

Universidad de las Ciencias Informáticas

Facultad 6



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

Título: Componente de transmisión de noticias del Sistema de
gestión y transmisión de contenido audiovisual.

Autor: Katia Roselló Díaz

Tutor: Ing. Abel Díaz Berenguer

Co-Tutor: Ing. Dayami Chávez Ayala

La Habana, 2012

Pensamiento.

"Quien de verdad sabe de qué habla, no encuentra razones para levantar la voz."

Leonardo DaVinci

"Intenta no volverte un hombre de éxito, sino volverte un hombre de valor."

Albert Einstein

Declaración de Autoría.

Declaro que soy el único autor de este trabajo y autorizo al departamento de Señales Digitales de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio. Para que así conste firmo la presente a los 28 días del mes de Junio del año 2012.

Autor: Katia Roselló Díaz

Tutor: Abel Díaz Berenguer

Firma del Autor

Firma del Tutor

Datos de Contacto.

Tutor(a): Nombre y apellidos: Abel Díaz Berenguer.

Sexo: M

Institución: Universidad de las Ciencias Informáticas.

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

Municipio: Boyeros.

Provincia: Ciudad de La Habana.

Correo electrónico: aberenguer@uci.cu.

Teléfono del trabajo:-

Teléfono particular:-

Cargo del trabajador: Profesor

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

Año de graduación: 2009.

Institución donde se graduó: Universidad de las Ciencias Informáticas.

Co-Tutor (a): Nombre y apellidos: Dayami Chávez Ayala.

Sexo: F

Institución: Universidad de las Ciencias Informáticas.

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

Municipio: Boyeros,

Provincia: Ciudad de La Habana.

Correo electrónico: dchavez@uci.cu.

Teléfono del trabajo: -

Teléfono particular: -

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

Año de graduación: 2010.

Institución donde se graduó: Universidad de las Ciencias Informáticas.

Agradecimientos.

A mi mamá y mi papá porque han sido las personas más importantes en mi vida, los que me han guiado y han estado a mi lado siempre que los he necesitado, gracias por todo su amor.

A mi abuela por su amor, por ser mi fuerza en los momentos de debilidad.

A mi hermano por siempre darme ánimo y apoyarme en los momentos difíciles.

A mis tías Ana, Miriam y Magali con las que he podido contar siempre y que me han tratado como una hija.

A mis tíos que siempre me han dado fuerza para seguir adelante.

A mis primas Tania, Yudy, Alianni y Meizel, porque más que mis primas son mis hermanas.

A mis primos por siempre estar ahí cuando les he pedido ayuda.

A Linet, Leanet y Anniel porque a pesar de sus pocos años de vida me han inspirado y me han dado fuerzas para superarme ante las dificultades.

A mis amigas de estos 5 años de carrera que me han aguantado todas mis locuras Indi y Maide a ustedes mil gracias por no dejarme sola cuando más lo necesitaba.

A Dainelis, Tailén, Miriam, Yuncy, Eniley, la chuchi, Yuri, Yanary, Saidy, Lisbet gracias por tener tanta paciencia conmigo y por estar siempre ahí para mí.

A Dainoui, Dorgis, Alejandro, Yoandry y Mauricio gracias por su apoyo durante estos 5 años de carrera.

A los integrantes del núcleo familiar del 110-101, gracias su apoyo.

A mi novio por aguantarme estos meses tan duros y de tanto estrés para los dos, por darme siempre su apoyo y no dejarme sola.

A mis tutores Dayami y Abel por el apoyo que me dieron a lo largo de este curso.

A todos los que a lo largo de estos 5 años me han ayudado y apoyado, mis compañeros de aula, los del proyecto, a todos muchas gracias.

Dedicatoria.

A mis padres por todo su amor, porque me han apoyado en todos los momentos de mi vida respetando mis decisiones.

A mi querida abuela por haberme inspirado siempre a luchar por las cosas que quiero y a nunca darme por vencida.

A toda mi familia porque siempre me han demostrado que puedo contar con ellos en las buenas y las malas.

Resumen.

La Universidad de las Ciencias Informáticas contribuye al desarrollo de software para informatizar el país, para ello ha creado una serie de herramientas que posibilitan el procesamiento de las señales digitales como el Sistema de gestión y transmisión de contenido audiovisual que surge a partir de la fusión de varios productos existentes. Dos de estos productos tienen como elemento común un subsistema de transmisión, uno dedicado a las noticias y otro a los canales virtuales, surgiendo la necesidad de hacer posibles ambas transmisiones desde un mismo subsistema. El objetivo propuesto fue alcanzado mediante la implementación de un componente que permite la transmisión de noticias a través de un canal virtual. El presente trabajo está estructurado en cuatro capítulos que recogen las etapas de desarrollo del sistema, las cuales van desde la fundamentación teórica de la investigación, a través de la propuesta y el modelado de la solución hasta llegar a la implementación y validación de la aplicación. Se obtiene un componente que realiza el procesamiento de las noticias gestionadas por la plataforma PRIMICIA de forma tal que pueden ser transmitidas y gestionadas desde el subsistema de transmisión del Sistema de gestión y transmisión de contenido audiovisual.

Palabras Claves: canal virtual, componente, transmisión.

Índice.

Introducción.	1
Capítulo 1. Fundamentos teóricos de la investigación.	5
1.1 Conceptos asociados al dominio del problema.	5
1.2 Objeto de estudio.	6
1.2.1 Descripción general del objeto de estudio.	6
1.2.3 Situación problemática.	8
1.3 Análisis de soluciones existentes.	9
X Frame TV.	9
VIDEOMA:	11
Estructure:	11
1.4 Conclusiones parciales.	13
Capítulo 2: Herramientas y tecnologías a utilizar.	14
2.1 Metodología de desarrollo.	14
2.1.1 Proceso Unificado de Desarrollo de Software (RUP).	15
2.2 Lenguaje Unificado de Modelado 2.0 (UML).	16
2.3 Herramientas CASE.	16
2.3.1 Visual Paradigm 8.0.	17
2.4.1 C++.	18
2.5 Marco de trabajo.	19
2.5.1 Qt 4.7.	19
2.6 Entorno de desarrollo integrado.	19
2.6.1 QT Creator 2.0.1.	19
2.7 Sistema gestor de base de datos.	20
2.7.1 PostgreSQL 8.4.	20
2.8.1 Open Source Computer Vision library.	21

2.8.2 GStreamer.	21
2.8.3 FFMPEG 4.0.6.	22
2.8.4 Mencoder 2.1.0.	22
2.9 Conclusiones parciales.	23
Capítulo 3: Presentación de la solución propuesta.	24
3.1 Modelo de dominio.	24
3.2 Requisitos.	25
3.2.1 Requisitos funcionales.	25
3.2.2 Requisitos no funcionales.	26
3.3 Descripción del sistema propuesto.	27
3.3.1 Descripción de los actores.	27
3.3.2 Diagrama de casos de uso del sistema.	28
3.3.3 Descripción de los casos de uso del sistema.	28
3.4 Diagramas de clases del análisis.	30
3.5 Conclusiones parciales.	31
Capítulo 4: Construcción de la solución propuesta.	32
4.1 Arquitectura de software.	32
4.2 Principios del diseño.	34
4.3 Patrones de diseño.	35
4.4 Diagrama de clases del diseño.	38
4.5 Diagrama de despliegue.	39
4.6 Modelo de implementación.	40
4.6.1 Diagrama de componentes.	40
4.7 Prueba de software.	41
4.7.1 Prueba de caja blanca.	41
4.8 Conclusiones parciales.	49
Conclusiones generales.	50

Recomendaciones.....	51
Bibliografía referenciada.....	52
Bibliografía consultada.....	55
Anexos.....	59

Índice de Tablas.

Tabla 1 Descripción de actores.	27
Tabla 2 Descripción de caso de uso Conformar Video.....	29
Tabla 3 Descripción de caso de uso Conformar Planificación.....	30

Índice de Figuras.

Figura 1 Diagrama modelo de dominio.	25
Figura 2 Diagrama de casos de uso del sistema.	28
Figura 3 Diagrama de clases del análisis del caso de uso Conformar Video.....	30
Figura 4 Diagrama de clases del análisis del caso de uso Conformar Planificación.....	31
Figura 5 Arquitectura en 3 capas.	34
Figura 6 Ejemplo de patrón Experto.	36
Figura 7 Ejemplo de patrón Creador.	37
Figura 8 Ejemplo de patrón Observador.....	38
Figura 9 Diagrama de clases del diseño del sistema.	39
Figura 10 Diagrama de despliegue.....	40
Figura 12 Diagrama de componentes del sistema.....	41

Introducción.

Con el surgimiento de la informática y su aplicación en numerosas áreas del conocimiento y actividad humana, la llamada sociedad de la información vive en una acelerada transformación impulsada para crear y comunicar mediante tecnologías digitales. Los medios de comunicación juegan un papel determinante en el desarrollo del mundo moderno, permitiendo una constante renovación de las necesidades vitales del hombre.

Durante las diferentes etapas por las que ha transcurrido la humanidad, diversas han sido las formas empleadas para establecer comunicación: los signos, sonidos, caracteres alfabéticos, ideogramas, gestos, señales y dibujos simbólicos son las más primitivas. Con el paso del tiempo se hizo indispensable renovar los mecanismos empleados para la comunicación por medios eficientes que permitieran acortar la distancia y facilitar la comunicación entre millones de personas al mismo tiempo.

El surgimiento de las nuevas Tecnologías de la Informática y las Comunicaciones (TIC) y su constante desarrollo, hace que los rápidos cambios en la tecnología y el considerable aumento de conocimientos creen en la sociedad una constante perspectiva de actualización. En los últimos años el uso de las TIC, que engloban a la prensa, la radio, la televisión, el cine y la red mundial¹ se ha incrementado. En especial cabe destacar el explosivo desarrollo de la televisión, medio que juega un papel primordial en la difusión de la información.

La televisión surge gracias a la conjunción de una serie de fenómenos e investigaciones simultáneas, aunque desarrolladas de forma aislada (1). Los procesos asociados a su transmisión en la actualidad se realizan de múltiples formas, y sobre todo en los países desarrollados el negocio que existe alrededor del medio es muy lucrativo. La unión del sonido e imagen brinda la posibilidad de conocer realidades remotas, contemplar acontecimientos en tiempo real, disfrutar de la integración de diferentes artes como la literatura, música, arquitectura, pintura, fotografía entre otras.

En Cuba las transmisiones televisivas comenzaron a inicios de los años 50, convirtiéndose en unos de los primeros países de América Latina en contar con este medio y el segundo después de Estados Unidos en realizar las transmisiones en colores (1). Después del triunfo de la revolución, Cuba cambia el enfoque de sus transmisiones televisivas, con el objetivo de lograr una cultura general integral y fortalecer la preparación ideológica del pueblo.

¹ Es un conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial.

En la actualidad el uso de las TIC y la televisión con un fin educativo constituyen un avance considerable para lograr el uso masivo de las mismas, como herramienta poderosa para lograr el desarrollo. Es por ello, que estos tiempos suponen un importante cambio para la televisión en Cuba, pues han de hacerse los preparativos para la inminente llegada de la televisión digital lo cual podría lograrse probablemente para el año 2020 (2). Implantándose como premisa, crear nuevos proyectos que permitan una mayor interactividad con las nuevas tecnologías, convirtiendo al hombre cubano en un sujeto activo del uso de los medios digitales.

La Universidad de las Ciencias Informáticas (UCI), como motor impulsor para el desarrollo de productos informáticos con el propósito de informatizar el país, se ha dado a la tarea de crear herramientas que posibiliten el procesamiento de las señales digitales y la Geoinformática. De esta forma contribuye con la formación de personal calificado y que a su vez respondan a las necesidades de progreso científico-técnico de Cuba (3).

La creación del departamento de Señales Digitales intenta agrupar las investigaciones y productos de software que se desarrollan en aras de alcanzar el procesamiento de las señales digitales. Entre sus productos se puede mencionar el Sistema de gestión y transmisión de contenidos audiovisuales (SIAV), el cual tuvo surgimiento a raíz de la fusión de diferentes productos existentes como: la plataforma VideoWeb, la plataforma de televisión informativa PRIMICIA y la plataforma de Transmisión Abierta para Radio y Televisión (PTARTV).

El sistema provee facilidades de personalización en dependencia de los requisitos que planteen los clientes ya que cuenta con varios subsistemas que pueden desplegarse de forma independiente o integrada en un solo sistema. Al realizar la integración se consolidó el trabajo realizado en estos proyectos, conformando un producto más completo.

PRIMICIA es una plataforma que admite la administración y transmisión de noticias y PTARTV una solución capaz de automatizar procesos que se ejecutan en una entidad dedicada a la gestión, procesamiento y transmisión de contenidos multimedia; cada una de ellas por separado cuenta con un subsistema de transmisión. Con la idea de tener un sistema que pueda desplegarse de forma independiente se hace necesario que se pueda hacer la transmisión de noticias y de canales virtuales desde el mismo subsistema.

De ahí que se plantee el siguiente problema de investigación: ¿Cómo aumentar las formas de distribución de noticias del Sistema de gestión y transmisión de contenidos audiovisuales? Para dar solución al mismo se establece como objetivo general: Desarrollar un componente para la transmisión

de noticias a través de un canal virtual. De ahí que se pueda plantear como objeto de estudio: Los procesos de transmisión de señales televisivas digitales. Enmarcado en el campo de acción: Procesos de transmisión de noticias a través de canales virtuales. A partir de lo planteado se defiende que el desarrollo de un componente para la transmisión de noticias a través de un canal virtual aumentará las modalidades de distribución de noticias del Sistema de gestión y transmisión de contenidos audiovisuales.

Para lograr el objetivo general propuesto y la resolución del problema planteado se establecen las siguientes tareas de investigación.

- Realizar estudio del arte sobre la evolución histórico-lógica de los procedimientos y técnicas para la transmisión de señales televisivas.
- Caracterizar las técnicas y procedimientos de transmisión de señales televisivas en el Sistema de gestión y transmisión de contenidos audiovisuales.
- Determinar las tendencias y tecnologías actuales más factibles para el desarrollo de la investigación.
- Desarrollar artefactos y documentación según la metodología de desarrollo seleccionada.
- Implementar las funcionalidades requeridas para la integración del canal informativo a las transmisiones televisivas en el Sistema de gestión y transmisión de contenidos audiovisuales.
- Validar la propuesta de solución.

Para obtener conocimientos necesarios que puedan hacer posible el cumplimiento del objetivo general, se realiza una investigación en la que se utilizan los siguientes métodos científicos:

Métodos Empíricos:

Entrevista: Se utiliza para obtener información sobre el funcionamiento actual de los procesos de transmisión en PRIMICIA. Tomando como personal a entrevistar profesores y estudiantes que trabajan en el proyecto, específicamente con este subsistema. Para ello se toma una población de 6 individuos, de los cuales se toma como muestra 3 para ser entrevistados. Se realiza una técnica de muestreo no probabilístico, específicamente muestreo intencional. (Ver Anexo 1)

Métodos Teóricos:

Analítico-Sintético: Con este método se analizan los procesos de transmisión de contenidos audiovisuales en las diferentes plataformas que forman parte de la integración que componen el SIAV. Se emplea también para buscar puntos de convergencia que puedan existir entre soluciones ya existentes que permitan un aprovechamiento de conceptos que puedan resultar relevantes en la investigación.

Modelación: Durante la investigación se utilizan los modelos de implementación, diagramas de componentes y modelo de despliegue para crear abstracciones que permitan comprender el funcionamiento del sistema.

Histórico-Lógico: Permite valorar el desarrollo, evolución y tendencias actuales de los procedimientos y técnicas para la gestión de la transmisión de contenido audiovisual.

Capítulo 1. Fundamentos teóricos de la investigación.

En este capítulo se abordan los conceptos fundamentales que servirán de soporte teórico para el desarrollo de la investigación. Caracterizando el objeto de estudio en el que se describen los procedimientos y técnicas para la transmisión de señales televisivas. Se realiza un análisis del dominio del problema y la situación problemática, por lo que se hace necesario caracterizar las técnicas y procedimientos de transmisión en las plataformas PRIMICIA y PTARTV.

1.1 Conceptos asociados al dominio del problema.

Transmisión:

La transmisión es considerada como acción y efecto de transmitir que dicho propiamente de una emisora de radio o de televisión puede ser difundir noticias, programas de música y espectáculos entre otras (4). Se puede definir también como un intercambio o el envío de información en formato analógico o digital. Se dice que la televisión y la radio transmiten sus programas, ya que estos llegan al público a través de antenas, cables y otros dispositivos (5). Resumiendo en el contexto de la presente investigación se considera como transmisión la acción de enviar información en formato digital.

Señal:

La señal es una variación de una corriente eléctrica u otra magnitud que se utiliza para transmitir información (4). Es definido también como una onda electromagnética que permite transmitir información a un circuito electrónico, este es el caso utilizado en la señal de televisión y la señal de radio (6). De forma general se puede definir como un valor que proporciona cierta información y que puede ser interpretada o utilizada en función de un objetivo determinado.

Video Digital:

Un video digital es cualquier archivo de ordenador que contenga una secuencia de video (con o sin audio) (7). El video digital se representa como una secuencia de imágenes conocidos también como

fotogramas, donde además de los parámetros de resolución espacial (número de píxeles²) y profundidad, hay que considerar un número adecuado de imágenes por segundo que permitan crear la ilusión de movimiento (8). Se puede decir entonces que un video digital es un archivo que se encuentra almacenado en un dispositivo (CD, DVD y disco duro entre otros) compuesto por una secuencia de imágenes por segundo que dan la sensación de movimiento.

1.2 Objeto de estudio.

Para la presente investigación se ha definido como objeto de estudio: “*los procesos de transmisión de señales televisivas digitales*”

1.2.1 Descripción general del objeto de estudio.

En los últimos años el hombre ha utilizado la televisión para promocionar y transmitir servicios e información, con el objetivo de satisfacer sus necesidades desde la perspectiva de espectador y consumidor. La transmisión de una señal analógica o digital se puede realizar a través del aire (ondas electromagnéticas) o mediante una línea de transmisión física (transmisión por cable) (9).

La tradicional consiste en operar un conmutador de señal, al cual llegan señales de video de distintas fuentes como son: grabadoras de cintas de video, señales directas del estudio o de otras televisoras. Para esto existe una persona encargada de realizar esta operación, guiándose por una escaleta que no es más que un esqueleto o esquema del video que ayuda a organizar el plan de grabación y facilita la escritura del pre-guion y del guion definitivo, permitiendo así especificar el orden en que se mostrará cada espacio y el tiempo de duración. Los avances tecnológicos de los últimos años tanto a nivel del hardware como del software han dado la posibilidad de crear sistemas que automatizan el proceso de transmisión. Esto brinda la posibilidad de gestionar la escaleta lo que permite tener un mayor control de los procesos asociados a la transmisión.

La presencia de la tecnología en el campo de la comunicación televisiva, la posibilidad de realizar la transmisión de señales digitales y la utilización de los satélites de difusión directa, han hecho posible el amplio uso de la televisión digital (DTV por sus siglas en inglés) (9). La DTV es la emisión de información binaria para ser mostrada en un receptor capaz de interpretar señales digitales televisivas, garantizando una gran eficacia en la transmisión, una mejor calidad audiovisual y la posibilidad de prestar servicios de información adicional basados en la interactividad (10).

² Es la menor unidad homogénea en color que forma parte de una imagen digital.

Además la televisión digital permite un enorme número de ventajas diferenciadas y funciones adicionales respecto a la difusión convencional de televisión tales como: (10)

- Incrementar notablemente el número de programas, servicios y señales actualmente disponibles.
- Mejorar la calidad de las imágenes y del sonido de las transmisiones y de las recepciones televisivas.
- Establecer servicios personalizados e interactivos (relación empresa televisiva-audiencia) de radiodifusión y telecomunicaciones.
- Facilitar la convergencia entre el sector audiovisual, las telecomunicaciones y la informática.

La DTV ofrece la posibilidad de aplicar distintos niveles de interactividad. El televidente puede consultar el parte del tiempo, publicaciones periódicas, informaciones diversas, datos sobre un juego de beisbol, o sobre los actores de un dramatizado. También permite al usuario interactuar en tiempo real en programas de participación e incluso, seleccionar el ángulo en que desea disfrutar de un evento deportivo. Estas opciones no estarán disponibles con solo aplicar la televisión digital, sino que será un valor agregado por la industria (2).

Existen tres formas de difusión de televisión digital: satélite, cable y televisión digital terrestre (TDT). El sistema de televisión por satélite proporciona una programación de ámbito nacional o internacional. La televisión por cable proporciona una programación que puede combinar la nacional con la local, aunque su cobertura se adapta mejor a núcleos densos de población (2).

En el caso de la televisión digital terrestre es una técnica de difusión de las señales televisivas gracias a la cual la imagen, el sonido y los contenidos interactivos se transforman en información digital, es decir, que en lugar de ser enviados en forma de complejas ondas, que requieren grandes anchos de banda, lo hacen como "ceros y unos". En este caso de la televisión digital terrestre la recepción se realiza a través de la antena de televisión terrenal convencional instalada en las instalaciones que cuenten con el servicio (10).

La diferencia entre televisión digital terrestre y televisión digital por satélite o por cable se puede encontrar en que todos los contenidos de la televisión digital terrestre se transmiten exactamente de la misma forma en que lo hace la televisión analógica convencional, es decir, mediante ondas electromagnéticas terrestres, y es recibida a través de las antenas convencionales, a diferencia de la televisión digital vía satélite o por cable, que utilizan antenas parabólicas o cables (10).

Las redes de televisión digital terrestre, como las redes de difusión por satélite, permiten fácilmente difundir programas de televisión y datos asociados a los programas o de otro tipo a un gran número de usuarios, con las ventajas añadidas de flexibilidad para cubrir zonas de mayor o menor tamaño según las necesidades de cobertura (nacional, autonómica o local) y de usar las antenas existentes de la televisión digital analógica (10).

1.2.2 Descripción actual del dominio del problema.

En el departamento de Señales Digitales perteneciente al centro GEYSED se encuentra la solución Sistema de gestión y transmisión de contenidos audiovisuales (SIAV), la cual surge con la unión de varias soluciones existentes en el departamento. De forma general el sistema constituye una solución compuesta por varios subsistemas que pueden desplegarse de manera independientemente o integrada en un mismo sistema. Entre sus principales funcionalidades se encuentran la publicación y transmisión de contenido multimedia bajo demanda en la web, así como la programación de transmisiones de contenido multimedia, realizar las transmisiones previamente programadas y monitorizar las mismas.

1.2.3 Situación problemática.

En la actualidad existen numerosos sistemas de producción de noticias así como de gestión y administración de procesos asociados a televisoras. Todo esto originado por la creciente competencia entre productos de alta tecnología dedicados a la televisión y sentados sobre las bases de una tecnología generalmente propietaria. Cuba está limitada al uso de las TIC dedicadas a la gestión y transmisión de contenidos audiovisuales que ofrece el mundo, debido a la gran suma de dinero que conlleva la obtención y mantenimiento de las mismas, lo que representa un obstáculo para el desarrollo de los medios televisivos en el país. Por tal razón, se han dirigido esfuerzos al desarrollo de herramientas y sistemas, basados en los principios de independencia tecnológica y con el objetivo de alcanzar la gestión de sus propios medios de comunicación.

En este proceso, la UCI tiene un lugar significativo ya que varios de sus proyectos de innovación investigación y desarrollo están dirigidos a implementar herramientas para proveer al país del soporte digital que necesita. Entre los avances obtenidos en la gestión y administración de procesos asociados a televisoras, es válido caracterizar la Plataforma de Transmisión Abierta para Radio y Televisión (PTARTV) y la Plataforma de televisión informativa (PRIMICIA):

PTARTV.

Constituye una solución informática capaz de automatizar procesos que se ejecutan en una entidad dedicada a la gestión, procesamiento y transmisión de contenidos multimedia. Está estructurado en subsistemas que se relacionan entre sí y actúan como un todo para brindar un resultado final eficiente y acorde a las necesidades de los usuarios. Entre los subsistemas con que cuenta el sistema se encuentran el de Monitorización, Programación y Transmisión.

El subsistema de transmisión cuenta con funcionalidades como: gestionar canal, realizar transmisión la cual puede ser de manera manual o automática, listar programación canal, listar canales, mostrar datos canal, mostrar datos de las medias y visualizar la transmisión de un canal.

PRIMICIA.

Plataforma que admite la administración y transmisión de noticias. Se ha venido desarrollando desde el año 2005 por la UCI, constituyendo una solución capaz de proveer un canal de televisión para la transmisión automática y constante de informaciones en distintos formatos. De forma general su uso sería especialmente útil en Televisoras, Agencias de noticias, Sedes ministeriales o empresariales, Terminales de transporte, Universidades, Centros de convenciones, Hoteles, entre otros.

La integración de PRIMICIA, PTARTV y VideoWeb se realizó con el objetivo de tener un producto que pudiera ser comercializado, con todos sus subsistemas o con parte de ellos atendiendo a las necesidades del cliente. El hecho de que PRIMICIA contara con un subsistema de transmisión dedicado a la difusión de noticias y PTARTV con uno de transmisión de canales virtuales trajo consigo la necesidad de que se pudiera realizar la transmisión de ambas desde un mismo subsistema. Para ello se cuenta con el subsistema de transmisión del SIAV que brinda un conjunto de funcionalidades pero no permite la transmisión de noticias.

1.3 Análisis de soluciones existentes.

X Frame TV:

Es el Sistema Digital de Gestión Integral para Televisión. Es la aplicación principal del sistema X Frame, de la empresa ASPA. Es un sistema integral multimedia que engloba aplicaciones y herramientas para llevar a cabo toda la actividad de cualquier emisora de radio o televisión. Los recursos del sistema (audio, video, noticias) son comunes y están disponibles para su utilización en radio o televisión (11).

Para el desarrollo del sistema se siguieron criterios de funcionalidad, teniendo en cuenta, sobre todo, el usuario, la facilidad de manejo y la aportación de un amplio abanico de opciones y utilidades (11). Cuenta con diferentes módulos como:

Grabación/Edición:

- Distintos modos de grabación.
- Distintas formas de salvar grabaciones y cortes.

Emisión:

- Incrustación de logos y titulaciones de forma automática y manual.
- Fichas de ajuste automático de tiempo.

Parrilla de programación:

- Programación de días especiales, sin necesidad de crear una nueva estructura de programación.
- Refrescado automático de la información visualizada.

Noticias:

- Editor de noticias, almacenando versiones de una noticia modificada.
- Inserción de noticias en la parrilla de programación (11).

Algunas de sus principales características son: (11)

- Trabaja con múltiples formatos de codificación de video (DV50, DV25, DVSD) de forma simultánea en la misma lista.
- Trabaja con múltiples formatos de codificación de audio (MP3, MP2, WAV) de forma simultánea.
- Interface de usuario (pantallas) configurable y definible por el usuario. (Colores, tamaños, números de listas, tipo de letra).
- Buzones de video con subcarpetas (estructura de árbol).
- Posibilidad de definir nuevos campos en la base de datos.
- Aplicación diseñada bajo el modelo cliente/servidor:
 1. Sistema basado en comunicaciones.
 2. Recuperación automática ante cualquier eventualidad.
 3. Control y supervisión de puestos locales y remotos.
 4. Servidor y cliente en aplicaciones distintas.

VIDEOMA:

Es una solución para la gestión digital de video, audio e imagen. Cuenta con un sistema que ofrece una amplia gama de módulos que pueden ser adaptadas a las necesidades de los clientes. Posee la capacidad de capturar y transcodificar cualquier tipo de formato. Cuenta con varios sistemas tales como: (12)

- **Videoma Broadcast Monitor:** sistema de monitorización de contenido procedente de radio y televisión.
- **Videoma Archivo:** sistema de archivo para la gestión profesional de videotecas, fonotecas y fototecas.
- **Videoma LOCAL TV:** se integra de manera transparente con otras tecnologías utilizadas en la edición, producción y postproducción. Permite la inserción de contenido digital automatizada y sin necesidad de supervisión. Conversión y gestión automática de formatos de video.
- **Videoma Broadcast Monitor Lite:** es un sistema de análisis y seguimiento de TV que realiza la grabación, catalogación y búsqueda automática de contenidos de TV.

De manera general la solución cuenta con una serie de ventajas tales como: (12)

- Grabación, monitorización y visualización de múltiples canales.
- Indexación automática por subtítulos.
- Búsquedas por palabra clave.
- Envío de alertas por emails y creación de clips automáticos cuando el sistema detecta información objeto de análisis.
- Importación y exportación de información por XML.
- Catalogación personalizada por medio de campos de metadatos.
- Plataforma disponible en español e inglés.

Estructure:

Es una empresa de desarrollo de sistemas para la producción y gestión audiovisual que proporciona soluciones tecnológicas avanzadas para la explotación de activos audiovisuales. Los procesos productivos están integrados y se conectan entre sí aprovechando al máximo el trabajo en grupo: desde las áreas técnicas mediante la digitalización y control de emisión, hasta el visionado, redacción, edición, documentación, Internet o control del archivo, ofreciendo un alto nivel de funcionalidad (13).

Cuenta con varias soluciones tales como:

- **ELOCALTV:** es una solución orientada a televisoras pequeñas y canales temáticos. Incluye una nueva tecnología de video en 2D y 3D en tiempo real, además de utilizar las principales herramientas estándar de diseño para la creación y composición gráfica.
- **EVIDEOTECA:** permite la consulta y visionado de contenidos de forma simultánea para todos los puestos conectados, en tiempo real, y en diferentes configuraciones de calidad. Incluso pueden conectarse salas de visionado con pantallas de gran formato.
- **EVIDEO-OBSERVER:** es un sistema para la grabación, registro y seguimiento de medios de comunicación. Permite capturar de forma simultánea, desatendida y programada, aquellos espacios seleccionados previamente. Las grabaciones se irán almacenando con las clasificaciones de canales otorgadas, permitiendo el acceso y visionado, por todos los usuarios con acceso (13).

Las características principales que definen las aplicaciones de Estructure son las siguientes: (13)

Sencillo e intuitivo:

- Agradable interfaz gráfica.
- Herramientas de trabajo visibles en todos los módulos.

Rápido y potente:

- Todos los procesos se realizan en tiempo real.
- Optimización de redes con altas tasas de transferencia.
- Sincronismo de Códigos de Tiempos en lectura y escritura.

Flexible y compatible:

- Soporte de múltiples formatos de entrada y salida: H.264, DV, DVCAM, DVCPPro, MPEG-4, FLV.
- Compatibilidad con sistemas de terceros.
- Desarrollos y modificaciones a medida (13).

Al realizar el estudio de los sistemas de gestión de contenido audiovisual se han podido precisar las características fundamentales presentes en sistema de este tipo, tales como:

- Trabajo con múltiples formatos de video como MPEG-2, AVI, HD, HDV.

- Están concebidas desde el punto de vista de los distintos usuarios del sistema, facilitando las tareas mediante una sencilla y agradable interfaz de trabajo.
- Funcionamiento independiente o integrado a un sistema.
- Trabajo con base de datos relacionales.

Además de un conjunto de elementos que demuestran que estas soluciones no serían factibles tales como:

- Las tecnologías que utilizan son desarrolladas y costosas. Como el Protocolo de Control de Disco Óptico (VDCP).
- Utilizan software propietario como el Adobe Premiere, Vegas, Final Cut entre otros.
- Necesitan redes de banda ancha.
- Necesitan un hardware específico.
- Algunos son aplicables solo a lugares con determinadas características este es el caso ELOCALTV que permite la configuración de hasta 15 estaciones de trabajo.

1.4 Conclusiones parciales.

El análisis de los principales conceptos relacionados con el dominio del problema, permitió obtener un mayor conocimiento de los elementos teóricos que sustentan los objetivos del trabajo. La idea de desarrollar un componente que permita transmitir las noticias gestionadas por PRIMICIA a través del subsistema de transmisión de SIAV, constituye una solución factible al problema planteado.

Capítulo 2: Herramientas y tecnologías a utilizar.

En el presente capítulo se realizará un análisis de las tecnologías actuales más factibles para el desarrollo del componente. Definiendo para esto la metodología de desarrollo, el lenguaje de modelado y la herramienta CASE que guiará el proceso de desarrollo del software. Se seleccionará el lenguaje de programación, marco de trabajo y entorno integrado de desarrollo que serán usados para la implementación de la solución.

2.1 Metodología de desarrollo.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y documentación para el desarrollo de productos software. En esta se indican paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando las personas que deben participar en el desarrollo y qué papel deben de tener en el mismo. Se puntualiza además la información necesaria para comenzar una actividad y la que se debe obtener como resultado (14).

Existen metodologías ágiles y tradicionales. Las ágiles se centran en la capacidad de las personas miembros del equipo de desarrollo. Son mejores en equipos de desarrollo pequeños y generalmente necesitan que el cliente forme parte del equipo de trabajo. Está orientada fundamentalmente al desarrollo de soluciones pequeñas.

Las metodologías de desarrollo tradicionales conceden un gran peso a la planificación y descripción del sistema que se intenta realizar. Es por lo que al comenzar la implementación se cuenta con una definición bastante clara de cada elemento que se desea desarrollar. Este tipo de metodología se aplica más a proyectos de gran envergadura con grandes equipos de desarrollo. Es recomendable cuando se hace necesario que la documentación generada en el proceso de desarrollo tenga un nivel de detalle elevado para posterior entendimiento.

Se decide utilizar como metodología de desarrollo, el Proceso unificado de desarrollo de software pues la cantidad de documentación y detalle que se alcanza con la realización de los diagramas de UML proporciona a los desarrolladores un alto nivel de entendimiento, permite además el modelado orientado a objetos. El equipo de desarrollo tiene experiencia en el trabajo con esta metodología. Además esta propone el proceso de desarrollo en iteraciones lo que posibilita la corrección de errores y las mejoras del software a medida que se avanza en el desarrollo del mismo.

2.1.1 Proceso Unificado de Desarrollo de Software (RUP).

El proceso de desarrollo de software es un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software, está basado en componentes (15).

- **Dirigido por casos de uso:** quiere decir que el proceso de desarrollo sigue un hilo que avanza a través de una serie de flujos de trabajo que parten de los casos de uso. Los casos de uso se especifican, se diseñan, siendo los casos de uso finales la fuente a partir de los cuales los ingenieros de prueba construyen sus casos de prueba (15). De forma general se puede concluir que los casos de uso son los que guían el proceso de diseño, prueba e implementación, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.
- **Centrado en la arquitectura:** la arquitectura incluye los aspectos significativos de un sistema, muestra una visión común en la que el cliente y equipo de desarrollo deben estar de acuerdo, se describen algunos elementos que permiten tener una perspectiva del sistema completo. El proceso se desarrolla mediante iteraciones, comenzando por los casos de usos relevantes desde el punto de vista de la arquitectura (16).
- **Iterativo e incremental:** para hacer más práctico se divide el trabajo en partes más pequeñas o miniproyectos. Cada uno de estos miniproyectos es una iteración que resulta en un incremento, estas iteraciones hacen referencia a pasos en el flujo de trabajo y los incrementos al crecimiento del producto (16).

Estos conceptos, los del desarrollo dirigido por casos de uso, centrado en arquitectura, iterativo e incremental, son de igual importancia. La arquitectura proporciona la estructura sobre la cual guiar las iteraciones mientras que los casos de uso definen los objetivos y dirigen el trabajo en cada iteración (15).

Cuenta con 4 fases en las que se pueden realizar diferentes iteraciones:

- **Conceptualización** (Concepción o Inicio): se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- **Elaboración:** se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la

comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.

- **Construcción:** se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene uno o varios entregables del producto que han pasado las pruebas. Se ponen estos entregables a consideración de un subconjunto de usuarios.
- **Transición:** el producto liberado ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores (15).

El proceso unificado utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema de software, de hecho UML es una parte esencial de este proceso.

2.2 Lenguaje Unificado de Modelado 2.0 (UML).

UML es un lenguaje de modelado visual que se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información generada en el desarrollo de software. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar (16).

El uso de este lenguaje en el modelado del componente va a permitir:

- **Visualizar:** permite expresar de una manera gráfica un sistema de forma que este se pueda entender.
- **Especificar:** especifica cuáles son las características que va a tener cierto sistema antes de ser construido.
- **Construir:** una vez que estén especificados los modelos se pueden construir los sistemas diseñados.
- **Documentar:** se pueden utilizar como documentación los elementos gráficos desarrollados y estos pueden servir para una futura versión (16).

Para realizar la modelación con este lenguaje se utilizan las herramientas CASE las cuales facilitan la planificación y gestión del proyecto informático.

2.3 Herramientas CASE.

CASE (Computer-Aided Software Engineering o Ingeniería de Software Asistida por Computadora). Están destinadas a realizar énfasis en la planificación, análisis y diseño lo cual tiene un fuerte impacto

y recompensa en la mejora de la calidad del producto obtenido y en el aumento de productividad (disminución de tiempos, costes y esfuerzos) en las actividades de desarrollo y mantenimiento (17).

El beneficio adicional obtenido por la utilización de un CASE actual (si se compara con la utilización de una metodología sin el uso de un CASE) se representa en los siguientes aspectos:

- Facilita la verificación y mantenimiento de la consistencia de la información del proyecto.
- Facilita el establecimiento de estándares en los procesos de desarrollo y documentación.
- Facilita el mantenimiento del sistema y las actualizaciones de su documentación.
- Facilita la aplicación de las técnicas de una metodología.
- Disponibilidad de funciones automatizadas tales como: obtención de prototipos, generación de código, generación de pantallas e informes, generación de diseños físicos de bases de datos, verificadores automáticos de consistencia.
- Facilita la aplicación de técnicas de reutilización y reingeniería.
- Facilita la planificación y gestión del proyecto informático (17).

2.3.1 Visual Paradigm 8.0.

Visual Paradigm es una herramienta CASE que propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño. El uso de Visual Paradigm permite: (18)

- Construcción de aplicaciones con buena calidad y a un menor coste.
- Dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.
- Hacer uso de un lenguaje común al equipo de desarrollo que facilita la comunicación y disponibilidad de diferentes versiones.
- Soportar que trabajen múltiples usuarios sobre el mismo proyecto. Brinda facilidades para la exportación e importación de componentes (18).

Teniendo en cuenta que Visual Paradigm:

- Es la herramienta con la que el equipo de desarrollo tiene mayor experiencia.
- Utiliza como lenguaje de modelado UML que es el escogido para realizar el modelado del componente.
- Cuenta con una amplia documentación que puede ser consultada.

Se decide utilizar como herramienta CASE para la modelación de la solución.

2.4 Lenguaje de programación.

Un lenguaje de programación es una secuencia de instrucciones que permite a una computadora procesar una información. Está constituido por un grupo de reglas gramaticales, semánticas y un grupo de símbolos que permiten la comunicación entre una persona y el ordenador (19).

Según la forma de ejecución se clasifican en:

- **Lenguajes de programación compilados:** son aquellos cuya función es traducir un programa escrito en un determinado lenguaje a un idioma que la computadora entienda.
- **Lenguajes de programación interpretados:** son aquellos que sólo realizan la traducción a medida que sea necesaria, típicamente, instrucción por instrucción, y normalmente no guardan el resultado de dicha traducción (19).

2.4.1 C++.

C++ es un lenguaje de programación, diseñado a mediados de los años 1980, por Bjarne Stroustrup, como extensión del lenguaje de programación C. El lenguaje C++ abarca tres paradigmas de la programación: la programación estructurada, la programación genérica y la programación orientada a objetos. Es un lenguaje de programación compilado que tiene como ventajas principales: (20)

Difusión: al ser uno de los lenguajes más empleados en la actualidad, posee un gran número de usuarios y existe una gran cantidad de libros, cursos, y páginas web dedicadas a él.

Versatilidad: C++ es un lenguaje de propósito general, por lo que se puede emplear para resolver cualquier tipo de problema.

Portabilidad: el lenguaje está estandarizado y un mismo código fuente se puede compilar en diversas plataformas.

Eficiencia: C++ es uno de los lenguajes más rápidos en cuanto a ejecución.

Herramientas: existe una gran cantidad de compiladores, depuradores, librerías, entre otros (20).

Los programas desarrollados en C++ son rápidos. El código es portable, es decir, podrá ejecutarse en diferentes máquinas y en cualquier sistema operativo. Existen diferentes marcos de trabajo y bibliotecas para el trabajo con imágenes y videos que se pueden vincular con este lenguaje lo cual es muy significativo para el desarrollo del componente (21), es por estas razones que se decide utilizar como lenguaje de programación.

2.5 Marco de trabajo.

Un marco de trabajo o framework (del inglés) es una estructura de soporte definida en la cual puede ser organizado y desarrollado otro proyecto de software. Puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software, para ayudar a desarrollar y unir los diferentes componentes de un proyecto de software (22).

2.5.1 Qt 4.7.

Qt es un marco de trabajo multiplataforma para el desarrollo de aplicaciones e interfaces de usuario. Utiliza el lenguaje de programación C++ de forma nativa. Cuenta con un gran número de bibliotecas que pueden ser empleadas para el desarrollo de software, tiene métodos para acceder a bases de datos mediante SQL, es uno de los más utilizados en la actualidad por lo que presenta una amplia bibliografía para su estudio. Contiene además módulos para el trabajo con protocolos de red. Es producido por la división de software Qt de Nokia, desarrollado bajo licencia LGPL y liberado como software libre y de código abierto (23).

Se decide hacer uso del marco de trabajo Qt pues permite la utilización del paradigma de programación orientada a objeto, usa como lenguaje de programación C++ de forma nativa, cuenta con una serie de herramientas y clases que permiten entre otras cosas el fácil acceso a bibliotecas para el procesamiento de imágenes, audio y video. Para tratamiento de imágenes en Qt, las dos clases más utilizadas son QImage y QPainter la primera permite trabajar con la imagen en crudo, es decir, acceder directamente a los píxeles y modificarlos, abrir y guardar en diferentes formatos, y la segunda permite crear composiciones de imágenes, escala, rotación, dibujo de formas básicas y pinturas.

2.6 Entorno de desarrollo integrado.

Un entorno de desarrollo integrado o IDE (Integrated Development Environment), es un programa informático que está compuesto por un conjunto de herramientas de programación. Consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica de usuario. Pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes (24).

2.6.1 QT Creator 2.0.1.

QT Creator es un IDE de desarrollo multiplataforma, permite crear aplicaciones de escritorio. Puede ser usado para desarrollar interfaces gráficas de usuario y también para el desarrollo de programas sin interfaz como herramientas de la consola y servidores. Es distribuido bajo los términos de GNU Lesser

General Public License (LGPL³). Tiene entre sus principales características un avanzado editor de código, provee soporte para edición de C++ y completamiento de código (25).

2.7 Sistema gestor de base de datos.

Un sistema gestor de bases de datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta (26).

- **Lenguaje de definición de datos (LDD O DDL):** se utiliza para especificar el esquema de la base de datos, las vistas de los usuarios y las estructuras de almacenamiento. Es el lenguaje que utilizan los diseñadores y los administradores de la base de datos.
- **Lenguaje de manipulación de datos (LMD o DML):** se utiliza para realizar operaciones de consulta y/o modificación de la base de datos. Es el lenguaje utilizado por los usuarios de las bases de datos.
- **Lenguaje de Consulta:** es un lenguaje informático usado para hacer consultas en bases de datos (27).

Algunos ejemplos de SGBD son Oracle, DB2, MS SQL Server, MySQL y PostgreSQL.

2.7.1 PostgreSQL 8.4.

PostgreSQL es un SGBD de código abierto que soporta casi todas las sintaxis SQL, incluyendo funcionalidades como consultas complejas, subconsultas, disparadores y el trabajo con grandes cantidades de datos. Es un gestor capaz de ajustarse al número de procesadores y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta. Cuenta con un alto soporte para casi todos los lenguajes de programación lo que incluye a C++ (27).

Se decide utilizar PostgreSQL como SGBD pues además de incluir un alto soporte para el lenguaje de programación seleccionado es el usado en el SIAV.

³ GNU Lesser General Public License es una licencia de software creada por la Free Software Foundation. Los contratos de licencia de la mayor parte del software están diseñados para jugar con su libertad de compartir y modificar dicho software.

2.8 Biblioteca.

Las Bibliotecas o library (en inglés) contienen datos y código que proporcionan servicios a programas independientes pasando a formar parte de estos. Son consideradas como conjunto de subprogramas utilizados para desarrollar software.

2.8.1 Open Source Computer Vision library.

Open Source Computer Vision library o OpenCV como se le conoce, es una biblioteca de código abierto que proporciona funciones para el procesamiento de imágenes. Permite crear aplicaciones poderosas en el dominio de la visión digital. Incluye funcionalidades como: realizar operaciones básicas en matrices, procesado de imágenes, análisis estructural, análisis de movimiento, reconstrucción 3D, calibración de la cámara, segmentación y reconocimiento de objetos (28).

Entre sus principales tipos de datos se encuentra el `IplImage` el cual es usado para la representación de imágenes en RGB y la función `CvMat` que es una de las más empleadas para operar con imágenes. El `cvCreateMat` permite configurar una estructura matricial de manera sencilla y el `CvSize` se utiliza para definir las dimensiones de un rectángulo en píxeles (28).

Para el trabajo con video se usa el `CvCreateVideoWriter` al cual se le especifican algunos elementos que conformará el video como la cantidad de cuadros por segundo y el códec de video. Con la función `CvWriteFrame` se pueden añadir los cuadros que van a ir conformando el mismo. La biblioteca cuenta además con una serie de funciones básicas para cargar y guardar imágenes este es el caso del `cvLoadImage` y el `CvSaveImage`. Estas imágenes pueden tener diferentes formatos como BMP, TIFF y JPEG (28).

2.8.2 GStreamer.

GStreamer es un framework multimedia, libre, multiplataforma escrito en el lenguaje de programación C, usando la biblioteca GObject. Permite crear aplicaciones audiovisuales, como de video, sonido y codificación. La función del núcleo de GStreamer es proveer un marco para complementos, flujo de datos y manejo/negociación de distintos tipos de medios. Provee una API para escribir aplicaciones. Basa su funcionamiento en complementos, los cuales pueden proveer de códecs o más funcionalidades (29).

2.8.3 FFMPEG 4.0.6.

FFmpeg constituye una solución multiplataforma para grabar, convertir y transmitir audio y video. Incluye varios códec de video y audio como MPEG-1, MPEG-2, H.264/MPEG-4 AVC, MP2, mp3, Vorbis, WMA entre otros. Puede además capturar y codificar en tiempo real desde una tarjeta de televisión. Está licenciado bajo la LGPL o GPL en función del tipo de opciones de configuración.

El paquete viene con tres programas:

- **ffmpeg**: es una herramienta en línea de comandos para convertir ficheros de video, flujos de red o la entrada de una tarjeta de TV a varios formatos de video.
- **ffserver**: es un servidor de flujo para todo lo que ffmpeg pueda usar como entrada (ficheros, flujos, entrada de la tarjeta de TV y cámara web).
- **ffplay**: es un reproductor de medios muy simples y portable que utiliza las librerías ffmpeg y la librería SDL (30).

2.8.4 Mencoder 2.1.0.

El Mencoder es un codificador de video libre que se encuentra liberado bajo licencia GPL y que se encuentra incluido en el reproductor mplayer. Permite convertir audio y video a los formatos que su reproductor sea capaz de reproducir (31).

Entre los códecs que soporta se evidencian:

Para audio: pcm, copy, mp3lame y lavc.

- pcm: similar a WAV, codifica la cadena de audio sin comprimirla.
- copy: copia la cadena de Audio sin volver a codificarla.
- mp3lame: codifica la cadena de audio en MP3 de 128 Kbps (Por defecto)
- lavc: codifica la cadena en múltiples formatos soportados por FFMPEG.

Para video: raw, copy, xvid, lavc, frameno, qtvideo y nuv.

- raw: video sin compresión.
- copy: copia la cadena de video sin volverla a codificar.
- xvid: codifica la cadena de video usando el códec Xvid.
- lavc: codifica la cadena de video en múltiples formatos soportados por FFMPEG, incluyendo WMV, Microsoft MPEG-4, DivX, MPEG.

2.9 Conclusiones parciales.

La metodología de desarrollo seleccionada permitirá guiar el proceso de construcción del componente, utilizando para ello UML como lenguaje de modelado y la herramienta CASE Visual Paradigm. Teniendo en cuenta que permite el uso del lenguaje de programación C++ y cuenta con clases y bibliotecas que permiten el procesamiento de audio y video se selecciona como marco de trabajo Qt y como entorno integrado de desarrollo Qt Creator.

Capítulo 3: Presentación de la solución propuesta.

En el presente capítulo se realiza un análisis de los principales conceptos asociados con el entorno del sistema. Se obtienen los artefactos asociados a la etapa de análisis según la metodología de desarrollo seleccionada. Se definen además los requisitos funcionales y no funcionales del sistema, así como el diagrama de casos de uso y la descripción de cada uno de ellos.

3.1 Modelo de dominio.

Modelo de dominio es un artefacto de la disciplina de análisis, construido con las reglas de UML durante la fase de concepción, en la tarea construcción del modelo de dominio, presentado como uno o más diagramas de clases conceptuales y que contiene, no conceptos propios de un sistema de software sino de la propia realidad física. Los modelos de dominio pueden utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema (32).

El modelo de dominio presenta los conceptos importantes relacionados con el mismo. Es una actividad clásica del desarrollo orientado a objetos que permite establecer cuáles son los objetos de interés en el dominio.

Algunos de los términos analizados pertenecientes al dominio son:

Pantallas: son los elementos por lo que se encuentra compuesta la noticia, estas pueden ser de diferentes tipos, texto, texto – imagen, imagen o video.

Noticia: está compuesta por pantallas que brindan información, son clasificadas en sesiones temáticas.

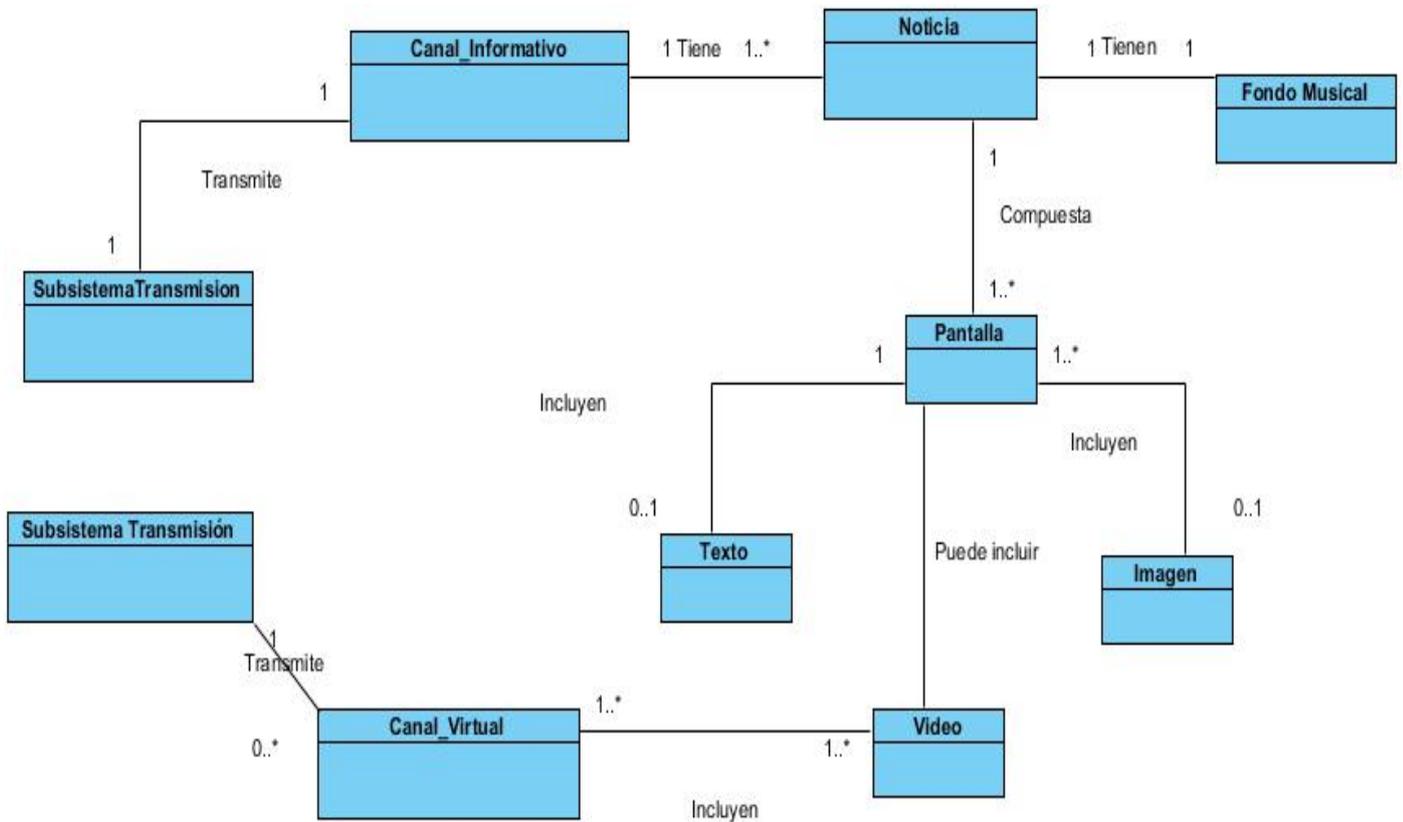


Figura 1 Diagrama modelo de dominio.

3.2 Requisitos.

La ingeniería de requisitos facilita el mecanismo apropiado para comprender lo que quiere el cliente, analizando necesidades, conformando su viabilidad, negociando una solución razonable, especificando la solución sin ambigüedad, validando la especificación y gestionando los requisitos para que se transformen en un sistema operacional (33).

Para el desarrollo del componente se hace necesario identificar los requisitos que se deben cumplir para satisfacer las necesidades del cliente. El tratamiento de requisitos es el proceso mediante el cual se especifican y validan los servicios que debe proporcionar el sistema, así como las restricciones sobre las que se deberá operar.

3.2.1 Requisitos funcionales.

RF 1: Conformar Video.

RF 2: Conformar Planificación.

3.2.2 Requisitos no funcionales.

Restricciones de software:

PC_Cliente.

Para el funcionamiento del sistema se deben instalar:

- El sistema operativo en el que se encuentre el componente debe ser GNU/Linux distribución UBUNTU 11.04.
- Mencoder 2:1.0.
- Ffmpeg 4:0.6.4.
- nfs-common 1:1.2.
- Libqxml-dev.
- libqt4-sql-psql.

Especificar en el fichero **fstab** que se encuentra ubicado dentro de **/etc** el url de la carpeta compartida y el lugar donde se va a montar.

PRIMICIA.

- libqxml-dev.

Repositorio de Medias.

- nfs-kernel-server.

Configurar el fichero **/etc/exports** para compartir la carpeta de almacenamiento y de publicación del servidor.

Servidor de Base de Datos

- PostgreSQL 8.4.

Requisitos de hardware:

Mínimos:

- Tarjeta de Red: 100 Mbps.
- Memoria RAM: 1GB.
- Procesador: Core 2 Duo.
- Disco Duro: 40 GB.

Recomendados:

- Tarjeta de Red: 100 Mbps o superior.
- Memoria RAM: 4 GB o superior.
- Procesador: Core i3 o superior.
- Disco Duro: 40 GB.

3.3 Descripción del sistema propuesto.

Después de realizado el modelo de dominio y el levantamiento de requisitos del sistema se describen los casos de uso. Para ello es necesario realizar una descripción representando de manera secuencial las acciones más significativas que realiza un actor en intercambio con el sistema.

3.3.1 Descripción de los actores.

El término significa el rol que algo o alguien juega cuando interactúa con el sistema. Un actor no es parte del sistema en desarrollo, es un agente externo que intercambia con el mismo para lograr un resultado esperado (34).

Actores	Descripción de los actores
Primicia	Representa el sistema Primicia, encargado de inicializar el proceso para la conformación del video asociado a la noticia.

Tabla 1 Descripción de actores.

3.3.2 Diagrama de casos de uso del sistema.

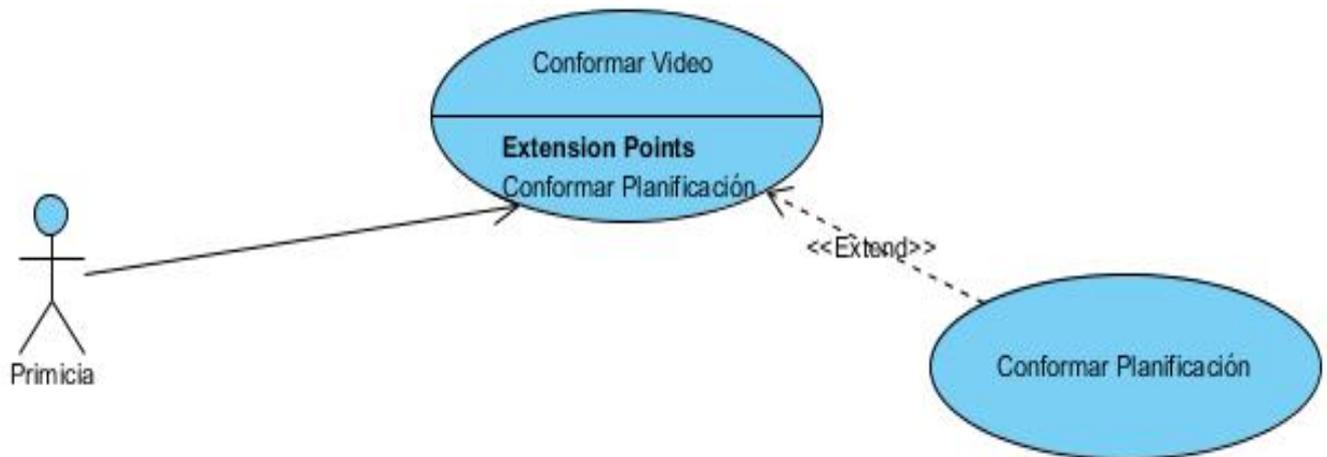


Figura 2 Diagrama de casos de uso del sistema.

3.3.3 Descripción de los casos de uso del sistema.

Caso de Uso:	Conformar Video	
Actores:	Primicia	
Resumen:	El caso de uso se inicia cuando el actor Primicia envía el identificador asociado a una noticia que acaba de ser publicada. Luego de recibido el identificador el sistema se encarga de conformar el video de la noticia. El caso de uso finaliza cuando se insertan los datos asociados al video en la tabla de la base de datos, archivo multimedia.	
Precondiciones:	Que exista una noticia publicada.	
Referencias	RF1	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1- El caso de uso se inicia cuando el actor Primicia envía el identificador de la noticia publicada.	1- Se cargan las imágenes que forman parte de la pantalla. 2- Se carga el texto que forma parte de la pantalla.	

CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

	<p>3- Se genera el video de la pantalla.</p> <p>4- Se unen los videos de las pantallas pertenecientes a la noticia en un solo archivo.</p> <p>5- Se le coloca al video el audio de la noticia.</p> <p>6- Después de conformado el video se guarda en el espacio reservado para el almacenamiento de las medias.</p> <p>7- Guarda los datos del video asociado a la noticia a la tabla archivo multimedia y así finaliza el caso de uso.</p>
Poscondiciones	Se almacena el video correspondiente a la noticia.

Tabla 2 Descripción del caso de uso Conformar Video.

Caso de Uso:	Conformar Planificación	
Actores:	Primicia	
Resumen:	El caso de uso se inicia cuando se reciben los datos del video asociado a una noticia para incluirlo en una planificación, se comprueba la existencia en la base de datos del canal de noticias, se inserta la planificación para el canal finalizando el caso de uso.	
Precondiciones:	El video de la noticia debe estar conformado.	
Referencias	RF2	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
	<p>1- Comprueba la existencia del canal de noticias en la base de datos.</p> <p>2- Inserta los datos del video en una planificación</p>	

	en el canal de noticias y así termina el caso de uso.
Poscondiciones	Se crea la planificación para el canal de noticias.

Tabla 3 Descripción del caso de uso Conformar Planificación.

3.4 Diagramas de clases del análisis.

El diagrama de clases del análisis es uno de los artefactos que se generan en el modelo del análisis. Las clases que se encuentran en estos diagramas se clasifican en Interfaz, Controladora y Entidad (15).

- **Las clases interfaz:** se encargan de modelar la interacción del actor con el sistema.
- **Las clases controladoras:** son usadas para modelar los aspectos dinámicos del sistema, de forma tal, que puedan coordinar las acciones y los flujos de control, además de representar derivaciones y cálculos complejos.
- **Las clases entidad:** se utilizan para modelar la información y comportamiento asociado a algún fenómeno o concepto como una persona, un objeto o suceso del mundo real (15).

Se presentan los diagramas de clases del análisis perteneciente a los casos de uso Conformar video y Conformar Planificación (Ver Figura 3 y Figura 4).

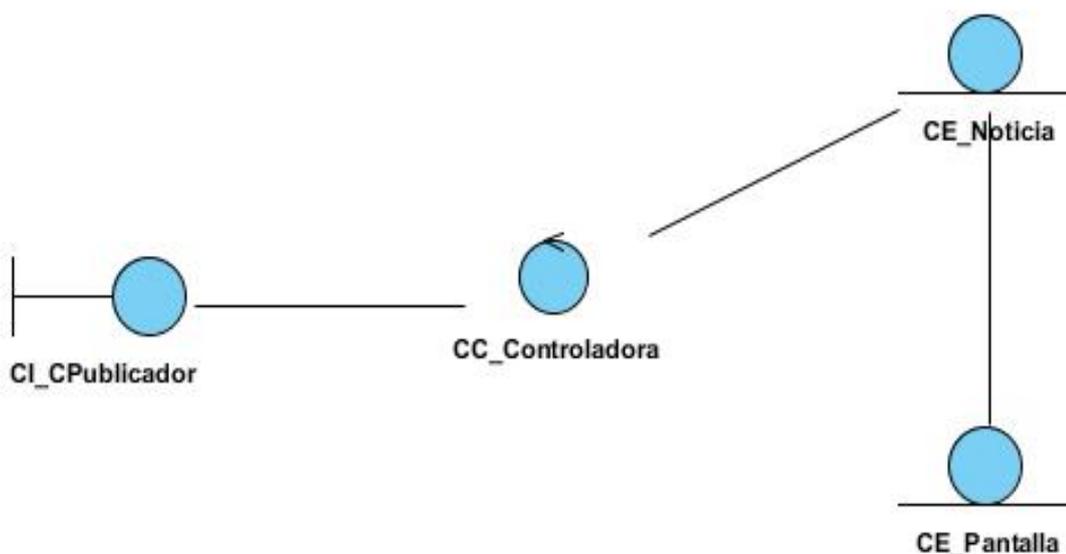


Figura 3 Diagrama de clases del análisis del caso de uso Conformar Video.

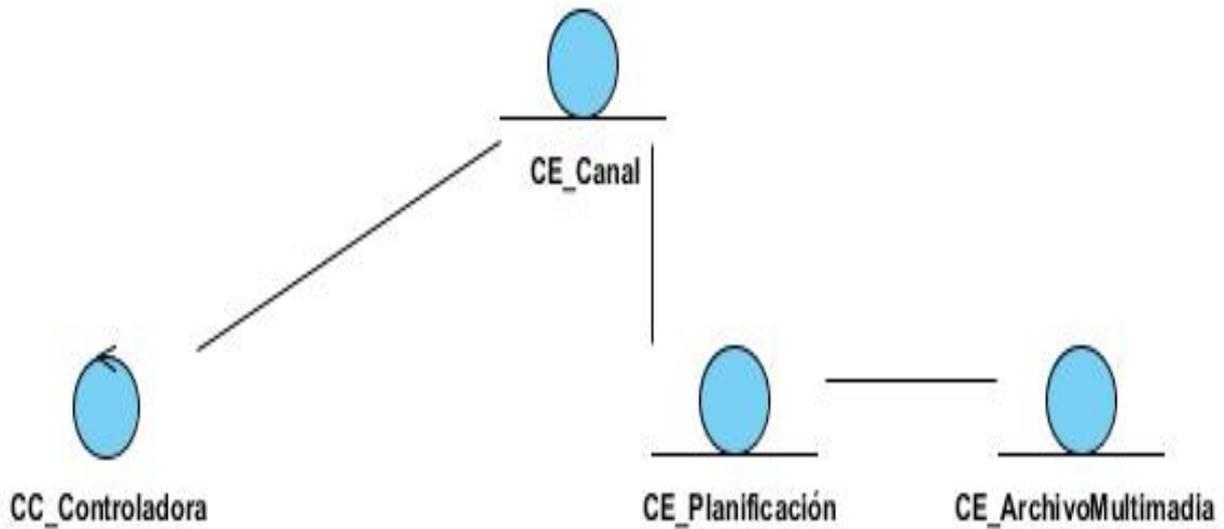


Figura 4 Diagrama de clases del análisis del caso de uso Conformar Planificación.

3.5 Conclusiones parciales.

Teniendo en cuenta que los procesos del negocio no se encuentran definidos con claridad se realizó el modelo de dominio, el cual permitió obtener un conocimiento del entorno donde será utilizada la solución propuesta. Los requisitos funcionales y no funcionales identificados describen las características que debe cumplir el componente para que se pueda realizar la transmisión de noticias mediante un canal virtual. Mediante la descripción de los casos de uso se detalla el flujo de sucesos y la manera en que se realiza la interacción entre el actor y el componente.

Capítulo 4: Construcción de la solución propuesta.

En este capítulo se abordan los aspectos relacionados con la construcción de la solución propuesta. Se definirá la arquitectura del sistema donde se describen las principales características de los patrones del diseño a utilizar. Se realizará el diseño de las funcionalidades identificadas en el capítulo anterior y seguidamente la implementación del sistema. Finalmente se describirán los resultados de las pruebas a la solución presentada.

4.1 Arquitectura de software.

Roger S. Pressman en el libro Ingeniería del software, Un enfoque práctico, define la arquitectura de software como: *“la estructura de las estructuras del sistema, la cual comprende los componentes del software, las propiedades de esos componentes visibles externamente y las relaciones entre ellos”* (35).

La arquitectura de software proporciona una visión global del sistema a construir. Describe las estructuras y la organización de los componentes del software, sus propiedades y las conexiones entre ellos. Es el resultado de ensamblar un cierto número de elementos arquitectónicos para satisfacer la funcionalidad y ejecución de los requisitos del sistema (36).

El modelo en capas organiza el sistema en capas. Cada una de las cuales proporciona un conjunto de servicios que puede pensarse como una máquina abstracta cuyo lenguaje máquina se define por los servicios proporcionados por la misma (36).

La arquitectura en capas soporta el desarrollo incremental de sistemas, a medida que se desarrolla una capa algunos de los servicios que esta provee pueden estar disponibles para los usuarios. Cada capa puede ocuparse exclusivamente de su nivel inmediatamente inferior, a quien solicita servicios y del nivel inmediatamente superior a quien devuelve los resultados (37).

En la solución se utilizará una arquitectura en capas definiéndose como capa de presentación, capa de lógica de negocio y la capa de datos brindando cada una servicios diferentes y necesarios para la implementación del software, las especificidades de cada una de estas se muestran a continuación. (Ver Figura 5)

Capa de presentación: en esta capa se encuentra la interfaz de comunicación del sistema.

Capa de lógica de negocio: esta capa extrae los datos brindados por la capa inferior, necesarios para realizar el procesamiento de las noticias.

Capa de acceso a datos: es la encargada de almacenar los datos y brindar el acceso a los mismos a la capa de negocio.

Ventajas de esta arquitectura:

- El desarrollo se puede llevar a cabo en varios niveles.
- Desarrollos paralelos (en cada capa).
- Aplicaciones más robustas debido al encapsulamiento.
- En caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado.
- Mantenimiento y soporte más sencillo.
- Mayor flexibilidad ya se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad.
- Alta escalabilidad. La principal ventaja de una aplicación distribuida bien diseñada es su buen escalabilidad, es decir, que puede manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware. El crecimiento es casi lineal y no es necesario añadir más código para conseguir esta escalabilidad (36).

Se decide utilizar esta arquitectura pues aumenta considerablemente la localización de errores, mejora el soporte del sistema y facilita la modularidad del mismo. Con el modelo en capas se logra obtener una mejor organización durante el desarrollo del software, cada nivel tiene funcionalidades diferentes lo que permite el diseño de una arquitectura escalable que puede ser ampliada en caso que sea necesario.

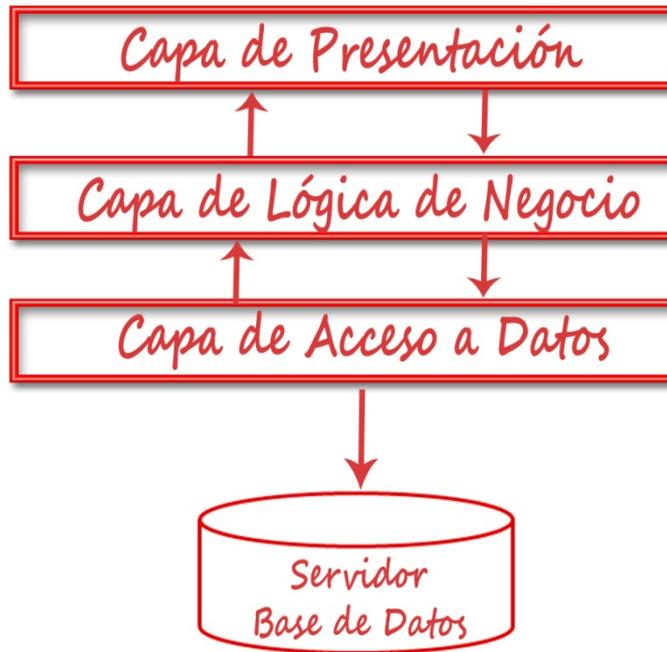


Figura 5 Arquitectura en 3 capas.

4.2 Principios del diseño.

El diseño de software es un proceso iterativo mediante el cual los requisitos se traducen en lo que se podría llamar un plano para construir el software. A medida que suceden estas iteraciones se va refinando el diseño lo que conduce a niveles de abstracción muchos más bajos. Proporcionando las representaciones de software que permitan evaluar la calidad del mismo. El diseño del software se puede considerar tanto un modelo como un proceso:

Proceso: es una secuencia de pasos que hacen posible que el diseñador describa todos los aspectos del software que se va a construir.

Modelo: se representa todo lo que se va a construir (33).

Según Roger S. Pressman *“los principios básicos del diseño de software son los que hacen posible al ingeniero de software navegar por el proceso de diseño”*.

- En el proceso de diseño no deberá utilizarse orejeras.
- El diseño no deberá inventar nada que ya esté inventado.
- El diseño deberá presentar uniformidad e integración.

- El diseño deberá estructurarse para admitir cambios.
- El diseño deberá evaluarse en función de la calidad mientras se va creando, no después de terminado.
- El diseño deberá revisarse para minimizar los errores conceptuales. (33)

4.3 Patrones de diseño.

Los patrones de diseño son los que permiten al diseñador crear la arquitectura del diseño integrando componentes reusables. Son las abstracciones genéricas similares entre aplicaciones que muestran los objetos abstractos, concretos y sus interacciones (37).

Según Ian Sommerville *“los patrones y los lenguajes de patrones son formas de describir las mejores prácticas, buenos diseños, y encapsulan la experiencia de tal forma que es posible para otros utilizar dicha experiencia”* (37).

Los patrones de diseño GRASP (General Responsibility Assignment Software Patterns) o patrones de asignación de responsabilidades describen los principios fundamentales de la asignación de responsabilidades a objetos, expresado en forma de patrones (38).

Los patrones GoF (Gang of Four) o pandilla de los cuatro como se le conoce son agrupados en tres grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento (39).

GRASP.

Patrón Experto.

El patrón Experto propone la asignación de la responsabilidad de experto a la clase que cuenta con la información necesaria para cumplir con la misma. Si esta asignación se hace de manera correcta se puede entender el sistema de una mejor manera siendo más fácil de mantener y ampliar, brindando la oportunidad de reutilizar sus componentes en futuras aplicaciones (38).

Este patrón fue usado en las clases creadas en la implementación del sistema para que cada una contenga los métodos relacionados con la información que posee. En algunos casos se manejan solo funcionalidades correspondientes con los atributos de cada una, en otros casos solo las que son necesarias para cumplir el objetivo con el que fueron creadas. Ejemplo es la clase CNoticia la cual contiene solo métodos afines con los datos con los que cuenta (Ver Figura 6).

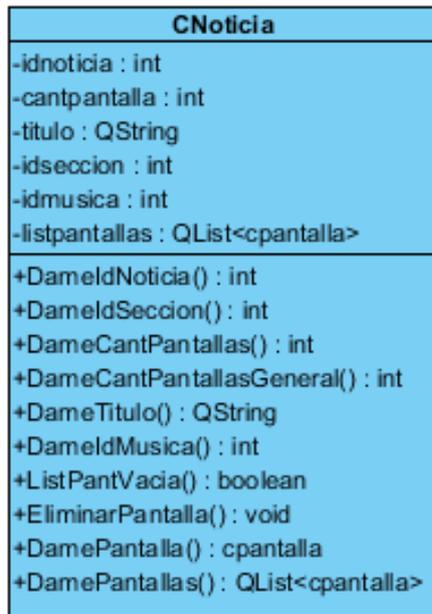


Figura 6 Ejemplo de patrón Experto.

Patrón Creador.

El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos. Se aplicará en todos los casos donde una clase tiene la responsabilidad de crear una nueva instancia de la otra. La correcta asignación permite que la aplicación pueda soportar un bajo acoplamiento, una mayor claridad, el encapsulamiento y reutilización (38).

Un ejemplo del empleo de este patrón se puede evidenciar en la clase CControladora la cual crea un objeto de la clase CProcesamiento para acceder a los métodos de esta clase que son los que realizan todas las operaciones con las imágenes, sonido y video (Ver Figura 7).

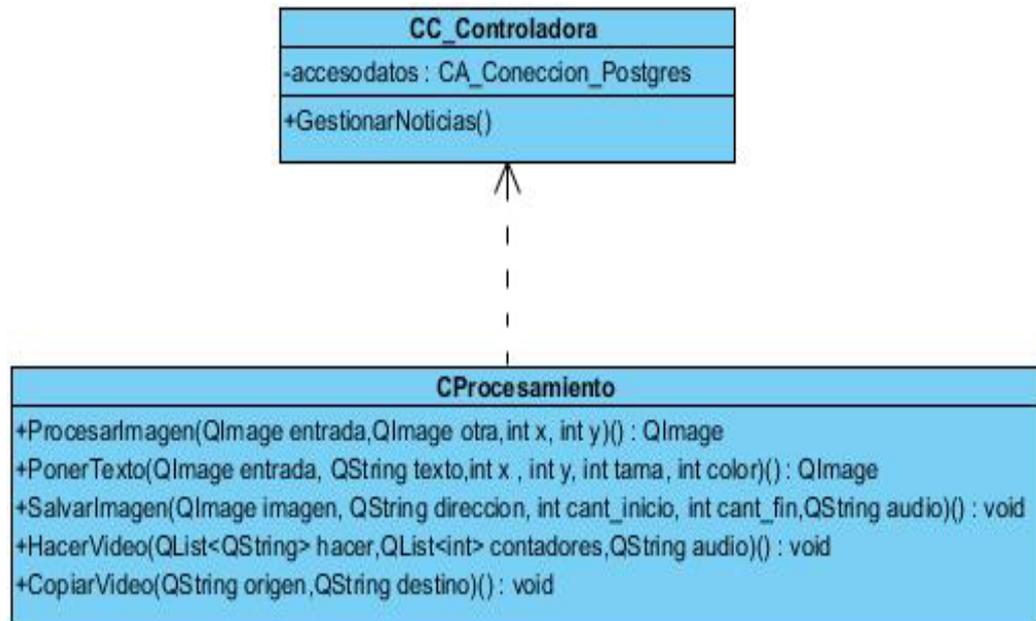


Figura 7 Ejemplo de patrón Creador.

Patrón Bajo Acoplamiento.

El patrón de Bajo Acoplamiento estimula a asignar responsabilidades de modo tal que una clase no dependa de muchas otras. Tener un bajo acoplamiento soporta el diseño de clases más independientes que reducen el impacto de los cambios y permite una mayor reutilización de código (38).

Este patrón se pone de manifiesto en todas las clases del componente. (Ver Figura 9)

GoF.

Patrón Observador.

El patrón observador define una dependencia del tipo “uno a muchos” entre objetos, de manera que cuando uno de los objetos cambia su estado, el observador se encarga de notificar este cambio a todos los otros dependientes (39) (Ver Figura 8).

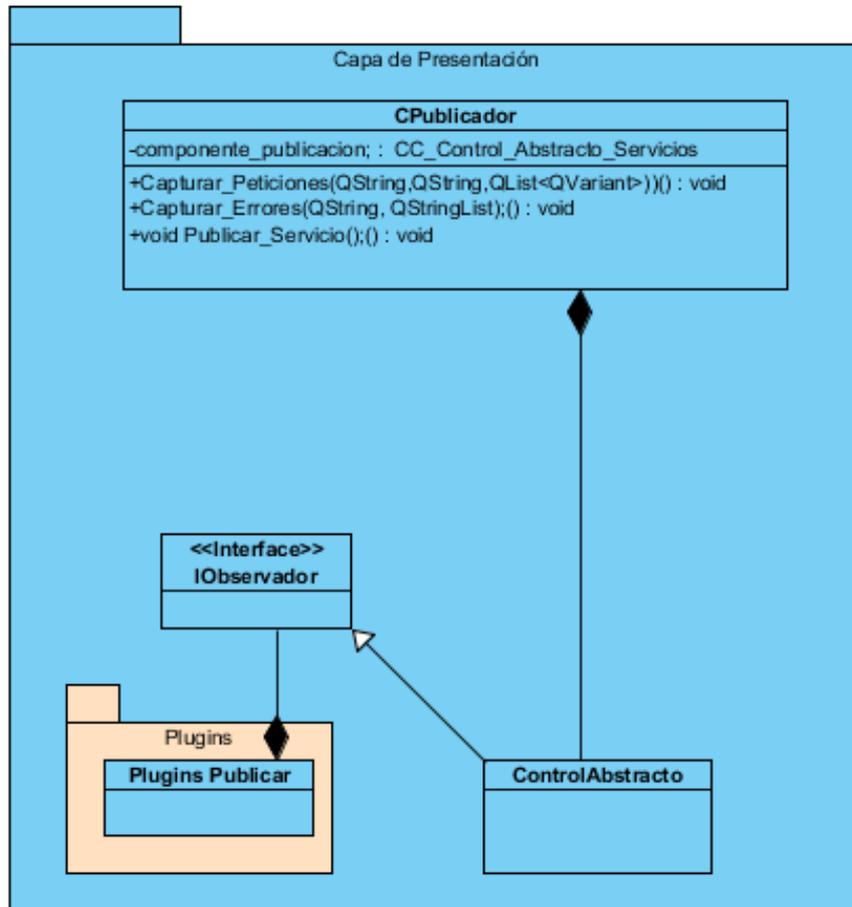


Figura 8 Ejemplo de patrón Observador.

4.4 Diagrama de clases del diseño.

Representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas. Sirve para visualizar las relaciones entre las clases que involucran el sistema. Un diagrama de clases está compuesto por los siguientes elementos:

- **Clase:** atributos, métodos y visibilidad.
- **Relaciones:** herencia, composición, agregación, asociación y uso.

Los diagramas de clases del diseño constituyen un elemento fundamental en la concepción de la aplicación que se propone, ya que servirán de guía a los desarrolladores al constituir una aproximación del sistema que se desea implementar, contribuyendo de esta forma a la calidad del producto final.

(35)

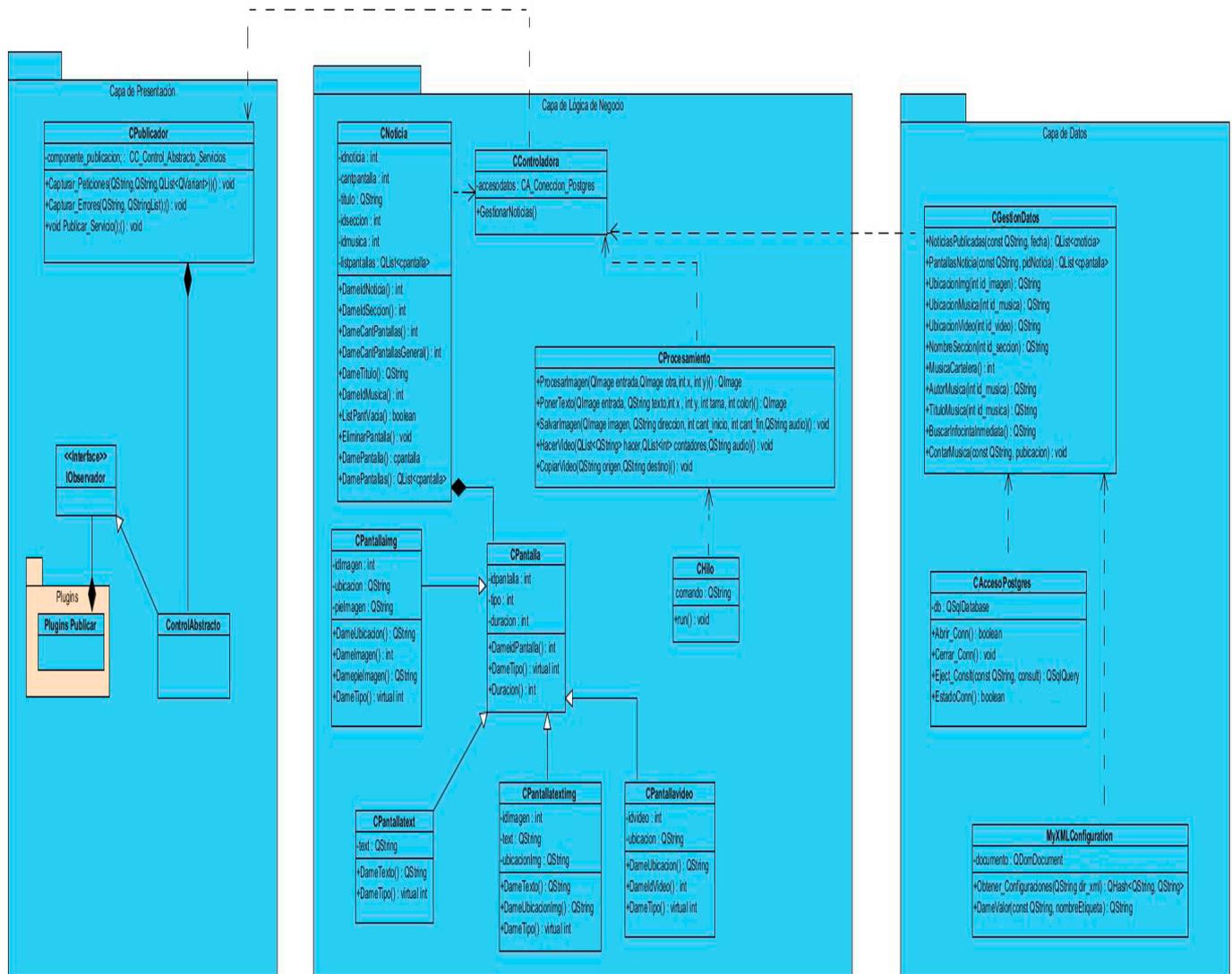


Figura 9 Diagrama de clases del diseño del sistema.

4.5 Diagrama de despliegue.

El diagrama de despliegue muestra el entorno computacional donde se encuentra instalado el sistema, visualiza la configuración de los elementos de procesamiento en tiempo de ejecución con sus respectivos procesos de software (16). El nodo Primicia envía por XML-RPC el identificador de la noticia que acaba de publicarse, en el nodo cliente donde se encuentra el componente se consulta el servidor de base de datos para solicitarle los datos de la noticia y el repositorio de medias para

acceder a los archivos que componen la noticia y comenzar el procesamiento de la información. Para establecer las conexiones al repositorio de medias y al servidor de base de datos se utiliza el protocolo TCP/IP que permite la transmisión de datos.

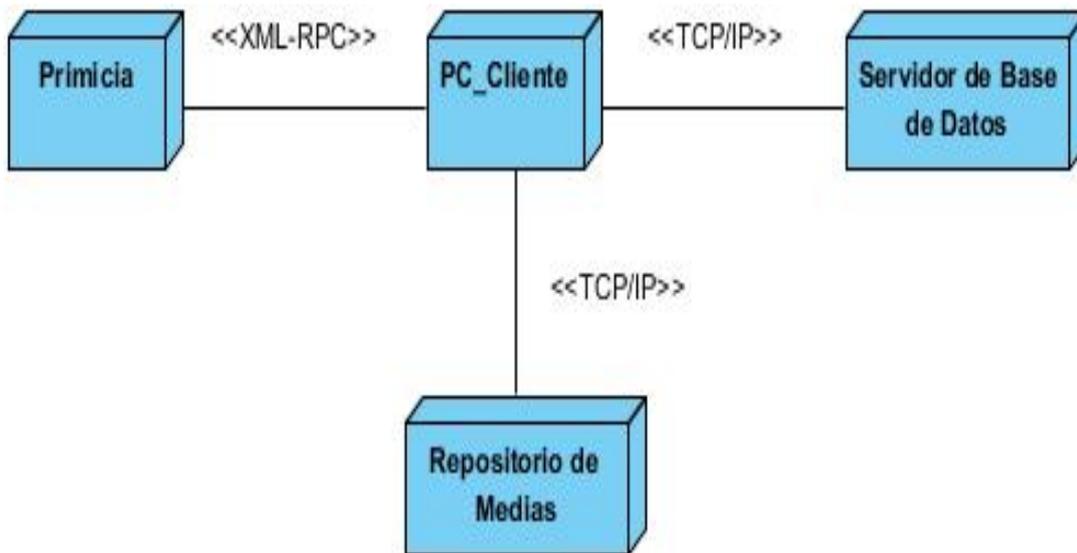


Figura 10 Diagrama de despliegue.

4.6 Modelo de implementación.

El modelo de implementación describe cómo las clases se implementan en términos de componentes, como ficheros de código fuente, ejecutables, entre otros. El modelo de implementación describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modelación disponible en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes uno de otros. (40)

4.6.1 Diagrama de componentes.

“Los diagramas de componentes describen los elementos del sistema y sus relaciones. Los componentes representan todos los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, bibliotecas cargadas dinámicamente, entre otros. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente” (41).

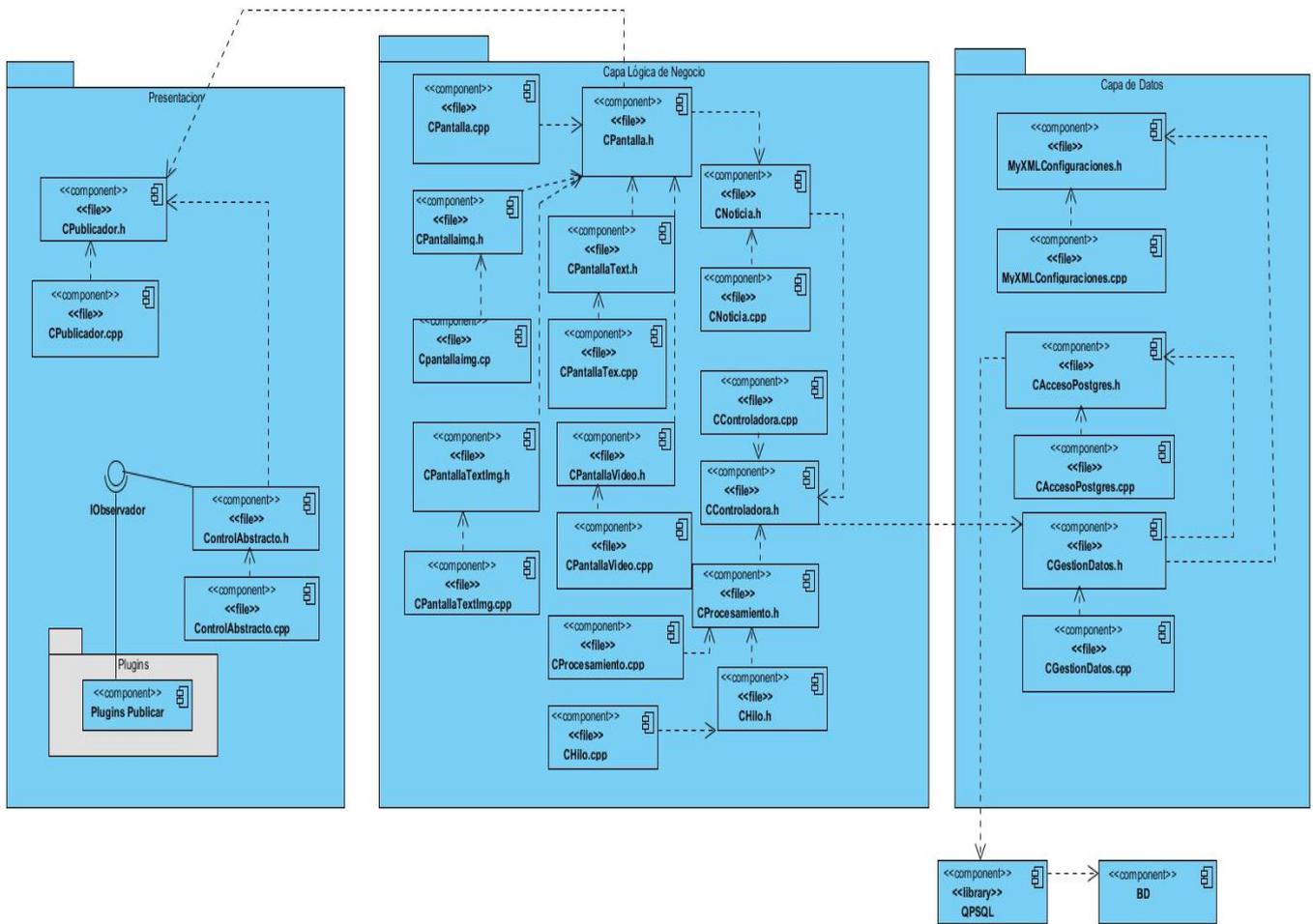


Figura 11 Diagrama de componentes del sistema.

4.7 Prueba de software.

Las pruebas de software consisten en una serie de acciones en las que el sistema es ejecutado bajo condiciones y requisitos específicos. Los resultados son observados y registrados realizando una evaluación de la calidad del producto para ser entregado (15).

Al componente se le realizaron pruebas de caja blanca las cuales se basan en un minucioso examen de los detalles procedimentales del código, por lo que es necesario conocer la lógica del programa.

4.7.1 Prueba de caja blanca.

Con la realización de pruebas de caja blanca se garantiza que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método así como todas las decisiones lógicas en las vertientes verdadera y falsa. Se ejecutan también todos los bucles en sus límites operacionales, además se ejercitan las estructuras internas de datos para asegurar su validez. Por

esto se considera que este tipo de pruebas permite que las aplicaciones cuenten con una mayor calidad y confiabilidad ya que las mismas contribuyen a disminuir el número de errores que puedan existir en los sistemas.

La prueba del camino básico propuesta por Tom McCabe es una técnica de prueba de Caja Blanca. Esta permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico.

Los pasos que se siguen para aplicar esta técnica son:

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

Un grafo de flujo está constituido por tres componentes fundamentales que permiten una mejor comprensión del mismo, además de brindar información para garantizar que el proceso se esté realizando de manera correcta.

Los componentes son:

Nodo: cada círculo se denomina nodo del Grafo de Flujo, este es usado para representar una o más secuencias procedimentales. Un solo nodo puede corresponder a una sentencia de decisión o secuencia de procesos.

Aristas: las flechas del grafo son denominadas como aristas y representan el flujo de control, son análogas a las representadas en un diagrama de flujo. Una arista debe terminar en un nodo, incluso aunque este nodo no represente ninguna sentencia procedimental.

Regiones: las regiones son las áreas delimitadas por las aristas, nodos y el área exterior del grafo, este se cuenta como una región más. Estas se enumeran y la cantidad de regiones es equivalente a la cantidad de caminos independientes del conjunto básico de un programa.

La Complejidad Ciclomática $V(G)$ se define como:

$$V(G) = A - N + 2$$

Donde A es el numero de aristas existentes en el grafo y N el de nodos. (33)

Prueba al método EfectuarProcesamiento de la clase CProcesamiento.

QImage CProcesamiento:: EfectuarProcesamiento(QImage entrada,QImage otra, int x, int y)

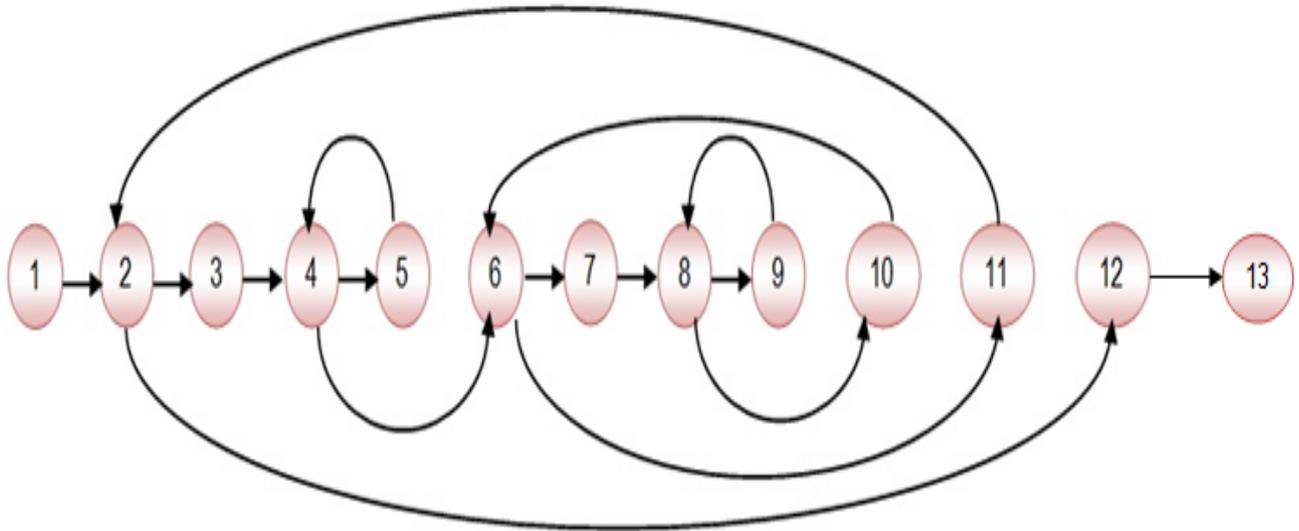
```

{
  QImage salida(entrada.size(), entrada.format()); 1
  for (int i = 0; i < entrada.height(); i++) 2
  {
    const QRgb* inPixels = reinterpret_cast<const QRgb*> (entrada.scanLine(i));
    QRgb* outPixels = reinterpret_cast<QRgb*> (salida.scanLine(i)); 3
    for (int j = 0; j < entrada.width(); j++) 4
    {
      int red = qRed(inPixels[j]);
      int green = qGreen(inPixels[j]);
      int blue = qBlue(inPixels[j]);
      int alpha = qAlpha(inPixels[j]);
      outPixels[j] = qRgba(red, green, blue, alpha); 5
    }

    for (int k = 0; k < otra.height(); k++) 6
    {
      const QRgb* inPixels1 = reinterpret_cast<const QRgb*> (otra.scanLine(k));
      QRgb* outPixels1 = reinterpret_cast<QRgb*> (entrada.scanLine(k+y)); 7
      for (int l = 0; l < otra.width(); l++) 8
      {
        int red = qRed(inPixels1[l]);
        int green = qGreen(inPixels1[l]);
        int blue = qBlue(inPixels1[l]);
        int alpha = qAlpha(inPixels1[l]);
        outPixels1[l+x] = qRgba(red, green, blue, alpha); 9
      } 10
    } 11
  } 12
}

```

```
return salida;
}
```



Complejidad Ciclomática.

$$V(G) = A - N + 2$$

$$V(G) = 16 - 13 + 2$$

$$V(G) = 5$$

Caminos Independientes.

R1: 1-2-12-13

R2: 1-2-3-4-6-11-2-12-13

R3: 1-2-3-4-5-4-6-11-2-12-13

R4: 1-2-3-4-5-4-6-7-8-10-6-11-2-12-13

R5: 1-2-3-4-5-4-6-7-8-9-8-10-6-11-2-12-13

Camino	Caso de prueba	Función del código	Resultado
R5: 1-2-3-4-5-4-6-7-8-9-8-10-6-11-2-12-13	EfectuarProcesamiento	La función debe de poner una imagen sobre otra a partir de	Satisfactorio

		unas coordenadas especificadas.	
--	--	---------------------------------	--

Prueba al método HacerVideo de la clase CProcesamiento.

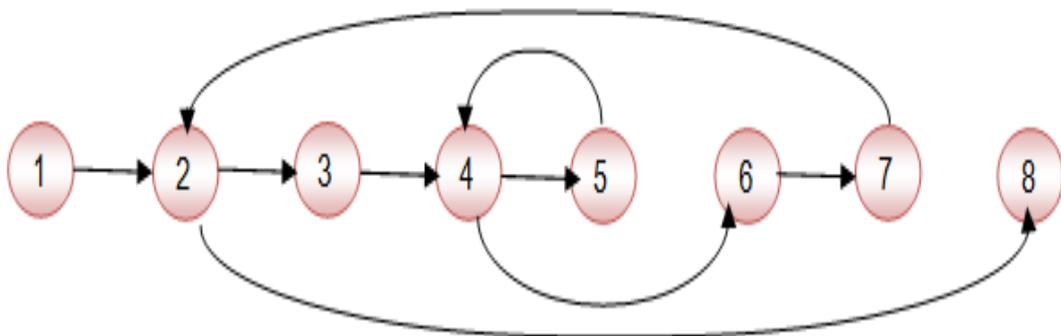
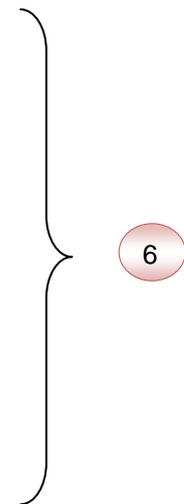
```
void CProcesamiento ::HacerVideo(QList<QString> hacer,QList<int> contadores, QString audio,int duracion, int inicio)
```

```
{
  QString cadena;
  QString cadena2;
  QString completa;
  QString elim;
  for (int i = 0; i < hacer.count();i ++){
    cadena2 = hacer.at(i);
    int cantidad =contadores.at(i);
    for (int j = 1; j < cantidad+1; j++){
      cadena= cadena + " "+cadena2+"/"+QString::number(j)+".webm" ;
    }
    completa = " mencoder -ovc copy "+ cadena + " -o "+ cadena2 +"/"+"completo.webm";
    Hilo *h = new Hilo(completa);
    h->run();
    int duracion=cantidad*12 ;
    QString cadena_audio= cadena2+"/"+"audio.ogg";
    QString musica= "ffmpeg -i "+ audio + " -t "+ QString::number(duracion) +
    " -ss "+ QString::number(inicio) + " -acodec libvorbis "+ cadena_audio;
    Hilo *m = new Hilo(musica);
    m->run();
    QString cadena_video = cadena2 +"/"+"completo.webm";
    QString video_audio= cadena2 +"/"+"noticia_.webm";
    QString video= "ffmpeg -i "+ cadena_audio + " -i "+ cadena_video+ " "+ video_audio;
    Hilo *v = new Hilo(video);
    v->run();
  }
}
```

```

elim = " rm " + cadena ;
Hilo *k=new Hilo(elim);
k->run();
cadena="";
cadena2="";
QString eliminar_video= " rm "+cadena_video;
Hilo *e= new Hilo(eliminar_video);
e->run();
QString eliminar_audio= " rm "+cadena_audio;
Hilo *a= new Hilo(eliminar_audio);
a->run();
}
}

```



Complejidad Ciclomática.

$$V(G) = A - N + 2$$

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

$$V(G) = 3$$

Caminos Independientes.

R1: 1-2-8

R2: 1-2-3-4-6-7-2-8

R3: 1-2-3-4-5-4-6-7-2-8

Camino	Caso de prueba	Función del código	Resultado
R3: 1-2-3-4-5-4-6-7-2-8	HacerVideo	La función debe unir en un solo archivo todos los videos que se encuentran en un directorio, ponerles el audio y eliminar todos los archivos que se encontraban en el directorio excepto el video que contiene el audio.	Satisfactorio

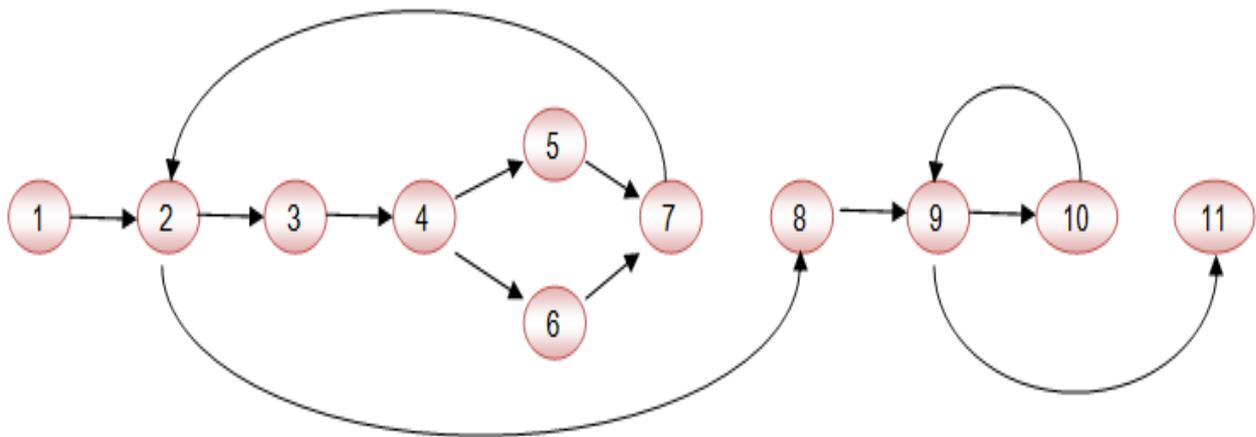
Prueba al método PonerTextoNuevo de la clase CProcesamiento.

```

QImage open::PonerTextoNuevo(QImage entrada, QString textonoticia, int x , int y, int tama, int color )
{
    QImage image=entrada;
    QImage tmplImage = image.convertToFormat(QImage::Format_ARGB32_Premultiplied);
    QPainter painter;
    painter.begin(&tmplImage);
    QString texto = textonoticia;
    int lon = 70;
    QStringList espacio = texto.split(' ');
    QString contenedor;
    QStringList *miparrafo = new QStringList();
    for (int i = 0; i < espacio.count(); i++)
    {
        int CantCaracteres = espacio.at(i).count();
        int CantRealContador = contenedor.count();
        if((CantCaracteres + CantRealContador)<=lon)
        {
            contenedor+=" "+espacio.at(i);
        }
        else
    }
}
    
```

```

{
miparrafo->append(contenedor);           } 6
contenedor = espacio.at(i);
}
} 7
miparrafo->append(contenedor);           } 8
QFont serifFont("Times", tama, QFont::Bold);
painter.setFont(serifFont);
for (int i = 0; i < miparrafo->count(); i++) 9
{
QColor col = QColor(0,0,0,255);         } 10
painter.setPen(Qt::white);
painter.drawText( x, y ,miparrafo->at(i));
y+=40;
}
painter.end();
image = tmpImage.convertToFormat(QImage::Format_RGB32); } 11
return image;
}
    
```



Complejidad Ciclomática.

$$V(G) = A - N + 2$$

$$V(G) = 13 - 11 + 2$$

$$V(G) = 4$$

Caminos Independientes.

R1: 1-2-8-9-11

R2: 1-2-3-4-5-7-2-8-9-11

R3: 1-2-3-4-6-7-2-8-9-11

R4: 1-2-8-9-10-9-11

Camino	Caso de prueba	Función del código	Resultado
R3: 1-2-3-4-5-4-6-7-2-8	PonerTextoNuevo	La función debe poner el texto sobre una imagen a partir de unas coordenadas específicas.	Satisfactorio

Los resultados de las pruebas aplicadas a los diferentes métodos fueron satisfactorios, ya que al ejecutarlos se recorren todos los nodos del camino al menos una vez. Después de realizado un análisis detallado de cada uno de los resultados anteriormente expuestos se llega a la conclusión de que el componente no presenta ningún problema en cuanto a las funcionalidades del código fuente.

4.8 Conclusiones parciales.

La utilización de la arquitectura en tres capas permite la comprensión de los flujos de los casos de uso del sistema brindando un modelo organizativo a la solución propuesta. La definición de las clases del diseño y el uso de patrones del diseño permite una correcta estructuración del código y una mejor comprensión a la hora de realizar la implementación. La realización de las pruebas para validar la solución propuesta, arrojaron como resultado que la aplicación cumple de manera satisfactoria con las funcionalidades identificadas en el proceso de desarrollo de software.

Conclusiones generales.

Con la terminación del proceso de investigación se concluye que:

- Se desarrolló un componente que permite realizar la transmisión de noticias a través de un canal virtual aumentando las modalidades de distribución de noticias del SIAV.
- La investigación permitió caracterizar los sistemas de gestión y transmisión de señales televisivas existentes en la actualidad, lo que permitió tener un mejor entendimiento de su funcionamiento.
- Las tecnologías y herramientas seleccionadas para el desarrollo del componente ofrecieron el soporte necesario para lograr un producto que cumpla con los requisitos.
- El proceso de desarrollo permitió generar todos los artefactos y documentación correspondientes al mismo, sirviendo como base para la implementación del componente.
- La implementación del componente permitió realizar la transmisión y gestión de noticias generadas por la plataforma PRIMICIA a través de canales virtuales.
- La realización de las pruebas arrojó resultados satisfactorios, por lo que el Departamento de señales digitales cuenta con un componente que hace posible la transmisión de noticias y de canales virtuales desde un mismo subsistema.

Recomendaciones.

Luego de culminada la investigación se recomienda a la institución:

- Continuar la investigación referente al trabajo con escenas en el marco de trabajo Qt para valorar la posibilidad de su uso en la generación de los videos asociados a las noticias.
- Utilizar el componente en una computadora que cuente con un procesamiento mayor a la empleada durante su desarrollo.

Bibliografía referenciada.

1. ICRT. Portal de la Televisión Cubana. [En línea] [Citado el: 13 de Enero de 2012.] <http://www.tvcubana.icrt.cu/seccion-historia/46-historia/236-1950-cuba-llego-la-television>.
2. Dábalos, Gabriel. El apagón de la televisión analógica en Cuba: 10 respuestas sobre la televisión digital. [En línea] 2010. [Citado el: 26 de Noviembre de 2011.] www.envivo.icrt.cu/tecnologia/77-el-apagon-de-la-television-analogica-en-cuba.
3. Universidad de las Ciencias Informáticas . Universidad de las Ciencias Informáticas . [En línea] 2012. [Citado el: 20 de Octubre de 2011.] www.uci.cu.
4. Real Academia Española. [En línea] [Citado el: 14 de Noviembre de 2011.] <http://www.rae.es/rae.html>.
5. Definición.de. [En línea] [Citado el: 14 de Noviembre de 2011.]
6. The Free dictionary. [En línea] [Citado el: 14 de Noviembre de 2011.]
7. Lorente Algora, Raul. Conceptos Básicos del video digital. [En línea] 2003. [Citado el: 1 de Abril de 2012.] <http://ciberconta.unizar.es/leccion/videos/index.htm>.
8. Agusti, Manuel, Benloch, Jose y Atienza, Vicente. Introducción a la edición de video. Compresión MPEG. Valencia, España : s.n., 2011.
9. Cruz, María Del Carmen Gómez. “Estudio del Streaming de audio y video sobre redes Heterogéneas. Madrid : s.n., 2010.
10. García, Joaquín. Posibilidades y retos de la television digital. [En línea] [Citado el: 14 de Febrero de 2012.] <http://www.razonypalabra.org.mx/anteriores/n45/jgarcia.html>.
11. X Frame Software, S.L. X Frame TV. [En línea] [Citado el: 29 de Noviembre de 2011.] <http://www.xfs.es/>.
12. ISID Gestión Inteligente de Activos Multimedia. Videoma. [En línea] [Citado el: 29 de Noviembre de 2011.] http://www.isid.es/Spanish/product/producto_v00.htm.
13. Estructure Media Systems. Estructure . [En línea] <http://www.estructure.es>.

14. La tecla de Escape . [En línea] [Citado el: 6 de Marzo de 2012.] <http://latecladeescape.com/articulos/1550-metodologias-de-desarrollo-del-software>.
15. Ivar Jacobson, Grady Booch, James Rumbaugh. El Proceso Unificado de Desarrollo de Software. Madrid : Pearson Education S.A, 2000.
16. —. El Lenguaje Unificado de Modelado.Manual de Referencia . s.l. : Addison Wesley.
17. Valencia, Universidad Politécnica de. Introducción a Herramientas CASE. Valencia : s.n., 2010.
18. Univ. Cantabria – Fac. de Ciencias. Trabajando con Visual Paradigm for. Cantabria.
19. Universidad de tecnología Nacional Facultad Regional Tucumán . Universidad de tecnología Nacional. [En línea] 2012. [Citado el: 6 de Marzo de 2012.] <http://www.frt.utn.edu.ar/sistemas/paradigmas/lenguajes.htm>.
20. Universidad de Alicante. [En línea] 2010. [Citado el: 2 de Febrero de 2012.] <http://publicaciones.ua.es/publica/Detalles.aspx?fndCod=LI9788479088880&idet=958>.
21. González, Carlos D. Introducción a C++ y a la resolución de problemas. Buenos Aires, Argentina : s.n., 2012.
22. CodeBox. CodeBox. [En línea] [Citado el: 14 de Febrero de 2012.] <http://www.codebox.es/glosario>.
23. Rus, Dr. Francisco Javier Melero. Desarrollo de aplicaciones con Qt. s.l. : Dpto. de Lenguajes y Sistemas Informáticos, Escuela Técnica Superior de Ingeniería Informática.
24. Soluciones vBulletin Inc. . Ighnetwork. [En línea] 2012. [Citado el: 5 de Marzo de 2012.] <http://foro.ighnetwork.net/showthread.php?15188-IDE-Entorno-integrado-de-desarrollo-%28Concepto-importante%29>.
25. Nokia. QT. [En línea] 2008-2012 Nokia Corporation. [Citado el: 6 de Febrero de 2012.] <http://qt.nokia.com/>.
26. Computer Audio y Video System Integrator. CAVSI. [En línea] [Citado el: 6 de Marzo de 2012.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
27. Sistemas Gestores de Bases de Datos . Ramírez, Raquel Zambrano. 14, Cordoba : Granada, 2008. ISSN 1988-6047.

28. Medrano, Raúl Igual Carlos. Tutorial de OpenCV. Teruel : s.n., 2008.
29. Gstreamer. Gstreamer. [En línea] [Citado el: 14 de Febrero de 2012.] <http://gstreamer.freedesktop.org/>.
30. FFmpeg. FFmpeg. [En línea] [Citado el: 13 de Febrero de 2012.] <http://ffmpeg.org/download.html>.
31. Mencoder. Ayuda de Mencoder. 2011.
32. Garcerant, Iván. Tecnología y Synergix. Modelo de Dominio. [En línea] [Citado el: 16 de Febrero de 2012.] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
33. Pressman, Roger S. Ingeniería de Software, Un enfoque práctico. . s.l. : s.l. : Mcgraw-hill - España. 5ta Edición.
34. RUP., Ayuda de. s.l. : s.l. : Corporación IBM, 2006.
35. Pressman, Roger S. La Ingeniería del software, un enfoque practico. La Habana : Felix Varela , 2005.
36. Kernelerror. Arquitectura en 3 Capas. [En línea] [Citado el: 27 de 3 de 2012.] <http://kernelerror.net/programacion/php/arquitectura-3-capas/>.
37. Sommerville, Ian. Ingeniería de Software, 7ma edición . Madrid España : Pearson Education S.A , 2005.
38. Larman, Craig. UML y Patrones Introducción al análisis y diseño orientado a objetos. México : PRENTICE HALL, 1999.
39. Geek the planet. [En línea] [Citado el: 10 de 5 de 2012.] <http://geektheplanet.net/5462/patrones-gof.xhtml>.
40. MeRinde V1.1.0 . [En línea] [Citado el: 10 de Abril de 2012.] merinde.net/index.php?option=com_content&task=view&id=495&Itemid=291.
41. Letelier Torres, Patricio. Desarrollo de software orientado a objetos usando UML. Valencia : Universidad Politécnica de Valencia, 2007.
42. Scripting News. [En línea] XML-RPC, 2004-2011 . [Citado el: 25 de 04 de 2012.] <http://xmlrpc.scripting.com/>.

Bibliografía consultada.

1. ICRT. Portal de la Televisión Cubana. [En línea] [Citado el: 13 de Enero de 2012.] <http://www.tvcubana.icrt.cu/seccion-historia/46-historia/236-1950-cuba-llego-la-television>.
2. Dábalos, Gabriel. El apagón de la televisión analógica en Cuba: 10 respuestas sobre la televisión digital. [En línea] 2010. [Citado el: 26 de Noviembre de 2011.] www.envivo.icrt.cu/tecnologia/77-el-apagon-de-la-television-analogica-en-cuba.
3. Universidad de las Ciencias Informáticas . Universidad de las Ciencias Informáticas . [En línea] 2012. [Citado el: 20 de Octubre de 2011.] www.uci.cu.
4. Real Academia Española. [En línea] [Citado el: 14 de Noviembre de 2011.] <http://www.rae.es/rae.html>.
5. Definición.de. [En línea] [Citado el: 14 de Noviembre de 2011.]
6. The Free dictionary. [En línea] [Citado el: 14 de Noviembre de 2011.]
7. Lorente Algora, Raul. Conceptos Básicos del video digital. [En línea] 2003. [Citado el: 1 de Abril de 2012.] <http://ciberconta.unizar.es/leccion/videos/index.htm>.
8. Agusti, Manuel, Benloch, Jose y Atienza, Vicente. Introducción a la edición de video. Compresión MPEG. Valencia, España : s.n., 2011.
9. Cruz, María Del Carmen Gómez. “Estudio del Streaming de audio y video sobre redes Heterogéneas. Madrid : s.n., 2010.
10. García, Joaquín. Posibilidades y retos de la television digital. [En línea] [Citado el: 14 de Febrero de 2012.] <http://www.razonypalabra.org.mx/anteriores/n45/jgarcia.html>.
11. X Frame Software, S.L. X Frame TV. [En línea] [Citado el: 29 de Noviembre de 2011.] <http://www.xfs.es/>.
12. ISID Gestión Inteligente de Activos Multimedia. Videoma. [En línea] [Citado el: 29 de Noviembre de 2011.] http://www.isid.es/Spanish/product/producto_v00.htm.
13. Estructure Media Systems. Estructure . [En línea] <http://www.estructure.es>.

14. La tecla de Escape . [En línea] [Citado el: 6 de Marzo de 2012.] <http://latecladeescape.com/articulos/1550-metodologias-de-desarrollo-del-software>.
15. Ivar Jacobson, Grady Booch, James Rumbaugh. El Proceso Unificado de Desarrollo de Software. Madrid : Pearson Education S.A, 2000.
16. —. El Lenguaje Unificado de Modelado.Manual de Referencia . s.l. : Addison Wesley.
17. Valencia, Universidad Politécnica de. Introducción a Herramientas CASE. Valencia : s.n., 2010.
18. Univ. Cantabria – Fac. de Ciencias. Trabajando con Visual Paradigm for. Cantabria.
19. Universidad de tecnología Nacional Facultad Regional Tucumán . Universidad de tecnología Nacional. [En línea] 2012. [Citado el: 6 de Marzo de 2012.] <http://www.frt.utn.edu.ar/sistemas/paradigmas/lenguajes.htm>.
20. Universidad de Alicante. [En línea] 2010. [Citado el: 2 de Febrero de 2012.] <http://publicaciones.ua.es/publica/Detalles.aspx?fndCod=LI9788479088880&idet=958>.
21. González, Carlos D. Introducción a C++ y a la resolución de problemas. Buenos Aires, Argentina : s.n., 2012.
22. CodeBox. CodeBox. [En línea] [Citado el: 14 de Febrero de 2012.] <http://www.codebox.es/glosario>.
23. Rus, Dr. Francisco Javier Melero. Desarrollo de aplicaciones con Qt. s.l. : Dpto. de Lenguajes y Sistemas Informáticos, Escuela Técnica Superior de Ingeniería Informática.
24. Soluciones vBulletin Inc. . Ighnetwork. [En línea] 2012. [Citado el: 5 de Marzo de 2012.] <http://foro.ighnetwork.net/showthread.php?15188-IDE-Entorno-integrado-de-desarrollo-%28Concepto-importante%29>.
25. Nokia. QT. [En línea] 2008-2012 Nokia Corporation. [Citado el: 6 de Febrero de 2012.] <http://qt.nokia.com/>.
26. Computer Audio y Video System Integrator. CAVSI. [En línea] [Citado el: 6 de Marzo de 2012.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
27. Sistemas Gestores de Bases de Datos . Ramírez, Raquel Zambrano. 14, Cordoba : Granada, 2008. ISSN 1988-6047.

28. Medrano, Raúl Igual Carlos. Tutorial de OpenCV. Teruel : s.n., 2008.
29. Gstreamer. Gstreamer. [En línea] [Citado el: 14 de Febrero de 2012.] <http://gstreamer.freedesktop.org/>.
30. FFmpeg. FFmpeg. [En línea] [Citado el: 13 de Febrero de 2012.] <http://ffmpeg.org/download.html>.
31. Mencoder. Ayuda de Mencoder. 2011.
32. Garcerant, Iván. Tecnología y Synergix. Modelo de Dominio. [En línea] [Citado el: 16 de Febrero de 2012.] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
33. Pressman, Roger S. Ingeniería de Software, Un enfoque práctico. . s.l. : s.l. : Mcgraw-hill - España. 5ta Edición.
34. RUP., Ayuda de. s.l. : s.l. : Corporación IBM, 2006.
35. Pressman, Roger S. La Ingeniería del software, un enfoque practico. La Habana : Felix Varela , 2005.
36. Kernelerror. Arquitectura en 3 Capas. [En línea] [Citado el: 27 de 3 de 2012.] <http://kernelerror.net/programacion/php/arquitectura-3-capas/>.
37. Sommerville, Ian. Ingeniería de Software, 7ma edición . Madrid España : Pearson Education S.A , 2005.
38. Larman, Craig. UML y Patrones Introducción al análisis y diseño orientado a objetos. México : PRENTICE HALL, 1999.
39. Geek the planet. [En línea] [Citado el: 10 de 5 de 2012.] <http://geektheplanet.net/5462/patrones-gof.xhtml>.
40. MeRinde V1.1.0 . [En línea] [Citado el: 10 de Abril de 2012.] merinde.net/index.php?option=com_content&task=view&id=495&Itemid=291.
41. Letelier Torres, Patricio. Desarrollo de software orientado a objetos usando UML. Valencia : Universidad Politécnica de Valencia, 2007.
42. Scripting News. [En línea] XML-RPC, 2004-2011 . [Citado el: 25 de 04 de 2012.] <http://xmlrpc.scripting.com/>.

Glosario de términos.

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

Actores: toda entidad externa al sistema o negocio, que guarda una relación con este y que demanda el uso de alguna de sus funcionalidades.

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

Clase: una clase representa un conjunto de objetos que tienen una estructura, un comportamiento y unas relaciones con propiedades parecidas.

Casos de uso: es una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema.

Anexos.

Fecha: 12/10/2011

Entrevista

Realizada por:

Nombre y Apellidos	Cargo
Katia Roselló Díaz	Estudiante.

Entrevistados:

Nombre y Apellidos	Cargo
Carlos De Jesús Andrés González	Líder de Proyecto
Félix Iván Romero Rodríguez	Desarrollador
Aramis Romero Carballea	Estudiante rol desarrollador

Objetivo:

Se utiliza para obtener información sobre el funcionamiento actual de los procesos de transmisión en PRIMICIA.

Pregunta 1: ¿Cómo funciona actualmente el subsistema de transmisión de PRIMICIA?

Pregunta 2: ¿Cómo están constituidas las noticas?

Pregunta 3: ¿Qué elementos son los que se tienen en cuenta a la hora de realizar la selección de las noticias a transmitir?

Pregunta 4: ¿Qué tecnologías utilizan para realizar las transmisiones?