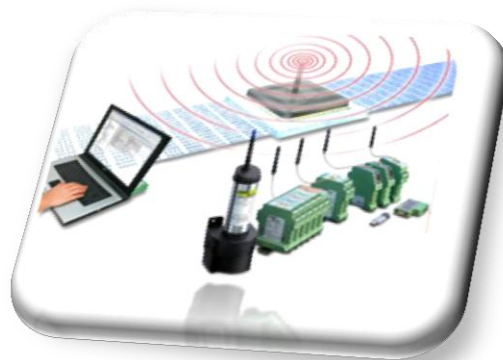


UCi

Universidad de las Ciencias Informáticas

Facultad 6

*Desarrollo del componente de transmisión de
noticias a través de la red para la plataforma
PRIMICIA*



Autor: Jorge Alejandro Puebla González

Tutor: Ing. Susana Yaque Rivera

La Habana, Junio 2015

Declaración de Autoría

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

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

Jorge Alejandro Puebla González

Susana Yaque Rivera

Firma del Autor

Firma del Tutor

Datos de Contacto

Tutora: Ing. Susana Yaque Rivera.

Graduada de Ingeniera en Ciencias Informáticas en el año 2013. Trabaja en el proyecto Plataforma de Televisión Informativa, PRIMICIA, del centro de Geoinformática y Señales Digitales (GEYSED) de la facultad 6 en la Universidad de las Ciencias Informáticas (UCI).

Correo electrónico: susanay@uci.cu

Teléfono: 837-2244

Pensamiento

“En la tierra hace falta personas que trabajen más y critiquen menos, que construyan más y destruyan menos, que prometan menos y resuelvan más, que esperen recibir menos y dar más, que digan mejor ahora que mañana.”
Che



Dedicatoria

A mi mamá por ser guía indispensable de mi vida, a ella que nunca escatimó en esfuerzos por regalarme un mañana mejor. Por sacrificarse a un nivel inimaginable para no cargar mi vida con problemas que atentaran contra mi formación. A ti Marlenis, con todo mi amor y agradecimiento te dedico el fruto de nuestros esfuerzos.

A mi novia Isi por apoyarme en todo, por inyectarme la pizca de sensatez que nunca he tenido, por hacer mis días más sencillos, por preocuparse por mí y regalarme amor incondicionalmente.

Agradecimientos

A mi mamá por poner toda su confianza en mí.

A mi novia por apoyarme en todo momento.

A mi tutor “empírico” y amigo Nelson Sánchez por sus horas invertidas explicándome cosas que acababa de entender, mil gracias chino.

A mi amigo Serguey Caballero por sus clases desinteresadas de Gstreaming.

A mi tutora por toda la ayuda ofrecida durante la investigación.

Al tribunal por sus críticas constructivas.

A la oponente por haber puesto todo su empeño en mejorar el trabajo de diploma.

A todos los que hoy no menciono y me ayudaron, gracias.

Resumen

Para perfeccionar la televisión (TV), el hombre ha llegado a vincularla con la informática y las telecomunicaciones dando lugar al surgimiento y evolución de plataformas de televisión informativa en todo el mundo. En Cuba, específicamente en la Universidad de las Ciencias Informáticas (UCI), se desarrolla la Plataforma de Televisión Informativa PRIMICIA; aplicación capaz de proveer un canal de TV para la transmisión de contenidos informativos a través de una red de TV.

PRIMICIA basa su funcionamiento en redes de TV por cable y satélite. Para que los contenidos multimedia transmitidos se visualicen en los televisores es necesario utilizar un dispositivo de conversión digital-analógica. Esta forma de transmitir aunque cumple el objetivo por el cual fue creado el proyecto, genera gastos adicionales, convierte a PRIMICIA en un producto que depende de un hardware y sólo es capaz de llegar hasta un número finito de televisores. Esto provoca que se vean limitadas las oportunidades de negocio del centro productivo GEYSED.

La siguiente investigación se realiza con el objetivo de desarrollar un componente informático para la plataforma PRIMICIA que permita transmitir las noticias a través de una red IP. De modo que en este documento se presenta el desarrollo del componente, describiéndose desde su concepción, diseño e implementación, hasta las pruebas realizadas.

El mismo fue diseñado con una arquitectura n-capas y desarrollado con herramientas y tecnologías libres, tributando a la independencia y soberanía tecnológica del país. La validación de la propuesta demuestra que PRIMICIA podrá transmitir las noticias a través de una red IP, no dependerá de un dispositivo CDA y brindará la opción de llegar a otros dispositivos de visualización aparte de los televisores. Lo que influye directamente a que no se vean limitadas las oportunidades de negocio del centro GEYSED.

Palabras Clave: componente informático, noticia, PRIMICIA, red IP, televisión, visualizar.

Abstract

To perfect the television (TV), man has come to link it to computer and telecommunications leading to the emergence and evolution of informative television platforms worldwide. In Cuba, specifically in the University of Information Sciences (UCI), the Platform for TV PRIMICIA Fact is developed; application that can provide a TV channel for transmission of information content through a TV network.

PRIMICIA bases its operation on cable and satellite TV. So that the transmitted multimedia contents are displayed on televisions it is necessary to use a device for digital-analog conversion. This form of transmitting even fulfills the purpose for which it was created the project generates additional costs, makes first in a product that depends on hardware and is only able to reach a finite number of televisions. This cause the business opportunities look GEYSED limited production center.

The following research is conducted with the aim of developing a software component for PREMIERE platform to broadcast news through an IP network. So herein development component, describing from conception, design and implementation to the tests carried out is presented.

It was designed with an n-tier architecture and developed with free tools and technologies, paying technological independence and sovereignty. The validation of the proposal shows that PRIMICIA may transmit news via an IP network, there will depend on a CDA device and provide an option to reach other display devices other than televisions. This directly influences to business opportunities not see GEYSED center limited.

Keywords: computer components, news, PRIMICIA, IP network, television, visualize.

Índice de Contenido

- Introducción - 1 -
- Capítulo I. Fundamentación Teórica del componente de transmisión de noticias a través de la red para la plataforma PRIMICIA - 5 -
 - 1.1. Introducción..... - 5 -
 - 1.2. Términos asociados al dominio del problema..... - 5 -
 - 1.3. Procesos de transmisión de señales digitales a través de una red IP..... - 6 -
 - 1.3.1. Situación Problemática - 7 -
 - 1.4. Análisis de soluciones existentes - 8 -
 - 1.4.1. Soluciones existentes a nivel internacional..... - 8 -
 - 1.4.2. Soluciones existentes a nivel nacional..... - 9 -
 - 1.5. Proceso y herramientas a utilizar en la modelación, diseño y construcción del componente informático - 10 -

El componente informático que se desea desarrollar formará parte del proyecto PRIMICIA. Teniendo en cuenta esta afirmación se debe utilizar el mismo proceso de desarrollo y apoyarse en las mismas herramientas definidas por el arquitecto de PRIMICIA. De esta forma se garantiza compatibilidad entre el proyecto y el producto informático en desarrollo. A continuación se puntualizan las principales características del proceso y las herramientas utilizadas. - 10 -

 - 1.5.1. Metodología de desarrollo de software - 10 -
 - 1.5.2. Lenguaje de Modelado Unificado..... - 10 -
 - 1.5.3. Herramienta para el modelado - 11 -
 - 1.5.4. Lenguaje de programación - 11 -
 - 1.5.5. Framework de desarrollo - 11 -
 - 1.5.6. Entorno de desarrollo (IDE) - 12 -
 - 1.6. Conclusiones parciales..... - 12 -
- Capítulo II. Análisis y diseño del componente de transmisión de noticias a través de la red para la plataforma PRIMICIA - 14 -
 - 2.1. Introducción..... - 14 -
 - 2.2. Modelo de dominio - 14 -
 - 2.2.1. Principales eventos del entorno..... - 14 -

2.2.2. Diagrama de clases del Modelo de Dominio	- 15 -
2.2.3. Principales conceptos asociados al dominio del problema	- 15 -
2.3. Especificación de los requisitos de software	- 16 -
2.3.1. Requisitos Funcionales.....	- 16 -
2.3.2. Requisitos No Funcionales	- 17 -
2.4. Descripción de la solución propuesta	- 18 -
2.4.1. Descripción de los actores del sistema	- 18 -
2.4.2. Diagrama de Casos de Uso del Sistema	- 18 -
2.4.3. Descripción de los Casos de Uso	- 19 -
2.5. Modelo del diseño	- 28 -
2.5.1. Arquitectura de software	- 29 -
2.5.2. Arquitectura en n capas.....	- 29 -
2.5.3. Patrones de diseño.....	- 29 -
2.5.4. Diagrama de clases del diseño	- 31 -
2.6. Conclusiones parciales.....	- 32 -
Capítulo III. Implementación y validación del componente de transmisión de noticias a través de la red para la plataforma PRIMICIA	- 34 -
3.1. Introducción.....	- 34 -
3.2. Modelo de implementación.....	- 34 -
3.2.1. Diagrama de componentes.....	- 34 -
3.3. Modelo de despliegue	- 35 -
3.3.1. Diagrama de despliegue.....	- 35 -
3.4. Estándar de codificación	- 35 -
3.5. Pruebas.....	- 36 -
3.5.1. Pruebas de rendimiento, carga y estrés.....	- 37 -
3.5.2. Prueba de caja negra	- 39 -
3.5.3. Prueba de caja blanca	- 41 -
3.5.4. Pruebas de aceptación	- 44 -
3.6. Conclusiones parciales.....	- 45 -
Conclusiones generales.....	- 46 -

Recomendaciones - 47 -
Referencias Bibliográficas..... - 48 -
Anexos..... - 51 -

Índice de Tablas

Tabla 1. Actores del sistema	- 18 -
Tabla 2. Relación entre Casos de uso y Requisitos Funcionales	- 19 -
Tabla 3. Descripción de Caso de uso Transmitir canal.....	- 20 -
Tabla 4. Descripción de Caso de uso Gestionar plantilla de configuración.....	- 28 -
Tabla 5. Rendimiento del CPU y la RAM.....	- 38 -
Tabla 6. Rendimiento del CPU y la RAM (2)	- 39 -
Tabla 7. Escenarios a probar en el caso de uso Configurar transmisión	- 41 -
Tabla 8. Escenarios a probar en el caso de uso Transmitir canal	- 41 -
Tabla 9. Personas que participan en la prueba de aceptación	- 44 -
Tabla 10. Descripción de Caso de uso Configurar transmisión	- 52 -
Tabla 11. Descripción de Caso de uso Detener transmisión	- 53 -

Índice de Figuras

Ilustración 1. Diagrama de clases del Modelo de Dominio	- 15 -
Ilustración 2. Diagrama de Casos de Uso del Sistema.....	- 18 -
Ilustración 3. Diagrama de Clases del Diseño.....	- 32 -
Ilustración 4. Diagrama de Componentes	- 34 -
Ilustración 5. Diagrama de Despliegue.....	- 35 -
Ilustración 6. Grafo del método MediaConfigure	- 42 -
Ilustración 7. Grafo del método InicioStream.....	- 44 -

Introducción

La televisión es un sistema de comunicación para la transmisión y recepción de imágenes en movimiento y sonido a distancia. Desde fines de los años cuarenta no deja de ampliar su mercado; los progresos de la electrónica, la informática y de las telecomunicaciones conjugan sus efectos para abrir a los medios de comunicación audiovisuales las perspectivas de una expansión acelerada. (Pierre, y otros, 2012).

Existen múltiples corporaciones informativas que realizan todos los procesos de transmisión de forma automática. Lo que posibilita que no sólo transmitan noticias, sino que también muestran informaciones adicionales como: estadísticas deportivas, comportamiento de la bolsa, partes meteorológicos, entre otros, manteniendo así más informado y actualizado al televidente sobre múltiples informaciones a la vez. La mayoría de estas entidades son desarrolladas sobre plataformas propietarias por lo que, casi siempre, hay que pagar un elevado precio por la prestación de sus servicios.

La Corporación Británica de Radiodifusión (BBC) es un ejemplo de estas corporaciones, la cual constituye un servicio público de radio, televisión e internet del Reino Unido con más de ocho décadas de trayectoria. La BBC transmite en emisoras AM y FM en más de 100 capitales del mundo por una red de más de 250 corresponsales y puede ser sintonizada desde cualquier parte del planeta (BBC Mundo, 2014).

Cuba no está excluida de lo que acontece en el mundo sobre el desarrollo de tecnologías en el campo de la transmisión de señales digitales. Varios sectores como la salud, la educación y las investigaciones están interesados en que sus trabajadores y/o clientes estén completamente informados de las actividades más relevantes de la organización. Un canal de noticias representa una vía factible para solucionar esta situación.

En la Universidad de las Ciencias Informáticas (UCI), específicamente en el centro productivo de Geoinformática y Señales Digitales (GEYSED) de la Facultad 6, se desarrollan sistemas informáticos capaces de brindar información mediante canales televisivos (CTVs). Uno de ellos es la Plataforma de Televisión Informativa PRIMICIA, esta aplicación es capaz de proveer un canal de televisión para la transmisión de contenidos informativos a través de una red de televisión (Andrés González, 2012).

PRIMICIA es un producto informático desarrollado por la necesidad de mantener a clientes y/o trabajadores informados de una manera clara, precisa y oportuna sobre cualquier tema de interés definido por la entidad que lo utilice. En el año 2014 el Ministerio de la Informática y las Telecomunicaciones (MINCOM) se

interesó en los servicios que brinda el proyecto. Sin embargo, no finalizaron las negociaciones porque el modo de transmisión de la plataforma informativa representa un inconveniente para dicho ministerio.

La transmisión de PRIMICIA basa su funcionamiento en redes de TV por cable y satélite. Los contenidos multimedia asociados a las noticias son transmitidos desde una computadora (PC) y se visualizan en los televisores (TVs) destino. De forma que si se quieren visualizar las noticias en un nuevo TV, antes se debe añadir a la red de transmisión cableada o satelital. Además, para permitir que las noticias transmitidas sean visualizadas, se utiliza un dispositivo de conversión digital-analógica (CDA¹).

El MINCOM, al igual que otros clientes, estimó inconveniente utilizar un producto informático que además de un software dependa de un hardware específico (CDA). Utilizar a PRIMICIA generaría gastos adicionales, por ejemplo: una tarjeta de video *GeForce 4 MX* tiene un precio en el mercado internacional de alrededor de 85 euros, mientras que el precio de un convertidor *VGA-PAL de AverMedia* oscila entre los 90 y 95 euros (MeriStation, 20015).

Estos datos reafirman que PRIMICIA corre el riesgo de perder clientes interesados en un canal informativo porque: las noticias sólo pueden ser visualizadas en los TVs, su transmisión depende de un hardware específico con costos variables en el mercado internacional y genera gastos adicionales. Lo antes expuesto evidencia que se ven limitadas las oportunidades del negocio.

Dada la **situación problemática** expuesta anteriormente, el presente trabajo de diploma plantea como **problema de la investigación**: El modo de transmisión de PRIMICIA depende de un dispositivo CDA y sólo está dirigido a los televisores, lo que provoca que se vean limitadas las oportunidades de negocio.

Para solucionar el problema planteado se define como **objetivo general**: Desarrollar un componente informático para la plataforma PRIMICIA que permita transmitir las noticias a través de una red IP².

Se define como **objeto de estudio**: Los procesos de transmisión de señales digitales a través de una red IP, presentando como **campo de acción**: Los procesos de transmisión de contenidos multimedia sobre redes IP.

Para dar cumplimiento al objetivo planteado se han propuesto las siguientes **tareas para la investigación**:

¹ Dispositivo encargado de convertir las señales digitales que origina la PC y convertirla a señales analógicas para que puedan visualizarse en los TVs.

² Es una red compuesta por nodos, los cuales pueden ser desde un ordenador hasta un router; estos se comunican por enlaces y se utiliza la conmutación de paquetes como técnica de comunicación entre ellos.

- ✚ Caracterización de los mecanismos utilizados para transmitir contenidos multimedia.
- ✚ Selección de la metodología y las herramientas a utilizar en el proceso de desarrollo del componente de transmisión de noticias a través de la red para la plataforma PRIMICIA.
- ✚ Definición de los requisitos funcionales y no funcionales para establecer las capacidades y restricciones con que debe cumplir el componente de transmisión de noticias a través de la red para la plataforma PRIMICIA.
- ✚ Realización del diseño ingenieril de los procesos para representar las funcionalidades del componente de transmisión de noticias a través de la red para la plataforma PRIMICIA.
- ✚ Implementación de las funcionalidades diseñadas para el componente de transmisión de noticias a través de la red para la plataforma PRIMICIA.
- ✚ Aplicación de pruebas al componente de transmisión de noticias para identificar posibles errores y corregirlos en el menor tiempo posible.

Preguntas de la investigación

- ✚ ¿Cuáles son los elementos principales de los procesos de transmisión de señales digitales a través de una red IP?
- ✚ ¿Qué características debe tener el componente informático que permita transmitir las noticias a través de una red IP?
- ✚ ¿Cómo organizar el proceso de desarrollo del componente de transmisión de noticias a través de la red?
- ✚ ¿El componente informático desarrollado permite la transmisión de noticias a través de la red?

Métodos científicos de la investigación

Para lograr el desarrollo y cumplimiento del objetivo planteado se hace uso de un conjunto de métodos científicos, los cuales se presentan a continuación:

Métodos teóricos

Análisis y Síntesis: se analizó la información disponible, permitiendo la extracción de los elementos más importantes que se relacionan con los procesos de transmisión de señales digitales a través de una red IP.

Modelación: se estudió la información referente a la transmisión de señales digitales en una línea lógica de tiempo. Se analizaron los principales avances que han tenido los procesos de transmisión desde sus

inicios hasta la actualidad. Finalmente, se identificaron elementos importantes a la hora de transmitir contenidos multimedia a través de una red IP, lo que permitió que el equipo de desarrollo elaborara su propio conocimiento de forma lógica y respondiendo a las características de la solución a desarrollar.

Estructura del trabajo de diploma:

Capítulo I. Fundamentación Teórica del componente de transmisión de noticias a través de la red para la plataforma PRIMICIA.

En este capítulo se expone la fundamentación teórica que da sustento al trabajo de diploma, abordando los elementos conceptuales que fundamentan la investigación incluyendo temas sobre el proceso de transmisión de multimedia. Se describen y definen las herramientas, lenguaje y metodología que guiará el desarrollo del componente informático.

Capítulo II. Análisis y diseño del componente de transmisión de noticias a través de la red para la plataforma PRIMICIA.

En este capítulo se describe el modelo de dominio para representar los conceptos más importantes de los objetos que se muestran en el dominio del problema. Se especifican los requisitos funcionales y no funcionales necesarios que debe cumplir el componente. Se realiza el diagrama de casos de uso y el modelo de diseño del sistema para permitir un mejor entendimiento de las funcionalidades del componente. Además se explica la arquitectura a utilizar para la solución.

Capítulo III. Implementación y validación del componente de transmisión de noticias a través de la red para la plataforma PRIMICIA.

En este capítulo se implementa el componente, donde se presentan los elementos fundamentales de esta fase. Además se realiza el proceso de validación del sistema a partir de las pruebas realizadas a la aplicación, para así verificar si la solución cumple con lo esperado.

Capítulo I. Fundamentación Teórica del componente de transmisión de noticias a través de la red para la plataforma PRIMICIA

1.1. Introducción

Los fundamentos teóricos constituyen la base sobre la cual se realiza la presente investigación, por ello en este capítulo se realiza una descripción detallada de cada elemento teórico que posibilitará comprender, definir y explicar la situación problemática que dio origen al problema a resolver y así guiar el desarrollo investigativo hacia una solución. Además se analizan los diferentes tipos de metodología de desarrollo de software para definir cuál utilizar, así como las herramientas y tecnologías que permiten llevar a cabo la modelación, diseño y construcción de la futura solución.

1.2. Términos asociados al dominio del problema

Para que sea mejor comprendida la investigación, a continuación se definen los principales conceptos asociados a los procesos de transmisión de señales digitales a través de la red.

Señales digitales

Son variables que toman un número finito de niveles o estados entre un mínimo y un máximo. Las más utilizadas son las binarias que sólo pueden tener dos niveles que se asignan a los números binarios 0 y 1. (Tomasi, 2003). (Miyara, 2004) Concuerdia con dicha afirmación diciendo que: son variables eléctricas con dos niveles bien diferenciados que se alternan en el tiempo transmitiendo información según un código previamente acordado. Cada nivel eléctrico representa uno de dos símbolos: 0 ó 1, V o F, etc. Teniendo en cuenta los puntos de vistas de diferentes autores y llevándolos al dominio del problema se puede decir que las señales digitales son creadas por el hombre y se codifican a conveniencia del mismo. Mediante las mismas se puede transmitir información y a medida que varíe el tiempo estas también van a variar.

Multimedia

Los archivos multimedia permiten la presentación de diferentes tipos de códigos y lenguajes, que van desde los textuales hasta los íconos sonoros e íconos visuales, tanto de forma estática como dinámica. Esta presentación no se realiza de forma lineal y secuencial, sino altamente ramificada, permitiendo que el sujeto en la interacción con el medio pueda avanzar por la información de forma personal. De esta forma el mismo puede construir de forma significativa el conocimiento, el cual responderá a las necesidades que en ese momento se plantee (Cabero, 2002).

Por otro lado (Salaverría, 2001) va un poco más lejos y divide el término multimedia en dos subconjuntos: el primer plano es el comunicativo, donde el adjetivo multimedia identifica a aquellos mensajes informativos transmitidos, presentados o percibidos unitariamente a través de múltiples medios. El segundo es el plano instrumental, donde multimedia equivale a los múltiples intermediarios que pueden participar en la transmisión de un producto informativo, tanto si este producto es multimedia en el sentido comunicativo como si no lo es.

Resumiendo lo antes expuesto y tomando como frontera los dominios de la investigación, se puede decir que **multimedia** es toda la información en formato de texto, imagen, audio o video que es utilizada para apoyar el proceso de información al personal.

Streaming

Es una técnica de encapsulamiento de la información para distribuir audio y/o video por internet de forma continua sin necesidad de ninguna descarga previa (Zavala, y otros, 2011). Además, (Prieto, 2008) agrega que se utiliza para acceder a contenidos multimedia que no se encuentran en la PC en la que se van a reproducir y que suelen tener un gran volumen de información. Una vez analizados los diferentes conceptos con el empleo de distintas fuentes de conocimientos, se puede definir al **streaming** como la técnica de encapsular contenidos multimedia para transmitirlos por Internet sin interrupción y estos puedan ser visualizados sin necesidad de ser descargados previamente en el ordenador cliente.

1.3. Procesos de transmisión de señales digitales a través de una red IP

Son los procesos encargados del envío de información a través de medios físicos de comunicación guiados (par trenzado, cable coaxial y fibra óptica) o no guiados (transmisión inalámbrica) en forma de señales digitales (Verdejo, y otros, 2003). Entre sus funciones no sólo está la transmisión inmediata, también deben permitir la transmisión de señales analógicas y para ello deben de digitalizarlas primero. Sin embargo, como la información digital no puede ser transmitida en forma de 0 y 1, debe ser codificada en forma de una señal digital, por ejemplo: dos niveles de voltajes con respecto a la conexión a tierra, la diferencia de voltaje entre dos cables y la presencia o ausencia de corriente en un cable. (Tomasi, 2003).

Entre las ventajas que presentan estos procesos se encuentran que: aunque una señal se distorsione al ser transmitida, puede ser reconstruida en el destino gracias a su carácter discreto. Brindan un grado de abstracción elevado, independizando la comunicación de la naturaleza de la información a enviar. Además, cuando se desea transmitir una señal a larga distancia no se necesita de repetidores de señal. (Rodríguez, y otros, 2012). Cuando lo que se transmite es un texto, una imagen, sonido, video o cualquier combinación entre ellos; se está en presencia de una transmisión de contenidos multimedia (Alonso, 2011).

1.3.1. Situación Problemática

La transmisión en PRIMICIA se basa en hacer llegar las noticias al cliente en diferentes formatos: texto, imagen, sonido y video. Estos contenidos multimedia almacenados previamente en la unidad de almacenamiento del ordenador origen deben viajar hasta varios TVs destino. Para ello se cuenta con un fichero XML (XML Configuración) en el cual se guardan una serie de parámetros de la transmisión como son: nombre del servidor, nombre de la base de datos, contraseña, puerto, entre otros. El mismo es de gran importancia, en casos en los que se desee desplegar a PRIMICIA; no se necesita recompilar el proyecto, solo se debe modificar el fichero con los datos convenientes

Para realizar una transmisión se necesita que la aplicación se esté ejecutando sobre una PC. Además, es necesario un dispositivo CDA para que los TVs puedan visualizar la información transmitida. Los contenidos multimedia viajan por una red de cable coaxial hasta los diferentes televisores cuando la transmisión es cableada. En caso de ser satelital, se necesita situar una estación de tierra en un lugar accesible para la señal procedente del centro de transmisión. Dicha estación debe estar equipada con una antena que establezca la conexión ascendente con el satélite.

Luego, la señal que recibe el satélite se retransmite hacia el área geográfica deseada por una antena adecuada que cumpla con las características necesarias: frecuencia y potencia. Esta señal la reciben los receptores situados en tierra, bien sea un usuario individual o una comunidad (Hernández García, y otros, 2007). De esta forma, PRIMICIA es capaz de brindarle un espacio de información al cliente de una forma continua; pero el contenido que se emite sólo llega hasta los televisores. No se pueden visualizar las noticias en otros dispositivos.

Además, utilizar a PRIMICIA significa tener que comprar dispositivos CDA, ya sean tarjetas gráficas o cualquier otro dispositivo que convierta las señales analógicas en digitales. Esto supone gastos adicionales y los clientes no estiman conveniente comprar un producto que depende de un hardware con costos variables en el mercado. Es contradictorio pagar por un software que no resuelva por sí solo los problemas por los que se compró. Esta situación provoca que se vean limitadas las oportunidades de negocio del centro GEYSED.

Es necesario crear las condiciones para que PRIMICIA sea compatible con otros dispositivos de visualización. Se necesita eliminar cualquier dependencia que tenga la plataforma para su transmisión, la misma debe ser capaz de transmitir sin necesidad de un dispositivo CDA. Teniendo en cuenta estos aspectos se necesita que el subsistema de transmisión modifique el procedimiento a la hora de transmitir recursos multimedia.

1.4. Análisis de soluciones existentes

A continuación se analizan diferentes formas de transmitir contenidos multimedia a través de la red con el objetivo de encontrar una solución que proporcione respuesta al problema planteado.

1.4.1. Soluciones existentes a nivel internacional

Servidor de video streaming de la empresa Komova.net

Estrategia utilizada por la empresa Komova.net para fortalecer el servicio de “Comunicación e Imagen”. Consiste en un servidor de video *streaming* que facilita la transmisión de contenidos multimedia a través de internet y no requieren descarga previa en el ordenador del usuario. Básicamente funciona de la siguiente manera: un usuario (emisor) envía una señal de video al servidor, el servidor (receptor-repartidor) recepciona esta señal y la reparte a todos los demás usuarios que deseen visualizar la transmisión mediante un flujo de *streaming*. Los usuarios finales (receptores) sólo deben de conectarse al servidor streaming directamente o mediante el sitio web perteneciente a la empresa (Komova.net) y tener instalado un reproductor de contenidos multimedia (VLC, Real Video, etc.).

Sin embargo, Komova.net no define detalladamente que procedimientos, técnicas y tecnologías utiliza para realizar el streaming. Dicha empresa se limita a ofrecer sus servicios, no a divulgar la forma en que lo hacen. Sería poco conveniente para PRIMICIA delegar la responsabilidad de transmitir sus contenidos multimedia a una entidad ajena al proyecto. Se vería comprometida la integridad de las noticias y no sería PRIMICIA quien controlaría el proceso de transmisión.

VLC Media Player

Es un reproductor multiplataforma, gratuito y de código abierto que no sólo se dedica a la reproducción de contenidos multimedia sino que puede realizar la función de servidor streaming. El mismo fue desarrollado originalmente por VideoLAN³ aunque actualmente sigue incorporando funcionalidades gracias a la extensa comunidad de colaboradores con que cuenta (VideoLAN, 2015). Básicamente funciona de la siguiente manera: un usuario (emisor) ejecuta el reproductor VLC Media Player en su PC, elige el recurso multimedia que desea transmitir y selecciona la opción emitir. La transmisión saldrá por la dirección ip⁴ de la PC del emisor y todos los usuarios (receptores) que se conecten directamente con la PC del emisor desde otras estaciones de trabajo podrán recibir el flujo de streaming.

³ Proyecto originalmente integrado por estudiantes de la Escuela Central de París en 1996. Actualmente desarrollado por colaboradores de todas partes del mundo. (VideoLAN, 2015).

⁴ Etiqueta numérica que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (generalmente a una computadora) dentro de una red que utilice el protocolo Internet Protocol (IP).

Sin embargo, una precondition necesaria para realizar un streaming con este reproductor es que se debe tener el recurso multimedia que se desea transmitir almacenado con antelación. PRIMICIA necesita que su propia transmisión sea un streaming, teniendo en cuenta que dicha transmisión se realiza en tiempo real es imposible utilizar a VLC Media Player como solución.

1.4.2. Soluciones existentes a nivel nacional

Módulo transmisión del Sistema de Transmisión de Canales Virtuales (STCV)

Módulo encargado de la gestión de los canales que se van a transmitir. Provee la transmisión a través de un flujo de *streaming* de los recursos multimedia según la planificación previamente establecida en el módulo de programación. Las transmisiones se realizan de forma automática atendiendo a las especificaciones realizadas durante la planificación y programación de la transmisión. Sin embargo, esta solución más que asegurar que una transmisión llegue sin problemas a un cliente o trabajador concentra sus esfuerzos en la gestión de varias transmisiones. Brinda prioridad a que varios canales puedan transmitirse armónicamente y coloca en un segundo plano que las noticias lleguen a su destino, por lo que deja de ser una alternativa factible para solucionar las limitaciones que presenta PRIMICIA.

Al finalizar el análisis de los ejemplos de transmisión anteriores no se encontró una solución para el problema planteado. Sin embargo, se comprendió a mayor escala la importancia de una solución y se identificó un elemento a la hora de transmitir recursos multimedia que puede incluirse en el componente informático a desarrollar. Proveer la transmisión a través de un flujo de *streaming* de los recursos multimedia es una solución bastante acertada dentro de la plataforma PRIMICIA.

Es evidente que trae grandes beneficios pues se pueden transmitir las noticias por la red; aumentando de esta forma el alcance de transmisión de las noticias y se dota al producto con la capacidad de llegar al cliente o al trabajador por vías alternativas como son: teléfonos móviles, tablas, laptop, etc. Se elimina la dependencia existente de un dispositivo CDA, aumentando así las oportunidades del negocio.

1.5. Proceso y herramientas a utilizar en la modelación, diseño y construcción del componente informático

El componente informático que se desea desarrollar formará parte del proyecto PRIMICIA. Teniendo en cuenta esta afirmación se debe utilizar el mismo proceso de desarrollo y apoyarse en las mismas herramientas definidas por el arquitecto de PRIMICIA. De esta forma se garantiza compatibilidad entre el proyecto y el producto informático en desarrollo. A continuación se puntualizan las principales características del proceso y las herramientas utilizadas.

1.5.1. Metodología de desarrollo de software

Para desarrollar el componente de transmisión de noticias a través de la red se utiliza la metodología Proceso Racional Unificado (*Rational Unified Process*, RUP). La misma es una metodología tradicional y es utilizada por el proyecto PRIMICIA. A continuación se presentan las características de RUP que se evidencian en el desarrollo del componente de transmisión de noticias a través de la red:

- ✚ Reconoce que las necesidades del usuario y sus requerimientos no se pueden definir completamente al principio. (Pressman, 2005).
- ✚ Genera una amplia documentación, bien detallada en cuanto a cómo las tareas serán ejecutadas y los resultados que se deben obtener. (Pressman, 2005).
- ✚ Permite evaluar tempranamente los riesgos en lugar de descubrir problemas en la integración final del sistema.

1.5.2. Lenguaje de Modelado Unificado

El Lenguaje de Modelado Unificado (UML 2.0) es un lenguaje para la especificación, visualización, construcción y documentación de los 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. (Jacques Garcías, 2012).

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. Permite especificar las características que tendrá el sistema, incluso antes de su construcción. (Coromoto, 2007).

1.5.3. Herramienta para el modelado

Visual Paradigm 8.0 es una herramienta profesional para el modelado con UML. Ideal para analistas y arquitectos que están interesados en construir sistemas informáticos a gran escala. Garantiza confiabilidad y estabilidad en el desarrollo orientado a objetos. Permite representar todos los tipos de diagramas de clases, generar código desde diagramas y documentación. Es fácil de instalar, actualizar y tiene compatibilidad entre ediciones. Posee una licencia gratuita y comercial. (visual-paradigm, 2015)

Además, permite que múltiples usuarios trabajen sobre el mismo proyecto y posibilita el control de versiones. Características a tener en cuenta ya que PRIMICIA constituye un proyecto con un equipo de desarrollo bastante grande. Teniendo en cuenta lo antes expuesto se decidió emplear Visual Paradigm para UML pues se ajusta a las necesidades del proyecto y soporta todo el ciclo de vida del software.

1.5.4. Lenguaje de programación

C++ 4.4 es un lenguaje de programación orientado a objetos al que se le añadieron características y cualidades de las que carecía el lenguaje C. Tiene una gran potencia en la programación a bajo nivel aunque posee elementos que permiten programar a alto nivel. Es un lenguaje de programación híbrido, por lo que existen varios IDEs que lo soportan como son el Eclipse y el QT Creator (Informática.com, 2006).

Principales características (Delgado, 2011)

- ✚ Genera programas eficientes.
- ✚ La posibilidad de poder ser compilado en una variedad de computadoras, con pocos cambios (portabilidad).
- ✚ Posee una bibliografía extensa y confiable para su aprendizaje.
- ✚ Es un lenguaje rápido en cuanto a tiempo de ejecución.

Además C++ tiene a su favor que es utilizado por un conjunto de herramientas libres de desarrollo, consta de operadores para el manejo de memoria persistente y algunas utilidades adicionales basadas en bibliotecas externas. Es un lenguaje multiplataforma y se utiliza actualmente en el subsistema de transmisión de PRIMICIA. El mismo cubre las necesidades existentes de seleccionar un lenguaje de programación rápido, eficiente y que brinde las condiciones necesarias para la construcción exitosa del componente informático.

1.5.5. Framework de desarrollo

Qt 5.3 es un framework de desarrollo integral con herramientas diseñadas para simplificar la creación de aplicaciones e interfaces de usuario para el escritorio, embebidos y plataformas móviles. Permite realizar

aplicaciones avanzadas y desplegarlas en escritorio y sistemas operativos integrados sin tener que reescribir el código fuente, contribuyendo de esta forma con el tiempo y el costo de desarrollo del producto. Es fácil de usar, aprender, mantener y de código reutilizable, además posee un alto rendimiento en tiempo de ejecución y ocupa poco espacio en disco (Garrido, 2009).

Qt ostenta varias herramientas y componentes que facilitan el trabajo a los desarrolladores, a continuación se muestran algunas (Garrido, 2009).

- ✚ Bibliotecas Qt: son clases escritas en C++ que facilitan el desarrollo del producto.
- ✚ QtDesigner: utilizado para diseñar y crear interfaces visuales de forma cómoda.
- ✚ QtAssistant: sencillo y potente visor de documentación que proporciona múltiples maneras de acceder a la información.

Para la implementación del componente de transmisión de noticias a través de la red se utiliza Qt 5.3 como framework de desarrollo.

1.5.6. Entorno de desarrollo (IDE)

Qt Creator 3.1 es un entorno de desarrollo integrado (en inglés Integrated Development Environment) multiplataforma que proporciona herramientas para diseñar y desarrollar aplicaciones informáticas utilizando el framework Qt. (Rischpater, 2014)

Características que se destacan de QT Creator (Rischpater, 2014):

- ✚ Avanzado editor de código C++.
- ✚ Tiene integrado diseñadores de interfaz gráfica de usuario.
- ✚ Ayuda sensible al contexto.
- ✚ Resaltado y autocompletado de código

Mediante las investigaciones desarrolladas anteriormente se puede ver que este IDE posee las características necesarias para el desarrollo de este componente informático. También se ha decidido elegirlo puesto que es utilizado por el subsistema de transmisión de PRIMICIA.

1.6. Conclusiones parciales

Mediante el desarrollo del capítulo se llegó a la conclusión que es necesario modificar el proceso de transmisión actual de PRIMICIA. El mismo provoca que se vean limitadas las oportunidades de negocio del centro productivo GEYSED. Proveer la transmisión de los recursos multimedia mediante un flujo de

streaming brinda grandes beneficios e incrementa las oportunidades del negocio. Las herramientas seleccionadas para el desarrollo del componente fueron las definidas por el arquitecto del proyecto PRIMICIA.

Finalmente, se logró comprender la investigación y se identificaron los contenidos bibliográficos necesarios a profundizar. De esta forma se creó una base de conocimiento sobre lo que acontece nacional e internacionalmente respecto al fenómeno estudiado, constituyendo un inicio acertado en el ciclo de vida del software.

Capítulo II. Análisis y diseño del componente de transmisión de noticias a través de la red para la plataforma PRIMICIA

2.1. Introducción

En este capítulo se realiza el análisis y diseño de la solución propuesta. Teniendo en cuenta que no se tienen bien definidos los procesos de negocio, se realiza el modelo de dominio para entender el funcionamiento de la plataforma PRIMICIA a la hora de transmitir las noticias. Además, se enumeran los requisitos funcionales y no funcionales que debe tener el componente de transmisión de noticias a través de la red partiendo de conocer las condiciones y cualidades deseadas por los usuarios. Posteriormente se realiza el diagrama de casos de usos del sistema y la descripción detallada de cada caso de uso. También se realizan los diagramas de clase de análisis y diseño que permitirán un mejor entendimiento de las funcionalidades del módulo.

2.2. Modelo de dominio

El Modelo de Dominio o Modelo Conceptual permite representar los conceptos más importantes de los objetos que se muestran en el dominio del problema. Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes software. No se trata de un conjunto de diagramas que describen clases software, u objetos software con responsabilidades, es el encargado de capturar los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema (Larman, 2003).

Este modelo de dominio constituye el punto de arranque para lograr realizar un diseño acertado del producto a desarrollar, definiendo a partir del mismo un conjunto de clases conceptuales u objetos existentes del mundo real en el dominio del problema planteado.

2.2.1. Principales eventos del entorno

PRIMICIA actualmente la conforman dos subsistemas que se relacionan entre sí, Administración y Transmisión. El primero es el encargado de administrar el canal y gestionar las noticias y recursos multimedia; mientras que el subsistema de transmisión se encarga de la visualización de las noticias y los materiales publicados. El proceso de transmisión de noticias inicia cuando la aplicación comienza a ejecutarse. La información llega a los usuarios de forma satisfactoria a través de una red de televisión en diferentes formatos (texto, imagen, audio y video).

2.2.2. Diagrama de clases del Modelo de Dominio

En la figura se muestran los principales conceptos asociados al dominio del problema y las relaciones que se establecen entre éstos.

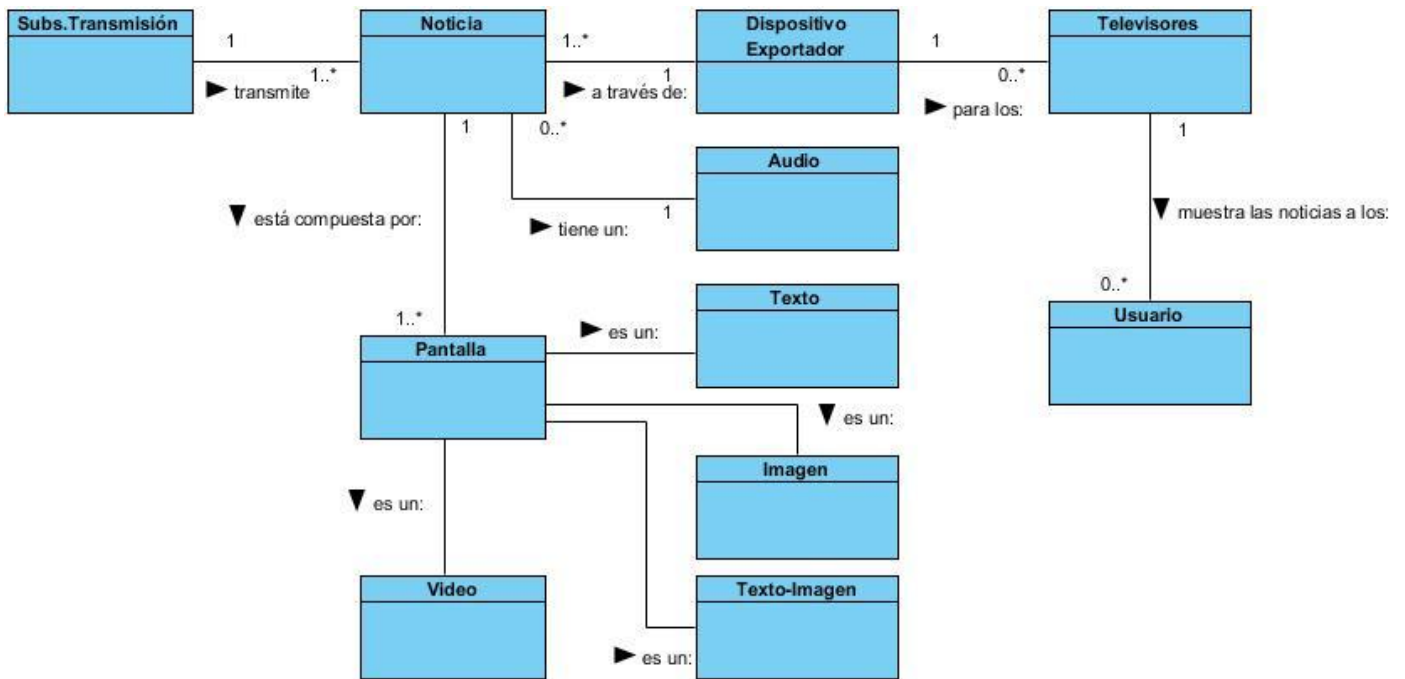


Ilustración 1. Diagrama de clases del Modelo de Dominio

2.2.3. Principales conceptos asociados al dominio del problema

A continuación se muestran los conceptos que sirven de base para el entendimiento de la presente investigación:

- ✚ *Noticia*: información que se desea dar a conocer.
- ✚ *Pantalla*: combinación de contenidos multimedia (texto, imagen y video) utilizada para conformar una noticia.
- ✚ *Dispositivo Exportador*: dispositivo que exporta la señal digital del ordenador a una señal analógica.
- ✚ *Televisores*: estaciones destino donde se visualizan los contenidos multimedia que transmite PRIMICIA.
- ✚ *Usuario*: persona interesada en las noticias que transmite PRIMICIA.
- ✚ *Texto*: secuencia de caracteres que brinda una información.
- ✚ *Imagen*: ilustración cuyo objetivo es apoyar el entendimiento de una información.

- ✚ *Texto-Imagen:* combinación de un texto con una imagen.
- ✚ *Video:* imagen en movimiento.
- ✚ *Audio:* sonido que acompaña la transmisión.

2.3. Especificación de los requisitos de software

Para llevar a cabo el desarrollo de un software, es necesario saber qué debe hacer dicho sistema y cómo debe hacerlo, tener bien claro cuáles son las expectativas del cliente con respecto al sistema que se va a desarrollar. Incluye un conjunto de casos de uso que describe todas las interacciones que tendrán los usuarios con el software.

2.3.1. Requisitos Funcionales

Los requisitos funcionales definidos para el desarrollo del componente informático son los siguientes:

RF 1: Configurar fotogramas por segundos.

El sistema debe permitir configurar cuantos fotogramas por segundo deben visualizarse durante la transmisión.

RF 2: Seleccionar el formato de salida del video.

El sistema debe permitir seleccionar los formatos de salida de video (mp4, webm, ogv).

RF 3: Seleccionar la cantidad de canales de salida del audio.

El sistema debe permitir seleccionar la cantidad de canales de salida del audio (1 ó 2).

RF 4: Crear plantilla de configuración.

El sistema debe permitir crear nuevas plantillas de configuración de la transmisión.

RF 5: Modificar plantilla de configuración.

El sistema debe permitir modificar los campos de las plantillas de configuración de la transmisión.

RF 6: Eliminar plantilla de configuración.

El sistema debe permitir eliminar las plantillas de configuración de la transmisión.

RF 7: Buscar plantilla de configuración.

El sistema debe permitir buscar las plantillas de configuración de las transmisiones que se encuentren creadas con antelación.

RF 8: Visualizar plantilla de configuración.

El sistema debe permitir visualizar las plantillas de configuración de la transmisión seleccionada.

RF 9: Notificar estado de la conexión.

El sistema debe notificar al usuario si se encuentra conectado a la red o no.

RF 10: Transmitir señal.

El sistema debe transmitir el contenido multimedia capturado de una aplicación en ejecución por una red IP.

RF 11: Detener transmisión.

El sistema debe permitir detener la transmisión de las noticias.

2.3.2. Requisitos No Funcionales

RNF 1: Requisitos de software del servidor de transmisión:

✚ El sistema operativo que soportará la solución será GNU/Linux Ubuntu 14.10.

RNF 2: Requisitos de software de los dispositivos de captura de la transmisión:

✚ El dispositivo (PC conectada a la red o cualquier otro dispositivo móvil capaz de visualizar contenidos multimedia y conectarse a una red IP) que capture la transmisión debe tener instalado el reproductor VLC en su versión 2.2.0 o superior.

RNF 3: Requisitos de hardware del servidor de transmisión:

Mínimos:

✚ Memoria RAM⁵: 2Gb.

✚ Procesador (CPU): Core i3 a 3.3GHz.

⁵ Memoria de Acceso Aleatorio: Se utiliza como memoria de trabajo de computadoras para el sistema operativo, los programas y la mayor parte del software.

Recomendados por el equipo de desarrollo:

- ✚ Memoria RAM: 32Gb o más.
- ✚ Procesador (CPU): Core i7-5960x a 3.3GHz.

2.4. Descripción de la solución propuesta

2.4.1. Descripción de los actores del sistema

Durante el desarrollo del componente informático se definió un único actor. A continuación se describe el mismo y se define el papel que juega en el sistema:

<i>Actor</i>	<i>Descripción</i>
Usuario	Es la persona que interactúa directamente con la aplicación, tiene acceso a todas las funcionalidades, no tiene que autenticarse.

Tabla 1. Actores del sistema

2.4.2. Diagrama de Casos de Uso del Sistema

A partir de la identificación del actor que interactúa con el componente informático, así como la recopilación del conjunto de funcionalidades agrupadas en casos de uso según sus particularidades, se conforma el Modelo de Casos de Uso que se representa a continuación.

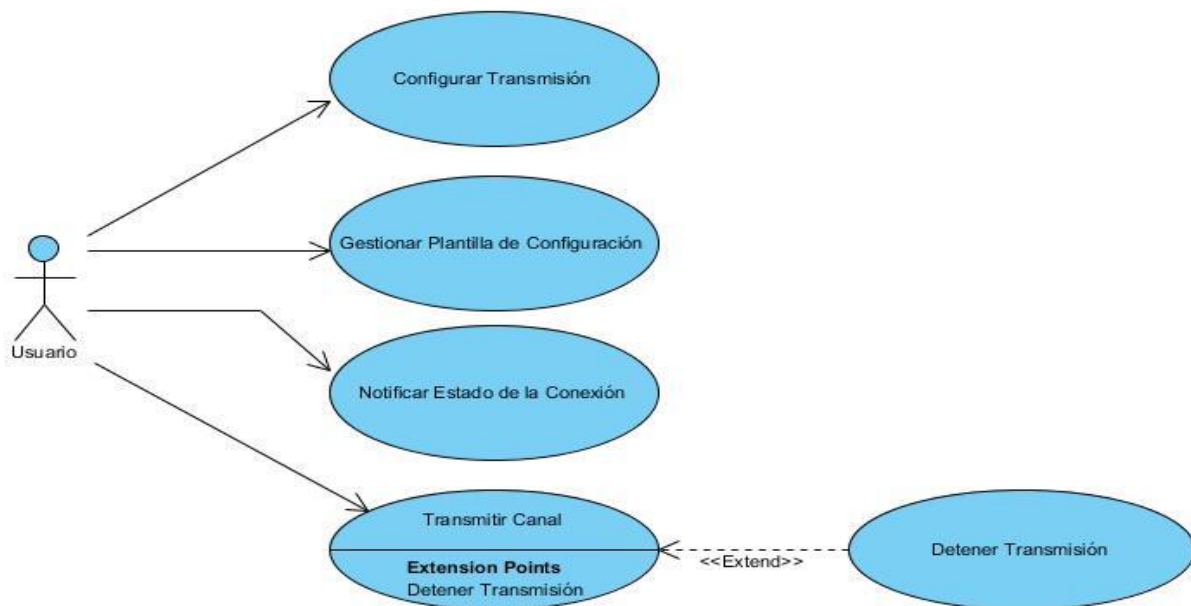


Ilustración 2. Diagrama de Casos de Uso del Sistema

Relación entre Casos de uso y Requisitos Funcionales

Casos de uso del sistema	Requisitos Funcionales
Configurar Transmisión.	RF 1: Configurar fotogramas por segundos. RF 2: Seleccionar el formato de salida del video. RF 3: Seleccionar la cantidad de canales de salida del audio.
Gestionar Plantilla de Configuración.	RF 4: Crear plantilla de configuración. RF 5: Modificar plantilla de configuración. RF 6: Eliminar plantilla de configuración. RF 7: Buscar plantilla de configuración. RF 8: Visualizar plantilla de configuración.
Notificar Estado de la Conexión.	RF 9: Notificar estado de la conexión.
Transmitir Canal.	RF 10: Transmitir señal.
Detener Transmisión.	RF 11: Detener transmisión.

Tabla 2. Relación entre Casos de uso y Requisitos Funcionales

2.4.3. Descripción de los Casos de Uso

En la especificación de los Casos de Uso (CU) se describe, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. A continuación se expone una descripción de los principales casos de uso que conforman el componente.

Descripción del CU Transmitir canal

Objetivo	Asegurar la transmisión de los contenidos multimedia.
Actores	Usuario: (Inicia) Iniciar Proceso.
Resumen	El CU inicia cuando el usuario indica al sistema que comience a transmitir (clic en el botón Iniciar Proceso). El CU termina cuando se inicia la transmisión en el sistema.
Complejidad	Alta
Prioridad	Alta
Precondiciones	Campos pertenecientes al CU “Configurar transmisión” no se encuentren vacíos (fps, formato de salida del video y la cantidad de canales que tendrá la salida del

	audio.).	
Poscondiciones	Transmisión iniciada en el sistema.	
Flujo de eventos		
Flujo básico Transmitir canal		
	Actor	Sistema
1.	Indica al sistema que comience a transmitir dando clic en el botón “Iniciar Proceso”.	
2.		Verifica que no exista una transmisión en curso.
3.		Desactiva todas las funcionalidades relacionadas con la gestión de las configuraciones.
5.		Comienza a transmitir con la configuración que tenga en ese momento. Termina el CU.
Flujos alternos		
2a Evento: Existe una transmisión en curso.		
	Actor	Sistema
3a		Muestra una alerta indicando al usuario que existe una transmisión en curso. Termina el CU.
Relaciones	CU incluidos	No existen.
	CU extendidos	Detener transmisión.
Requisitos no funcionales	No procede.	
Asuntos pendientes	No procede.	

Tabla 3. Descripción de Caso de uso Transmitir canal

Descripción del CU Gestionar plantilla de configuración

Objetivo	Crear, modificar, eliminar o buscar plantillas de configuración de la transmisión.	
Actores	Usuario: (Inicia) Crea, modifica, elimina y busca las plantillas de configuración de la transmisión.	
Resumen	El CU inicia cuando el usuario crea, modifica, elimina o busca una plantilla de configuración, el sistema valida los datos insertados en los caso que corresponda. El CU termina cuando se inserta, modifica, elimina o busca una plantilla de configuración en el sistema.	
Complejidad	Media	
Prioridad	Media	
Precondiciones		
Poscondiciones	Plantilla de configuración creada exitosamente. Datos de plantilla de configuración modificados exitosamente. Plantilla de configuración eliminada exitosamente. Datos de Plantilla de configuración encontrados y visualizados exitosamente.	
Flujo de eventos		
Flujo básico Gestionar plantilla de configuración		
	Actor	Sistema
1.	Selecciona una de las siguientes opciones: <ul style="list-style-type: none"> • Crear plantilla de configuración. Ver paso 2 del flujo básico Gestionar usuarios. • Modificar plantilla de configuración. Ver Sección 1: Modificar plantilla de configuración. • Eliminar plantilla de configuración. Ver Sección 2: Eliminar plantilla de 	

	<p>configuración.</p> <ul style="list-style-type: none"> • Buscar plantilla de configuración. Ver Sección 3: Buscar plantilla de configuración. 	
2.		Muestra el formulario correspondiente para crear una nueva plantilla de configuración.
3.	<p>Inserta los datos de la plantilla de configuración: identificador (nombre), fps, formato de salida del video y la cantidad de canales que tendrá la salida del audio.</p>	
4.	<p>Selecciona la opción Aceptar.</p>	
5.		<p>Verifica que el formato del identificador introducido sea el correcto (Sólo letras y números, específicamente en ese orden.).</p>
6.		<p>Verifica que el identificador introducido no exista.</p>
7.		<p>Verifica que ningún campo haya quedado vacío.</p>
8.		<p>Genera de forma automática un número para la plantilla de configuración, que sea incremental con respecto a la última plantilla de configuración creada.</p>
9.		<p>Registra la nueva plantilla de configuración, finalizando el CU.</p>
Flujos alternos		
4a Evento Selecciona la opción Cancelar.		
	Actor	Sistema
5a		<p>Cancela la operación. Termina el CU.</p>
Flujos alternos		
5ª Evento El formato del identificador introducido es incorrecto.		

	Actor	Sistema
6a		Muestra una alerta indicando al usuario que el formato del identificador es incorrecto. Regresar al paso 3 del flujo básico Gestionar plantilla de configuración.
Flujos alternos		
6a Evento El identificador insertado ya existe.		
	Actor	Sistema
7a		Muestra una alerta indicando al usuario que ya existe una plantilla de configuración registrada en el sistema con ese identificador. Regresar al paso 3 del flujo básico Gestionar plantilla de configuración.
Flujos alternos		
7a Evento Existen campos vacíos.		
	Actor	Sistema
8a		Muestra una alerta indicando al usuario que existen campos vacíos. Regresar al paso 3 del flujo básico Gestionar plantilla de configuración.
Sección 1: “Modificar plantilla de configuración”		
Flujo básico Modificar plantilla de configuración		
	Actor	Sistema
2.		Muestra un listado con las plantillas de configuración gestionadas hasta ese momento.
3.	Selecciona una plantilla de configuración de la lista mostrada.	
4.	Selecciona la opción Modificar plantilla de	

	configuración.	
5.		Muestra un formulario con los datos de la plantilla de configuración seleccionada.
6.	Modifica los datos de la plantilla de configuración seleccionada: identificador (nombre), fps, formato de salida del video y la cantidad de canales que tendrá la salida del audio.	
7.	Selecciona la opción Aceptar.	
8.		Verifica que el formato del identificador introducido sea el correcto
9.		Verifica que el identificador introducido no exista.
10.		Verifica que ningún campo haya quedado vacío.
11.		Actualiza los datos de la planilla de configuración seleccionada y finaliza el CU.

Flujos alternos

3a Evento El usuario no selecciona una plantilla de configuración.

	Actor	Sistema
4a		Muestra una alerta indicando al usuario que debe seleccionar una configuración para poder modificarla. Regresar al paso 2 de la sección 1 Modificar plantilla de configuración.

Flujos alternos

7a Evento Selecciona la opción Cancelar.

	Actor	Sistema
8a		Cancela la operación. Termina el CU.

Flujos alternos		
8a Evento El formato del identificador introducido es incorrecto.		
	Actor	Sistema
9a		Muestra una alerta indicando al usuario que el formato del identificador es incorrecto. Regresar al paso 6 de la sección 1 Modificar plantilla de configuración.
Flujos alternos		
9a Evento El identificador introducido ya existe.		
	Actor	Sistema
10a		Muestra una alerta indicando al usuario que ya existe una plantilla de configuración registrada en el sistema con ese identificador. Regresar al paso 6 de la sección 1 Modificar plantilla de configuración.
Flujos alternos		
10a Evento Existen campos vacíos.		
	Actor	Sistema
11a		Muestra una alerta indicando al usuario que existen campos vacíos. Regresar al paso 6 de la sección 1 Modificar plantilla de configuración.
Sección 2: “Eliminar plantilla de configuración”		
Flujo básico Eliminar plantilla de configuración		
	Actor	Sistema
2.		Muestra un listado con las plantillas de configuración gestionadas hasta ese momento.
3.	Selecciona una plantilla de configuración	

	de la lista de configuración.	
4.	Selecciona la opción Eliminar plantilla de configuración.	
5.		Muestra un mensaje de confirmación.
6.	Selecciona la opción Aceptar.	
7.		Elimina la plantilla de configuración seleccionada.
8.		Actualiza el listado de las plantillas de configuración. Termina el CU.

Flujos alternos

3a Evento El usuario no selecciona una plantilla de configuración.

	Actor	Sistema
4a		Muestra una alerta indicando al usuario que debe seleccionar una configuración para poder eliminarla. Regresar al paso 2 de la sección 2 Eliminar plantilla de configuración.

Flujos alternos

6a Evento Selecciona la opción Cancelar.

	Actor	Sistema
7a		Cancela la operación. Termina el CU.

Sección 3: “Buscar plantilla de configuración”

Flujo básico Buscar plantilla de configuración

	Actor	Sistema
2.		Muestra un formulario para que el usuario decida el criterio de búsqueda.
3.	Elige el criterio de búsqueda: identificador	

	o formato de salida del video.	
4.	Selecciona la opción Aceptar.	
5.		El sistema busca todas las plantillas de configuración que coincidan con el criterio de búsqueda especificado.
6.		El sistema muestra un listado con todas las plantillas de configuración resultantes de la búsqueda.
7.	Selecciona la plantilla de configuración de la cual desea visualizar sus datos.	
8.		El sistema visualiza los datos de la plantilla de configuración seleccionada. Finaliza el CU.

Flujos alternos

4a Evento Selecciona la opción cancelar.

	Actor	Sistema
5a		Cancela la operación. Termina el CU.

Flujos alternos

6a Evento No existe plantilla de configuración que coincida con el criterio de búsqueda.

	Actor	Sistema
7a		Muestra un mensaje indicando que no existe ninguna plantilla de configuración con el parámetro especificado. Regresar al paso 3 sección 3 Buscar plantilla de configuración.

Relaciones	CU incluidos	No existen.
	CU extendidos	No existen.

Requisitos no funcionales	No procede.
Asuntos pendientes	No procede.

Tabla 4. Descripción de Caso de uso Gestionar plantilla de configuración

2.5. Modelo del diseño

Una vez especificados los requisitos y definidos los casos de usos del sistema, normalmente se realiza la fase de análisis. La fase de análisis tiene dos propósitos fundamentales: refinar los casos de uso con más detalles y establecer la asignación inicial de funcionalidad del sistema a los objetos para que estos proporcionen el comportamiento.

El propósito y objetivo del análisis debe alcanzarse de algún modo en todo el proyecto. La manera exacta de ver y de emplear el análisis puede diferir de un proyecto a otro. Una de las variantes que se pueden emplear es no utilizar en absoluto este modelo para describir los resultados del análisis. En cambio el proyecto analiza los requisitos como parte integrada de la captura de requisitos o en el diseño (Pressman, 2005).

La metodología utilizada para el desarrollo del componente (RUP) es un proceso configurable, por lo que no se está obligado a hacer uso de todas las actividades y artefactos definidos. El proceso se puede configurar para que se adapte a aquellas partes que se consideran necesarias, por lo que se decidió prescindir de la realización del Modelo de Análisis en el desarrollo de esta investigación debido a que:

- ✚ Es posible obtener un mayor formalismo en el modelo de casos de uso pues el cliente es capaz de comprender los resultados que estos pueden proporcionar.
- ✚ Los requisitos son simples, bien conocidos y se cuenta con cierta comprensión de los mismos.
- ✚ Se logra evitar el costo en tiempo y recursos de mantener este flujo.
- ✚ El lenguaje de programación, el framework y las tecnologías en general sobre las cuales se estará desarrollando el sistema son elementos dominados.

Teniendo en cuenta que se decidió no realizar el modelo de análisis, se puede pasar directamente al Modelo de Diseño (MD). El MD es un modelo de objetos que describe la relación física de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además, el modelo de diseño

sirve de abstracción de la implementación del sistema y es, de ese modo, utilizada como una entrada fundamental de las actividades de implementación (Pressman, 2005).

2.5.1. Arquitectura de software

La arquitectura del software constituye una especificación de las principales ideas del diseño proporcionando una descripción más detallada de cómo realizar dicho sistema. Contiene varias vistas la Vista Lógica, la Vista de Despliegue, la Vista de Implementación y la Vista de Casos de Uso. Es un conjunto de patrones que proporcionan un marco de referencia necesario para servir de guía en la construcción del software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación. Es considerada el nivel más alto en el diseño de la arquitectura de un sistema puesto que establecen la estructura, funcionamiento e interacción entre las partes del software.

2.5.2. Arquitectura en n capas

La arquitectura en capa define el patrón en capas como una organización jerárquica, posibilitando un diseño basado en niveles de abstracción creciente, facilitándole a los implementadores, particionar un problema en una secuencia de pasos incrementales. Este estilo de desarrollo en varios niveles, provee que en caso que ocurra algún cambio, sólo se tendrían que realizar las correcciones necesarias en el nivel requerido sin tener que revisar código de otros niveles. (Introducción a la Arquitectura del Software II, 2013).

Se escogió este tipo de arquitectura ya que es la utilizada por PRIMICIA actualmente. El desarrollo se puede llevar a cabo en varios niveles. En caso de que acontezca algún cambio, solo se ataca al nivel requerido sin tener que revisar entre el código mezclado. Es más sencillo darle mantenimiento y soporte.

Se definió la arquitectura en tres capas, denominadas como: *capa acceso a datos*, *capa presentación* y *capa lógica de negocio*. La capa de presentación es la encargada de contener la interfaz del sistema y tiene el objetivo de mostrar a los usuarios las informaciones deseadas, esta capa se comunica únicamente con la capa de negocio llevando y trayendo los datos o registros necesarios, la capa de lógica de negocio es la encargada de establecer todos los procesos que deben realizar, por otra parte la capa de acceso a datos es donde residen los datos y es la encargada de acceder a los mismos.

2.5.3. Patrones de diseño

A la hora de desarrollar un Software es importante el uso de Patrones de diseño, pues son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. En otras palabras, brindan una solución

ya probada y documentada a problemas de desarrollo de software, constituyen una guía para resolver problemas comunes en programación (Microsoft, 2003).

Los patrones de diseño se dividen en dos grupos: GRASP⁶ y GOF⁷. Los patrones GRASP son patrones generales de software para asignación de responsabilidades y los patrones GOF, se clasifican en 3 categorías: creacionales, estructurales y de comportamiento. Para el desarrollo del componente fueron utilizados los siguientes patrones:

Patrones GRASP

Experto: es el encargado de asignar una responsabilidad al experto en información, constituyendo este la clase que cuenta con la información necesaria para cubrir la responsabilidad (Montero Vals, y otros, 2013). En este trabajo este tipo de patrón es utilizado a la hora determinar las responsabilidades con las cuales debe de cumplir cada clase y las funcionalidades que realizan cada una de ellas. Este patrón es utilizado pues las clases que intervienen en la aplicación (**Formulario**, **Control**, **Emisor** y **Fichero**) poseen implementadas solo las funcionalidades que responden a la responsabilidad que deben asumir.

Creador: soluciona el problema de ¿Quién debería ser responsable de crear una nueva instancia? Guía la asignación de responsabilidades relacionadas a la creación de objetos, una tarea muy común en sistemas orientados a objetos. El intento básico de este patrón es encontrar un creador que necesite estar conectado al objeto creado en un evento en particular. Separa la construcción de un objeto complejo de su representación, de forma que el mismo proceso de construcción pueda crear diferentes representaciones. Este patrón se pone de manifiesto en la clase **Formulario**. (Montero Vals, y otros, 2013).

Controlador: esta es la responsable de asignar a clases específicas la responsabilidad de controlar la salida de sucesos en el sistema, permitiendo a través de esta interactuar con las demás clases del componente (Montero Vals, y otros, 2013). La utilización de este patrón se evidencia en la clase **Control**.

Alta Cohesión: la alta cohesión es la medida en la que un componente se dedica a realizar solo la tarea para la cual fue creado, delegando las tareas complementarias a otros componentes. (Una clase debe de hacer lo que respecta a su entidad, y no hacer acciones que involucren a otra clase o entidad) (Montero Vals, y otros, 2013).

⁶ GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (Patrones generales de software para asignar responsabilidades).

⁷ GOF es un acrónimo que significa Gang of Four (Grupo de los cuatro).

Bajo Acoplamiento: Soluciona el problema de ¿Cómo dar soporte a las bajas dependencias y al incremento de la reutilización? Plantea tener las clases lo menos ligadas entre sí, de tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización y disminuyendo la dependencia entre las mismas. (Montero Vals, y otros, 2013).

Patrones GOF

Fábrica Abstracta: el problema a solucionar de este patrón es el de crear diferentes familias de objetos, por ejemplo la creación de interfaces gráficas de distintos tipos (ventana, menú, botón, etc.) (Prieto, 2008). Este patrón se evidencia en la clase **Formulario**. En esta clase crean distintos objetos con el objetivo de poder manejar la información necesaria para gestionar la transmisión.

Observador: Define una dependencia entre objetos de forma que si un objeto cambia de estado los demás sean notificados y se actualicen automáticamente (Prieto, 2008). Este patrón se evidencia en la clase **Formulario**. De acuerdo al estado que tengan los botones **Iniciar transmisión** y **Detener transmisión**, cambiará el estado de varios objetos dentro de la clase.

2.5.4. Diagrama de clases del diseño

Un diagrama de clases es un diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Son utilizados durante el proceso de análisis y diseño de los sistemas para lograr una mejor interpretación de la implementación del mismo.

A continuación se muestra una representación del diagrama de clases del diseño de los casos de usos más importantes representados siguiendo la arquitectura definida. Se representan las clases y las relaciones existentes entre estas, los atributos, métodos y dependencias.

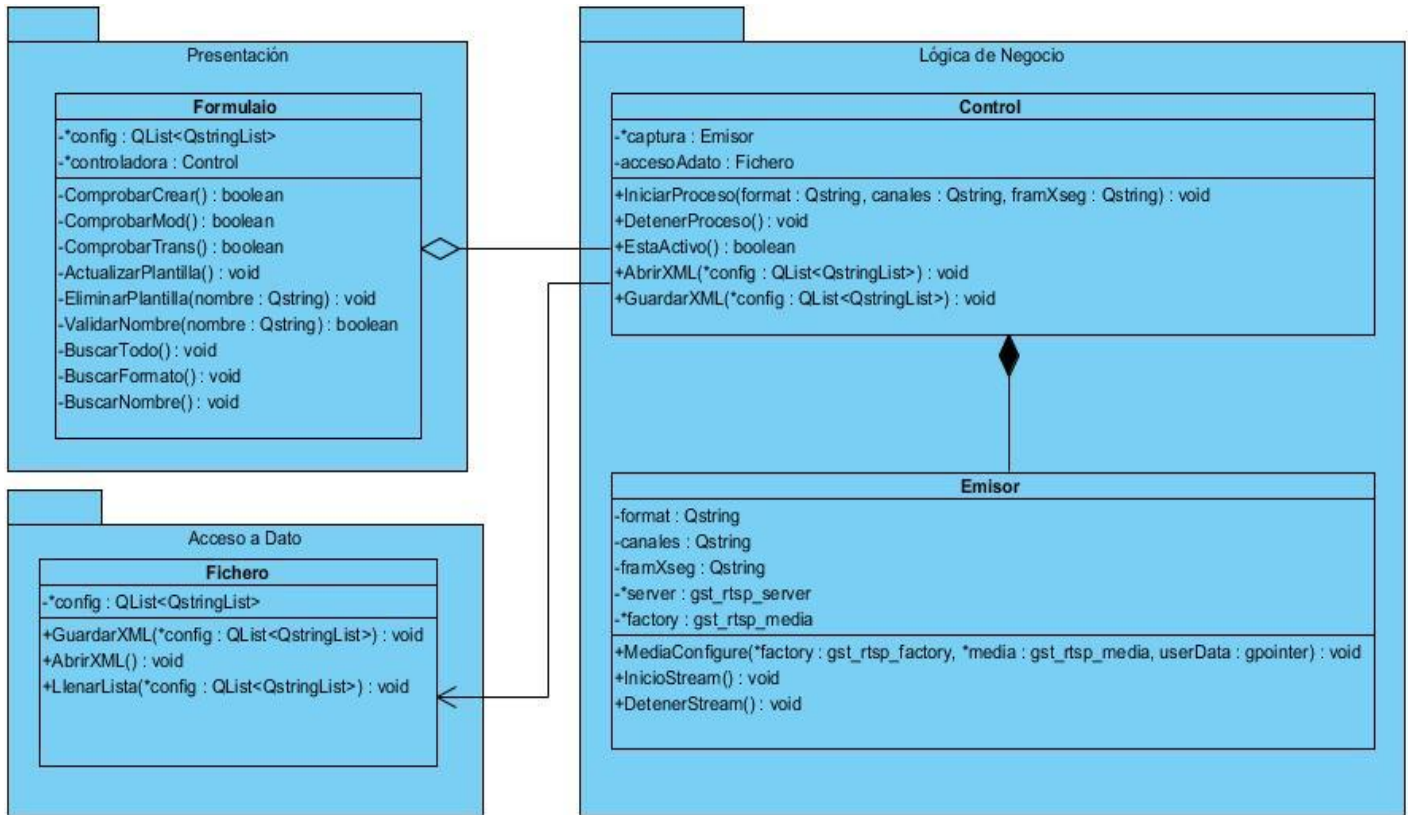


Ilustración 3. Diagrama de Clases del Diseño

La clase **Form1** permite la interacción con el usuario facilitando la gestión de las plantillas de configuración de la transmisión y el control del proceso de transmisión. Para ello utiliza una instancia de la clase **Control**, la cual a través de una instancia de la clase **Ficheros**, permite guardar y cargar el archivo donde se almacenan las configuraciones. **Control** también es la encargada de recibir los parámetros de la configuración para ejecutar la transmisión a través de una instancia de la clase **Emisor** que es la encargada de crear la tubería de ejecución de la librería de Gstreamer de forma dinámica.

2.6. Conclusiones parciales

Actualmente en PRIMICIA no existe un proceso de negocio visible y bien definido. Sin embargo; se logró comprender mejor el funcionamiento del sistema utilizando como apoyo el modelo de dominio, pues se identificaron y detallaron los principales eventos, términos y conceptos presentes en el entorno donde éste trabajará.

Quedaron plasmadas las prestaciones y características del componente de transmisión de noticias a través de la red mediante el proceso de captura de requisitos. Se determinó que la transmisión de contenidos

multimedias es la función principal del software esperado y se identificaron los casos de usos del sistema como elemento clave en el desarrollo del producto informático según la metodología RUP.

En el diagrama de clases del diseño se representaron las clases y las relaciones existentes entre ellas de manera detallada. Estas clases fueron agrupadas teniendo en cuenta la arquitectura en tres capas, lo que permitió tener una correcta estructura de los elementos de software, simplificar la complejidad y tener una organización en capas. Como resultado principal de este capítulo quedó plasmada la propuesta de solución para el problema actual del sistema, detallada a través de los actores, casos de usos y sus diagramas, así como las especificaciones de los casos de uso.

Capítulo III. Implementación y validación del componente de transmisión de noticias a través de la red para la plataforma PRIMICIA

3.1. Introducción

En este capítulo se describe la implementación y validación del sistema. Se realizan los diagramas de componentes y despliegue del componente de transmisión de noticias a través de la red. Además se desarrollan las pruebas a la aplicación con el objetivo de validar que el sistema cumpla con la calidad requerida.

3.2. Modelo de implementación

Modelo donde se describe como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados. Además refleja la dependencia entre los componentes (Ornelas, 2003).

3.2.1. Diagrama de componentes

Diagrama que muestra la organización y las dependencias entre un conjunto de componentes. Puede ser un tipo especial de diagrama de clases que se centra en los componentes físicos del sistema. Se utiliza para modelar la gestión de la configuración de los archivos de código fuente, tomando como productos de trabajo precisamente estos archivos (Grau, y otros, 2011). A continuación se muestra el diagrama de componentes perteneciente a la aplicación a desarrollar:

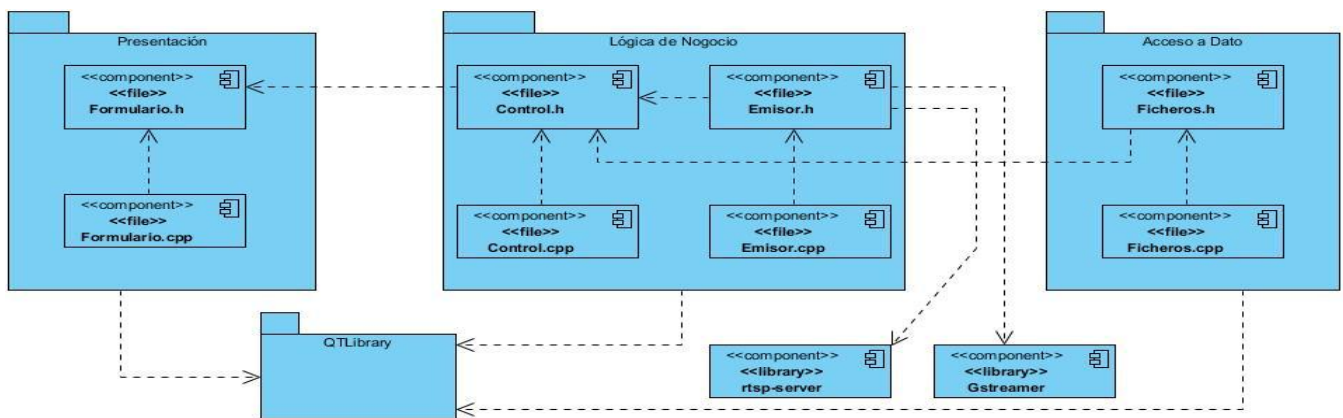


Ilustración 4. Diagrama de Componentes

3.3. Modelo de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo hardware similar. Los nodos poseen relaciones que representan medios de comunicación entre ellos. El modelo de despliegue además puede describir diferentes configuraciones de red (García Peñalvo, y otros, 2008).

3.3.1. Diagrama de despliegue

Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria (Marca Huallpara, y otros, 2003).

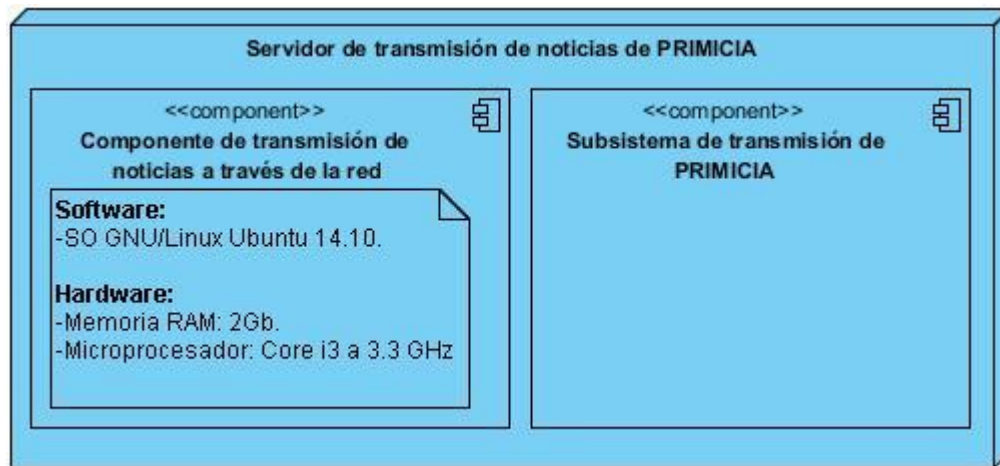


Ilustración 5. Diagrama de Despliegue

Servidor de transmisión de noticias de PRIMICIA: representa la PC donde se encuentra instalado el subsistema de transmisión de PRIMICIA. Al mismo tiempo es donde se ejecutará el componente informático desarrollado.

3.4. Estándar de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo

práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez (Microsoft, 2003).

La utilización de un estándar de codificación constituye una buena práctica de programación que reduce considerablemente la posibilidad de cometer errores, finalmente se obtiene una aplicación con código legible y comprensible por otras personas del equipo de desarrollo. Para la elaboración del estándar de codificación utilizado se tuvo en cuenta las características del framework de desarrollo Qt el entorno integrado de desarrollo Qt Creator y el lenguaje de programación C++. A continuación se muestran algunas de las características con las cuales cumple la implementación del subsistema de transmisión de noticias a través de la red:

- ✚ En la definición del nombre de las clases se utiliza *UpperCamelCase*⁸ como estándar de codificación.
- ✚ En la nomenclatura de los métodos se utiliza *lowerCamelCase*⁹ como estándar de codificación.
- ✚ Dentro de la estructura del código las líneas pueden tener hasta 150 caracteres.
- ✚ Para separar palabras se usa la diferenciación entre mayúscula y minúscula.
- ✚ Las funciones implementadas llevan comentarios describiendo el objetivo de la función si se considera necesario, pero no la descripción del funcionamiento.
- ✚ Se declaran las propiedades de una clase antes que los propios métodos de la misma.

3.5. Pruebas

Las pruebas realizadas a una aplicación son el proceso que permite verificar y revelar la calidad de un software, básicamente es una fase en el desarrollo de software donde se prueban las aplicaciones. Pueden integrarse dentro de las diferentes fases del ciclo del software y determinan el nivel de calidad de un producto informático mediante medidas o pruebas que permiten comprobar el grado de cumplimiento respecto a las especificaciones iniciales del sistema (Ruiz Tenorio, 2010).

Algunos objetivos de las pruebas (Ruiz Tenorio, 2010):

- ✚ Ejecución del programa con la intención de descubrir errores.
- ✚ Confección de casos de pruebas con altas probabilidades de mostrar un error no descubierto hasta entonces.

8 UpperCamelCase: Estilo de codificación que se aplica a frases o palabras compuestas donde la primera letra de cada una de las palabras es mayúscula.

9 lowerCamelCase: Estilo de codificación que se aplica a frases o palabras compuestas igual que la anterior con la excepción de que la primera letra es minúscula.

- ✚ Definición de hasta qué punto las funciones del software funcionan de acuerdo con las especificaciones y cuán cerca se está de alcanzar los requisitos de rendimiento.
- ✚ Definición de la fiabilidad del software a partir de los datos obtenidos en las pruebas.

3.5.1. Pruebas de rendimiento, carga y estrés

Las pruebas de rendimiento son un subconjunto de la ingeniería de pruebas, una práctica informática que se esfuerza por mejorar el rendimiento del sistema englobándose en el diseño y la arquitectura. Además puede servir para investigar, medir, validar o verificar otros atributos de la calidad del sistema; tales como la escalabilidad, la fiabilidad y el uso de recursos. (Echeverría Perez, y otros, 2014).

Las pruebas de carga son usadas para determinar y validar la respuesta de la aplicación cuando es sometida a una carga de usuarios y/o transacciones que se espera en el ambiente de producción. Mientras que las pruebas de estrés se utilizan para encontrar el volumen de datos o tiempo en que la aplicación comienza a fallar o es incapaz de responder a las peticiones que se le realizan. (Echeverría Perez, y otros, 2014).

Estos tipos de pruebas deben realizarse siempre antes que el software entre en producción para reducir la probabilidad de eventos inesperados. Por ejemplo: diversos estudios han indicado que los usuarios no toleran una interfaz con tiempo de respuesta mayor a 8 segundos (Echeverría Perez, y otros, 2014), por lo que se puede afirmar que el tiempo de respuesta es una pauta fundamental para aumentar las oportunidades del negocio.

Diseño de prueba de rendimiento, carga y estrés

Para disminuir las probabilidades de que el componente de transmisión de noticias a través de la red funcione incorrectamente, se diseñó una prueba para: monitorizar el consumo de memoria RAM y de la Unidad Central de Procesamiento (CPU) mientras se ejecuta la aplicación por un tiempo prolongado. Observar el comportamiento de dicho consumo mientras aumenta la cantidad de usuarios conectados y determinar el número máximo de peticiones que soporta antes de colapsar.

Servidor de transmisión (estación de trabajo 1):

- ✚ Sistema Operativo:
GNU/Linux Ubuntu 14.10.
- ✚ Hardware:
Procesador (CPU): Core i5 a 3.3GHz.

Memoria RAM: 4Gb.

Tiempo de transmisión (horas).	1	2	3
Usuarios Conectados.	2	4	6
Porcentaje de uso del CPU (%).	31-37	31-39	33-37
Memoria RAM consumida (Mb).	1.300	2.150	3.900

Tabla 5. Rendimiento del CPU y la RAM

La aplicación funciona sin ninguna dificultad pero con el incremento de los elementos multimedia (principalmente videos) y las conexiones al servidor, puede ocurrir una caída de los FPS¹⁰ en el contenido multimedia que se visualizará en los dispositivos conectados. El uso de la memoria RAM varía entre los 1.300 y 3.900 MB y el porcentaje de uso del CPU oscila entre 31 y 39%; aumentando o disminuyendo en dependencia de la cantidad de peticiones que se le hagan al servidor y la cantidad de elementos multimedia mostrados en la pantalla.

Servidor de transmisión (estación de trabajo 2):

🚦 Sistema Operativo:

GNU/Linux Ubuntu 14.10.

🚦 Hardware:

Procesador (CPU): Core i3 a 3.3GHz.

Memoria RAM: 2Gb.

Tiempo de transmisión (horas).	1	2	3
Usuarios Conectados.	2	3	4
Porcentaje de uso del CPU(%).	52-54	52-58	52-57
Memoria RAM consumida	1.214	1.950	-

¹⁰ Fotogramas por segundos.

(Mb).			
-------	--	--	--

Tabla 6. Rendimiento del CPU y la RAM (2)

La aplicación responde correctamente a un número limitado de peticiones realizadas al servidor. El flujo de *streaming* llega de forma continua a los dispositivos que se conectan al servidor mientras las peticiones realizadas no excedan la capacidad de respuesta de la memoria RAM.

3.5.2. Prueba de caja negra

Es un tipo de prueba realizada sobre la interfaz del software para validar las funciones del mismo. Verifica que la entrada de datos al sistema sea de forma adecuada y que se produzca un resultado correcto. Examina algunos aspectos del modelo fundamental del sistema sin prestar mucha atención a la estructura lógica interna del software y se centra en los requisitos funcionales de la aplicación. (Pressman, 2005). Estas pruebas permiten encontrar:

- ✚ Errores de interfaz.
- ✚ Errores de rendimiento.
- ✚ Errores en estructuras de datos.

Diseño de prueba de caja negra

En la fase de prueba se le realizaron dos iteraciones al sistema, se encontró una no conformidad en la primera iteración. El caso de uso Transmitir canal no estaba funcionando correctamente, la aplicación permitía que se comenzara una transmisión existiendo campos vacíos. Esta no conformidad se analizó y se le dio solución inmediatamente. Luego, se realizó una segunda iteración de prueba para verificar que se había erradicado el mal funcionamiento identificado. Esta última concluyó de manera satisfactoria, no se encontró ningún problema a la hora de transmitir las noticias. A continuación se presentan los casos de pruebas resultantes de la segunda iteración referentes a los casos de uso Configurar transmisión y Transmitir canal.

V1: Formato de salida del video.

V2: Fotogramas por segundo de la transmisión.

V3: Cantidad de canales que tendrá la salida del audio.

Escenarios a probar en el caso de uso Configurar transmisión

Escenario	Descripción	V1	V2	V3	Respuesta	Flujo central
-----------	-------------	----	----	----	-----------	---------------

					<i>del sistema</i>		
EC	2.1	El usuario introduce los datos pertenecientes a la configuración de la transmisión.	V mp4	V 25	V 2	Configuración lista para transmitir.	- Se ejecuta la aplicación. Se introducen los datos pertenecientes a la configuración de la transmisión. -Se ejecuta la aplicación. Se selecciona una configuración del listado existente. Se modifican los valores que tomarán los campos pertenecientes a la configuración a conveniencia del usuario.
EC	2.2	El usuario introduce los datos pertenecientes a la configuración de la transmisión. (Queda al menos un campo vacío.)	I vacío	V 15	V 1	Configuración incorrecta, no se podrá transmitir	-Se ejecuta la aplicación. Se introducen los datos pertenecientes a la configuración de la transmisión. -Se ejecuta la aplicación. Se selecciona una configuración del listado existente. Se modifican los valores que tomarán los campos pertenecientes a la configuración a conveniencia del usuario.
	V ogv		I vacío	V 2			
	V webm		V 25	I vacío			

Tabla 7. Escenarios a probar en el caso de uso Configurar transmisión**Escenarios a probar en el caso de uso Transmitir canal**

<i>Escenario</i>	<i>Descripción</i>	<i>Respuesta de sistema</i>	<i>Flujo central</i>
EC 1.1 Transmitir canal correctamente.	El usuario selecciona la opción de comenzar a transmitir.	La aplicación comienza a transmitir las noticias.	Se ejecuta la aplicación, clic en el botón Iniciar Proceso.
EC 1.2 Transmitir canal de forma incorrecta.	El usuario selecciona la opción de comenzar a transmitir. (Existen campos de la configuración de la transmisión vacíos.)	La aplicación no comienza a transmitir las noticias.	Se ejecuta la aplicación, clic en el botón Iniciar Proceso.

Tabla 8. Escenarios a probar en el caso de uso Transmitir canal**3.5.3. Prueba de caja blanca**

La prueba de caja blanca es un método de diseño de caso de prueba que usa la estructura de control de diseño procedimental para obtener los casos de prueba (CP). Este método permite obtener los CP que garanticen que (Ruiz Tenorio, 2010):

- ✚ Se ejercitan al menos una vez todos los caminos independientes de cada módulo.
- ✚ Se ejercitan todas las decisiones lógicas en sus vertientes verdaderas y falsas.
- ✚ Se ejecutan todos los bucles en sus límites y con sus límites operacionales.
- ✚ Se ejercitan las estructuras internas de datos para asegurar su validez.

Una de las técnicas utilizadas para realizar este tipo de prueba es la del camino básico, esta técnica permite al diseñador de casos de pruebas obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Los CP obtenidos del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa (Ruiz Tenorio, 2010).

Los pasos a seguir para la realización de las pruebas de caja blanca son los siguientes:

1. Generar el grafo de flujo de datos.
2. Calcular la complejidad ciclomática, $V(G)$.

$$V(G) = NA \text{ (Número de aristas)} - NN \text{ (Número de nodos)} + 2.$$

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

$$V(G) = \text{Número de regiones.}$$

La complejidad ciclomática es una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa. Cuando se usa en el contexto del método de prueba del camino básico, define el número de caminos independientes del conjunto básico de un programa y especifica un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez (Ruiz Tenorio, 2010).

3. Determinar los caminos independientes o básicos.
4. Generar un CP para cada camino de ejecución.

Diseño de prueba de caja blanca

Para realizar la prueba de caja blanca con la técnica de camino básico según los pasos analizados anteriormente se necesita seleccionar el código fuente al cuál se le desea aplicar el procedimiento.

Método MediaConfigure de la clase Emisor (ver Anexo 2)

Paso 1: Generar el grafo.

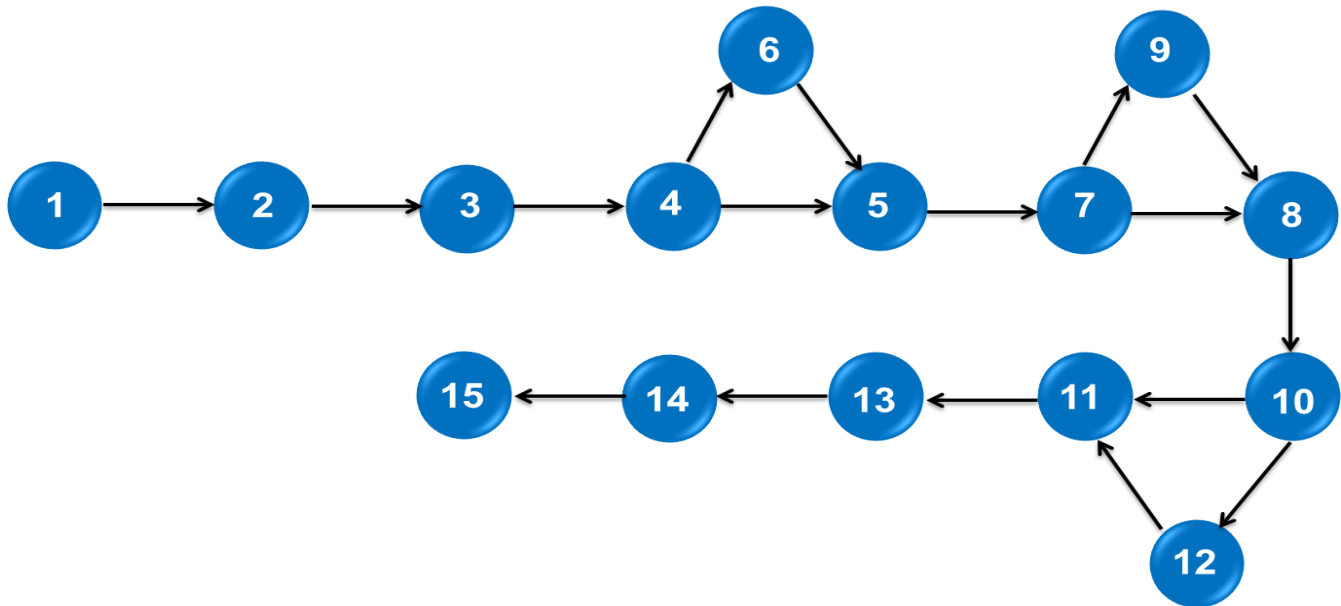


Ilustración 6. Grafo del método MediaConfigure

Paso 2: Calcular la complejidad ciclomática.

$V(G) = NA$ (Número de aristas) – NN (Número de nodos) + 2.

$V(G) = 17 - 15 + 2 = 4$

Paso 3: Determinar los caminos básicos.

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

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

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

CB 4: 1-2-3-4-5-7-8-10-11-13-14-15

Paso 4: Caso de prueba para el camino básico.

✚ *Caso de prueba para el camino básico 1*

Entrada: `if(formato.toLatin1().data()==".mp4").`

Resultado esperado: Transmisión iniciada con formato mp4.

Resultado de la prueba: Satisfactorio.

✚ *Caso de prueba para el camino básico 2*

Entrada: `if(formato.toLatin1().data()==".webm").`

Resultado esperado: Transmisión iniciada con formato webm.

Resultado de la prueba: Satisfactorio.

✚ *Caso de prueba para el camino básico 3*

Entrada: `if(this->formato.toLatin1().data()==".ogv").`

Resultado esperado: Transmisión iniciada con formato ogv.

Resultado de la prueba: Satisfactorio.

✚ *Caso de prueba para el camino básico 4*

Entrada: No se elige un formato de salida para la transmisión.

Resultado esperado: No se inicia la transmisión.

Resultado de la prueba: Satisfactorio.

Método InicioStream de la clase Emisor (ver Anexo 2)

Paso 1: Generar el grafo.



Ilustración 7. Grafo del método InicioStream

Paso 2: Calcular la complejidad ciclomática.

$$V(G) = NA \text{ (Número de aristas)} - NN \text{ (Número de nodos)} + 2.$$

$$V(G) = 2 - 3 + 2 = 1$$

Paso 3: Determinar los caminos básicos.

CB 1: 1-2-3

Paso 4: Caso de prueba para el camino básico.

🚩 *Caso de prueba para le camino básico 1*

Entrada: `gst_rtsp_mount_point_add_factory*(mount, "/primicia", factory).`

Resultado esperado: Transmisión iniciada.

Resultado de la prueba: Satisfactorio.

3.5.4. Pruebas de aceptación

Realizadas con el objetivo de validar que el sistema cumple con el funcionamiento esperado por el cliente y brinda la oportunidad a la directiva del proyecto PRIMICIA de determinar si está o no conforme con la solución obtenida atendiendo a la funcionalidad y rendimiento del software.

<i>Especialista</i>	<i>Cargo</i>
Ing. Yanary Hernández Sosa	Jefe de Proyecto
Ing. Susana Yaque Rivera	Analista
Ing. Aramis Romero Carballea	Programador

Tabla 9. Personas que participan en la prueba de aceptación

3.6. Conclusiones parciales

En el diagrama de componentes se describieron los componentes a construir y su organización. Además se representó la dependencia entre los nodos físicos en los que funcionará la aplicación. Durante la validación del componente mediante la prueba de rendimiento, carga y estrés; se detectó que el desempeño gráfico correspondiente a la aplicación se sustenta sobre la capacidad operacional tanto del microprocesador como de la memoria RAM. Se necesita un ordenador que posea buenas prestaciones respecto a estas características puesto que PRIMICIA requiere de una capacidad elevada de procesamiento gráfico y al servidor que transmite los contenidos multimedia se le harán varias peticiones.

Con la realización de las pruebas de caja blanca se obtuvieron casos de pruebas que garantizaron que se ejercitara por lo menos una vez todos los caminos independientes de las funcionalidades más importantes de la clase **Emisor** de forma correcta. Con la prueba de aceptación se pudieron ejecutar pruebas de software en un ambiente real y bajo la supervisión del equipo de desarrollo. Luego de haber realizado estas pruebas con resultados satisfactorios se puede garantizar que se disminuyeron las posibilidades de que el software presente algún error funcional.

Conclusiones generales

En el presente trabajo se generó una serie de artefactos que sirven de garantía para la continuidad del proyecto (mantenimiento y evolución). Posibilitando añadirle nuevas funcionalidades al producto informático obtenido tras la investigación realizada para dar cumplimiento al objetivo general planteado inicialmente. El uso de la metodología y las herramientas seleccionadas, posibilitó realizar y documentar el proceso de diseño e implementación del componente de transmisión de noticias a través de la red para la plataforma PRIMICIA. La utilización de las mismas fue adecuada para lograr obtener un producto final fiable y de calidad; avalado por las pruebas de aceptación, cumpliendo con todas las especificaciones planeadas por el cliente.

En PRIMICIA, la transmisión de contenidos multimedia no sólo se realiza mediante redes de TV o satélite, sino que pueden transmitirse por una red IP posibilitando su visualización por vías alternativas. Se puede afirmar que: con el desarrollo del componente de transmisión de noticias a través de la red para la plataforma PRIMICIA, se ha alcanzado satisfactoriamente el objetivo y se han cumplido todas las tareas propuestas para el presente trabajo de diploma.

Recomendaciones

El objetivo general trazado en el presente trabajo de diploma fue alcanzado satisfactoriamente; no obstante, se recomienda:

- ✚ Realizar un estudio con el objetivo de que se puedan realizar operaciones en el servidor de transmisión de contenidos multimedia sin afectar la información que reciben los dispositivos conectados a dicho servidor.

Referencias Bibliográficas

BBC Mundo. 2014. BBC Mundo. *La BBC en 10 preguntas.* [En línea] 2014. <http://www.bbc.co.uk>.

Cabero, Julio Almenara, Duarte, Ana Hueros. 2002. *Evaluación de medios y materiales en soporte multimedia.* Sevilla,Huelva : s.n., 2002.

Carrillo Pérez, Isaías, Pérez González, Rodrigo y Rodríguez Martín, Aurelino David. 2008. *Metodología de desarrollo del software.* 2008.

Coromoto, Janeth. 2007. *APLICACIÓN DE LA INGENIERÍA INVERSA DE SOFTWARE SOBRE EL SISTEMA DE APROVISIONAMIENTO DE LA RED BANDA ANCHA DE CANTV.* 2007.

Del Manifiesto Ágil sus valores y principios. **ELIÉCER HERRERA URIBE, LUZ ESTELA VALENCIA. 2007.** 2007, Vol. 34.

Delgado, Jordan. 2011. Teoría de Programación. [En línea] 2011. <http://teoria-de-programacion.globered.com>.

Echeverría Perez, MsC.Delvis, y otros. 2014. *Evaluación de la Eficiencia en Productos de Software. Experiencia desde el Departamento de Evaluación de Productos de Software de CALISOFT.* 2014.

estructuretv. [En línea]. [En línea] [Citado el: 22 de 10 de 2014.] <http://www.estructuretv.com>.Enterprises, MECK. eNewsRoom . eNewsRoom. [En línea] .

FUNDAMENTOS DE INGENIERÍA DE SOFTWARE. FUNDAMENTOS DE INGENIERÍA DE SOFTWARE. [En línea] [Citado el: 26 de 5 de 2015.] <https://sites.google.com/site/funddeingdesoftwareamairiani/home>.

García Peñalvo, Francisco José, Conde González, Miguel Ángel y Bravo Martín, Sergio. 2008. *Diseño orientado a objetos.* 2008.

Garrido, Salvador Alemany. 2009. *Introducción a QT. Programación grafica en C++ con Qt4.* 2009.

Grau, Xavier Ferré y Segura, María Isabel Sánchez. 2011. *Desarrollo Orientado a Objetos con UML.* 2011.

Guía a Rational Unified Process.

Hernández García, Ruber y Montaner Hernández, Yuniór. 2007. *Sistema Automatizado de Teletexto para la Plataforma de Televisión Digital Satelital Cubana.* La Habana : s.n., 2007.

Informática.com, Grupo de trabajo de La Revista. 2006. La Revista Informática.com. [En línea] 2006. <http://www.larevistainformatica.com..>

Internet, Software.com.ar. Available from. <<http://www.software.com.ar/visual-paradigm-para-uml.html>>. [En línea] [Citado el: 24 de 1 de 2013.]

Introducción a la transmisión digital. **Kizokea.net. 2014.** 2014.

Jacques Garcías, Fausto Abraham. 2012. *DESARROLLO Y APLICACIÓN DE UN SISTEMA DE SOPORTE A LA DECISIÓN DIANÓSTICO COGNITIVO EN WEB PARA LA GESTIÓN DE CONTENIDOS EN UN SISTEMA DE EDUCACIÓN VIRTUAL.* 2012.

Larman, Craig. 2003. *UML y Patrones.* 2003. 2ª Edición.

Marca Huallpara, Hugo Michael y Quisbert Limachi, Nancy Susana. 2003. *Diagrama de Despliegue.* 2003.

MARTÍNEZ, A. M. Y. R. *Guía a Rational Unified Process. Guía a Rational Unified Process.*

Martínez, Alejandro y Martínez, Raúl. 2002. *Proceso Unificado de Rational.* s.l. : Escuela Politécnica Superior de Albacete –Universidad de Castilla la Mancha, 2002.

Microsoft. 2003. *Microsoft Developer Network.* [En línea] 2003. <https://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.

Miyara, Federico. 2004. *Convertidores D/A y A/D. Electrónica III.* 2004.

Montero Vals, Ing. Adrián y Sospedra López, Ing. Diannet. 2013. *SISTEMA DE SUPERVISIÓN PARA CALL CENTERS.ARQUITECTURA, HERRAMIENTAS Y TECNOLOGÍAS.* 2013.

Ornelas, Raquel Ochoa. 2003. *Simulador gráfico de algoritmos de programación para computadoras.* 2003.

Penadés, M^a Carmen, Canó, José y Letelier, Patricio. 2003. *Metodologías Ágiles en el Desarrollo de Software.* Universidad Politécnica de Valencia : s.n., 2003.

Pérez Yuste, Antonio. 2006. *Baird y Jenkins: Pioneros de la Televisión Mecánica.* 2006.

Pierre, Albert y Tudesq, André-Jean. 2012. *Historia De La Radio y La Televisión.* 2012.

Pressman, Rogger. 2005. *Ingeniería de software. Un enfoque práctico.* 2005.

Prieto, Felix. 2008. *Programación III.ITI. de Sistemas. Patrones de Diseños.* 2008.

Prieto, Fernando Posada. 2008. *Que es el Streaming. Diseño de Materiales Multimedia_Web 2.0.* [En línea] 2008. <http://www.ite.educacion.es>.

Qt Creator. 2014. *Application Development with Qt Creator.* 2014.

Quintero, Juan Pablo Ortiz y Castro, Cristian Andrey Serna. 2006. *EVALUACION DE SERVIDORES DE STREAMING DE VIDEO ORIENTADO A DISPOSITIVOS MOVILES.* Medellín : s.n., 2006.

Rodríguez, Felix Ivan Romero. 2011. *Plataforma de televisión Informativa Primicia:Implementación del módulo de Redacción informativa PRIMICIA: Implementación del módulo de Redacción .* Ciudad de la Habana : Universidad de las Ciencias Informáticas. : s.n., 2011. s.n Tesis.

Ruiz Tenorio, Roberto. 2010. *Las Pruebas de Software y su Importancia en las Organizaciones.* 2010.

Salaverría, Ramón. 2001. *Aproximación al concepto de multimedia desde los planos comunicativo e instrumental.* Navarra : Estudios sobre el mensaje periodístico, 2001.

SONAPS. SONAPS Sistema de Producción en Red. *Sistema de Producción en Red. [En línea]* . [En línea] [Citado el: 22 de 10 de 2014.] [http://www.scribd.com/doc/8328747/Sonaps -Sistema-de-Produccion-en-Red](http://www.scribd.com/doc/8328747/Sonaps-Sistema-de-Produccion-en-Red).

Sonaps.Sonaps. [En línea] [Citado el: 22 de 10 de 2014.] [http://www.sony.fi/biz/lang/en/fi /product/network-..](http://www.sony.fi/biz/lang/en/fi/product/network-..)

Tendencias actuales en el uso de dispositivos móviles en educación. **Cantillo, Carmen Valero, Roura, Margarita Redondo y Sánchez, Ana Palacín. 2012.** 2012, La educ@ción digital megazine.

visual-paradigm. 2015. visual-paradigm.com. [En línea] 2015. <http://www.visual-paradigm.com/support/documents/vpuserguide.jsp>.

Zavala, Esequiel Huavel, y otros. 2011. *Streaming de Video en Vivo por Internet.* Lima : s.n., 2011.

Anexos

Anexo I. Descripción de casos de usos.

CU 3. Configurar transmisión

Objetivo	Asegurar que los parámetros necesarios para una transmisión tengan los valores específicos que desea el usuario antes de transmitir.	
Actores	Usuario: (Inicia) Ejecuta aplicación.	
Resumen	El CU inicia cuando un usuario ejecuta la aplicación y el sistema carga los datos que contiene el fichero de configuración.	
Complejidad	Baja	
Prioridad	Media	
Precondiciones		
Poscondiciones	Transmisión configurada en el sistema.	
Flujo de eventos		
Flujo básico Configurar transmisión		
	Actor	Sistema
1.		Carga los datos que se encuentran almacenados en el fichero de configuración (XML).
2.		Muestra la interfaz de configuración, el valor que toman los campos pertenecientes a la configuración de la transmisión son los valores pertenecientes a la última configuración seleccionada con la que se transmitió. Además, se actualiza una lista con los identificadores de todas las configuraciones registradas previamente en el sistema.
3.	Modifica los campos necesarios para una transmisión: fps, formato de salida del video (mkv, avi, mpeg) y número de canales (1 ó 2). Termina el CU.	
Flujos alternos		
1a Evento: El fichero de configuración no se encuentra disponible o está en blanco.		
	Actor	Sistema
2a		Muestra la interfaz de configuración con los campos vacíos. Regresar al paso 3 del flujo básico Configurar transmisión.

Flujos alternos		
3a Evento: El usuario no desea modificar ningún campo.		
	Actor	Sistema
4a		Termina el CU.
Flujos alternos		
3a Evento: El usuario selecciona un ID de la lista de configuraciones.		
	Actor	Sistema
4a		El valor que toman los campos pertenecientes a la configuración de la transmisión son los valores pertenecientes a la configuración del ID seleccionado. Regresar al paso 3 del flujo básico Configurar transmisión.
Relaciones	CU incluidos	No existen.
	CU extendidos	No existen.
Requisitos no funcionales	No procede.	
Asuntos pendientes	No procede.	

Tabla 10. Descripción de Caso de uso Configurar transmisión

CU 4. Detener transmisión

Objetivo	Detener la transmisión de los contenidos multimedia.
Actores	Usuario: (Inicia) Detener Proceso.
Resumen	El CU inicia cuando el usuario indica al sistema que deje de transmitir (click en el botón Detener Proceso). El CU termina cuando se detiene la transmisión en el sistema.
Complejidad	Baja
Prioridad	Media
Precondiciones	Transmisión iniciada.
Poscondiciones	Transmisión detenida en el sistema.

Flujo de eventos		
Flujo básico Autenticar usuarios		
	Actor	Sistema
1.	Indica al sistema que deje de transmitir dando click en el botón "Detener Proceso".	
2.		Verifica que exista una transmisión en curso.
3.		Deja de transmitir los contenidos multimedia.
4.		Activa todas las funcionalidades relacionadas con la gestión de las configuraciones. Termina el CU.
Flujos alternos		
2a Evento: No existe una transmisión en curso.		
	Actor	Sistema
3a		Muestra una alerta indicando al usuario que no existe una transmisión en curso. Termina el CU.
Relaciones	CU incluidos	No existen.
	CU extendidos	No existen.
Requisitos no funcionales	No procede.	
Asuntos pendientes	No procede.	

Tabla 11. Descripción de Caso de uso Detener transmisión

Anexo 2. Código de métodos utilizados en las pruebas de caja blanca.**Método MediaConfigure**

```

void emisor::MediaConfigure(gpointer *factory, gpointer *media_ptr, gpointer data_ptr)
{
qDebug()<<"AAAAAAAAAAAA";
GstElement *element, *pay,*pay2,
*origenVideo,*accionVideo,*codificacionVideo,*origenAudio,*accionAudio,*codificacionAudio,*encasulamien
to, *x2;

/* get the element used for providing the streams of the media */
GstRTSPMedia *media=(GstRTSPMedia*)media_ptr;
element=gst_rtsp_media_get_element (media);

/* get our appsrc, we named it 'mysrc' with the name property */
pay = gst_bin_get_by_name_recurse_up (GST_BIN (element), "pay0");
pay2 = gst_bin_get_by_name_recurse_up (GST_BIN (element), "pay1");

//gchar *asd="ximagesrc use-damage=0 startx=0 starty=0 endx=800 endy=600 xname=\"Inkscape\"
! videoconvert ! x264enc name=na";
origenVideo=gst_parse_launch("ximagesrc use-damage=0 startx=0 starty=0
xname=\"Inkscape\"",NULL);
if(formato.toLatin1().data()==" .mp4")
{
accionVideo=gst_parse_launch("videoconvert",NULL);
codificacionVideo=gst_parse_launch("x264enc name=na",NULL);
gst_bin_add_many(GST_BIN(element),origenVideo,accionVideo,codificacionVideo,NULL);
gst_element_link_many(origenVideo,accionVideo,codificacionVideo,pay,NULL);
}
if(formato.toLatin1().data()==" .webm")
{
accionVideo=gst_parse_launch("vp8enc",NULL);
codificacionVideo=gst_parse_launch("quality=10 speed=2",NULL);
}
}

```

```
gst_bin_add_many(GST_BIN(element),origenVideo,accionVideo,codificacionVideo,NULL);
gst_element_link_many(origenVideo,accionVideo,codificacionVideo,pay,NULL);
}
if(this->formato.toLatin1().data()=="ogv")
{
accionVideo=gst_parse_launch("theoraenc",NULL);
gst_bin_add_many(GST_BIN(element),origenVideo,accionVideo,NULL);
gst_element_link_many(origenVideo,accionVideo,pay,NULL);
}
```

```
origenAudio=gst_parse_launch("pulsesrc device=\"alsa_output.pci-0000_00_1b.0.analog-
stereo.monitor\"",NULL);
accionAudio=gst_parse_launch("audioconvert",NULL);
codificacionAudio=gst_parse_launch("vorbisenc",NULL);
encasulamiento=gst_parse_launch("rtpvorbispay name=pay1 pt=97",NULL);
gst_bin_add_many(GST_BIN(element),origenAudio,accionAudio,codificacionAudio/*,r*/,NULL);
gst_element_link_many(origenAudio,accionAudio,codificacionAudio,pay2/*,r*/,NULL);
}
```

Método InicioStream

void emisor::iniciarTransmision()

```
{
GstRTSPServer *server = (GstRTSPServer*) this->server;
GstRTSPMountPoints *mounts = gst_rtsp_server_get_mount_points (server);
this->factory=gst_rtsp_media_factory_new ();
GstRTSPMediaFactory *factory=(GstRTSPMediaFactory*) this->factory;
gst_rtsp_media_factory_set_launch (factory,"( rtpH264pay name=pay0 pt=96 rtpvorbispay
name=pay1pt=97 )");
g_signal_connect (factory, "media-configure", (GCallback) mediaConfigure,NULL);
```

```
gst_rtsp_mount_points_add_factory (mounts, "/primicia" , factory);  
g_object_unref (mounts);  
}
```