarchivada->setFechaConfeccion ($this->objNot-
>getFechaConfeccion ());
12     $noticia_desarchivada->setTitulo ($this->objNot->getTitulo ());
12     $noticia_desarchivada->setMusicaId ($this->objNot->getMusicaId ());
12     $noticia_desarchivada->setSeccionId ($this->objNot->getSeccionId ());
12     $noticia_desarchivada->setUsuarioId ($this->objNot->getUsuarioId ());
12     $noticia_desarchivada->save ();

12     $idNoticia= $noticia_desarchivada->getIdNoticia ();

12     $criterio= new Criterias ();
12     $criterio->add (PantallaArchivadaPeer::NOTICIA_ARCHIVADA_ID, $this->id);
12     $pantalla= PantallaArchivadaPeer::deselete($criterio);

13     foreach ($pantalla as $obj)
    {
14         $pantallaN = new Pantalla ();
14         $pantallaN->setTexto($obj->getTexto());
14         $pantallaN->setTipo($obj->getTipo());
14         $pantallaN->setVideoId($obj->getVideoId ());
14         $pantallaN->setImagenId($obj->getImagenId());
14         $pantallaN->setOrden($obj->getOrden ());
14         $pantallaN->setNoticiaId($idNoticia);
14         $pantallaN->save ();
    }

15     $this->objNot->delete ();

15     $objNotPub= new NoticiaPublicada ();
15     $objNotPub->setFechaPublicacion (date ("d-m-Y"));
15     $objNotPub->setFechaInicio (str_replace ('/', '-', $this-
>getRequestParameter ('fechaini')));
15     $objNotPub->setFechaFin (str_replace ('/', '-', $this-
>getRequestParameter ('fechafin')));
15     $objNotPub->setUsuarioId ($this->getUser () ->getAttribute
('usuario_login'));
15     $objNotPub->setNoticiaId ($idNoticia);
15     $objNotPub->save ();
16 }
17 }
18 }
```

1. Grafo de Flujo del código fuente del método Publicar noticia.

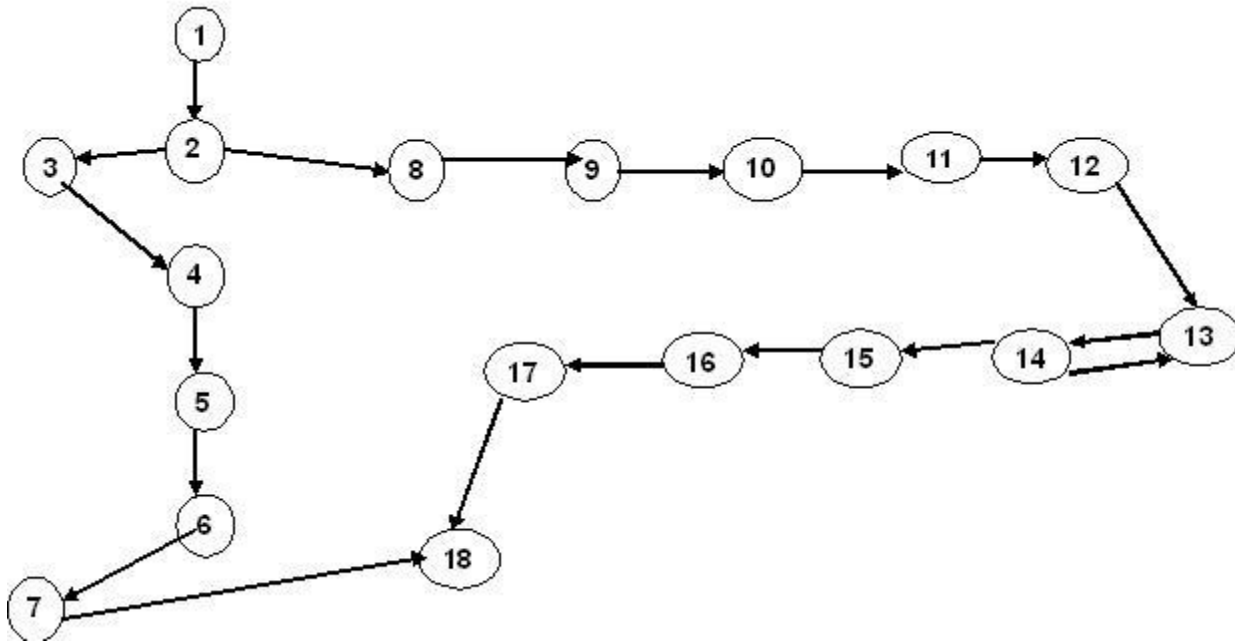


Figura 5: Flujo de grafo para el método Publicar noticia.

2. Cálculo de la complejidad ciclomática:

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

3. Caminos básicos

CB1: 1-2-3-4-5-6-7-18.

CB2: 1-2-8-9-10-11-12-13-14-15-16-17-18

4. Casos de prueba para los caminos básicos.

CB1: 1-2-3-4-5-6-7-18.

Caso de prueba: publicar una noticia.

Entrada: Seccion= Deportes, Noticia=Juegos Deportivos, Fecha de inicio =8/04/2009 15:05,
Fecha de fin= 10/04/2009 12:20

Resultado Esperado: La noticia pasa a estar publicada con el tiempo previsto.

Condiciones: Exista la noticia.

CB2: 1-2-8-9-10-11-12-13-14-15-16-17-18

Caso de prueba: Archivar una noticia.

Entrada: Noticia que se desea archivar

Resultado Esperado: La noticia deseada pasa al archivo de noticias.

Condiciones: Exista noticia.

Conclusiones

En este capítulo se describe de forma clara las características del producto Primicia, el sistema sometido al proceso de pruebas. A partir de esto, se detallaron los módulos y dentro de los mismos las funcionalidades que dan pie al buen desarrollo del software. Para llevar a cabo el diseño y la aplicación de estas pruebas se tienen en cuenta una serie de aspectos que son abordados también en este capítulo como son:

- Elaboración del plan de pruebas.
- Las pruebas se diseñan para comprobar que existen errores, no para demostrar que no existen.
- A partir de esto se determinan las características más importantes a probar.
- Las pruebas de caja negra se basan en la especificación del sistema.
- Las pruebas de código identifican casos de prueba que permiten ejecutar todos los "camino" en un programa, asegurando que todas las sentencias se ejecutan al menos una vez.

Capítulo 3 Resultados

Introducción

En el proceso de pruebas es necesario documentar los resultados obtenidos de la aplicación de las mismas. En este capítulo se realiza un análisis de los errores encontrados al aplicar los diseños de caso de prueba desarrollados para las funcionalidades del sistema, y así corregirlos antes de ser entregado el producto al usuario final.

3.1 Resultados obtenidos de aplicar las pruebas al módulo Seguridad.

Funcionalidad: **Autenticar usuario**

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Sección: "Autenticar Usuario"		<i>El sistema debe mostrar un formulario con dos campos para introducir el usuario y la contraseña. Debe verificar que los datos ingresados por el usuario sean correctos. En ese caso aparecerá la página principal del sistema con las opciones que puede realizar este actor según su rol. En caso que los datos no sean correctos deberá aparecer un mensaje informándole al usuario que no tiene acceso al sistema.</i>	<i>Se muestra el formulario correctamente. Todo funciona bien. No se muestra el mensaje si la contraseña o el usuario son incorrectos.</i>	<i>El botón debe llamarse Entrar según lo descrito.</i>

Tabla 14 Resultado de la sección Autenticar usuario

3.2 Resultados obtenidos de aplicar las pruebas al módulo Editorial.

Funcionalidad: **Gestionar sección.**

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Sección: "Gestionar Sección"		<i>Se debe mostrar un listado con las secciones existentes e información de ellas como orden de transmisión, horario en que se mostrarán y si está o no habilitada. Además se mostrará un botón salvar para guardar todos los cambios que se realicen. El sistema debe permitir hacer modificaciones en el caso del horario de transmisión: Se debe habilitar el campo para establecer el horario de transmisión de la sección y en el caso de Habilitar o deshabilitar:</i>	<i>Las modificaciones se hacen directamente sin mostrar las opciones para modificar, ni se habilitan los campos para hacer esta acción como está especificado en las descripciones. El sistema no valida si las modificaciones hechas están bien. En el campo horario se puede</i>	<i>La palabra sección está sin tilde al igual que algunos de los nombres de las secciones mostradas.</i>

CAPÍTULO 3: Resultados

		<i>Se habilita el campo que tiene esta opción para modificarla.</i>	<i>poner cualquier cosa y no se muestra ningún mensaje informando el error, lo que aparecen son líneas de código.</i>	
<i>Sección: "Modificar orden"</i>		<i>Muestra una ventana emergente con el listado de las secciones en el orden actual que tienen, dando la posibilidad de moverlas para reordenarlas.</i>	<i>Las reestructuraciones se realizan correctamente.</i>	<i>Las imágenes que van en los botones de subir y bajar no se muestran</i>

Tabla 15 Resultado de Gestionar sección

Funcionalidad: **Gestionar Infocinta**

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
<i>Sección "Gestionar Infocintas"</i>		<i>Mostrar las opciones Listado de infocintas y Crear infocintas.</i>	<i>No se muestra ninguna opción</i>	
<i>Sección: "Listado de infocintas"</i>		<i>Mostrar un campo desplegable con las secciones temáticas. Luego de seleccionada una mostrar un listado con el texto y el orden de transmisión de las infocintas existentes en la sección seleccionada, así como opciones para ordenar, modificar y eliminarlas.</i>	<i>Al lado del botón Listar se muestra un mensaje "Esta sección no tiene infocintas" sin haber seleccionado ninguna.</i>	<i>Las secciones temáticas no están escritas correctamente, les faltan las tildes.</i>
<i>Sección: "Ordenar Infocinta"</i>		<i>Muestra una ventana emergente con un listado de las infocintas de la sección seleccionada, por el orden actual que tienen, dando la posibilidad de moverlas para reordenarlas. Al terminar la reestructuración del orden de las infocintas, guardar los cambios y cerrar la ventana.</i>	<i>La reestructuración funciona correctamente pero no se muestra ninguna ventana emergente para realizar esta acción como se especifica en la descripción.</i>	<i>Los botones de subir y bajar no están bien alineados dando un aspecto desorganizado a la página</i>
<i>Sección: "Modificar Infocinta"</i>		<i>Se muestra una nueva ventana con la información de la infocinta seleccionada: Texto, sección temática, Fecha de inicio y fin de publicación y estado, dando la posibilidad de ser modificados.</i>	<i>Esta funcionalidad funciona correctamente</i>	
<i>Sección: "Eliminar Infocinta"</i>		<i>Muestra una confirmación sobre la eliminación de la infocinta.</i>	<i>Se elimina correctamente la infocinta pero no se muestra ningún mensaje confirmando la acción por tanto no se le da la posibilidad al usuario de cancelar esta acción.</i>	

Tabla 16 Resultado de Gestionar infocinta

3.3 Resultados obtenidos de aplicar las pruebas al módulo Reporte.

Funcionalidad: **Generar reporte.**

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Sección "Generar Reporte"		Mostrar dos opciones: Reporte de redactores y Reporte temático	Se mostraron correctamente las dos opciones	En la opción reporte temático la palabra temático está sin tilde y en el menú principal la palabra Gestión también está sin tilde
Sección: "Reporte de redactores"		Mostrar un listado con los nombres y apellidos de los redactores del sistema y un campo para seleccionarlos, se debe mostrar una opción en caso de que desee seleccionarlos todos. Además debe existir un campo para filtrar el contenido de la lista, así como dos campos para definir la fecha de inicio y fin del período que se desea conocer las actividades realizadas por el o los redactores señalados.	No todos los redactores aparecen con su apellido. Existe el campo de filtrar pero no funciona correctamente ya que muestra los redactores cuyo nombre no contiene la subcadena escrita en el campo. La funcionalidad no funciona totalmente pues no se muestra la ventana con el reporte generado del o los redactor(es), además no se muestra ningún mensaje si no se seleccionan los datos necesarios para generar un reporte. Se muestran correctamente las opciones de Exportar o Imprimir el reporte, pero la opción Exportar no está implementada.	
Sección: "Reporte temático"		Mostrar un formulario para realizar la búsqueda de las noticias atendiendo a diferentes criterios: Sección temática, Palabras clave, Título y Fecha de inicio y fin del período del que se desea	Encima de la página de Reporte Temático salen líneas de código. En el campo de la sección temática no aparece ninguna opción para seleccionar.	Las palabras Título, sección y temático están sin tilde. El nombre del botón debería ser solamente Buscar

		generar el reporte.	No está validado para cuando no se introduce ningún criterio de búsqueda ni tampoco para cuando se introducen datos erróneos , además cuando la fecha de inicio es un día mayor al de la fecha de fin ambas con el mismo mes y el mismo año no se le informa al usuario que cometió un error. Se muestran correctamente las opciones de Exportar o Imprimir el reporte, pero la opción Exportar no está implementada.	no Buscar OK.
--	--	---------------------	--	---------------

Tabla 17 Resultado de Generar reporte

Funcionalidad: **Exportar Reporte**

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Sección: "Exportar información"				Esta funcionalidad no está implementada

Tabla 18 Resultado de Exportar reporte

3.4 Resultados obtenidos de aplicar las pruebas al módulo Medias.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
----------------	------------------	--------------------	------------------------	---------------

<p>Sección: "Agregar recurso multimedia"</p>		<p>El sistema debe mostrar un formulario con los tipos de recursos que se pueden agregar: Imagen, Video y Música. Además debe mostrar un formulario con las distintas informaciones que son necesarias para agregar el recurso al sistema, así como un campo para buscar y añadir el recurso físicamente. Para el caso específico de la música debe mostrar dos botones, Agregar país y Agregar género, para el caso en que no existan los deseados por el actor. Si no está disponible el nombre del país, el género al que pertenece la música que va a guardar.</p> <p>Selecciona el botón agregar.</p>	<p>Se muestran correctamente los tipos de recurso que se pueden agregar. En el caso de la música se muestran correctamente los dos botones Agregar país y Agregar género.</p> <p>En el campo país hay países repetidos y otros que no existen al igual que en el campo género.</p> <p>Esta funcionalidad no funciona totalmente ya que no adiciona imágenes ni videos ni música.</p>	<p>Las palabras Título, ubicación, género y país están sin tilde</p>
--	--	--	--	--

Tabla 19 Resultado de Administrar archivos de recursos multimedia

3.5 Resultados obtenidos de aplicar las pruebas al módulo Redacción.

Funcionalidad: **Redactar noticia.**

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
<p>Sección: "Redactar noticia"</p>		<p>El sistema debe mostrar un formulario para que el usuario seleccione la sección, el título y le relacione un fondo musical, así como una opción para editar las pantallas que conformarán la noticia. Para la búsqueda del fondo musical se pone un botón buscar.</p>	<p>Se muestra el formulario correctamente. El botón Buscar no funciona, como no funciona no se pudo probar el botón Reproducir.</p>	<p>Si copio la url para otra página, da un error pero sale la sección de todas formas, por tanto hay que revisar la seguridad. En los nombres de los campos, las palabras sección, música, título aparecen sin tilde.</p>
		<p>Al hacer clic en la opción Editar pantalla. Muestra un formulario para seleccionar el tipo de pantalla que se va a editar.</p>	<p>La opción Editar Pantalla no existe, el botón se llama Nueva Pantalla. Permite escoger el tipo de pantalla correctamente.</p>	
<p>Sección: "Pantalla texto"</p>		<p>El sistema debe mostrar un campo para entrar el texto de la pantalla y el botón Salvar. El sistema guarda la pantalla y muestra las opciones de</p>	<p>Se muestran los campos correctos, pero el botón Salvar no muestra las opciones Continuar o Terminar solamente, sino</p>	<p>En el botón Terminar la Redacción, la palabra Redacción aparece sin tilde.</p>

		<i>Continuar o Terminar la redacción.</i>	<i>que se muestra nuevamente el campo titulo, y salen los botones Buscar, Reproducir y Nueva Pantalla. El botón Continuar y Terminar la Redacción funcionan correctamente.</i>	
<i>Sección: "Pantalla Texto-Imagen"</i>		<i>El sistema debe mostrar un campo donde aparecerá el nombre de la imagen después de que se escoja, un botón para buscar la imagen y otro campo para entrar el texto correspondiente a la pantalla, además se muestra el botón Salvar.</i>	<i>Se muestran los campos descritos, pero la funcionalidad del botón Buscar no sirve, por tanto Visualizar tampoco. El botón Salvar, debido a que no se puede seleccionar ninguna imagen da un error.</i>	
		<i>El sistema guarda la pantalla y muestra las opciones de Continuar o Terminar la redacción. Al seleccionar Continuar, regresa a la acción 4 de la sección Redactar noticia y si escoge Terminar redacción, guarda la noticia.</i>	<i>No se puede revisar estas funcionalidades, pues con el error que da el botón salvar no salen estos campos.</i>	
<i>Sección: "Pantalla imagen"</i>		<i>El sistema debe mostrar un campo donde aparecerá el nombre de la imagen después de que se escoja, un botón para buscar la imagen y otro campo para entrar el texto correspondiente al cintillo de la imagen, además se muestra el botón salvar.</i>	<i>Se muestran los campos descritos, pero la funcionalidad del botón Buscar no sirve, por tanto Visualizar tampoco. El botón Salvar, debido a que no se puede seleccionar ninguna imagen da un error.</i>	
		<i>El sistema guarda la pantalla y muestra las opciones de Continuar o Terminar la redacción. Si oprime Continuar, regresa a la acción 4 de la sección Redactar noticia, si oprime Terminar redacción, guarda la noticia.</i>	<i>No se puede revisar estas funcionalidades, pues con el error que da el botón salvar no salen estos campos.</i>	
<i>Sección: "Pantalla Video"</i>		<i>El sistema debe mostrar un campo donde aparecerá el nombre del video después de que se escoja, un botón para buscar el video, además se muestra el botón Salvar.</i>	<i>Se muestran los campos descritos, además del botón Visualizar que no está descrito. El botón Buscar no funciona, por tanto el Visualizar tampoco. El botón Salvar da un error debido a que no se puede seleccionar ningún video.</i>	

		<i>El sistema debe guardar la pantalla y muestra las opciones de Continuar o Terminar la redacción. Si oprime Continuar, debe regresar a la acción 4 de la sección Redactar noticia y si escoge Terminar redacción, debe guardar la noticia.</i>	<i>No se puede revisar estas funcionalidades, pues con el error que da el botón salvar no salen estos campos.</i>	
--	--	--	---	--

Tabla 20 Resultado de Redactar noticia

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
<i>Sección: "Gestionar noticias redactadas"</i>		<i>El sistema debe mostrar un listado con los títulos de las noticias que han sido redactadas por el usuario que se autenticó, dándole la posibilidad de: Modificar, Eliminar y Ver modificaciones de la noticia.</i>	<i>Se muestra el formulario correctamente. Aparece un campo que no está descrito, Volver a Redactar. Las funcionalidades Modificar y Mostrar, re direccionan correctamente.</i>	<i>Si copio la url para otra página, da un error pero sale la sección de todas formas, por tanto hay que revisar la seguridad.</i>
<i>Sección: "Eliminar"</i>		<i>El sistema debe mostrar un mensaje de confirmación para eliminar la noticia, dando la posibilidad de Aceptar o Cancelar la operación. Si oprime Aceptar, elimina la noticia, si oprime Cancelar cancela la acción.</i>	<i>No muestra el mensaje de confirmación para eliminar la noticia.</i>	

Tabla 21 Resultado de Gestionar noticias redactadas

3.6 Resultados de las pruebas de funcionalidad y de acceso a datos

De modo general la interfaz permite el acceso y navegación a todas las funcionalidades permitidas de acuerdo al rol de cada usuario.

Dentro del marco de pruebas se le hicieron comprobaciones a la Base de Datos específicamente pruebas de volumen, para el llenado voluminoso de la misma se utilizó la herramienta EMS Data Generator 2005 for PostgreSQL, el mismo permite generar datos para una o varias tablas a la vez, además permite definir el rango de valores admisibles para cada campo, así como brinda la posibilidad de generar los datos a través de consultas SQL o de escogerlos por medio de listas de valores. El EMS Data Generator respeta la integridad referencial, ya que genera los datos que provienen de otras tablas para evitar errores. Esto permitió comprobar el comportamiento de la base de datos frente a las consultas más frecuentes que se realicen a la misma, se evidenció la eficiencia de esta y su funcionalidad; donde las inserciones, modificaciones y eliminaciones se ejecutaron correctamente y la

información se actualiza y guarda adecuadamente según los cambios que presupone cada ejecución de las funcionalidades.

3.7 Resultados Generales

En el proceso de ejecución de las pruebas, a modo general, se encontraron errores en cada uno de los módulos, estos errores se han enmarcado en ortográficos, en algunas ocasiones no muestran los mensajes de confirmación como en casos de eliminar, hay algunos botones que es necesario verificar porque la implementación es incompleta como son algunas funcionalidades para salvar y buscar, existen problemas a la hora de la descripción de los casos de uso, pues existen funcionalidades que no están descritas previamente, o que están descritas y no implementadas.

Conclusiones

Al finalizar este capítulo, podemos concluir que el proceso de pruebas se llevó a cabo de forma satisfactoria, pues los casos de prueba desarrollados, mostraron muchos de los errores del sistema. Estos se pueden clasificar como no catastróficos, previendo que estos puedan solucionarse de forma rápida y sin mayores problemas, pues están debidamente documentados, para una rápida comprensión por el equipo de desarrollo, y seguidamente pueda ser desplegado para pruebas de aceptación de usuarios.

Conclusiones generales.

En toda la labor realizada en este trabajo de diploma al planificar y aplicar pruebas al software PRIMICIA se obtuvieron los siguientes resultados:

- Se obtuvo un amplio conocimiento acerca de las técnicas de pruebas existentes, la estrategia para su aplicación y los pasos a seguir.
- Se definió el plan de prueba especificando fundamentalmente el alcance, la estrategia y los recursos.
- Se ejecutaron y evaluaron 12 casos de prueba diseñados y se obtuvo la lista de defectos con un resumen de errores y sugerencias.

De forma general, el flujo de trabajo 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 informaron y fueron arreglados, colaborando de esta manera con el proceso de Gestión de Cambios. Se lograron los objetivos propuestos, que el producto PRIMICIA a partir de las pruebas realizadas y su corrección posee una mayor calidad y un alto nivel técnico.

Recomendaciones

Una vez concluido el proceso de pruebas y dada la importancia que este tiene en la corrección de errores de un sistema se recomienda:

- Hacer extensiva la fase de prueba a los demás proyectos de la facultad, llevando la documentación necesaria cada vez que se le apliquen.
- Se recomienda realizarle pruebas de carga, stress que son pruebas de sistema, a la hora de desplegarlo.

Bibliografía

REFERENCIAS BIBLIOGRÁFICAS CITADAS

- (1) R. S. Pressman. *Ingeniería del software. Un enfoque práctico*. 4ª Edición. McGrawHill (1998)
- (2) NOVATICA. Número 137, Enero-Febrero 1999. *Monográfico Calidad del Software / Software de calidad*.
- (3) Carrasco, O.M.F, D.G. León, and A.B. Benavides, *Un enfoque actual sobre la calidad del software*, 2004 [Cite 2005 17/11/2005]; http://bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm.
- (4) IEEE *Standard Glossary of Software Engineering Terminology*, 1990.
Collazo, Manuel. *Técnicas de prueba del software. Estrategias de prueba del software*. 2003.
- (5) <http://www.calidadyssoftware.com>
- (6) <http://www.sqs.es/>
- (7) Norma ISO 8402 UNE (30 páginas)

REFERENCIAS CONSULTADAS

- (1) Comisión de Calidad. *CMMI: su aplicación en Empresas en general y en Pymes, beneficios u cuestionamientos*. Argentina : s.n., 2006.
- (2) Ailyn Febles, Sofía Álvarez, Humberto Fernández. *Calidad de software y la empresa, enseñanza de un tema imprescindible para el Ingeniero Informático*.
- (3) García, Lourdes, Álvarez, Sofía. *Una visión general sobre la certificación de calidad y la evaluación del proceso de desarrollo de software*. Rev. Ingeniería Industrial. No.
- (4) Johanna Rojas Rojas, Emilio José Barrios. *Investigación sobre estado del arte en diseño y aplicación de pruebas de software*. [En línea] 2007. [Citado el: 20 de enero de 2009.]
<http://www.udistrital.edu.co>.
- (5) Pressman, R. *Ingeniería del software: un enfoque práctico*. McGraw-Hill, 2002.
- (6) Aineris Reyes Nápoles, Mavis Rivera Nieblas. *Diseño y aplicación de pruebas al sistema IPC*. CIUDAD DE LA HABANA : s.n., 2007.
- (7) Noa, Eliane Paz. *Diseño y aplicación de pruebas al producto "Árbol Genealógico."*. Ciudad de la Habana, : s.n., 2007.
- (8) Lesdey Saavedra López, Yohandris Pupo Blez. *Diseño y aplicación de pruebas al producto Registro Cubano de Discapacitados*. Ciudad de la Habana, : s.n., 2007.

- (9) Alvarez, Juan Carlos Granja. *Controles y Metricas Tecnicas del software*. La Habana : s.n., 2007.
- (10) Darlen Hornia, Yisel Jorge. *Diseño, ejecución y evaluación de casos de prueba del Registro de Población de la Atención Primaria del Sistema de Información para la Salud*. Ciudad de la Habana : s.n., 2007.
- (11) R. S. Pressman. *Ingeniería del software. Un enfoque práctico*. 4ª Edición. McGrawHill (1998)
- (12) NOVATICA. Número 137, Enero-Febrero 1999. *Monográfico Calidad del Software / Software de calidad*.
- (13) Carrasco, O.M.F., D.G. León, and A.B. Benavides, Un enfoque actual sobre la calidad del software, 2004 [Cite 2005 17/11/2005]; http://bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm.
- (14) IEEE *Standard Glossary of Software Engineering Terminology*, 1990.
- Collazo, Manuel. *Tecnicas de prueba del software. Estrategias de prueba del software*. 2003.
- (15) <http://www.calidadyssoftware.com>
- (16) <http://www.sqs.es/>
- (17) *Norma ISO 8402 UNE (30 páginas)*
- (18) Mary Beth Chrissis, Mike Konrad, Sandy Shrum, "*CMMI® for Development, v1.2*", **2006**.
- (19) Pruebas de software, <http://lml.ls.fi.upm.es/ftp/ed2/0203/Apuntes/pruebas.ppt>