



Universidad de las Ciencias Informáticas
Facultad 9

“Desarrollo del Canal Informativo del Ministerio del Poder Popular para la Energía y Petróleo de Venezuela: Subsistema de Transmisión”

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

Autor:

José Andrés Hernández Bustio

Tutor:

Ing. Ruber Hernández García

Ciudad de La Habana, junio 2008
“Año 50 de la Revolución”

***"Los conceptos y principios fundamentales de la ciencia son
invenciones libres del espíritu humano."***

Albert Einstein

A José Andrés...

A mi mamá Yoyi...

A mi papá José Andrés...

A mi familia...

A los amigos de toda la vida...

AGRADECIMIENTOS

A todos los profesores que he tenido a lo largo de mi vida estudiantil, que de una forma u otra me han formado e influido en todos los logros que he obtenido en ella, especialmente María Elena y Tomás.

A toda mi familia por darme su cariño y confiar en mí en todo momento.

A mis amigos que han estado presente en el momento que los he necesitado.

A mis compañeros de grupo por enseñarme como se puede ser mejor persona.

A todos los integrantes de mi proyecto por demostrar seriedad en la realización del trabajo que se nos encomendó, especialmente a Geovanys que trabajo en el subsistema conmigo.

A Ruber por ayudarme en todo momento y ser más que tutor, mi amigo, compañero.

A dos personas que no por ponerlas de último son las menos importantes, al contrario son la razón de mi existir y mi fuente de inspiración. Son las personas que se merecen todo mi esfuerzo para seguir adelante y que ninguna meta sea imposible para mí. Son los que han hecho de mí una persona de bien y a los que le debo todo en mi vida. Son lo que más quiero, para mis padres...

MUCHAS GRACIAS...

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) y al Ministerio del Poder Popular para la Energía y Petróleo de Venezuela (MENPET) para que hagan el uso que estimen pertinente con este trabajo.

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

Autor: José Andrés Hernández Bustio

Tutor: Ing. Ruber Hernández García

OPINIONES Y AVALES

OPINIÓN DEL TUTOR SOBRE EL TRABAJO DE DIPLOMA

El tutor del presente Trabajo de Diploma considera que durante el período que se evalúa el estudiante mostró las cualidades que a continuación se detallan.

Durante las etapas de investigación y desarrollo el estudiante ha demostrado un alto nivel de independencia y laboriosidad, expresado en los aspectos que ha tenido que estudiar para desarrollar los objetivos del trabajo, entre los que se encuentran contenidos de Televisión, Periodismo, *software* libre, Video *Streaming*, entre otros que han contribuido a la formación profesional e investigativa del mismo.

Las soluciones a los diferentes obstáculos que aparecieron demostraron una elevada creatividad y originalidad que se reflejó en el trabajo individual y en la relación con el resto del colectivo del proyecto que tributa a este trabajo de diploma, apoyado en un arduo y continuo trabajo que demostró la gran responsabilidad asumida por el estudiante ante las tareas que impuso el desarrollo de la tesis.

En la obtención de la solución el autor mostró una gran receptividad ante las críticas y sugerencias efectuadas, así como capacidad para apropiarse de los conocimientos que necesitaba para enfrentar las dificultades surgidas en el proceso de creación, logrando una alta experiencia en el desarrollo de *software* para la TV.

El documento realizado contiene la documentación de las etapas de análisis y diseño del sistema, cumpliendo con los requisitos planteados para este tipo de trabajo.

El trabajo posee una gran calidad científico-técnica reflejada en las distintas presentaciones en eventos científicos que se han realizado, así como en el empleo, modificación y reutilización de tecnologías libres, para contribuir de esta manera al perfeccionamiento del producto que se comercializa y a la soberanía tecnológica del país. La solución tiene un alto valor agregado para la entidad cliente la que sin dudas se beneficiará mucho de este producto con pos de la revolución que llevan a cabo. Por otra parte es válido mencionar que el proyecto generó como aporte bruto al país más del 85% del monto total con su desarrollo.

Por todo lo anteriormente expresado considero que el estudiante ha vencido las tareas previstas para el corte, así como cumplido los objetivos propuestos; y propongo que se le otorgue al Trabajo de Diploma en el actual corte de tesis, la calificación de **5**.

Tutor: Ing. Ruber Hernández García

Firma

Fecha

RESUMEN

Es una necesidad primaria de los seres humanos mantenerse informados y así estar al tanto del entorno que los rodea. Muchas han sido las formas de comunicación que se han apreciado a lo largo de la historia de la humanidad, donde la necesidad de perfeccionarlas han dado origen a medios tan avanzados como la radio y la televisión (TV), destacándose este último como uno de los principales medios de comunicación masiva.

Sin embargo en ocasiones existen situaciones especiales que imposibilitan la transmisión de informaciones en distintos formatos a una gran masa de personas. En este caso se pueden encontrar entornos cerrados con una densidad poblacional significativa, siendo necesario transmitir informaciones en formato texto en tiempo real o dinámicamente; ejemplos de estos son sedes ministeriales o empresariales, universidades, centros de convenciones, hoteles y terminales de transportación.

En el presente trabajo se definen los conceptos que soportan el desarrollo del subsistema de transmisión de un canal de televisión interno, para la sede del Ministerio del Poder Popular para la Energía y Petróleo de Venezuela (MENPET), con el uso de tecnologías informáticas, posibilitando de este modo la integración de diferentes informaciones textuales y audiovisuales capaces de actualizar informacionalmente a los usuarios de manera rápida y constante. Se describen además las características que presentará la solución dada.

PALABRAS CLAVES: televisión, canal, informativo, teletexto, audiovisuales, tv.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1	4
Introducción.....	4
1.1 Conceptos relacionados con la solución.....	4
1.1.1 Procesos de transmisión en canales de Televisión.....	4
1.1.2 Teletexto.....	5
1.2 Objeto de Estudio.....	6
1.2.1 Descripción General.....	6
1.2.2 Descripción actual del dominio del problema.....	7
1.2.3 Situación Problemática.....	7
1.3 Análisis de otras soluciones existentes.....	8
1.3.1 Sistemas de Teletexto en Europa.....	8
1.3.2 Subsistemas de transmisión de canales informativos en el área nacional.....	10
Conclusiones.....	11
CAPÍTULO 2	12
Introducción.....	12
2.1 Descripción de las tendencias y tecnologías actuales sobre las que se apoya la propuesta.....	12
2.1.1 Aplicaciones Multimedia.....	12
2.1.2 AJAX.....	13
2.1.3 Streaming.....	16
2.2 Lenguajes de programación.....	18
2.2.1 Lenguajes de programación del lado cliente.....	18
2.2.2 Lenguajes de Programación del lado servidor.....	20
2.3 Framework.....	22
2.3.1 CakePHP.....	23
2.3.2 Zend Framework.....	24
2.3.3 Symfony.....	25
2.3.4 Fundamentación del <i>framework</i> escogido.....	28
2.4 Sistema de Gestión de Bases de Datos.....	29
2.4.1 MySQL.....	29
2.4.2 PostgreSQL.....	30
2.4.3 Fundamentación del SGBD escogido.....	31
2.5 Metodologías de desarrollo de software.....	32
2.5.1 Programación Extrema.....	32
2.5.2 Proceso Unificado de Desarrollo.....	33
2.5.3 Fundamentación de la metodología de desarrollo de software escogido.....	34
2.6 Herramientas CASE para la modelación UML.....	35
2.6.1 ArgoUML.....	35
2.6.2 VisualParadigm.....	36
2.6.3 Fundamentación de la herramienta CASE para la modelación UML escogido.....	36
2.7 Sistemas de Streaming.....	36
2.7.1 MPEG4IP.....	37
2.7.2 FFMPEG.....	37
2.7.3 VideoLAN.....	38

2.7.4	Fundamentación de la aplicación de Streaming escogida.....	39
2.8	Sistema operativo.....	40
2.8.1	Nova LNX.....	40
	Conclusiones.....	41
CAPÍTULO 3	42
	Introducción.....	42
3.1	Modelo de Negocio.....	42
3.1.1	Procesos del Negocio.....	42
3.1.2	Actores del negocio.....	43
3.1.3	Diagrama de Casos de Uso del Negocio.....	43
3.1.4	Descripción textual de los Casos de Uso de Negocio.....	44
3.1.5	Diagramas de Actividades para cada Caso de Uso del Negocio.....	46
3.2	Requerimientos Funcionales.....	46
3.3	Requerimientos No Funcionales.....	47
3.4	Descripción del Sistema Propuesto.....	49
3.4.1	Descripción de los actores del sistema.....	49
3.4.2	Casos de Uso del Sistema.....	50
3.4.3	Descripción textual de casos de usos del sistema.....	50
CAPÍTULO 4	56
	Introducción.....	56
4.1	Diagramas de Clases del diseño.....	56
4.1.1	Diagrama de Clases del diseño “Subsistema de transmisión”.....	57
4.2	Principios de diseño.....	58
4.2.1	Estándares de la interfaz.....	58
4.3	Estándares de Codificación.....	59
4.4	Diseño de la base de datos.....	59
4.4.1	Diagrama de clases persistentes.....	60
4.4.2	Modelo de datos.....	61
4.5	Modelo de Despliegue.....	61
4.6	Modelo de Implementación.....	62
	Conclusiones.....	63
CONCLUSIONES	64
RECOMENDACIONES	65
REFERENCIAS BIBLIOGRÁFICAS	66
GLOSARIO DE TÉRMINOS	68
ANEXOS	70

ÍNDICE DE TABLAS

Tabla 1. Límites físicos de PostgreSQL.	31
Tabla 2. Descripción de los actores del negocio.	43
Tabla 3. Descripción detallada del CU del negocio Transmitir canal.	45
Tabla 4. Requerimientos de Hardware.	49
Tabla 5. Descripción de los actores del sistema.	49
Tabla 6. Descripción detallada del CU del sistema Transmitir canal.	54
Tabla 7. Descripción de la BD: Tabla infocinta.	71
Tabla 8. Descripción de la BD: Tabla noticia.	71
Tabla 9. Descripción de la BD: Tabla música.	72
Tabla 10. Descripción de la BD: Tabla sección.	72
Tabla 11. Descripción de la BD: Tabla pantalla.	73
Tabla 12. Descripción de la BD: Tabla genero.	73
Tabla 13. Descripción de la BD: Tabla video.	74
Tabla 14. Descripción de la BD: Tabla imagen.	74
Tabla 15. Descripción de la BD: Tabla país.	75
Tabla 16. Descripción de la BD: Tabla noticias_publicadas.	75
Tabla 17. Descripción de la BD: Tabla cambios_programados.	76
Tabla 18. Descripción de la BD: Tabla senal.	76
Tabla 19. Descripción de la BD: Tabla tipo_pantalla.	76
Tabla 20. Comparación entre <i>framework</i> para PHP.	78

ÍNDICE DE FIGURAS

Figura 1. Imagen del Sistema de Teletexto de Televisión Española.....	9
Figura 2. Tecnologías agrupadas bajo el concepto de AJAX.	14
Figura 3. Tipo de transmisión <i>unicast</i>	16
Figura 4. Tipo de transmisión <i>multicast</i>	17
Figura 5. Características de otros <i>framework</i> presentes en Symfony.....	26
Figura 6. Diagrama del proceso de transmisión.	43
Figura 7. Diagrama de Casos de Usos del Negocio.....	44
Figura 8. Diagrama de Actividades para el CU del Negocio Transmitir canal.....	46
Figura 9. Diagrama de CUS Subsistema de Transmisión.	50
Figura 10. Diagrama de Clases del diseño “Subsistema de transmisión”.....	57
Figura 11. Diseño de Interfaz: Subsistema de Trasmisión.	58
Figura 12. Diagrama de clases persistentes.	60
Figura 13. Modelo de datos.	61
Figura 14. Modelo de despliegue.	62
Figura 15. Diagrama de componentes del modelo de implementación.	63
Figura 16. Página de estado del FFServer.....	70

INTRODUCCIÓN

A lo largo de la historia del ser humano se ha visto como ha logrado evolucionar sus formas de comunicación: desde rudimentarios métodos como la escritura jeroglífica, pasando por la invención del alfabeto y del papel, dando un leve salto hasta la llegada de la imprenta, y apenas uno más para la aparición del teléfono, el cine, la radio, la televisión. Todos estos medios han sido ciertamente un avance en las formas de comunicación del hombre y tienen un papel fundamental en la socialización de los ciudadanos.

En esta evolución de los medios de comunicación algunos han contado con mayor preferencia por las personas, apoyándose en las características que tienen cada uno de ellos. Sin lugar a duda la televisión se destaca dentro del grupo, al tener la posibilidad de vincular a las informaciones el poder seductor de la imagen y el sonido. Aprovechando las facilidades que brinda este medio la Oficina de Comunicación y Relaciones Institucionales del MENPET transmite un canal de televisión interno con recepción en diferentes locales de su sede que comprende salas de espera de elevadores, comedores y oficinas; donde se colocan algunos materiales informativos y se encadena mayormente con la televisora del gobierno Venezolana de Televisión (VTV) y un canal que se transmite en la Sede Central de Petróleos de Venezuela, SA (PDVSA).

Este canal complementa la labor informativa-formativa sobre los trabajadores y visitantes del ministerio, que comprende distintos temas relacionados con la actividad ministerial y presidencial en el ámbito nacional e internacional. Las informaciones son publicadas además en otros medios de divulgación con que cuentan, los que incluyen la publicación escrita y digital.

Por la importancia que tiene la difusión de las informaciones que se brindan por el canal de televisión con todo el proceso revolucionario que se lleva a cabo en Venezuela y el desarrollo institucional, la principal tarea de este trabajo estaría encaminada a resolver el siguiente **problema**:

El MENPET no cuenta con una aplicación que permita la trasmisión automática y eficiente de las informaciones generadas para su canal de televisión interno.

A partir de esta problemática se espera, como aportes prácticos, dotar al MENPET de una aplicación informática capaz de transmitir la información que será visualizada a través de la televisión interna, de esta manera se favorecen los trabajadores que en ella radican y visitantes que acuden al centro.

En este trabajo es **objeto de estudio** los procesos de transmisión del canal de televisión interno del MENPET. El **campo de acción** está centrado en la automatización de los procesos de transmisión del canal de televisión interno del MENPET.

El sistema informático resultado del presente trabajo tendrá una gran repercusión en la División de Prensa, al informatizar el proceso de transmisión de su canal de televisión, además de brindar el servicio las 24 horas del día. También se logrará que los trabajadores de la institución puedan tener un medio para su información y mucho más importante para su formación.

Se propone como **objetivo general** desarrollar un Sistema Automatizado para el MENPET que facilite la transmisión automática y eficiente de las informaciones generadas para su canal de televisión interno utilizando *software* libre.

Para cumplir el objetivo general del trabajo se trazan los siguientes **objetivos específicos**:

- Investigar sobre las plataformas existentes y las herramientas necesarias para la implementación de la solución.
- Realizar el análisis y diseño del sistema.
- Implementar el Módulo de Trasmisión del canal informativo.

Para dar cumplimiento a los objetivos planteados anteriormente se deben cumplir las siguientes **tareas**:

- Investigar sobre el sistema operativo Linux.
- Investigar sobre lenguajes de Programación, *frameworks* y entornos de desarrollo integrado
- Investigar sobre herramientas para trabajo con video a demanda en *software* libre.
- Capturar y clasificar de requisitos funcionales.
- Capturar y clasificar de requisitos no funcionales.
- Identificar, clasificar y describir casos de uso del sistema.
- Estructurar el modelo de sistema.
- Estructurar modelo de diseño.
- Implementar el sistema.

Para el presente trabajo se realizó una investigación en la que se utilizaron algunos de los métodos de investigación, tanto teóricos como empíricos, existentes. La investigación se dividió en distintas partes para llevarla a cabo, donde posteriormente, delimitando las relaciones que existían entre ellas, se unieron como un todo (*Analítico sintético*). Por otra parte se trazó la estrategia de estudiar algunos sistemas que hayan existidos, que tengan algo en común con lo que se pretende hacer, aprovechando de ella algunos elementos teóricos que aportaran para el desarrollo de la aplicación (*Análisis Histórico – lógico*).

Para obtener una visión real de la situación existente, en la problemática planteada, determinando, clasificando y consignando los acontecimientos pertinentes de acuerdo con el esquema previsto se utilizó la *Observación*, así como para obtener información valiosa para la investigación, brindada de forma directa por el cliente se realizaron una serie de *entrevistas*, siendo este otro método utilizado en la investigación.

CAPÍTULO 1

Fundamentación Teórica

Introducción.

En este capítulo se aborda como objetivo fundamental, los aspectos que se utilizan como soporte teórico del subsistema diseñado. Se realiza un análisis de la forma en que se lleva a cabo el proceso de transmisión de los canales de televisión. De igual manera se describe la situación actual en la transmisión del canal interno del MENPET y por último se analizan algunas soluciones existentes en el ámbito internacional y nacional que se encuentran en el campo del problema planteado.

1.1 Conceptos relacionados con la solución.

1.1.1 Procesos de transmisión en canales de Televisión.

Haciendo una breve referencia a la historia de la televisión, es necesario referenciar los años 20 del siglo pasado donde surgen los dos primeros modelos: por un lado, la televisión mecánica, por otro, la electrónica. Ambas se desarrollaron de forma paralela y accidentada en un período caracterizado por la lucha, fundamentalmente en los Estados Unidos y en Gran Bretaña, entre distintas compañías e inventores por la adopción de un estándar técnico en los sistemas de difusión y recepción de imágenes. Frente a la competencia existente, el Gobierno británico decide nombrar una comisión investigadora para definir la posición del Estado en materia televisiva. En enero de 1935, dicha comisión adoptó una definición mínima de 240 líneas y 25 imágenes por segundo de forma que el sistema mecánico quedaba relegado frente a la calidad de las imágenes electrónicas.

Los años 50 han sido calificados por muchos analistas como aquellos del “gran salto de la televisión en el mundo”, puesto que es entonces cuando los servicios regulares de televisión se extendieron gradualmente por las grandes urbes del mundo. En este período surgen numerosas emisoras para brindarle el servicio a la población, solamente en los Estados Unidos se estima para 1952 unas 108, alimentando las pantallas de unos 21 millones de televisores.

Actualmente el proceso de transmisión de las televisoras puede ser de forma tradicional o automatizada. La tradicional consiste en operar un *switch*, al cual llegan señales de video de distintas

fuentes como son, grabadoras de cintas de video (VTR, siglas en inglés), señales directas del Estudio o de otras televisoras. La persona encargada de realizar estas operaciones se guía por una escaleta¹ que fue confeccionada previamente, especificando el orden en que se mostrará cada espacio y el tiempo de duración. En el territorio nacional el canal Cubavisión esta implementado de esta manera.

El avance de la informática, tanto a nivel del *hardware* como del *software* han dado la posibilidad de crear sistemas que automatizan el proceso de transmisión, siendo posible gestionar la escaleta, tener control total de las líneas de exterior y entradas de satélites, con grabación automática, titulación y gráficos, control de VTRs, reproducción de videos desde disco duro en formatos como MPEG-2², AVI³ entre otros. Algunos de los sistemas que permiten estos procesos son *Vsnmatic*⁴, *VideoMat MPro+*⁴ y *TVmaker*⁴.

1.1.2 Teletexto.

El teletexto se define como un sistema de transmisión de textos escritos por medio de la televisión (18). El ingeniero de la BBC⁴ Peter Rainger es el que tiene la iniciativa de crear estos sistemas, realizándose las primeras emisiones en el año 1973. En los inicios solo se pretendía buscar un sistema que fuera capaz de emitir rótulos o subtítulos en los televisores para ser leídos por personas con discapacidad auditiva, pero resultó que se convertiría en un nuevo medio de comunicación.

El teletexto permite acceder a una información codificada a través del mando a distancia. El contenido editorial convive en sus páginas ofreciendo al receptor una completa información de toda índole, fácil de consultar y de forma gratuita. Para acceder a ella es necesario disponer de un televisor con esta facilidad (9). La característica que marca la diferencia a la hora de hacer una comparación con la prensa tradicional es que la información que se muestra está en constante renovación. Las secciones más destacadas como son las noticias Nacionales, Internacionales, Económicas, Culturales o Deportivas se actualizan prácticamente al instante (9).

¹ nombre para el guión de televisión y radio.

² es la designación para un grupo de estándares de codificación de audio y video acordado por el grupo de expertos en imágenes en movimiento (MPEG, siglas en inglés), y publicados como estándar ISO 13818.

³ audio y Video Entrelazados (AVI, siglas en inglés). Es el formato estándar más utilizado en el mundo del video digital para Windows que utiliza para almacenar video con sonido incorporado.

⁴ Ver glosario de términos.

Los usos y las posibilidades del teletexto son incontables. En principio, todo aquello que sea capaz de despertar el interés del usuario tiene cabida en este medio. Los temas que interesan a una gran mayoría se jerarquizan, aunque el interés de la minoría también es tenido en cuenta. Si a esto se añade que la información se recibe de forma inmediata y en el momento en que se desee en la pantalla del propio televisor, se aumentan las opciones de uso del teletexto de forma ilimitada (9).

1.2 Objeto de Estudio.

1.2.1 Descripción General.

La Oficina de Comunicación y Relaciones Institucionales del MENPET tiene entre sus funciones la programación, diseño, coordinación y realización de documentales y reportajes de carácter divulgativo, tanto escritos como audiovisuales, a nivel nacional, de la labor que ejecute el Ministerio. La División de Prensa ejerce esta labor informativa a través de diferentes medios de divulgación con que cuentan, los que incluyen la publicación escrita, digital y la emisión de un canal televisivo en distintos locales de su sede central.

Seguidamente se expone cómo se realiza la transmisión del canal interno de televisión del MENPET de manera general:

- Los materiales a transmitir por el canal son grabados en casetes de video VHS⁵ desde una computadora.
- Los materiales disponibles en video son alternados con la señal en vivo de VTV y el canal interno de la sede de PDVSA.
- El encargado de las transmisiones del canal, utilizando un video VHS, reproduce los materiales seleccionados. Cuando se desea cambiar a la televisión en vivo se cambia el video de canal, operaciones que son visualizadas en los telerreceptores.
- La señal es monitorizada mediante dos televisores ubicados en el local de transmisiones, el cual no tiene las condiciones necesarias para estas operaciones.

⁵ sistema de video casero (VHS, siglas en inglés) es un sistema de grabación de audio y video analógico.

1.2.2 Descripción actual del dominio del problema.

La División de Prensa del MENPET, adscrita a la Oficina de Comunicación y Relaciones Institucionales, se encarga de coordinar la elaboración y difusión de avisos, notas de prensa, comunicados, notificaciones, documentos y otros servicios escritos y audiovisuales dirigidos al público, donde se abordan aspectos vinculados a la gestión del Ministerio en materia de hidrocarburos, energía en general, petroquímica, carboquímica, similares y conexas.

La Oficina de Comunicación y Relaciones Institucionales ejecuta sus programas comunicacionales; de publicidad y relaciones públicas utilizando los diferentes medios que posee, entre los que se encuentran la Red de Televisión Interna, con receptores distribuidos en la recepción, salas de espera de elevadores, comedores y oficinas; la que es de vital relevancia en el dominio del problema que se presenta. Además publica la revista "Petróleo y Revolución", publicada trimestralmente; una Cartelera Informativa impresa con una frecuencia semanal que se encarga de abordar la actividad presidencial; el Portal Web del Ministerio; folletos y plegables informativos.

Los materiales audiovisuales transmitidos por la red de televisión del ministerio son editados por un equipo de realización integrado por periodistas, editores de video y diseñadores gráficos. A pesar que el canal de televisión se transmite regularmente no cumple con las expectativas y necesidades de la División de Prensa, al no poder tener una constante presencia informativa.

1.2.3 Situación Problemática.

La Oficina de Comunicación y Relaciones Institucionales del MENPET, junto a su División de Prensa, tiene como misión difundir a sus trabajadores y a diferentes sectores de la población información sobre la formación política, la conservación del medio ambiente, el cambio climático, el uso de energías alternativas, la revolución energética, la nacionalización petrolera y otros temas de la actividad ministerial.

La División de Prensa transmite a través de la red de televisión interna un canal donde se publican, sin una programación definida, materiales audiovisuales y se alternan con la emisión de la señal en vivo de VTV y el canal interno de la sede de PDVSA.

El canal interno del ministerio es transmitido manualmente con la utilización de un video y dos televisores de referencia. Los materiales son grabados en un casete de video para ser reproducidos posteriormente. Las operaciones de cambios de canales y materiales al ser realizadas de forma manual son visualizadas en los receptores.

La Oficina de Comunicación y Relaciones Institucionales desde su labor informativa - formativa tiene interés en transmitir informaciones en formato texto a través de la televisión para de esta manera poder lograr una programación regular del canal, sin embargo el ministerio no tiene hoy una infraestructura que permita lo expresado anteriormente.

Finalmente se puede plantear que la División de Prensa de la Oficina de Comunicación y Relaciones Institucionales del MENPET, no tiene una herramienta que les permita la transmisión automática y eficiente de su canal interno en correspondencia con los intereses institucionales.

1.3 Análisis de otras soluciones existentes.

Para lograr transmitir información rápida haciendo uso de la TV como soporte y teniendo como base un sistema automático, existen en el mundo actualmente algunas soluciones. Ejemplo son los sistemas de teletexto que se difunden en las cadenas de televisión digital de países europeos.

Otras soluciones son los Canales Informativos en el ámbito nacional, *Señal 3* de la Universidad de las Ciencias Informáticas (UCI) y *Señal ACN* de la Agencia Cubana de Noticias (ACN), los cuales están orientados a brindar informaciones las 24 horas a través de un sistema de teletexto adaptado.

Seguidamente se explican y analizan las características de estas soluciones de manera detallada para una mejor comprensión del tema.

1.3.1 Sistemas de Teletexto en Europa.

La mayor parte de los sistemas de teletextos existentes en el mundo se concentran en Europa, países como España, Inglaterra, Suiza, Holanda e Italia se destacan en un grupo de más de 45 países.

La emisión del 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 (9). Esta característica limita en gran parte la información que se puede emitir, admitiendo sólo textos y mínimamente gráficos. Al poseer este inconveniente no son capaces de satisfacer los requerimientos planteados para el subsistema a desarrollar debido a que se necesita un producto que resulte atractivo para los televidentes y que permita la utilización de imágenes y materiales en formato de video.

Para que se emitan las páginas, los sistemas están permanentemente conectados a través de una línea de datos con el centro de transmisión de la cadena televisiva; las páginas se almacenan allí en

una máquina que automáticamente mezcla la señal de audio y video con la del teletexto, proporcionando así una emisión en línea (*on-line*).

Cuando se selecciona la opción de teletexto de las cadenas que lo tienen implementado aparece de manera automática en la pantalla una página, donde se encuentra referencias a las distintas secciones. Cada página se identifica con un número de tres cifras, que figura en la línea de encabezamiento junto a la fecha y la hora exacta (9). Para visualizar una noticia debe seleccionarse a través del mando el número de página que le corresponde a la noticia para que esta sea mostrada.



Figura 1. Imagen del Sistema de Teletexto de Televisión Española.

Estos sistemas de teletexto implementados en Europa, ofrecen enormes ventajas desde el punto de vista de la televisión digital, organización por secciones y actualización automática mediante un software que facilita este trabajo. Todas estas características se ajustan a las exigencias de la institución cliente, sin embargo, la posibilidad que le brindan al usuario de decidir mediante su mando a distancia qué noticias desea ver dentro del sinnúmero de informaciones que se ofrecen, hacen que no se ajuste para darle solución a la problemática planteada, debido que en ella se requiere que las noticias sean transmitidas sin la necesidad de la interactividad con el usuario.

1.3.2 Subsistemas de transmisión de canales informativos en el área nacional.

En el país se cuenta con dos principales antecedentes la solución que se propone, estos son: el subsistema de transmisión de *Señal 3*, canal Informativo de la UCI y el de *Señal ACN* de la Agencia Cubana de Noticias, los cuales ofrecen importantes aportes al presente trabajo por su función informativa y los procesos del negocio que se implementan. Estos subsistemas tienen como elemento común la posibilidad de brindar información a los usuarios las 24 horas del día, utilizando como base los teletextos, con el empleo además, de imágenes y videos.

Los subsistemas de transmisión de ambos canales son capaces de conectarse automáticamente a una base de datos obteniendo las noticias y recursos que posteriormente visualiza en dependencia de las secciones activas y el tipo de pantalla que presente la noticia, las cuales pueden ser de Texto, Imagen, Texto-Imagen o Video. Para que sea más placentera la lectura se reproduce música instrumental y en la pantalla se muestra el nombre del tema y su autor como información adicional. Esta señal televisiva es difundida, mediante la red interna de televisión por cable (*Señal 3*) y por la Plataforma de Televisión Satelital (*Señal ACN*).

Estos sistemas resuelven parcialmente el problema presente, tienen características que se aproxima a lo deseado por el cliente como son: trasmisión de texto a través de la televisión y la utilización de recursos multimedia, su organización por secciones temáticas, el empleo de distintos tipos de pantallas para las noticias y un fondo musical.

Sin embargo, la existencia de algunas particularidades en el problema planteado imposibilita resolver el mismo con estos subsistemas. El enlace en vivo con las televisoras nacionales, no es contemplado en los mismos. Además para la realización de la solución se utilizaron *software* como el Director, creado por Macromedia⁶, pero actualmente propiedad de Adobe Systems⁷ y SQL Server que son propietarios y disponibles solo para el sistema operativo *Windows*. Estas características hacen que se descarten completamente al tener como requisitos indispensable el uso de herramientas libres y que sea capaz de correr en el sistema operativo Linux.

⁶ era una empresa de software de gráficos y desarrollo web formada en 1992. El 18 de abril de 2005, *Adobe Systems*.

⁷ es una empresa de software fundada en Diciembre de 1982 por John Warnock y Charles Geschke. Destaca en el mundo del software por sus programas de edición de páginas web, video e imagen digital.

Conclusiones.

En este capítulo se abordaron los elementos teóricos que sustentan el problema científico y los objetivos del trabajo. Se realizó un estudio crítico y valorativo, sentando las bases teóricas que apoyan la solución propuesta. Con la descripción de los procesos de transmisión en canales de Televisión y los teletextos se logra dar una referencia de conceptos básicos asociados a la solución. Se analizaron otras soluciones existentes, concluyendo que no resuelven la problemática planteada siendo inconciliables con la misma.

Al plasmar en este capítulo el objetivo de estudio se logra tener un mejor entendimiento de la situación que llevó a la realización del presente trabajo, siendo muy útil para la comprensión de los capítulos siguientes.

CAPÍTULO 2

Tendencias y tecnologías actuales a desarrollar

Introducción.

En este capítulo se realiza un análisis detallado de los principales conceptos y tecnologías que se adecuan a la solución propuesta. Se abordan temas relacionados con el uso de tecnologías sobre aplicaciones multimedia, *JavaScript* asíncrono + XML (AJAX, siglas en inglés) y *Streaming*. Se hace un estudio de los lenguajes de programación más apropiados para el desarrollo del subsistema, el *framework* y Sistema de Gestión de Bases de Datos (SGBD) que se utilizarán. Se referencia también las herramientas necesarias para la obtención del producto final deseado. Además se considera la metodología a utilizar para el análisis y diseño del sistema teniendo en cuenta las facilidades que puede aportar al trabajo.

2.1 Descripción de las tendencias y tecnologías actuales sobre las que se apoya la propuesta.

Con la utilización de más de un medio de comunicación para presentar la información, las aplicaciones multimedia encuentra su utilidad en varias áreas, entre las que se destacan el arte, la educación y el entretenimiento. Como aspecto revolucionario en el mundo de presentar los contenidos multimedia en la Web, así como la utilización de los videos, han surgido dos nuevas alternativas, AJAX y Streaming, las cuales hacen de la Web un campo sin fronteras en el mundo de la multimedia.

Estos tres conceptos, aplicaciones multimedia, AJAX y *Streaming*; están estrechamente ligados a la solución que se propone. A continuación se realiza una caracterización del por qué el uso de las tecnologías que se describen, donde se pueden observar las ventajas que ofrecen para el desarrollo del sistema.

2.1.1 Aplicaciones Multimedia.

Antes de definir el concepto de aplicación multimedia se hace necesario conocer el de multimedia. Una de las definiciones que se tiene de aplicación multimedia es: "Cualquier combinación de texto, arte gráfico, sonido, animación y video que llega a usted por cualquier computadora u otros medios". (21)

Desde otro punto de vista, Monet [1995], después de un estudio panorámico sobre la multimedia, ofrece las siguientes definiciones:

- Noción simplista: alianza de las capacidades de comunicación de la televisión y, por extensión, del vídeo, con la potencia y la interactividad de las computadoras.
- Noción tecnológica: "media o de comunicación integrada a interactiva". Fusión de al menos dos de los soportes de la comunicación (texto, voz, sonido, imagen, fotografía, animación gráfica, video) en el seno de programas profesionales, de servicios o de obras electrónicas, lúdicas o pedagógicas. La información ofrecida, algunas veces a distancia, puede ser visualizada y organizada inmediatamente por el usuario mediante un material y un programa que permiten actuar sobre la presentación desplegada. (14)

Una vez conocido a lo que se refiere el término multimedia, se puede definir a las aplicaciones multimedia como: un sistema informático donde se relacionan los distintos soportes de comunicación (texto, sonido, imagen y videos) con un determinado propósito.

Apoyándose en los fundamentos de los conceptos anteriormente expuestos, se puede afirmar que una aplicación multimedia le permite al usuario obtener una gran cantidad de información, como la que se mostrará en el canal interno del MENPET con la publicación de las noticias de distintos formatos, sin que se tome excesivamente monótono. Además es necesario desarrollar posibilidades técnicas, estéticas y de comunicación completamente novedosas. Por estas razones se decide implementar la visualización automática de las noticias mediante una aplicación multimedia.

2.1.2 AJAX.

Se acuñó por primera vez el termino *AJAX* en el artículo "*AJAX: A New Approach to Web Applications*" publicado por Jesse James Garrett el 18 de Febrero de 2005. Hasta ese momento, no existía un término normalizado que hiciera referencia a un nuevo tipo de aplicación Web que estaba apareciendo.

Como una primera definición del término *AJAX*, se tiene la expuesta por James en su artículo: "*AJAX* no es una tecnología en sí mismo. En realidad, se trata de la unión de varias tecnologías que se desarrollan de forma autónoma y que se unen de formas nuevas y sorprendentes" (8).

Las tecnologías que forman *AJAX* son:

- XHTML⁸ y CSS¹¹, para crear una presentación basada en estándares.
- DOM¹¹, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT⁹ y JSON¹⁰, para el intercambio y la manipulación de información.
- XMLHttpRequest¹¹, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

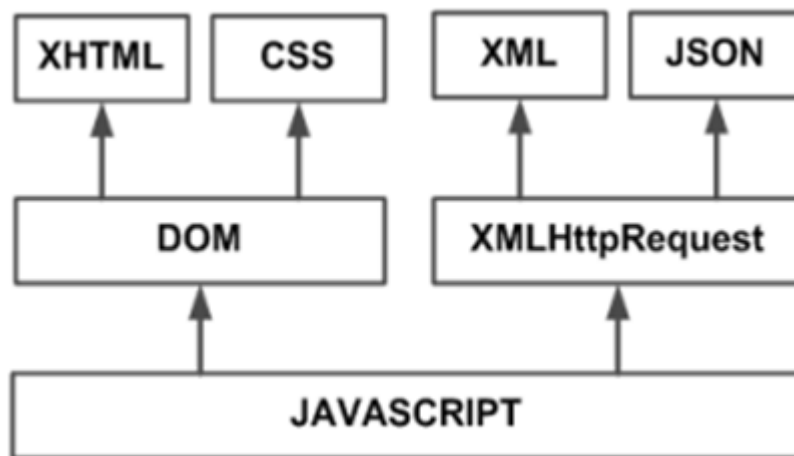


Figura 2. Tecnologías agrupadas bajo el concepto de AJAX.

En las aplicaciones Web tradicionales, las acciones sobre la página, desencadenan llamadas al servidor. Una vez procesada la petición, el servidor devuelve una nueva página al navegador. Esta técnica tradicional para crear aplicaciones Web funciona correctamente, pero no crea una buena sensación al usuario. Al realizar peticiones continuas al servidor, el usuario debe esperar a que se recargue la página con los cambios solicitados. Si la aplicación debe realizar peticiones continuas, la aplicación Web se convierte en algo más molesto que útil.

Las aplicaciones construidas con *AJAX* eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de *AJAX* mejora la

⁸ Ver glosario de términos.

⁹ lenguaje de hojas extensibles de transformación (XSLT, siglas en inglés), se utiliza para tratar estructuras de datos en formato XML.

¹⁰ notación de objetos JavaScript (JSON, siglas en inglés), es un formato ligero para el intercambio de datos.

¹¹ es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores WEB.

respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

Las peticiones *HTTP*¹² al servidor se transforman en peticiones *JavaScript* que se realizan al elemento encargado de *AJAX*. Las peticiones más simples no requieren intervención del servidor, por lo que la respuesta es inmediata. Si la interacción del servidor requiere la respuesta del servidor, la petición se realiza de forma asíncrona mediante *AJAX*. En este caso, la interacción del usuario tampoco se ve interrumpida por recargas de página o largas esperas por la respuesta del servidor.

Desde su primera definición, se han creado cientos de aplicaciones basadas en *AJAX* que en la mayoría de los casos pueden sustituir completamente a otras técnicas como *Flash*¹³ y en el caso de las aplicaciones más avanzadas, pueden sustituir a complejas aplicaciones de escritorio.

En una investigación realizada en busca de la manera de desarrollar la aplicación multimedia que diera solución a la problemática planteada, se analizaron algunas propuestas de sistemas de autor libres y propietarios disponibles. Como resultado obtenido de este análisis se llegó a la conclusión que no existe aún un sistema de autor libre para el desarrollo de la aplicación. Se refiere específicamente a los sistemas de autor libre porque se tiene como exigencia del cliente que el producto sea completamente desarrollado sobre *software* libre.

Teniendo por descartada la posibilidad de utilizar un sistema de autor los cuales son diseñados para un propósito específico, donde muchas de las necesidades más habituales para la creación de aplicaciones multimedia han sido previstas de antemano y son fáciles de implantar, se analizó la otra vertiente para la construcción de aplicaciones multimedia, la cual constituía en utilizar un lenguaje de programación de propósito general.

Los lenguajes de programación de propósito general, existentes en variedad en todas las distribuciones *GNU/Linux*, permiten el control del *hardware* de la máquina a un nivel más bajo logrando aplicaciones más eficientes y robustas; pero en cambio requieren de un esfuerzo extra en la manipulación de los recursos audiovisuales o el desarrollo de herramientas de apoyo a la producción para automatizar el proceso.

¹² es el protocolo usado en cada transacción de la Web.

¹³ es una aplicación en forma de estudio de animación que trabaja sobre fotogramas destinado a la producción y entrega de contenido interactivo.

Al utilizar las características de *AJAX* y la ayuda de un *framework* se torna muy fácil lograr la interacción de los recursos multimedia para obtener los efectos visuales necesarios en la aplicación a construir, simulando lo que se podía obtener con los sistemas de autor. Además en el lado del servidor se utilizaría un lenguaje de programación de propósito general para lograr la comunicación con la base de datos que respaldará al sistema.

Después del análisis realizado se llega a la conclusión de que con *AJAX* se puede lograr la simulación de una aplicación de escritorio que presente la interactividad de distintos elementos multimedia con el uso de efectos visuales, constituyendo esta, una solución para la aplicación multimedia que genera el contenido a transmitir por la señal del canal de televisión que se desea implementar.

2.1.3 Streaming.

Se conoce como *Streaming* a la distribución de contenidos multimedia por medio de un flujo constante a través de la red sin que sea necesario que el usuario descargue el archivo a su computador. (15)

Mucho ha influido el avance tecnológico que se ha experimentado en el procesamiento de datos y redes para que se pueda lograr implementar esta tecnología. Entre los principales tipos de transmisión de datos están:

Unicast: Es la transmisión punto a punto, utiliza la arquitectura cliente/servidor donde se transmite múltiples flujos de datos. Esta arquitectura consume mayor ancho de banda.

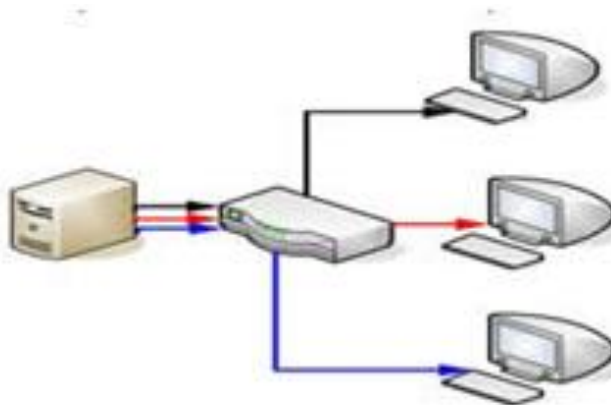


Figura 3. Tipo de transmisión *unicast*.

Multicast: Es la transmisión punto a multipunto, utiliza la arquitectura cliente/servidor para un grupo de usuarios determinados por el servidor, esto se le puede hacer mediante *software* y una lista de acceso por direcciones IP^{14}/MAC^{15} . Solo se transmite un solo flujo de datos multimedia.

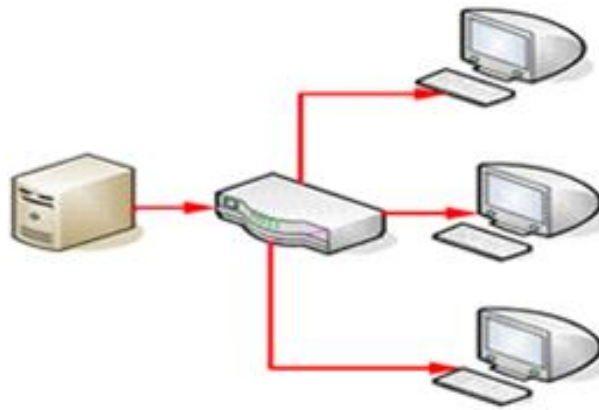


Figura 4. Tipo de transmisión *multicast*.

Para poder transmitir por *Streaming*, se necesitan los siguientes componentes:

- **Servidor de Streaming:** *software* encargado de la gestión de las peticiones de los usuarios para acceder el video. Los servidores pueden proporcionar dos tipos de contenidos:
 - **Bajo demanda** (VOD, siglas en inglés): Son peticiones realizadas por clientes individuales a ficheros almacenados en el servidor, cada cliente que solicita una secuencia suele tener el control total de la misma y puede aplicar las características de avance rápido, rebobinado, pausa y reinicio del contenido. Esto se debe a que se proporcionan una sola ruta de acceso a los datos para cada cliente que solicita el contenido.
 - **Difusión** (*broadcast*): Es la transmisión de video en vivo. Mediante este método se realiza una transmisión simultánea de datos a una audiencia de gran volumen. Los clientes que reciben una difusión no pueden controlar el inicio del contenido, la velocidad de reproducción, el avance rápido ni rebobinar la secuencia. Es el servidor el que tiene el control de la secuencia. Los contenidos pueden ser creados en ese

¹⁴número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo de internet (IP, siglas en inglés).

¹⁵es un número hexadecimal fijo que es asignado a la tarjeta o dispositivo de red por el fabricante.

momento en vivo o almacenados previamente en el servidor. El sistema de difusión tiene analogías con los canales de TV.

- **Reproductor:** *software* encargado de la reproducción de los archivos suministrados por el servidor de *Streaming*. Es necesario tenerlo instalado en el ordenador para recibir los videos del servidor.

Haciendo uso de las posibilidades que brinda esta tecnología se puede dar solución a parte de la problemática planteada en el presente trabajo como es la transmisión de señales de televisión en vivo y la reproducción de materiales audiovisuales.

2.2 Lenguajes de programación.

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente.

Para la realización del subsistema serán utilizados varios de los lenguajes de programación existentes, de acuerdo al fin para el cual fueron construidos. A continuación se brinda una breve descripción de cada uno de ellos.

2.2.1 Lenguajes de programación del lado cliente.

Los lenguajes de programación del lado cliente están insertados en la página Web y son interpretados y ejecutados por el navegador. Para la implementación del subsistema se deben tener en cuenta alguno de ellos, los cuales son referenciados a continuación.

2.2.1.1 HTML.

Marcas de *Hipertexto* (HTML, siglas en inglés) es el lenguaje que todos los programas navegadores usan para presentar información en la *World Wide Web* (WWW). Este es un lenguaje muy sencillo que se basa en el uso de etiquetas, consistentes en un texto *ASCII*¹⁶ encerrado dentro de un par de paréntesis angulares (<...>). El texto incluido dentro de los paréntesis dará una explicación de la utilidad de la etiqueta. Las etiquetas podrán incluir una serie de atributos o parámetros, en su mayoría opcionales, que permitirá definir diferentes posibilidades o características de la misma.

¹⁶ código Estándar para el Intercambio de Información.

Entre otras cosas, el manejo de estas etiquetas permitirá:

- Definir la estructura lógica del documento *HTML*.
- Aplicar distintos estilos al texto.
- La inclusión de imágenes y ficheros multimedia (gráficos, vídeo, audio).

Todas estas características ofrecen importantes ventajas para el desarrollo de la aplicación que soportará el subsistema de transmisión en la visualización de los contenidos, debido a la gran capacidad multimedia que se genera con ellas.

2.2.1.2 JavaScript.

Se conoce a JavaScript como un lenguaje de programación que se utiliza principalmente para crear páginas Web dinámicas. Una página Web dinámica es aquella que incorpora efectos, como textos que aparecen y desaparecen, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario (17).

Las características más importantes del lenguaje JavaScript son:

- Es un lenguaje interpretado, es decir, no requiere compilación. El navegador del usuario se encarga de interpretar las sentencias JavaScript contenidas en una página HTML y ejecutarlas adecuadamente.
- Es un lenguaje orientado a eventos. Cuando un usuario selecciona un enlace o mueve el puntero sobre una imagen se produce un evento. Mediante JavaScript se pueden desarrollar scripts que ejecuten acciones en respuesta a estos eventos.
- Es un lenguaje orientado a objetos. El modelo de objetos de JavaScript está reducido y simplificado, pero incluye los elementos necesarios para que los scripts puedan acceder a la información de una página y puedan actuar sobre la interfaz del navegador.

Al ser un lenguaje de programación que se ejecuta en el lado cliente, posibilita la reducción de carga del servidor, siendo esta una característica muy importante para la aplicación que se quiere desarrollar, al poderse implementar distintas funcionalidades con recursos multimedia los cuales resultan “pesados” a la hora de su uso. Además la vinculación estrecha que existe entre este lenguaje y AJAX hace necesario su utilización.

2.2.2 Lenguajes de Programación del lado servidor.

Los lenguajes del lado servidor son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Entre los más usados actualmente se encuentran Java mediante las Páginas de Servidor Java (JSP, siglas en inglés) y PHP de los que se brinda una breve descripción a continuación.

2.2.2.1 Java

Java es un lenguaje de programación orientado a objetos desarrollado por la empresa SUN Microsystems¹⁷ en 1995. Es un lenguaje de alto nivel y propósito general similar a C++, con marcadas características de seguridad y transportabilidad. Este lenguaje define una máquina virtual independiente de la plataforma donde se ejecuta, que procesa programas llamados *Applets*, descargados desde el servidor Web. Además, debido al modo de ejecución de los *Applets*, este lenguaje es muy seguro frente a la presencia y ataque de virus informáticos (5). En la actualidad existen varias formas de usar Java dentro de un servidor Web, mediante *Servlets* o JSPs.

Los *Servlets* son componentes del servidor. Estos componentes pueden ser ejecutados en cualquier plataforma o en cualquier servidor debido a la tecnología Java que se usa para implementarlos. Los *Servlets* incrementan la funcionalidad de una aplicación Web. Se cargan de forma dinámica por el entorno de ejecución Java del servidor cuando se necesitan. Cuando se recibe una petición del cliente, el contenedor/servidor Web inicia el *Servlet* requerido. El *Servlet* procesa la petición del cliente y envía la respuesta de vuelta al contenedor/servidor, que es enrutada al cliente.

JSP es una tecnología similar a los *Servlets* que ofrece una conveniente forma de agregar contenido dinámico a una página Web por utilizar código escrito en Java dentro de la página utilizando *tags* especiales que son procesados por el servidor Web antes de enviarlos al cliente. La posibilidad de usar APIs¹⁸ de Java se obtiene la ventaja de la programación orientada a objeto, como creación de clases especiales llamadas componentes o Java *Beans*, independencia de la plataforma propia de la programación en Java, etc.

La diferencia entre *Servlets* y JSP es que los *Servlets* son clases que deben implementar la clase abstracta *HttpServlet*, en especial el método *doGet()* o *doPost()* y deben ser previamente compilados,

¹⁷ es una empresa informática de Silicon Valley, fabricante de semiconductores y software.

¹⁸ interfaz de programación de aplicaciones (API, siglas en inglés): una serie de funciones que están disponibles para realizar programas para un cierto entorno.

mientras que los archivos JSP contienen código Java entre código HTML utilizando los símbolos `<%` y `%>`.

2.2.2.2 PHP.

PHP es un lenguaje interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas, similar al ASP de *Microsoft* o el JSP de *SUN*, embebido en páginas HTML y ejecutado en el servidor. La mayor parte de su sintaxis ha sido tomada de *C*, *Java* y *Perl* con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas.

Este lenguaje dispone de gran cantidad de características que lo convierten en el lenguaje de programación del lado del servidor ideal para usar.

- **Multiplataforma:** el único requisito para poder ejecutar PHP es contar con un servidor Web. En entornos Unix o Linux, a través de *Apache*; y en entornos *Windows* a través de *IIS*¹⁹. Con esto el código generado en una plataforma es perfectamente válido para otra. Las mejores prestaciones se consiguen a través de *Apache* como servidor Web, ya que el interprete de PHP se compila como un módulo de *Apache*, mejorando la velocidad de ejecución con un consumo de recursos bajo.
- **Licencia de *software* libre:** PHP está basado en *software* libre, lo que permite distribuirlo libremente y agregar nuevas funcionalidades si se necesita. Además no hay que pagar licencias por su uso.
- **Sintaxis cómoda:** La sintaxis de PHP es similar a *C*, *C++* o *Perl*. En PHP 5, trabajando con programación orientada a objetos se encuentra una sintaxis parecida a la de *Java*. Cabe señalar la forma de crear variables, en PHP no es necesario definir el tipo de éstas, lo cual permite guardar cualquier valor dentro de ellas.
- **Soporta objetos y herencia:** PHP soporta programación orientada a objetos, sobre todo con las mejoras introducidas en PHP 5, y también la herencia, aunque no múltiple.

¹⁹ es una serie de servicios para los ordenadores que funcionan con *Windows*.

- Extensa librería de funciones: Las librerías de funciones de PHP facilitan el trabajo de desarrollo, y casi siempre es posible encontrar una función en sus librerías que se adapte a lo que se necesita, reduciendo así los tiempos de desarrollo.
- Compatibilidad con bases de datos: PHP ofrece soporte para un amplio abanico de bases de datos: dBase, DB2, mSQL, Oracle, PostgreSQL, MySQL, etc.
- Soportado por una gran comunidad de desarrolladores: como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.

2.2.2.3 Fundamentación del lenguaje de programación de lado del servidor escogido.

Como lenguaje de programación para el lado del servidor se escogió PHP, al garantizar con sus características, obtener un producto que se pueda montar en un entorno *Unix* o *Linux* con *software* totalmente libres sin la necesidad de pagar licencias por el uso del lenguaje de programación, facilidad para el grupo de desarrolladores en el aprendizaje de la sintaxis de *PHP* al estar muy relacionada con la de *C++*, agilidad en la elaboración del producto al contar con librerías donde están implementadas las funciones necesarias para dar solución a gran cantidad de funcionalidades del sistema, compatibilidad con *PosgreSQL* sistema de gestión de bases de datos que se utilizará, los cuales constituyen requisitos imprescindibles para obtener una aplicación que satisfaga las necesidades del cliente.

2.3 Framework

Un *framework* simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un *framework* proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un *framework* facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

Entre los *framework* que más se usan para las aplicaciones Web donde se utilice el lenguaje de programación PHP se encuentran CakePHP, Zend Framework y Symfony de los cuales se brinda a continuación una pequeña descripción, exponiendo además, cual fue elegido para realizar el subsistema.

2.3.1 CakePHP.

CakePHP es un *framework* de desarrollo rápido de aplicaciones de código abierto en PHP, inspirado en Ruby on Rails. Es un *framework* para la construcción de sitios Web que utilizan una base de datos como fuente de recursos, posee una infraestructura que tiene como finalidad permitir el desarrollo de manera ágil y estructurada, sin perder flexibilidad.

Entre las características más destacables de CakePHP se incluyen:

- Arquitectura basada en el patrón Modelo Vista Controlador (MVC) y orientada a objetos: define clases modelo, vista y controlador con funcionalidades básicas y de las cuales heredan todas las clases que se ajustan a este patrón y que son usadas en la aplicación construida con el *framework*.
- Una comunidad activa de usuarios: creada tras la publicación del *framework* en 2005 y que ha contribuido a mejorar el *framework* y difundir su uso.
- Licencia flexible: es distribuido bajo la licencia X11, más conocida entre los desarrolladores de software como MIT *License*.
- Compatible con PHP4 y PHP5: aunque en PHP4 se requiere especificar algunos parámetros de configuración adicionales en las clases a implementar.
- Operaciones básicas en base de datos (creación, obtención, actualización y borrado): estas operaciones están integradas para la interacción con la base de datos y la simplificación de consultas.
- Estructura de aplicaciones (*Application Scaffolding*): permite al programador hacer uso de un conjunto de convenciones aplicables a la estructura de la base de datos de la aplicación y el *framework* se encarga de generar el código para la interacción a lo largo de todas las capas de la aplicación.
- Despachador de peticiones: permite acceder a la aplicación a través de URLs²⁰ amigables y configurables.

²⁰localizador uniforme de Recursos (URL, siglas en inglés), es la cadena de caracteres con la cual se asigna dirección única a cada uno de los recursos de información disponibles en Internet.

- Incorporación de validaciones a lo largo del *framework*.
- Generación de plantillas de manera rápida y flexible: usando la sintaxis de PHP y con asistentes o *helpers*.
- Incorporación de asistentes de construcción de vistas: para la automatización de la generación de código en AJAX, JavaScript, formularios HTML, entre otros.
- Componentes de seguridad, manejo de sesiones y de peticiones: que reúnen las mejoras prácticas estandarizadas por la industria del software.
- Listas de control de acceso flexibles: para gestionar el ingreso de usuarios a la aplicación construida con el *framework*.
- Verificación de ingreso de datos permitidos: facilita determinar qué datos pueden ser ingresados y darle el formato adecuado a aquellos que no cumplen las reglas de validación.
- Almacenamiento en caché de las vistas: para acelerar la descarga de las páginas Web.
- Trabaja en cualquier subdirectorio de un servidor Web: requiere poca o nula configuración del servidor Apache donde se instalará.

2.3.2 Zend Framework

Zend Framework, es un intento dirigido por la compañía responsable del desarrollo del lenguaje PHP, Zend Technologies Ltd, y mantenido por una comunidad de voluntarios. Le destaca el hecho de que no sólo busca facilitar la programación a través del patrón MVC, sino también automatizar tareas más específicas, como el acceso a base de datos, el filtrado de datos ingresados a la aplicación o la búsqueda en un sitio Web ordenando resultados por relevancia. Entre sus metas se encuentran:

- Proveer un repositorio de componentes de alta calidad y que cuenten con soporte activo.
- Proveer un sistema completo para el desarrollo de aplicaciones Web elaboradas en PHP5.
- Facilitar el aprendizaje en el uso del *framework* sin tener que aprender un nuevo lenguaje de programación.
- Organizar la colaboración de la comunidad para una programación avanzada en PHP5.

Los componentes con los que cuenta este *framework* pueden ser agrupados en las siguientes categorías:

- Infraestructura del núcleo del *framework*: componentes requeridos por otros bloques del *framework*, como memoria caché, configuración del ambiente de trabajo, operación por línea de comandos, registro de actividades y gestión de memoria, entre otros.
- Autenticación y autorización de acceso: responsables de la configuración de listas de control de acceso, autenticación de usuarios y manejo de sesiones.
- Base de datos: clases de acceso, modificación de tablas, obtención de datos mediante consultas SQL²¹ y lectura en formato XML.
- Internacionalización y Localización: para configurar la fecha y hora, la ubicación geográfica que tomará como base la aplicación Web, las unidades de medida a usar y la posibilidad de traducir la información a otros idiomas.
- Emails, formatos y búsquedas: generación de archivos PDF, mensajes de correo electrónico en formato de texto y extensiones de correo internet multipropósito (MIME, siglas en inglés).
- Modelo-Vista-Controlador: centra su atención en el desarrollo de controladores genéricos y específicos: de acciones, de atención de peticiones, de generación de URLs; además de una clase para las vistas.
- Web y Servicios Web: permite que la aplicación Web pueda hacer uso de servicios Web ofrecidos por aplicaciones externas y exponga servicios Web propios.
- Documentación: intenta facilitar la lectura del código fuente, brindar ejemplos de su uso, promover equipos de traducción de la documentación y elaborar un tutorial del desarrollo de una aplicación Web.

A pesar de que este *framework* esta dirigido por la compañía responsable del desarrollo de PHP, aún no consigue niveles de eficacia y adopción similares a los *frameworks* en PHP.

2.3.3 Symfony.

Symfony es un completo *framework* diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones Web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación Web. Proporciona varias herramientas y clases encaminadas a reducir

²¹lenguaje de consulta estructurado (SQL, siglas en inglés), es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.

el tiempo de desarrollo de una aplicación Web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación Web (7).

Su creador, Fabien Potencier, toma prestadas las mejores ideas de cualquier *framework* (da igual si está escrito en *Perl*, *Python* o *Ruby*) y las adapta para *Symfony*, además de añadir las suyas propias.



Figura 5. Características de otros *framework* presentes en *Symfony*

La lista de ideas que incorpora *Symfony* y que tienen su origen en otros *frameworks* y aplicaciones es la siguiente:

- *Ruby on Rails*: sistema de enrutamiento, *helpers*, *scaffolding*²², archivos de configuración en formato *YAML*²³ y la herramienta *Rake* (que en *Symfony* se llama *Pake*²⁴).
- *Django*: el nuevo mecanismo de formularios y *widgets* que incorpora la versión 1.1 de *Symfony*.
- *Propel* y *Doctrine*: son los dos *ORM*²⁵ principales de *Symfony*, sobre todo *Propel*, que se encuentra completamente integrado.

²² es una estructura básica de acciones y plantillas para poder realizar las operaciones CRUD en una tabla de la base de datos

²³ es un formato de serialización de datos legible, inspirado en lenguajes como XML, C, Python, Perl.

²⁴ es una utilidad para la línea de comandos.

²⁵ mapeo objeto-relacional (ORM, siglas en inglés,) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos.

- Prado: todo lo relacionado con la internacionalización (i18n) y la localización (l10n).
- Cocoa: el nuevo sistema de eventos de Symfony 1.1 se basa completamente en este *framework* de Apple²⁶.
- *Test::More*: la herramienta Lime, para crear pruebas unitarias, está basada en el *framework* *Test::More* de Perl.
- *Prototype*, *script.aculo.us*, *TinyMCE*: utilidades relacionadas con AJAX y *JavaScript*.
- Al centralizar estas características en uno solo, hacen que el uso de este *framework* posibilite la implementación de una mayor cantidad de funcionalidades de forma sencilla, en las aplicaciones que se desarrollen.

Entre las características generales del *framework* se pueden citar:

- Fácil de instalar y configurar: ha sido probado con éxito en plataformas Windows y derivadas de Unix.
- Independiente del manejador de base de datos: utiliza Propel, una capa de abstracción que le permite interactuar con varias bases de datos.
- Simple de usar: y al mismo tiempo lo suficientemente flexible para adaptarse a escenarios complejos.
- Basado en la premisa de "convención sobre configuración": el desarrollador sólo necesita configurar aquellos aspectos sobre los cuales no hay una tendencia definida.
- Cumple con la mayoría de las mejores prácticas en diseño Web y patrones de diseño.
- Utilizable en entornos empresariales: puede adaptarse a políticas y arquitecturas ya existentes en tecnologías de información, y es lo suficientemente estable para proyectos de largo plazo.
- Código legible: con comentarios en phpDocumentor²⁷ para su fácil mantenimiento.
- Fácil de extender: permitiendo la integración con otras librerías.

²⁶ es una empresa estadounidense de tecnología informática. Fundada en 1976, tuvo un rol prominente en la revolución informática de los años 70 con su computadora Apple II.

²⁷ es un sistema para crear documentación de aplicaciones creadas con PHP.

- Incorpora herramientas que facilitan la prueba y depuración de aplicaciones: como unidades de generación de código, pruebas del funcionamiento del *framework*, panel de depuración, interfaz por línea de comandos y configuración en tiempo real.
- Incorpora una capa de internacionalización: que posibilita la traducción de datos e interfaces, así como la localización de contenido en función de la ubicación geográfica del usuario.
- Uso de plantillas: las cuales pueden ser elaboradas por diseñadores de páginas Web que desconocen el resto de detalles técnicos del *framework*.
- Asistentes de vistas: que reducen la cantidad de código de presentación al escribir grandes bloques de código con simples llamadas a funciones.
- Validación y regeneración automática de formularios: lo que asegura una buena calidad de los datos en la base de datos y una mejor experiencia de usuario.
- Verificación de la salida enviada por la aplicación: que ofrece una protección frente a ataques por datos corruptos.
- Manejo de memoria caché: lo cual reduce el uso del ancho de banda y la carga en servidor.
- URLs inteligentes: que permiten que las direcciones de las páginas Web sean parte de la interfaz y resulten amigables a los motores de búsqueda.

Las interacciones usando AJAX son fáciles de implementar gracias a los asistentes de una sola línea de código que encapsulan efectos en JavaScript compatibles con varios navegadores.

2.3.4 Fundamentación del *framework* escogido.

Los *frameworks* anteriormente mencionados poseen de forma general grandes mejoras para poder optimizar el trabajo de los desarrolladores en el lenguaje PHP. A la hora de escoger uno de ellos es necesario ver cual tiene mayor cantidad de características que se ajusten específicamente, a la aplicación que se quiere obtener. Teniendo en cuenta lo anteriormente planteado se decide utilizar Symfony, al constituir un *framework* muy maduro, que permitiría lograr migrar de *SGBD* sin hacer cambios en el código fuente de la aplicación, obtener un código comentado, logrando que cada uno de los desarrolladores tenga una guía cuando se realice cualquier cambio en la aplicación. Otra de las características que lo distinguen es el uso de los *helpers*, los cuales permiten un manejo de los elementos multimedia sin tener que hacer grandes usos de *JavaScript*, lo cual resultaría más complicado, además de brindar la facilidad de no notar la recarga de las páginas cada vez que se

realice un cambio en ellas. Por último y no menos importante es la maravillosa documentación (sobre todo en español) que posee, la cual incluye un libro que explica detalladamente todos sus conceptos, la documentación de la API y la Wiki, los cuales son muy útiles en la obtención de ayuda cuando no se tenga conocimiento de lo que se quiere realizar, teniendo una gran fuente donde buscar. Para tener una mejor visión de la comparación entre estos *framework*, ver [Anexo III](#).

2.4 Sistema de Gestión de Bases de Datos.

Para los sistemas que necesitan procesar gran cantidad de información resulta imprescindible usar una herramienta que les permita almacenar, acceder y modificar estos datos. Para el propósito anteriormente planteado surgen los Sistemas de Gestión de Bases de Datos (SGBD).

Stanley Su en su libro *“Database Computers: Principles, Architectures and Techniques”* define un Sistema de Gestión de Bases de Datos como un paquete de software que proporciona todas las facilidades para la creación, recuperación, manipulación y mantenimiento de bases de datos, asegurando su integridad, confidencialidad y seguridad. (19)

Entre los SGBD más populares para los sistemas operativos GNU/Linux se encuentran MySQL y PostgreSQL, de los cuales se brinda una pequeña descripción a continuación.

2.4.1 MySQL.

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU²⁸. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, actualmente fue comprado por SUN Microsystems.

Aunque MySQL es software libre, posee una versión comercial, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Las principales características de este gestor de bases de datos son las siguientes:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de APIs en gran cantidad de lenguajes (C, C++, Java, PHP, etc).

²⁸ proyecto iniciado por Richard Stallman con el objetivo de crear un sistema operativo completamente libre.

- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.

2.4.2 PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley. PostgreSQL es una derivación libre (código abierto) de este proyecto, y utiliza el lenguaje SQL92/SQL99, así como otras características.

Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales. PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.

A continuación se enumeran las principales características de este gestor de bases de datos:

- Implementación del estándar SQL92/SQL99.
- Por su arquitectura de diseño, escala muy bien al aumentar el número de CPUs²⁹ y la cantidad de RAM³⁰.
- Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial).
- Tiene mejor soporte para *triggers*³¹ y procedimientos en el servidor.
- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.

²⁹unidad central de procesamiento (CPU, siglas en inglés), es el componente en una computadora digital que interpreta las instrucciones y procesa los datos contenidos en los programas de computadora.

³⁰memoria de acceso aleatorio, o memoria de acceso directo (RAM, siglas en inglés), se compone de uno o más chips y se utiliza como memoria de trabajo para programas y datos.

³¹ver glosario de términos.

- Implementa el uso de *rollback's*³¹, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz.
- Numerosos tipos de datos, posibilidades de definir nuevos tipos
- Soporta el almacenamiento de objetos binarios grandes (gráficos, videos, sonido, ...)
- Replicación asíncrona.
- Copias de seguridad en caliente.
- APIs para programar en C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, PHP y muchos otros lenguajes.
- Completa documentación

Algunos de los límites físicos de PostgreSQL son: (10)

Limite	Valor
Máximo tamaño base de dato	Ilimitado (Depende del sistema de almacenamiento)
Máximo tamaño de tabla	32 TB
Máximo tamaño de fila	1.6 TB
Máximo tamaño de campo	1 GB
Máximo número de filas por tabla	Ilimitado
Máximo número de columnas por tabla	250 - 1600 (dependiendo del tipo)
Máximo número de índices por tabla	Ilimitado

Tabla 1. Límites físicos de PostgreSQL.

2.4.3 Fundamentación del SGBD escogido.

Teniendo en cuenta que MySQL puede tener un futuro incierto con su valor de código abierto, al ser comprado por SUN Microsystems, la cual ya ha enseñado su visión futurista con el producto, al plantear que algunas partes de MySQL sólo serían ofrecidas a clientes corporativos que paguen una suscripción por el sistema de base de datos, y que éstos componentes sólo estarían disponibles como código abierto cerrado (20), se decide utilizar PostgreSQL en la aplicación que se quiere obtener, al ser uno de los SGBD más potente del momento y que en sus últimas versiones empieza a no tener que envidiarle nada a otras bases de datos comerciales, además de ser ideal para sistemas serios en las que la consistencia de la base de datos sea fundamental.

2.5 Metodologías de desarrollo de software.

Las metodologías de desarrollo de *software* abarcan todo el ciclo de vida del *software*, y se definen como un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo *software* (4), adoptando la misma se obtiene un producto *software* más predecible y permite ciertas características deseables como:

- Existencias de reglas bien definidas.
- Verificaciones intermedias.
- Planificación y control.
- Comunicación efectiva.
- Utilización sobre un abanico amplio de proyectos.
- Fácil formación.
- Herramientas CASE³².
- Actividades que mejoren el proceso de desarrollo.
- Soporte al mantenimiento.
- Soporte de la reutilización de software.

En los últimos años se han desarrollado dos corrientes en lo referente a los procesos de desarrollo, los llamados métodos ligeros o ágiles y los pesados. A continuación se brinda una breve descripción de un método ligero conocido como Programación Extrema (XP, siglas en inglés) y un método pesado llamado Proceso Unificado de Desarrollo (RUP, siglas en inglés)

2.5.1 Programación Extrema

XP nace “oficialmente” en un proyecto desarrollado por Kent Beck en *DaimlerChrysler*, después de haber trabajado varios años con *Ward Cunningham* en busca de una nueva aproximación al problema del desarrollo de software que hiciera las cosas más simples de lo acostumbrado por las metodologías existentes.

Como toda metodología ágil, trata de reducir la complejidad del software, por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción. Minimiza el riesgo de fallo del proceso por medio de la disposición permanente de un representante

³² ingeniería de Software Asistida por Ordenador (CASE, siglas en inglés).

competente del cliente a merced del equipo de desarrollo. Este representante debería estar en condiciones de contestar rápida y correctamente a cualquier pregunta del equipo de desarrollo de forma que no se retrase la toma de decisiones, de ahí lo de competente.

El ciclo de desarrollo consiste, a grandes rasgos, en los siguientes pasos (2):

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

2.5.2 Proceso Unificado de Desarrollo.

RUP como metodología es un proceso de desarrollo de software que define *quién* esta haciendo *qué*, *cuándo* y *cómo* alcanzar un determinado objetivo, además es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto (11).

Para la realización del producto *software* la metodología RUP divide el trabajo en 9 flujos de trabajo y 4 fases y tiene como características fundamentales ser iterativo e incremental, dirigido por casos de usos y centrado en la arquitectura.

El ser iterativo e incremental le permite al equipo de desarrollo en un tiempo estimado obtener una pequeña parte del producto de acuerdo a una fase que pase por todas las disciplinas, esto se conoce por iteración, permite además según cada iteración un crecimiento del producto, conocido como incremento.

Sobre esto recae gran importancia pues se puede ir observando poco a poco el adelanto que va teniendo el producto que se está desarrollando y que se puedan corregir a tiempo las deficiencias.

La característica de ser dirigido por casos de uso ayuda también al equipo de trabajo a desarrollar el sistema según los requisitos solicitados por el cliente pues a través de los mismos estos se modelan.

Por último, no por ello menos importante, posee la característica de estar centrado en la arquitectura, esta indica como puede construirse el sistema y en que orden. Todas estas características fueron destacadas por los autores del proceso unificado desde sus inicios.

El Proceso Unificado de Desarrollo además contiene las 6 mejores prácticas de la ingeniería de *software* que están siendo utilizadas actualmente para el desarrollo exitoso de un proceso de desarrollo de *software*, desplegando las mismas ofrece al equipo de desarrollo un número significativo de ventajas pues lo provee de guías, plantillas y herramientas para una mejor forma de trabajo, estas mejores prácticas son (12):

- Desarrollo de software iterativo.
- Administrar requerimientos.
- Usar arquitectura de componentes.
- Modelar el software visualmente.
- Verificar continuamente la calidad del software.
- Controlar cambios del software.

2.5.3 Fundamentación de la metodología de desarrollo de software escogido.

Analizando las características de las dos metodologías anteriormente expuestas se llegó a la conclusión de que la más aceptada para la aplicación que se desea obtener es RUP, al tener XP algunas limitantes como estar dirigida a equipos pequeños o medianos (Beck sugiere que el tamaño de los equipos se limite de 3 a 20 como máximo, otros dicen no más de 10 participantes), necesidad de tener un representante del cliente a total disposición del equipo de desarrollo, genera poca documentación del proyecto así como el modelado y se pueden introducir muchos cambios en las funcionalidades los cuales se acuerda directamente con el representante del cliente. Con RUP se logra tener una planificación del proyecto basada en el plan de desarrollo, plan de iteración, plan de calidad entre otros con los que se controla el desarrollo del software. A través de un esquema de escalabilidad y gestión de riesgos predefinidos, se pueden reconocer previamente problemas y fallos de forma temprana y prevenirlos o corregirlos, no es necesario tener un representante del cliente a total

disposición del equipo de desarrollo, esta pensado para proyectos y equipos grandes, en cuanto a tamaño y duración, define en cada momento del ciclo de vida del proyecto, que artefactos, con que nivel de detalle, y por qué rol, se deben crear. Además con RUP se definirán que artefactos son necesarios para poder realizar una actividad y que artefactos se deberán crear durante dicha actividad. En general RUP garantiza obtener una mejor calidad del producto, así como el control del proceso de desarrollo.

2.6 Herramientas CASE para la modelación UML.

Las herramientas CASE tienen como objetivo automatizar los aspectos claves del proceso de desarrollo de sistemas desde su inicio hasta el final. Las consecuencias de esta automatización se observan en el aumento de la productividad por parte de los desarrolladores de sistema, en la mejora de la comunicación entre usuarios y especialistas, el sostenimiento de una estandarización en las actividades de un proyecto y en las facilidades que brindan para el mantenimiento del sistema.

Dentro de las Herramientas CASE multiplataforma utilizadas hoy en día para la modelación de los diagramas UML se encuentran ArgoUML y Visual Paradigm. De estas dos herramientas se brinda una pequeña descripción a continuación.

2.6.1 ArgoUML

ArgoUML es una herramienta para el modelado de proyectos por medio de UML creado por Jason Robbins y su equipo de colaboradores de la Universidad de California. El primer lanzamiento de esta herramienta se realizó en 1998 y más de 100 000 descargas hasta mediados de 2001 mostraron el impacto de este proyecto.

ArgoUML está codificado en Java, por lo que basta tener instalado el JRE para ejecutarlo. Esto lo hace independiente de la plataforma. Es una herramienta de código abierto muy fácil de usar, la licencia que presenta es BSD³³.

Soporta diversos tipos de diagramas de UML:

- Diagrama de Casos de Uso.
- Diagrama de Clases.
- Diagrama de Secuencia.

³³ distribución de software Berkeley (BSD, siglas en inglés), licencia de software libre.

- Diagrama de Colaboración.
- Diagrama de Estados.
- Diagrama de Actividades.
- Diagrama de Despliegue.

2.6.2 VisualParadigm.

Esta herramienta acelera el desarrollo de aplicaciones, ya que sirve de puente visual entre arquitectos, analistas y diseñadores de sistema, haciendo el trabajo más fácil y dinámico. La herramienta también automatiza tareas tediosas que pueden distraer a diseñadores del desarrollo. Ofrece los siguientes beneficios:

- Varios idiomas.
- Fácil de instalar y actualizar.
- Compatibilidad entre ediciones.
- La navegación intuitiva entre el código y el modelo visual.
- Poderoso generador de informes PDF/HTML.
- Tiempo real en la demanda.
- Un ambiente modelador visual superior.
- Sofisticado diseño de diagramas.

2.6.3 Fundamentación de la herramienta CASE para la modelación UML escogido.

Entre las herramientas anteriormente expuestas fue escogido para utilizar Visual Paradigm al contar con una mayor aceptación entre los desarrolladores de aplicaciones informáticas del mundo. Brinda gran cantidad de funcionalidades en la construcción de los Diagramas, así como tener un desarrollo más completo.

2.7 Sistemas de Streaming.

En la actualidad existen múltiples herramientas de código abierto que permiten la distribución de contenidos multimedia por medio de un flujo constante a través de la red. Algunas de ellas son multiplataforma, lo que permite tener un muy buen nivel de compatibilidad entre el extremo transmisor y receptor cuando se trabaja con distintos sistemas operativos. Otras basan su fortaleza en la adopción

de estándares de compresión y en normas internacionales. A continuación se describen tres de estas alternativas las cuales son de las más reconocidas.

2.7.1 MPEG4IP

El proyecto MPEG4IP provee un conjunto de herramientas de código abierto que permiten implementar servidores y clientes de Streaming basados en estándares exentos de protocolos y extensiones propietarios. Su desarrollo está focalizado a GNU/Linux, pero también ha sido portado a otras plataformas. Dentro de los muchos programas que se incluyen en el proyecto se encuentran el reproductor *gmp4player* y el servidor *mp4live*. Éste permite codificar y transmitir flujos de video y audio en tiempo real, obtenidos a través de la captura desde dispositivos compatibles con V4L³⁴. El servidor *mp4live* dispone de una Interfaz Gráfica de Usuario (GUI, siglas en inglés) que permite configurar los parámetros necesarios para generar el Streaming. La configuración puede ser también leída desde un archivo llamado *.mp4live_rc*. Para aumentar la cantidad de *codecs* soportados por esta herramienta es posible aplicarle parches al código fuente. De todas maneras existe soporte nativo para el formato de video MPEG-4 y para los formatos de audio MP3 y AAC. El resultado de la captura puede ser transmitido por la red vía *unicast* o *multicast*, escrito localmente a un archivo mp4 o ambos simultáneamente. En el caso de ser transmitido por la red, se utiliza el protocolo RTP sobre datagramas UDP.

2.7.2 FFMPEG

FFMPEG es una solución completa para grabar, convertir y generar flujos de audio y video. Con este sistema se puede emitir contenidos en vivo, pregrabados o conectar una videocámara en un sistema de monitorización de vídeo. El proyecto está compuesto de diversos componentes: la herramienta para capturar y/o codificar videos en tiempo real se llama *ffmpeg*, el servidor de Streaming se denomina *ffserver* y además se cuenta con *ffplay*, un reproductor basado en la biblioteca gráfica SDL. Esta aplicación también incluye la poderosa biblioteca multimedia *libavcodec*, la cual en conjunto con *libavformat* son el núcleo de este sistema y han sido desarrolladas desde cero para permitir un mejor desempeño y un alto grado de reutilización.

³⁴ es una API de captura de video para Linux.

El servidor *ffserver* se configura a través de un archivo, el cual debe ser invocado como argumento durante su inicialización. Cada secuencia de video y audio es alimentada a través del programa *ffmpeg* u otro *ffserver*, siendo codificadas en una gran cantidad de formatos y resoluciones. El servidor también posee una página Web de monitoreo y distribución (ver [Anexo I](#)), donde se señalan los flujos disponibles y los *codecs* utilizados.

Dentro de los parámetros configurables por el servidor se encuentra el número del puerto de escucha de peticiones, la cantidad máxima de clientes y de ancho de banda máximo a utilizar, los *codecs* de audio y video para cada *Streaming*, así como también la tasa de bits y la resolución del video, entre otras opciones.

2.7.3 VideoLAN

El proyecto VideoLAN presenta al VLC el cual es un término que proviene de las siglas de la frase en inglés *VideoLAN Client*, el cual puede ser usado como servidor para transmitir archivos MPEG-1, MPEG-2 y MPEG-4, DVDs y video en vivo sobre la red en *unicast* o *multicast*, o usado como cliente para recibir, decodificar y visualizar flujos MPEG sobre varios sistemas operativos. Vale destacar que inicialmente solo fue concebido como aplicación cliente del proyecto VideoLAN motivo por el cual esta reflejado en su nombre la palabra cliente. (11)

VLC trabaja sobre muchas plataformas: Linux, Windows, Mac OS X, BeOS, *BSD, Solaris, Familiar Linux, Yopy/Linupy y QNX.

Puede leer:

- Archivos MPEG-1, MPEG-2 y MPEG-4 / DivX desde un disco duro, un CD-ROM, ...
- DVDs y VCDs,
- Desde tarjetas receptoras de satélite (DVB-S),
- Flujos MPEG-1, MPEG-2 y MPEG-4 desde la red enviados por la salida de VLS³⁵ o VLC's.

VLC también puede ser usado como servidor para transmitir:

- Archivos MPEG-1, MPEG-2 y MPEG-4 / DivX ,
- DVDs,

³⁵ Inicialmente fue el servidor de Streaming del proyecto VideoLAN.

- Desde una tarjeta codificadora MPEG. (11)

Las transmisiones pueden realizarse a una computadora, conocida como *unicast* o a un grupo dinámico de máquinas a las que el cliente puede conectarse o desconectarse, denominándola *multicast*.

El VLC cuenta con *VideoLAN Manager* (VLM) un pequeño administrador de media diseñado para el control de transmisiones múltiples con una sola instancia de VLC. El VLM puede ser controlado desde dos interfaces, la telnet o la http.

Existen dos tipos en la que se realizan 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.
- *Broadcast*: los usuarios no pueden interactuar con el recurso, solo se puede manejar desde el servidor. (11)

El proyecto VideoLAN también presenta un *plugins* para el Mozilla³⁶ del VLC, el cual permite embeber el reproductor en el explorador Web.

2.7.4 Fundamentación de la aplicación de Streaming escogida.

Después de un análisis de las dos propuestas se decide utilizar la presentada por el proyecto VideoLAN al poseer una excelente documentación y una interfaz más amistosa, lo que permite una rápida adaptación por parte de usuarios menos experimentados. Además, con este sistema se puede utilizar el VLC como servidor de Streaming, jugando un papel fundamental en la aplicación a desarrollar cuando se va a transmitir la señal en vivo de los canales de televisión externos, los materiales audiovisuales y la música de fondo para las pantallas de textos, texto-imagen o imagen de las noticias.

Para las señales en vivo de los canales de televisión externos se aprovechará la posibilidad de transmitir de forma *Broadcast* por el servidor, quedando un flujo constante que se transmite, el cual puede ser utilizado por la aplicación cuando se quiera hacer un enlace con ellos.

³⁶es un navegador de Internet, con interfaz gráfica de usuario, desarrollado por la Corporación Mozilla y un gran número de voluntarios externos.

Por otro lado para los videos de las pantallas de tipo video y las canciones que estarán de fondo en las pantallas de texto, texto-imagen o imagen de las noticias se utilizará la transmisión VOD, disponible también en el VLC, lo que permitirá utilizar los recursos cuando corresponda reproducirlos. Para garantizar poner a disposición los recursos de forma VOD automáticamente, se implementará en la aplicación, a la hora de crear la cartelera del ciclo noticioso, una función que genera las líneas de comando para cada uno de los recursos que se tenga que utilizar. Las líneas de comando son ejecutadas una vez generadas.

Por último, el *plugins* que trae para Mozilla es un recurso de suma importancia al poderlo utilizar, con el objetivo de embeber el reproductor en la pagina Web, que en si constituirá la aplicación, para mostrar los Streaming que se generarán en el servidor.

2.8 Sistema operativo.

El sistema operativo es el programa más importante de un ordenador. Para que funcionen los otros programas, cada ordenador de uso general debe tener un sistema operativo.

Los sistemas operativos realizan tareas básicas, tales como reconocimiento de la conexión del teclado, enviar la información a la pantalla, no perder de vista archivos y directorios en el disco, y controlar los dispositivos periféricos tales como impresoras, escáner, etc. (3)

En la actualidad existen sistemas operativos libres y propietarios. Básicamente lo que hace la diferencia entre estos dos tipos de sistemas operativos es que para los libres no es necesario pagar por su uso, mientras todo lo contrario para los propietarios.

Al ser requisito imprescindible desarrollar y montar la aplicación sobre un sistema operativo libre se buscó una alternativa que fuera apropiada para ello. Nova LNX fue el sistema operativo escogido, del cual se hace una referencia a continuación.

2.8.1 Nova LNX.

Nova LNX es una personalización de la distribución GNU/Gentoo Linux desarrollada por el proyecto Nova de la Universidad de las Ciencias Informáticas.

El proyecto Nova garantiza obtener Nova LNX adaptado para el entorno escritorio de las maquinas de desarrollo, y otro para los servidores donde se montará la aplicación, con facilidades para su instalación y mejoras para incrementar su rendimiento. Este aspecto anteriormente señalado fue de mucho peso para la elección de este sistema operativo, ya que todas las distribuciones Linux tienen

sus “sabores” característicos y son orientadas a las necesidades de sus desarrolladores, las cuales no se adaptan en su forma nativa a las necesidades del entorno en el cual van a ser utilizadas, por lo que pueden aparecer un sinnúmero de problemas de compatibilidad (16). Además de este aspecto, se encuentra la posibilidad de soporte, el cual estaría garantizado en todo momento, tanto en el desarrollo, como en el despliegue de la aplicación.

Conclusiones.

En este capítulo se ha realizado un análisis de las tecnologías y tendencias actuales sobre las que se trabajará para obtener la aplicación que se desea, profundizándose en el conocimiento de conceptos necesarios para la comprensión del trabajo. Se dio a conocer como influyen todos estos elementos en la confección del mismo y las características que hacen que sean escogidos.

CAPÍTULO 3

Presentación de la solución propuesta

Introducción.

En este Capítulo se da a conocer el modelado del negocio, el mismo incluirá los procesos y actores del negocio, los casos de uso del negocio y el diagrama de actividades correspondiente por cada caso de uso. Además se expondrán los requerimientos funcionales y no funcionales que se identificaron para el sistema. Posteriormente se describe el sistema propuesto donde se incluye la descripción de los actores del sistema, diagrama de casos de uso del sistema y la descripción textual de los casos de uso.

3.1 Modelo de Negocio.

Cuando se va a desarrollar un sistema para una organización, institución o empresa es necesario comprender la estructura y la dinámica que tienen, así como los problemas actuales que presentan, para con ello, identificar las mejoras potenciales que se pueden realizar. Para lograr esos propósitos, el modelado del negocio permite obtener una visión de la organización, institución o empresa describiendo cada proceso del negocio, especificando sus datos, actividades, roles y reglas de negocio.

3.1.1 Procesos del Negocio.

Un proceso del negocio es el conjunto estructurado de las actividades que han sido diseñadas para producir un resultado específico para un cliente o el mercado. Debe haber un enfoque a la lógica del negocio de dicho proceso, desde la perspectiva del producto.

A continuación se muestra una descripción del proceso del negocio propuesto:

Proceso de Transmisión: se realiza la transmisión de los materiales audiovisuales y la señal de televisoras externas como Venezolana de Televisión por el canal de televisión interno.

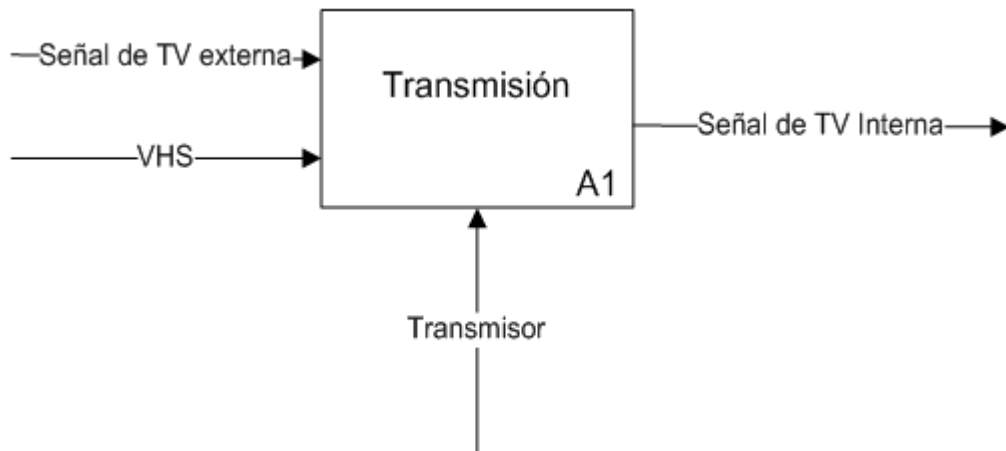


Figura 6. Diagrama del proceso de transmisión.

3.1.2 Actores del negocio.

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa (1). Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados.

Actores del Negocio	Descripción
Operador Técnico	Es el que inicia y mantiene la transmisión de la señal del canal de televisión, poniendo los materiales audiovisuales en los equipos reproductores, VHS y DVD, además de realizar el encadenamiento con las señales de televisión externas.

Tabla 2. Descripción de los actores del negocio.

3.1.3 Diagrama de Casos de Uso del Negocio.

Para tener una visión general de los diferentes procesos de negocio, se construye un diagrama de casos de uso del negocio, en el cual aparece cada proceso del negocio como un caso de uso. El diagrama de casos de uso del negocio representa gráficamente a los procesos del negocio y su interacción con los actores del negocio. (1)



Figura 7. Diagrama de Casos de Usos del Negocio

3.1.4 Descripción textual de los Casos de Uso de Negocio.

Nombre del CU	Transmitir canal
Actores	Operador técnico (Inicia).
Propósito	Realizar la transmisión del canal interno, para brindar información de interés a los trabajadores y visitantes del Ministerio del Poder Popular para la Energía y Petróleo de Venezuela.
Resumen	El caso de uso se inicia cuando el operador técnico pone un material en algunos de los equipos que tiene (VHS y DVD) o pone alguno de los 2 cables por los que llegan las señales de televisoras externas (televisión interna de PDVSA y señal de los canales nacionales) al VHS, en la entrada que tiene para cable coaxial, la cual es difundida por la red de televisión interna del ministerio, conectando la salida del VHS al transmisor de la institución. La transmisión se continua haciendo cambios entre los componentes con los que se cuenta, durante todo el día, ya sea suministrando nuevos materiales al DVD y VHS o conectando alguna de las televisoras externas. Se concluye el caso de uso cuando el Operador técnico termina su horario laboral y desconecta todos los equipos.

Curso normal de los eventos	
Acciones del actor	Respuestas del proceso del negocio
1. El Operador técnico inserta el material audiovisual en el reproductor que lo soporta (VHS o DVD).	2. Se realiza la transmisión de la señal del canal de televisión interno.
3. Monitorea la señal del canal.	
4. Realiza el cambio del contenido que se está transmitiendo. Pasa a la acción 1	
Curso alternativo de los eventos	
Acciones del actor	Respuestas del proceso del negocio
Acción1. El Operador técnico conecta el cable de la señal de televisión externa al VHS.	1. Pasa a la acción 2 del curso normal de eventos.
Acción 4. Termina la transmisión del canal de televisión interno.	
Prioridad	Media
Mejoras	<p>Se automatizará:</p> <ul style="list-style-type: none"> • La reproducción de los materiales audiovisuales. • Los cambios entre la señal del canal interno y la de los externos. • La transmisión de noticias informativas en forma de texto. • La visualización de infocintas promocionando eventos de importancia para la institución, noticias relevantes que se transmitirán o información en general para los televidentes.

Tabla 3. Descripción detallada del CU del negocio Transmitir canal.

3.1.5 Diagramas de Actividades para cada Caso de Uso del Negocio.

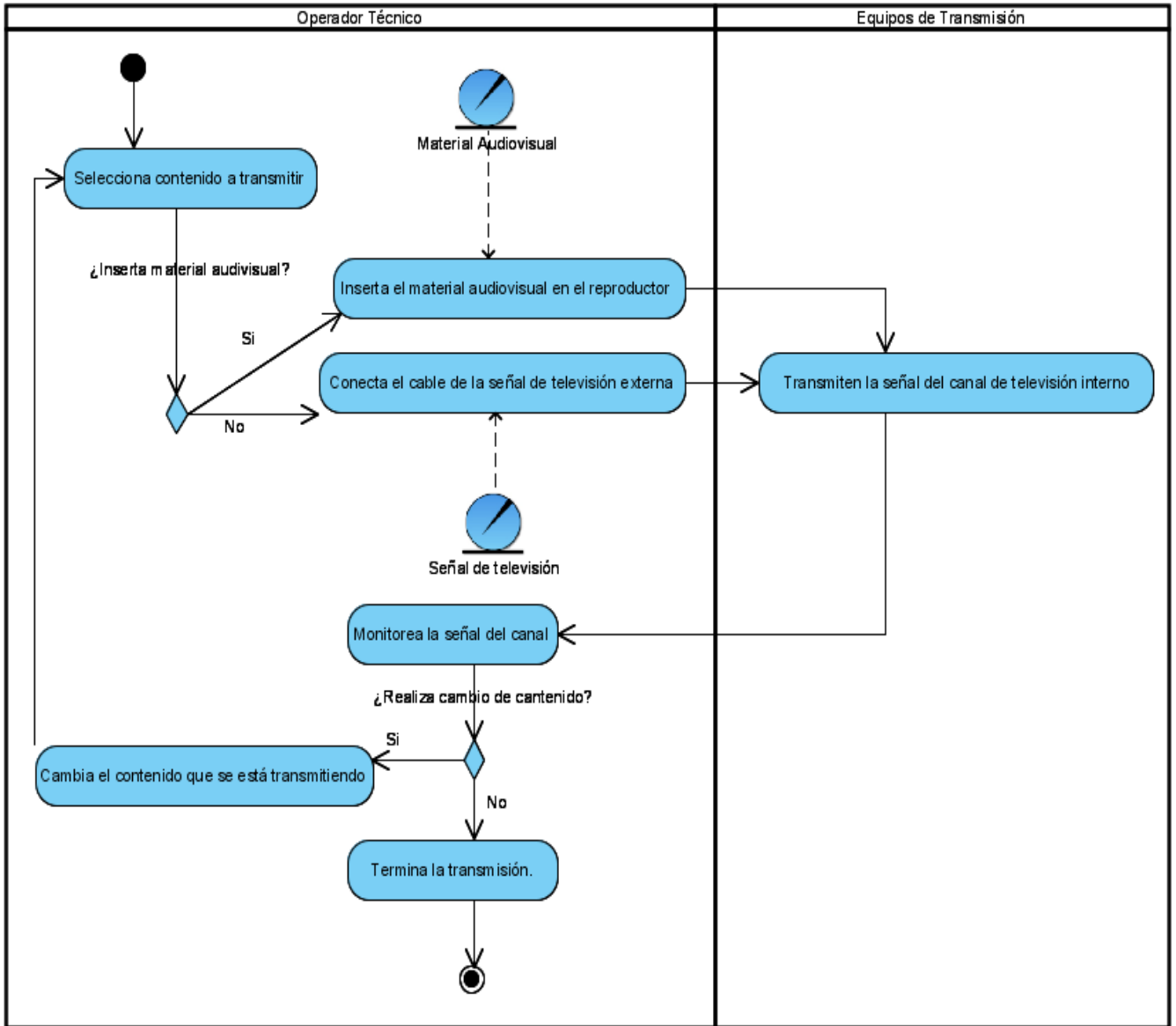


Figura 8. Diagrama de Actividades para el CU del Negocio Transmitir canal

3.2 Requerimientos Funcionales.

Los requerimientos funcionales definen las acciones que debe realizar el sistema desde el punto de vista de las necesidades del usuario (1). A continuación se relacionan los que se establecieron para el subsistema:



- R1.** Verificar señal: deberá identificar el estado de la señal a transmitir y en dependencia de esto visualizará la señal determinada por el Operador de Transmisiones.
- R2.** Mostrar cartelera: el sistema deberá visualizar una cartelera del ciclo de trasmisión como inicio de cada ciclo, mostrando para cada noticia la sección temática y el titular, en el orden que serán mostrados posteriormente.
- R3.** Visualizar noticias: deberá mostrar las noticias publicadas teniendo en cuenta su período de publicación y el formato de sus pantallas. Se mostrará la sección temática a la que pertenece la noticia, el titular y el contenido de la pantalla, en el caso de las pantallas de tipo imagen y video se mostrarán los mismos a pantalla completa, ocultando todo el contenido del canal. Para las pantallas de tipo imagen se mostrará el pie de imagen especificado en la redacción de la noticia.
- R4.** Mostrar información adicional: se mostrará información adicional para la cartelera y pantallas de tipo texto y texto-imagen, mostrándose la fecha en formato abreviado y la hora. En el caso específico de las pantallas de texto y texto-imagen se mostrará además el tiempo restante de visualización de la pantalla y titular de la próxima noticia y sección temática.
- R5.** Reproducir fondo musical: se deberá reproducir un fondo musical para el canal el cuál estará dado por la música especificada opcionalmente para cada noticia, en caso que no se haya definido tema musical para la noticia entonces se escuchará un tema escogido aleatoriamente de un grupo genérico. En caso que se muestre un video el fondo musical del canal no se escuchará.
- R6.** Mostrar infocintas: se visualizarán las infocintas publicadas teniendo en cuenta su período de publicación y la sección temática a la que pertenece.
- R7.** Trasmitir video en vivo: deberá ser capaz de transmitir la señal de video externa.
- R8.** Mostrar patrón del canal: en caso que el canal pase a fuera de servicio se deberá mostrar una imagen de patrón de canal.

3.3 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 o confiable (1). A continuación se relacionan los requerimientos no funcionales que debe cumplir el sistema clasificados según el conjunto de aspectos que representan.

- **Usabilidad.**

- El uso del diseño debe ser fácil de entender, atendiendo a la experiencia, conocimiento y habilidades del usuario.

- **Apariencia o interfaz externa.**

- El diseño debe comunicar de manera eficaz la información necesaria.
- El diseño debe estar en correspondencia con la identidad de la institución.
- El formato de salida debe ser de 16:9 panorámico.
- Establecer una división en secciones temáticas de manera visual y no por bloques de noticias de forma tal que se puedan alternar noticias de diferentes naturalezas sin que por esto se pierda la referencia temática de las mismas.
- Los tipos de pantallas para las noticias se definen de cuatro maneras: solo texto, texto e imagen, solo imagen y video.
- Uso de video de presentación y despedida de la transmisión.

- **Seguridad.**

- La información será objeto de cuidadosa protección contra la corrupción y estado inconsistente.
- Deberán existir mecanismos de chequeo de integridad.
- Se deberán hacer copias de respaldo que puedan restaurar el sistema en caso de fallo crítico o pérdida total de la información.

- **Software.**

- Los servidores tendrán instalada la distribución Nova/Linux como sistema operativo.
- El Gestor de Base de Datos a utilizar debe ser PostgreSQL.

- **Hardware.**

Servidor	Procesador	Memoria RAM	Disco Duro	Tarjeta de Red	Tarjeta de Video
Base de Datos y de Streaming.	Dual-Core Xeon 2.33 GHz	4 Gb	500 Gb	2 puertos Ethernet Gigabit	Capturadora Hauppauge WinTV PVR 350
Transmisión	Dual-Core Xeon 1.60 GHz	1 Gb	80 Gb	Ethernet Gigabit	Exportadora ATI Radeon X300

Tabla 4. Requerimientos de Hardware.

3.4 Descripción del Sistema Propuesto.

Después de describir el proceso de negocio y especificar las actividades a automatizar en el sistema, se puede pasar a modelar y dar la propuesta del sistema.

3.4.1 Descripción de los actores del sistema.

Un actor del sistema es un rol que un usuario juega con respecto al sistema. Es importante destacar que un actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema, es decir que un equipo o un sistema automatizado, son parte del sistema, y pueden intercambiar información con él o ser recipientes pasivos de información.

Actores del Sistema	Descripción
Reloj	Controla las acciones necesarias para lograr la visualización de los contenidos en el canal de manera automática.

Tabla 5. Descripción de los actores del sistema.



3.4.2 Casos de Uso del Sistema.

El diagrama de casos de uso representa la forma en cómo un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso) (1). Se puede decir que es una técnica para capturar información de cómo un sistema o negocio trabaja, o de cómo se desea que trabaje. Los Casos de Uso según describen, bajo la forma de acciones y reacciones, el comportamiento de un sistema desde el punto de vista del usuario. Además permiten definir los límites del sistema y las relaciones entre el sistema y el entorno. (1)



Figura 9. Diagrama de CUS Subsistema de Transmisión.

3.4.3 Descripción textual de casos de usos del sistema.

3.4.3.1 Descripción de Casos de Uso Transmitir canal.

Caso de Uso: CU-1	<i>Transmitir canal.</i>
Actores:	Reloj.
Resumen:	El caso de uso se inicializa cuando el actor Reloj detecta que existe alguna señal activada para transmitir. Se muestra inicialmente el video de presentación del canal. Si la señal activa es la de las televisoras nacionales o la del canal de PDVSA se muestra la presentación del enlace y seguidamente la señal. En el caso de que sea la propia del canal, se muestra la cartelera generada para el ciclo noticioso correspondiente. Al terminar de mostrar la cartelera se mostrarán las noticias según el orden que se estableció. Al inicio de cargar cada pantalla que conforman la noticia se verificará si existe algún

	cambio de señal en ese instante o en el intervalo que duraría mostrar la pantalla. Se mostrarán además las infocintas publicadas que estén relacionadas a la sección temática a la que pertenece la noticia.
Referencia:	R1, R2, R3, R4, R5, R6, R7, R8.
CU asociados:	<i>Buscar Noticia</i>
Precondiciones:	Existe una señal de las posibles a transmitir activa.
Flujo Normal de Eventos	
Sección: “Transmitir canal”	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicializa cuando el actor Reloj detecta alguna señal activa para transmisión.	2. Muestra el video de presentación del canal. Si la señal activa es: <ul style="list-style-type: none"> a. <i>Señal de canales externos</i>, ver sección Visualizar señal de canales externos. b. <i>Señal del canal</i>, ver sección Visualizar canal. c. <i>Fuera de servicio</i>, ver sección Fuera de servicio.
Sección: “Visualizar señal de canales externos”	
Acción del Actor	Respuesta del Sistema
	1. Muestra la presentación del enlace con otra televisora y posteriormente la señal del canal externo.
Sección: “Visualizar canal”	
Acción del Actor	Respuesta del Sistema
	1. Se buscan las noticias publicadas que su intervalo de publicación coincide con la fecha y hora actual.
	2. Crea la cartelera con las noticias resultantes de la búsqueda que se



	<p>realizó, ordenándolas descendientemente según la fecha de publicación, mostrándola posteriormente. En la cartelera se mostrará número del orden de aparición de la noticia, sección temática a la que pertenece y su título. También estará acompañada de una música de fondo.</p>
	<p>3. Visualiza cada una de las noticias del ciclo noticioso. Para cada noticia se muestran las pantallas que la componen, en caso de que sea:</p> <ul style="list-style-type: none"> a. <i>Pantalla texto</i>, se muestra el texto que conforma la pantalla. Muestra la infocinta correspondiente, según el orden de visualización, del listado de infocintas de la sección temática de la noticia que se está transmitiendo. b. <i>Pantalla texto-imagen</i>, se muestra la imagen correspondiente, más el texto que conforma la pantalla. La imagen tendrá un espacio establecido donde se mostrará, por lo que la resolución será ajustada a él. Muestra además la infocinta correspondiente, según el orden de visualización, del listado de infocintas de la sección temática de la noticia que se está transmitiendo. c. <i>Pantalla imagen</i>, se muestra la imagen más el cintillo que tendrá la

	<p>información correspondiente a ella.</p> <p>d. Pantalla video, se muestra el video que tiene relacionado la pantalla.</p> <p>La señal del canal tendrá elementos que serán generales para todas las pantallas excepto para la de tipo video e imagen, los cuales son: fecha en formato abreviado, hora, tiempo restante de la pantalla, sección temática a la que pertenece, titulo de la noticia y titular de la próxima.</p> <p>Durante la visualización de las noticias se reproducirá una música de fondo, excepto en las pantalla de tipo video. La noticia puede tener un tema musical definido, en caso contrario se seleccionará uno de forma aleatoria de una lista genérica de canciones.</p>
<p>3. Al finalizar la visualización de cada pantalla, verifica si existe algún cambio de señal programado en ese instante de tiempo o durante el período de muestra de la pantalla.</p>	<p>4. Si:</p> <p>a. Existe cambio: regresa a la acción 1 de la sección Transmitir canal.</p> <p>b. No existe cambio: Continúa con la visualización de las noticias y sus pantallas.</p>
	<p>5. Al finalizar el ciclo noticioso regresa a la acción 1 de la sección Visualizar canal.</p>
<p>Sección: “Fuera de servicio”</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>

1. Muestra una imagen indicando que el canal esta fuera de servicio.

Prototipo de Interfaz



Pantalla de tipo Texto.



Pantalla de tipo Texto-Imagen.



Pantalla de tipo Imagen.



Pantalla de tipo Video.



Pantalla de la Cartelera.

Poscondiciones:	Se logra la transmisión del canal.
Prioridad:	Alta.
Especificaciones Complementaria:	En la acción 1 se buscan las noticias publicadas para lo cual hay que ver el CU Buscar noticia .

Tabla 6. Descripción detallada del CU del sistema Transmitir canal.



Conclusiones.

En este capítulo con la descripción del negocio se obtiene una mejor comprensión de como ocurren los procesos en la realidad dentro de la instalación estudiada. Se plantean los requisitos que se deberán cumplir, resultando ocho requisitos funcionales y diversos no funcionales. Además se describe la propuesta del sistema donde se ve cómo se resuelve el problema planteado.

CAPÍTULO 4

Construcción de la solución propuesta

Introducción.

En este capítulo se expone el Diagrama de clases del diseño que respalda al subsistema, el cual constituye la base para el flujo de implementación. Se detallan las pautas que se definieron para estandarizar la interfaz del subsistema. Además se presenta el diseño de la Base de datos, el cual se obtendrá a partir del modelo de datos resultante del diagrama de clases persistentes. Por último se muestran los modelos de despliegue e implementación, que resultaron del diseño realizado de los casos de uso del sistema.

4.1 Diagramas de Clases del diseño.

El diagrama de clase que se muestran a continuación brinda un mayor acercamiento a la forma y al contenido de la solución propuesta. De este modo se ha elaborado el diagrama de acuerdo con la descripción del sistema brindada en el capítulo anterior.

4.1.1 Diagrama de Clases del diseño “Subsistema de transmisión”.

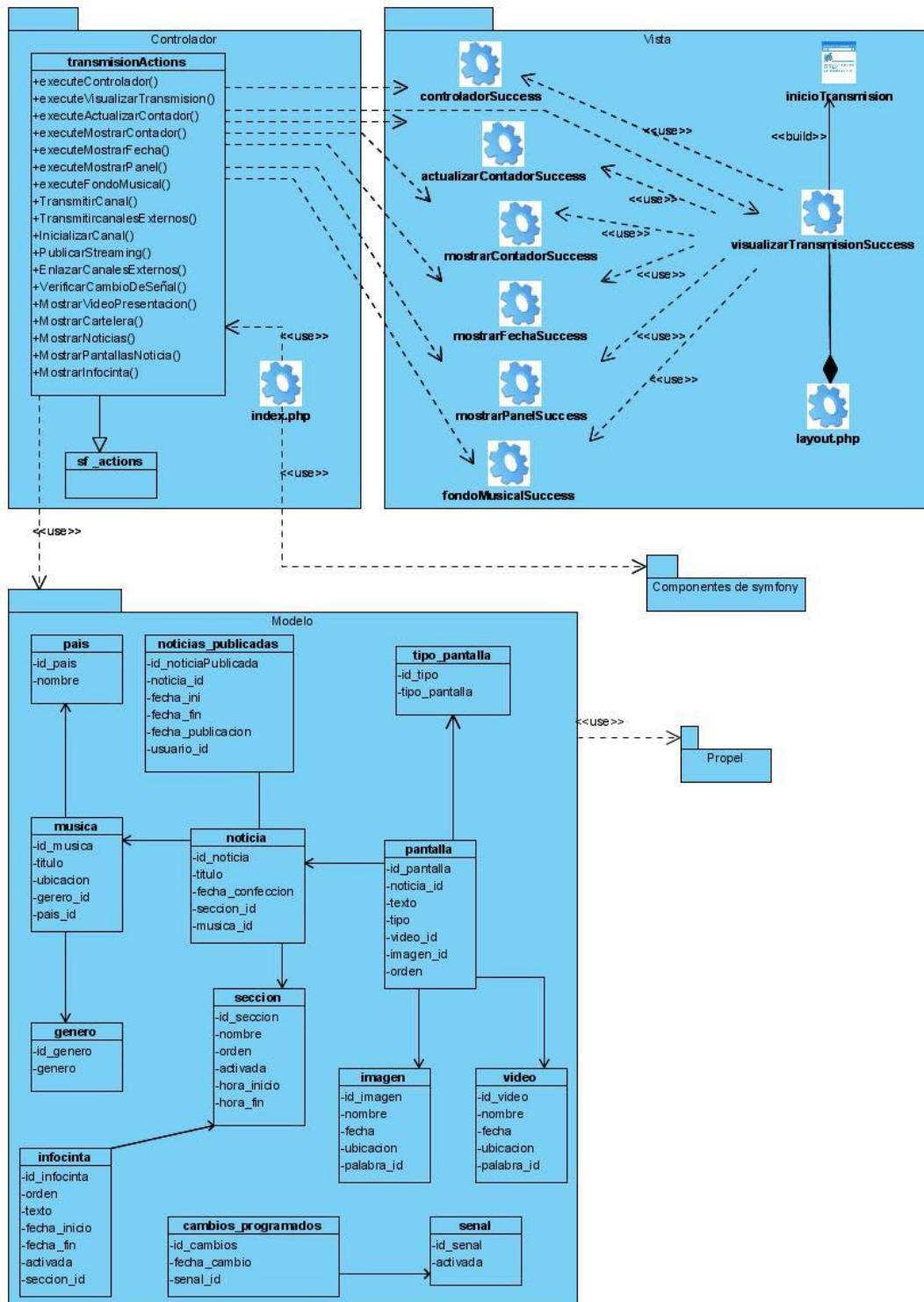


Figura 10. Diagrama de Clases del diseño “Subsistema de transmisión”.



4.2 Principios de diseño.

4.2.1 Estándares de la interfaz.

Para el diseño de la interfaz se utilizó el esquema Cabecera-Navegador-Contenido que constituye uno de los más usados actualmente. Se ofrece en la parte superior derecha información acerca de la fecha y hora (A) y en la izquierda el nombre del canal (B). En la parte derecha se brindan datos relacionados con la noticia que se está transmitiendo como son: la sección temática a la que pertenece, número de la pantalla que se está visualizando del total que tiene la noticia, el tiempo restante de visualización de dicha pantalla, así como la sección temática y el titular de la noticia que se mostrará a continuación (C). Las pantallas de tipo texto y texto-imagen serán mostradas en la parte central (E), mientras que las de imágenes y videos se visualizarán a pantalla completa.

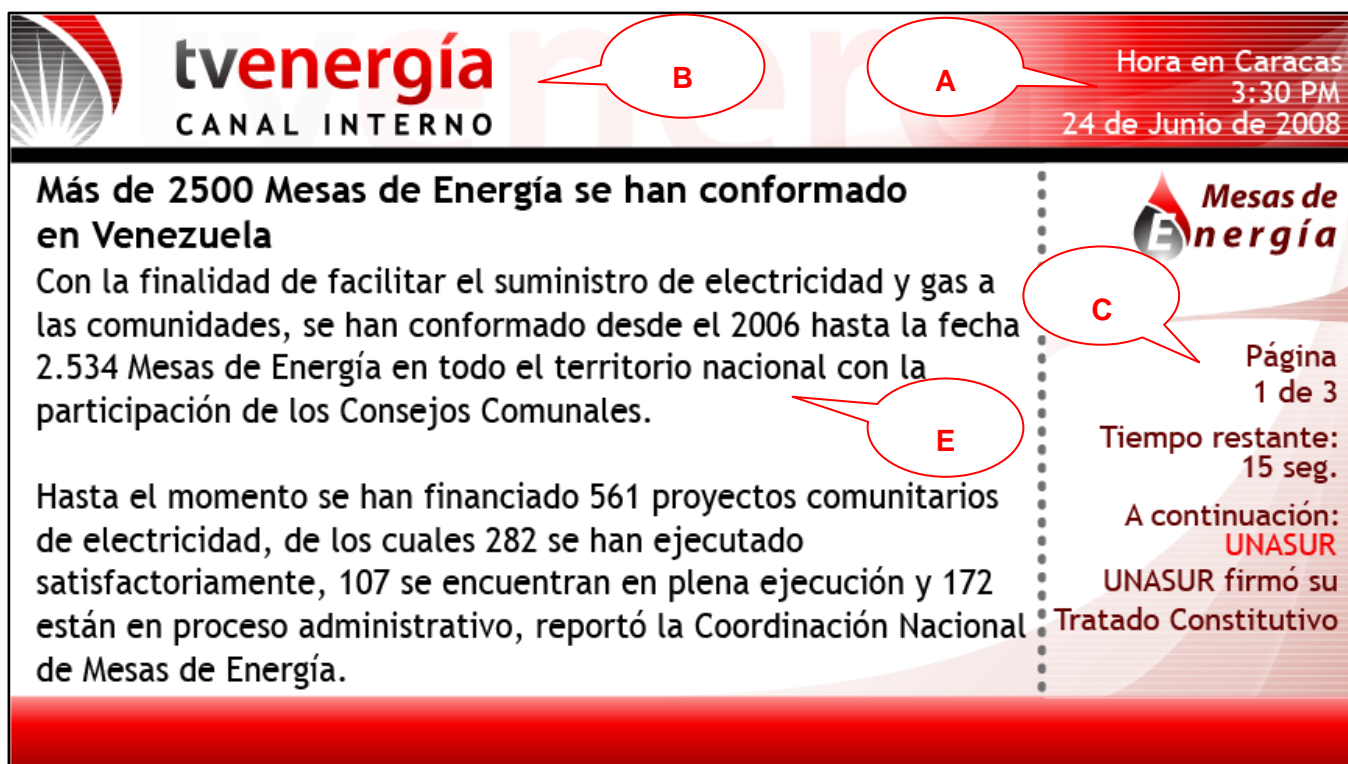


Figura 11. Diseño de Interfaz: Subsistema de Trasmisión.

Se definieron logotipos que identifican a cada una de las secciones temáticas. Para establecer el tipo y tamaño de fuente de los distintos elementos de cada página se crea una hoja de estilo. Se utiliza en general la familia de fuentes *Trebuchet MS* de tamaño entre 24 píxeles para los textos hasta 28 para los títulos. ¡Error! Marcador no definido.

4.3 Estándares de Codificación.

Para que el código generado en la implementación del subsistema sea de un mejor entendimiento es necesario establecer un estándar de codificación a usar.

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código (6). En aras de obtener el subsistema con una homogeneidad en su código se establece lo siguiente:

- Empleo de comentarios en todas las funciones donde se explique el objetivo de la misma.
- Utilización de nombres para los métodos y variables, que expresen total o parcialmente su significado.
- Los nombres de los métodos tienen la forma encamellada, se capitalizan las palabras que componen el nombre del método excepto la primera. Ejemplo: mostrarInfocinta ().
- Colocar espacios en blanco entre operadores lógicos aritméticos y sus operandos.
- Organización del código de forma estructurada, en bloques de código, para una mejor lectura del mismo.

4.4 Diseño de la base de datos.

La base de datos es el sistema utilizado para el almacenamiento de información y acceso controlado a ella. Para diseñar la base de datos se utiliza el diagrama de clases persistentes y el modelo de datos, este último se corresponde con su representación física. En este epígrafe se muestra el diseño de la base de datos del subsistema propuesto a través del diagrama de clases persistentes y el modelo de datos generados a partir de este. En el [Anexo II](#) se pueden consultar las descripciones de las tablas de la base de datos.

4.4.1 Diagrama de clases persistentes.

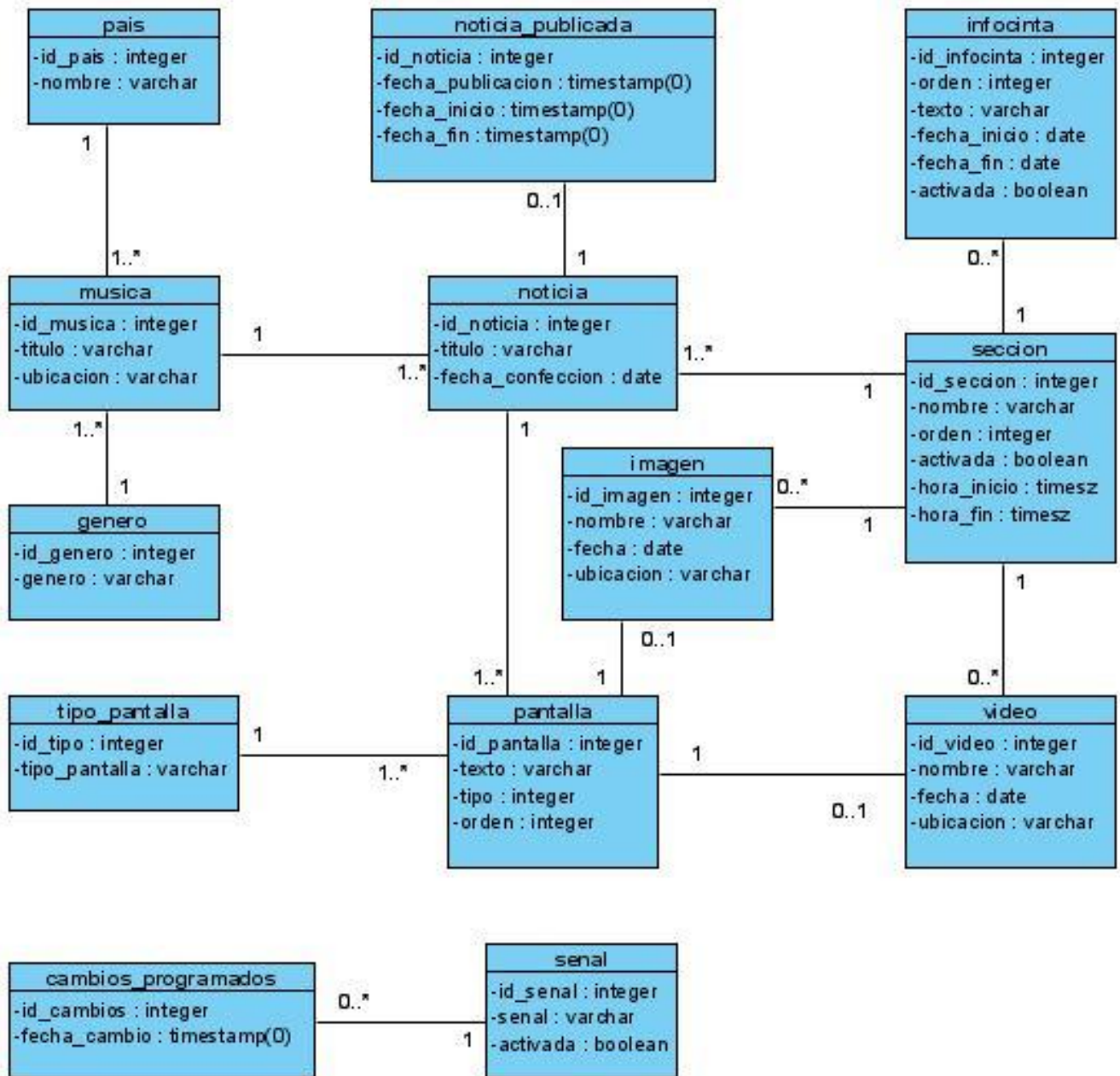


Figura 12. Diagrama de clases persistentes.

4.4.2 Modelo de datos.

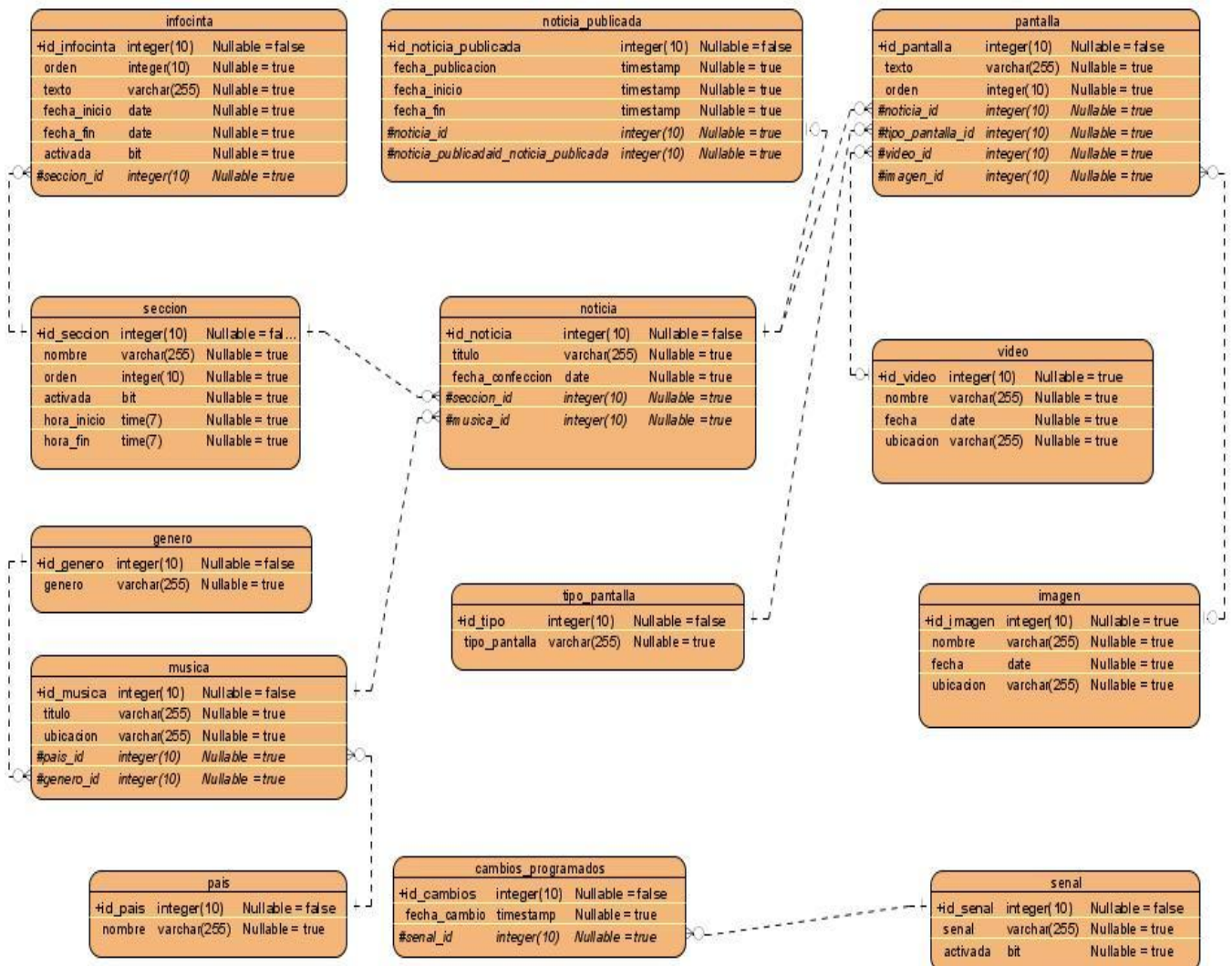


Figura 13. Modelo de datos.

4.5 Modelo de Despliegue.

El modelo de despliegue describe la distribución física del sistema, muestra cómo están repartidos los componentes de software entre los distintos nodos de cómputo. Este modelo lo forma una colección de nodos y arcos; donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de *hardware* similar. La obtención de este modelo permite comprender la correspondencia entre la arquitectura *software* y la arquitectura *hardware*.



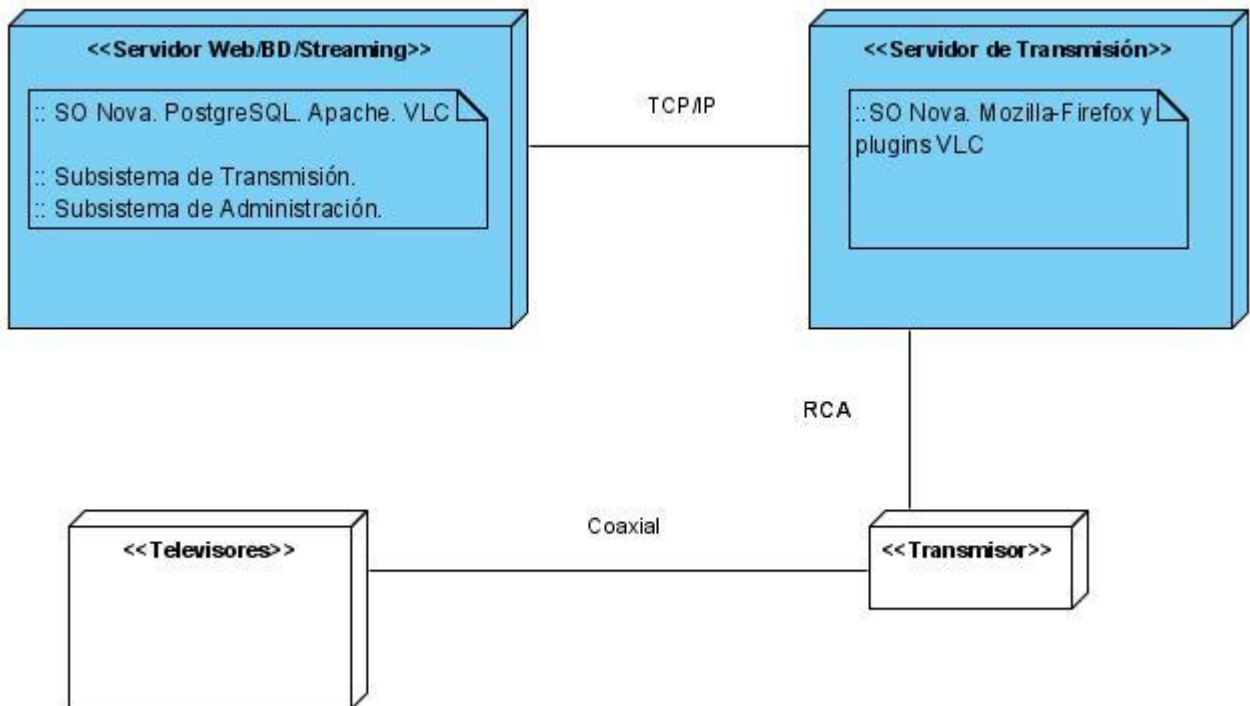


Figura 14. Modelo de despliegue.

4.6 Modelo de Implementación.

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes se pueden encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos. (13)

Este artefacto describe cómo se implementan los componentes, congregándolos en subsistemas organizados en capas y jerarquías, y señala las dependencias entre éstos. Seguidamente se expone el diagrama de componente del subsistema.

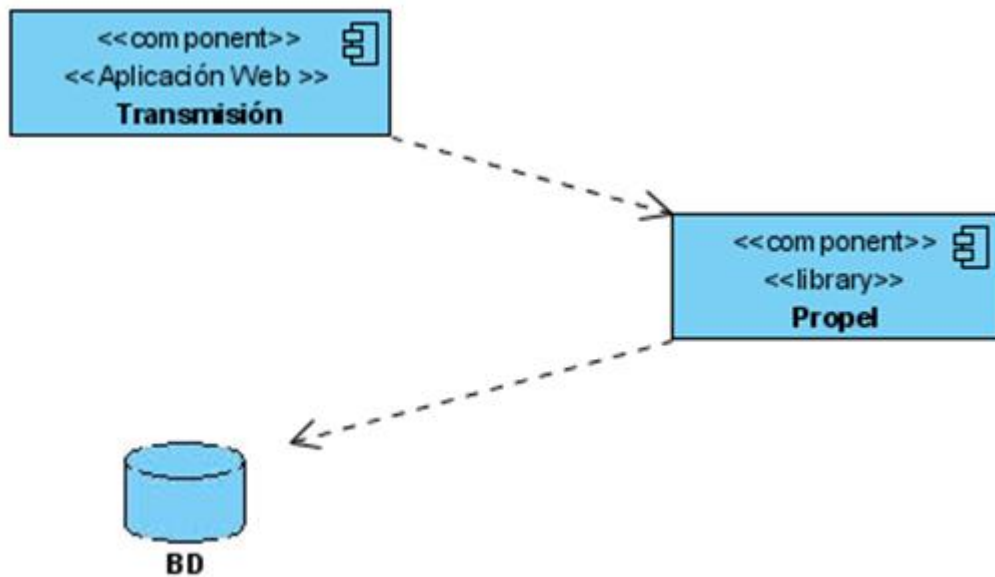


Figura 15. Diagrama de componentes del modelo de implementación.

Conclusiones.

En el capítulo se mostraron los resultados de la etapa de construcción del sistema. Con los resultados alcanzados y a modo de evolución se obtuvieron los diagramas de clases, así como el de clases persistentes y de este último el modelo de datos. Se plantearon los principios de diseño y estándares de interfaz, elementos que ayudan a una mejor recepción de la información por el usuario. Como culminación al diseño se presentó el modelo de implementación describiendo la distribución física del sistema y sus componentes.

CONCLUSIONES

Con la realización de este trabajo de diploma se obtuvo una propuesta que da cumplimiento al objetivo general planteado, al lograr desarrollar un Sistema Automatizado para el MENPET que facilita la transmisión automática y eficiente de las informaciones generadas para su canal de televisión interno, utilizando software libre. Para llegar a estos resultados:

- Se dio una propuesta de las herramientas necesarias para la implementación de la solución.
- Se realizó el análisis y diseño del sistema propuesto, quedando completa la documentación de Ingeniería de Software del mismo.
- Se Implementó el subsistema de Trasmisión del canal informativo.

La adquisición del producto por el MENPET, lo dotará de un medio más para realizar la labor informativa-formativa sobre los trabajadores y visitantes a su sede central en Caracas. Al término del despliegue del sistema aportará al país más del 85 % de su monto general y aportará importantes conocimientos y resultados a los esfuerzos que se realizan por lograr la soberanía tecnológica de la Industria Cubana del Software.

RECOMENDACIONES

Al término de este trabajo se recomienda:

- Dar continuidad al desarrollo y soporte del subsistema, incorporándole el uso de la colocación automática de un cintillo informativo sobre la transmisión en vivo de otras televisoras, en caso de que no se desee transmitir la información que ellas visualizan.
- Implantar esta nueva solución de software libre en los sistemas que actualmente se usan en la Universidad de las Ciencias Informáticas y la Agencia de Cubana de Noticias.
- Proponer la implantación de sistemas de este tipo en hospitales, universidades, hoteles y otras sedes empresariales y ministeriales.

REFERENCIAS BIBLIOGRÁFICAS

1. **Jacobson G. B. Ivar, Rumbaugh James.** *Proceso Unificado de Desarrollo de Software.* Madrid : s.n., 2000. ISBN.
2. **Canós José H., Letelier Patricio y Penadés M^a Carmen.** *Métodologías Ágiles en el Desarrollo de Software.* [Documento] Valencia : Universidad Politécnica de Valencia.
3. Definición de Sistema Operativo. *masadelante.* [En línea] [Citado el: 4 de mayo de 2008.] <http://www.masadelante.com/faq-sistema-operativo.htm>.
4. *Eduangi telecom.* [En línea] [Citado el: 26 de 05 de 2008.] <http://web.madritel.es/personales3/edcollado/ingsw/tema2/2-4.htm>.
5. Enciclopedia Encarta. [En línea] [Citado el: 26 de abril de 2008.] http://es.encarta.msn.com/encyclopedia_961537147/JAVA_lenguaje.html.
6. Estándar para codificación en lenguaje c++. *iteso.* [En línea] [Citado el: 7 de mayo de 2008.] <http://progra.iteso.mx/estandares/estandar%20codificacion%20c++/estandardcodificacion.pdf>.
7. **Fabien, Potencier y François, Zaninotto.** *Symfony, la guía definitiva.* *librosweb.* [En línea] <http://www.librosweb.es/symfony/index.html>.
8. **Garrett, Jesse James.** *Ajax: Un Nuevo acercamiento a las Aplicaciones Web.* [En línea] 18 de febrero de 2005. [Citado el: 25 de abril de 2008.] <http://mmugcba.com.ar/foro/viewtopic.php?id=36>.
9. *Imsol.* [En línea] [Citado el: 25 de abril de 2008.] http://www.imsol.es/20/content_2060.html.
10. Introducción a PostgreSQL - Instalación e inicialización. *Linux-es.* [En línea] [Citado el: 29 de abril de 2008.] <http://www.linux-es.org/node/536>.
11. **Lattre, Alexis de, y otros.** *VideoLAN Streaming Howto.* s.l. : VideoLAN, 2002-2005.
12. **M., Mendoza Luis Eduardo.** *Sistemas de información II Teoría.* [En línea] [Citado el: 25 de 05 de 2008.] <http://prof.usb.ve/lmendoza/Documentos/PS-6116/Teoria%20PS6116%20O-O%20y%20RUP.pdf>.
13. Modelo de despliegue. *merinde.* [En línea] [Citado el: 12 de 05 de 2008.] http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=96&Itemid=297.
14. **Monet, Dominique.** *Le multimedia.* París : Flammarion, 1995. pág. 127 .

15. Newnet. [En línea] [Citado el: 25 de abril de 2008.] <http://www.newnet.cl/streaming.php>..
16. **Orama, Angel Goni.** *NOVA LNX COMO PLATAFORMA DE DESARROLLO*. La Habana : Universidad de las Ciencias Informáticas, 2007.
17. **Pérez, Javier Eguíluz.** Introducción a JavaScript. *librosweb*. [En línea] [Citado el: 26 de abril de 2008.] <http://www.librosweb.es/javascript/>.
18. *Real Academia Española* . [En línea] [Citado el: 25 de 05 de 2008.] http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=teletexto .
19. **Su, Stanley.** *Database Computers: Principles, Architectures and Techniques*. Nueva York : McGraw-Hil, 1988. pág. 511.
20. Sun cerrará algunos componentes nuevos de MySQL. *diarioti*. [En línea] [Citado el: 3 de mayo de 2008.] <http://diarioti.com/gate/n.php?id=17422>.
21. **Vaughan, Tay.** *Todo el poder de multimedia*. México, D.F : McGraw Hill., 1994.

GLOSARIO DE TÉRMINOS

Clases Persistentes	Son aquellas cuyos objetos deben ser almacenados en algún repositorio como una base de datos relacional.
Cliente-Servidor	Esta arquitectura consiste básicamente en que un programa, el Cliente informático realiza peticiones a otro programa, el servidor, que les da respuesta.
Televisión mecánica	Modelo que se basó en el disco de Nipkow mediante el cual una imagen era explorada mecánicamente por un haz luminoso que repercute en una celda fotoeléctrica produciendo una corriente eléctrica variable.
Noticias	Contenidos que se muestran en el canal, estructurado en distintas pantallas según su extensión.
Pantalla de Texto	Pantalla que muestra la información de forma textual.
Pantalla de Texto e Imagen	Pantalla que muestra la información de forma textual, acompañada de una imagen.
Pantalla de Imagen	Pantalla que muestra una imagen acompañada de un cintillo.
Pantalla de Video	Pantalla que muestra la reproducción de un video.
Televisión Digital	Es la difusión de las señales de TV que utiliza la más moderna tecnología digital para transmitir de forma optimizada imagen y sonido de mayor calidad, permitiendo ofrecer adicionalmente otros servicios interactivos o de acceso a la Sociedad de la Información.
Sistemas de autor	Son un tipo de programas informáticos que facilitan la creación de productos multimedia a usuarios sin amplios conocimientos de programación.
Codecs	Son pequeñas aplicaciones que se instalan en el ordenador para reconocer ciertos tipos de vídeos y audios.
Multiplataforma	Es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas combinaciones de hardware y software.
Programación Orientada a Objetos	Es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora. Está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo, y encapsulamiento.
XHTML	Lenguaje extensible de marcado de hipertexto (XHTML, siglas en inglés), es el lenguaje de marcado que tiene, básicamente, las mismas funcionalidades que HTML, pero cumple las especificaciones, más estrictas, de XML.

CSS	Hojas de estilo en cascada (CSS, siglas en inglés), es un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).
DOM	Modelo de objetos en documentos (DOM, siglas en inglés), es una interface independiente de la plataforma y del lenguaje que permite que los programas y scripts tengan acceso dinámicamente y actualicen el contenido, la estructura y estilo de los documentos.
Vsnmatic	Es el playout multiformato y servidor de video diseñado para automatizar una sala de Continuidad 24x7 y controlar un canal de TV automáticamente y sin fallos.
VideoMat MPro+	Sistema de continuidad automatizada para TV.
TVmaker	Es un sistema con tecnologías MPEG2 que aporta soluciones realmente eficaces, estables y de altas prestaciones a la automatización de continuidad.
BBC	Corporación Británica de Radiodifusión (BBC, siglas en inglés) s el servicio público de radio, televisión e internet del Reino Unido, y opera bajo un Estatuto Real que garantiza la independencia de controles comerciales y/o políticos.
Triggers	Es un evento que se ejecuta cuando se cumple una condición establecida al realizar una operación de inserción (INSERT), actualización (UPDATE) o borrado (DELETE).
Rollback's	Es una operación que devuelve a la base de datos a algún estado previo.

ANEXOS

ANEXO I. PÁGINA DE ESTADO DEL SERVIDOR FFSERVER.

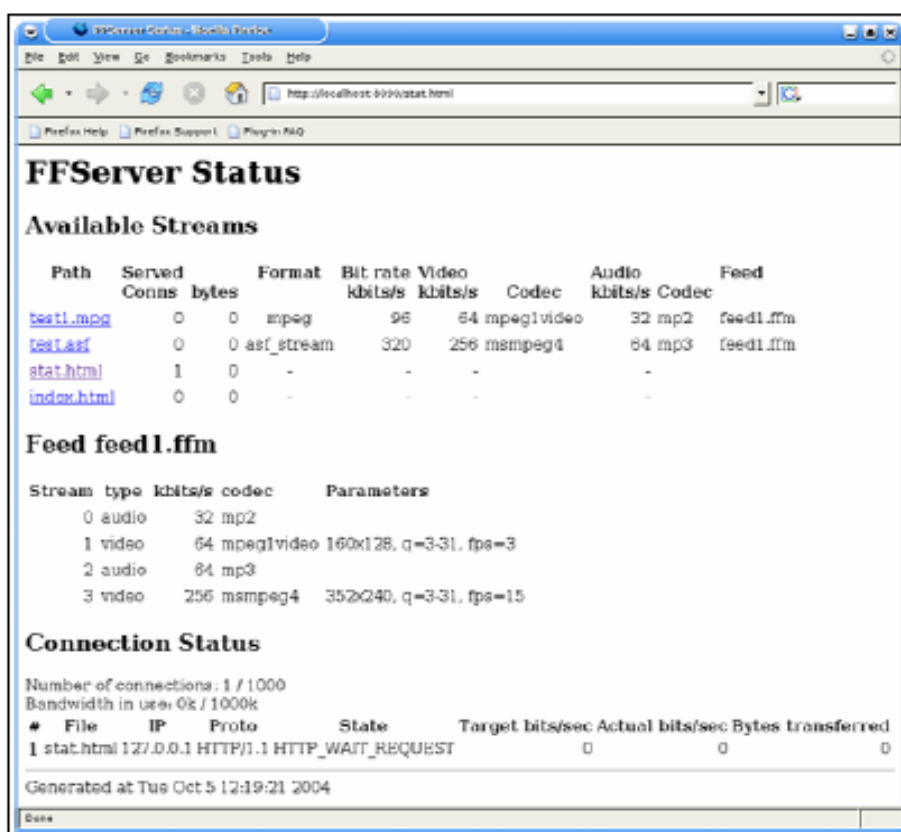


Figura 16. Página de estado del FFServer.

ANEXO II. DESCRIPCIÓN DE LAS TABLAS DE LA BD

Nombre: infocinta		
Descripción: Esta tabla contiene las infocintas que se mostrarán en cada sección		
Atributo	Tipo	Descripción
id_infocinta	integer(10)	Identificador de la infocinta
orden	integer(10)	Orden en que se mostrarán
texto	varchar(255)	Texto de la infocinta
fecha_inicio	date	Fecha de inicio de publicación de la infocinta
fecha_fin	date	Fecha de fin de publicación de la infocinta
activada	bit	Determina si se mostrará
sección_id	integer(10)	Identificador de la sección asociada

Tabla 7. Descripción de la BD: Tabla infocinta.

Nombre: noticia		
Descripción: Esta tabla contiene todas las noticias del sistema		
Atributo	Tipo	Descripción
id_noticia	integer(10)	Identificador de la noticia
titulo	varchar(255)	Título de la noticia
fecha_confeccion	date	Fecha en que se redactó la noticia
usuario_id	integer(10)	Identificador del usuario asociado
seccion_id	integer(10)	Identificador de la sección asociada
musica_id	integer(10)	Identificador de la música asociada

Tabla 8. Descripción de la BD: Tabla noticia.

Nombre: musica		
Descripción: Esta tabla contiene todos los datos de los temas musicales.		
Atributo	Tipo	Descripción
id_musica	integer(10)	Identificador de la música
titulo	varchar(255)	Título de la música
ubicacion	varchar(255)	Ubicación física de la música
genero_id	integer(10)	Identificador del género asociado
pais_id	integer(10)	Identificador del país asociado

Tabla 9. Descripción de la BD: Tabla música.

Nombre: seccion		
Descripción: Esta tabla contiene todas las secciones del sistema		
Atributo	Tipo	Descripción
id_seccion	integer(10)	Identificador de la sección
nombre	varchar(255)	Nombre de la sección
orden	integer(10)	Orden en que se encuentran las secciones
activada	bit	Determina si se mostrará
hora_inicio	time(7)	Hora en que comienza a mostrarse la sección
hora_fin	time(7)	Hora en que termina de mostrarse la sección

Tabla 10. Descripción de la BD: Tabla sección.

Nombre: pantalla		
Descripción: Esta tabla contiene las pantalla de las noticias		
Atributo	Tipo	Descripción
id_pantalla	integer(10)	Identificador de la pantalla
noticia_id	integer(10)	Identificador de la noticia
texto	varchar(255)	Texto de la pantalla
tipo	integer(10)	Tipo de pantalla
video_id	integer(10)	Identificador del video asociado
imagen_id	integer(10)	Identificador de la imagen asociado
orden	integer(10)	Orden en que se mostrarán las pantallas

Tabla 11. Descripción de la BD: Tabla pantalla.

Nombre: genero		
Descripción: Esta tabla contiene los géneros musicales		
Atributo	Tipo	Descripción
id_genero	integer(10)	Identificador del género musical
genero	varchar(255)	Nombre del género musical

Tabla 12. Descripción de la BD: Tabla genero.

Nombre: video		
Descripción: Esta tabla contiene todos los datos de los videos		
Atributo	Tipo	Descripción
id_video	integer(10)	Identificador del video
nombre	varchar(255)	Nombre del video
fecha	date	Fecha relaciona con el video
ubicacion	varchar(255)	Ubicación física del video
palabra_id	integer(10)	Identificador de la palabra clave asociada

Tabla 13. Descripción de la BD: Tabla video.

Nombre: imagen		
Descripción: Esta tabla contiene todos los datos de las imágenes		
Atributo	Tipo	Descripción
id_imagen	integer(10)	Identificador de la imagen
nombre	varchar(255)	Nombre de la imagen
fecha	date	Fecha relacionada con la imagen
ubicacion	varchar(255)	Ubicación física de la imagen
palabra_id	integer(10)	Identificador de la palabra clave asociada

Tabla 14. Descripción de la BD: Tabla imagen.

Nombre: pais		
Descripción: Esta tabla contiene los países a los que puede pertenecer una música		
Atributo	Tipo	Descripción
id_pais	integer(10)	Identificador del país
nombre	varchar(255)	Nombre del país

Tabla 15. Descripción de la BD: Tabla país.

Nombre: noticias_publicadas		
Descripción: Esta tabla contiene las noticias publicadas		
Atributo	Tipo	Descripción
id_noticiaPublicada	integer(10)	Identificador de la noticia publicada
noticia_id	integer(10)	Identificador de la noticia
fecha_ini	timestamp	Fecha de inicio de publicación
fecha_fin	timestamp	Fecha de fin de publicación
fecha_publicacion	timestamp	Fecha en que se publicó la noticia
usuario_id	integer(10)	Identificador del usuario asociado

Tabla 16. Descripción de la BD: Tabla noticias_publicadas.

Nombre: cambios_programados		
Descripción: Esta tabla contiene los cambios de señal		
Atributo	Tipo	Descripción
id_cambios	integer(10)	Identificador de los cambios programados
fecha_cambio	timestamp	Fecha en que realizará el cambio
senal_id	integer(10)	Identificador de la señal asociada al cambio

Tabla 17. Descripción de la BD: Tabla cambios_programados.

Nombre: senal		
Descripción: Esta tabla contiene las señales que se pueden transmitir		
Atributo	Tipo	Descripción
id_senal	integer(10)	Identificador de la señal
senal	varchar(255)	Nombre de la señal
activada	bit	Determina si se mostrará

Tabla 18. Descripción de la BD: Tabla senal.

Nombre: tipo_pantalla		
Descripción: Esta tabla contiene los tipos de pantallas		
Atributo	Tipo	Descripción
id_tipo	integer(10)	Identificador del tipo de pantalla
tipo_pantalla	varchar(255)	Nombre del tipo de pantalla

Tabla 19. Descripción de la BD: Tabla tipo_pantalla.

ANEXO III. COMPARACIÓN DE LOS FRAMEWORK PARA PHP SEGÚN WWW.LINUX.PUCP.EDU.PE.

Características	Cake PHP	Symfony	Zend FW
Arquitectura de aplicaciones			
Incorporación del patrón Modelo Vista Controlador orientado a objetos.	X	X	X
Operaciones CRUD (Create, Retrieve, Update y Delete) asociadas a patrón Active Record.	X	X	
Mapeado de objetos a bases de datos relacionales (ORM).	X	X	
Independiente del manejador de base de datos.	X	X	X
Estructura por defecto para aplicaciones (scaffolding).	X	X	
Archivos de configuración de la aplicación.	.php	.yml*	.php
Acceso vía Web			
Despachador de peticiones HTTP.	X	X	X
Generación de URLs amigables.	X	X	X
Implementación de código HTML			
Uso de plantillas en PHP.	X	X	X
Posibilidad de uso de plantillas en Smarty.	X	X	
Implementación de ayudantes de plantillas (helpers).	X	X	
Seguridad			
Manejo propio de sesiones por usuarios.	X	X	X
Manejo de privilegios de acceso a secciones de la aplicación (Access Control List).	X		X
Verificación de la salida generada en HTML por procesamiento de peticiones (Data Sanitization).	X	X	
Usabilidad y acceso rápido			
Almacenamiento en caché de las vistas.	X	X	
Almacenamiento en caché de configuración de las aplicaciones.		X	
Documentación para su uso			
Manual de referencia.	X	X	X
Documentación de la Interfaz de Programación de Aplicaciones (API).	X	X	X
Herramientas de programación			
Generación de código PHP.	X	X	
Herramientas de prueba y depuración.		X	X
Interfaz de línea de comandos para la creación y mantenimiento de aplicaciones.	X	X	

Almacenamiento de logs de funcionamiento del framework.		X	
Extensibilidad y opciones adicionales			
Integración con otras herramientas a través de plugins.	X	X	
Implementación propia de llamadas Asynchronous JavaScript and XML (AJAX)	X	X	
Soporte para Web Services.			X
Soporte para envío de correo electrónico.		X	X
Generación de contenido sindicalizado (RSS).		X	X
Generación de archivos PDF.	X	X	X
Soporte para internacionalización y localización de contenidos.		X	X
Soporte PHP			
Soporte para PHP4.	X		
Soporte para PHP5.	X	X	X
Características adicionales			
Licencias libres.	X11	X11	BSD
Comunidad activa de usuarios.	X	X	X

Tabla 20. Comparación entre *framework* para PHP.