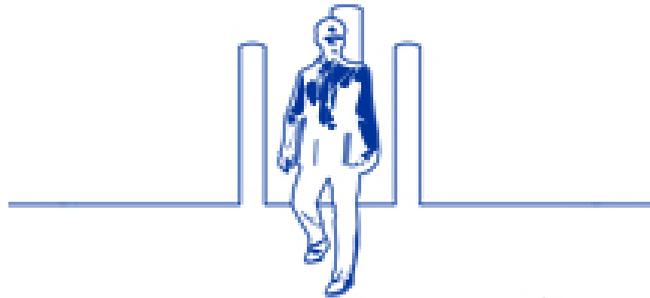


Universidad de las Ciencias Informáticas.



Título: Procedimientos y técnicas de transmisión de señales televisivas para la plataforma PTARTV.

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

Autor: Jeans Manuel Arias Arévalo.

Tutores:
Ing. Abel Díaz Berenguer.
Ing. Dunier Domínguez Mora.

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo al departamento de Señales Digitales de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los _____ días del mes de _____ del año 2011.

Jeans Manuel Arias Arévalo.

Ing. Abel Díaz Berenguer.

Ing. Dunier Domínguez Mora.

Datos de contacto

Tutor(a):

Nombre y apellidos: Abel Díaz Berenguer.

Sexo: M

Institución: Universidad de las Ciencias Informáticas.

Dirección de la institución: Carretera a San Antonio de los Baños, Km. 2 1/2,

Reparto: Torrens.

Municipio: Boyeros.

Provincia: Ciudad de La Habana.

Correo electrónico: aberenguer@uci.cu.

Teléfono del trabajo:-

Teléfono particular:-

Cargo del trabajador: Profesor

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Año de graduación: 2009.

Institución donde se graduó: Universidad de las Ciencias Informáticas.

Tutor(a):

Nombre y apellidos: Dunier Dominguez Mora.

Sexo: M

Institución: Universidad de las Ciencias Informáticas.

Dirección de la institución: Carretera a San Antonio de los Baños, Km. 2 1/2,

Reparto: Torrens.

Municipio: Boyeros.

Provincia: Ciudad de La Habana.

Correo electrónico: duniermail2006@gmail.com.

Teléfono del trabajo:-

Teléfono particular:-

Cargo del trabajador: Profesor

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Año de graduación: 2009.

Institución donde se graduó: Universidad de las Ciencias Informáticas.

Dedicatoria

*Quiero dedicar este trabajo a la persona que estuvo presente cuando todos se habían ido, que con mucha humildad me enseñó a amar la vida y siempre seguir adelante.
Ella es mi madre.*

Agradecimientos

Primero que todo agradecer a mi mamá por apoyarme incondicionalmente durante toda mi vida. Por representar la marca a la que siempre he querido llegar. Por ser la que siempre alimentó todos mis sentimientos y sueños. Gracias por tanto amor y comprensión, si volviera a nacer volvería a ser tu hijo. Te amo.

A Dios por acompañarme en los momentos más difíciles.

A mi abuela por siempre confiar en mí y darme ánimos cuando las cosas no iban bien y me parecía que todo se derrumbaba.

A mi papá por todo su apoyo y amor.

Zuiero agradecer también a mi tío Beto y a mis tías Marta, Chela y Carmen por ayudarme tanto durante mi vida y el desarrollo de mi carrera.

A mis hermanos Angel Antonio y Elizabet por darme el privilegio de compartir su amor y cariño.

Zuiero hacer llegar mi gratitud a mi segundo padre, que es mi abuelo Mañiño por tantos consejos y la educación que siempre me dio. Gracias.

A Mandy y Eugenia por darme tanto amor y cariño sin recibir nada cambio.

A mi novia, amante y amiga por todo su apoyo y comprensión, sin ti no hubiese sido posible. Gracias por estar.

A mis amigos Alfredo, Alejandro y José Carlos por ser como hermanos para mí.

A todos mis amigos, los de Bayamo y los de la UCI, por hacer de cada momento una historia que contar. Nunca los olvidaré.

Un agradecimiento especial a mis tutores Abel y Dunier por todo el apoyo profesional que me dieron, las horas de dedicación y por enseñarme a ser receptivo a las críticas sobre mi trabajo, pero más que todo por ser mis amigos. Sin ustedes no hubiera sido posible la realización de este trabajo.

A mi país por darme la posibilidad de formarme como profesional.

Opinión del tutor sobre el trabajo de diploma

La Plataforma de Transmisión Abierta para Radio y TV ha sido concebida como una aplicación informática que permita unificar los servicios radiales y televisivos de la Universidad de las Ciencias Informáticas y de cualquier otra institución. Dentro de este sistema las funcionalidades de transmisión televisiva son fundamentales en aras de lograr los objetivos de la plataforma.

El presente Trabajo de Diploma abordó de manera satisfactoria el desarrollo del subsistema Transmisión. Durante su ejecución los tutores consideran que el estudiante mostró las cualidades que a continuación se detallan.

Alta responsabilidad, compromiso, esfuerzo y laboriosidad.

Alta independencia, originalidad y creatividad.

El tesista trabajó con mucho empeño para lograr el cumplimiento de las tareas asignadas por el cronograma, generando la documentación necesaria para la investigación.

Las actividades desarrolladas en esta etapa han contribuido de manera significativa al desarrollo cognoscitivo sobre el tema en el proyecto PTARTV.

Prototipos funcionales de esta aplicación han sido expuestos en eventos como Informática 2011 y FESI 2011 avalando la investigación y desarrollo.

El trabajo también se presentó en la Jornada Científica Estudiantil donde obtuvo premio Relevante a nivel de Universidad.

El aporte científico-técnico del presente trabajo de diploma es alto y constituye un paso de avance en la informatización de la sociedad cubana pues en un futuro cercano la Dirección de Televisión Universitaria podrá aplicar los resultados obtenidos.

La premisa del uso de estándares abiertos es una propuesta novedosa para la UCI y el país.

Otras instituciones (incluyendo al ICRT) también pueden beneficiarse del trabajo realizado lo cual representa un logro significativo.

El documento presentado se encuentra bien estructurado y cuenta con el rigor necesario. La bibliografía es abundante, actualizada, confiable y se encuentra expuesta correctamente.

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero en Ciencias Informáticas; y se propone otorgarle al Trabajo de Diploma la calificación de 5 puntos. Se recomienda la publicación de los resultados obtenidos en eventos científicos.

Ing. Abel Díaz Berenguer. _____

Ing. Dunier Domínguez Mora. _____

Reconocimientos y avales

Jornada
**Científica
Estudiantil**



9na
Jornada
científica
estudiantil

UCI

Se le otorga
el presente

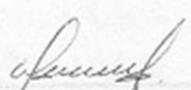
Reconocimiento

*A. Procedimientos y técnicas de transmisión de
señales televisivas para la plataforma PTAR TV.*

Por haber obtenido la categoría de

Relevante

En la 9na edición de la Jornada Científica Estudiantil


Alexis Camue Henandez
Presidente Fed-6


Luis Manuel Vidal Piña
Presidente Consejo Científico
Fac-6


Yanet Villanueva Armenteros
Decano Facultad-6

Universidad de las Ciencias
Informáticas
UCI
Federación de las Ciencias Informáticas
Federación Estudiantil Universitaria

Otorga el presente:

Reconocimiento

A. Procedimientos y técnicas de transmisión de señales
televisivas para la plataforma PTARTV
Por haber obtenido:

Relevante

"El conocimiento nos hace responsable.."
Ernesto Ché Guevara



Lester González López
Presidente FEJUICI

Ángel Vega García
Vicerrector de Investigación y Postgrado



Resumen

El creciente desarrollo de las comunicaciones y las redes computacionales ha proporcionado que muchas de las actividades realizadas por las personas se integren a diferentes aplicaciones informáticas. La televisión se ha beneficiado mucho de esta fusión. El envío y recepción de contenidos audiovisuales apoyándose en ordenadores ha mejorado mucho la realización de la televisión actual, brindando servicios cada vez más fiables e interactivos.

En el presente trabajo se exponen las bases de la implementación del subsistema Transmisión de la Plataforma de Transmisión Abierta para Radio y Televisión. Esta plataforma tiene como principal objetivo la distribución de contenidos audiovisuales haciendo uso de los sistemas de transmisión de señales digitales televisivas empleándose tecnología streaming.

Este documento recoge resultados y conclusiones de la investigación realizada con el objetivo de caracterizar las tecnologías, técnicas de programación y herramientas necesarias para el desarrollo del subsistema. También se describen las funciones de la aplicación exponiéndose a través del modelo de implementación y los diagramas de componentes. Una vez programada la aplicación se prueban sus principales funcionalidades y se recomiendan puntos de gran importancia para la implementación de versiones posteriores.

Palabras claves:

Administración, Plataforma, Streaming, Televisión, Transmisión, Video.

Índice

Introducción	1
Capítulo 1. Fundamentación teórica	5
1.1 Introducción	5
1.2 Conceptos asociados al dominio del problema	5
1.2.1 Administración:	5
1.2.2 Digital:	6
1.2.3 Señal:	6
1.2.4 Sistema:	6
1.2.5 Transmisión:	7
1.3 Objeto de estudio.....	7
1.3.1 Descripción general del objeto de estudio.....	7
1.3.2 Descripción actual del dominio del problema	10
1.3.3 Situación problemática.....	11
1.4 Análisis de otras soluciones existentes	13
1.4.1 Soluciones similares	14
1.5 Conclusiones	15
Capítulo 2. Herramientas y tecnologías a utilizar	16
2.1 Introducción	16
2.2 Metodología de desarrollo	16
2.2.1 Proceso Unificado de Desarrollo de Software	16
2.3 Lenguaje de modelado	16
2.3.1 Lenguaje Unificado de Modelado.....	17
2.4 Herramienta de Ingeniería de Software Asistida por Computadoras	17
2.4.1 Visual Paradigm.....	17
2.5 Paradigma de programación.....	18
2.5.1 Programación Orientada a Objetos.....	18

2.6	Lenguaje de programación	19
2.6.1	Java.....	19
2.6.2	C++.....	20
2.6.3	Selección del lenguaje de programación	21
2.7	Marco de Trabajo.....	21
2.7.1	Qt	21
2.8	Entorno Integrado de Desarrollo	21
2.8.1	QT Creator.....	22
2.9	Sistema Gestor de Base de Datos	22
2.9.1	PostgreSQL	23
2.10	Biblioteca Libvlc	23
2.11	Conclusiones	23
Capítulo 3. Descripción y validación de la solución propuesta		25
3.1	Introducción	25
3.2	Descripción del subsistema Transmisión	25
3.3	Diagrama de componentes	26
3.4	Modelo de implementación	27
3.5	Estándar de codificación.....	28
3.5.1	Comentarios	28
3.5.2	Nombres de identificadores	29
3.5.3	Organización Visual del Programa.....	31
3.6	Validación de la solución propuesta.....	32
3.6.1	Elementos del proceso de pruebas.....	32
3.6.2	Niveles de pruebas	32
3.6.3	Tipos de prueba.....	33
3.6.4	Pruebas de caja blanca	35
3.6.5	Diseño de pruebas de caja blanca.....	36
3.6.6	Pruebas funcionales	41

Índice

3.6.7	Diseño de pruebas funcionales.....	41
3.6.8	Pruebas de rendimiento.....	47
3.6.9	Diseño de pruebas de rendimiento	47
3.6.10	Resultado de las pruebas	49
3.6.11	Segunda iteración de pruebas funcionales para el caso de uso Gestionar canal.....	50
3.7	Conclusiones	53
	Conclusiones generales.....	54
	Recomendaciones	55
	Trabajos citados	56
	Bibliografía consultada.....	60
	Anexos.....	64
	Glosario de términos.....	65

Índice de ilustraciones

Ilustración 1. Diagrama de componentes para el subsistema Transmisión.	27
Ilustración 2. Modelo de implementación.	28

Índice de tablas

Tabla 1. Tipos de prueba que se le aplican a un software.....	35
Tabla 2. Secciones a probar en el caso de uso Gestionar canal.	42
Tabla 3. Descripción de variables para el caso de uso Gestionar canal.....	43
Tabla 4. Matriz de datos para la sección Crear nuevo canal.	44
Tabla 5. Secciones a probar en el caso de uso Autenticar usuario	46
Tabla 6. Descripción de variables para el caso de uso Autenticar usuario.....	46
Tabla 7. Matriz de datos para la sección Autenticar usuario.	47
Tabla 8. Niveles de consumo de memoria y de la CPU para la estación de trabajo 1.....	48
Tabla 9. Niveles de consumo de memoria y de la CPU para la estación de trabajo 2.....	49
Tabla 10. Matriz de datos 2 para la sección Crear nuevo canal.	51
Tabla 11. Matriz de datos 2 para la sección Modificar canal.	53

Introducción

La televisión (TV), es uno de los medios de difusión más usado por el hombre para mantener su contacto con el acontecer global. Desde su surgimiento, se buscaron disímiles formas de mejorar las transmisiones televisivas. Años más tarde llegaría la informática, elemento que representaba un paso trascendental en el mundo de la gestión de información. La misma, se fue desarrollando de forma muy rápida, comenzando a integrarse a un número muy grande de procesos desarrollados por los seres humanos. La TV no queda fuera de esta integración, llegando a un punto de convergencia donde el hombre a través de la informática busca vías que contribuyan a mejorar la forma en que se desarrolla la televisión actual.

Varias empresas del mundo guían sus investigaciones a la creación de algoritmos para la gestión, almacenamiento y procesamiento del contenido audiovisual, propiciando a las cadenas televisivas soluciones que automatizan y hacen más fácil el arte de hacer televisión.

Cuba comienza a dar sus primeros pasos en la industria del software. El país se encuentra en medio de un proceso de informatización que le ha permitido incursionar en muchas de las ramas del desarrollo de aplicaciones, destacándose algunas como: software educativo, software para la gestión de procesos, inteligencia artificial y televisión. Para apoyar este proceso se crea la Universidad de las Ciencias Informáticas (UCI) en el año 2002, compuesta actualmente por diez facultades, siete de ellas ubicadas en la sede central en La Habana y el resto en las provincias de Granma, Ciego de Ávila y Artemisa. Entre estas facultades se encuentra la Facultad 6, la cual cuenta con el centro Geoinformática y Señales Digitales (GEYSED). Este centro tiene como misión desarrollar productos, servicios y soluciones informáticas en el campo de la Geoinformática y el procesamiento de Señales Digitales, contribuyendo a la formación integral de profesionales que respondan a las necesidades del progreso científico técnico y socioeconómico, permitiendo un posicionamiento en el mercado nacional e internacional.

La Universidad de las Ciencias Informáticas también posee una Dirección de Televisión Universitaria (DTU), la cual tiene como misión, apoyar el proceso de enseñanza- aprendizaje y la formación cultural e integral de la comunidad universitaria. En esta entidad televisiva se ejecutan diariamente una serie de procesos que en su mayoría funcionan de forma manual e independiente. Entre ellos se encuentran: el inicio de una transmisión, la construcción de las programaciones y elaboración de un reporte. La transmisión es uno de los elementos más importantes ya que esta es el resultado final después de un arduo trabajo de producción de los programas y es el producto que llega a los televidentes. La DTU posee varios canales de programas, estos se transmiten apoyándose en una solución informática desarrollada con tecnologías privativas y que no cumple con las necesidades y expectativas de la UCI.

Introducción

Periódicamente se producen errores de reproducción y fallos en la transmisión. Estos sucesos son apreciados por los televidentes y en ocasiones, también es visible el proceso de solución de los problemas. El orden en el que se transmiten los contenidos audiovisuales es determinado manualmente por el transmisor, guiándose por un documento con la planificación previamente elaborada. Esta situación da lugar a errores humanos que pueden provocar transmisiones fuera del horario planificado, o que no lleguen a transmitirse los programas planificados. Lo descrito anteriormente trae consigo que los telespectadores no estén conformes con el servicio que se les brinda y que la DTU no cumpla su misión con la calidad requerida.

Para revertir esta situación la UCI asigna al centro GEYSED, el desarrollo de un sistema capaz de automatizar determinados procesos que se ejecutan en la Dirección de Televisión Universitaria. Con el objetivo de construir dicho sistema surge el proyecto de investigación, innovación y desarrollo Plataforma de Transmisión Abierta para Radio y Televisión (PTARTV). El mismo constituye una solución informática que cuenta con doce subsistemas, de los cuales dos, Transmisión y Administración de la Transmisión, son los responsables de gestionar los procesos asociados a la emisión televisiva. De acuerdo a la estructura actual de la DTU estos subsistemas dependen para su funcionamiento de una estación de trabajo por cada canal que exista (clientes de transmisión). Otra estación es la encargada de administrar qué canal sale al aire, así como la adición, modificación o eliminación de un canal (servidor de transmisión).

El modo de transmisión de los subsistemas es basado en señales digitales, específicamente tecnología streaming¹. Aunque las señales inicialmente son digitales, deben llegar analógicamente a los equipos de multiplexación, combinación y amplificación de señales de la DTU, que en este caso son todos analógicos. Para que un medio analógico sea capaz de interpretar una señal digital, necesita de un dispositivo que sea capaz de convertir esta señal. En este caso se utiliza un Set-top box² conectado a cada uno de los clientes para recibir una señal streaming. Esta señal es emitida por el servidor de transmisión hacia los clientes.

Al mismo tiempo, el intérprete del lenguaje con que se desarrolló el sistema consume un porcentaje muy grande de memoria, por lo que se ve afectado el rendimiento de la aplicación.

Actualmente la UCI no tiene la posibilidad de adquirir el conjunto de Set-top-box que necesitan los subsistemas para poder funcionar. En busca de una solución, PTARTV guía sus investigaciones a desarrollar un nuevo sistema con un costo computacional y económico relativamente bajo. Además

¹Streaming: tecnología utilizada para la ejecución de audio y video por la red de datos permitiendo escuchar y visualizar los archivos mientras se están descargando.

²Set-top-box: dispositivo multifunción que permite la recepción y distribución de señales procedentes de diversos tipos de redes de comunicación, entre ellos: radio, televisión, cable, satélite e Internet.

Introducción

de ser capaz de adaptarse a un nuevo entorno en caso de adquirir el conjunto de Set-top box necesario para transmitir televisión en la UCI. Atendiendo a la situación planteada, en el presente trabajo de diploma se formula como **problema de investigación**: *los procesos de transmisión y administración de la transmisión de la plataforma PTARTV no satisfacen las necesidades requeridas de la Dirección de Televisión Universitaria*. Para dar solución al problema de investigación se establece como **objetivo general**: *Migrar los subsistemas de transmisión televisiva de la plataforma PTARTV y personalizarlos para la DTU*. Por lo que se delimita como **objeto de estudio**: *los procesos de transmisión y administración de la transmisión de señales televisivas digitales*. Como **campo de acción**: *migración y personalización de los procedimientos y técnicas de transmisión y administración de la transmisión de señales televisivas digitales empleados en la plataforma PTARTV*. Por lo anteriormente planteado se tiene la siguiente **idea a defender**: *la migración y personalización de los subsistemas de transmisión y administración de la transmisión mejorará los procesos de transmisión televisiva en la plataforma PTARTV*.

En función de dar cumplimiento al objetivo general se definen las siguientes tareas de la investigación:

1. Caracterizar los procesos relacionados con la transmisión de señales en la DTU.
2. Caracterizar los procedimientos y técnicas relacionados con la transmisión y administración de la transmisión en la Plataforma de Transmisión Abierta para Radio y Televisión.
3. Caracterizar sistemas de transmisión televisiva.
4. Determinar las herramientas y tecnologías de desarrollo a utilizar en la construcción del software.
5. Desarrollar artefactos y documentación según la metodología de desarrollo seleccionada.
6. Implementar las funcionalidades de los subsistemas Transmisión y Administración de la Transmisión.
7. Validar la propuesta de solución.

Con el objetivo de obtener conocimientos necesarios que hagan posible la materialización del objetivo general, se ejecuta una investigación en la que se utilizan los siguientes métodos científicos:

Métodos Teóricos:

1. **Analítico-Sintético**: Con este método se analizan los procesos de transmisión de contenidos audiovisuales. Se emplea también para caracterizar soluciones similares que

Introducción

permitan aprovechar puntos en común y conceptos teóricos que sean de relevancia para la investigación.

2. **Histórico-Lógico:** Permite valorar el desarrollo, evolución y tendencias actuales de los procedimientos y técnicas para la gestión de la transmisión de contenido audiovisual.
3. **Modelación:** Durante la investigación se utiliza para crear abstracciones que permitan comprender el funcionamiento del sistema mediante los modelos de implementación, diagramas de componentes y modelo de despliegue.

Métodos Empíricos:

1. **Entrevista:** Se utiliza para obtener información sobre la forma en que funcionan actualmente los procesos de transmisión en la DTU. Tomando como personal a entrevistar a los trabajadores de esta entidad. Para ello se ha tomado como población a 5 trabajadores del departamento Programación, específicamente en el área de Transmisión. De ellos se toman como muestra 3 individuos para ser entrevistados, lo que representa el 60% de la población. Se realiza una técnica de muestreo no probabilístico, específicamente muestreo intencional.
2. **Observación:** Permite corroborar la información obtenida en la entrevista a los trabajadores de la DTU. Específicamente se utiliza el tipo de observación "De acuerdo con el grado de formalización de la observación" y dentro de esta: observación no estructurada.

Capítulo 1. Fundamentación teórica

1.1 Introducción

Hoy día la industria del software se encuentra en un constante desarrollo, ya que las exigencias de la sociedad son cada vez mayores. Aspectos como la integridad de los datos y la flexibilidad, son elementos que no deben pasarse por alto al desarrollar un sistema informático.

En este capítulo se estudian e identifican los conceptos y argumentos teóricos que permiten una mejor comprensión del contenido de la investigación, sirviendo de base para la construcción los subsistemas transmisión televisiva de la plataforma PTARTV. Se caracterizan los procesos relacionados con la transmisión de señales televisivas de la DTU y la plataforma PTARTV. Además de la realización de un análisis de soluciones existentes con características similares al software que se desea desarrollar.

1.2 Conceptos asociados al dominio del problema

A continuación se enuncian un conjunto de conceptos que servirán para trazar la línea base de la presente investigación, brindando definiciones básicas y específicamente en el ámbito informático.

1.2.1 Administración:

Según la Real Academia Española (RAE) **administración** es la acción y efecto de administrar. Conceptualizando administrar como ordenar, disponer u organizar bienes (1).

Por su parte el doctor en técnicas de administración Idalberto Chiavenato, plantea que **administración** es "el proceso de planear, organizar, dirigir y controlar el uso de los recursos para lograr los objetivos organizacionales" (2).

Reinaldo O. Da Silva, consultor en Desarrollo Organizacional/Administrativo y en Instrumentación de Sistemas de Calidad, define la **administración** como "un conjunto de actividades dirigido a aprovechar los recursos de manera eficiente y eficaz con el propósito de alcanzar uno o varios objetivos o metas de la organización" (3).

De acuerdo a los criterios estudiados se concluye que **administración** es el grupo de actividades que realiza un ente determinado para disponer o determinar sobre un conjunto de objetos con un fin establecido.

Teniendo en cuenta lo anteriormente expresado, se puede asumir que en el ámbito de la informática **administración** sería el grupo de instrucciones que recibe un software o hardware, con el objetivo de ejecutar un algoritmo o acción sobre determinados datos o herramientas.

1.2.2 Digital:

Digital, se refiere a información que se almacena utilizando una serie de unos y ceros. Las computadoras son máquinas digitales, ya que sólo puede leer la información como encendido o apagado - 1 o 0. Este método de cálculo, también conocido como el sistema binario, puede ser usado para representar grandes volúmenes de datos (4).

La revista digital MasterMagazine define **digital** como información convertida al código binario, el lenguaje de números (ceros y unos) que emplean los ordenadores para almacenar y manipular los datos (5).

Puede decirse que **digital** es el conjunto de valores en código binario que representan un dato o conjunto de datos, permitiendo ser manipulados e interpretados de acuerdo a su codificación.

1.2.3 Señal:

El diccionario en línea Definición.de establece **señal** como un signo, seña, marca o medio que informa, avisa o advierte algo. Este aviso permite dar a conocer una información, realizar una advertencia o constituirse como un recordatorio (6).

Se puede decir que **señal** es la transmisión de información a través de la modulación de una onda electrónica (7).

Por otra parte una **señal** puede ser manifestada a través de la variación de una corriente eléctrica o cualquier otra magnitud física que se utilice, con el objetivo de transmitir información (8).

Finalmente se puede asumir que **señal** es un valor que proporciona cierta información, que puede ser interpretada o utilizada en función de un objetivo determinado.

1.2.4 Sistema:

La RAE define que sistema es el conjunto de cosas que relacionadas entre sí ordenadamente contribuyen a determinado objeto (1).

La IEEE Standard Dictionary of Electrical and Electronic Terms da la siguiente definición: "**sistema** es un todo integrado, aunque compuesto de estructuras diversas, interactuantes y especializadas. Cualquier sistema tiene un número de objetivos, y los pesos asignados a cada uno de ellos pueden variar ampliamente de un sistema a otro. Un sistema ejecuta una función imposible de realizar por cualquiera de las partes individuales. La complejidad de la combinación está implícita" (9).

Capítulo 1. Fundamentación teórica

Se puede concluir que **sistema** es la interrelación de varias estructuras, cada una de ellas se especializa en una función determinada, tributando a un resultado final que es fruto de la unión de todas sus especializaciones.

Informáticamente hablando, un **sistema** sería la combinación de hardware y software para obtener una solución integral, compuesta por varios componentes que trabajen como un todo, en función de brindar un resultado que no podrían obtener cada uno de forma independiente.

1.2.5 Transmisión:

Al conceptualizar **transmisión**, la RAE lo considera como la acción y efecto de transmitir. Definiendo transmitir como hacer llegar a alguien mensajes o noticias (1).

También puede definirse **transmisión** como la transferencia de datos a través de un canal de comunicaciones (10).

Por su parte The Free Dictionary conceptualiza **transmisión** como el acto o proceso de enviar de un lugar a otro un mensaje, una imagen o cualquier otra información por medio de ondas de radio, señales eléctricas, señales luminosas, entre otras (11).

Enfocándose en el ambiente de las señales digitales televisivas **transmisión** sería la recepción y envío de una señal digital de video o audio a través de una red de datos³.

1.3 Objeto de estudio

En el presente trabajo de diploma se ha definido como **objeto de estudio**: *los procesos de transmisión y administración de la transmisión de señales televisivas digitales.*

1.3.1 Descripción general del objeto de estudio

Las señales digitales han pasado a ocupar un lugar cimero en el mundo de las telecomunicaciones, mientras que la vida de las tecnologías analógicas cada vez es más corta; en el mundo se conoce este proceso como “**apagón analógico**” (12).

Una señal digital es un signo electrónico transmitido en código binario que puede ser la presencia o ausencia de pulsos de corriente, altos y bajos voltajes en una frecuencia particular (13).

Las computadoras se han convertido en uno de los equipos que más se usan en el trabajo con señales digitales, abriéndose paso en diversas esferas de la sociedad. La TV como medio de

³ Red de datos: es un sistema que enlaza dos o más puntos (terminales) por un medio físico, el cual sirve para enviar o recibir un determinado flujo de información.

Capítulo 1. Fundamentación teórica

difusión ha aprovechado al máximo esta tecnología, dando paso al surgimiento de la TV Digital (DTV, por sus siglas en inglés).

La DTV consiste en la emisión de información binaria para ser mostrada en un receptor capaz de interpretar señales digitales televisivas, garantizando una mejor calidad de imagen y sonido; conteniendo la posibilidad de prestar servicios de información adicional basados en la interactividad (14).

Muchos países, principalmente los desarrollados, han avanzado mucho en la producción de TV digital. Estos cuentan con equipos digitales sofisticados que son capaces de transmitir con calidad de imagen de definición estándar (SD, por sus siglas en inglés) y calidad de imagen de alta definición (HD, por sus siglas en inglés) (15).

La televisión digital impone un cambio. El tránsito del dominio analógico al digital es un proceso inevitable. Los equipos analógicos se dejarán de producir, no habrá piezas de repuesto, y esta tecnología quedará completamente obsoleta (16).

Una señal digital puede almacenarse fácilmente en un soporte magnético (disco o cinta) o ser enviada por una red de datos. El hecho de ser digital disminuye el riesgo de que exista deterioro o pérdida en la fidelidad de la señal.

Las televisoras que hacen transmisiones digitales suministran al televidente una calidad de imagen y sonido superior a la de las transmisiones analógicas. Más que limitarse a ofrecer un programa analógico, una televisora podrá ofrecer un programa digital de HD con un porcentaje de resolución muy alto, o varios programas digitales de SD simultáneamente mediante el proceso llamado "multitransmisión". La multitransmisión permite transmitir varios canales de programación digital al mismo tiempo, usando el mismo espectro⁴ que requiere un solo programa analógico. Por ejemplo, mientras que una estación que transmite en formato analógico por el canal 7, sólo puede ofrecer a los televidentes un programa, una estación que transmite en formato digital puede ofrecer un programa digital en el canal 7-1, un segundo programa digital en el Canal 7-2, un tercero en el canal 7-3 y así sucesivamente (17).

La DTV ofrece la posibilidad de aplicar distintos niveles de interactividad. El televidente puede consultar el parte del tiempo, publicaciones periódicas, informaciones diversas, datos sobre un juego de beisbol, o sobre los actores de un dramatizado. También permite al usuario interactuar en tiempo real en programas de participación e incluso, seleccionar el ángulo en que desea disfrutar de un evento deportivo. Estas opciones no estarán disponibles con solo aplicar la televisión digital, sino que será un valor agregado por la industria (16).

⁴ Espectro: Gama de frecuencias electromagnéticas utilizada en la transmisión de datos, audio y video.

Capítulo 1. Fundamentación teórica

La seguridad en un Sistema de Transmisión de Señales Digitales Televisivas (STSDT) es un término que está compuesto por varios complementos. Un STSDT debe cumplir con aspectos como: integridad, disponibilidad, flexibilidad y poseer una vía de administración. A continuación se describen brevemente los elementos antes mencionados.

✓ **Integridad:**

La transmisión debe brindar al televidente la garantía de que los materiales audiovisuales que a él están llegando se trasladan por una vía confiable. Sin correr el riesgo de que agentes externos al sistema de transmisión puedan modificar el contenido de la emisión.

✓ **Disponibilidad:**

Mucha importancia se le confiere a la seguridad en una transmisión digital televisiva. Es necesario que el emisor garantice más de una fuente de transmisión, previendo que si existe algún fallo sea apreciado lo menos posible por los televidentes. Se debe lograr también que las transmisiones se efectúen en los horarios planificados, brindando al televidente puntualidad y seriedad en el servicio.

✓ **Flexibilidad:**

La facilidad de adaptar la transmisión a las condiciones dadas en el momento en que se está ejecutando, es un aspecto esencial. Una transmisión digital televisiva debe permitir que personal autorizado sea capaz de manipular la misma, en función de solucionar un problema, o simplemente responder a un cambio emergente. Un ejemplo es la interrupción de un programa para emitir una noticia de última hora.

✓ **Administración**

La administración, es un aspecto fundamental para el funcionamiento de cualquier sistema de transmisión televisiva. Desde el punto de vista de las televisoras, tener el control sobre las señales que se transmiten, cómo y cuándo se transmiten, es un principio inviolable. De disímiles maneras puede manifestarse la administración en un sistema de transmisión; desde iniciar una transmisión, hasta designar hardware para una nueva vía de emisión.

La implementación de una estructura para la transmisión de señales digitales televisivas requiere de una combinación de hardware y software que funcione como un todo, teniendo como objetivo principal, obtener señales que puedan ser manipuladas y utilizadas con facilidad por los operadores de transmisión.

1.3.2 Descripción actual del dominio del problema

La Universidad de las Ciencias Informáticas es una institución que forja como objetivo principal la producción de software para abrirse paso en el mercado internacional. Además de la formación integral de ingenieros que sean la fuerza motriz para la informatización de la sociedad cubana.

Como parte de su proceso de formación de profesionales, la UCI posee como asignatura rectora de su proceso de enseñanza-aprendizaje la Práctica Profesional. Para esto, se crearon los centros de producción, constituidos por departamentos y estos a su vez por proyectos de investigación, innovación y desarrollo. La Facultad 6, perteneciente a la UCI posee dos centros de producción: GEYSED (Geoinformática y Señales Digitales) y DATEC (Centro de Tecnologías de Gestión de Datos). Dentro del centro GEYSED existe el departamento de Señales Digitales, que asume el desarrollo del proyecto PTARTV. Este proyecto es el encargado de desarrollar un sistema que automatice todos los procesos que se realizan a diario en una televisora.

La DTU es la entidad encargada de hacer llegar a toda la comunidad de la UCI, tanto la televisión nacional como los canales internos de la universidad. Para que un programa interno pueda ser apreciado por los televidentes debe antecederle un proceso de producción que puede abarcar desde la filmación, hasta la transmisión de ese programa.

Hoy día, el principal abastecedor de software de la DTU es el centro GEYSED. Se considera que la construcción de la plataforma PTARTV dotará a la DTU de una herramienta capaz de automatizar un gran número de procesos que se efectúan normalmente para la producción de TV. El presente trabajo se enmarca específicamente en el desarrollo de dos de los subsistemas que forman parte de la plataforma PTARTV y que tienen como función automatizar el proceso de transmisión de materiales audiovisuales. Estos subsistemas son nombrados: Transmisión y Administración de la Transmisión.

Actualmente existen otras soluciones que brindan servicios similares a los que desarrolla PTARTV. Estas soluciones son propietarias y representan un elevado costo. La UCI es una de las principales impulsoras en la lucha por la independencia y soberanía tecnológica. La producción de software, tanto para empresas nacionales como para extranjeras es una ayuda considerable a la economía del país. Se considera que la construcción de PTARTV proveerá a la DTU de una solución económicamente factible, ya que optimiza los recursos materiales y humanos de esta entidad. Por su parte los subsistemas Transmisión y Administración de la Transmisión constituyen elementos de vital importancia para el desarrollo y despliegue de PTARTV.

1.3.3 Situación problemática

La UCI es un centro que basa su proceso de enseñanza-aprendizaje en el uso de las Tecnologías de la Información y las Comunicaciones (TICs). De ahí que la formación cultural de los miembros de la comunidad universitaria también se ve muy ligada a las TICs, es por ello que en la Universidad se ha creado la DTU. Entidad que posibilita la difusión de teleclases, noticias y muchos otros programas que contribuyen significativamente a la formación cultural de la comunidad universitaria.

La DTU posee varios departamentos, de todos ellos, el departamento de Programación es el encargado de ejecutar la transmisión de materiales audiovisuales. Dicho departamento está dividido en tres áreas: Programación, Centro de Gestión de Información Audiovisual (CGIA) y Transmisión.

El proceso comienza por el área de Programación, donde se realiza una revisión de todos los materiales para formar parte del banco de audiovisuales de la DTU. Estos se clasifican en aptos o no aptos. Si el material pasa la revisión satisfactoriamente (material apto), el CGIA procede a la catalogación y almacenamiento del mismo. Desde este momento es posible incluir el audiovisual en un programa de transmisión (comúnmente denominado planificación de transmisión). Cuando Programación incluye el material en una de las planificaciones, entonces se transfiere también al área de Transmisión.

Transmisión inicialmente archiva todos los materiales audiovisuales en un servidor, que es la fuente para la posterior transferencia hacia cada una de las estaciones de trabajo en las que se apoya la transmisión de los canales que posee la DTU (estaciones de transmisión).

Al comenzar una jornada laboral el transmisor copia las medias hacia cada una de las estaciones de transmisión, guiándose por un documento con la planificación elaborada por el área Programación. Una vez concluida la copia, el transmisor establece el orden de reproducción de las medias de los canales, rigiéndose también por el documento antes mencionado. Esta operación se realiza a través de un software para conectarse a las estaciones de transmisión, visualizar todas las medias que fueron copiadas anteriormente y generar las listas de reproducción correspondientes. Estas listas se reproducen de forma automática por un reproductor en ejecución constante y a pantalla completa.

La transmisión de un audiovisual depende totalmente de la correcta ejecución de los procesos mencionados anteriormente. Se puede apreciar que dichos procesos en su mayoría se realizan de forma manual e independiente, provocando que se materialicen errores que afectan de forma considerable la transmisión de un material. Por ejemplo: si al realizar la transferencia de los

Capítulo 1. Fundamentación teórica

ficheros hacia las estaciones de transmisión no se copian todos los ficheros que corresponden, no estarán disponibles a la hora que se deba efectuar su reproducción. Cuando una de las medias no está en la estación de transmisión o simplemente el fichero está defectuoso, el reproductor continúa hacia la media siguiente sin emitir ningún tipo de aviso del evento. Esto trae consigo que se alteren los horarios de la salida al aire de los programas que siguen a continuación.

En cada canal existe una carpeta con materiales de emergencia previendo que el reproductor proporcione un error o esté reproduciendo de forma indebida algún fichero. Si el transmisor se percata de lo sucedido, procede entonces a sustituirlo por un material de emergencia. Para sustituir un material, hacer una interrupción o pausar la transmisión, el transmisor debe conectarse de forma remota a la estación correspondiente y hacerlo directamente en el reproductor. Estos eventos son apreciados por los televidentes desde sus telerreceptores.

La aplicación de la que se auxilia el transmisor para visualizar la lista de medias en una estación de transmisión y generar las listas de reproducción posee una interfaz poco amigable.

Todos los elementos anteriormente mencionados propiciaron un descontento por parte de la comunidad universitaria con respecto al servicio que presta la DTU.

Con el objetivo de dar solución a estos problemas la UCI asigna al proyecto PTARTV el desarrollo de un software que automatice varios de los procesos que se ejecutan en la Dirección de Televisión Universitaria.

PTARTV es una plataforma que tiene como objetivo principal automatizar los procesos que se llevan a cabo en una televisora. Al adaptar dicho sistema a la DTU, quedó estructurado en doce subsistemas. Dos de éstos: Transmisión y Administración de la Transmisión, son los que están involucrados con el proceso de emisión televisiva. La DTU posee una computadora por cada canal de programa que envían hacia red de transporte de señales televisivas de la UCI.

Al adaptar los subsistemas Transmisión y Administración de la Transmisión a las condiciones actuales de transmisión audiovisual de la DTU, se estructura de la siguiente forma: una Computadora Personal (PC, por sus siglas en inglés) por cada canal que vaya a transmitirse (cliente de transmisión) y otra para cumplir la función de administradora del sistema (servidor de transmisión).

En cada uno de los clientes debe estar desplegado un servidor streaming. Este servidor streaming es el encargado de publicar un flujo de video que llegue desde otra estación. Habrá un Set-top-box conectado a cada uno de los clientes de transmisión. La función del Set-top-box será tomar el flujo de video que publique el servidor streaming correspondiente a un cliente de transmisión y enviarlo

Capítulo 1. Fundamentación teórica

hacia los equipos de modulación, combinación y amplificación de señales. Finalmente estas señales se envían a la red de transporte de señales televisivas de la universidad.

En el servidor de transmisión están desplegados los subsistemas Transmisión (Tx) y Administración de la Transmisión (ATx). Administración de la Transmisión tiene como función, adicionar, eliminar, o modificar los datos de un canal. Tx provee a los servidores clientes de los flujos streaming correspondientes para su posterior publicación. Otras de las funcionalidades de este subsistema son: parar, reanudar, pausar e iniciar una transmisión.

En el caso de ATx, cuando se desee adicionar un nuevo canal de programas se deben llenar los campos correspondientes a un canal en la interfaz de la aplicación. Además se debe agregar un nuevo hardware a la estructura (una nueva estación para desempeñar el papel de cliente de transmisión) y se le debe instalar el paquete de software correspondiente a un servidor de transmisión.

Al iniciar una transmisión, el flujo streaming se genera en el servidor de transmisión. Teniendo en cuenta la información almacenada en la base de datos el sistema carga la programación correspondiente a ese día y a esa hora. Una vez cargada la programación el sistema identifica a que canal corresponde la misma y entonces comienza a hacer streaming en vivo dirigido hacia el cliente correspondiente y generando un fichero de configuración en la carpeta de publicación del servidor streaming montado en el cliente de transmisión.

Hasta el momento la UCI no alberga la posibilidad de adquirir el conjunto necesario de Set-top-box para desplegar los subsistemas Transmisión y Administración de la Transmisión. El soporte del lenguaje con el que se desarrolló el software consume un porcentaje muy alto de memoria, lo que afecta considerablemente el rendimiento del sistema.

1.4 Análisis de otras soluciones existentes

En el ciclo de investigación se examinan las reseñas del proyecto de software que se propone. En el mercado internacional pueden encontrarse aplicaciones que poseen algunas funcionalidades similares a la solución propuesta. Las mismas son aplicaciones privativas que se venden por un elevado costo y que en su mayoría requieren de un hardware específico para la solución. Para efectuar una prueba de la eficiencia de estos productos habría que adquirirlos en el mercado. Los fabricantes de estos sistemas no brindan grandes volúmenes de información a menos que se cierre un contrato legal, lo que provoca que aparezca muy poca información sobre estas aplicaciones. A continuación se presentan las características más relevantes de algunos de los sistemas existentes en la actualidad.

1.4.1 Soluciones similares

1.4.1.1 X Frame TV

Sistema Digital de Gestión Integral para Televisión. Es la aplicación principal del sistema X Frame, de la empresa ASPA. Permite realizar la programación del audio o el video y toda la gestión de escaletas o parrillas. Incluye módulos de grabación y edición de audio y video, permitiendo realizar la emisión de ambos. X Frame TV es usado por más de 100 estaciones entre emisoras de radio y televisoras (18).

Características principales:

- Sistema con estructura cliente servidor que permite un control más pormenorizado de cualquier circunstancia ocurrida en el sistema.
- El trabajo con cualquier tarjeta de audio (profesional o doméstica).
- La posibilidad de gestionar múltiples tarjetas de audio o video, en el mismo equipo o en equipos remotos.
- Posee buzones de video con subcarpetas (estructura de árbol).
- Gestión de usuarios y grupos garantizando la seguridad del sistema.

X Frame en emisión:

- Cuenta con carga automática de programación y con emisión automática.
- Emisión de múltiples formatos de video en la misma lista.
- Incrustación de logos y titulaciones de forma automática y manual.

1.4.1.2 Dayang automation

Esta solución está diseñada para automatizar el proceso de emisión. El sistema es flexible y modular, capaz de satisfacer un amplio grupo de necesidades de los organismos de difusión. Es fiable y fácil de usar. Controla una amplia gama de dispositivos, incluyendo magnetoscopios, conmutadores, servidores de video y generadores de caracteres. Automatización Dayang es de gran versatilidad, y continuamente mejorado para cumplir con los requisitos del cliente. Este se ha vendido a más de 72 clientes de todo el mundo, lo que representa a más de 400 canales (19).

1.4.1.3 TVCONT ADDI

Es un sistema caracterizado por el control de hasta treinta y dos dispositivos externos RS-232⁵, entendiéndose como tales a magnetoscopios y visualizadores de logos. Brinda salida de video

⁵ RS-232: Recommended Standard 232, es una interfaz que designa una norma para el intercambio serie de datos binarios entre un equipo terminal de datos y un equipo de comunicación de datos.

Capítulo 1. Fundamentación teórica

compuesto y salida de audio estéreo balanceado. Existen dos versiones de TVCONT ADDI: BASIC y FULL. Como el software permite el control de hasta 32 dispositivos, a partir de la configuración FULL, se personaliza el hardware en función de lo que el cliente necesite. Además permite el consumo desde el mismo ordenador, posibilitando integrar el sistema con el módulo de TELEPRONTER (valor agregado del software) (20).

El estudio realizado a los sistemas de transmisión de señales digitales televisivas ha aportado una visión más clara de la línea a seguir durante el desarrollo del presente trabajo. Se han definido características que pueden tenerse en cuenta para la implementación de los subsistemas Transmisión y Administración de la Transmisión de la plataforma PTARTV. Además de un conjunto de elementos que no son factibles para la administración, entre los que se encuentran:

- ✓ Las aplicaciones similares estudiadas son todas propietarias y de un alto costo.
- ✓ Requieren de un hardware específico para su despliegue.
- ✓ Su despliegue debe ejecutarse sobre plataformas propietarias.

1.5 Conclusiones

Con el estudio de soluciones similares a la solución que se desea desarrollar, se ha obtenido una visión más clara de los elementos que deben tenerse en cuenta para su posterior construcción. Así como las características que debe cumplir un sistema informático hecho para transmitir televisión. Además, el estudio de algunas soluciones informáticas para TV arrojó que estas no constituyen un medio factible para la UCI porque son aplicaciones que tienen un alto costo en el mercado y no representan un factor adaptable a las condiciones de la DTU. Por lo que se debe proceder a la personalización de los *subsistemas de transmisión televisiva de la Plataforma PTARTV*.

Capítulo 2. Herramientas y tecnologías a utilizar

2.1 Introducción

En el presente capítulo se abordan las características fundamentales de las tecnologías a utilizar para dar solución al problema de la investigación, realizando un estudio del conjunto de tecnologías definidas en la arquitectura del sistema. Además se establece el estándar de codificación a usar en la construcción de la solución propuesta.

2.2 Metodología de desarrollo

Una metodología de desarrollo de software es un conjunto de procedimientos, técnicas, herramientas y soporte documental que guían el proceso de fabricación de una aplicación informática, arrojando un producto de calidad y en el tiempo requerido (21).

2.2.1 Proceso Unificado de Desarrollo de Software

RUP (por sus siglas en inglés) es una infraestructura flexible de desarrollo de software que proporciona prácticas recomendadas probadas. Se podrán alcanzar resultados predecibles unificando el equipo con procesos comunes que optimizan la comunicación y crean un entendimiento común para todas las tareas, responsabilidades y artefactos (22).

Es una de las metodologías de desarrollo de software más utilizada, es robusta, define quién, cómo, cuándo y qué debe realizarse en el proyecto. Es un proceso de software genérico que puede especializarse para una gran variedad de sistemas de software y diferentes tamaños de proyecto. Como lenguaje de modelado usa Lenguaje Unificado de Modelado, permitiendo preparar todos los esquemas de un sistema de software. Está dirigido por casos de uso ya que el proceso de desarrollo sigue una misma línea. Avanza a través de flujos de trabajos que parten de los casos de usos y es centrado en la arquitectura, pues incluye los aspectos estáticos y dinámicos más significativos del sistema. Además es iterativo e incremental, pues el trabajo se divide en pequeñas partes, lo que facilita el control de los pasos en el flujo de trabajo y beneficiando al avance y crecimiento del producto (23).

2.3 Lenguaje de modelado

En los años iniciales de la informática, los programadores no consumaban un análisis minucioso sobre el problema a resolver. Se poseía una pobre noción de técnicas de modelado y en el mejor de los casos se escribía el programa conforme se requería, sin tener en cuenta modelos que representaran el estilo y estructura que debían tener los códigos. Actualmente, los productos informáticos tienden a ser más complejos en cuanto a fabricación se trata. Por ello la necesidad de

Capítulo 2. Herramientas y tecnologías a utilizar

diseños sólidos ha traído consigo la creación de lenguajes y modelos que sirvan como pautas a seguir por los analistas, desarrolladores y clientes. Logrando unificar la visión y proyección del equipo de desarrollo hacia un producto de calidad.

2.3.1 Lenguaje Unificado de Modelado

UML (por sus siglas en inglés) es un lenguaje de modelado visual utilizado para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML incluye conceptos semánticos, notación, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. La especificación de UML no define un proceso estándar pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos (24).

2.4 Herramienta de Ingeniería de Software Asistida por Computadoras

Una herramienta CASE⁶ (por sus siglas en inglés) es un conjunto de programas y ayudas que brindan asistencia a los analistas, ingenieros de software y desarrolladores, durante el ciclo de desarrollo de un software (25).

2.4.1 Visual Paradigm

Visual Paradigm permite modelar todos los artefactos que se obtienen a partir del análisis del negocio y el sistema. Es una herramienta multiplataforma que soporta completamente el ciclo de desarrollo de un software: análisis y diseño, construcción, pruebas y despliegue. Posibilita el modelado de base de datos, requerimientos, proceso de negocio, permite realizar todo tipo de diagramas de clases, ingeniería inversa, generar código desde diagramas y generar documentación. Brinda funcionalidades como (26):

- ✓ Captura, especificación y análisis de requisitos mediante la creación de Modelos de Análisis Textual.

⁶ CASE: Computer Aided Software Engineering, en español(Ingeniería de Software Asistida por Computadoras).

Capítulo 2. Herramientas y tecnologías a utilizar

- ✓ A partir de las descripciones contenidas en los análisis textuales se pueden identificar los elementos candidatos a incluir en los diagramas de casos de uso.
- ✓ Una vez identificados estos elementos candidatos se pueden utilizar para conformar los diagramas de casos de uso.
- ✓ Es una herramienta colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto.
- ✓ Permite control de versiones.
- ✓ Genera la documentación del proyecto automáticamente en varios formatos como HTML o PDF.

2.5 Paradigma de programación

Puede describirse como el conjunto de ideas que indican una forma de programar. A través de ellos se pueden obtener diversas formas de "ver" y "pensar" un programa antes de escribirlo (27). Estas ideas han evolucionado a la par de los lenguajes de programación haciéndose muy común el uso de paradigmas en la construcción de aplicaciones informáticas. Un ejemplo vigente es la Programación Orientada a Objetos.

2.5.1 Programación Orientada a Objetos

La Programación Orientada a Objetos (POO), intenta simular el mundo real a través de objetos que contienen características y funciones. Se basa en la idea de un objeto, que es una combinación de variables y procedimientos llamados métodos que juntos conforman una entidad de programación (28). Para ello se utilizan un grupo de elementos que se detallan a continuación (29).

Clase: Es un tipo definido por el usuario que determina las estructuras de datos y las operaciones asociadas con ese tipo. Es posible especificar como una colección de objetos similares y un objeto constituye una instancia de la definición de una clase.

Encapsulación: Se usa para describir la combinación de estructuras de datos y de métodos que son manipulados por el objeto. La llamada a un objeto es lo que se denomina pasar un "aviso" a un objeto.

Abstracción: Es un término externo al objeto, que controla la forma en que es visto por los demás.

Modularidad: Un programa grande siempre será más complicado que la suma de varios programas pequeños, con lo que se considera ventajoso dividir un gran sistema en diversos módulos.

Jerarquía: Consiste en la clasificación y organización de las abstracciones según su naturaleza. El más claro ejemplo de jerarquía es la herencia.

Capítulo 2. Herramientas y tecnologías a utilizar

Herencia: Se puede definir como una jerarquía de abstracciones y la relación entre clases, donde se comparte la estructura y el comportamiento de una o más clases consideradas como clases superiores.

Polimorfismo: Habilidad de llamar de la misma forma a dos funciones o procedimientos diferentes. El programa decide qué función utilizar según los parámetros que recibe.

2.6 Lenguaje de programación

Se define como lenguaje de programación al elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que se pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes (30).

En el presente trabajo se formuló como **objetivo general**: *Migrar los subsistemas de transmisión televisiva de la plataforma PTARTV y personalizarlos para la DTU*. El trabajo con ficheros de audio y video exige altas velocidades de procesamiento. Una razón para que los lenguajes de programación que son reconocidos por su velocidad en tiempo de ejecución son recomendados para el desarrollo de aplicaciones que procesan video. La versión anterior de estos subsistemas fue desarrollada en el lenguaje Java, por lo que en este epígrafe se realiza una comparación entre Java y C++, este último fue el lenguaje escogido por los arquitectos del centro GEYSED para la construcción de sus componentes.

2.6.1 Java

Java es un lenguaje de programación interpretado, desarrollado por SUN Microsystems a principio de la década del 90. El objetivo de los diseñadores de Java fue crear un lenguaje orientado a objetos, sencillo, distribuido, seguro, robusto, con grandes prestaciones en cuanto a su rendimiento y que fuera multitarea. Está basado en C++. El lenguaje fue creado con la idea de desarrollar aplicaciones con un elevado nivel de fiabilidad, un ejemplo de ello se puede observar en las comprobaciones que realiza en tiempo de ejecución y durante la compilación del código. Es multiplataforma, posibilitando el trabajo en sistemas operativos diferentes y arquitecturas variadas. Es muy robusto ya que la Máquina Virtual de Java (JVM por sus siglas en inglés) está capacitada para revelar un elevado número de errores (31).

Aunque Java tiene como ventaja la portabilidad, el método que utiliza para la creación del código es ineficiente. Cuando Java es compilado y pasado a código en bytes, el intérprete correspondiente (la JVM), especialmente diseñado para una arquitectura de hardware, corre el programa. Al tener que interpretar el código, el proceso se hace más lento. El código compilado, muestra una serie de instrucciones que tienen que ver directamente con las instrucciones del

Capítulo 2. Herramientas y tecnologías a utilizar

procesador. Por lo tanto, el intérprete debe primero transcribir el código binario de java al equivalente de las instrucciones para el tipo de procesador. Este proceso toma algo de tiempo y no importa el tamaño del programa este será más lento que realizar la misma misma operación directamente en el código de la máquina (32).

2.6.2 C++

C++ es un lenguaje de programación basado en C que soporta directamente conceptos de la Orientación a Objetos. Comenzó su desarrollo en 1980 cuando Bjarne Stroustrup de los Bell Labs de AT&T necesitaba un lenguaje para desarrollar grandes programas de simulación. El resultado inicial fue un lenguaje denominado C con Clases. Este lenguaje continuó evolucionando y se le fueron añadiendo nuevos recursos, aunque manteniendo los recursos cercanos a la máquina que tiene C y el nombre fue cambiado por el de C++. Es por esto que es considerado un lenguaje híbrido, en él coexiste la forma clásica de programación en C con los recursos y conceptos de la POO (33).

Desde el punto de vista de la POO C++ ofrece una buena cantidad de recursos. Como son las formas de encapsulamiento, la sobrecarga de operadores y funciones, la herencia y el polimorfismo. Los mayores beneficios de C++ se apreciarán a largo plazo: alta calidad del software, mayor reutilización de código y mayor facilidad de adaptación, esta última apreciada sobre todo en grandes proyectos (33).

El lenguaje de programación C++ es uno de los más empleados en la actualidad. Puede emplearse mediante programación basada en eventos para crear programas que usen interfaz gráfica de usuario.

Este lenguaje se encuentra estandarizado a nivel internacional con el estándar ISO/IEC 14882:1998 con el título "Information Technology – Programming Languages - C++", publicado el 1 de septiembre de 1998 (34).

Las principales ventajas que presenta el lenguaje C++ son (34):

- ✓ Difusión: al ser uno de los lenguajes más empleados en la actualidad, posee un gran número de usuarios y existe una gran cantidad de libros, cursos y páginas web dedicados a su estudio.
- ✓ Versatilidad: C++ es un lenguaje de propósito general, por lo que se puede emplear para resolver cualquier tipo de problema.
- ✓ Portabilidad: está estandarizado y un mismo código fuente se puede compilar en diversas plataformas.
- ✓ Eficiencia: es uno de los lenguajes más rápidos en cuanto a tiempo de ejecución.

Capítulo 2. Herramientas y tecnologías a utilizar

- ✓ Herramientas: existe una gran cantidad de compiladores, depuradores y librerías para el trabajo con C++.

2.6.3 Selección del lenguaje de programación

Al realizar un análisis comparativo entre los lenguajes Java y C++, se pudo detectar que C++ es el que más se ajusta a las necesidades del proyecto PTARTV. Este lenguaje cuenta con grandes potencialidades además de ser compatible en varias plataformas incluyendo GNU/Linux. Es un lenguaje de programación libre para la presentación de mayor velocidad de ejecución en comparación con otros lenguajes como Java. La máquina virtual de Java es lenta. Java todavía no está al nivel del código nativo compilado de C++. El lenguaje C++ ha sido ampliamente adoptado por los desarrolladores y no desearán cambiar el lenguaje si las aplicaciones que ellos desarrollan con Java no están a la altura de sus estándares y los de sus clientes.

2.7 Marco de Trabajo

En el desarrollo de software un marco de trabajo o framework (del inglés) es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto (35).

2.7.1 Qt

Qt es un marco de trabajo multiplataforma para el desarrollo de aplicaciones e interfaces de usuario. Presenta un gran número características entre las que se destacan las siguientes: garantiza la portabilidad entre sistemas operativos embebidos y de escritorio, presenta herramientas de desarrollo integrado con entorno integrado de desarrollo multiplataforma y un alto desempeño en dispositivos embebidos. Entre las posibilidades que brinda este marco de trabajo están la del trabajo con hilos independientemente del sistema operativo, contiene un módulo para el trabajo con protocolos de red, soporte para aplicaciones orientadas a componentes y trabajo con los gestores de bases de datos más conocidos. Hasta el momento ha sido liberado bajo dos licencias, la LGPL y la comercial (36).

2.8 Entorno Integrado de Desarrollo

Un entorno integrado de desarrollo (en inglés Integrated Development Environment o IDE) es un programa compuesto por una serie de herramientas que utilizan los programadores para escribir códigos. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos. Las herramientas que normalmente componen un entorno de desarrollo integrado son las siguientes: un editor de texto, un compilador,

un intérprete, unas herramientas para la automatización, un depurador, un sistema de ayuda para la construcción de interfaces gráficas de usuario y, opcionalmente, un sistema de control de versiones (37).

2.8.1 QT Creator

Qt Creator es un entorno integrado de desarrollo (IDE) multiplataforma adaptado a las necesidades de los desarrolladores. Permite a los desarrolladores crear aplicaciones de escritorio y plataformas de dispositivos móviles. A continuación se detallan sus características más relevantes (38):

✓ Editor de código sofisticado

Avanzado editor de código de Qt Creator provee soporte para edición de C++, sensible al contexto de ayuda y completamiento de código.

✓ El control de versiones

Qt Creator se integra con la mayoría de los sistemas de control de versiones populares, incluyendo Git, Subversion, Perforce, CVS y Mercurial.

✓ Los diseñadores de interfaz de usuario

Qt Creator ofrece el editor visual integrado: Qt Diseñador, muy intuitivo para el desarrollo de interfaces.

✓ Proyecto y Construcción Gestión

Si se importa un proyecto existente o se crea uno desde cero, Qt Creator genera todos los archivos necesarios.

✓ Escritorio y objetivos móviles

Qt Creator ofrece soporte para crear y ejecutar aplicaciones para equipos de escritorio y dispositivos móviles.

2.9 Sistema Gestor de Base de Datos

Un Sistema Gestor de base de datos (en inglés database management system o DBMS) es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. Por tanto debe permitir (39):

- ✓ Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- ✓ Construir la base de datos: guardar los datos en algún medio controlado por el mismo DBMS
- ✓ Manipular la base de datos: realizar consultas, actualizarla y generar informes.

2.9.1 PostgreSQL

El DBMS relacional orientado a objetos conocido como PostgreSQL cuenta con más de 15 años de desarrollo activo y una arquitectura probada. Se ha ganado una sólida reputación de fiabilidad, corrección e integridad de datos. Tiene soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados. Incluye una gran cantidad de tipos de datos, incluyendo INTEGER, BOOLEAN, CHAR, VARCHAR, DATE e INTERVAL. Además de soportar almacenamiento de objetos binarios grandes, como imágenes, sonidos o video (40). Al contar con interfaces nativas de programación para C / C + +, Java, .Net, Perl, Python, Ruby, entre otros, es uno de los DBMS más conocido y usado. Por esta razón la documentación y el número de sitios destinados a él es muy amplio.

2.10 Biblioteca Libvlc

La biblioteca Libvlc representa la API⁷ subyacente del reproductor Video LAN (VLC). El mismo es un simple envoltorio de acceso a Libvlc. Los desarrolladores pueden utilizar la biblioteca con el objetivo de explotar las complejas funcionalidades implementadas por Video LAN. Las mismas brindan la posibilidad de reproducir contenido audiovisual y hacer streaming del mismo, lo que es explotado por el subsistema Transmisión para realizar las emisiones televisivas. Libvlc se distribuye de forma compartida, lo que permite al desarrollador de aplicaciones acceder a las funcionalidades del reproductor y usarlas a su conveniencia (41).

2.11 Conclusiones

Para fabricar una aplicación informática debe tenerse en cuenta cuál es la tecnología más apropiada, no se puede seguir esta pauta sino se tiene la información más actualizada sobre este tema. Cada día las demandas del mercado del software son más exigentes, el número de herramientas y tecnologías para la fabricación de aplicaciones aumenta vertiginosamente. Lo que obliga a los desarrolladores a realizar estudios minuciosos sobre qué usar y por qué. Una vez concluida la descripción de las tecnologías definidas por los arquitectos del centro GEYSED y específicamente PTARTV, los programadores de PTARTV concuerdan con la decisión de los arquitectos de software. La utilización de un marco de trabajo y un entorno integrado de desarrollo especialmente diseñados para trabajar en conjunto facilitará la codificación, así como explotar el conjunto de funcionalidades que su relación provee. Es muy importante mencionar que aunque las tecnologías escogidas poseen un alto grado de profesionalidad, no sería una mala práctica realizar periódicamente un estudio del soporte arquitectónico de las soluciones que ofrece el centro

⁷ API: Application Programming Interface (en español “interfaz de programación de aplicaciones”).

Capítulo 2. Herramientas y tecnologías a utilizar

GEYSED, ya que constantemente las bibliotecas y herramientas se actualizan y en la mayoría de los casos simplifican operaciones que anteriormente tenían un mayor grado de complejidad.

Capítulo 3. Descripción y validación de la solución propuesta

3.1 Introducción

Para migrar y personalizar los *subsistemas Transmisión y Administración de la Transmisión de la plataforma PTARTV* se realizó un estudio detallado de la versión anterior de estos subsistemas. En el mismo se detectó que el nivel de responsabilidad que tenía el subsistema *Administración de la Transmisión* se limitaba solo a insertar, eliminar o modificar los datos de un canal en el sistema. Debido a esto se decidió en la solución propuesta, centralizar todas las funcionalidades de ambos subsistemas en un único subsistema llamado “Transmisión”. Al fundir ambos subsistemas, se hace menor el número de ventanas mostradas por la aplicación y se crea un ambiente más agradable para el usuario. En este capítulo se realizará una breve descripción de dicho subsistema así como su representación en el modelo de implementación y por componentes. Partiendo de los diagramas de componentes se estructura el modelo de implementación, el mismo trata de mostrar el funcionamiento de todos los subsistemas de un software funcionando como un todo. También se mostrará el estándar de codificación que será usado en la construcción de la solución propuesta y se representarán un conjunto de elementos que permiten validar la solución que en este capítulo se describe.

3.2 Descripción del subsistema Transmisión

El subsistema Transmisión es el encargado de efectuar todo el proceso de transmisión de un material, así como tener el control sobre el número de canales con los que desee trabajar el usuario. El mismo es capaz de conformar las listas de reproducción correspondientes a las planificaciones de transmisión, así como iniciar la emisión de un canal de forma automática o manual (de acuerdo a la elección del usuario). Una vez puesta en marcha una emisión el transmisor es capaz de parar, pausar o reanudar la misma. El sistema también brinda la posibilidad de visualizar la señal de salida de un canal determinado, sirviendo como opción de monitoreo. La aplicación podrá funcionar teniendo un Set-top box por cada canal que exista, en caso contrario y adaptándose a las condiciones actuales de DTU se provee al cliente de una aplicación que visualizará a pantalla completa una transmisión, en este caso la señal se tomaría de la tarjeta de video instalada en la estación que vaya a usarse como cliente de transmisión. En la interfaz de usuario se visualiza todo tipo de información referente a los canales con transmisión, por ejemplo la dirección de difusión por la que se realiza la transmisión, el nombre del canal, la lista de reproducción correspondiente o si está en emisión. Este subsistema también es el

encargado de gestionar la información de los canales, considérese esto como la inserción, modificación, o eliminación de los datos de un canal.

3.3 Diagrama de componentes

Al utilizar la metodología RUP debe conocerse que la misma hace énfasis en el uso de diagramas de componentes para modelar las dependencias entre los componentes de los subsistemas de implementación.

“Los diagramas de componentes describen los elementos del sistema y sus relaciones. Los componentes representan todos los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, bibliotecas cargadas dinámicamente, entre otros. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente (42).”

Los principales estereotipos estándares que presentan los componentes dentro de un modelo son

- <<executable>>
Representa programas que se ejecutan en un nodo.
- <<file>>
Son ficheros de datos o código fuente.
- <<library>>
Modelan librerías estáticas o dinámicas.
- <<table>>
Constituyen tablas de bases de datos.
- <<document>>
Simbolizan documentos.

✓ Diagrama de componentes para el subsistema Transmisión.

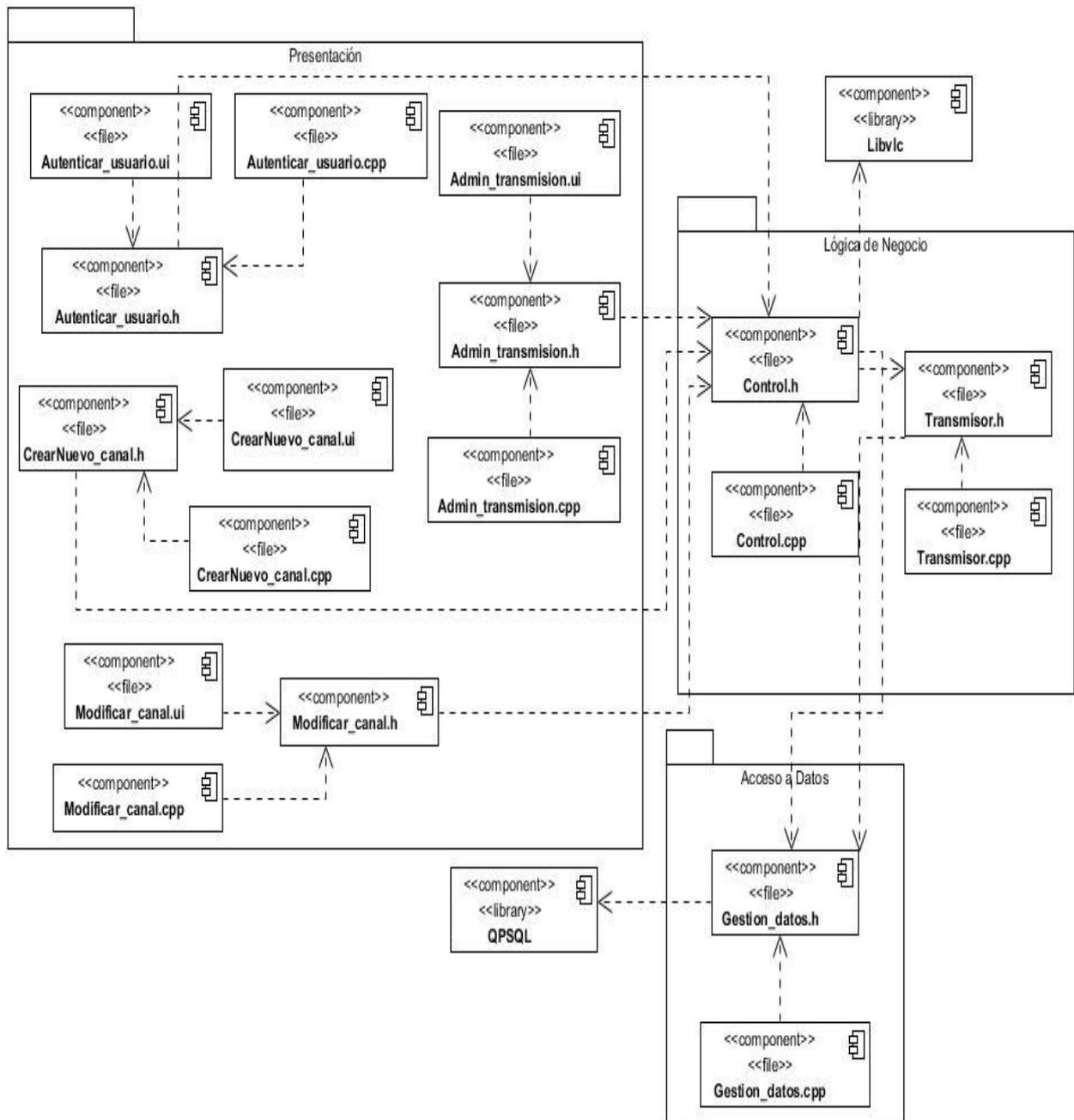


Ilustración 1. Diagrama de componentes para el subsistema Transmisión.

3.4 Modelo de implementación

El modelo de implementación representa la composición física de la implementación en términos de subsistemas de implementación y elementos de implementación. Identifica los componentes físicos de la implementación para que puedan comprenderse y gestionarse mejor (43).

Capítulo 3. Descripción y validación de la solución propuesta

A continuación se muestra el modelo de implementación correspondiente a la plataforma PTARTV. Puede apreciarse la relación entre el subsistema Transmisión y los subsistemas Reporte y Seguridad. En el caso de Reporte será el subsistema encargado de brindar todos los reportes relacionados con la transmisión. Seguridad es el subsistema encargado de autenticar a los usuarios y concederles acceso o no al sistema de transmisión.

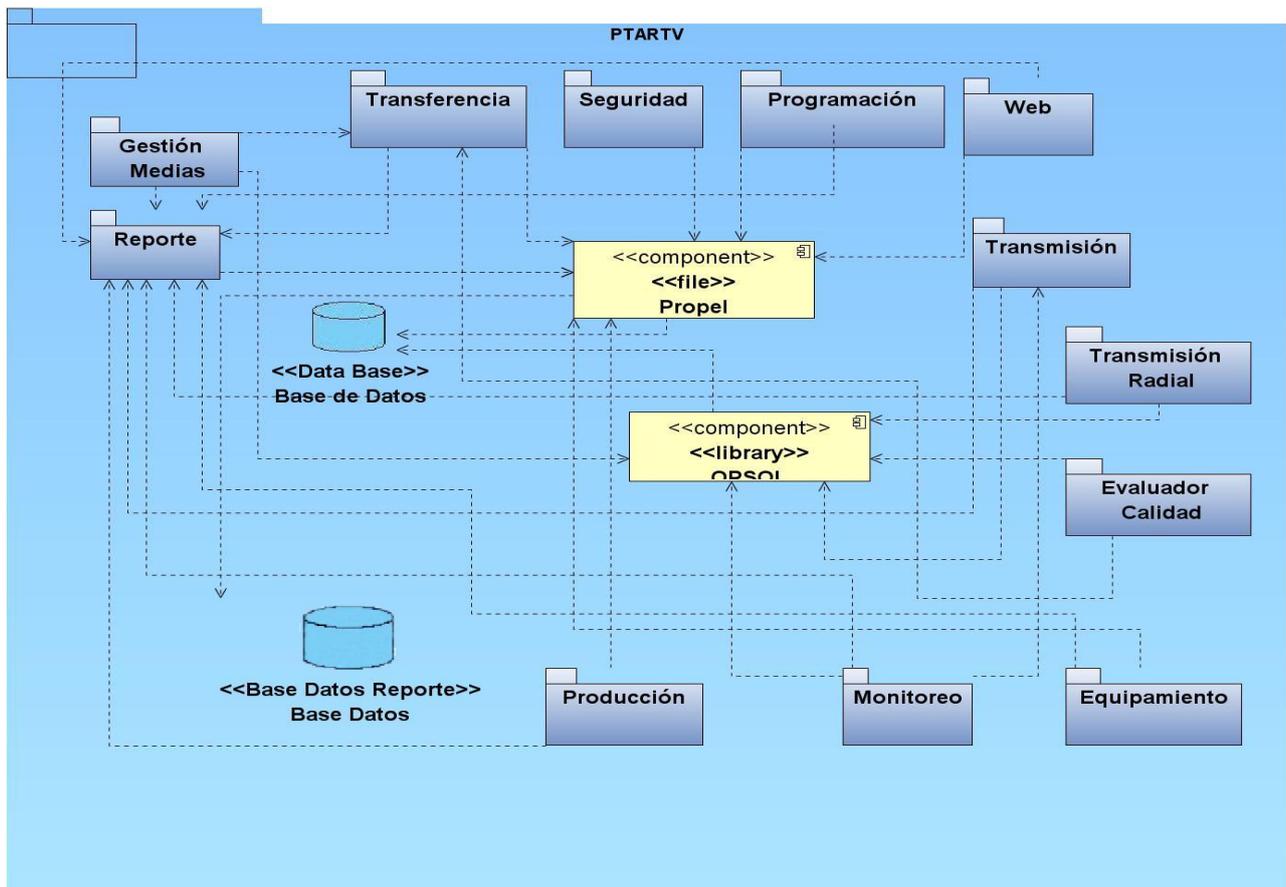


Ilustración 2. Modelo de implementación.

3.5 Estándar de codificación.

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. A continuación se presentan las convenciones establecidas por el centro GEYSED para la programación en lenguaje C++.

3.5.1 Comentarios

Cada programa deberá comenzar con un comentario que incluya:

- ✓ Autor.

Capítulo 3. Descripción y validación de la solución propuesta

- ✓ Fecha.
- ✓ Objetivo, o problema que resuelve el programa.
- ✓ Fecha de creación y bitácora de versiones con las dos últimas fechas de modificación.
- ✓ Algoritmo.

Cada función debe tener un encabezado que contenga:

- ✓ Objetivo de la función y no descripción del procedimiento.
- ✓ Comentarios de apoyo a variables, llamadas a función o inclusión de archivos que no sean obvios al proceso.
- ✓ Explicación de uso de argumentos (parámetros) no obvios.
- ✓ Explicación de uso de valores devueltos (de retorno).

3.5.2 Nombres de identificadores

Se considera como identificador a los nombres de variables (arreglos, matrices, apuntadores), funciones, así como cualquier tipo de dato definido por el usuario (estructura, clase). Dichos identificadores deberán seguir las siguientes normas, además de las definidas por el propio lenguaje.

- ✓ Deberán tener un nombre significativo para que por su simple lectura, pueda conocerse su función, sin tener que consultar manuales o hacer demasiados comentarios.
- ✓ Para nombres que se usen con frecuencia o para términos largos, se recomienda usar abreviaturas estándares para que éstos tengan una longitud razonable. Si se usa abreviaturas deben manejarse la misma lógica en todo el programa.
- ✓ Evitar identificadores que comiencen con uno o dos caracteres de subrayado para evitar que se confundan con los que el compilador selecciona.
- ✓ Cada identificador de función, variable o procedimiento deberá ser precedido por la abreviación del tipo de dato de que es la variable, o si se trata de una función o procedimiento del tipo de dato que regresa.

3.5.2.1 Identificadores de variables

Comenzarán siempre con la primera letra minúscula correspondiente a su tipo de dato.

Para distinguir palabras dentro del nombre deberá emplearse una letra mayúscula o un guión bajo (_), sin mezclar ambas formas en un mismo programa.

Ejemplos:

temperatura_de_vapor

3.5.2.2 Identificadores de punteros (apuntadores)

Su nombre deberá comenzar con la letra p.

Ejemplo: p_obj_alumno

Dónde *p_obj_alumno* es un puntero que podrá tener la dirección del lugar donde se almacena información de un alumno.

Tipo de dato en abreviaturas

- ✓ integer I
- ✓ float f .
- ✓ double d.
- ✓ arreglo ar.
- ✓ char c
- ✓ enumeración e
- ✓ estructura st
- ✓ constantes todas las letras del identificador con mayúsculas.
- ✓ punteros p

3.5.2.3 Identificadores de variables dimensionadas(arreglos, matrices)

Su nombre deberá comenzar con las letras ar.

Ejemplo: ar_alumnos

Donde *ar_alumnos* es un arreglo de datos para guardar información de alumnos.

3.5.2.4 Identificadores de datos constantes

Serán declaradas en letras mayúsculas.

Ejemplo: const IVA=0.15;

3.5.2.5 Identificadores de funciones

La primera letra deberá ser mayúscula.

Ejemplo: void Funcion();

3.5.2.6 Identificadores de tipos definidos por el usuario

La primera letra será mayúscula.

Ejemplo: class Clase o struct Estructura

3.5.3 Organización Visual del Programa

3.5.3.1 Generales

- ✓ No manejar en los programas más de una instrucción por línea.
- ✓ Declarar las variables en líneas separadas
- ✓ Añadir comentarios descriptivos junto a cada declaración de variables, si es necesario.

3.5.3.2 Sangrías

- ✓ Las sangrías tendrán una longitud de tres espacios.
- ✓ Para las llaves que definen el cuerpo de una función, se dejará un nivel de sangría.

Ejemplo: void Funcion ()

```
{//Instrucciones de la función}
```

- ✓ Se dejará sangría para las instrucciones del cuerpo de cada estructura de control.

Ejemplo: for (int x = 0; x < 5; x++)

```
{//Instrucciones a ejecutar}
```

- ✓ Se evitará codificar más de tres niveles de sangría.

3.5.3.3 Líneas y espacios en blanco

- ✓ Insertar una línea en blanco antes y después de una declaración de datos que aparezca entre instrucciones ejecutables.

```
a=b+c;
```

```
//línea en blanco
```

```
int f; //declaración entre instrucciones
```

```
// línea en blanco
```

```
f = a;
```

- ✓ Las declaraciones de datos dentro de una función, deberán ir al inicio y separadas de las instrucciones ejecutables de la función por medio de una línea en blanco.
- ✓ Deben incluirse espacios en ambos lados de los operadores binarios.

Ejemplo:

```
y = 50 + 15 - x;
```

Capítulo 3. Descripción y validación de la solución propuesta

- ✓ Es posible distribuir una instrucción grande sobre varias líneas. Si lo hace, seleccione puntos de ruptura que tengan sentido, como después de una coma en el caso de una lista, o después de un operador en el caso de una expresión larga.

Ejemplo:

```
cout << "Ejemplo de ruptura de una instrucción en más de una"  
<< " línea de comandos";
```

- ✓ Los operadores unarios (++ , -- , etc.) deben ponerse junto a sus operandos, sin espacios intermedios.
- ✓ Antes y después de cada estructura de control deberá poner una línea en blanco.

3.5.3.4 Paréntesis

- ✓ Para hacer más clara una expresión, es aceptable agregarle paréntesis innecesarios. Dichos paréntesis se llaman paréntesis redundantes.

3.6 Validación de la solución propuesta

La prueba de una aplicación informática es un elemento fundamental para avalar la eficacia del software y representa una revisión de las especificaciones, del diseño y de la codificación de los elementos que componen la aplicación. Existen varias tipologías de pruebas, cada una de ellas con metas y estrategias bien definidas. En el presente capítulo se describirán las pruebas funcionales y de rendimiento realizadas al subsistema Transmisión.

3.6.1 Elementos del proceso de pruebas

Cuando se van a realizar las pruebas es necesario analizar cuáles son los elementos que hay que tener en cuenta para realizar una prueba, como los niveles y tipos de pruebas a realizar. A continuación se mostrarán dichos elementos.

3.6.2 Niveles de pruebas

Prueba de desarrollador: Indica los aspectos de diseño e implementación de las pruebas más adecuadas que debe llevar a cabo el equipo de desarrolladores, a diferencia de la prueba independiente. En la mayoría de los casos, la ejecución de la prueba se produce inicialmente con el grupo de pruebas de desarrollador que la diseñó e implementó; aunque es recomendable que los desarrolladores creen las pruebas de forma que estén disponibles para que las ejecuten grupos de pruebas independientes (43).

Capítulo 3. Descripción y validación de la solución propuesta

Prueba independiente: Indica el diseño y la implementación de la prueba realizada más adecuadamente por alguien ajeno al equipo de desarrolladores. Puede considerar esta distinción un súper conjunto, que incluye validación y verificación independientes. En la mayoría de los casos, la ejecución de la prueba se produce inicialmente con el grupo de pruebas independientes que la diseñó e implementó; aunque los verificadores independientes deberían crear sus pruebas de forma que estén disponibles para que las ejecuten los grupos de pruebas de desarrollador (43).

Prueba de unidad: Se centra en la verificación de los elementos más pequeños del software que se puedan probar. Normalmente, las pruebas de unidad se aplican a componentes representados en el modelo de implementación para verificar que se cubren los flujos de control y los flujos de datos y que funcionan como se esperaba. El implementador realiza la prueba de unidad mientras se desarrolla la unidad. Los detalles de la prueba de unidad se describen en la disciplina de implementación (43).

Prueba de integración: Las pruebas de integración se realizan para garantizar que los componentes del modelo de implementación funcionan correctamente cuando se combinan para ejecutar un guión de uso. El destino de la prueba es un paquete o un conjunto de paquetes del modelo de implementación. A menudo, los paquetes que se combinan proceden de diferentes empresas de desarrollo. Las pruebas de integración exponen el estado incompleto o los errores de las especificaciones de la interfaz del paquete (43).

Prueba del sistema: Normalmente, la prueba del sistema se realiza cuando el software funciona en su totalidad. Un ciclo vital repetitivo permite que las pruebas del sistema se realicen mucho antes, en cuanto se hayan implementado subconjuntos bien formados del comportamiento de guiones de uso. Normalmente, el destino son los elementos en funcionamiento de extremo a extremo del sistema (43).

Prueba de aceptación: La prueba de aceptación del usuario es la última acción de prueba antes de desplegar el software. El objetivo de la prueba de aceptación es comprobar si el software está preparado y lo pueden utilizar los usuarios para realizar las funciones y tareas para las que se diseñó (43).

Después de un análisis de los niveles de prueba por los que puede pasar una aplicación informática, se debe tener conocimiento de cuales son los tipos de pruebas que se aplican en cada uno de estos niveles.

3.6.3 Tipos de prueba

Tipos de Pruebas que se aplican al software (43)

Capítulo 3. Descripción y validación de la solución propuesta

Dimensión de calidad	Tipo de pruebas
Funcionalidad	<p><i>Prueba de función:</i> pruebas que se centran en la validación de funciones del destino de la prueba, proporcionan los guiones de uso, los métodos y los servicios necesarios. Esta prueba se implementa y se ejecuta en diferentes destinos de la prueba, incluidas las unidades, las unidades integradas, las aplicaciones y los sistemas.</p>
	<p><i>Prueba de seguridad:</i> pruebas que se centran en garantizar que los datos del destino de la prueba (o sistemas) sólo son accesibles para los actores a los que se dirigen. Esta prueba se implementa y ejecuta en varios destinos de la prueba.</p>
	<p><i>Prueba de volumen:</i> pruebas que se centran en la verificación de la capacidad del destino de la prueba para manejar grandes cantidades de datos, ya sean de entrada y salida o residentes, en la base de datos. La prueba de volumen incluye estrategias de prueba como la creación de consultas que devolverán el contenido completo de la base de datos, o que tendrán tantas restricciones que no devolverán ningún dato, o en las que la entrada de datos tiene la cantidad máxima de datos para cada campo.</p>
Fiabilidad	<p><i>Prueba de integridad:</i> pruebas que se centran en la evaluación de la fuerza del destino de la prueba (resistencia a los errores) y la conformidad técnica del lenguaje, la sintaxis y la utilización de recursos. Esta prueba se implementa y se ejecuta en diferentes destinos de la prueba, incluidas las unidades y las unidades integradas.</p>
	<p><i>Prueba de estructura:</i> pruebas que se centran en la evaluación de la adherencia del destino de la prueba a su diseño y formación. Normalmente, esta prueba se realiza en aplicaciones habilitadas para web y garantiza que todos los enlaces están conectados, se muestra el contenido adecuado y no hay ningún contenido huérfano.</p>
	<p><i>Prueba de tensión:</i> se trata de un tipo de prueba de fiabilidad que se centra en la evaluación de cómo responde el sistema en circunstancias anormales. Las tensiones del sistema pueden ser cargas de trabajo extremas, memoria insuficiente, servicios y hardware no disponible o recursos compartidos limitados. Estas pruebas suelen realizarse para saber mejor cómo y en qué áreas fallará el sistema, de forma que se puedan planificar y presupuestar los planes de contingencia y el mantenimiento de las actualizaciones con bastante antelación.</p>
	<p><i>Prueba de puntos de referencia:</i> se trata de un tipo de prueba de rendimiento que compara el rendimiento de un destino de la prueba nuevo o desconocido con una referencia conocida, carga de trabajo y sistema.</p>
	<p><i>Prueba de contienda:</i> pruebas que se centran en la validación de la capacidad del</p>

Capítulo 3. Descripción y validación de la solución propuesta

	destino de la prueba para manejar de forma aceptable varias demandas del actor en el mismo recurso (registros de datos, memoria, etc.).
Rendimiento	<i>Prueba de carga:</i> se trata de un tipo de prueba de rendimiento que se utiliza para validar y evaluar la aceptabilidad de los límites operativos de un sistema bajo cargas de trabajo variables, mientras el sistema que se está probando permanece igual. En algunas variantes, la carga de trabajo permanece igual y se modifica la configuración del sistema que se está probando. Las medidas suelen tomarse en función del rendimiento de la carga de trabajo y el tiempo de respuesta de las transacciones en línea. Las variaciones de la carga de trabajo suelen incluir la emulación del pico y el promedio de cargas de trabajo que se producen dentro de la tolerancia operativa normal.
	<i>Perfil de rendimiento:</i> se trata de una prueba en la que se controla el perfil de tiempo del destino de la prueba, incluidos el flujo de la ejecución, el acceso de datos, las llamadas del sistema y de funciones para identificar y tratar los cuellos de botella de rendimiento y los procesos ineficaces.
	<i>Prueba de configuración:</i> pruebas que se centran en garantizar que las funciones del destino de la prueba son las adecuadas en diferentes configuraciones de hardware y software. Esta prueba también se puede implementar como una prueba de rendimiento del sistema.
Capacidad de soporte	<i>Prueba de instalación:</i> pruebas que se centran en garantizar que el destino de la prueba se instala correctamente en diferentes configuraciones de hardware y software, y en condiciones diferentes (como, por ejemplo, espacio de disco insuficiente o interrupciones de la alimentación). Esta prueba se implementa y ejecuta en aplicaciones y sistemas.

Tabla 1. Tipos de prueba que se le aplican a un software.

Al conocer los niveles y tipos de pruebas que pueden aplicarse a un producto de software, se puede seleccionar cuál es el tipo de prueba que se aplicará para comprobar la calidad de la solución construida. Finalmente se decidió aplicar pruebas a nivel de unidad (pruebas de caja blanca), pruebas funcionales y pruebas de rendimiento.

3.6.4 Pruebas de caja blanca

Las pruebas de caja blanca ejercitan que se cumplan todos los caminos lógicos del programa. Con la aplicación de la técnica de camino básico se realiza un grafo de flujo del código fuente seleccionado previamente.

Capítulo 3. Descripción y validación de la solución propuesta

Los pasos para la realización de esta técnica son los siguientes:

Paso 1. Generar el grafo de flujo de datos a partir del código/diseño procedimental (Nodos, Aristas, Regiones)

Paso 2. Calcular la complejidad ciclomática $V(G)$: métrica del software que da una medición de la complejidad de un programa.

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

$$V(G) = P \text{ (Nodos predicados)} + 1$$

$$V(G) = \text{Número de regiones}$$

Paso 3. Definir un conjunto básicos de caminos de ejecución a partir de la complejidad ciclomática.

Paso 4. Generar un Caso de Prueba para cada camino de ejecución.

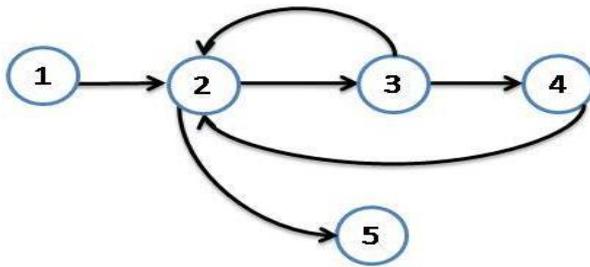
3.6.5 Diseño de pruebas de caja blanca

A continuación se muestran los diseños de casos de prueba de caja blanca para las funciones críticas implementadas en la solución a probar.

✓ Función 1. Obtener la posición de un transmisor en la lista de transmisores.

```
int control::Posicion_Transmisor_Lista(int id_canal)
{
1  int i_pos_salida=-1;
2  for(int i=0;i<lista_transmisores.length();i++)
   {
3   if(lista_transmisores.at(i)->ID_Canal()==id_canal)
      {
4     i_pos_salida=i;
      }
   }
5  return i_pos_salida;
}
```

Paso 1: Generar grafo de flujo de datos.



Paso 2: Cálculo de complejidad ciclomática.

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

$V(G) = 6 - 5 + 2 = 3$

Paso 3: Caminos básicos.

CB1: 1 2 3 4 2 5

CB2: 1 2 5

Paso 4: Caso de prueba para el camino básico.

CB1: 1 2 3 4 2 5

Caso de prueba: Posición de un Transmisor en la lista.

Entrada: id_canal=2.

Resultado esperado: 3.

Resultado de la prueba: Satisfactoria.

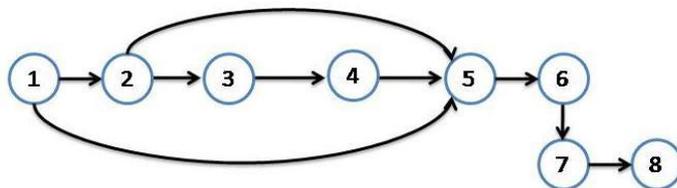
- ✓ **Función 2. Actualizar el valor de la variable que permite saber la posición en la lista de reproducción de la media que se está transmitiendo.**

```
void transmisor::Actualizar_Pivote(bool incrementar)
{
1  if(incrementar)
    {
2  if(i_pivote<lista_reproduccion.length()-1)
    {
3  i_pivote++;
4  emit sPivote_Incrementado();
    }
    }
}
```

Capítulo 3. Descripción y validación de la solución propuesta

```
5 if(!incrementar)
  {
6   if(i_pivote>0)
    {
7     i_pivote--;
8     emit sPivote_Decrementado();
    }
  }
}
```

Paso 1: Generar Grafo de flujo de datos.



Paso 2: Cálculo de complejidad ciclomática.

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

$V(G) = 9 - 8 + 2 = 3$

Paso 3: Caminos básicos.

CB1: 1 2 3 4 5

CB2: 1 2 5

CB3: 1 5 6 7 8

CB4: 1 5 6

Paso 4: Caso de prueba para el camino básico.

CB1: 1 2 3 4 5

Caso de prueba: Actualizar pivote.

Entrada: incrementar = true.

Resultado esperado: i_pivote= 6.

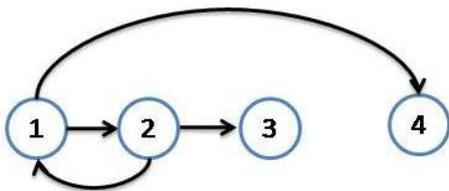
Resultado de la prueba: Satisfactoria.

✓ **Función 3. Saber si un canal se encuentra en la lista de transmisores.**

Capítulo 3. Descripción y validación de la solución propuesta

```
bool control::Buscar_Transmisor_Lista(int id_canal)
{
1  for(int i=0;i<lista_transmisores.length();i++)
    {
2      if(lista_transmisores.at(i)->ID_Canal()==id_canal)
        {
3          return true;
        }
    }
4  return false;
}
```

Paso 1: Generar Grafo de flujo de datos.



Paso 2: Cálculo de complejidad ciclomática.

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

$V(G) = 4 - 4 + 2 = 2$

Paso 3: Caminos básicos.

CB1: 1 2 3

CB2: 1 2 1 4

CB3: 1 4

Paso 4: Caso de prueba para el camino básico.

CB1: 1 2 3

Caso de prueba: Buscar un transmisor en la lista.

Entrada: id_canal=6.

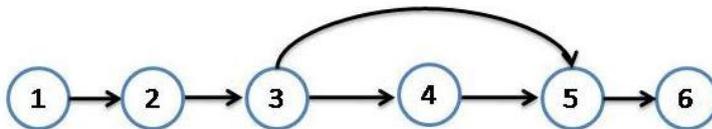
Resultado esperado: false.

Resultado de la prueba: Satisfactoria.

✓ Función 4. Iniciar una transmisión.

```
void control::Iniciar_Transmision(transmisor *p_obj)
{
1  p_obj->Transmitir();
2  lista_transmisores.append(p_obj);
3  if(!lista_transmisores.isEmpty())
   {
4    connect(lista_transmisores.last(), SIGNAL(sTransmision_Finalizada(int)), this,
SIGNAL(sCanal_Transmision_Finalizada(int)));
   }
5  emit sCanal_Transmitiendo(p_obj->ID_Canal());
6  p_obj_gestion_datos->Cambiar_Estado_Canal(p_obj->ID_Canal(),1);
}
```

Paso 1: Generar Grafo de flujo de datos.



Paso 2: Cálculo de complejidad ciclomática.

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

$V(G) = 6 - 6 + 2 = 2$

Paso 3: Caminos básicos.

CB1: 1 2 3 4 5 6

CB2: 1 2 3 5 6

Paso 4: Caso de prueba para el camino básico.

CB1: 1 2 3 4 5 6

Caso de prueba: Iniciar transmisión.

Entrada: puntero a un objeto de la clase transmisor.

Resultado esperado: puesta en marcha de una transmisión.

Resultado de la prueba: Satisfactoria.

3.6.6 Pruebas funcionales

Se denominan pruebas funcionales a las pruebas de software que tienen por objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados, es común que este tipo de pruebas sean desarrolladas por analistas de pruebas con apoyo de algunos usuarios finales, esta etapa suele ser la última etapa de pruebas y al dar conformidad sobre esta el paso siguiente es el pase a producción (44).

A este tipo de pruebas se les denomina también pruebas de comportamiento o pruebas de caja negra, ya que los analistas de pruebas, no enfocan su atención a cómo se generan las respuestas del sistema, básicamente el enfoque de este tipo de prueba se basa en el análisis de los datos de entrada y en los de salida, esto generalmente se define en los casos de prueba preparados antes del inicio de las pruebas (44).

Las pruebas de caja negra se centran en los requisitos funcionales de software, permitiendo obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Este tipo de prueba intenta encontrar errores de las siguientes categorías:

- Función incorrecta o ausente
- Errores de interfaz
- Errores en estructura de datos
- Errores de rendimiento
- Errores de inicialización y terminación.

3.6.7 Diseño de pruebas funcionales

Con el objetivo de validar el correcto funcionamiento de las interfaces de usuario del subsistema Transmisión se realizaron pruebas de caja negra a los casos de uso Gestionar canal y Autenticar usuario.

A continuación se relacionan los casos de pruebas para los casos de uso seleccionados:

3.6.7.1 Secciones a probar en el caso de uso Gestionar canal

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

Capítulo 3. Descripción y validación de la solución propuesta

SC 1: Crear nuevo canal	EC 1.1: Crear nuevo canal satisfactoriamente	El sistema verifica la validez de los datos y que no existan campos vacíos de los requeridos por el formulario y luego inserta los datos del canal en la base de datos.
	EC 1.2: Crear nuevo canal falla.	El sistema verifica la no validez de los datos requeridos en el formulario y no se adiciona el canal a la base de datos. Muestra un mensaje informando el error que se ha producido.
SC 2: Modificar datos de un canal	EC 2.1: Modificar datos de un canal satisfactoriamente.	El sistema verifica que el canal no esté en transmisión y de no ser así permite la modificación de los datos. Posteriormente salva los datos en la base de datos, muestra un mensaje indicando que la acción fue ejecutada satisfactoriamente.
	EC 2.2: Modificar datos de un canal falla	Si el canal está en transmisión el sistema muestra un mensaje indicando que no se pueden modificar sus datos, de lo contrario el sistema valida que los datos entrados sean correctos y que no existan campos vacíos, de existir datos incorrectos o campos vacíos el sistema muestra un mensaje informando el error ocurrido y no actualiza los cambios en la base de datos.

Tabla 2. Secciones a probar en el caso de uso Gestionar canal.

Capítulo 3. Descripción y validación de la solución propuesta

3.6.7.2 Descripción de variables para el caso de uso Gestionar canal

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	nombre	Campo de texto (alfanumérico)	No	El usuario introducirá en el campo el nombre del canal.
2	puerto	Campo numérico	No	El usuario introducirá en el campo el puerto del canal.
3	dirección IP	Campo numérico	No	El usuario introducirá en el campo la dirección IP del canal.
4	url del logo		No	
5	descripción		No	

Tabla 3. Descripción de variables para el caso de uso Gestionar canal.

3.6.7.3 Matriz de datos para la sección Crear nuevo canal

Escenario	nombre	puerto	dirección IP	url del logo	descripción	Respuesta del sistema	Resultado de la prueba
Crear nuevo canal satisfactoriamente.	V	V	V	V	V	El sistema valida que los datos introducidos por el usuario sean correctos, inserta los mismos en la base de datos y muestra un mensaje indicando que la acción fue ejecutada satisfactoriamente.	Insatisfactorio.
Crear nuevo canal falla.	I	V	V	V	V	El sistema valida los datos introducidos por	Satisfactorio.

Capítulo 3. Descripción y validación de la solución propuesta

	V	I	V	V	V	el usuario y:	
	V	V	I	V	V	Si los datos introducidos por el usuario están incorrectos el sistema muestra un mensaje indicando el error ocurrido.	
	V	V	V	I	V	Si existen campos obligatorios vacíos el sistema muestra un mensaje indicando que no pueden existir campos vacíos.	
	V	V	V	V	I		

Tabla 4. Matriz de datos para la sección Crear nuevo canal.

3.6.7.4 Matriz de datos para la sección Modificar canal

Escenario	nombre	puerto	dirección IP	url del log	descripción	Respuesta del sistema	Resultado de la prueba
Modificar canal satisfactoriamente.	V	V	V	V	V	El sistema valida que los datos introducidos por el usuario sean correctos, inserta los mismos en la base de datos y muestra un mensaje indicando que la acción fue	Satisfactorio

Capítulo 3. Descripción y validación de la solución propuesta

						ejecutada satisfactoriamente.	
Modificar canal falla.	I	V	V	V	N/A	El sistema valida que el canal seleccionado por el usuario no esté en transmisión, de ser así muestra un mensaje indicando que no se pueden modificar los datos. Si el canal no está en transmisión el sistema valida los datos modificados por el usuario y: Si los datos introducidos por el usuario están incorrectos el sistema muestra un mensaje indicando el error ocurrido. Si existen campos obligatorios vacíos el sistema muestra un mensaje	Insatisfactorio.
	V	I	V	V	N/A		
	V	V	I	V	N/A		
	V	V	V	I	V		

Capítulo 3. Descripción y validación de la solución propuesta

						indicando que no pueden existir campos vacíos.	
--	--	--	--	--	--	--	--

3.6.7.5 Secciones a probar en el caso de uso Autenticar usuario

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Autenticar usuario	EC 1.1: Autenticar usuario satisfactoriamente	El usuario escribe su nombre de usuario y contraseña y presiona el botón “Entrar” para entrar en el sistema. El sistema verifica la validez del usuario y la contraseña introducida. Verifica los permisos que tiene asignados y en dependencia de los permisos le permite acceder a la aplicación mostrando la interfaz correspondiente.
	EC 1.2: Autenticar usuario falla.	Si los datos de autenticación introducidos por el usuario no son válidos, el sistema muestra el siguiente mensaje: “Usuario o contraseña incorrecta”. El sistema da la posibilidad de volver a introducir los datos.

Tabla 5. Secciones a probar en el caso de uso Autenticar usuario

3.6.7.6 Descripción de variables para el caso de uso Autenticar usuario

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	usuario	Campo de texto	No	El usuario introducirá en este campo su usuario para autenticarse.
2	contraseña	Campo de texto	No	El usuario introducirá en este campo su contraseña para autenticarse.

Tabla 6. Descripción de variables para el caso de uso Autenticar usuario.

Capítulo 3. Descripción y validación de la solución propuesta

3.6.7.7 Matriz de datos para la sección Autenticar usuario

Escenario	usuario	contraseña	Respuesta del sistema	Resultado de la prueba
Autenticar usuario satisfactoriamente.	V	V	El sistema verifica que el usuario y la contraseña existan en la Base de Datos y se correspondan. Verifica los permisos que tiene asignados ese usuario y en dependencia de los permisos le permite acceder a la aplicación mostrando la interfaz correspondiente.	Satisfactorio.
Autenticar usuario falla.	I	V	Si los datos de autenticación introducidos por el actor no son válidos, el sistema muestra el siguiente mensaje: "Usuario o contraseña incorrecta". El sistema da la posibilidad de volver a introducir los datos.	Satisfactorio.

Tabla 7. Matriz de datos para la sección Autenticar usuario.

Las pruebas funcionales que se le aplicaron al subsistema Transmisión de la plataforma PTARTV arrojaron resultados que permitieron saber hasta que punto la aplicación cumplía con los requisitos funcionales establecidos en la fase inicial de desarrollo.

3.6.8 Pruebas de rendimiento

En la Ingeniería del Software, las pruebas de rendimiento son aquellas que son realizadas para determinar que tan rápido un sistema realiza una tarea bajo ciertas condiciones pre-planificadas de trabajo. Estas pruebas también son utilizadas para validar y verificar diferentes aspectos de la calidad de software, como por ejemplo, escalabilidad, fiabilidad y el buen uso de los recursos. Las pruebas de rendimiento constituyen un subconjunto de la Ingeniería de Pruebas, la cual se esfuerza en mejorar el rendimiento, basándose en el diseño y la arquitectura de un sistema, antes de la realización del proceso de codificación (45).

3.6.9 Diseño de pruebas de rendimiento

Para probar el rendimiento del subsistema Transmisión de la plataforma PTARTV se levantó el proceso correspondiente al subsistema y se realizó un monitoreo del consumo de memoria y de la

Capítulo 3. Descripción y validación de la solución propuesta

CPU. El proceso se ejecutó en dos estaciones de trabajo con características diferentes, lo que permitió establecer límites en cuanto a la cantidad de canales a transmitir respondiendo a un hardware específico. A continuación se muestran las características de ambas estaciones, además de tablas que representan los niveles de consumo de memoria y de la CPU del subsistema durante su funcionamiento:

Estación de trabajo 1

- ✓ Sistema Operativo:
 - Ubuntu Release 10.4 (lucid).
 - Kernel Linux 2.6.32-21 generic.
 - GNOME 2.30.0.
- ✓ Hardware:
 - Memoria: 984 MiB.
 - Procesador 0: Intel(R) Core(TM)2 Duo CPU E4500 @ 2.20GHz.
 - Procesador 1: Intel(R) Core(TM)2 Duo CPU E4500 @ 2.20GHz.
- ✓ Estado del sistema:
 - Espacio libre en disco: 5.4 GiB.

Niveles de consumo de memoria y de la CPU para la estación de trabajo 1.

Cantidad de canales en transmisión.	Porcentaje consumido de la CPU.	Memoria consumida (MiB).
4	45	200.6
8	80	369.5
12	95	710.0

Tabla 8. Niveles de consumo de memoria y de la CPU para la estación de trabajo 1.

Estación de trabajo 2

- ✓ Sistema Operativo:
 - Ubuntu Release 10.10 (maverick).
 - Kernel Linux 2.6.35-22 generic.
 - GNOME 2.32.0.
- ✓ Hardware:
 - ✓ Memoria: 2.9 GiB.
 - ✓ Procesador 0: Intel(R) Core(TM) i3 CPU M 350 @ 2.27 GHz.

Capítulo 3. Descripción y validación de la solución propuesta

- ✓ Procesador 1: Intel(R) Core(TM) i3 CPU M 350 @ 2.27 GHz.
- ✓ Procesador 2: Intel(R) Core(TM) i3 CPU M 350 @ 2.27 GHz.
- ✓ Procesador 3: Intel(R) Core(TM) i3 CPU M 350 @ 2.27 GHz.
- ✓ Estado del sistema:
 - ✓ Espacio libre en disco: 5.4 GiB.

Niveles de consumo de memoria y de la CPU para la estación de trabajo 2.

Cantidad de canales en transmisión.	Porcentaje consumido de la CPU.	Memoria consumida (MiB).
4	20	115.3
8	40	180.7
12	50	238.9
16	55	324.0
20	65	396.2
24	80	451.6

Tabla 9. Niveles de consumo de memoria y de la CPU para la estación de trabajo 2.

3.6.10 Resultado de las pruebas

En este epígrafe se detallan las no conformidades encontradas durante las pruebas funcionales, así como determinar los límites de carga de la aplicación y el equipamiento necesario para su funcionamiento.

3.6.10.1 Resultado de las pruebas de unidad.

Al realizar pruebas de unidad pudieron obtenerse resultados satisfactorios lo que comprobó el buen funcionamiento de los más pequeños elementos que se componen el software.

3.6.10.2 Resultado de las pruebas funcionales.

Durante la fase de pruebas, específicamente las funcionales, se pudieron encontrar elementos que afectaban el buen desenvolvimiento de la aplicación. A continuación se detallan los elementos encontrados por los desarrolladores:

- ✓ El sistema permitía dejar campos vacíos al insertar un nuevo canal.
- ✓ Si no había conexión con la base de datos el sistema no mostraba ningún mensaje.
- ✓ El sistema permitía modificar un canal que estuviera en transmisión.

Capítulo 3. Descripción y validación de la solución propuesta

- ✓ El sistema permitía dejar campos vacíos al modificar los datos de un canal.

Al detectar este grupo de deficiencias el equipo de desarrollo trató de solucionarlas rápidamente y decidió realizar una nueva iteración de pruebas funcionales para el caso de uso Gestionar canal.

3.6.10.3 Resultado de las pruebas de rendimiento

Luego de realizar las pruebas de rendimiento se detectó que el procesamiento correspondiente a la aplicación se sustenta sobre la capacidad operativa del microprocesador. Si el procesador puede atender mayor número de instrucciones por frecuencia de reloj se consumirá menor cantidad de memoria. Los niveles de consumo de memoria aumentan drásticamente cuando la carga de trabajo del procesador se acerca a su límite.

3.6.11 Segunda iteración de pruebas funcionales para el caso de uso Gestionar canal

3.6.11.1 Matriz de datos 2 para la sección Crear nuevo canal

Escenario	nombre	puerto	dirección IP	url del log	descripción	Respuesta del sistema	Resultado de la prueba
Crear nuevo canal satisfactoriamente.	V	V	V	V	V	El sistema valida que los datos introducidos por el usuario sean correctos, inserta los mismos en la base de datos y muestra un mensaje indicando que la acción fue ejecutada satisfactoriamente.	Satisfactorio.
Crear nuevo canal falla.	I	V	V	V	N/A	El sistema valida los datos	Satisfactorio.

Capítulo 3. Descripción y validación de la solución propuesta

	V	I	V	V	N/A	introducidos por el usuario y:	
	V	V	I	V	V	Si los datos introducidos por el usuario están incorrectos el sistema muestra un mensaje indicando el error ocurrido.	
	V	V	V	I	V	Si existen campos obligatorios vacíos el sistema muestra un mensaje indicando que no pueden existir campos vacíos.	

Tabla 10. Matriz de datos 2 para la sección Crear nuevo canal.

3.6.11.2 Matriz de datos 2 para la sección Modificar canal

Escenario	nombre	puerto	dirección IP	url del log	descripción	Respuesta del sistema	Resultado de la prueba
Modificar canal satisfactoriamente.	V	V	V	V	V	El sistema valida que los datos introducidos por el usuario sean correctos, inserta los mismos en la base de datos y muestra un mensaje indicando que la	Satisfactorio

Capítulo 3. Descripción y validación de la solución propuesta

						acción fue ejecutada satisfactoriamente.	
Modificar canal falla.	I	V	V	V	N/A	El sistema valida que el canal seleccionado por el usuario no este en transmisión, de ser así muestra un mensaje indicando que no se pueden modificar los datos. Si el canal no está en transmisión el sistema valida los datos modificados por el usuario y: Si los datos introducidos por el usuario están incorrectos el sistema muestra un mensaje indicando el error ocurrido. Si existen campos obligatorios vacíos el sistema muestra un	Satisfactorio.
	V	I	V	V	N/A		
	V	V	I	V	N/A		
	V	V	V	I	V		

Capítulo 3. Descripción y validación de la solución propuesta

						mensaje indicando que no pueden existir campos vacíos.	
--	--	--	--	--	--	---	--

Tabla 11. Matriz de datos 2 para la sección Modificar canal.

Después de realizar la segunda iteración de pruebas del caso de uso Gestionar_canal se obtuvieron resultados satisfactorios lo que confirma que las deficiencias encontradas en la primera fase de pruebas fueron erradicadas.

3.7 Conclusiones

Los clientes de la industria del software cada vez son más exigentes en cuanto a la calidad de los productos que se ofertan. Conocer qué tipos de pruebas pueden aplicársele a un sistema informático permitió seleccionar las que se le aplicarían a la solución propuesta. A través de las pruebas seleccionadas se pudo validar el cumplimiento de los requisitos funcionales establecidos en la fase inicial del proceso de desarrollo del producto, constatando el correcto funcionamiento del subsistema Transmisión de la plataforma PTARTV. Los resultados obtenidos durante la fase de pruebas del subsistema representaron un elemento fundamental para corregir las deficiencias encontradas y así dar por concluida la construcción de una nueva versión de los procedimientos y técnicas de transmisión de señales televisivas de la plataforma PTARTV.

Conclusiones generales

Con la caracterización de los procesos relacionados con la transmisión de señales televisivas en la DTU y en la Plataforma de Transmisión Abierta para Radio y Televisión, se pudo tener mayor claridad para implementar las funcionalidades correspondientes a la solución propuesta en el presente trabajo. Para desarrollar el sistema se caracterizaron las herramientas y tecnologías de desarrollo establecidas en la arquitectura del producto, permitiendo reconocer las facilidades que las mismas brindan y de qué forma explotarlas para el posterior desarrollo. Mientras se desarrollaba el subsistema el equipo de trabajo aumentó el dominio sobre el marco de trabajo Qt y la biblioteca Libvlc.

Un momento clave de la investigación fue la búsqueda y descripción de aplicaciones similares a la solución propuesta. Lo que permitió tomar sus elementos positivos e implantarlos en el nuevo subsistema.

En aras de adquirir sincronía entre los miembros del equipo de trabajo se desarrolló el modelo de implementación, así como el estándar de codificación a aplicar para la construcción del subsistema. Finalmente, para examinar la calidad y correcto funcionamiento del sistema, se diseñaron y ejecutaron los casos de prueba, estos arrojaron resultados satisfactorios, demostrando el cumplimiento de los requerimientos funcionales establecidos en la fase inicial del proceso de desarrollo del producto. Con el objetivo también de examinar la calidad, se efectuaron pruebas de rendimiento que le permitieron al equipo determinar el hardware que se necesita para desplegar el subsistema Transmisión respondiendo a la cantidad de canales necesitados por el cliente.

Por lo anteriormente expuesto se constata el cumplimiento del objetivo general de la investigación proveyendo a la UCI y a la DTU de una herramienta que permite el montaje de varios canales de TV sin grandes inversiones monetarias.

Recomendaciones

- ✓ Continuar con el desarrollo y soporte del subsistema construido.
- ✓ Sumar al sistema funcionalidades que permitan la configuración de la interfaz visual de la aplicación y así lograr una interfaz gráfica más amigable e intuitiva.
- ✓ Mejorar algunas de las funcionalidades del VLC como servidor streaming para que sea aún más factible su utilización en las aplicaciones desarrolladas por el centro GEYSED.

Trabajos citados

1. Diccionario de la Lengua Española - Vigésima segunda edición. [En línea] Real Academia Española, 2001. [Citado el: 18 de noviembre de 2010.] <http://buscon.rae.es/drael/>.
2. **Chiavenato, Idalberto.** *Introducción a la Teoría General de la Administración.* 7ma. Bogotá : McGraw-Hill Interamericana, 2004. pág. 10.
3. **Silva, Reinaldo Oliveira Da.** *Teorías de la Administración.* s.l. : International Thomson Editores, 2002. pág. 6.
4. Tech Terms.com. [En línea] Sharpened Productions, 2010. [Citado el: 6 de diciembre de 2010.] <http://www.techterms.com/definition/digital>.
5. MasterMagazine. [En línea] [Citado el: 6 de diciembre de 2010.] <http://www.mastermagazine.info/termino/4618.php>.
6. Definición.de. [En línea] 2008. [Citado el: 6 de diciembre de 2010.] <http://definicion.de/senal/>.
7. Dictionarist tr. [En línea] 2010. [Citado el: 6 de diciembre de 2010.] <http://definicion.dictionarist.com/se%C3%B1al>.
8. Definición ABC. [En línea] 2008. [Citado el: 6 de diciembre de 2010.] <http://www.definicionabc.com/general/senal.php>.
9. DAEDALUS. [En línea] [Citado el: 2 de diciembre de 2010.] <http://www.daedalus.es/inteligencia-de-negocio/sistemas-complejos/ciencia-de-sistemas/que-es-un-sistema/>.
10. Glosario.net. [En línea] 2010. [Citado el: 6 de diciembre de 2010.] <http://tecnologia.glosario.net/terminos-tecnicos-internet/transmisi%C3%B3n-1631.html>.
11. The Free Dictionary. [En línea] 2010. [Citado el: 6 de diciembre de 2010.] <http://translate.google.com/cu/translate?u=http://www.thefreedictionary.com/transmission&sl=en&tl=es&hl=&ie=UTF-8>.
12. Dolceta. [En línea] 2010. [Citado el: 24 de noviembre de 2010.] <http://www.dolceta.eu/espana/Mod4/spip.php?article203>.
13. The Free Dictionary. [En línea] 2010. [Citado el: 3 de diciembre de 2010.] <http://encyclopedia2.thefreedictionary.com/digital+signal>.
14. DaveChaffey.com. [En línea] 2010. [Citado el: 24 de noviembre de 2010.] <http://www.davechaffey.com/E-marketing-Glossary/Digital-television.htm>.
15. **Villamar, Roberto Carlos y Cos Cordero, Pedro Mario.** *SISTEMA DE VIDEO DIGITAL EN UN ESTUDIO DE TELEVISIÓN.* Ciudad habana : s.n., 2010.

16. *El apagón de la televisión analógica en Cuba: 10 respuestas sobre la televisión digital.* **Dábalos, Gabriel.** [ed.] Lázaro de Jesús. La Habana : s.n., 2010, En Vivo. Esta revista tiene un sitio web->www.envivo.icrt.cu.
17. DTV.gov. [En línea] [Citado el: 22 de noviembre de 2010.] http://www.dtv.gov/spanish/outreach_samplenews1.html.
18. XF Software. [En línea] 2009. [Citado el: 2 de diciembre de 2010.] <http://www.xfs.es/xframe.html>.
19. Dayang. [En línea] 2009. [Citado el: 4 de diciembre de 2010.] <http://www.dayang.com/list.php?classid=10&id=7>.
20. ADDItelecom. [En línea] [Citado el: 4 de diciembre de 2010.] http://www.additelecom.com/elemento_detalle.php?pk_cd_elemento=131&op=&fil_fk_cd_categoria=43.
21. Scribd. [En línea] 2010. [Citado el: 15 de febrero de 2011.] <http://www.scribd.com/doc/2050925/metodologias-de-desarrollo-software>.
22. Grupo de Soluciones GSInnova. [En línea] [Citado el: 9 de febrero de 2011.] <http://www.rational.com.ar/herramientas/rup.html>.
23. **Jacobson y otros.** 2000.
24. **Ivar Jacobson, Grady Booch y James Rumbaugh.** *El Lenguaje Unificado de Modelado.Manual de Referencia.* 2000. 78-84-7829-087-1.
25. Scribd. [En línea] 2010. [Citado el: 9 de febrero de 2011.] <http://www.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE>.
26. Free Download Manager. [En línea] [Citado el: 11 de febrero de 2011.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.
27. **Mora, Dunier Domínguez.** *Implementación de los Subsistemas Web y Transferencia de la Plataforma de Transmisión Abierta para Radio y Televisión.* Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2009. págs. 13,14.
28. Lenguajes de Programación. [En línea] 2009. [Citado el: 9 de febrero de 2011.] <http://www.lenguajes-de-programacion.com/programacion-orientada-a-objetos.shtml>.
29. **Moreno, Francisco.** *Introducción a la OOP.* s.l. : EIDOS, 2000.

30. Definición.org. [En línea] [Citado el: 9 de febrero de 2011.] <http://www.definicion.org/lenguaje-de-programacion>.
31. Sun Microsystems. [En línea] [Citado el: 3 de febrero de 2011.] <http://www.java.com/es/about/>.
32. 50 Megs. [En línea] [Citado el: 15 de febrero de 2011.] <http://arechiga.50megs.com/tpoo2/javah1.html>.
33. **Katrib, Miguel.** *Programacion Orientada a Objetos en CPP*. Ciudad de la Habana : s.n.
34. **Mora, Sergio Luján.** Grupo de Investigación en Procesamiento(GPLSI). [En línea] 2009. [Citado el: 15 de febrero de 2011.] <http://gplsi.dlsi.ua.es/~slujan/materiales/cpp-muestra.pdf>.
35. CodeBox. [En línea] 2008. [Citado el: 9 de febrero de 2011.] <http://www.codebox.es/glosario>.
36. **Gómez, Reynier Pupo.** *Título: Gestor de ficheros para el Entorno de Escritorio*. Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2010.
37. Sitio web de la EU de Ingeniería Técnica de Ovideo. [En línea] [Citado el: 9 de febrero de 2011.] <http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf>.
38. **Nokia.** QT. [En línea] 2011. [Citado el: 9 de febrero de 2011.] <http://qt.nokia.com/products/developer-tools>.
39. Garbage Collector. [En línea] [Citado el: 9 de febrero de 2011.] http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php.
40. PostgreSQL. [En línea] 2011. [Citado el: 16 de febrero de 2011.] <http://www.postgresql.org/about>.
41. **Gabriel, Leonardo.** *Los Nuevos medios en la Era digital.Convergencias e industrias de Streaming*. 2008.
42. **Letelier Torres, Patricio.** *Desarrollo de Software Orientado a Objeto usando UML*. s.l. : Departamento Sistemas Informáticos y Computación. Universidad Politécnica de Valencia.
43. *Ayuda de RUP*. s.l. : IBM Corp., 2006.
44. **Oré B., Ing. Alexander.** CalidadSoftware.com. *FUNCTIONAL TESTING - PRUEBAS FUNCIONALES*. [En línea] 2009. [Citado el: 4 de Mayo de 2010.] http://www.calidadyssoftware.com/testing/pruebas_funcionales.php.

45. Corporación Sybven-Integración Tecnológica. [En línea] 2011. [Citado el: 9 de mayo de 2011.] http://www.corporacionsybven.com/portal/index.php?option=com_content&view=article&id=246.

Bibliografía consultada

1. Diccionario de la Lengua Española - Vigésima segunda edición. [En línea] Real Academia Española, 2001. [Citado el: 18 de noviembre de 2010.] <http://buscon.rae.es/drae/>.
2. **Chiavenato, Idalberto.** *Introducción a la Teoría General de la Administración.* 7ma. Bogotá : McGraw-Hill Interamericana, 2004. pág. 10.
3. **Silva, Reinaldo Oliveira Da.** *Teorías de la Administración.* s.l. : International Thomson Editores, 2002. pág. 6.
4. Tech Terms.com. [En línea] Sharpened Productions, 2010. [Citado el: 6 de diciembre de 2010.] <http://www.techterms.com/definition/digital>.
5. MasterMagazine. [En línea] [Citado el: 6 de diciembre de 2010.] <http://www.mastermagazine.info/termino/4618.php>.
6. Definición.de. [En línea] 2008. [Citado el: 6 de diciembre de 2010.] <http://definicion.de/senal/>.
7. Dictionarist tr. [En línea] 2010. [Citado el: 6 de diciembre de 2010.] <http://definicion.dictionarist.com/se%C3%B1al>.
8. Definición ABC. [En línea] 2008. [Citado el: 6 de diciembre de 2010.] <http://www.definicionabc.com/general/senal.php>.
9. DAEDALUS. [En línea] [Citado el: 2 de diciembre de 2010.] <http://www.daedalus.es/inteligencia-de-negocio/sistemas-complejos/ciencia-de-sistemas/que-es-un-sistema/>.
10. Glosario.net. [En línea] 2010. [Citado el: 6 de diciembre de 2010.] <http://tecnologia.glosario.net/terminos-tecnicos-internet/transmisi%C3%B3n-1631.html>.
11. The Free Dictionary. [En línea] 2010. [Citado el: 6 de diciembre de 2010.] <http://translate.google.com/cu/translate?u=http://www.thefreedictionary.com/transmission&sl=en&tl=es&hl=&ie=UTF-8>.
12. Dolceta. [En línea] 2010. [Citado el: 24 de noviembre de 2010.] <http://www.dolceta.eu/espana/Mod4/spip.php?article203>.
13. The Free Dictionary. [En línea] 2010. [Citado el: 3 de diciembre de 2010.] <http://encyclopedia2.thefreedictionary.com/digital+signal>.
14. DaveChaffey.com. [En línea] 2010. [Citado el: 24 de noviembre de 2010.] <http://www.davechaffey.com/E-marketing-Glossary/Digital-television.htm>.
15. **Villamar, Roberto Carlos y Cos Cordero, Pedro Mario.** *SISTEMA DE VIDEO DIGITAL EN UN ESTUDIO DE TELEVISIÓN.* Ciudad habana : s.n., 2010.

Bibliografía consultada

16. *El apagón de la televisión analógica en Cuba: 10 respuestas sobre la televisión digital.* **Dábalos, Gabriel.** [ed.] Lázaro de Jesús. La Habana : s.n., 2010, En Vivo. Esta revista tiene un sitio web->www.envivo.icrt.cu.
17. DTV.gov. [En línea] [Citado el: 22 de noviembre de 2010.] http://www.dtv.gov/spanish/outreach_samplenews1.html.
18. XF Software. [En línea] 2009. [Citado el: 2 de diciembre de 2010.] <http://www.xfs.es/xframe.html>.
19. Dayang. [En línea] 2009. [Citado el: 4 de diciembre de 2010.] <http://www.dayang.com/list.php?classid=10&id=7>.
20. ADDItelecom. [En línea] [Citado el: 4 de diciembre de 2010.] http://www.additelecom.com/elemento_detalle.php?pk_cd_elemento=131&op=&fil_fk_cd_categoria=43.
21. Scribd. [En línea] 2010. [Citado el: 15 de febrero de 2011.] <http://www.scribd.com/doc/2050925/metodologias-de-desarrollo-software>.
22. Grupo de Soluciones GSInnova. [En línea] [Citado el: 9 de febrero de 2011.] <http://www.rational.com.ar/herramientas/rup.html>.
23. **Jacobson y otros.** 2000.
24. **Ivar Jacobson, Grady Booch y James Rumbaugh.** *El Lenguaje Unificado de Modelado.Manual de Referencia.* 2000. 78-84-7829-087-1.
25. Scribd. [En línea] 2010. [Citado el: 9 de febrero de 2011.] <http://www.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE>.
26. Free Download Manager. [En línea] [Citado el: 11 de febrero de 2011.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.
27. **Mora, Dunier Domínguez.** *Implementación de los Subsistemas Web y Transferencia de la Plataforma de Transmisión Abierta para Radio y Televisión.* Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2009. págs. 13,14.
28. Lenguajes de Programación. [En línea] 2009. [Citado el: 9 de febrero de 2011.] <http://www.lenguajes-de-programacion.com/programacion-orientada-a-objetos.shtml>.
29. **Moreno, Francisco.** *Introducción a la OOP.* s.l. : EIDOS, 2000.

Bibliografía consultada

30. Definición.org. [En línea] [Citado el: 9 de febrero de 2011.] <http://www.definicion.org/lenguaje-de-programacion>.
31. Sun Microsystems. [En línea] [Citado el: 3 de febrero de 2011.] <http://www.java.com/es/about/>.
32. 50 Megs. [En línea] [Citado el: 15 de febrero de 2011.] <http://arechiga.50megs.com/tpoo2/javah1.html>.
33. **Katrib, Miguel.** *Programacion Orientada a Objetos en CPP*. Ciudad de la Habana : s.n.
34. **Mora, Sergio Luján.** Grupo de Investigación en Procesamiento(GPLSI). [En línea] 2009. [Citado el: 15 de febrero de 2011.] <http://gplsi.dlsi.ua.es/~slujan/materiales/cpp-muestra.pdf>.
35. CodeBox. [En línea] 2008. [Citado el: 9 de febrero de 2011.] <http://www.codebox.es/glosario>.
36. **Gómez, Reynier Pupo.** *Título: Gestor de ficheros para el Entorno de Escritorio*. Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2010.
37. Sitio web de la EU de Ingeniería Técnica de Ovideo. [En línea] [Citado el: 9 de febrero de 2011.] <http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf>.
38. **Nokia.** QT. [En línea] 2011. [Citado el: 9 de febrero de 2011.] <http://qt.nokia.com/products/developer-tools>.
39. Garbage Collector. [En línea] [Citado el: 9 de febrero de 2011.] http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php.
40. PostgreSQL. [En línea] 2011. [Citado el: 16 de febrero de 2011.] <http://www.postgresql.org/about>.
41. **Gabriel, Leonardo.** *Los Nuevos medios en la Era digital. Convergencias e industrias de Streaming*. 2008.
42. **Letelier Torres, Patricio.** *Desarrollo de Software Orientado a Objeto usando UML*. s.l. : Departamento Sistemas Informáticos y Computación. Universidad Politécnica de Valencia.
43. *Ayuda de RUP*. s.l. : IBM Corp., 2006.
44. **Oré B., Ing. Alexander.** CalidadSoftware.com. *FUNCTIONAL TESTING - PRUEBAS FUNCIONALES*. [En línea] 2009. [Citado el: 4 de Mayo de 2010.] http://www.calidadsoftware.com/testing/pruebas_funcionales.php.

Bibliografía consultada

45. Corporación Sybven-Integración Tecnológica. [En línea] 2011. [Citado el: 9 de mayo de 2011.] http://www.corporacionsybven.com/portal/index.php?option=com_content&view=article&id=246.
46. **Berenguer, Abel Días y Román Vieito, Alberto Ramón.** *Sistema de Administración y Configuración de la solución de Captura y Catalogación de Medias.* Ciudad de la Habana : s.n., 2009.
47. **García Pañeda, Xabiel, Marín Prendes, Ignacio y Melendi Palacios, David.** Universidad de Oviedo. *Área de Ingeniería Telemática, Departamento de Informática.* [En línea] [Citado el: 29 de noviembre de 2010.] http://www.it.uniovi.es/old/material/cursos/streaming_EU_122004/Intro.pdf.
48. **Alonso Guerrero, Zorlin y Leyva González, Genry.** *Sistema de Captura y Transcripción de Audio.* Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2009.
49. **Marín Abreu, Angel Dayán.** *Desarrollo de la Plataforma VideoWeb. Rol Analista.* La Habana : s.n., 2009.
50. **Senso, Jose A. y Piñeiro, Antonio de la Rosa.** *El concepto de metadato. Algo más que descripción de recursos electrónicos.* Brasília : s.n., 2003.
51. **Domínguez Mora, Dunier.** *Implementación de los Subsistemas Web y Transferencia de la Plataforma de Transmisión Abierta para Radio y Televisión.* La Habana : s.n., 2009.
52. **Alegsa.** Diccionario Informático. *Definición de UML.* [En línea] Alegsa. [Citado el: 15 de Marzo de 2010.] <http://www.alegsa.com.ar/Dic/uml.php>.
53. **Ing. Alexander Oré B.** CalidadySoftware.com. [En línea] 2009. [Citado el: 6 de mayo de 2011.] http://www.calidadysoftware.com/testing/pruebas_funcionales.php.

Anexos

Anexo 1. Entrevista realizada al equipo de trabajo del departamento “Programación” de la DTU.

Preguntas:

¿Qué elementos se tienen en cuenta para iniciar una transmisión?

¿Qué protocolo es más conveniente utilizar para realizar streaming de video y audio en la DTU?

¿Qué operaciones puede realizar el transmisor durante una transmisión?

¿Existe algún estándar para la codificación de las medias?

¿Cuáles son los beneficios que brinda utilizar streaming para las transmisiones en la DTU?

Glosario de términos

Aplicación: Es una clase de programa informático creado para facilitar al usuario un determinado tipo de trabajo. Esto lo caracteriza frente a otros programas como los sistemas operativos, las utilidades y los lenguajes de programación.

Cliente: Es una aplicación informática cuya función es acceder a los servicios que ofrece un servidor, haciendo uso generalmente de una red de telecomunicaciones.

Ficheros: Directorios. Agrupación de archivos de datos, atendiendo a su contenido, a su propósito o a cualquier otro criterio.

Multiplataforma: Se utiliza el término para denominar a los programas, lenguajes de programación u otra clase de software que pueden brindar sus prestaciones funcionando sobre diversas combinaciones de hardware y software.

Permisos: Tareas o acciones que puede realizar un usuario dentro de un sistema en dependencia de su nivel de acceso.

Servidor: Software u ordenador que provee servicios a otros programas o equipos denominados clientes.

Streaming: Tecnología que permite la reproducción de sonido o vídeo sin que sea necesario descargar previamente todo el archivo de recurso.

Televisión: Medio de comunicación que combina los mensajes de imágenes fijas y en movimiento con voz, música, efectos sonoros y especiales.

Transmisión: Emitir la señal de radio o televisión desde una estación emisora a otra que actúa como receptora de la misma.

Transmisión Abierta: Señal de libre acceso por la cual no es necesario pagar para consumirla.

Video: Es una tecnología de captura electrónica, tratamiento, almacenamiento, transmisión, y reconstrucción de una sucesión de imágenes y sonidos que representan escenas en movimiento.