

UNIVERSIDAD DE LA CIENCIAS INFORMÁTICAS

# Análisis, diseño e implementación del subsistema de Transmisión de la plataforma de televisión informativa PRIMICIA versión 2.0

---

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

**Autor: Raúl Lugo Rodríguez.**

17-06-11

## **DECLARACIÓN DE AUTORÍA**

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_

\_\_\_\_\_

Firma de la autor:

Firma del tutor:

Raúl Lugo Rodríguez

Ing. Geovanys Bermúdez Valdés

## Resumen

La Plataforma de Televisión Informativa, PRIMICIA, con varios resultados alcanzados en disímiles eventos se encuentra en un proceso de constante perfeccionamiento y búsqueda de nuevas funcionalidades. Los desarrolladores de esta solución requieren, a corto plazo, dar solución a las crecientes necesidades de los clientes, en aras de alcanzar mayor inmediatez, dinamismo y personalización de la gestión informativa. La solución propuesta es una nueva versión de la Plataforma, con una arquitectura más flexible y escalable, que sea capaz de soportar las nuevas funcionalidades requeridas y la incorporación de aquellas que surjan debido al constante perfeccionamiento y evolución del sistema. Los esfuerzos de este trabajo estuvieron centrados mayormente en la comprensión de los procesos de la Plataforma de Televisión Informativa, pues fueron el punto de partida para lograr una solución adecuada para el producto PRIMICIA.

**PALABRAS CLAVES:** escalabilidad, arquitectura, PRIMICIA, escalable.

# Índice

ÍNDICE DE TABLAS Y FIGURAS .....	VI
Introducción .....	1
Capítulo 1: Fundamentación Teórica. ....	5
1.1 Introducción.....	5
1.2 Conceptos asociados al dominio del problema. ....	5
1.2.1 Televisión.....	5
1.2.2 Procesos de transmisión en canales de Televisión. ....	6
1.2.3 Teletexto .....	7
1.2.4 Aplicación de escritorio. ....	8
1.2.5 Aplicaciones Web.....	8
1.2.6 Aplicaciones multimedia.....	9
1.2.7 Software Libre.....	10
1.3 Objeto de Estudio. ....	12
1.3.1 Descripción General.....	12
1.3.2 Descripción actual del dominio del problema. ....	14
1.3.3 Situación problemática. ....	14
1.5 Análisis de soluciones existentes. ....	15
1.5.1 Ámbito nacional. ....	15
1.5.2 Ámbito internacional.....	17
1.5.3 Valoración del autor .....	20
1.6 Conclusiones Parciales. ....	21
Capítulo 2: Tendencias y tecnologías actuales. ....	22
2.1 Introducción.....	22

2.2	Caracterización de metodología de desarrollo escogida para desarrollar la solución.....	22
2.3	<i>Aplicaciones en las que se apoya para la solución:</i> .....	23
2.3.1	Lenguaje Unificado de Modelado (UML). .....	23
2.3.2	Herramienta de modelado de software.....	24
2.3.3	Lenguaje de programación.....	25
2.3.4	Framework.....	27
2.3.5	Entorno de Desarrollo Integrado (IDE). .....	30
2.4	Descripción de otras tecnologías necesarias en la propuesta.....	32
2.4.1	Soporte para el uso de XML.....	32
2.4.2	Soporte de medias .....	34
2.5	Conclusiones Parciales. ....	37
Capítulo 3: Presentación de la solución propuesta.....		38
3.1	Introducción.....	38
3.2	Modelo de Dominio.....	38
3.2.1.	Conceptos y principales eventos del entorno. ....	38
3.2.2.	Términos del Dominio. ....	39
3.3.	Especificación de los requisitos de software.....	40
3.3.1	Requerimientos Funcionales.....	41
3.3.2	Requerimientos No Funcionales .....	43
3.4	Descripción del Sistema Propuesto .....	46
3.4.1	Actores Propuestos para el Sistema. ....	46
3.4.2	Diagrama de Casos de Uso del Sistema.....	46
3.4.3	Especificación de los Casos de Uso.....	46
3.6	Conclusiones Parciales. ....	47

Capítulo 4: Construcción de la Solución Propuesta.....	48
4.1 Introducción.....	48
4.2 Modelo de Análisis. ....	48
4.2.1 Clases del Análisis.....	48
4.2.2 Diagramas de clases del análisis. ....	48
4.3 Modelo de Diseño.....	50
4.3.1 Patrones de diseño .....	50
4.3.2 Clases del Diseño. ....	51
4.4 Modelo de Despliegue. ....	55
4.5 Diagrama de Componentes. ....	55
4.6 Prueba.....	58
4.6.1 Elementos del proceso de pruebas .....	58
4.6.2 Niveles de pruebas .....	58
4.6.3 Tipos de prueba.....	59
4.6.4 Pruebas de rendimiento.....	61
4.6.5 Diseño de pruebas de rendimiento.....	62
4.6.6 Resultado de las pruebas de rendimiento .....	63
4.6.7 Pruebas funcionales.....	63
4.6.8 Diseño de pruebas funcionales.....	64
4.6.9 Resultado de las pruebas funcionales.....	66
4.6.10 Diseño de Casos de Prueba de caja blanca.....	67
4.7 Conclusiones. ....	71
Capítulo 5: Validación de la propuesta.....	72
5.1 Introducción .....	72

5.2	Planificación del criterio de expertos .....	72
5.3	Confirmación de la participación de los expertos .....	74
5.4	Elaboración del cuestionario .....	74
5.5	Análisis y procesamiento de la información .....	74
5.6	Resultados Graficados. ....	77
5.7	Conclusiones: .....	78
	Conclusiones generales.....	78
	Recomendaciones .....	79
	Bibliografía.....	80

## ÍNDICE DE TABLAS Y FIGURAS

Figura 3:	DCA - CU Transmitir. ....	49
Figura 4:	DCA - CU Verificar cambios programados. ....	49
Figura 5:	DCA - CU Gestión de información.....	50
Figura 6:	DCD - Gestión de información. ....	51
Figura 7:	DCA - CU Transmitir (1).....	52
Figura 8:	DCA - CU Transmitir (2).....	53
Figura 9:	DCA - CU Verificar cambios programados. ....	54
Figura 10:	Diagrama de Despliegue.....	55
Figura 11:	DCMP_CU Gestión de información. ....	56
Figura 12:	DCMP_CU Verificar cambios programados. ....	56
Figura 13:	DCMP_CU Transmitir. ....	57
Figura 14:	Grafo de flujo de datos Caso 1.....	68
Figura 15:	Grafo de flujo de datos Caso 2.....	70

Tabla 1: Requerimientos de hardware .....	45
Tabla 2: Requerimientos de hardware del Servidor de Administración.....	45
Tabla 3: Requerimientos de hardware del Servidor de Transmisión.....	45
Tabla 4: Niveles de consumo de memoria y de la CPU para la estación de trabajo 1.....	62
Tabla 5: Niveles de consumo de memoria y de la CPU para la estación de trabajo 2.....	63
Tabla 6: Escenarios a probar en el caso de uso Transmitir.....	65
Tabla 7: Escenarios a probar en el caso de uso Administrar información.....	65
Tabla 8: Descripción de variables para el caso de uso Transmitir.....	66
Tabla 9: Descripción de variables para el caso de uso Administrar información.....	66
Tabla 10: Criterios de expertos .....	76
Tabla 11: Rj y Cc .....	76

# Introducción

De forma vertiginosa ha ido avanzando el mundo la tecnología y con ella los medios de comunicación, permitiendo que las informaciones sean transmitidas por todo el planeta con una alta velocidad. Estos medios desde sus orígenes han formado parte de las preferencias de la humanidad, debido principalmente al aporte de los mismos al conocimiento del entorno social y opiniones de temas de la actualidad. También permiten modificar la opinión pública sobre los temas actuales, y establecer nuevas formas de vínculo y relaciones sociales.

En la actualidad, las actividades humanas más cotidianas están siendo influenciadas por una revolución significativa que tiene como base las Tecnologías de la Información y las Comunicaciones, innovaciones que favorecen enormemente el flujo de información y que, por supuesto, mejoran también en gran medida las posibilidades de comunicación humana. El aumento acelerado de Internet hace que el caudal de información al cual se tiene acceso sea inmenso.

En los países desarrollados los medios de comunicación masiva se han adaptado a su propio entorno, manteniéndose en menor medida las transmisiones libres. Esto determina que la mayor parte del contenido que se emite valla dirigido al consumismo, provocando que gran cantidad de información sea poco relevante para quienes la ven y por tanto no transforme la conducta de los individuos.

En Cuba desde el año 1959 con el triunfo de la Revolución, los medios de comunicación han adquirido un carácter gratuito, divulgando solamente contenido social y educativo, con el objetivo de contribuir a la formación cultural y educativa de los ciudadanos. A pesar del bloqueo económico impuesto por más de 50 años, Cuba no se mantiene exenta de los adelantos en el mundo de la tecnología de la información.

Los medios de comunicación masivos, han evolucionado rápidamente sus métodos tradicionales hasta adquirir una presencia digital a través de las redes de computadoras. En medio de esto se crea la Plataforma de Televisión Informativa, PRIMICIA. La plataforma es un producto informático creado en el 2005 por la Universidad de las Ciencias Informáticas(UCI), específicamente en el seno del Polo productivo de Video y Sonido Digital de la Facultad 9, hoy perteneciente al centro de desarrollo GEYSED de la facultad 6. PRIMICIA ha permanecido en constante evolución constituyendo una solución integral capaz

de proveer un canal de televisión para la transmisión automática y constante de informaciones en distintos formatos.

PRIMICIA surge de la generalización de un conjunto de soluciones informáticas creadas para la transmisión de noticias. Sus principales antecedentes son Señal 3 (sistema informativo creado para la red de televisión interna de la UCI), Señal ACN (sistema para la transmisión de noticias a los colaboradores cubanos en el exterior y habitantes de las zonas de silencio de la geografía cubana) y TV Energía (desarrollado para mantener informados a los trabajadores y visitantes de la sede central del Ministerio del Poder Popular para la Energía y Petróleo de Venezuela (MENPET)). La plataforma permite la administración y transmisión de noticias de un canal de televisión informativo.

PRIMICIA esta soportada por herramientas para el desarrollo de aplicaciones web. Esto permite una gran interacción del usuario con la aplicación y ciertas flexibilidades dado sus características. Sin embargo constituye una desventaja al hacerse un uso insuficiente de las animaciones y efectos gráficos. Además la aplicación disminuye su rendimiento con el uso del navegador, que es el recurso utilizado por este tipo de aplicaciones para su visualización.

Como consecuencia de lo antes mencionado y teniendo en cuenta la inserción de nuevos requisitos identificados, se concibe necesario incorporarle al producto PRIMICIA, las potencialidades que proponen las tecnologías de escritorio en el manejo de la transmisión de las noticias. Es por ello que se determinó el siguiente **problema a resolver**: ¿Cómo dar solución a las deficiencias presentes en el subsistema de transmisión de la Plataforma de Televisión Informativa, Primicia?

Atendiendo al problema planteado y sobre la base de la necesidad de investigación en el marco de esta situación, se tiene como **objeto de estudio** los procesos de la Plataforma de Televisión Informativa, PRIMICIA.

El **campo de acción** son las tecnologías empleadas en el subsistema de transmisión de la Plataforma de Televisión Informativa, PRIMICIA.

Esta investigación tiene como **objetivo general** desarrollar un subsistema de transmisión para la Plataforma de Televisión Informativa, PRIMICIA, en su versión 2.0.

Con el desarrollo de una nueva versión del subsistema de transmisión de PRIMICIA, se eliminarán las deficiencias presentes en el producto, lo que constituye la **hipótesis** de esta investigación.

En función del cumplimiento del problema y para dar solución al objetivo general se deben realizar las siguientes **tareas de la investigación**:

- Caracterización de la Plataforma de Televisión Informativa, Primicia.
- Definición de los requisitos funcionales y no funcionales del sistema.
- Caracterización de las herramientas y metodologías para el desarrollo.
- Modelación del sistema haciendo uso de una herramienta de modelado.
- Implementación de las funcionalidades diseñadas.
- Validación de la propuesta.

Los **posibles resultados** a obtener son:

- Artefactos vinculados a las metodologías utilizadas.
- Código fuente de la aplicación.

Para cumplir con las tareas del trabajo se utilizaron los siguientes métodos científicos.

Para cumplir con las tareas investigativas del trabajo se utilizaron los siguientes **métodos científicos**:

- Dentro de los métodos científicos generales, los **métodos teóricos** crean las condiciones para ir más allá de las características fenoménicas y superficiales de la realidad, permiten explicar los hechos y profundizar en las relaciones esenciales y cualidades fundamentales de los procesos y fenómenos. En esta investigación se emplean los métodos:
  - **Análisis Histórico-Lógico:** Con este método se realizó un estudio de los sistemas de transmisión en el pasar de los años. De esta manera se puede determinar una tendencia del uso de las tecnologías, para determinar cuáles serán las empleadas en la construcción del subsistema de transmisión de noticias como una aplicación de escritorio.
  - **Analítico-Sintético:** Este método permitió el análisis de los procesos de transmisión de noticias que presentan diversos sistemas informativos, para de esta forma

encontrar las características más idóneas que serán incorporadas a la Plataforma de Televisión Informativa.

- **Inductivo–deductivo:** Permitió llegar a un grupo de conocimientos generalizadores acerca de los posibles elementos a incluir en la aplicación, así como determinar aquellos procesos más específicos implicados en los nuevos cambios que se necesitan para una correcta transmisión.
- Para las operaciones prácticas con el objeto y con el fin de adquirir información procedente fundamentalmente de la experiencia, se emplean los **métodos empíricos:**
- **Observación:** Es método históricamente a constituido la forma básica de obtener información científica, se utiliza a la hora de recopilar información acerca del comportamiento funcional de PRIMICIA en los actuales ambientes de desarrollo.
  - **Observación bibliográfica:** Es utilizado en todo la investigación pues con él se recoge información de los aspectos tratados en la tesis desde el punto de vista de otros autores así como sus definiciones y resultados para permitir realizar un análisis del arte en ese aspecto.

# Capítulo 1: Fundamentación Teórica.

## 1.1 Introducción.

El objetivo esencial de este capítulo es abordar los conceptos que van a servir como soporte teórico al problema científico. También se realiza una caracterización de la Plataforma de Televisión Informativa, PRIMICIA, así como un análisis de soluciones existentes (más conocido como estado de arte o del conocimiento) a nivel internacional, nacional y de la Universidad.

## 1.2 Conceptos asociados al dominio del problema.

### 1.2.1 Televisión.

Desde los años 50, la televisión se ha convertido en el medio de comunicación por excelencia. La prehistoria de este medio comienza a partir de los descubrimientos técnicos más elementales, que hicieron posible la transmisión y recepción de imágenes en movimiento y sonido a distancia. Durante el siglo pasado surgen los dos primeros modelos inventados por el hombre: la televisión mecánica y la electrónica.

Las primeras transmisiones experimentales dieron a luz en Estados Unidos. Fue en Julio de 1928 cuando desde la estación experimental W3XK de Washington, se comenzó a transmitir imágenes exploradas de películas. Las primeras emisiones públicas de televisión las efectuó la BBC<sup>1</sup> en Inglaterra en 1927 y la CBS<sup>2</sup> y NBC<sup>3</sup> en Estados Unidos en 1930. En ambos casos se utilizaron sistemas mecánicos y los programas no se emitían con un horario regular. Las emisiones con programación se iniciaron en Inglaterra en 1936 (1).

Es a mediados del siglo XX donde la televisión se convierte en bandera tecnológica de los países europeos y cada uno de ellos va desarrollando sus sistemas de TV nacionales y privados. En 1953 se

---

<sup>1</sup> La British Broadcasting Corporation (Corporación Británica de Radiodifusión), más conocida como BBC, es la primera empresa de televisión y radio del Reino Unido. También es la cadena más grande de transmisión de noticias en el mundo.

<sup>2</sup> CBS (*Columbia Broadcasting System*) es una cadena estadounidense de radio y televisión. Está constituida por varias divisiones: televisión, radio, noticiarios, deportes y entretenimientos.

<sup>3</sup> La National Broadcasting Company, Incorporated, es una empresa estadounidense de medios de comunicación.

crea Eurovisión<sup>4</sup> que asocia a varios países de Europa conectando sus sistemas de TV mediante enlaces de microondas<sup>5</sup>. Unos años más tarde, en 1960, se crea Mundovisión<sup>6</sup> que comienza a realizar enlaces con satélites geoestacionarios<sup>7</sup> cubriendo todo el mundo (2).

## 1.2.2 Procesos de transmisión en canales de Televisión.

El proceso de transmisión en canales de Televisión puede ser de forma tradicional o automatizada. La tradicional consiste en operar un *switch*<sup>8</sup> que recibe señales de video<sup>9</sup> de distintas fuentes. Quien realiza las operaciones se guía por una escaleta, que no es más que un esquema o esqueleto del video que ayuda a organizar el plan<sup>10</sup> de grabación<sup>11</sup> y facilita la escritura del pre-guión y del guión<sup>12</sup> definitivo, permitiendo así especificar el orden en que se mostrará cada espacio y el tiempo de duración. Gracias al avance de la informática, tanto como del *hardware*<sup>13</sup> como del *software*<sup>14</sup> se ha mostrado la posibilidad de crear sistemas para automatizar el proceso de transmisión. Esto ha posibilitado una mejora de:

- ➔ La gestión de la escaleta.
- ➔ El control total de las señales recibidas.
- ➔ La titulación y gráficos de la transmisión.
- ➔ La reproducción de videos desde disco duro en formatos como MPEG<sup>15</sup>, AVI<sup>16</sup> entre otros.

---

<sup>4</sup> Unión Europea de Radiodifusión, también llamado UER o EBU: organismo europeo que aglutina a las emisoras de radio y televisión del continente y su zona de influencia.

<sup>5</sup> Ondas electromagnéticas definidas en un rango de frecuencias determinado.

<sup>6</sup> Red satelital mundial creada en 1960.

<sup>7</sup> Los satélites geoestacionarios están situados exactamente sobre el ecuador de la Tierra y giran alrededor de la Tierra en una órbita circular.

<sup>8</sup> Dispositivo de conmutación que permite el control de distintos equipos.

<sup>9</sup> El vídeo o video es la tecnología de la captación, grabación, procesamiento, almacenamiento, transmisión y reconstrucción por medios electrónicos digitales o analógicos de una secuencia de imágenes que representan escenas en movimiento.

<sup>10</sup> Un plan suele referirse a un programa o procedimiento para conseguir un determinado objetivo.

<sup>11</sup> Proceso de capturar datos o convertir la información a un formato almacenado en un medio de almacenamiento.

<sup>12</sup> Un guión es un texto en que se expone, con los detalles necesarios para su realización, el contenido de una película, historieta o de un programa de radio o televisión.

<sup>13</sup> Conjunto de los componentes que integran la parte material de una computadora.

<sup>14</sup> Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

<sup>15</sup> Grupo de trabajo del ISO/IEC encargado de desarrollar estándares de codificación de audio y vídeo.

<sup>16</sup> Formato contenedor de audio y video lanzado por Microsoft en 1992.

### 1.2.3 Teletexto

Es un medio de comunicación gratuito con vocación de servicio público que se sirve de la televisión para hacer llegar sus mensajes a los al usuario. En el conviven contenidos editoriales y comerciales, ofreciéndole al receptor una información completa de toda índole y fácil de consultar. Se podría comparar con un gran periódico de más de 2.000 páginas que el usuario puede seleccionar y leer cuantas veces desee. Su característica principal es que, a diferencia del periódico tradicional, su información está en constante renovación.

El teletexto nace gracias a la iniciativa del ingeniero de la BBC Peter Rainger. Desde 1970, Rainger buscaba un sistema que emitiera rótulos o subtítulos para ser leídos por personas con discapacidad auditiva en sus televisores. En 1972, Sir James Redmond, director de Ingeniería de la BBC, resolvió que este invento podría ser incorporado en todos los aparatos, con lo que además de ayudar a la comunidad sorda, se creaba un nuevo medio de comunicación. Para que la idea fructificase debían implicarse los fabricantes de televisores, cosa que no tardó en llegar.

Las primeras emisiones de teletexto las realizó la BBC en 1973. Al principio, tan sólo se emitían frases intemporales, pero pronto la cadena británica apostó por formar un equipo que se encargara de proporcionar noticias y todo tipo de contenidos a través de este sistema. Para poder recibir la señal era necesario invertir en un decodificador, lo que impidió el desarrollo del teletexto, aunque cada vez eran más las personas que se fiaban del recién nacido medio.

Los usos y las posibilidades del teletexto son incontables. En principio, todo aquello que sea susceptible de despertar el interés del usuario tiene cabida en este medio (3). Se define específicamente como teletexto el sistema que permite la transmisión de información textual utilizando la propia señal de televisión convencional.

Un término muy asociado al teletexto es el servicio “videotexto”, el cual es definido por la Real Academia Española como un sistema de comunicación que transmite a una pantalla textos informatizados a través de una señal telefónica o televisiva. Otras definiciones de este término aportan elementos que lo asocian directamente a un servicio unidireccional para la transmisión texto a través de la televisión (4).

### 1.2.4 Aplicación de escritorio.

Para la informática, una aplicación es uno de diversos tipos de programas de computación diseñados especialmente para cumplimentar una función o actuar como herramienta para acciones puntuales del usuario. A diferencia de otros programas como los sistemas operativos, los lenguajes de programación y otros, la aplicación tiene el único y principal fin de realizar una tarea específica, a menudo básica y de rápido y fácil uso para el usuario común no avanzado (5).

Las aplicaciones de escritorio son creadas para ejecutarse en un ordenador, sobre una plataforma o sistema operativo de interfaz. Sus principales desventajas son:

- ➔ El desarrollo sobre una plataforma regularmente implica que la aplicación "no" pueda ser ejecutada en otras.
- ➔ Su rendimiento está en dependencia de las configuraciones de hardware como memoria RAM<sup>17</sup>, y memoria de video.

A pesar de sus dependencias, este tipo de aplicación brindaría a la Plataforma de Televisión Informativa ventajas como:

- ➔ Mayor capacidad gráfica visual.
- ➔ Menor tiempo de respuesta (aplicación más rápida).
- ➔ Mayor personalización.

### 1.2.5 Aplicaciones Web.

En la ingeniería software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web, y en la que se confía la ejecución de la aplicación al navegador (6).

Hoy por hoy gozan de gran popularidad por su práctico uso, al permitir una comunicación más fluida y dinámica en la computación Cliente-Servidor. El alto grado de desarrollo actual permite la actualización y

---

<sup>17</sup> La memoria de acceso aleatorio (en inglés: *random-access memory*, cuyo acrónimo es RAM) es la memoria desde donde el procesador recibe las instrucciones y guarda los resultados.

el mantenimiento (vía Internet) de dichas Aplicaciones, sin que se deba distribuir e instalar software específico o versiones individuales a un usuario cada vez.

Una página Web puede contener elementos que permiten una comunicación activa entre el usuario y la información (vía Servidor); logrando el usuario acceso a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones.

Las aplicaciones web generan dinámicamente una serie de páginas en un formato estándar, como HTML<sup>18</sup> o XHTML<sup>19</sup>, que soportan por los navegadores web comunes. Se utilizan lenguajes interpretados en el lado del cliente, tales como JavaScript<sup>20</sup>, para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página web en particular se envía al cliente como un documento estático, pero la secuencia de páginas ofrece al usuario una experiencia interactiva.

Los principales inconvenientes de este tipo de aplicación, es que habitualmente brindan menos funcionalidades que las aplicaciones de escritorio. Esto se debe a que las funcionalidades que se pueden realizar desde un navegador son más limitadas que las que se pueden realizar desde el sistema operativo. Por otra parte la disponibilidad depende de un tercero, el proveedor de la conexión a internet o el que proporciona el vínculo entre el servidor de la aplicación y el cliente. Así que la disponibilidad del servicio está sujeta al proveedor. También es cierto que carece de la riqueza gráfica de las aplicaciones de escritorio que cuentan con controles inteligentes que dan mayor fluidez al trabajo del usuario (7).

Mediante el desarrollo de una aplicación web y gracias a sus características, fue posible obtener un sistema de administración para la Plataforma de Televisión que posibilita de manera eficiente las diferentes funcionalidades requeridas.

### 1.2.6 Aplicaciones multimedia

Es cualquier combinación de texto, arte gráfico, sonido, animación y vídeo que llega a nosotros por computadora u otros medios electrónicos (8).

---

<sup>18</sup> HyperText Markup Language (*Lenguaje de Marcado de Hipertexto*), es el lenguaje de marcado predominante para la elaboración de páginas web.

<sup>19</sup> Acrónimo en inglés de eXtensible Hypertext Markup Language (lenguaje extensible de marcado de hipertexto)

<sup>20</sup> Javascript es un lenguaje de programación interpretado dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Las posibilidades de aplicar el concepto multimedia a la vida cotidiana del hombre son muy numerosas y la mayoría de ellas se encuentran en vías de desarrollo. Las áreas en las que la multimedia cobra más importancia son: la educación, la cultura y el ocio, sin dejar de lado las aplicaciones profesionales. En el ámbito educativo, estas aplicaciones ofrecen la posibilidad de ampliar las características favorables de los ordenadores en dicha área. La utilización las mismas, unida a la propiedad de interacción de los ordenadores, convierten a estos últimos en elementos muy importantes en el desarrollo cognitivo de los individuos (9).

De acuerdo a lo anterior se puede definir a un sistema automatizado de televisión aquel que integre en su visualización distintos soportes de comunicación como texto, imagen, video y sonido como una aplicación multimedia. De esta manera se plantea el desarrollo de un sistema multimedia para la transmisión de las informaciones a través de la televisión.

### 1.2.7 Software Libre.

Software libre es aquel que se puede ejecutar, copiar, distribuir, estudiar, cambiar y mejorar. Las cuatro libertades esenciales que tienen los usuarios que lo poseen son:

- La libertad de usar el programa, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a tus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias del programa, con lo cual puedes ayudar a tu prójimo (libertad 2).
- La libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie (libertad 3). (10)

Este tipo de software tiene licencias, donde la más utilizada es la GPL<sup>21</sup>. La misma establece entre otras reglas que la modificación o extensión del software debe ser bajo la misma licencia, donde esto implica que si en un proyecto se combinan códigos con alguna otra licencia, ésta desaparecería y la licencia resultante sería la GPL.

---

<sup>21</sup> La Licencia Pública General de GNU o más conocida por su nombre en inglés *GNU General Public License* o simplemente sus siglas del inglés GNU GPL, es una licencia creada por la Free Software Foundation en 1989 (la primera versión), y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

Software libre no significa que no sea comercial. Un programa libre debe estar disponible para el uso comercial, la programación comercial y la distribución comercial. La programación comercial de software libre ya no es inusual; tal software libre comercial es muy importante. Puede haber pagado dinero para obtener copias de software libre, o puede haber obtenido copias sin costo. Pero sin tener en cuenta cómo obtuvo sus copias, siempre tiene la libertad de copiar y modificar el software, incluso de vender copias. (11)

Una vez comprendidas en cierta medida las características del software libre se puede concluir en algunas de sus ventajas:

- Tiende a ser muy eficiente y robusto, pues mucha gente puede arreglarlo, optimizarlo, mejorarlo.
- Tiende a ser muy diverso: cada persona que contribuye tiene necesidades diferentes y esto hace que el software sea adaptado a una cantidad más grandes de problemas.
- Se logra tener un mayor control sobre el software al contar con el código fuente, lo que se puede traducir en un mayor control sobre la introducción de código malicioso, un rápido desarrollo y una alta calidad del producto.
- Permite que el número de personas que pueden utilizar el software se maximice a través de la traducción del software en cualquier lenguaje.
- El desarrollo y soporte de un software puede ser continuado aún después que se creador haya abandonado el proyecto, ya sea una persona o una institución, lo que garantiza la durabilidad de las soluciones. (12)

Algunas de las desventajas más significativas son:

- Por lo general el software libre no tiene garantías provenientes del autor.
- Se necesita dedicar recursos a la reparación de errores. Sin embargo en el software propietario es imposible arreglar errores y hay que esperar a que saquen a la venta otras versiones.
- La mayoría de las configuraciones de hardware no son intuitivas, se requieren conocimientos previos acerca del funcionamiento del sistema operativo y fundamentos del equipo a conectar para lograr un funcionamiento adecuado.

- La incompatibilidad entre tipos de archivos a los que no se puede tener acceso, patentados por licencias no libres, como los que siguen los estándares de Microsoft (12).

La gran diversidad de software libre es considerada una ventaja para muchos, para otros una desventaja. La realidad es que se pueden encontrar soluciones de todo tipo para pequeños dispositivos móviles o destinadas a cubrir necesidades de alto procesamiento y almacenamiento. A veces la interacción con los sistemas operativos libres se hace difícil. Existen sistemas fácilmente maniobrables por el usuario estándar y sistemas avanzados que requieren de gran adiestramiento.

Cuba ha apostado en los últimos años por la utilización de tecnologías y plataformas libres para el desarrollo de la industria cubana del software. Todo ello con vistas a lograr de esta manera una soberanía tecnológica en este sentido, evitar la compra de costosas licencias que imponen los monopolios informáticos y que en muchas ocasiones le es vetada la compra de las mismas a causa del bloqueo económico al que se ha visto sometida por más de 50 años. Son estas las principales razones por lo que se plantea el desarrollo de la solución integral completamente sobre plataformas libres.

## **1.3 Objeto de Estudio.**

### **1.3.1 Descripción General.**

La Plataforma de Televisión Informativa PRIMICIA es un producto informático creado en el año 2005 por la Universidad de las Ciencias Informáticas UCI en el Polo productivo de Video y Sonido Digital, actualmente Centro de Desarrollo GEYSED<sup>22</sup>. El sistema se orienta a proveer un canal de televisión básicamente con características informativas, donde se integran contenidos de texto, imagen fija y video.

La plataforma constituye una solución integral que permanece en constante evolución. Cuenta con funcionalidades genéricas fácilmente escalables, que no dependen de un entorno dado y que no están ligadas a un diseño gráfico específico. El sistema se encuentra estructurado en dos subsistemas que se relacionan entre sí y actúan como un todo para brindar un resultado final. El subsistema de administración es el responsable de que a través de él se realice la administración del canal y toda la gestión de las

---

<sup>22</sup> Geoinformática y Señales Digitales.

noticias y recursos. Por otra parte el de transmisión es el encargado de visualizar las noticias y materiales publicados.

**El subsistema de administración tiene las siguientes prestaciones generales:**

- Gestión de los usuarios del sistema que permitan registrar y eliminar usuarios, así como establecer o modificar los permisos de acceso en el sistema.
- Gestión de las sesiones temáticas del canal que admitan establecer orden de las secciones, horario en que serán mostradas y habilitarlas o deshabilitarlas.
- Funcionalidades para la redacción de noticias según los formatos definidos para las pantallas; la publicación de las noticias teniendo en cuenta fecha de inicio y fin de la publicación; la gestión de las noticias del canal que permitan modificar, eliminar y archivar las noticias; la administración del archivo de noticias del canal que permitan reutilizar y eliminar las mismas.
- Almacenamiento, administración y reproducción de recursos multimedia como imágenes, música y video.
- Funcionalidades para la creación y administración de cintillos informativos o infocintas. La administración de los cintillos incluirá establecer orden de prioridad de muestra y la habilitación o deshabilitación de los mismos.
- Generación de reportes sobre la actividad del sistema. Los reportes se realizarán sobre la actividad de los trabajadores del sistema, realizar búsquedas de noticias publicadas atendiendo distintos criterios como fecha de publicación, temática, palabras claves y título. Ofrece facilidades para la impresión de los reportes y la exportación de los reportes a formato digital.
- Administración de la señal del canal que permite cambiar entre la señal de televisión en vivo y la señal del canal.
- Administración del diseño gráfico del canal permitiendo realizar cambios en las distribuciones gráficas del sistema informativo de televisión. (12)

### **Prestaciones del subsistema de transmisión:**

- Generar una cartelera del ciclo de transmisión, mostrando para cada noticia la sección temática y el titular, en el orden que se van a visualizar.
- Visualizar noticias compuestas por pantallas de tipo Texto, Texto-Imagen, Imagen y Video.
- Reproducir un fondo musical mientras se muestran las noticias, excepto cuando se muestra un video.
- Mostrar en las pantallas de tipo Imagen un comentario que orienta al televidente acerca de lo que está observando.
- Mostrar cintillos informativos o infocintas para promocionar eventos de última hora o acontecimientos de gran importancia.
- Mostrar como información adicional a la noticia: la fecha, hora, tiempo restante de la pantalla, el titular de la próxima noticia y la sección temática.
- Transmitir televisión en vivo proveniente de una señal externa.
- Mostrar el patrón del canal cuando éste se encuentre fuera de servicio (4).

### **1.3.2 Descripción actual del dominio del problema.**

### **1.3.3 Situación problemática.**

La Plataforma de Televisión Informativa PRIMICIA surge de la generalización de un conjunto de soluciones informáticas creadas para la transmisión de noticias. Sus principales antecedentes son Señal 3<sup>23</sup>, Señal ACN<sup>24</sup> y TV Energía<sup>25</sup>. La plataforma permite la administración y transmisión de noticias de un canal de televisión informativo.

El desarrollo total del sistema se realizó utilizando software libre, empleando e integrando las siguientes herramientas y lenguajes de programación:

- Sistema Operativo: NOVA/LINUX.

---

<sup>23</sup> Sistema informativo creado para la red de televisión interna de la UCI.

<sup>24</sup> Sistema para la transmisión de noticias a los colaboradores cubanos en el exterior y habitantes de las zonas de silencio de la geografía cubana.

<sup>25</sup> Sistema desarrollado para mantener informados a los trabajadores y visitantes de la sede central del Ministerio del Poder Popular para la Energía y Petróleo de Venezuela.

- ➔ Herramienta de Gestión de Proyecto: dotProject.
- ➔ Herramienta de Control de Versiones: Subversion.
- ➔ Herramienta de Modelado: Visual Paradigm.
- ➔ Entorno de Desarrollo Integrado (IDE): Eclipse PDT.
- ➔ Framework de Desarrollo: Symfony.
- ➔ Sistema Gestor de Base de Datos: PostgreSQL.
- ➔ Lenguajes de Programación: PHP, JavaScript, HTML. (2)

La Plataforma está soportada con herramientas para el desarrollo de aplicaciones web que permiten ciertas flexibilidades. Sin embargo constituye una desventaja para el subsistema de transmisión al poner lenta la aplicación y disminuir su propio rendimiento con la utilización del navegador, que es el recurso utilizado por este tipo de aplicaciones para su visualización. No obstante, las tecnologías de escritorio permiten dar una mejora, mediante las potencialidades que brindan, posibilitando un eficiente desempeño tanto de hardware como de software.

## **1.5 Análisis de soluciones existentes.**

En disímiles universidades, centros de convenciones, hoteles, terminales de transportación, sedes ministeriales o empresariales en las que existen circuitos cerrados de televisión se brindan informaciones a través de sistemas de teletexto. Ejemplo de soluciones son los canales informativos en el ámbito nacional, Señal 3 de la Universidad de las Ciencias Informáticas, Señal ACN de la Agencia Cubana de Noticias y la Plataforma de Televisión Informativa PRIMICIA como una conceptualización de todos estos. Estos sistemas tienen como elemento común la posibilidad de brindar información a los usuarios las 24 horas del día, utilizando el empleo de imágenes y videos en la mayoría de los casos.

### **1.5.1 Ámbito nacional.**

Cuba cuenta con antecedentes a la solución que se formula, las cuales son Señal 3 de la Universidad de las Ciencias Informáticas, Señal ACN de la Agencia Cubana de Noticias, TV Energía (MENPET) y el sistema El Viajero de la terminal de ómnibus ASTRO.

Señal 3 es un sistema diseñado para transmitir las 24 horas, el acontecer de la Universidad de las Ciencias Informáticas. Fue la base para la construcción del sistema Señal ACN. Está compuesto por dos

módulos: administración y transmisión y estos por secciones. Las noticias están agrupadas por distintos tipos como: solo texto, texto e imagen, solo imagen, video y animación flash. Esta aplicación satisface en parte el problema presente, tiene características aproximadas a lo que se desea como son: su fácil gestión a través de la web, transmisión de texto y recursos multimedia por medio de la televisión, organización por secciones, empleo de múltiples tipos de noticias y un fondo musical. Sin embargo, la presencia de algunas particularidades en el problema trazado imposibilita resolver el problema planteado con el apoyo de este sistema, pues Señal 3:

- No está encaminado solo hacia la reproducción de noticias, sino incluye también informaciones que facilitan socialmente la comunidad universitaria, como pérdida, felicitaciones y convocatorias.
- El diseño gráfico del canal no coincide con los estándares requeridos para el lenguaje televisivo.

Señal ACN de la Agencia Cubana de Noticias. Su principal objetivo es la transmisión de información nacional a los cooperantes cubanos que cumplen misión en el extranjero y a los cubanos que residen en lugares inaccesibles de la geografía cubana y reciben la señal de Televisión Satelital cubana. Al igual que Señal 3 esta aplicación está desarrollada completamente utilizando software propietario (Windows, ASP, SQL, Macromedia Director) y ninguna de las dos permite realizar transmisiones en vivo.

El Viajero de la terminal de ómnibus ASTRO es un sistema que brinda información a todas las personas que se transportan de un lugar a otro en el país. Está desarrollado con las siguientes tecnologías: C# como lenguaje de programación, UML como Lenguaje Unificado de Modelado, RUP como metodología de desarrollo de software, como herramienta de modelado Rational Rose, MySql como Sistema Gestor de Base de Datos y como Entorno Integrado de desarrollo IDE Microsoft Visual Studio 2005. Este al igual que Señal 3 y Señal ACN, fue creado con la utilización de tecnologías propietarias no compatibles con el sistema a desarrollar, ya que se desea una aplicación que permita la transmisión de información y que esté desarrollado sobre tecnologías libres. Además hace solamente empleo de pantallas que pueden contener texto e imágenes.

La Plataforma de Televisión Informativa PRIMICIA proporciona un canal de televisión que es soportado y transmitido utilizando computadoras. La solución ha sido pensada para que sea adaptable a distintos

entornos. Todo el desarrollo del sistema se enfocó hacia la obtención de funcionalidades genéricas fácilmente escalables que no dependan de un entorno dado y no atadas a un diseño gráfico específico. Un elemento adicional es que en el estudio de requerimientos de los clientes se puede acordar la personalización o eliminación de alguna funcionalidad (4). El desarrollo del sistema se realizó completamente utilizando software libre. La versión presente de la plataforma es la misma que la actual de TV Energía (MENPET).

Estos sistemas tienen particulares que se acercan a lo anhelado: transmisión de texto a través de la televisión, uso de recursos multimedia, organización por secciones temáticas, empleo de diferentes tipos de pantallas para las noticias y un fondo musical. Sin embargo, solo se tomara como apoyo la versión actual de Primicia, siendo esta la base fundamental de estudio.

### **1.5.2 Ámbito internacional.**

A nivel mundial es escasa la información encontrada sobre sistemas afines al que se desea. Se localizan incontables sistemas para la transmisión de información, pero en la mayoría de los casos estos procesos no se corresponden con el entorno televisivo. En las búsquedas de información se hallaron algunos sistemas de teletexto<sup>3</sup> que integran herramientas para la transmisión.

La inmensa mayoría de los Sistemas de teletextos existentes en el mundo se concentran situados en Europa. Entre los hallados que ofrecen información técnica sobre sus soluciones se encuentran “Fingertext” y “Cyclone”, dos plataformas de procedencia española e inglesa respectivamente. A continuación se elabora un análisis de Fingertext teniendo en cuenta que existe una abundante documentación sobre sus funcionalidades y que son similares a las de Cyclone.

Fingertext es un sistema modular de teletexto y subtítulos por teletexto, el cual permite una amplia gama de configuraciones. Debido a su bajo coste de explotación y mantenimiento (automatismos) permite una recuperación de la inversión de forma prácticamente inmediata. Incorpora, además, servicios que se reutilizarán en la Tv Digital. Fue desarrollado por la empresa Anglatècnic S.L., fundada en enero de 1999 por Enric Torres i Feixas. Anglatècnic está formada por un equipo de Ingenieros de Telecomunicaciones e Ingenieros Informáticos que se dedican a la realización de proyectos en los sectores de Emisoras de Televisión y Tecnologías de la Información. (13)

### **Algunas de las tecnologías empleadas en su construcción:**

- JavaScript: Son scripts incluidos en el código HTML de las páginas que interpreta el navegador. Muy adecuadas para la validación de datos de formularios o para comunicarse con Applets de Java. Anglatècnic lo utiliza en Fingertext para validar los datos de los formularios.
- Applets Java: Son aplicaciones que ejecuta el navegador programadas en lenguaje Java mediante estos se puede acceder a ficheros, bases de datos, crear aplicaciones gráficas con animaciones etc. Anglatècnic aprovecha su potencialidad para crear en Fingertext banners interactivos, gráficos dinámicos, programas dentro de páginas web, etc.
- Flash: La tecnología Flash permite crear páginas web interactivas, introducir animaciones en ellas y todo esto permite al Flash crear aplicaciones sofisticadas pero con interfaces agradables que cautiven fácilmente al usuario. En Fingertext se utiliza esta tecnología para presentar ejercicios en cursos interactivos a través de Internet.
- PHP: Permiten independizar el diseño de la página de la programación. Está especialmente indicado para trabajar conjuntamente con una base de datos que contiene los datos que se quieren presentar. Anglatècnic se apoya en este lenguaje para realizar en Fingertext las oficinas virtuales, redacciones virtuales, etc.
- Librerías PHP: En el lenguaje de programación PHP se pueden utilizar librerías que facilitan tareas de desarrollo. Anglatècnic emplea estas colecciones para generar dinámicamente imágenes de páginas del teletexto de las emisoras de televisión, mapeadas adecuadamente a fin que el usuario pueda saltar a las otras páginas indicadas.
- XML y XSL: Con XML se puede estructurar fácilmente la información que se quiere presentar. Mediante páginas XSL se puede presentar la información almacenada en formato XML en el formato que más se desee, HTML, texto, wml (WAP), etc. De esta forma se consigue total independencia entre el diseño de las páginas y los datos que se publican. El grupo de expertos de Anglatècnic utiliza esta tecnología para publicar en Fingertext datos meteorológicos. (2)

### **Características del sistema Fingertext:**

- Permite que los usuarios se autentiquen a través de la web. Verificando que sólo puedan realizar las tareas permitidas según su perfil. El cual define las carpetas y contenidos sobre los que puede actuar y cuales son las operaciones que puede realizar. Estas carpetas tendrán las páginas y subtitulaciones que posteriormente serán mostrados en los receptores.
- Brinda la posibilidad de la administración y gestión del sistema completo desde cualquier ubicación física que tenga acceso al servidor.
- Permite la creación de las páginas, utilizando para ello un editor de teletexto, el cual permite que estas sean creadas y editadas (cuando se habla de página se está haciendo alusión al teletexto que será mostrado en los televisores). Este software utilizado para la creación de los teletextos da la posibilidad de utilizar todas las características de los estándares establecidos internacionalmente. Además permite la configuración a nivel de página y de subpágina.
- Este sistema posibilita crear dos tipos de subtitulaciones: la directa y la diferida. Para realizar algunas de estas se utilizan diferentes softwares especializados en dependencia del tipo.
- El sistema además da la posibilidad a los usuarios con ciertos privilegios de editar las páginas de teletexto, poder eliminarlas, moverlas o cambiar su contenido, estas mismas operaciones pueden ser realizadas con las subtitulaciones. (14)

Estos sistemas de teletexto implementados en Europa brindan enormes ventajas desde el punto de vista de la televisión digital, organización por secciones y actualización automática mediante un software que facilita este trabajo. Sin embargo existen funcionalidades importantes en la solución a la problemática planteada que no son soportadas por estas plataformas.

### **Entre las funcionalidades no toleradas están:**

- Tienen como principio fundamental que el usuario es el encargado de decidir mediante su mando a distancia qué noticias desea ver dentro del sinnúmero de informaciones que se ofrecen hace que no se ajuste, debido que se requiere que las noticias sean transmitidas sin la necesidad de la interactividad con el usuario.
- Las noticias soportan varios formatos de pantalla (texto, imagen, texto-imagen, video). En este caso no se necesita de ningún software para realizar esta operación, todo el trabajo es realizado dentro de la misma aplicación dinámicamente.
- La emisión del propio teletexto se realiza junto con la señal televisiva haciendo uso de las llamadas líneas del intervalo de cancelación de cuadro de la señal de video, es decir, las líneas no utilizadas para la señal de imagen de televisión. Esta característica limita en gran parte la información que se puede emitir, admitiendo sólo textos y mínimamente gráficos.

Luego del anterior análisis se concluye que los sistemas de teletexto internacionales no pueden dar solución ni apoyo al problema planteado.

### **1.5.3 Valoración del autor**

En el mundo la mayor parte de los sistemas de teletexto que permiten la transmisión están realizados con tecnologías web, ya que así pueden ser accedidos desde cualquier parte y ofrecen ciertas posibilidades tanto al usuario como al creador del sistema. Actualmente la tendencia a desarrollar este tipo de sistemas con técnicas web ha adquirido un gran auge, sin embargo las de escritorio ofrecen distintas potencialidades y riquezas que no conviene ser ignoradas. Si se efectúa una indagación detallada en la red de redes, se hallará que el dictamen de numerosos especialistas entendidos del tema y otras personas en el mundo, concuerdan en su mayoría en desarrollar estos sistemas con tecnologías web.

Sin embargo se considera que el uso de una u otra tecnología no se debe específicamente a cuál de ellas es mejor para hacer estos sistemas, sino a las mejoras o beneficios que se desee brinde el sistema, pues ambas tecnologías brindan disímiles bondades. Se concluye realizar una aplicación de escritorio para PRIMICIA, ya que se quiere mejorar del sistema elementos como animación y efectos pues a través de la web este proceso se realiza más lento; así como también un mayor rendimiento al no tener que depender de un navegador para la ejecución de la aplicación.

## **1.6 Conclusiones Parciales.**

Al concluir el capítulo se logra tener un mejor entendimiento de las razones por las que se realiza el presente trabajo. Se conceptualizaron elementos teóricos que sirven como base al problema científico y objetivo del trabajo. Con la descripción general del problema científico, el análisis de la Plataforma y el estudio de otras soluciones existentes, se logra poseer un mejor entendimiento de la situación que llevó a la elaboración del presente trabajo, siendo muy útil para la comprensión de los capítulos siguientes.

# Capítulo 2: Tendencias y tecnologías actuales.

## 2.1 Introducción.

En este capítulo se realiza un análisis detallado de los principales conceptos y tecnologías que se propondrán para el desarrollo de la plataforma. Se abordan temas relacionados con el uso de tecnologías para desarrollar una aplicación de escritorio, soporte para el uso de XML, así como de medias. Es importante destacar que para la selección de estas tecnologías se tuvo en cuenta tres aspectos específicos definidos previamente: animación, efectos y rendimiento. Basado en estas tres variables, que son las que se necesitan potenciar principalmente en el subsistema de transmisión de Noticias, se procede finalmente a la selección de las mismas.

## 2.2 Caracterización de metodología de desarrollo escogida para desarrollar la solución.

Una Metodología de desarrollo de software en ingeniería de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. La metodología que se empleará para guiar el proceso de desarrollo del software es RUP<sup>26</sup>.

Rational Unified Process es una infraestructura flexible de desarrollo de software que proporciona prácticas recomendadas probadas y una arquitectura configurable. Es un Proceso Práctico (15). Junto con el Lenguaje Unificado de Modelado UML<sup>27</sup>, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es un conjunto de metodologías adaptables al contexto y necesidades de cada organización, que define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto. Como 3 características esenciales está dirigido por los Casos de Uso: que orientan el proyecto a la importancia para el usuario y lo que este quiere, está centrado en la arquitectura: que Relaciona la toma de decisiones que indican cómo tiene que ser

---

<sup>26</sup> *Rational Unified Process* en inglés, habitualmente resumido como RUP.

<sup>27</sup> LUM o UML, por sus siglas en inglés, *Unified Modeling Language*.

construido el sistema y en qué orden, y es iterativo e incremental: donde divide el proyecto en mini proyectos donde los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada (16).

### **Principales características.**

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.
- Administración de requisitos.
- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software.

RUP sigue una estructura bidimensional dividiendo el proceso en fases y éstas en flujos de trabajo. En cada una de sus fases realiza una serie de artefactos que sirven para comprender mejor tanto el análisis como el diseño del sistema.

## ***2.3 Aplicaciones en las que se apoya para la solución:***

### ***2.3.1 Lenguaje Unificado de Modelado (UML).***

El Lenguaje Unificado de Modelado es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. UML captura decisiones y conocimiento sobre los sistemas a construir, así como incluye conceptos semánticos, notación y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML posee elementos gráficos para soportar la captura de requisitos, el análisis, el diseño, la implementación, y las pruebas. Sin embargo es una notación y no un proceso o método, es decir, es una herramienta útil para representar los modelos del sistema en desarrollo, pero no ofrece ningún tipo de guía o criterios acerca de cómo obtener esos modelos. Indica qué es lo que supuestamente hará el sistema, pero no cómo lo hará. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG<sup>28</sup>. UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

### 2.3.2 Herramienta de modelado de software.

El modelado de sistemas software es una técnica para tratar con la complejidad inherente a estos sistemas. El uso de modelos ayuda al ingeniero de software a visualizar el sistema a construir. Además, los modelos a un nivel de abstracción mayor pueden utilizarse para la comunicación con el cliente. Por último, las herramientas de modelado pueden ayudar a verificar la corrección del modelo.

Para la modelamiento de la solución se escogió como herramienta de modelado de software a Visual Paradigm.

Visual Paradigm es una herramienta para la modelación UML muy potente y de fácil uso. Te permite sobre UML dibujar todo tipo de diagramas, revertir código fuente a modelos, generar código fuente desde los diagramas, y es además un software un libre. Visual Paradigm for UML incluye los objetos más recientes de UML, además de diagramas de casos de uso, diagramas de clase, diagramas de componentes, reversa instantánea para Java, C++, DotNet Exe/dll, XML, XML Schema, y Corba IDL, ofrece soporte para Rational Rose, integración con Microsoft Visio, además permite generar reportes y documentación en HTML/PDF (17).

Dentro de sus principales características de Visual Paradigm se pueden mencionar:

➤ Para UML, es una Herramienta CASE que soporta las últimas versiones de UML, así como el modelado de procesos de negocios.

---

<sup>28</sup> Object Management Group.

➔ En adición al soporte de Modelado UML esta herramienta provee de un generador de mapeo de objetos-relacionales para los lenguajes de programación Java .NET y PHP.

Para desarrolladores independientes existe una versión llamada Community Edition la que se caracteriza por ser de uso No Comercial (18).

Para maximizar la interoperabilidad de los productos de Visual Paradigm con otras aplicaciones, se le introdujo la funcionalidad de importar y exportar de modelos de proyecto desde o hasta un formato XML. Además, la herramienta es colaborativa, es decir, soporta múltiples usuarios trabajando sobre este proyecto. Permite además el control de versiones y practicar ingeniería inversa (crear modelo a partir código). Generan documentación automática como Web o PDF. Se seleccionó Visual Paradigm como herramienta CASE<sup>29</sup> por las potencialidades mencionadas.

### 2.3.3 Lenguaje de programación.

Un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana (19). Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa informático se le llama programación.

Para la implementación del subsistema de transmisión de Primicia se escoge C++ como lenguaje de programación debido a sus disimiles potencialidades ligadas a la solución del problema a resolver.

C++ es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. En principios era llamado "C con clases", ya que es una extensión de C. La intención de su creación fue extender el exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. Desde el punto de vista de los lenguajes orientados a objetos, C++ es un lenguaje híbrido. Se trata simplemente del sucesor de un lenguaje de programación hecho por programadores de alto nivel para programadores. C++, como su ancestro, sigue muy ligado al hardware subyacente, manteniendo una

---

<sup>29</sup> *Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora.

considerable potencia para programación a bajo nivel y en el manejo de la memoria del sistema, pero se la han añadido elementos que le permiten también un estilo de programación con alto nivel de abstracción.

En la actualidad, C++ es un lenguaje versátil, potente y general. Su éxito entre los programadores profesionales le ha llevado a ocupar el primer puesto como herramienta de desarrollo de aplicaciones. El C++ mantiene las ventajas del C en cuanto a riqueza de operadores, expresiones, flexibilidad, concisión y eficiencia. Además, ha eliminado algunas de las dificultades y limitaciones del C original.

Este lenguaje de programación ha experimentado un extraordinario éxito desde su creación. Muchos sistemas operativos, compiladores e intérpretes han sido escritos en C++ (el propio Windows y Java). Una de las razones de su éxito es ser un lenguaje de propósito general que se adapta a múltiples situaciones. Ha servido de justificación para el diseño de otros lenguajes que intentan eliminar sus inconvenientes y al mismo tiempo mantener sus virtudes.

**Para su elección se tuvo en cuenta que:**

- ➔ En tiempo de ejecución, supera a su equivalente en otros lenguajes como C# y Java, ya que no importa la optimización de JIT<sup>30</sup>, nada va a ir más rápido que el acceso de puntero directo a memoria.
- ➔ El modelo de memoria, los tipos atómicos y el almacenamiento local de hilos brindan las garantías básicas para soportar una buena librería de hilos.
- ➔ Es considerado por expertos como el lenguaje más potente para el trabajo con el hardware de la máquina.
- ➔ Potencia el manejo directo de interrupciones o la manipulación directa de hardware, es considerado la solución ideal en cuanto a la obtención de un mejor rendimiento del hardware.
- ➔ Es un lenguaje de plataforma de bajo nivel, con el que los desarrolladores alcanzan acceso a recursos primarios como sólo este lenguaje puede ofrecer.

---

<sup>30</sup> Just in Time Compilation.

- ➔ Potencia un mejor rendimiento de la caché de las aplicaciones permitiendo que estas sean más rápidas que las desarrolladas en otros lenguajes de programación.
- ➔ Si se tienen datos contiguos en la memoria, con el empleo correcto de punteros y el manejo directo de memoria se logra un rendimiento excelente que no tiene comparación con los de ningún otro lenguaje.
- ➔ Permite el tratamiento intensivo de gráficos superior al de Java y C# ya que posee una perfecta integración con OpenGL<sup>31</sup> al estar ésta programada en C, con ello se potencia la aceleración de hardware en 3D. Algunos de estos efectos son niebla, transparencias, reflexiones, texturas 3D, sombras volumétricas, efecto de movimiento y otros.

### 2.3.4 Framework.

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. También facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Para la implementación del subsistema de transmisión de noticias Primicia se escoge Qt 4.7.0 como framework de desarrollo.

Qt es un framework para el desarrollo de aplicaciones multiplataforma. **Dentro de sus características se encuentran:**

- ➔ Compatibilidad multiplataforma con un sólo código fuente.
- ➔ Performance de C++.
- ➔ Disponibilidad del código fuente.

---

<sup>31</sup> Open Graphic Library.

- ➔ Excelente documentación (Qt Assistant).
- ➔ Internacionalización de aplicaciones (Qt Linguist).
- ➔ Soporte para XML.
- ➔ Conexión a bases de datos.
- ➔ OpenGL.
- ➔ Programación para redes.
- ➔ Biblioteca de clases intuitiva.
- ➔ Herramientas de desarrollo integrado.
- ➔ Integración con Visual Studio.
- ➔ Utiliza los controles de ActiveX en aplicaciones Qt.
- ➔ Integra Direct3D<sup>32</sup> para el hardware de gráficos acelerados.
- ➔ Soporte para aplicaciones 64-bits en la plataforma Mac<sup>33</sup> con la API Cocoa<sup>34</sup>.
- ➔ Librerías para analizar la carga de las aplicaciones haciendo benchmark<sup>35</sup> de las mismas.

La librería Qt posee numerosos módulos que permite trabajar con bases de datos, gráficos, imágenes, sonidos, etc.

Las aplicaciones basadas en Qt tienen una buena respuesta y un buen uso de la memoria. El desarrollo con este framework es muy apropiado para proyectos de larga escala, tanto comerciales como de libre

---

<sup>32</sup> Es parte de DirectX, una API propiedad de Microsoft disponible tanto en los sistemas Windows de 32 y 64 bits, como para sus consolas Xbox y Xbox 360 para la programación de gráficos 3D.

<sup>33</sup> Macintosh (abreviado Mac) es el nombre con el que actualmente nos referimos a cualquier computadora personal diseñada, desarrollada, construida y comercializada por Apple Inc.

<sup>34</sup> Conjunto de frameworks orientados a objetos que permiten el desarrollo de aplicaciones nativas para Mac OS X.

<sup>35</sup> Técnica utilizada para medir el rendimiento de un sistema o componente de este.

distribución. Por ejemplo, el escritorio KDE<sup>36</sup> y FreeBSD<sup>37</sup>, también el paquete de KOffice<sup>38</sup>, y así miles de programas tanto de libre distribución como comerciales.

El conjunto de herramientas Qt tiene un enfoque más eficiente para facilitar la tarea de gestión de memoria para sus programadores: cuando un objeto es suprimido, todos los objetos dependientes se eliminan automáticamente. El enfoque de Qt es no interferir con la libertad del programador para borrar manualmente cuando lo deseen.

La biblioteca Qt está concebida como multiplataforma, permitiendo escribir código que se compilará y ejecutará en distintas plataformas, incluyendo Unix, Linux, FreeBSD o incluso Windows. También decir que funciona en todas las principales plataformas y tiene un amplio apoyo. El API de la biblioteca cuenta con métodos para acceder a bases de datos mediante SQL, así como uso de XML, gestión de hilos, soporte de red, una API multiplataforma unificada para la manipulación de archivos y una multitud de otros métodos para el manejo de ficheros, además de estructuras de datos tradicionales.

Qt incluye un framework de animación que ayuda a construir grandes animaciones, interfaces gráficas de alto rendimiento sin la molestia de la gestión de estructuras complejas, temporizadores, etc. También permite crear animaciones personalizadas y funciones de interpolación. Tiene una clase que define las funciones de la funcionalidad compartida por todas las animaciones, la clase QAbstractAnimation que es la base de todas las animaciones. QAbstractAnimation proporciona las funciones virtuales puras utilizadas por las subclases para seguir el progreso de la animación: la duración y updateCurrentTime (). Las vistas gráficas no están omitidas, se puede animar utilizando QGraphicsWidgets y QGraphicsObject las cuales heredan de QGraphicsItem. Cuenta con efectos que pueden ser utilizados para alterar la apariencia de los elementos de la interfaz de usuario. Las animaciones son controladas usando las curvas fáciles y pueden ser agrupadas. Esto permite que las animaciones sean de complejidad arbitraria.

El API es fácil de entender con funciones tales como start (), stop (), pause (), y currentTime (). Qt cuenta actualmente con un sistema de triple licencia: GPL v2/v3 para el desarrollo de software de código abierto y software libre, la licencia de pago QPL para el desarrollo de aplicaciones comerciales, y a partir de la versión 4.5 una licencia gratuita pensada para aplicaciones comerciales, LGPL.

---

<sup>36</sup> Un conocido entorno de escritorio para Linux.

<sup>37</sup> Sistema operativo libre para computadoras basado en las CPU de arquitectura Intel.

<sup>38</sup> Una alternativa para el MS Office.

### 2.3.5 Entorno de Desarrollo Integrado (IDE).

Un entorno de desarrollo informático (en inglés *integrated development environment*) es un programa informático compuesto por un conjunto de herramientas de programación, que proveen facilidades a los programadores para agilizar el proceso de desarrollo de software.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación. Consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Algunas veces contiene también un sistema para llevar a cabo el control de versiones. Muchos de los IDEs modernos llevan integrados además un navegador de clases, un inspector de objetos y diagramas de herencia de clases para ser usados en el desarrollo orientado a objetos. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. Estos proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación, tales como, C++, C#, Java Object Pascal, Visual Basic, entre otros. En muchos casos se puede dedicar a un sólo lenguaje de programación, por ejemplo, Borland C++ o puede utilizarse para varios como es el caso de Eclipse.

Para la implementación del subsistema de transmisión de noticias Primicia se escoge Qt Creator 2.0.1 IDE.

Como su nombre lo indica, está basado en la librería Qt, la biblioteca multiplataforma de interfaces gráficas de usuario. **Cuenta con las siguientes características principales:**

- ➔ Editor avanzado para C++.
- ➔ Diseñador de formularios (GUI) integrado.
- ➔ Herramientas para la administración y construcción de proyectos.
- ➔ Completado automático.
- ➔ Depurador visual.

Qt Creator es una aplicación integral y un marco para el desarrollo de aplicaciones de interfaz de usuario de Windows que también se puede implementar en muchos otros escritorios y sistemas integrados de explotación sin reescribir el código fuente. La codificación puede resultar uno de los factores que insumen

más tiempo en las aplicaciones de desarrollo, pero el editor de códigos C++ de Qt presenta la torna más fácil y más rápida. Posee revisor de código estático y suministra sugerencias de estilo a medida que se tipea. Se puede obtener ayuda sensible al contexto, combinación de paréntesis y modos de selección, dos funciones muy útiles que permitirán terminar el trabajo más rápido.

Se escoge Qt Creator como Entorno Integrado de Desarrollo ya que:

- Es uno de los mejores IDEs recomendado para trabajar con el framework Qt.
- Brinda apoyo a la refactorización de código fuente para C++.
- A través de la biblioteca de clases Qt permite construir interfaces de usuario complejas de una forma visual y rápida, lo que deja más tiempo para centrarse en la parte más importante
- Permite utilizar los controles de ActiveX en aplicaciones Qt e integra Direct3D para el hardware de gráficos acelerados.
- Se centra en proporcionar características que ayudan a los nuevos usuarios de Qt a desarrollar más rápido, y también aumentar la productividad de los desarrolladores con experiencia.
- Al estar perfectamente integrado con la biblioteca Qt escrita genuinamente en código C++ nativo, potencia el manejo de efectos, transparencias, sombras de forma fácil empleando las clases *QGraphicsItem*, *QGraphicsWidgets* y *QGraphicsObjects*. Además se potencia el manejo de videos y la animación de objetos a través de las clases *QObject*, *QAbstractAnimation*, *QAbstractVideoSurface*.
- Integra completamente a Qt Designer, herramienta que permite el desarrollo de formularios y presentaciones gráficas para las aplicaciones.
- Permite la visualización previa de la interfaz, soporte para widgets y un editor de propiedades poderoso.
- Acelera el desarrollo de interfaces de alto rendimiento, a la vez que proporciona una forma fácil de diseñar interfaces gráficas de usuario avanzadas generando el código fuente para las mismas, lo que permite al desarrollador ajustarlo a sus necesidades.

## 2.4 Descripción de otras tecnologías necesarias en la propuesta.

Para la implementación del subsistema de transmisión de la Plataforma de Televisión Informativa PRIMICIA se hace necesario describir otras tecnologías que apoyarán la propuesta de solución.

### 2.4.1 Soporte para el uso de XML.

Al realizar un análisis de los procesos que se llevan a cabo en el subsistema de transmisión, se hace indispensable optimizar el proceso de almacenamiento de la información. Las propiedades de los elementos configurables en el proceso de redacción de la noticia cambian frecuentemente durante la redacción de la misma. Es por ello que no es conveniente que sean almacenadas en la base de datos, ya que provocaría sucesivas actualizaciones y produciría una sobrecarga que atentaría contra el mantenimiento y rendimiento de la aplicación en general. Para almacenar la configuración de todos los elementos que conforman la noticia se estructurará en un formato XML.

**Entre las potencialidades que tiene el uso de XML están:**

▣ ***Independencia entre la Presentación y los Datos:*** es la separación entre el contenido (los *datos*) y la presentación (*el formato de los datos*). Esta característica presente en los documentos XML, posibilita que una vez generado el documento que contiene todos los detalles de la forma en que se mostrará la noticia, pueda ser presentada lo mismo en un ambiente web como en uno escritorio, todo depende del analizador de XML que se emplee.

▣ ***Comunicación entre aplicaciones:***

Permite que exista una comunicación entre diferentes aplicaciones sin que el formato esté ligado a ninguna presentación. En muchas ocasiones, empresas de un cierto sector, definen un vocabulario XML común para la interacción entre ellos (documentos de pedidos, facturas, etc). En el caso de Primicia servirá como medio de comunicación entre el subsistema de administración, donde se lleva a cabo la redacción de la noticia y la creación del documento XML y el subsistema de transmisión que es el encargado de leer el documento XML generado y mostrar la noticia.

▣ ***Configuración:*** para guardar los datos de configuración de las aplicaciones.

Un aspecto importante cuando se trabaja con información representada en XML, es que abundan las herramientas para analizar, acceder, tratar texto XML y transformar los datos que contiene. Para ello se utilizan componentes que ofrecen una o las dos API estandarizadas: SAX<sup>39</sup> y DOM<sup>40</sup> llamados Analizadores o Parseres.

SAX es muy utilizado en el procesamiento de XML basado en eventos y tiene como característica principal que lee el documento de principio a fin sin tener que cargarlo completamente en memoria. Por su parte DOM tiene como característica fundamental que carga todo el documento en memoria en una estructura de árbol, lo que facilita el acceso a cada uno de los elementos del documento (nodos del árbol), lo que requiere más memoria, pero hace que sea más fácil de manipular la estructura del documento.

SAX es una interfaz estándar basada en eventos para los analizadores XML. La interfaz Qt sigue el diseño de la implementación de Java SAX2 (20).

La interfaz SAX2 es un mecanismo orientado a eventos para proporcionar al usuario información del documento. Un "evento" en este contexto, significa algo reportado por el analizador, por ejemplo, se ha detectado una etiqueta de inicio o una etiqueta de cierre.

Si bien este es un rápido y sencillo enfoque para leer documentos XML, la manipulación es difícil porque los datos no se almacenan, simplemente son manipulados y descartados. El módulo Qt XML proporciona una clase abstracta, *QXmlReader*, que define una interfaz para los potentes lectores SAX2. Qt incluye una lectura de implementación *QXmlSimpleReader*, que es fácil de adaptar a través de subclases.

La clase *QXmlInputSource* proporciona los datos de entrada para las subclases *QXmlReader*. Todas las subclases de *QXmlReader* leen la entrada del documento XML desde ésta clase.

Otra de las API es DOM, que proporciona una interfaz para acceder y modificar el contenido y la estructura de un archivo XML. Tiene una vista jerárquica del documento (una vista de árbol). Es un modelo de objeto del documento que está residente en memoria, es por ello que se hace mucho más fácil su manipulación. Todos los nodos en el árbol de documentos DOM son subclases de *QDomNode*. El documento en sí mismo es representado como un objeto *QDomDocument*.

---

<sup>39</sup> *Simple API for XML*.

<sup>40</sup> Document Object Model.

La clase *QDomNode* es la clase base para todos los nodos en un árbol DOM. Muchas funciones en el DOM retornan un *QDomNode*. La clase *QDomDocument* representa un documento XML completo. Conceptualmente es la raíz del árbol de documento y provee el acceso primario de los datos del documento.

Otro aspecto importante a tener en cuenta cuando se va a desarrollar aplicaciones usando XML es la validación de este mediante XML-Schemas. Estos elementos le proporcionan al parser una gramática por la cual se va a regir a la hora de crear o validar los documentos XML.

Hay dos formas de QtXmlPatterns que se pueden utilizar para validar esquemas: puede utilizar la API C++ en su aplicación Qt usando las clases *QXmlSchema* y *QXmlSchemaValidator* o puede emplearse la utilidad de línea de comando denominado *xmlpatternsvalidator*.

La clase *QXmlSchema* carga, compila y valida archivos XML Schema W3C. Se utiliza para representar los esquemas que se ajustan al esquema XML 1.0 (21).

La clase *QXmlSchemaValidator* valida documentos con instancias de XML frente a un esquema XML de W3C. Las cargas de la clase *QXmlSchemaValidator*, analiza un documento de instancia XML y lo valida contra un esquema XML de W3C que ha sido compilado con *QXmlSchema*.

## 2.4.2 Soporte de medias

Los objetos media son una instancia de la clase *mediaobject*, permiten iniciar, pausar y detener la reproducción de un flujo de medios de comunicación, es decir, que proporcionan un control básico de la reproducción. Se puede pensar en el objeto como un sencillo reproductor multimedia. Un objeto media es siempre instanciado con el constructor por defecto y se suministra con una fuente media. Como complemento a los objetos media, *Phonon*<sup>41</sup> también proporciona la clase *MediaController*, que proporciona control sobre funciones que son opcionales para una media determinada.

---

<sup>41</sup> Framework multimedia estándar de KDE 4, también parte de Qt desde la versión 4.4.

### 2.4.2.1 Audio

La clase *QSound* proporciona acceso a la plataforma de audio .Qt provee la operación de audio más comúnmente requerida en aplicaciones GUI: reproducir asincrónicamente un archivo de sonido. Para ello, es muy fácil utilizar la función estática *play ()*.

Una vez creado el objeto *QSound* puede ser consultado por su *fileName ()* y un número total de *loops ()* que indica el tiempo que se estará reproduciendo. El número de repeticiones puede ser alterado usando la función *setLoops ()*. Mientras el sonido se esté reproduciendo la función *loopsRemaining ()* retornará el número restante de repeticiones. Para determinar si el sonido ha terminado de reproducirse se debe emplear la función *isFinished ()*.

La reproducción del sonido usando el objeto *QSound* puede usar más memoria que la función estática, pero éste también puede reproducirse de forma inmediata. Para determinar si existen facilidades de sonido se puede usar la función estática *isAvailable ()*.

### 2.4.2.2 Video

Para la reproducción de un video, se proporciona *VideoWidget*. Los widget automáticamente elegirán un dispositivo disponible para la reproducción del video, que suele ser una tecnología entre la aplicación Qt y la tarjeta gráfica, como DirectShow en Windows.

El *VideoWidget* no tiene por qué ajustarse a una categoría, es clasificado automáticamente por *VideoCategory*, sólo hay que asegurar que el audio también se incluye en la misma categoría. Para realizar el flujo de video, Phonon lee la dirección URL del archivo de video dondequiera que ésta esté.

Existen dos formas de realizar las transmisiones:

VOD (video bajo demanda): el usuario puede interactuar con el recurso que se pone a disposición haciéndole peticiones al servidor, entiéndase como, reproducir, poner pausa o parar.

*Brocadas*: los usuarios no pueden interactuar con el recurso, solo se puede manejar desde el servidor (22).

Muchos son los protocolos que permiten la transmisión por streaming como por ejemplo RTSP, MMS, RTP, UDP y otros. Hasta el momento solo se han probado dos de ellos en la plataforma PRIMICIA: RTSP Y MMS.

### 2.4.2.3 Texto

Qt contiene la clase *QGraphicsTextItem* que proporciona un elemento de texto que se puede agregar a un *QGraphicsScene* para mostrar texto con formato.

Para cambiar el texto del elemento, se debe pasar un objeto *QString* al constructor de la clase *QGraphicsTextItem*, o llamar la función *setHtml ()* / *setPlainText ()*. El ancho de texto del elemento puede ser cambiado usando *setTextWidth ()* y obtenido mediante *textwidth ()*. Es posible crear un documento editable mediante la configuración de bandera de *Qt::TextEditorInteraction* utilizando la función *setTextInteractionFlags ()*.

### 2.4.2.4 Imagen

Qt proporciona cuatro clases para el manejo de datos de la imagen: *QImage*, *QPixmap*, *QBitmap* y *QPicture*. *QImage* está diseñado y optimizado para la manipulación y acceso directo a píxeles, mientras que *QPixmap* está diseñado y optimizado para mostrar imágenes en pantalla. *QBitmap* es sólo una clase de conveniencia que hereda de *QPixmap* y que facilita la implementación. Por último, la clase *QPicture* es un dispositivo que graba y reproduce comandos *QPainter* (23).

La clase *QPixmap* es una representación de la imagen fuera de pantalla que se puede utilizar como un dispositivo de pintura. Es la clase que se utiliza actualmente para la manipulación de la imagen ya que no se desea modificarla sino mostrarla.

Un objeto *QPixmap* se puede visualizar fácilmente en la pantalla usando la clase *QLabel* que incluye la propiedad *pixmap* o una de las subclases *QAbstractButton* (como *QPushButton* y *QToolButton*).

Además de los constructores ordinarios, un objeto *QPixmap* se puede construir utilizando el *grabWidget* estática () y *grabWindow* (), funciones que crea un *QPixmap* y pinta el widget dado. Hay funciones para convertir entre *QImage* y *QPixmap*. Por lo general, la clase *QImage* se utiliza para cargar un archivo de imagen, opcionalmente manipular los datos de la imagen antes que el objeto *QImage* sea convertido en un *QPixmap* para ser mostrado en pantalla. Alternativamente, si no se desea la manipulación, el archivo

de imagen se puede cargar directamente en un QPixmap. En Windows, la clase QPixmap soporta la conversión entre HBITMAP y QPixmap.

### 2.4.3 Uso del timer

La clase *QTimer* proporciona temporizadores repetitivos y con muy corta duración. También esta clase proporciona una interfaz de programación de alto nivel para los temporizadores. Para usarlo, cree un objeto *QTimer*, conecte su señal de *timeout ()* a los slots correspondientes, y llame la función *start ()*. A partir de entonces se emitirá el *timeout ()* de la señal a intervalos constantes.

En las aplicaciones multiproceso, puede utilizar *QTimer* en cualquier subproceso que tiene un ciclo de eventos. Para iniciar un ciclo de eventos de un hilo sin interfaz gráfica, use *QThread::exec ()*. Qt usa la afinidad del temporizador de hilo para determinar qué hilo emitirá el *timeout ()* de la señal. Debido a esto, debe iniciar y detener el temporizador en su hilo, no es posible iniciar un temporizador desde otro hilo (24).

Una alternativa al uso *QTimer* es llamar *QObject::startTimer ()* por su objeto y implementa de nuevo el *QObject::TimerEvent ()* del controlador de eventos en su clase (que debe heredar *QObject*). La desventaja es que *TimerEvent ()* no admiten las funciones de tan alto nivel como contadores de tiempo de disparo único o señales. Otra alternativa al uso de *QTimer* es utilizar *QBasicTimer*. Típicamente es menos incómodo que el uso de *QObject::startTimer ()* directamente.

## 2.5 Conclusiones Parciales.

En este capítulo se realizó un análisis de las tecnologías sobre las cuales se trabajará para el desarrollo del subsistema de transmisión de la Plataforma de Televisión Informativa Primicia , realizando para ello su correspondiente caracterización y comparación para escoger los lenguajes de programación, sistema gestor de base de datos, framework y otras. Al terminar este capítulo se logra dominar la fundamentación teórica necesaria para comprender la construcción del subsistema.

# Capítulo 3: Presentación de la solución propuesta.

## 3.1 Introducción

En este capítulo se dará una introducción a la solución que se formula. Al no identificarse un negocio se presenta la conceptualización del entorno mediante un modelo de dominio. Se examina cada una de las entidades y conceptos presentes en el contexto donde trabajará el sistema, además de las relaciones existentes entre cada uno de estos. Se especifican los requisitos funcionales y no funcionales que deberá cumplir la solución propuesta. Además se describen los actores y se detallan los casos de usos.

## 3.2 Modelo de Dominio

El modelo de dominio puede utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño. Similares a los mapas conceptuales utilizados en el aprendizaje, el modelo de dominio es utilizado por el analista como un medio para comprender el sector industrial o de negocios al cual el sistema va a servir. (25)

### 3.2.1. Conceptos y principales eventos del entorno.

PRIMICIA es una plataforma que admite la administración y **transmisión** de **noticias** en diferentes formatos mediante un canal de televisión, con el objetivo de mantener actualizada cualquier comunidad, empresa, o entidad que requiera de información clara, precisa y de forma inmediata.

La transmisión del canal puede realizarse de tres formas diferentes, por:

**Sección temática** permite establecer el orden en que serán mostradas las secciones, se comienza con la primera sección temática y las noticias asociadas a la misma, una vez que termine con esta se pasa a la otra y así sucesivamente, creando un ciclo mientras exista vigencia en las noticias publicadas. En caso que exista una sección temática que no tenga noticias asociadas esta no es mostrada.

**Bloque de noticias** funciona estableciendo el horario en que saldrá al aire un bloque de noticias y durante ese tiempo se muestran todas las noticias asociadas al mismo, creando de esta manera un ciclo durante el horario establecido. Cuando concluye el período establecido se dejan de transmitir.

**Continua** es aquella donde las noticias se muestran en el orden en que sean publicadas en el sistema, creando un ciclo de transmisión mientras existan.

La entidad que haga uso de la plataforma debe seleccionar cual es el tipo que desea usar de las antes

Las **noticias** son gestionadas desde el Subsistema de Administración de la plataforma, cuando se redactan se asignan a un bloque de noticias o a una sección temática en caso que la transmisión se esté realizando por una de estas dos variantes. En otro caso no se le asocia a nada en específico.

El sistema también permite gestionar **canales externos** los cuales sirven para enlazar la transmisión de la plataforma con estos canales en el momento deseado, ya sea para la transmisión directa de una información inmediata o simplemente para completar la programación del canal cuando no existan noticias.

A continuación se muestra el modelo de dominio, obtenido gracias a los conocimientos adquiridos de los principales conceptos que sustentan el problema actual. [Diagrama de clases de modelo de dominio.](#)

### 3.2.2. Términos del Dominio.

El siguiente glosario de términos ayudará a una mejor comprensión de los conceptos presentes en el entorno del problema.

**Video:** Un video es un sistema de grabación y reproducción de imágenes, que pueden estar acompañadas de sonidos

**Imagen:** Se refiere a la figura, representación, semejanza o apariencia de algo. Por ejemplo: *“Esta imagen representa la caída del muro de Berlín”, “Eres la viva imagen de tu padre”, “Necesito una imagen para ilustrar mi idea”*. Una imagen también es la representación visual de un objeto a través de técnicas de la fotografía, la pintura, el diseño, el video u otras disciplinas.

**Texto:** Conjunto coherente de enunciados, ya sean escritos u orales. Se trata de una composición de signos que está codificada en un sistema y que forma una unidad de sentido. El texto tiene intención comunicativa: a través de sus signos, busca transmitir un cierto mensaje que adquiere sentido de acuerdo al contexto. La extensión del texto es muy variable, desde unas pocas palabras hasta millones de ellas. De hecho, un texto es virtualmente infinito.

**Noticia:** Es el contenido de una comunicación antes desconocida, un hecho divulgado o la divulgación de una doctrina. En otras palabras, la noticia es un conocimiento o una noción. En los medios de comunicación masivos, una noticia es una redacción o un relato que informa al público sobre un hecho novedoso o atípico, ocurrido dentro de una comunidad o determinado ámbito específico, lo que amerita su divulgación.

**XML:** XML proviene de eXtensible Markup Language (“Lenguaje de Marcas Extensible”). Se trata de un metalenguaje (un lenguaje que se utiliza para decir algo sobre otro lenguaje) extensible de etiquetas que fue desarrollado por el World Wide Web Consortium (W3C), un consorcio internacional que elabora recomendaciones para la World Wide Web.

**Base Datos:** Una base de datos o banco de datos (en ocasiones abreviada con la sigla *BD* o con la abreviatura *b. d.*) es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.

**Streaming De Video:** Consiste en la distribución de video por la red. La palabra *streaming* se refiere a que se trata de una corriente continua (sin interrupción). Este tipo de tecnología permite que se almacenen en un búfer lo que se va viendo.

### 3.3. Especificación de los requisitos de software.

La primera etapa de este proceso ingenieril es identificar los requisitos con los que debe cumplir el software, esto ayuda a que se logre una mejor modelación y diseño, además de contribuir al éxito en el desarrollo del producto. (26)

Con los requisitos se puede establecer una guía para el desarrollo de cualquier software hacia un sistema adecuado. A través de estos se concretan los objetivos generales de forma tal que se beneficien tanto el negocio como sus actores; se consigue una descripción correcta de lo que debe realizar el sistema y delimita su alcance; se acrecienta la comunicación entre un cliente y el grupo de desarrollo; y sirven de base para la validación, verificación y procedimientos de aceptación de cualquier producto de software. A continuación se mencionan a través de requerimientos funcionales y no funcionales, las acciones que el sistema debe ser capaz de realizar.

### 3.3.1 Requerimientos Funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Se mantienen invariables sin importar con que propiedades o cualidades se relacionen. (27)

#### **Requisito funcional 1      Transmitir canal.**

El sistema debe ser capaz de posibilitar la transmisión de un canal de televisión.

#### **Requisito funcional 2      Cargar configuración:**

El sistema debe ser capaz de leer ficheros XML.

#### **Requisito funcional 3      Visualizar noticia personalizada:**

Se visualizarán noticias de cualquier formato de pantalla, obteniendo su configuración de un documento XML generado en el subsistema de administración en el momento de confección de la noticia.

#### **Requisito funcional 4      Reproducir fondo musical:**

Se deberá reproducir un fondo musical si es necesario.

#### **Requisito funcional 5      Visualizar cartelera:**

Componer y transmitir una cartera del contenido generado en la escaleta de transmisión correspondiente a la señal interna.

**Requisito funcional 6      Visualizar infocintas:**

El sistema será capaz de mostrar cintillos informativos inmediatos o programados con anterioridad. Se tendrá en cuenta el tiempo de duración y la sección temática a la que pertenecen.

**Requisito funcional 7      Visualizar información adicional:**

El sistema permitirá mostrar información adicional relacionada con los contenidos que se estén transmitiendo. La información adicional será acerca de:

- ➔ La noticia o la pantalla (duración, tiempo restante, cantidad de pantallas, número de la pantalla actual, sección temática, nombre del redactor, foto de redactor, etc.).
- ➔ Música (autor, nombre, género, etc.).
- ➔ Sección temática (logo, nombre).
- ➔ Imagen (pie de imagen, autor).
- ➔ Título noticia a continuación, sección temática a continuación.
- ➔ Fecha, hora.
- ➔ Logo del canal.

**Requisito funcional 8      Transmitir señal externa:**

Se detendrá la transmisión de la señal del canal y se comenzará a transmitir la señal proveniente de la fuente externa.

**Requisito funcional 9      Mostrar patrón:**

Se mostrará el patrón del canal cuando no se esté transmitiendo ningún material o cuando se encuentre fuera de servicio.

**Requisito funcional 10      Posicionar elementos:**

Los elementos serán posicionados teniendo en cuenta la resolución con la que se va a transmitir la señal del canal, por lo que serán ubicados de forma relativa.

**Requisito funcional 11      Comunicar subsistemas:**

Se establece una comunicación entre los subsistemas administración y transmisión.

**Requisito funcional 12      Verificar cambios programados:**

Debe existir una constante verificación sobre la señal a transmitir. Esto incluye determinar si se mantiene la transmisión de la señal actual, o se establece un cambio programado.

**Requisito funcional 13      Transmitir señal interna:**

Consiste en la función principal del sistema. Esta transmisión interna está compuesta principalmente por noticias.

**Requisito funcional 14      Mostrar videos transicionales:**

Se transmitirán videos al inicio y fin de transmisión, en los cambio de Señal.

**Requisito funcional 15      Visualizar noticia ordinaria:**

Mostrar noticias compuestas por pantallas de tipo Texto, Texto-Imagen, Imagen y Video.

**Requisito funcional 16      Buscar información almacenada**

Se debe establecer una comunicación en busca de información.

**Requisito funcional 17      General escaleta de transmisión:**

Se genera una escaleta de transmisión con el contenido de las noticias a transmitir.

### **3.3.2 Requerimientos No Funcionales**

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable (28).

Requisitos no funcionales del software que se está proponiendo:

**Requisito no funcional 1 Confiabilidad:**

Se debe garantizar un tratamiento adecuado de las excepciones y validación de los datos obtenidos por el sistema desde la base de datos.

**Requisito no funcional 2 Restricciones en el diseño y la implementación:**

Se debe utilizar el lenguaje UML y como herramienta case Visual Paradigm. La arquitectura debe soportar migrar la interfaz de forma rápida, para lograr visualizar cualquiera de los cambios que se produzcan.

**Requisito no funcional 3 Seguridad:**

Solo se tendrá acceso desde las terminales que poseen instalado el software. Se deberán hacer copias de respaldo que puedan restaurar el sistema en caso de fallo crítico o pérdida total de la información.

**Requisito no funcional 4 Software:**

El sistema operativo que soportará la solución será Nova/Linux. Será imprescindible un servidor FTP para proveer los archivos multimedia que se utilizarán en la confección de las noticias.

**Requisito no funcional 5 Legales:**

La aplicación estará sujeta a las mismas restricciones legales que el resto de la Plataforma de Televisión Informativa, PRIMICIA. Sus derechos de autor y otros están determinados por la empresa comercializadora del producto, ALBET S.A y la entidad desarrolladora, UCI.

**Requisito no funcional 6 Interfaces Software:**

Se reutilizan las funcionalidades brindadas por el phonon para la creación transmisión de los videos y música que se visualizará en el canal.

**Requisito no funcional 7 Hardware:**

Variante I: Utilizar un servidor para la gestión de los procesos de administración y transmisión.

Requerimientos de hardware
Dual-Core Xeon 1.60 GHz
4 Gb RAM
HDD 500 Gb
Ethernet 10/100 Mbs
Capturadora Hauppauge WinTV PVR 350
Exportadora ATI Radeon X300

**Tabla 1:** Requerimientos de hardware

Variante II: Utilizar dos servidores especializados para la gestión independiente de los procesos de administración y transmisión.

Requerimientos de hardware del Servidor de Administración.
Dual-Core Xeon 2.33 GHz
4 Gb RAM
HDD 500 Gb
2 x Ethernet Gigabit
Capturadora Hauppauge WinTV PVR 350

**Tabla 2:** Requerimientos de hardware del Servidor de Administración

Requerimientos de hardware del Servidor de Transmisión
Dual-Core Xeon 1.60 GHz
1 Gb RAM
HDD 80 Gb
Ethernet Gigabit
Exportadora ATI Radeon X300

**Tabla 3:** Requerimientos de hardware del Servidor de Transmisión

**Requisito no funcional 8 Usabilidad:**

Las funcionalidades deben ser claras y se debe mostrar la información de forma lógica y correctamente estructurada.

**Requisito no funcional 9 Interfaces de Comunicación:**

Se utilizará la red interna de televisión para la transmisión de la señal del canal.

**Requisito no funcional 10 Fiabilidad:**

El sistema debe estar disponible los 7 días de la semana y las 24 horas del día, garantizando el buen funcionamiento de la Plataforma.

## 3.4 Descripción del Sistema Propuesto

### 3.4.1 Actores Propuestos para el Sistema.

Actor	Descripción
Sistema	Encargado de iniciar la aplicación para lograr la visualización de los contenidos en el canal.

### 3.4.2 Diagrama de Casos de Uso del Sistema.

[Anexo 3: Diagrama de casos de uso del sistema.](#)

### 3.4.3 Especificación de los Casos de Uso.

[Anexo 4: descripción textual de los CUS.](#)

### **3.6 Conclusiones Parciales.**

En este capítulo teniendo ya un conocimiento de las herramientas que van a servir como apoyo para dar solución al problema planteado, se da paso a la presentación de la solución propuesta. Donde teniendo en cuenta las características de la misma se determina la confección de un modelo de dominio, donde este dio paso a la confección de la fase de requerimientos. Como resultado principal de este capítulo quedó plasmada la propuesta de solución para el problema actual del sistema, detallada a través de los actores, casos de usos, diagrama de casos de uso, así como las especificaciones de los casos de uso. El resultado de este capítulo proporciona una visión más ampliada y detallada de los requerimientos de la aplicación.

# Capítulo 4: Construcción de la Solución Propuesta.

## 4.1 Introducción.

Una vez definidos todos los requisitos y conseguido un entendimiento de lo que se quiere realizar, en el presente capítulo se da paso a modelar la propuesta de solución. Para hacer entendible la forma en que está concebido el sistema, se modelan los diagramas de clases del análisis y los diagramas de clases del diseño así como se refinan y estructuran los requisitos obtenidos con anterioridad. Por último se muestra el modelo de implementación mediante los diagramas de componentes y despliegue.

## 4.2 Modelo de Análisis.

Está constituido por las clases del análisis y sus objetos organizados que colaboran. Tiene como característica que ayuda a tener un mejor control sobre el problema a resolver ya que en él se modela la solución de los mismos. En la construcción del modelo de análisis se identifican las clases que describen la realización de los casos de uso, los atributos y las relaciones que existen entre ellas.

### 4.2.1 Clases del Análisis.

Las clases del análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Están constituidas por atributos y entre ellas se establecen relaciones de asociación, agregación/composición, generalización/especialización y tipos asociativos. RUP las clasifica en: Entidad, Interfaz y Control.

### 4.2.2 Diagramas de clases del análisis.

Los diagramas de clases del análisis permiten identificar las clases que utiliza el sistemas y sus relaciones, así como los atributos y sus métodos, estas a su vez normalmente se convierten en las tablas de la Base de Datos, por eso este diagrama es muy consultado en la fase de implementación, donde el desarrollador tendrá que construir las tablas con sus campos basados en este. A continuación se muestran los diagramas de clases del análisis de todos los casos de uso del sistema.

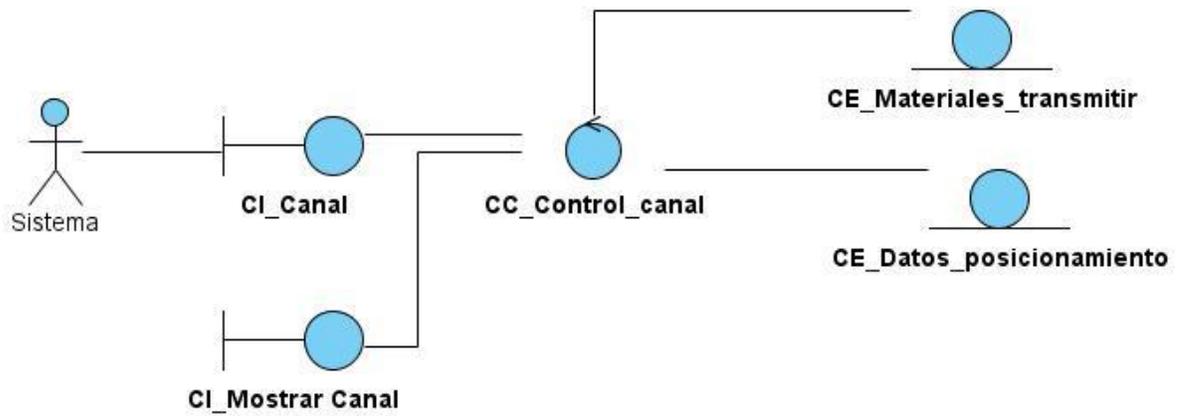


Figura 1: DCA - CU Transmitir.

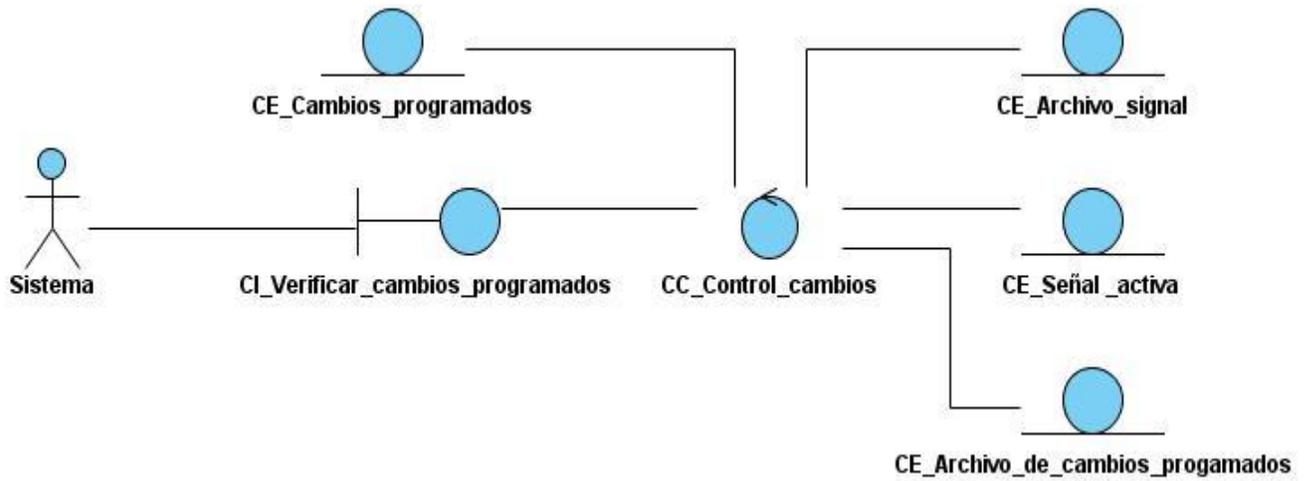


Figura 2: DCA - CU Verificar cambios programados.

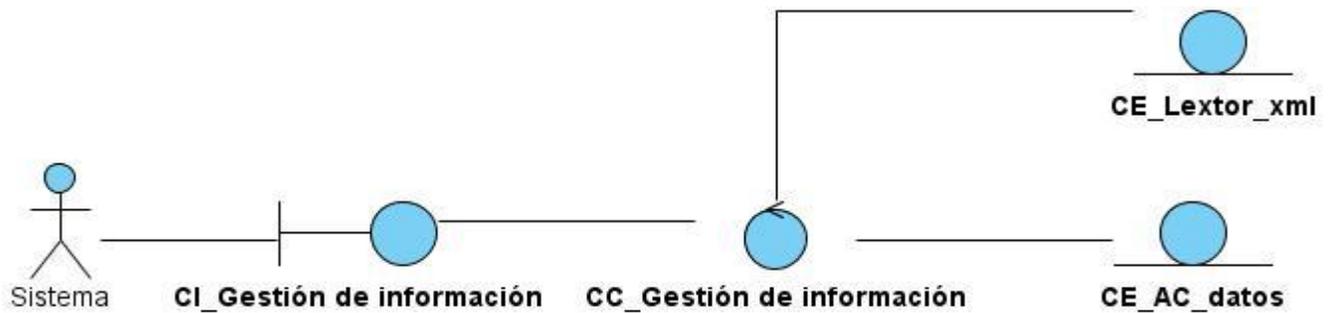


Figura 3: DCA - CU Administrar información.

## 4.3 Modelo de Diseño.

El modelo de diseño es una abstracción de la implementación del sistema. Se utiliza para concebir y para documentar el diseño del sistema de software. Es un producto de trabajo integral y compuesto que abarca todas las clases de diseño, subsistemas, paquetes, colaboraciones y las relaciones entre ellos.

### 4.3.1 Patrones de diseño

Un patrón es una descripción de un problema y la solución, a la que se da un nombre, y que se puede aplicar a nuevos contextos; idealmente, proporciona consejos sobre el modo de aplicarlo en varias circunstancias, y considera los puntos fuertes y compromisos. (29)

Los patrones de diseño deben ser adaptados al software que se está desarrollando, pues constituyen una guía para resolver problemas comunes en programación. Su utilización ayuda a obtener un software de calidad, que sea reutilizable y que permita ampliar sus funcionalidades. Los patrones de diseño son muchos y muy variados, resulta casi imposible revisar todos los que existen a la hora de desarrollar una aplicación.

Los patrones de diseño a tener en cuenta son los GRASP (Patrones para Asignar Responsabilidades), los cuales describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Los mismos están clasificados en Experto, Creador, Bajo Acoplamiento, Controlador y Alta Cohesión.

### 4.3.2 Clases del Diseño.

Es una representación más concreta que el diagrama de clases del análisis. Representa la parte estática del sistema, las clases y sus relaciones. Están compuestas por clases asociadas y atributos, interfaces con sus operaciones y constantes, métodos, información sobre los tipos de atributos, navegabilidad y dependencias. A continuación se muestra una representación de los diagramas de clases del diseño.

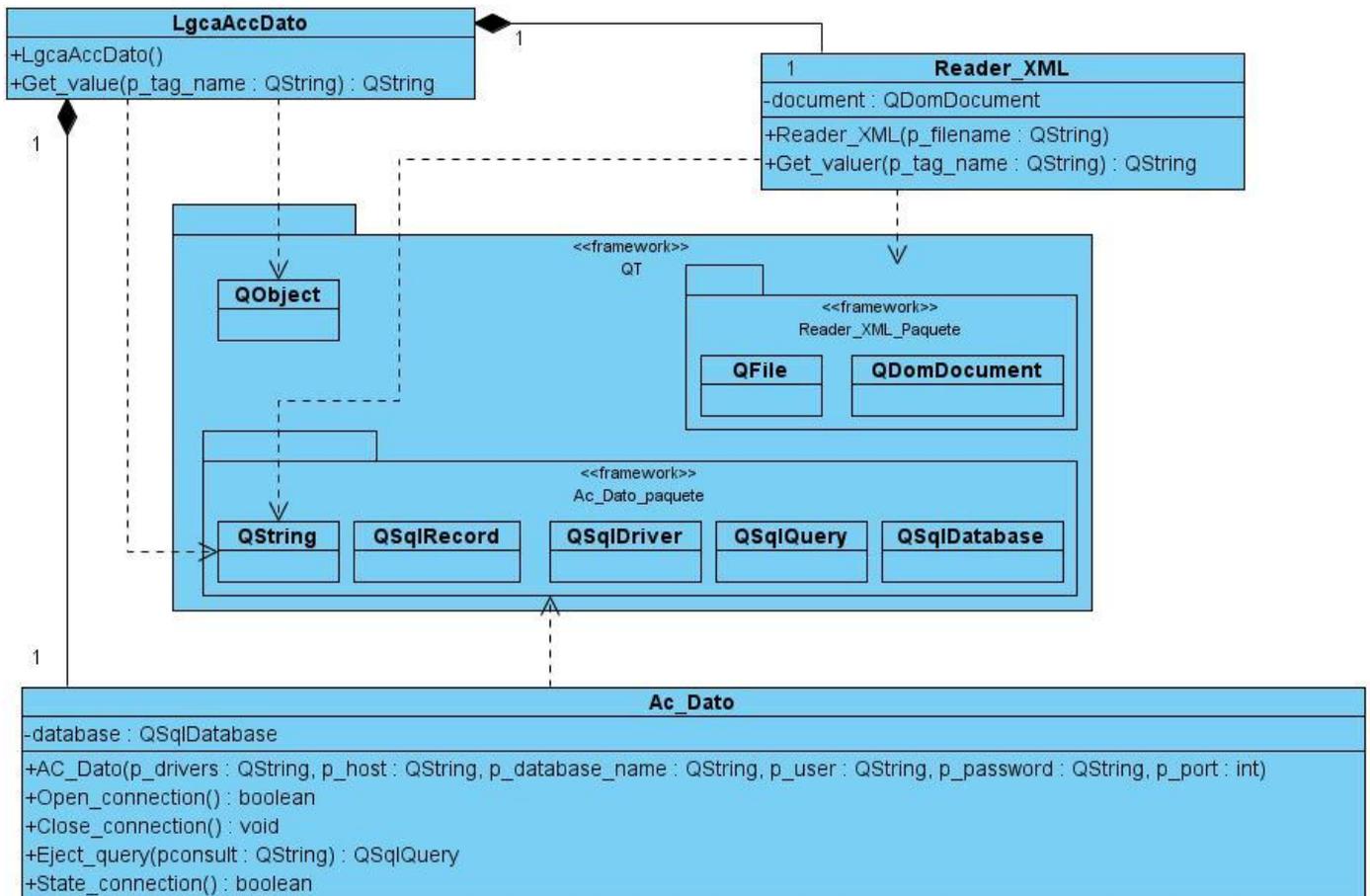


Figura 4: DCD - Administrar información.

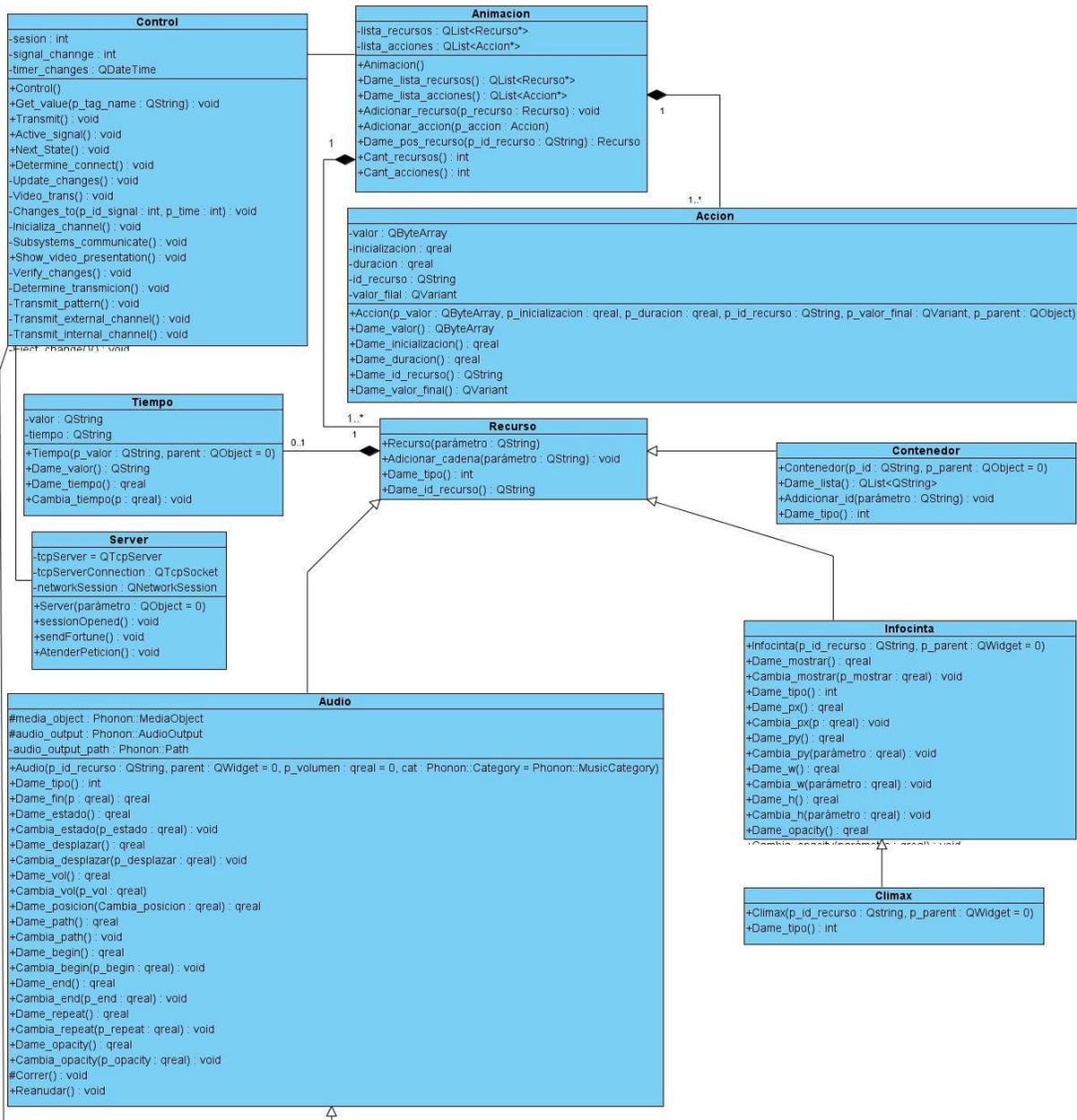


Figura 5: DCA - CU Transmitir (1).

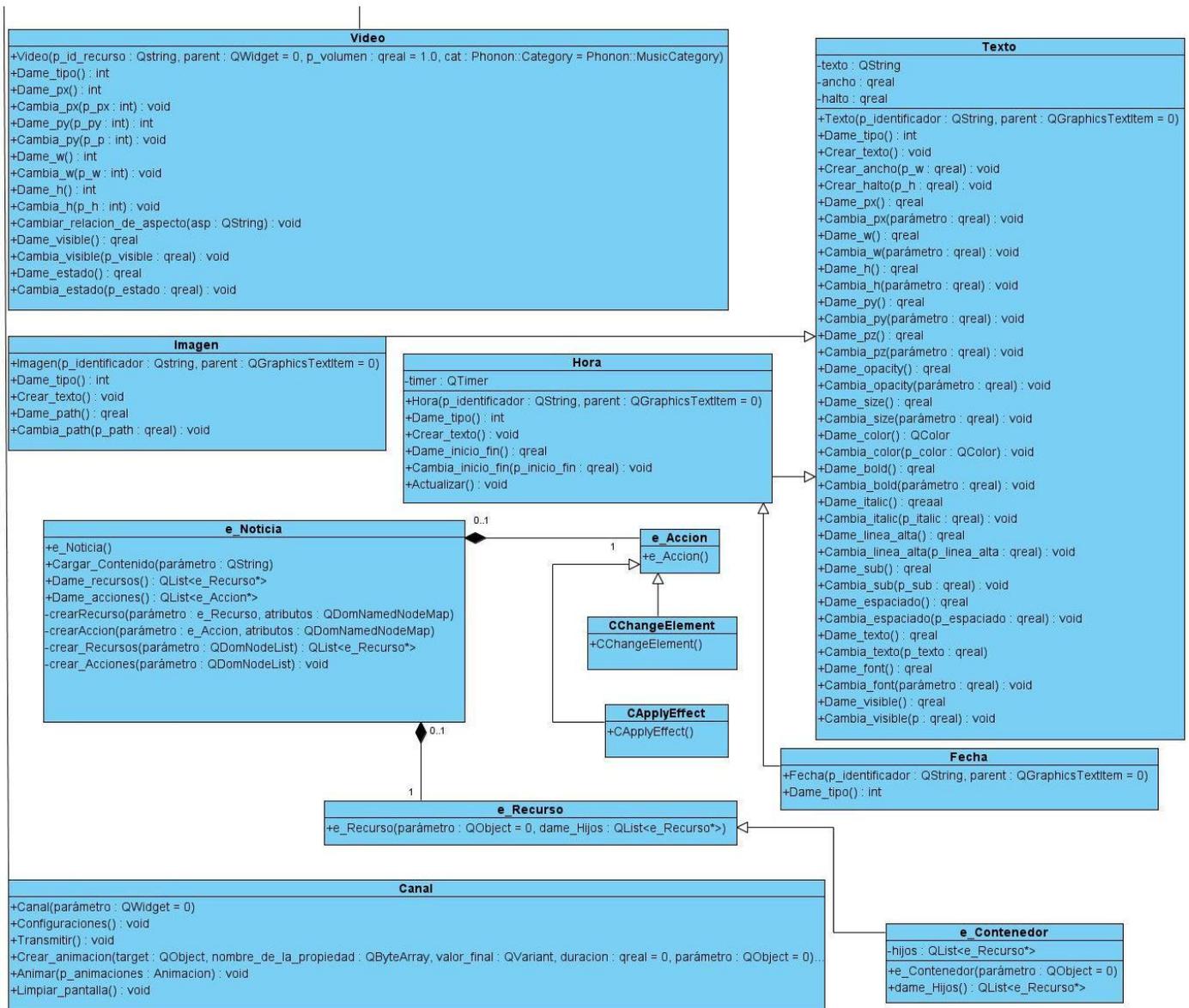


Figura 6:DCA - CU Transmitir (2).

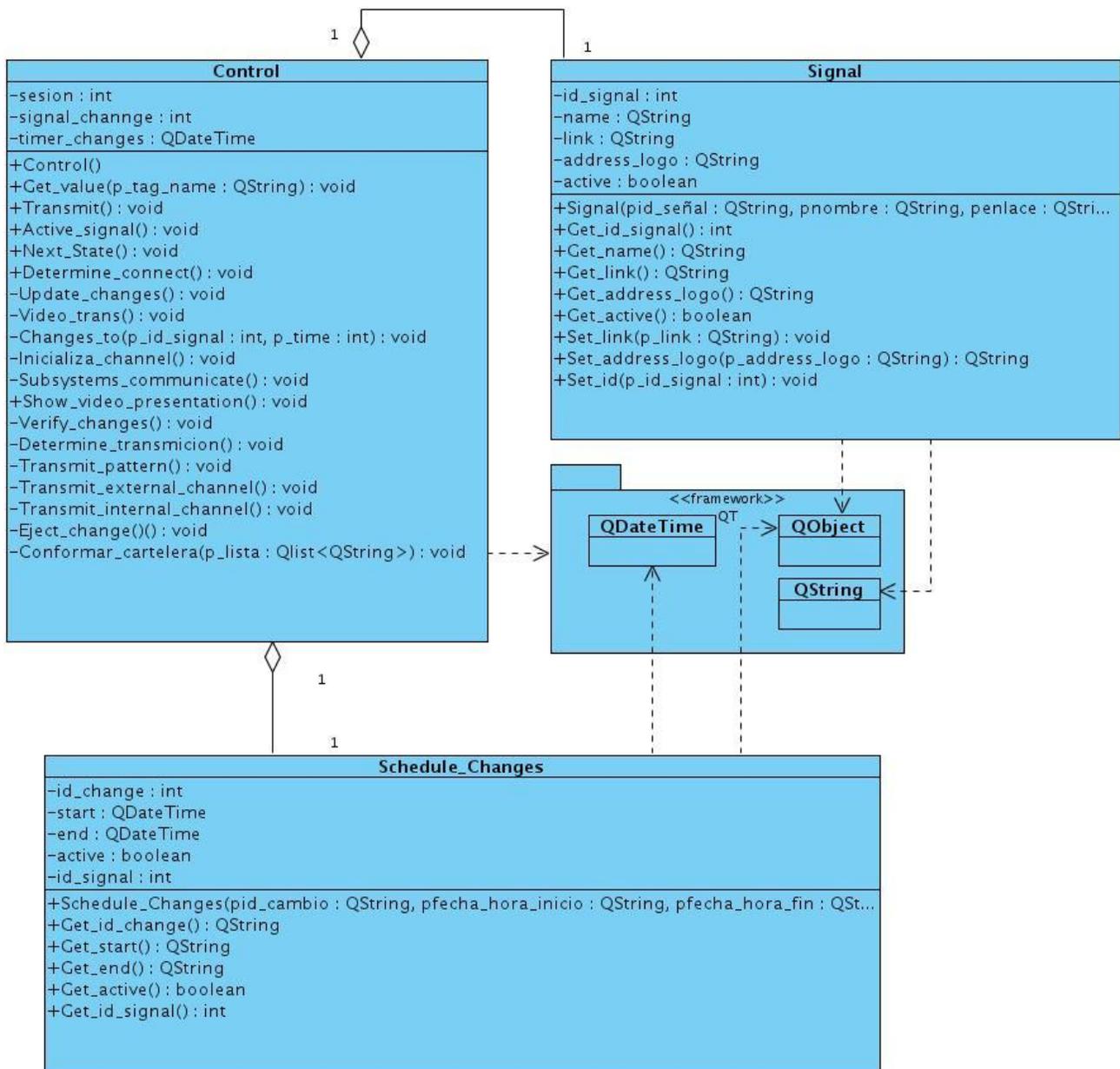


Figura 7: DCA - CU Verificar cambios programados.

## 4.4 Modelo de Despliegue.

El modelo de despliegue describe la distribución física del sistema, muestra cómo están distribuidos los componentes de software entre los distintos nodos de cómputo. Permite comprender la correspondencia entre la arquitectura software y la arquitectura hardware.

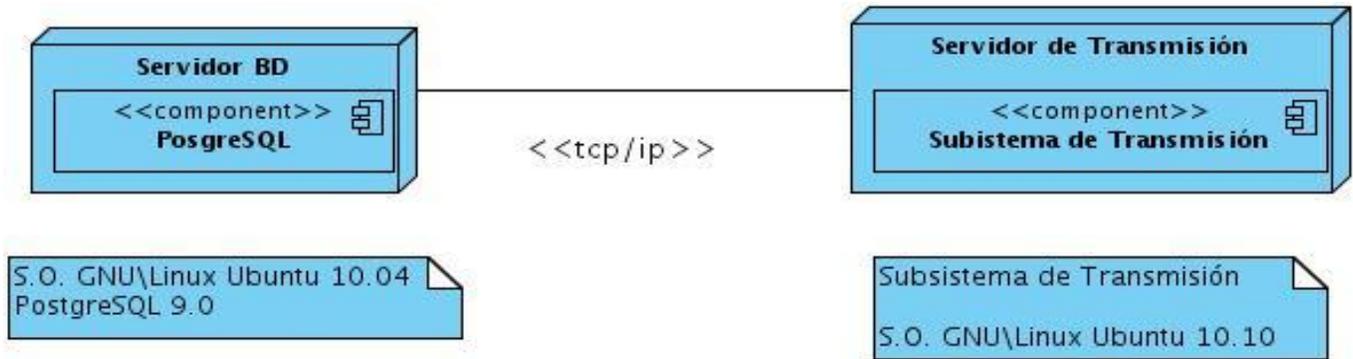


Figura 8: Diagrama de Despliegue.

## 4.5 Diagrama de Componentes.

El Diagrama de Componente muestra la relación entre los componentes del software, sus dependencias, comunicaciones, localización y otras condiciones. Los Diagramas de Componentes son usados para estructurar los componentes en los sistemas del software. Ellos examinan y controlan las dependencias entre componentes o interfaces de los componentes. Un componente representa una parte modular, desplegable y reutilizable de un sistema. (30)

A continuación se muestran los diagramas de componentes confeccionados por cada caso de uso.

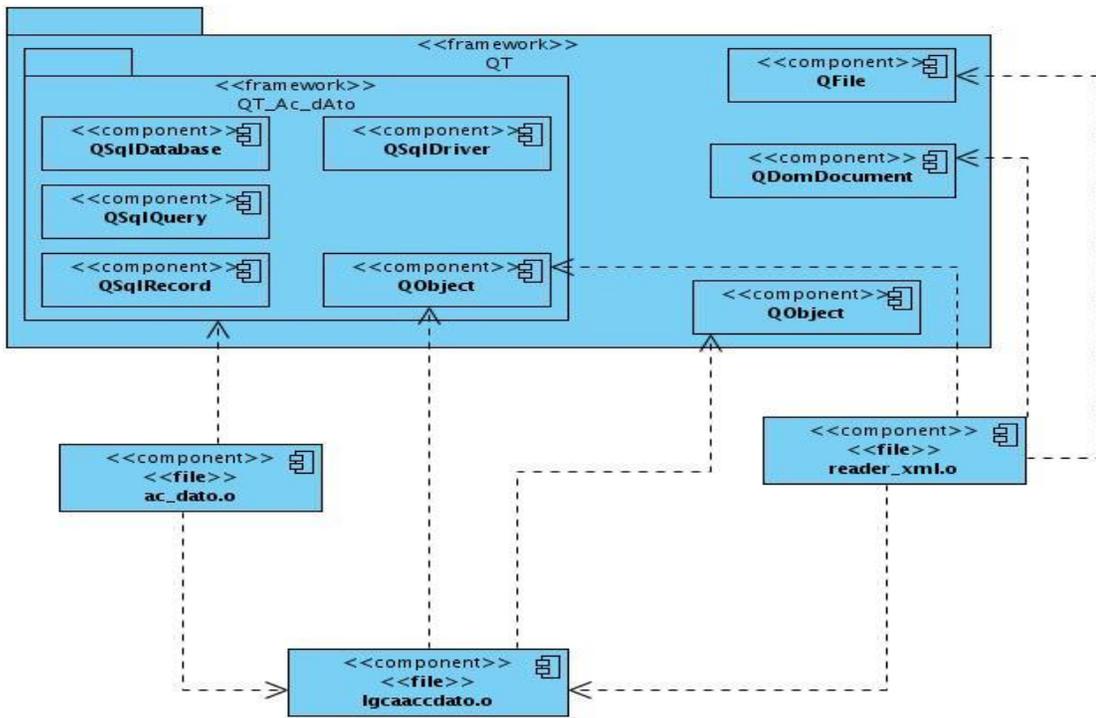


Figura 9: DCMP\_CU Gestión de información.

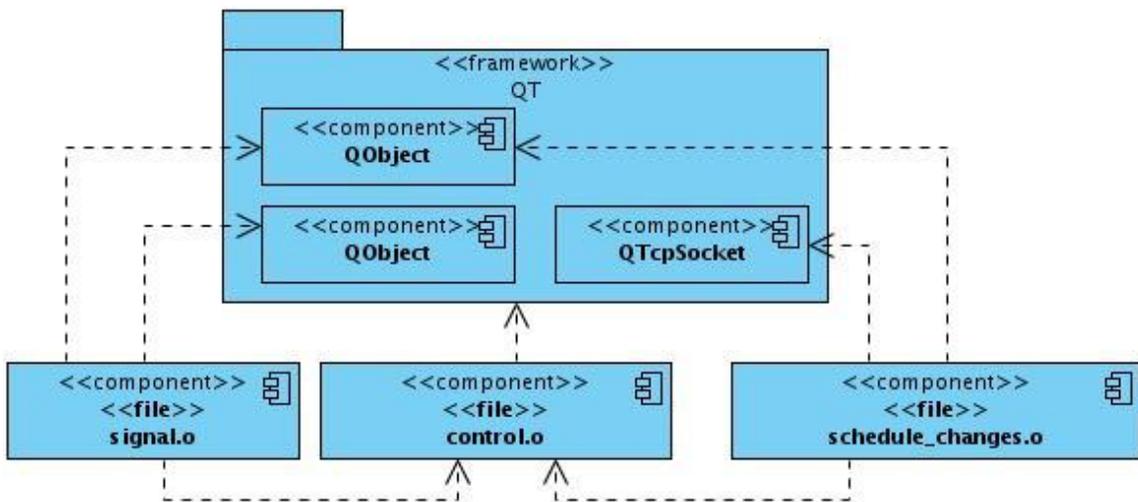


Figura 10: DCMP\_CU Verificar cambios programados.

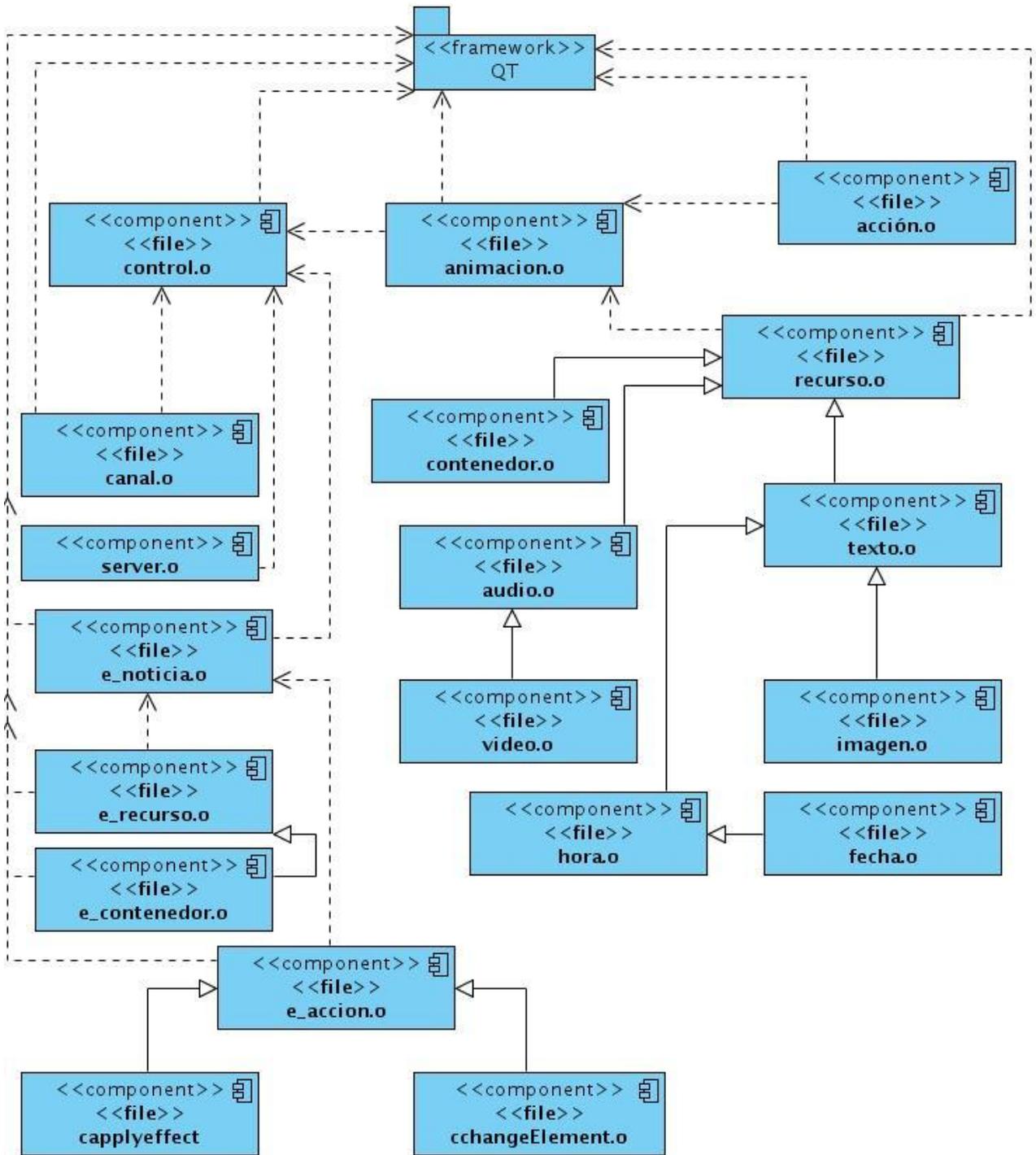


Figura 11: DCMP\_CU Transmitir.

## 4.6 Prueba

La prueba de una aplicación informática es un elemento fundamental para avalar la eficacia del software y representa una revisión de las especificaciones, del diseño y de la codificación de los elementos que componen la aplicación. Existen varias tipologías de pruebas, cada una de ellas con metas y estrategias bien definidas.

### 4.6.1 Elementos del proceso de pruebas

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

### 4.6.2 Niveles de pruebas

**Prueba de desarrollador:** Indica los aspectos de diseño e implementación de las pruebas más adecuadas que debe llevar a cabo el equipo de desarrolladores, a diferencia de la prueba independiente. En la mayoría de los casos, la ejecución de la prueba se produce inicialmente con el grupo de pruebas de desarrollador que la diseñó e implementó; aunque es recomendable que los desarrolladores creen las pruebas de forma que estén disponibles para que las ejecuten grupos de pruebas independientes (31).

**Prueba independiente:** Indica el diseño y la implementación de la prueba realizada más adecuadamente por alguien ajeno al equipo de desarrolladores. Puede considerarse esta distinción un súper conjunto, que incluye validación y verificación independientes. En la mayoría de los casos, la ejecución de la prueba se produce inicialmente con el grupo de pruebas independientes que la diseñó e implementó; aunque los verificadores independientes deberían crear sus pruebas de forma que estén disponibles para que las ejecuten los grupos de pruebas de desarrollador (31).

**Prueba de unidad:** Se centra en la verificación de los elementos más pequeños del software que se puedan probar. Normalmente, las pruebas de unidad se aplican a componentes representados en el modelo de implementación para verificar que se cubren los flujos de control y los flujos de datos y que funcionan como se esperaba. El implementador realiza la prueba de unidad mientras se desarrolla la unidad. Los detalles de la prueba de unidad se describen en la disciplina de implementación (31).

**Prueba de integración:** Las pruebas de integración se realizan para garantizar que los componentes del modelo de implementación funcionan correctamente cuando se combinan para ejecutar un guión de uso. El destino de la prueba es un paquete o un conjunto de paquetes del modelo de implementación. A menudo, los paquetes que se combinan proceden de diferentes empresas de desarrollo. Las pruebas de integración exponen el estado incompleto o los errores de las especificaciones de la interfaz del paquete (31).

**Prueba del sistema:** Normalmente, la prueba del sistema se realiza cuando el software funciona en su totalidad. Un ciclo vital repetitivo permite que las pruebas del sistema se realicen mucho antes, en cuanto se hayan implementado subconjuntos bien formados del comportamiento de guiones de uso. Normalmente, el destino son los elementos en funcionamiento de extremo a extremo del sistema (31).

**Prueba de aceptación:** La prueba de aceptación del usuario es la última acción de prueba antes de desplegar el software. El objetivo de la prueba de aceptación es comprobar si el software está preparado y lo pueden utilizar los usuarios para realizar las funciones y tareas para las que se diseñó (31).

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

### 4.6.3 Tipos de prueba

#### 4.6.3.1 Funcionalidad:

**Prueba de función:** pruebas que se centran en la validación de funciones del destino de la prueba, proporcionan los guiones de uso, los métodos y los servicios necesarios. Esta prueba se implementa y se ejecuta en diferentes destinos de la prueba, incluidas las unidades, las unidades integradas, las aplicaciones y los sistemas.

**Prueba de seguridad:** pruebas que se centran en garantizar que los datos del destino de la prueba (o sistemas) sólo son accesibles para los actores a los que se dirigen. Esta prueba se implementa y ejecuta en varios destinos de la prueba.

**Prueba de volumen:** pruebas que se centran en la verificación de la capacidad del destino de la prueba para manejar grandes cantidades de datos, ya sean de entrada y salida o residentes, en la base de datos. La prueba de volumen incluye estrategias de prueba como la creación de consultas que devolverán el

contenido completo de la base de datos, o que tendrán tantas restricciones que no devolverán ningún dato, o en las que la entrada de datos tiene la cantidad máxima de datos para cada campo.

#### 4.6.3.2 Fiabilidad

**Prueba de integridad:** pruebas que se centran en la evaluación de la fuerza del destino de la prueba (resistencia a los errores) y la conformidad técnica del lenguaje, la sintaxis y la utilización de recursos. Esta prueba se implementa y se ejecuta en diferentes destinos de la prueba, incluidas las unidades y las unidades integradas.

**Prueba de estructura:** pruebas que se centran en la evaluación de la adherencia del destino de la prueba a su diseño y formación. Normalmente, esta prueba se realiza en aplicaciones habilitadas para web y garantiza que todos los enlaces están conectados, se muestra el contenido adecuado y no hay ningún contenido huérfano.

**Prueba de tensión:** se trata de un tipo de prueba de fiabilidad que se centra en la evaluación de cómo responde el sistema en circunstancias anormales. Las tensiones del sistema pueden ser cargas de trabajo extremas, memoria insuficiente, servicios y hardware no disponible o recursos compartidos limitados. Estas pruebas suelen realizarse para saber mejor cómo y en qué áreas fallará el sistema, de forma que se puedan planificar y presupuestar los planes de contingencia y el mantenimiento de las actualizaciones con bastante antelación.

**Prueba de puntos de referencia:** se trata de un tipo de prueba de rendimiento que compara el rendimiento de un destino de la prueba nuevo o desconocido con una referencia conocida, carga de trabajo y sistema.

**Prueba de contienda:** pruebas que se centran en la validación de la capacidad del destino de la prueba para manejar de forma aceptable varias demandas del actor en el mismo recurso (registros de datos, memoria, etc.).

#### 4.6.3.3 Rendimiento

**Prueba de carga:** se trata de un tipo de prueba de rendimiento que se utiliza para validar y evaluar la aceptabilidad de los límites operativos de un sistema bajo cargas de trabajo variables, mientras el sistema que se está probando permanece igual. En algunas variantes, la carga de trabajo permanece igual y se

modifica la configuración del sistema que se está probando. Las medidas suelen tomarse en función del rendimiento de la carga de trabajo y el tiempo de respuesta de las transacciones en línea. Las variaciones de la carga de trabajo suelen incluir la emulación del pico y el promedio de cargas de trabajo que se producen dentro de la tolerancia operativa normal.

**Perfil de rendimiento:** se trata de una prueba en la que se controla el perfil de tiempo del destino de la prueba, incluidos el flujo de la ejecución, el acceso de datos, las llamadas del sistema y de funciones para identificar y tratar los cuellos de botella de rendimiento y los procesos ineficaces.

**Prueba de configuración:** pruebas que se centran en garantizar que las funciones del destino de la prueba son las adecuadas en diferentes configuraciones de hardware y software. Esta prueba también se puede implementar como una prueba de rendimiento del sistema.

#### 4.6.3.4 Capacidad de soporte

**Prueba de instalación:** pruebas que se centran en garantizar que el destino de la prueba se instala correctamente en diferentes configuraciones de hardware y software, y en condiciones diferentes (como, por ejemplo, espacio de disco insuficiente o interrupciones de la alimentación). Esta prueba se implementa y ejecuta en aplicaciones y sistemas.

Al conocer los niveles y tipos de pruebas que pueden aplicársele a un producto de software, se puede seleccionar cuál es el tipo de prueba que se desarrollará para comprobar la calidad de la solución propuesta.

#### 4.6.4 Pruebas de rendimiento

En la Ingeniería del Software, las pruebas de rendimiento son aquellas que son realizadas para comprobar que tan rápido un sistema efectúa una tarea bajo ciertas circunstancias pre-planificadas de trabajo. Estas pruebas también son utilizadas para validar y verificar diferentes aspectos de la calidad de software, como por ejemplo, escalabilidad, fiabilidad y el buen uso de los recursos. Las pruebas de rendimiento constituyen un subconjunto de la Ingeniería de Pruebas, la cual se esfuerza en mejorar el rendimiento, basándose en el diseño y la arquitectura de un sistema, antes de la realización del proceso de codificación (32).

## 4.6.5 Diseño de pruebas de rendimiento

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

### 4.6.5.1 Estación de trabajo 1

- ✓ Sistema Operativo:
  - Ubuntu Release 10. 10 (maverick).
  - Kernel Linux 2.6.32-22 generic.
  - GNOME 2.32.0.
- ✓ Hardware:
  - ➔ Memoria: 984 MiB.
  - ➔ Procesador 0: Intel(R) Core(TM)2 Duo CPU E4500 @ 2.20GHz.
  - ➔ Procesador 1: Intel(R) Core(TM)2 Duo CPU E4500 @ 2.20GHz.
- ✓ Estado del sistema:
  - ➔ Espacio libre en disco: 6.2 GiB.

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

Cantidad de animaciones en transmisión.	Porcentaje consumido de la CPU.	Memoria consumida (MiB).
4	30	19.4
8	65	30.0
12	80	80,2

Tabla 4: Niveles de consumo de memoria y de la CPU para la estación de trabajo 1.

#### 4.6.5.2 Estación de trabajo 2

- ✓ Sistema Operativo:
  - Ubuntu Release 10.4 (lucid).
  - Kernel Linux 2.6.32-21 generic.
  - GNOME 2.30.0.
- ✓ Hardware:
  - ➔ Memoria: 4,00 GB(2,93 GB utilizables).
  - ➔ Procesador 0: IntelGenuine Intel(R) CPU T1600 @ 1.66GHz, 1662 MHz.
  - ➔ Procesador 1: IntelGenuine Intel(R) CPU T1600 @ 1.66GHz, 1662 MHz.
- ✓ Estado del sistema:
  - ➔ Espacio libre en disco: 20.6 GiB.
- ➔ **Niveles de consumo de memoria y de la CPU para la estación de trabajo 2.**

Cantidad de animaciones en transmisión.	Porcentaje consumido de la CPU.	Memoria consumida (MiB).
4	45	19.4
8	80	32.5
12	95	90.0

Tabla 5: Niveles de consumo de memoria y de la CPU para la estación de trabajo 2.

#### 4.6.6 Resultado de las pruebas de rendimiento

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

#### 4.6.7 Pruebas funcionales

Se denominan pruebas funcionales a las pruebas de software que tienen por objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados, es

común que este tipo de pruebas sean desarrolladas por analistas de pruebas con apoyo de algunos usuarios finales.

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

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

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

#### 4.6.8 Diseño de pruebas funcionales

Con el objetivo de validar el correcto funcionamiento del Subsistema Transmisión se realizaron pruebas de caja negra a los casos de uso Transmitir y Administrar información.

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

##### 4.6.8.1 Escenarios a probar en el caso.

Nombres	Escenarios	Descripción de la funcionalidad
CU 1: Transmitir.	EC 1: Transmitir canal.	El sistema verifica la señal a transmitir, y la transmite.

	EC 1.2: Transmitir canal falla.	El sistema transmite las informaciones correspondientes al patrón del canal.
--	---------------------------------	--

Tabla 6: Escenarios a probar en el caso de uso Transmitir

Nombres	Escenarios	Descripción de la funcionalidad
CU 1: Administrar información.	EC 1: Ac_Dato.	Se establece una lectura del archivo XML en busca de datos de conexión a la base de datos.
	EC 1.2: Ac_Dato falla.	Se emite un mensaje de alerta y se busca alternativas de transmisión.

Tabla 7: Escenarios a probar en el caso de uso Administrar información.

#### 4.6.8.2 Descripción de variables.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	<i>id</i>	Campo de texto (alfanumérico)	No	Se obtiene de la Base de datos
2	<i>nombre</i>	Campo de tipo cadena	No	Se obtiene de la Base de datos
3	<i>Link</i>	Campo de tipo cadena	No	Se obtiene de la Base de datos
4	<i>url_logo</i>	Campo de tipo cadena	No	Se obtiene de la Base de datos

5	<i>activado</i>	Campo de tipo cadena	No	Se obtiene de la Base de datos
---	-----------------	----------------------	----	--------------------------------

Tabla 8: Descripción de variables para el caso de uso Transmitir.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	<i>drivers</i>	Campo de tipo cadena	No	Se obtiene de un archivo XML
2	<i>host</i>	Campo de tipo cadena	No	Se obtiene de un archivo XML
3	<i>name</i>	Campo de tipo cadena	No	Se obtiene de un archivo XML
4	<i>user</i>	Campo de tipo cadena	No	Se obtiene de un archivo XML
5	<i>password</i>	Campo de tipo cadena	No	Se obtiene de un archivo XML
6	<i>port</i>	Campo de tipo entero	No	Se obtiene de un archivo XML

Tabla 9: Descripción de variables para el caso de uso Administrar información.

#### 4.6.8.3 Matriz de datos

[Matriz de datos para el caso de uso Transmitir](#)

[Matriz de datos para el caso de uso Administrar información](#)

#### 4.6.9 Resultado de las pruebas funcionales.

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

- Si no se encuentra un fichero XML con las informaciones generales del canal este pasa a la transmisión del patrón, sin poder cumplir la función principal del sistema.
- Los datos recibidos de la base de datos tienen que estar bien para poder definir la transmisión.

#### 4.6.10 Diseño de Casos de Prueba de caja blanca

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

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

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

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

$V(G) = NA$  (Número de Aristas) -  $NN$  (Número de Nodos) + 2.

$V(G) = P$  (Nodos predicados) + 1.

$V(G) =$  Número de regiones.

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

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

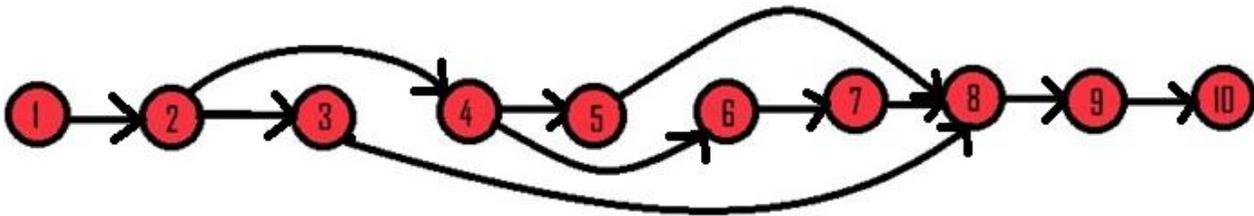
```
Signal::Signal (int p_id_signal)
{
1  id_signal = new int (p_id_signal);
2  if (p_id_signal==1)
3    name = new QString ("Canal Interno");
4  else if (p_id_signal==2)
```

```

5   name=new QString ("Patron");
6   else
7   name=new QString ("Senal Externa");
8   active = new bool (true);
9   link = new QString ("");
10  address_logo = new QString ("");
}

```

**Paso 1:** Generar Grafo de flujo de datos.



**Figura 12:** Grafo de flujo de datos Caso 1.

**Paso 2:** Cálculo de complejidad ciclomática

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

$V(G) = 11 - 10 + 2 = 3$

**Paso 3:** Caminos básicos

CB1: 1 2 3 8 9 10

CB2: 1 2 4 6 7 8 9 10

CB3: 1 2 4 5 8 9 10

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

CB3: 1 2 4 5 8 9 10

**Caso de prueba:** (Constructor de señal).

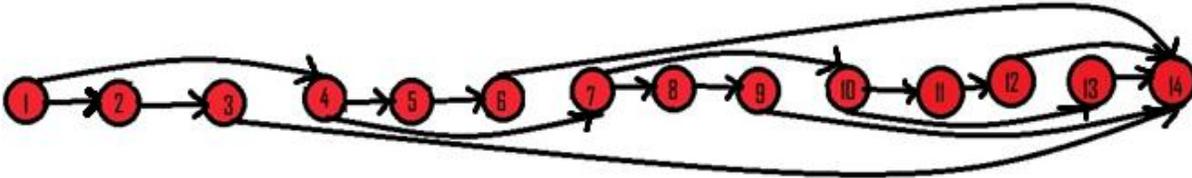
**Entrada:** posición 2.

**Resultado esperado:** Se determina al patrón como señal.

**Resultado de la Prueba:** Satisfactoria.

```
void Control::Message_received (QString args)
{
1   if (args=="Actualizar cambios")
    {
2       Update_changes ();
3       Verify_changes ();
    }
4   else if (args=="Cambiar a señal internal")
    {
5       Activar_senal (1);
6       emit Transmit_internal_channel ();
    }
7   else if (args=="Cambiar a patron")
    {
8       Activar_senal (2);
9       emit Transmit_pattern ();
    }
10  else if (args=="Cambiar a señal external")
    {
11     Activar_senal(3);
12     emit Transmit_external_channel();
    }
    else
13     qDebug("Nada");
14}
```

**Paso 1:** Generar Grafo de flujo de datos.



**Figura 13:** Grafo de flujo de datos Caso 2.

**Paso 2:** Cálculo de complejidad ciclomática

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

$V(G) = 17 - 14 + 2 = 5$

**Paso 3:** Caminos básicos

CB1: 1 2 3 14

CB2: 1 4 5 6 14

CB3: 1 4 7 8 9 14

CB4: 1 4 7 10 11 12 14

CB5: 1 4 7 10 13 14

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

CB1: 1 2 3 14

**Caso de prueba:** (Mensaje recibido).

**Entrada:** `args Actualizar cambios`.

**Resultado esperado:** Se manda a actualizar los cambios programados y luego se verifican.

**Resultado de la Prueba:** Satisfactoria.

## **4.7 Conclusiones.**

En este capítulo se mostraron los resultados de la etapa de construcción del sistema, realizándose un análisis completo de la solución que se está proponiendo y se elaboró la modelación de las clases del diseño ajustadas al framework que se utilizará. Se mencionan los principios de diseño y por último se muestra el modelo de despliegue y los diagramas de componentes. A partir del estudio realizado se pudo llegar a la siguiente conclusión. El proceso ingenieril, específicamente las fases de análisis y diseño de las funcionalidades expuestas propició una vía de solución para lograr la documentación necesaria que ayudará a la implementación en el desarrollo de la aplicación. Con el desarrollo de pruebas sobre la aplicación se obtuvieron sustanciales conocimientos sobre la aplicación, propiciando un criterio contundente para el desarrollo del próximo capítulo correspondiente a la valoración de un conjunto de expertos sobre la nueva versión confeccionada.

# Capítulo 5: Validación de la propuesta

## 5.1 Introducción

Para la validación y aceptación de la propuesta de la versión 2.0 del subsistema de transmisión para la Plataforma de Televisión Informativa, PRIMICIA, se utilizó el método Delphy, que es una técnica de investigación social que tiene como objetivo la obtención de una opinión grupal a partir del criterio de varios expertos.

En esencia, el método consiste en la realización de un diálogo anónimo entre los expertos, que son consultados individualmente mediante cuestionarios en aras de obtener un consenso general o, en caso contrario, conocer al menos los motivos de la discrepancia. La confrontación de opiniones se lleva a cabo mediante una serie de interrogantes sucesivas que permiten que la información obtenida pase por un proceso estadístico-matemático que permita la evaluación cuantitativa de los criterios.

## 5.2 Planificación del criterio de expertos

Para la planificación del criterio de expertos se plantea el objetivo a alcanzar, después se realiza la selección del tipo de Delphy a utilizar y se produce una buena selección de los especialistas como pasos previos a la aplicación directa de los criterios.

### **Objetivo a alcanzar**

El principal objetivo a alcanzar con la aplicación del método Delphy es que basándose en la opinión de un grupo de expertos, se logre la validación de la propuesta de solución planteada de una manera rápida e íntegra, dilucidando su efectividad y aplicación desde el punto de vista de los entendidos.

### **Selección del tipo de Delphy a utilizar**

El método Delphy se puede clasificar de diferentes maneras teniendo en cuenta varios aspectos; por objetivos, por conducción, entre otros tipos.

Para realizar la selección del tipo idóneo para esta investigación se tuvieron en cuenta las siguientes características: disminución de tiempo y flexibilidad de las respuestas.

Después de un estudio realizado se llega a la conclusión que el método Delphy conocido como Delphy cara-cara, es el seleccionado para realizar la validación de la solución propuesta.

En esta variante, el cuestionario se lleva personalmente a cada integrante del panel, a quien se le hace la encuesta en forma individual, lo cual permite aumentar la flexibilidad de las respuestas, pues el entrevistador puede resolver cualquier duda o ambigüedad que se le presente al panelista en relación a las preguntas del cuestionario. Por otra parte se logra considerables ventajas de tiempo y se logra disminuir el porcentaje de deserción de los expertos.

### **Selección de expertos**

Existen varios métodos para determinar la competencia de los expertos, entre los que se encuentran:

- Autovaloración
- Coeficiente de competencia K
- Efectividad de la actividad profesional

En este caso se aplicó el último de los mencionados por ser el que logra mayor objetividad en la evaluación del resultado y a la vez es el más cómodo para proceder a realizar la selección de los expertos.

Los expertos se seleccionaron según las siguientes características: labor que realiza, años de experiencia, especialidad, categoría docente, categoría científica y el coeficiente de competencia sobre el tema abordado de cada experto.

La elección de los expertos siguiendo las características antes mencionadas favorece agenciar resultados con calidad y aceptación, sumados a otros valores propios de estas personas como son: la honestidad, la responsabilidad y la integridad entre otros valores.

Esta selección se realizó además atendiendo a las posibilidades reales de participación de los candidatos, siendo todos profesionales de la UCI que tienen como promedio 3 años de experiencia en el proceso productivo de la Universidad. Poseen amplios conocimientos en temas relacionados con la propuesta a

evaluar, lo que ha permitido que las opiniones brindadas sean confiables y válidas para avalar el trabajo realizado.

### **5.3 Confirmación de la participación de los expertos**

Una vez conformado el panel de expertos se le comunicó a cada uno ellos, por diálogo directo, la intención de que participaran en el proceso de validación y aceptación de la propuesta realizada. En ese primer encuentro se les explicó cuáles eran los objetivos del proceso y qué consistía el trabajo. El cien por ciento de los invitados aceptó la tarea, por lo que se pasó a elaborar el listado final de los integrantes del panel

### **5.4 Elaboración del cuestionario**

Las preguntas del cuestionario se elaboraron con un enfoque investigativo. Inicialmente fueron de tipo abiertas, lo que permitió al encuestado brindar criterios sobre el tema en cuestión sin estar limitado, en sus respuestas, por un margen. Este tipo de preguntas puede ofrecer información importante de la que no se haya tenido referencias durante la investigación. Sin embargo también fueron incluidas preguntas de tipo cerradas, pues están enfocadas a intereses específicos de la investigación y desde este punto de vista aportan una mayor riqueza a las respuestas. ([Ver Anexo I: Encuesta a miembros del Panel de Expertos](#)).

### **5.5 Análisis y procesamiento de la información**

A la hora de procesar la información obtenida del cuestionario se tuvo en cuenta para cada pregunta si la misma era de tipo abierta o cerrada. En el caso de las abiertas se analizó cada una de las respuestas desde un punto de vista cualitativo, por lo que se realizó una valoración de los elementos más comunes. A las preguntas cerradas se le aplicó un análisis cuantitativo con la asociación de un valor numérico a cada criterio.

El cuestionario quedó compuesto por 5 preguntas que de forma general dan la posibilidad de conocer si podrían ser aplicados los resultados que se presentan en este trabajo. En este sentido, y con el fin de concretar el proceso de evaluación de la propuesta se tuvieron en cuenta los criterios expuestos en la pregunta # 3. A continuación se muestra cada uno de los planteamientos a evaluar:

- a) \_\_ El empleo del framework Qt en la solución PRIMICIA representa un valor agregado para el producto al poder brindar mejores interfaces gráficas, y completa soberanía tecnológica.
- b) \_\_ La inclusión de las noticias personalizadas en la nueva versión representa un avance, por la posibilidad de la edición de los contenidos a utilizar en la transmisión.
- c) \_\_ La comunicación directa entre subsistema permite mejor rendimiento en la aplicación y eficacia en las operaciones vinculadas principalmente con el acceso a la base datos.
- d) \_\_ La versión 2.0 del subsistema de transmisión agrega un mayor dinamismo e inmediatez en la gestión informativa por la posibilidad de contar con noticias e información en sentido general de carácter relevante o prioritario.
- e) \_\_ La arquitectura de la solución propuesta hace una solución más escalable y flexible al incorporar soporte de plugins o componentes, para la adición o modificación de funcionalidades.
- f) \_\_ La aplicación resulta ser más agradable, intuitiva, con inclusión de efectos y métodos de visualización de los contenidos.

Para cada planteamiento se estableció una escala de 1 a 5 para hacer corresponder con el nivel de importancia que el experto le atribuye, siendo 5 el valor de mayor significación. El proceso se realizó mediante la entrega personal del cuestionario a cada uno de los expertos, a los cuales se les pidió que lo completaran en un mínimo de tiempo, para más tarde realizar el análisis y procesamiento de los datos que se obtendrían de sus respuestas. La tabla que a continuación se muestra contiene los criterios emitidos por los diferentes expertos en relación a cada uno de los planteamientos de la pregunta 3.

Criterios	Expertos							
	E1	E2	E3	E4	E5	E6	E7	E8
a)	5	5	4	4	4	5	5	4
b)	5	4	5	5	5	5	5	5
c)	4	5	5	5	5	5	4	5
d)	4	4	4	5	5	5	5	5
e)	5	5	5	5	5	5	3	5
f)	5	4	4	5	5	5	5	5

Tabla 10: Criterios de expertos

Luego de ser asignados los valores a cada uno de los enunciados se procede a calcular el coeficiente de concordancia mediante la siguiente fórmula:

$$C_c = (1 - V_n / V_t) * 100$$

Donde:

C<sub>c</sub>: coeficiente de concordancia.

V<sub>n</sub>: cantidad de expertos en contra del criterio predominante.

V<sub>t</sub>: cantidad total de expertos.

Después de aplicar la fórmula se obtuvieron los siguientes valores:

Alternativas	a)	b)	c)	d)	e)	f)
R <sub>j</sub>	36	39	38	37	38	38
C <sub>c</sub>	50	87,5	75	62,5	87,5	75

Tabla 11: R<sub>j</sub> y C<sub>c</sub>

Siendo  $R_j = \sum C_{ij}$

C<sub>ij</sub>: Opinión del experto i para la alternativa j.

Una vez obtenido estos resultados se pasa a ordenar los planteamientos teniendo en cuenta el valor de R<sub>j</sub>, en orden descendente, pues el valor de mayor significación es 5. Para cada pregunta se entenderá por

consenso cuando un criterio acumula al menos el 70% de los votos. Ordenamiento resultante: E, B, C, F, A, D.

## 5.6 Resultados Graficados.

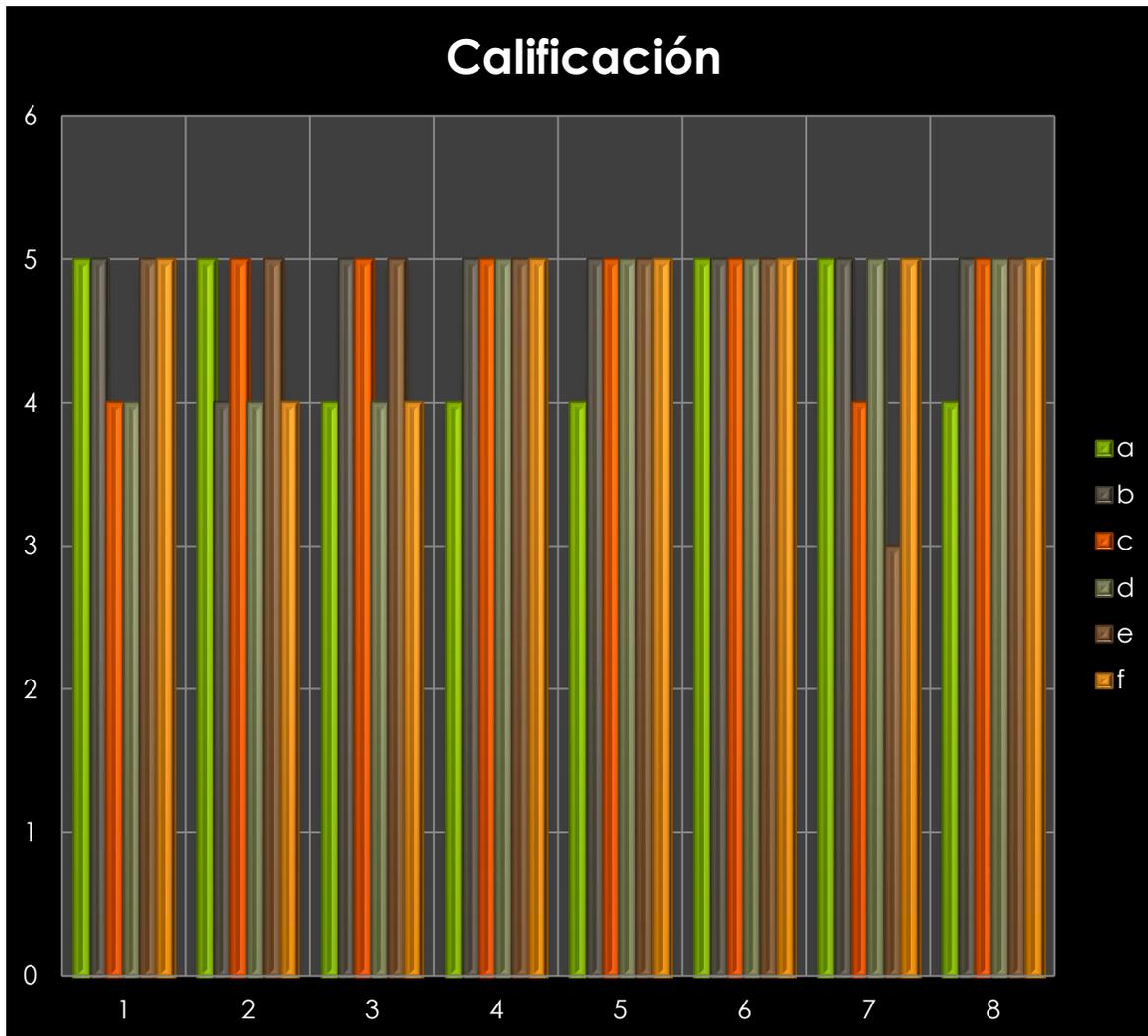


Figura 14: Encuesta.

En la representación se especifica un color por cada pregunta. En sentido general las respuesta con 5 representan el 62,5 %, las de 4 representan el 25 %, las de 3 un 2% y menos un cero por ciento.

Por estos datos se puede determinar que la solución en sentido general ha cumplido con su objetivo.

## 5.7 Conclusiones:

En este capítulo se ha validado la propuesta de la versión 2.0 del subsistema de transmisión para la Plataforma de Televisión Informativa, PRIMICIA, utilizando las pruebas realizadas en el capítulo anterior, así como el método Delphy mediante el panel de expertos. Los panelistas tuvieron un alto grado de concordancia en todas las preguntas por lo que solo fue necesario aplicar una sola ronda de entrevistas de donde se concluye según el criterio de los expertos que, la propuesta de versión 2.0 del subsistema de transmisión para Primicia es adecuada.

## Conclusiones generales

Con la realización de este trabajo de diploma se dio cumplimiento al objetivo general de desarrollar un subsistema de transmisión para la Plataforma de Televisión Informativa, Primicia, en su versión 2.0 que permitió dar solución a las necesidades de los clientes, en aras de alcanzar mayor inmediatez, dinamismo y personalización de la gestión informativa. Para alcanzar este resultado de forma satisfactoria se:

- Caracterizo la plataforma.
- Definieron los requisitos funcionales y no funcionales del sistema.
- Caracterizaron las herramientas y metodologías para el desarrollo a utilizar.
- Modelo el sistema haciendo uso de las herramientas de modelado.
- Implementaron las funcionalidades diseñadas.
- Valido la aplicación.

Además de las mejoras que aporta a PRIMICIA en cuestiones de flexibilidad, posibilidad de lectura de archivos XML, inclusión de transmisión en vivo, personalización de la noticia y escalabilidad, la solución brindada, representa un paso más hacia la emisión de informaciones de forma digital.

## Recomendaciones

A lo largo de este trabajo se apreció como se cumplieron cada uno de los objetivos trazados en el mismo. Además se observa claramente que el desarrollo obtenido potenciará La Plataforma de Televisión Informativa (PRIMICIA), no obstante, se realizan varias recomendaciones que darán continuación a dicho trabajo:

- Realizar un estudio de posibles mejoras para facilitar la adaptabilidad del sistema a las necesidades de diferentes clientes.
- Optimizar la codificación de la aplicación y actualizar la versión del framework empleado.
- Aumentar el conjunto de animaciones actualmente soportadas.
- Realizar un estudio del QML, en vista de la inclusión de futuras animaciones.
- Proponer la implantación de sistemas de este tipo en hospitales, universidades, hoteles, otras sedes empresariales y ministeriales.

# Bibliografía

1. **Romero, Milagros.** La televisión. *Monografias*. [En línea] [Citado el: 18 de Diciembre de 2010.] <http://www.monografias.com/trabajos13/televis/televis.shtml>.
2. **Terrero, Yaniley Méndez.** *Conceptualización del subsistema de transmisión de noticias como aplicación de escritorio para la Plataforma de Televisión Informativa*. Ciudad de la Habana : s.n., 2010.
3. **Solution, IMS. Interactive Media.** Preguntas Frecuentes. [En línea] 2010. [Citado el: 21 de Diciembre de 2010.] [http://www.intv.es/20/content\\_2060.html](http://www.intv.es/20/content_2060.html).
4. **Hernández García, Ruber, y otros, y otros.** PRIMICIA, Plataforma de Televisión. *Electroweb*. [En línea] 2010. [Citado el: 19 de Diciembre de 2010.] <http://electroweb.cujae.edu.cu/revista/index.php/rieac/article/view/52>.
5. Definición ABC una guía única en la red. [En línea] [Citado el: 20 de Diciembre de 2010.] <http://www.definicionabc.com/tecnologia/aplicacion.php>.
6. Definición Y Tipos De Aplicaciones Web. *Definición de Aplicación WEB*. [En línea] [Citado el: 19 de Diciembre de 2010.] <http://www.buenastareas.com/ensayos/Definicion-Y-Tipos-De-Aplicaciones-Web/317130.html>.
7. **Rios, Alexander.** Aplicaciones Web vs. Escritorio (Desktop) vs. legacy o consola. *Desarrollo de aplicaciones*. [En línea] 4 de Agosto de 2006. [Citado el: 19 de Diciembre de 2010.] <http://alxplus.blogspot.com/2006/08/aplicaciones-web-vs.html>.
8. **Zambrano Rodriguez, Douglas Francisco.** MULTIMEDIA. *Monografias*. [En línea] [Citado el: 22 de Diciembre de 2010.] <http://www.monografias.com/trabajos10/mmedia/mmedia.shtml>.
9. Aplicaciones multimedia. [En línea] [Citado el: 19 de Diciembre de 2010.] <http://www.agendistas.com/computadoras/aplicaciones-multimedia.html>.
10. Hacia la sociedad del conocimiento libre. *HispaLinux*. [En línea] [Citado el: 2010 de Diciembre de 2010.] <http://www.hispalinux.es/SoftwareLibre>.

11. LA DEFINICIÓN DE SOFTWARE LIBRE. *GNU Operating System*. [En línea] [Citado el: 20 de Diciembre de 2010.] [HTTP://WWW.GNU.ORG/PHILOSOPHY/FREE-SW.ES.HTML](http://www.gnu.org/philosophy/free-sw.es.html).
12. **Bermúdez Valdés, Geovanys**. *Propuesta de Sistema Operativo con kernel Linux para la Plataforma de Televisión Informativa, PRIMICIA*. Ciudad de la Habana : s.n., 2008.
13. FINGERTEXT (SISTEMA TELETETO). *Anglatécnic S. L.* [En línea] [Citado el: 19 de Diciembre de 2010.] <http://www.anglatecnic.com/es-6-Fingertext-%28Sistema-Teletexto%29.html>.
14. **Olivares Tamayo, Jorge Daniel y Rey Almaguer, Bernardo**. Desarrollo del Canal Informativo del Ministerio del Poder Popular para la Energía y Petróleo de Venezuela: Subsistema de Administración. *Presentación de la solución propuesta*. Ciudad de La Habana : s.n., 2008.
15. Rational Unified Process. *GSI*. [En línea] [Citado el: 15 de Diciembre de 2010.] <http://www.rational.com.ar/herramientas/rup.html>.
16. **Gomez Gallego, Juan Pablo y Galves, Jorge**. SOBRE EL PROCESO RACIONAL UNIFICADO. [En línea] 16 de Septiembre de 2007. [Citado el: 15 de Diciembre de 2010.] <http://www.scribd.com/doc/297224/RUP>.
17. Herramienta CASE para modelamiento UML. *Visual Paradigm for UML*. [En línea] 2010. [Citado el: 19 de Diciembre de 2010.] <http://www.todoprogramas.com/programalinux/visualparadigmforuml>.
18. Visual Paradigm For Uml. *slideshare*. [En línea] 2010. [Citado el: 19 de Diciembre de 2010.] <http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>.
19. Programación Java. *Lenguajes de programación*. [En línea] 2009. [Citado el: 19 de Diciembre de 2010.] <http://www.lenguajes-de-programacion.com/programacion-java.shtml>.
20. Java SAX2. *Java SAX2*. [En línea] 2010. <http://doc.qt.nokia.com/4.6/xml-sax.html>.
21. QXmlSchema. *QXmlSchema*. [En línea] 2010. <http://doc.qt.nokia.com/4.6/qxmlschema.html>.
22. **Lattre, Alexis**. *VideoLAN Streaming Howto*. s.l. 2005.

23. QImage. *QImage*. [En línea] 2010. [Citado el: 23 de Noviembre de 2010.] <http://doc.qt.nokia.com/4.6/qimage.html>.
24. QTimer. [En línea] 2010. [Citado el: 23 de Noviembre de 2010.] <http://doc.qt.nokia.com/4.6/qtimer.html>.
25. **Garcerant, Iván**. Modelo de Dominio. [En línea] 10 de Octubre de 2007. [Citado el: 22 de Enero de 2010.] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
26. **Herrera J, Lizka Johany**. Ingeniería De Requerimientos Ingeniería De Software 1. *Monografias 1*. [En línea] [Citado el: 22 de Enero de 2010.] <http://www.monografias.com/trabajos6/resof/resof.shtml>.
27. **Andrés González, Carlos de Jesús**. *Diseño del Subsistema de Configuración de la Plataforma de Televisión Informativa, PRIMICIA*. Ciudad de La Habana : s.n., 2010.
28. **Informáticas., Universidad de las Ciencias**. Flujo de trabajo de requerimientos. La Habana : s.n., 2007.
29. **IMS**. Preguntas Frecuentes. IMS. Interactive Media Solution. [En línea] 2007. [Citado el: 10 de 1 de 2007.] [http://www.imsol.es/index\\_2.html](http://www.imsol.es/index_2.html).
30. **Vaughan, Tay**. [aut. libro] Hill McGraw. *Todo el poder de la Multimedia*. México D.F : s.n., 1994, pág. 561.
31. **Solution, IMS. Interactive Media**. Preguntas Frecuentes. [En línea] [Citado el: 17 de Diciembre de 2010.] [http://www.intv.es/20/content\\_2060.html](http://www.intv.es/20/content_2060.html).
32. **Magazine, Master**. Actualidad Informática, Diccionario, Definición de Teletexto. [En línea] 2007. [http://www.ucm.es/info/emp/Numer\\_06/6-4-Inve/6-4-08.htm](http://www.ucm.es/info/emp/Numer_06/6-4-Inve/6-4-08.htm).
33. Planeta codigo. *Planeta codigo*. [En línea] 2010. [Citado el: 19 de Diciembre de 2010.] <http://www.planetacodigo.com/>.
34. **Larman, Craig**. UML y Patrones. Vol. 2.