



**Universidad de las Ciencias Informáticas
Facultad 9**

**Conjunto de funcionalidades de los Subsistemas Web y
Administración de la Programación de la Plataforma de Transmisión
Abierta para Radio y Televisión**

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

Autor:

Olivier García Domínguez

Tutor:

Ing. Yandy León Núñez

“Año 52 de la Revolución”

Declaración de Autoría

Declaro ser Olivier García Domínguez el único autor del presente trabajo de diploma y autorizo al Departamento de Señales Digitales la Facultad #9 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio y reconozco a la misma los derechos patrimoniales de este trabajo, con carácter exclusivo. Para que así conste firmo la presente a los 1 días del mes de julio del año 2010.

Olivier García Domínguez

Ing. Yandy León Núñez

Resumen

En la Universidad de las Ciencias Informáticas se realizan una serie de procesos relacionados con la transmisión de audio y video a toda la comunidad universitaria, todos ellos gestionados por la Dirección de Televisión Universitaria (DTU). Muchos de estos procesos se realizan de forma manual y por separados, por lo que en el Departamento de Señales Digitales de la Facultad 9 se comenzó a realizar la Plataforma de Transmisión Abierta para Radio y Televisión, sistema que unifica dichos procesos. Hoy en día la plataforma está formada por 8 subsistemas como son: Administración de la Transmisión, Transferencia, Reporte, Gestión de Medias, Seguridad, Transmisión, Web y Administración de la Programación, estos dos últimos no cuentan con una serie de funcionalidades que le son de vital importancia para el despliegue en su primera versión, siendo de esta forma necesaria su correcta implementación para la culminación de la plataforma, la cual está basada en herramientas libres.

En el presente trabajo se realizó un análisis profundo de las diferentes técnicas de programación, metodologías, frameworks y plataformas que las soportan. Luego se describieron las funcionalidades necesarias para los Subsistemas Web y Administración de la Programación, separándolas de manera lógica en distintos módulos, teniendo en cuenta la estructura y función que representa. Se describió el estilo de programación empleado en la implementación de cada subsistema y se realizaron las pruebas unitarias de las funcionalidades implementadas.

Palabras Claves

Plataforma, web, televisión, radio, herramientas libres, subsistemas, framework, pruebas unitarias.

ABSTRACT

A number of processes related to the transmission of audio and video to the entire university community are carried out at the University of Information Sciences, all managed by the Department of University Television (DTU). Many of these processes are performed manually and separately, that is why the Department of Digital Signal of our Faculty began to make the Open Platform for Radio Broadcasting and Television, a system that unifies these processes. Today the platform is made up of eight subsystems such as: Administration Transmission, Transfer, Reports, Middle Management, Security, Broadcasting, Web Programming and Administration, the latter two do not have a set of functionalities that are of main importance to the deployment in its original version, thus being necessary the correct implementation for the completion of the platform, which is based on free tools.

In this work we made a deep analysis of the different programming techniques, methodologies, frameworks and platforms that support them. He then described the features needed for Web Subsystems and Administration, Programming, logically separated into several modules, taking into account the structure and function it represents. It described the programming style used in the implementation of each subsystem and performed unit testing functionality implemented.

Introducción.....	1
CAPÍTULO 1	7
Introducción	7
1.1 Subsistemas Web y Administración de la Programación.....	7
1.1.1 Funcionalidades del Subsistema Web.....	7
1.1.2 Funcionalidades del Subsistema de Administración de la Programación.	8
1.2 Tendencias y Tecnologías Actuales	10
1.2.1 Aplicaciones Web.....	11
1.2.2 Arquitectura Cliente-Servidor	12
1.2.3 Ajax.....	14
1.3 Metodología de Desarrollo	16
1.4 Proceso Unificado de Rational (Rational Unified Process RUP).....	17
1.5 Lenguaje Unificado de Modelado (Unified Model Language UML)	19
1.6 Herramientas Case para la Modelación.....	20
1.6.1 Visual Paradigm	20
1.7 Paradigma de la Programación	21
1.7.1 Programación Orientada a Objeto.....	22
1.7.2 Programación Lógica	22
1.7.3 Programación Estructurada.....	23
1.7.4 Programación Funcional.....	23

1.7.5	Criterio de Selección del Paradigma de Programación.....	24
1.8	Lenguajes de Programación.....	24
1.8.1	Lenguajes de Programación del Lado del Cliente	25
1.8.2	Lenguaje de Programación del Lado del Servidor	28
1.9	Framework.....	29
1.9.1	Symfony.....	29
1.10	Patrón Arquitectónico	30
1.11	Entorno Integrado de desarrollo	31
1.11.1	NetBeans IDE 6.8.....	32
1.12	Servidor Web	33
1.12.1	Apache.....	33
1.13	Sistema de Gestión de Base de Datos.....	35
1.13.1	PostgreSQL 8.4.....	35
1.14	Sistema Operativo.....	36
1.13.1	Distribución de Linux	36
1.15	Conclusiones Parciales	38
CAPÍTULO 2		40
Introducción		40
2.1 Descripción de los Subsistemas		40
2.1.1 Funcionalidades del Subsistema Web		40
2.1.2 Funcionalidades del Subsistema Administración de la Programación.....		42
2.2 Modelo de Implementación		43

2.2.1 Diagramas de Componentes	44
2.3 Estilos de Programación	48
2.4 Conclusiones Parciales	52
CAPÍTULO 3	54
Introducción	54
3.1 Automatización de Pruebas	54
3.2 Pruebas Unitarias y Funcionales.....	54
3.2.1 Herramientas Empleadas en las Pruebas.....	56
3.3 Diseño de Pruebas.....	58
3.4 Conclusiones Parciales.....	63
CONCLUSIONES GENERALES	64
RECOMENDACIONES	66
REFERENCIAS BIBLIOGRÁFICAS	67
BIBLIOGRAFÍAS	70
GLOSARIO DE TÉRMINOS	71

ÍNDICE DE FIGURAS

Figura 1. Fases e Iteraciones de la Metodología RUP.....	17
Figura 2. Representación del patrón arquitectónico Modelo Vista Controlador.....	31
Figura 3. Logo del Servidor Apache.	34
Figura 4. Logo de PostgreSQL.....	35
Figura 5. Modelo de Implementación de la PTARTV.....	43
Figura 6. Estructura del Subsistema Web.....	44
Figura 7. Estructura Subsistema Administración de la Programación.....	44
Figura 8. Diagrama de componentes de caso de uso Gestionar Espacio Televisivo.	45
Figura 9. Diagrama de componentes de caso de uso Gestionar Espacio de Radio.....	45
Figura 10. Diagrama de componentes de caso de uso Gestionar Programación de Televisión.....	45
Figura 11. Diagrama de componentes de caso de uso Gestionar Programación de Radio.....	46
Figura 12. Diagrama de componentes del módulo Encuesta.....	46
Figura 13. Diagrama de componentes del módulo Cartelera.....	46
Figura 14. Diagrama de componentes del módulo Estadísticas.....	47
Figura 15. Logo de phpBB.....	48
Figura 16. Ejemplo de Indentación.....	49
Figura 17. Ejemplo de Saltos de Líneas.....	50
Figura 18. Ejemplo de Líneas y Espacios en Blanco.....	50
Figura 19. Fragmento de caso de prueba desarrollado con Selenium.....	56
Figura 20. Fragmento del TestSuite.html.....	57

Figura 21. Elementos del Selenium IDE.	57
Figura 22. Resultado de Caso de Prueba # 1.	58
Figura 23. Resultado de Caso de Prueba # 2.	59
Figura 24. Resultado de Caso de Prueba # 3.	60
Figura 25. Resultado de Caso de Prueba # 4.	61
Figura 26. Resultado de Caso de Prueba # 5.	61
Figura 27. Resultado de Caso de Prueba # 6.	62

ÍNDICE DE TABLAS

Tabla 1. Caso de Prueba # 1.	58
Tabla 2. Caso de Prueba # 2.	59
Tabla 3. Caso de Prueba # 3.	60
Tabla 4. Caso de Prueba # 4.	60
Tabla 5. Caso de Prueba # 5.	61
Tabla 6. Caso de Prueba # 6.	62
Tabla 7. Resultado del las Pruebas de cada Caso de Uso.	63

Introducción

Ha transcurrido un largo camino desde el inicio de la televisión (TV) hasta la actualidad y durante este proceso han ocurrido muchos avances en su desarrollo. Sus inicios en la prehistoria se emprenden desde finales del siglo XIX hasta 1935, llegándose a convertir en el avance tecnológico más importante de ese siglo. Su nacimiento parte de la unión de una serie de investigaciones y fenómenos en conjunto, pero desarrolladas de forma individual para lograr convertirse en una mezcla de audio y un conjunto de imágenes en movimiento con ausencia de otros colores que no fueran el blanco y el negro. Este proceso permitió que se lograra convertir un invento de laboratorio en un medio al alcance del ser humano. (1)

Fue en julio de 1928 donde a partir de la estación experimental W3XK en Washington, se lograron las primeras transmisiones experimentales, donde transmitían imágenes exploradas principalmente de películas con cierta regularidad. Un año más tarde en 1929 la British Broadcasting Corporation (BBC) de Londres comienza a interesarse por las investigaciones de Jonh Logie Baird, quien en 1928 había logrado grandes avances acerca de la transmisión de imágenes desde Londres hasta New York, además de haber demostrado la TV en color, la exterior con luz natural y la estéreo, pero todas estas de manera muy primitiva (2). Es por este motivo que la BBC en este mismo año comenzó a realizar una transmisión de imágenes de forma regular, pero con cierto desgano hasta llegar al año 1936 donde Gran Bretaña inauguró el primer ciclo de emisiones regulares de televisión. La Segunda Guerra mundial impidió continuar con el desarrollo que estaba alcanzando la TV a nivel mundial, ya que comenzó a concentrar todo el esfuerzo tecnológico en el conflicto militar. Una vez terminado el conflicto, la industria de la TV tomó un nuevo ímpetu y comenzó a tomar el gran protagonismo que había logrado durante los años anteriores (3).

Terminada la etapa de nacimiento y consolidación, llegaría la época del calor y, tras ella, la internacionalización de la televisión y sus medios, permitiendo que se extendiera a través de los distintos continentes hasta llegar a las Américas y comenzar a desarrollarse en la hermosa nación: Cuba. Fue la Unión Radio Televisión quien el 24 de octubre de 1950 transmitió la primera señal de televisión comercial por el canal 4, siendo el tercer país del hemisferio en introducir este medio. Al pasar 7 años de experiencia es decir, 1957, ya alrededor de 50 países tenían televisión y un año más tarde el país ya contaba con 25 transmisores de televisión instalados en distintas provincias.

El triunfo de la Revolución ha permitido que Cuba se interese por el desarrollo de la sociedad para multiplicar los conocimientos de la nueva generación. Este motivo logró que las nuevas Tecnologías de la Informática y las Comunicaciones (TICs), comenzaran a formar parte del proceso educacional para

desarrollar de forma masiva una cultura general integral, permitiendo la utilización de medios audiovisuales en las escuelas cubanas, lo cual produjo la necesidad de crear programas de televisión destinados a las escuelas y otros sectores de la sociedad. Continuar con un prolongado desarrollo de las TIC es un objetivo fundamental del gobierno y para esto cuenta con recursos humanos bien preparados, para la creación de distintos proyectos que responden a ese propósito. Fue así como surgió dentro del programa de la batalla de ideas, la Universidad de Ciencias Informáticas (UCI) en septiembre del 2002, la cual tiene como objetivo formar profesionales comprometidos con la patria cubana y calificados en la rama de la informática, los cuales permitirán el desarrollo de la producción de software.

Con el surgimiento de la UCI, dentro de ella se dio origen al nacimiento de canales televisivos orientados fundamentalmente al desarrollo de la enseñanza y el aprendizaje de sus estudiantes, no siendo éstos solamente empleados para fines educativos, sino también para el entretenimiento de toda la comunidad universitaria. Estos canales transmiten distintos tipos de materiales audiovisuales en diferentes horarios, ejemplo de estos son: series, películas, documentales, musicales e informativos. Muy relacionado al mundo de la televisión se creó un proyecto orientado a servicios televisivos tomando el nombre de UCITeVe, el cual en la actualidad pertenece al Centro de Geoinformática y Señales Digitales, teniendo dentro de sus objetivos, la automatización de procesos que resuelvan distintos problemas que presenta la Dirección de Televisión Universitaria (DTU¹). La universidad mediante la red interna permite que los usuarios puedan acceder a un portal web quien se encarga de hospedar todo el contenido multimedia, sin embargo, dicho portal está confeccionado con herramientas propietarias, mientras que el país se encuentra inmerso en la migración hacia el software libre, por lo que es necesario confeccionar aplicaciones con herramientas libres. Existen innumerables ventajas que el software no propietario brinda, entre las más conocidas se encuentran: la libertad de uso y distribución, la independencia tecnológica y la cantidad de aplicaciones que posee, por consiguiente, si se realiza una migración se podrían desarrollar productos con un gran número de funcionalidades, en aras de mejorar la eficiencia y funcionamiento de los servicios que se ofrecen, además no habría que pagar costosas licencias. Todo esto trae como consecuencia que los directivos del Centro de Geoinformática y Señales digitales comenzaran con la creación de herramientas propias bajo estándares libres.

Dentro del Departamento de Señales Digitales surge la idea de realizar una plataforma con el objetivo de integrar en una única solución, los procesos televisivos que se encuentran actualmente dispersos en la universidad, la misma llevaría el nombre de Plataforma de Transmisión Abierta para Radio y Televisión

¹ **DTU:** Entidad encargada de la gestión de los procesos de contenido multimedia transmitidos en la comunidad universitaria de la UCI.

(PTARTV). De esa manera se lograría conformar una solución factible para la DTU, de forma que se pueda informatizar los distintos procesos que aún se realizan de forma manual.

Esta entidad está formada por diferentes departamentos. Dentro de estos se encuentra el encargado de la programación de los canales, y en él se realizan las tareas correspondientes de una manera informativa y manual, debido a que la estructura de la programación es realizada por una persona haciendo uso de un documento en formato de hoja de cálculo. Todo esto conlleva en varias ocasiones a la existencia de errores humanos que han traído consigo la retransmisión de materiales audiovisuales en un período relativamente breve, producto al desconocimiento de lo que se proyectó en ocasiones anteriores y luego surge la insatisfacción por parte de los televidentes. La televisión cubana cuenta con una gran variedad de espacios televisivos, así como la radio nacional con sus espacios radiales, los cuales no son incluidos en la programación que se transmite mediante los canales internos de la UCI, provocando que la comunidad universitaria no pueda visualizar dichos espacios en vivos durante su transmisión.

Los directivos de la DTU no cuentan con un servicio que le permita conocer las inquietudes, necesidad y la opinión de la comunidad universitaria respecto a los distintos programas que son transmitidos por los canales de la UCI, para luego poder mejorar el servicio acorde a lo que requiere el cliente, de la misma manera que tampoco les ofrecen a los televidentes toda la información necesaria de los programas que se transmitirán. Es por este motivo que la PTARTV pretende resolver el problema de la gestión de contenido multimedia hacia el servidor de publicación, facilitando su visualización, la interacción del usuario y a la vez una comodidad al administrador de la programación para la realización de su trabajo, así como integrar todos los servicios televisivos y de radio en un sistema que facilite su gestión. La PTARTV realiza un proceso para la gestión de medias hacia el servidor de publicaciones, donde se crea una ficha técnica para cada uno de los materiales audiovisuales, los cuales son recibidos en distintos formatos como: AVI², MPEG-1³ y MPEG-2⁴; luego se le realiza la conversión al formato único que utiliza la plataforma, de manera tal que se encuentre listo para que el administrador de la programación pueda utilizar estos materiales para la planificación televisiva o radial que se le enviará al transmisor; quien seguidamente comenzará a transmitir por los canales establecidos. La plataforma está compuesta por 8 subsistemas, dentro de estos se encuentra el Subsistema Web y el de Administración de la Programación, los cuales se encuentran en desarrollo, pues existe un conjunto de funcionalidades que no se encuentran implementadas como lo es casi todo el trabajo que realizará el administrador de la

² **AVI:** (*Audio Video Interleave*), en español intercalado de audio y video, es un formato de archivo contenedor de audio y vídeo.

³ **MPEG – 1:** nombre de un grupo de estándares de codificación de audio y vídeo normalizados por el grupo *MPEG (Moving Pictures Experts Group)*.

⁴ **MPEG – 2:** similar a *MPEG-1*, pero también proporciona soporte para vídeo entrelazado (formato utilizado por las televisiones).

programación y otros servicios que ofrecerá el portal web. Es por ello que resulta de vital importancia su culminación ya que permitirá obtener un producto final mucho más estable, potente, seguro, con más prestaciones de servicios y con una interfaz gráfica mucho más amigable al cliente.

Todo lo antes mencionado demuestra una situación problemática que define el siguiente **problema científico**: Ausencia de un conjunto de funcionalidades que manejen la publicación y visualización de contenido multimedia así como la administración de la programación de la Plataforma de Transmisión Abierta para Radio y Televisión.

Toda la información antes brindada permite proponer como **objetivo general**: Implementar un conjunto de funcionalidades en los Subsistemas Web y Administración de la Programación de la Plataforma de Transmisión Abierta para Radio y Televisión.

Una vez definido el objetivo central del presente trabajo se plantea como **objeto de estudio**: Los procesos de publicación de contenidos multimedia en la Web y la Administración de la Programación de la PTARTV y como **campo de acción** se encuentra la informatización de los procesos presentes en los Subsistemas Web y Administración de la Programación.

Por lo anteriormente expresado, con la implementación de un sistema automatizado que permita la administración de la programación y visualización de contenido multimedia en la PTARTV, se produciría una mejora de dichos procesos y un aumento de la productividad, quedando de esta forma establecida la **idea a defender**.

Para dar cumplimiento a la investigación científica, se definieron un conjunto de **tareas** las cuales permitirán darle cumplimiento al objetivo planteado anteriormente:

1. Caracterizar tendencias y tecnologías actuales.
2. Definir entornos de desarrollo integrado y lenguajes de programación.
3. Caracterizar las funcionalidades en el Subsistema Web y Administración de la Programación.
4. Caracterizar las herramientas para el trabajo con la administración de la programación en software libre.
5. Construir el modelo de implementación.
6. Implementar los Subsistemas Web y Administración de la Programación.
7. Desarrollar los casos de prueba que certifiquen la veracidad de los algoritmos empleados.
8. Documentar los resultados de las pruebas unitarias y funcionales de los subsistemas.

Para lograr obtener los conocimientos necesarios que hagan posible el cumplimiento del objetivo propuesto en el trabajo, se lleva a cabo una investigación en la que se utilizan algunos de los métodos científicos existentes, tanto teóricos como empíricos.

- Analítico-Sintético: son dos procesos cognoscitivos que permiten dividir la investigación en partes para su desarrollo, logrando luego la unión a través de las relaciones existentes y se utiliza para la extracción de los elementos más importantes de los procesos a implementar en la PTARTV.
- Histórico-Lógico: estudia la trayectoria real de los fenómenos y acontecimientos en el de cursar de su historia por lo que se utiliza para caracterizar soluciones existentes relacionada a los subsistemas de implementación a desarrollar, entendiendo su evolución y conceptos teóricos que sean de relevancia.
- Modelación: permite una reproducción simplificada del fenómeno que se está estudiando y se utiliza para crear abstracciones de los distintos procesos a desarrollar y así poder explicar la realidad de éstos. Se hace realidad en el trabajo al crear modelos como el de implementación.
- Observación: es la percepción directa del objeto de investigación y se utiliza para conocer de una manera mucho más fácil el panorama real relacionado con las distintas funcionalidades a implementar mediante la percepción directa.

Capítulo I

Fundamentación Teórica

CAPÍTULO 1

Fundamentación Teórica

Introducción

En el presente capítulo se realiza un análisis profundo acerca de la implementación de los Subsistemas Web y Administración de la Programación de la Plataforma de Transmisión Abierta de Radio y Televisión, cuyas bases se encuentran fundamentadas por la previa realización de su análisis y diseño. Serán descritos los procesos relacionados al conjunto de funcionalidades necesarias para la culminación del Subsistema Web y el de Administración de la Programación de la PTARTV, así como las distintas técnicas de programación, metodologías y frameworks utilizados.

1.1 Subsistemas Web y Administración de la Programación.

1.1.1 Funcionalidades del Subsistema Web.

En el presente epígrafe se pretende explicar las distintas funcionalidades que se incorporarán al Subsistema Web y al de Administración de la Programación para su correcto desarrollo, logrando de esta manera brindar diversos servicios al cliente como:

Visualización de la Cartelera: este servicio posibilitará que el cliente pueda observar la cartelera de la programación que brindará la plataforma, no solo la de la fecha actual sino la de la semana en que se encuentra, posibilitándole planificar mejor su tiempo y seleccionar los materiales que podrá ver.

Visualizar Estadísticas: este servicio permite que el cliente y el administrador puedan visualizar la cantidad de veces que un determinado programa ha sido visto por los usuarios, dando una idea general de cuáles son los programas o materiales más gustados entre la comunidad.

Adicionar Encuesta: este servicio permitirá que el administrador del sistema publique temas de encuesta necesarios para conocer la opinión de los clientes.

Modificar Encuesta: este servicio le facilitará al administrador del sistema, poder modificar cualquiera de los temas de encuesta creados con anterioridad.

Eliminar Encuesta: este servicio posibilita que el administrador pueda eliminar un determinado tema de encuesta publicado en la plataforma.

Buscar Encuesta: este servicio le ofrecerá la posibilidad al cliente de poder buscar una determinada encuesta que se encuentre publicada en la plataforma, por esta vía podrá acceder a la misma de una manera mucho más rápida.

Adicionar Tema de Foro: este servicio permitirá publicar un tema de foro dentro de la plataforma, para que se comience a realizar un intercambio de información relacionado al tema en cuestión

Modificar Tema de Foro: este servicio le permitirá al administrador poder modificar los temas del foro que se publiquen en la plataforma.

Eliminar Tema de Foro: este servicio le posibilitará al administrador eliminar un tema de foro que se encuentre almacenado en la plataforma.

Una vez descritas las funcionalidades del Subsistema Web, es necesario realizar una explicación de las funcionalidades que debe realizar el Subsistema Administración de la Programación, encargado de establecer los horarios de los materiales que se transmitirán por los diferentes canales de la plataforma.

1.1.2 Funcionalidades del Subsistema de Administración de la Programación.

La programación que se brindará en la PTARTV es realizada por el administrador de la transmisión, quien es el encargado de organizar todas las medias que se visualizarán mediante la plataforma. Muchos de estos procesos son muy tediosos y no se encuentran automatizados. A continuación se hace una descripción general de cada funcionalidad agregada al Subsistema de Administración de la Programación.

Adicionar Programación para la Radio: posibilitará al administrador poder adicionar una nueva programación de radio a la plataforma para que los usuarios puedan escucharla. En este proceso se seleccionará el canal por el cual se realizará la futura transmisión, una fecha así como el horario que tendrá definido y después se agregarán los materiales de audio que formarán parte de esta programación, definiendo para cada uno de ellos la hora en que comenzarán.

Modificar Programación para la Radio: permitirá que el administrador pueda modificar una determinada programación de radio que se encuentre almacenada o publicada en la plataforma, ya sea eliminando algún material de audio publicado, moviéndolo de horario o agregando nuevas medias a dicha programación.

Eliminar Programación para la Radio: brindará al administrador la opción de eliminar una programación determinada que se encuentre publicada en la plataforma.

Buscar Programación para la Radio: ofrecerá al usuario la opción de buscar una programación en la plataforma. Esta función es utilizada por las tres funcionalidades descritas anteriormente debido a que de ella depende el correcto funcionamiento de las mismas.

Las programaciones para la televisión (TV) tienen un comportamiento muy similar a las de radios analizadas con anterioridad, la diferencia radica en el formato de los materiales que son publicados. En el caso de las programaciones de radio, solamente se programan medias que contengan audio mientras que en las de televisión se tienen en cuenta los materiales que contienen audio y video.

Adicionar Programación de Televisión: este servicio posibilitará que el administrador adicione una nueva programación de televisión a la plataforma para que esta pueda ser utilizada por el transmisor y emitido por el canal seleccionado. El proceso ocurre igual que para la programación de radio.

Modificar Programación de Televisión: permitirá que el administrador pueda modificar una determinada programación de televisión que se encuentre almacenada o publicada en la plataforma.

Eliminar Programación de Televisión: posibilitará al administrador eliminar una determinada programación de televisión publicada o almacenada en la plataforma.

Buscar Programación de Televisión: este servicio permite que el usuario pueda localizar una programación de televisión en específico de una manera más rápida.

Un espacio de radio o de televisión, es un momento reservado dentro de una programación que no puede ser alterado o sobrescrito. Ejemplo de ello lo son en el caso de la TV cubana: el noticiero, las aventuras o las novelas, que tienen un horario preestablecido e inalterable.

Adicionar Espacio de Radio: esta funcionalidad permitirá que el administrador pueda adicionar un espacio de radio a la plataforma para ser publicado. Cada espacio contiene los canales por los que se transmitirá, el horario de inicio y fin y los días de la semana que saldrá al aire.

Eliminar Espacio de Radio: este servicio le posibilitará al administrador eliminar un espacio de radio que se encuentre publicado o almacenado en la plataforma.

Modificar Espacio de Radio: brinda la posibilidad de modificar un determinado espacio de radio al administrador, cambiando los parámetros que tiene establecidos.

Buscar Espacio de Radio: ofrecerá al cliente la posibilidad de buscar un espacio de radio en específico que se encuentre publicado en la plataforma y de esa manera acceder mucho más rápido a ese espacio.

Adicionar Espacio de Televisión: este servicio permite que el administrador del sistema pueda agregar un nuevo espacio de televisión para luego poder incluirlo a la programación de un canal.

Eliminar Espacio de Televisión: esta funcionalidad le brinda la posibilidad de eliminar un espacio televisivo que se encuentre en la base dato, una vez que ya no se utilice en alguna programación.

Modificar Espacio de Televisión: este servicio le ofrece la posibilidad al administrador del sistema de modificar un determinado espacio televisivo.

Buscar Espacio Televisivo: posibilita que se pueda buscar de manera directa un determinado espacio televisivo.

Una vez explicado el objetivo de las funcionalidades que se necesitan adicionar a los subsistemas descritos, se hace necesaria una correcta selección de las tendencias y tecnologías que se utilizarán para la implementación.

1.2 Tendencias y Tecnologías Actuales

Sin duda alguna en la informatización a nivel mundial se exponen elementos que permiten un mayor acercamiento a las tendencias actuales más relevantes en el desarrollo de la televisión digital, donde sus medios tradicionales están convergiendo con los nuevos y cada vez son más las personas que siguen la televisión desde sus ordenadores. Este motivo permite aumentar el desarrollo de aplicaciones web, direccionadas a la creación de sistemas que manejen un mejor procesamiento de la radio y la televisión.

Para una correcta implementación de los Subsistemas Web y Administración de la Programación de la PTARTV es de suma importancia realizar una correcta selección de las nuevas tendencias y tecnologías, de esta forma, se logrará obtener un mejor resultado del producto final, por lo que a continuación se

comenzará caracterizando a las aplicaciones web, siendo el estilo definido para la plataforma en su etapa de análisis y diseño.

1.2.1 Aplicaciones Web

La WWW o World Wide Web, es decir, web, se ha convertido, con el pasar de los tiempos en el principal servicio de internet, dejando de ser una amplia “biblioteca” de páginas con información estática, para convertirse en un servicio que permite acceder a muchas prestaciones, funciones e infinidad de servicios.

Tim Bernés-Lee en el año 1989 trabajaba para la CERN (Centro Europeo de Investigación Nuclear) y tuvo la necesidad de comenzar a elaborar una plataforma que permitiera un fácil acceso a la documentación de la CERN. Fue aquí donde comenzó el surgimiento de la Web (4), quien en la actualidad tiene varias definiciones sobre dichas aplicaciones, por lo que no es posible tener una correcta definición que agrupe todos los procesos que la conforman.

Una aplicación web es un software que se ejecuta en un servidor web, el cual sirve para responder a las diferentes peticiones de los navegadores, proporcionando los recursos que sean solicitados usando el protocolo HTTP (Hyper Text Transfer Protocol) (en español Protocolo de Transferencia de Hipertexto) quien define la sintaxis y la semántica que utilizan la arquitectura web (“Cliente-Servidor”) para comunicarse. Es empleado HTML (Hyper Text Markup Language) (en español Lenguaje de Marcado de Hipertexto) como lenguaje de marcado predominante para su construcción y URI (Universal Resource Identifiers) (en español Identificador Uniforme de Recurso) quien permite definir una sintaxis para codificar el nombre o dirección a la hora de establecer un vínculo (5).

Las aplicaciones web se han popularizado debido a lo práctico del navegador web como cliente ligero, así como a la facilidad para actualizar y mantenerlas sin distribuir e instalar software a miles de usuarios y más aún, con el aumento del uso del internet quien le ha permitido un mayor desarrollo. Los usuarios para poder utilizarlas solo tienen que acceder a un servidor mediante internet o una intranet.

El desarrollo de las aplicaciones web ha crecido producto al gran número de ventajas que presenta, entre ellas se mencionan algunas como son:

- **Compatibilidad Multiplataforma:** diversas plataformas tanto hardware como software, permiten las ejecuciones de aplicaciones web, solo con el uso de un software y un ejecutable.
- **Actualización:** producto de que los usuarios de la aplicación utilizan un solo programa que se relaciona con un servidor, los usuarios siempre utilizarán la versión más actualizada del sistema.

- **Inmediatez de acceso:** los usuarios para utilizar el servicio que brinda la aplicación no necesitan descargarlo, instalarlo o configurarlo, solo acceder y beneficiarse de las prestaciones de la aplicación web.
- **Dócil de Integrar con otros sistemas:** producto a que se basa en protocolos estándares, la información manejada por el sistema puede ser accedida con mayor facilidad por otros sistemas.
- **Múltiples usuarios concurrentes:** estas aplicaciones permiten que múltiples usuarios puedan utilizarla al mismo tiempo.
- **Acceso cambiable:** el usuario puede acceder a la aplicación desde cualquier parte que tenga acceso a la red privada de la organización o internet y a la vez que tenga los permisos de acceso necesarios según las políticas de dicha organización. (6)

Estas ventajas le proporcionan a la plataforma mecanismos de comunicación estándares entre los diferentes módulos dentro de ella, que interactúan entre sí para presentar información dinámica al usuario. Además, para proporcionar interoperabilidad y extensibilidad entre estos módulos, para que al mismo tiempo sea posible su combinación y poder realizar operaciones complejas.

Para el desarrollo de toda aplicación web, es necesario definir una arquitectura correcta, la cual permita un intercambio de información entre un cliente y un servidor, por lo que el siguiente epígrafe argumenta la arquitectura cliente servidor utilizada para la implementación de ambos subsistemas.

1.2.2 Arquitectura Cliente-Servidor

El mundo de las comunicaciones entre computadoras usando aplicaciones web se rige fundamentalmente por lo que se llama modelo "Cliente/Servidor", quien intenta desarrollar una mayor usabilidad, interoperabilidad, escalabilidad y flexibilidad entre las comunicaciones. En estos momentos existen una gran cantidad de definiciones acerca de este modelo, pero la empresa International Business Machines (IBM) la define como: "Es tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicios hechos por estaciones de trabajo inteligentes o "clientes", resultan en un trabajo realizado por otros computadores llamados servidores (7).

Entre las ventajas que ofrece esta tecnología para el desarrollo del trabajo figuran que es una infraestructura versátil, modular y basada en mensajes que mejora la portabilidad, la interoperabilidad y la escalabilidad de los sistemas. Modificar la infraestructura puede resultar sencillo pues integrar más

ordenadores o sistemas, no constituye un problema. Con su uso se produce un aumento de la interactividad y se facilita el mantenimiento de las aplicaciones.

Principales características de la arquitectura “Cliente/Servidor”

- El servidor y el cliente pueden actuar como una sola entidad, o como entidades separadas, realizando tareas o actividades independientes.
- Tanto el cliente como el servidor pueden realizar funciones en distintas plataformas o en la misma.
- Un servidor presta servicios a varios clientes en forma concurrente.
- La interrelación entre el hardware y el software esta basado en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.
- Si se realizaran modificaciones en el servidor, estas producirían poca o ninguna afectación al cliente (7).

Esta arquitectura se divide en dos aspectos fundamentales:

- El **servidor** quien es la máquina a la cual se le introducen diversos servicios y se encuentra a la espera de las peticiones del cliente y una vez realizada, busca la información solicitada y le envía la respuesta.
- El **cliente** es quien realiza una petición al servidor, donde durante este proceso se efectúan funciones de software basándose en el hardware, pero de no tener la capacidad de procesar los datos necesarios, recurre al servidor y espera a que este le envíe la información solicitada (8).

La tecnología denominada **cliente/servidor**, es utilizada por muchas aplicaciones dentro de una red. Por esta razón y teniendo en cuenta que la Plataforma de Transmisión Abierta para Radio y Televisión en sus inicios prestará sus servicios dentro de una intranet, ya que un grupo de usuarios podrá acceder simultáneamente a este recurso, solicitando diversos servicios mientras que desde un servidor central se le podrá ofrecer respuesta a los mismos, se justifica la utilización de esta tecnología para el desarrollo del sistema. Con su uso se produce un incremento de la interactividad y se facilita el mantenimiento de las aplicaciones. Además, que implicaría un mejor manejo de errores e implementación de estrategias para garantizar la integridad de los datos de una aplicación, si se quisiera modificar la infraestructura del sistema podría resultar sencilla, pues integrar más ordenadores o sistemas externos, no constituiría un problema.

Para la correcta implementación de los subsistemas se hizo necesario vincular la arquitectura cliente servidor y la aplicación web con un conjunto de tecnologías denominada AJAX, donde en el siguiente epígrafe se hace alusión a sus principales características.

1.2.3 Ajax

Asociada a la tecnología web y cliente-servidor aparece el modelo de presentación de datos *AJAX* (*JavaScript* Asíncrono y *XML*⁵) que ofrece comodidad visual para el usuario en término de interacción con los datos.

Según Jesse James Garrett en el artículo "Ajax: A New Approach to Web Applications", publicado por primera vez el 15 de febrero del 2005, "Ajax no es una tecnología, sino el conjunto de varias tecnologías que unidas pueden lograr cosas realmente maravillosas lo cual lo convierten en algo mucho más interesante" (9).

¿Por qué AJAX?

La simplicidad de las aplicaciones le permite que estas aumenten en su desarrollo, aunque ofrecen una menor interactividad y usabilidad en comparación con las aplicaciones de escritorio, producto de que la interacción del usuario con las aplicaciones web se interrumpe cada vez que se realiza una petición al servidor, pero sus principales características permiten crear aplicaciones con un sin número de ventajas con respecto a las aplicaciones web basadas completamente en servidor ya que ofrecen:

- Una mayor eficiencia, ya que en el explorador de una página web se realizan sus principales procesos.
- Existe una gran compatibilidad con todos los exploradores más populares y utilizados por los usuarios, como son el Microsoft Internet Explorer, Mozilla Firefox y Apple Safari.
- La interfaz de usuarios posee una gran familiaridad con sus elementos, como indicadores de progreso, información sobre herramientas y ventanas emergentes.
- Se realizan actualizaciones parciales de la página, las cuales actualizan sólo las partes de la página web que han cambiado (10).

Ajax le permite al usuario lograr una mejora completamente en su relación con la aplicación, ya que elimina las recargas continuas de la página, producto a que se utiliza un segundo plano para el intercambio de información con el servidor.

⁵ **XML**: lenguaje de marcas extensible (siglas en inglés eXtensible Markup Language).

Las Tecnologías de AJAX incorporan:

- Estructura y presentación de la información basada en estándares mediante HTML o XHTML⁶ y Hojas de Estilo en Cascada (CSS) (en inglés Cascading Style Sheets).
- Document Object Model (DOM) (una traducción al español no literal, pero apropiada, podría ser Modelo de Objetos del Documento) para interactuar dinámicamente con los datos.
- Uso de XML⁷ y XSLT⁸ para el intercambio y manipulación de datos.

Navegadores que no permiten AJAX:

- Opera 7 y anteriores.
- Microsoft Internet Explorer para Windows versión 4.0 y anteriores.
- Microsoft Internet Explorer para Macintosh, todas las versiones.
- Navegadores basados en texto como Lynx y Links.
- Navegadores para incapacitados visuales (braille).

Las aplicaciones AJAX eliminan las paradas en la ejecución, donde interfiere un intermediario (el motor AJAX) entre el usuario y el servidor. En vez de cargar al iniciar sesión una página web, lo que hace el navegador es cargar el motor AJAX (escrito en JavaScript y normalmente escondido dentro de un marco oculto). La introducción de una capa a la aplicación le produce un gran número de ventajas la cual le provoca ser más versátil.

Lograr un sin número de desafíos en el proceso de la creación de aplicaciones Ajax, no está dado por las técnicas para su desarrollo, ya que las tecnologías son maduras, estables y bien conocidas, son los diseñadores de estas aplicaciones quienes tienen el peso fundamental para lograr diversos desafíos: olvidar lo que creemos saber sobre las limitaciones de la web, y comenzar a imaginar un rango más amplio y rico de posibilidades logrará ser más divertido (11).

Diferencias de Ajax con las aplicaciones Web tradicionales:

En las aplicaciones web tradicionales los usuarios interactúan mediante formularios, que al ser enviados, realizan una petición al servidor web, donde luego este servidor se comporta según lo enviado y

⁶ **XHTML**: Lenguaje extensible de marcado de hipertexto (acrónimo en inglés de eXtensible Hypertext Markup Language).

⁷ **XML**: Lenguaje de marcas extensible (siglas en inglés de eXtensible Markup Language).

⁸ **XSTL**: Extensible hoja de estilo en transformaciones del lenguaje (siglas en inglés eXtensible Style Sheet Language Transformations).

responde al mensaje realizando un envío de una nueva página web. Esto desperdicia mucho ancho de banda, ya que gran parte del HTML enviado en la segunda página web, ya se encontraba presente en la primera. De esta manera, no es posible desarrollar aplicaciones con un grado de interacción similar al de las aplicaciones de escritorio.

El desarrollo de aplicaciones con Ajax permite realizar peticiones al servidor web para obtener solamente la información que se desea, mediante la utilización de algún lenguaje para servicios web basados en XML, y uso de JavaScript por parte del cliente para procesar la respuesta del servidor web. Esto produce una mayor interacción producto a la reducción de información intercambiada entre el cliente y el servidor y a que parte del proceso de la información lo realiza el propio cliente a través del JavaScript, librando al servidor de esta labor. Todo lo contrario es que la descarga inicial de la página es más lento al tener que bajar todo el código JavaScript (12).

Una vez seleccionado un conjunto de tendencias y tecnologías para el desarrollo del presente trabajo se hace necesario el correcto uso de una determinada metodología de desarrollo.

1.3 Metodología de Desarrollo

El proceso de desarrollo de software no es una tarea sencilla. Muchos proyectos de los que se desarrollan tarde o temprano caen en problemas pues se salen del presupuesto, tienen importantes retrasos, o simplemente no cumplen con las expectativas del cliente. Como solución a esta problemática surge la metodología, la cual se puede definir como un conjunto de actividades encaminadas a lograr un fin específico. Las metodologías imponen un trabajo disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente.

Las metodologías de desarrollo de software conforman un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos de software. En ellas se van indicando paso a paso todas las actividades a realizar para lograr un buen producto informático, indicando además qué personas deben participar en el desarrollo de dichas actividades y el rol que deben ocupar cada una de ellas. Igualmente definen la información que se debe producir como resultado de una actividad y la necesaria para comenzarla (13).

En la actualidad existen un gran número de propuestas que definen este marco de trabajo, se habla de metodologías tradicionales o robustas como Proceso Unificado de Rational (RUP) (siglas en inglés Rational Unified Process), Microsoft Solutions Framework (MSF) y Métrica 3.0 o metodologías ágiles como Extreme Programming (XP) (siglas en español Programación Extrema), Scrum, Cristal Methods y

Feature Driven Development, pero las experiencias indican que no existe una receta mágica y no se trata simplemente de seleccionar una de las propuestas y seguirla. Se hace necesario entender los procesos de desarrollo, tener pleno dominio de factores como el tiempo con el que se cuenta, presupuesto y sobre todo cual metodología se ajusta más al entorno y a lo que se desea desarrollar.

Durante la investigación de este presente trabajo se estableció con anterioridad que la metodología correcta para realizar ambos subsistemas es el Proceso Unificado de Rational (RUP). A continuación se describe las características que las distingue de otras metodologías existentes.

1.4 Proceso Unificado de Rational (*Rational Unified Process RUP*)

El Proceso Unificado de Desarrollo (RUP) es un marco de trabajo genérico que se especializa en una gran variedad de sistemas de software, en distintas áreas de aplicación, tipos de organizaciones, niveles de aptitud y tamaños de proyecto (14).

RUP es un proceso, que en su modelación define como principales elementos: trabajadores que conforman el (quién); actividades que representan el (cómo); artefactos que son los productos tangibles del proyecto que concretan el (qué); y el flujo de actividades que precisan la secuencia de actividades (cuándo). Además, está compuesto por nueve flujos de trabajo y cuatro fases, cada una de ellas con un número variable de iteraciones como se muestra en la figura 1 a continuación.

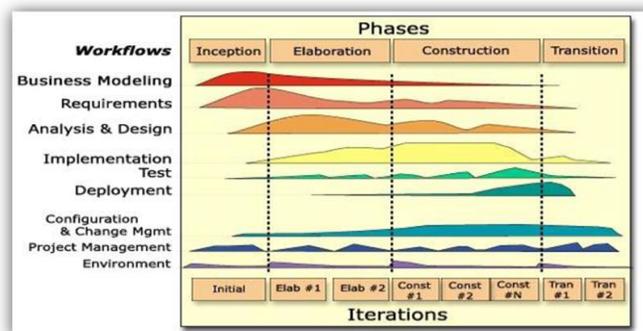


Figura 1. Fases e Iteraciones de la Metodología RUP.

RUP en el flujo de trabajo de implementación tiene los siguientes objetivos:

- Definir la organización del código en términos de subsistemas de implementación organizados en capas.
- Implementar los elementos de diseño en términos de elementos de implementación (archivos fuentes, binarios y ejecutables).
- Probar los componentes desarrollados independientemente como unidades.

- Integrar los resultados producidos por desarrolladores independientes o equipos en un sistema ejecutable.

Los artefactos y las actividades que se generan para los diferentes roles que ocupan las personas durante cada una de las fases en un proceso de desarrollo de software. Se realizará un énfasis en las actividades que son responsabilidad del rol de implementador durante el flujo de trabajo de implementación, las cuales son:

- Implementar elementos de diseño.
- Implementar los subsistemas de implementación.
- Ejecutar pruebas de desarrollador.
- Implementar las pruebas de desarrollador.
- Corregir la Implementación (15).

Para el rol de implementador el proceso unificado de desarrollo de software genera un grupo de artefactos como son:

- Modelo de implementación.
- Subsistema de implementación.
- Prueba de desarrollador.

A continuación se describen cada uno de los artefactos mencionados.

Modelo de implementación: representa la composición física de la implementación en términos de subsistemas de implementación, y elementos de implementación (directorios y archivos, incluyendo código fuente, datos y archivos ejecutables). En él están representados ambos subsistemas en una vista global con todos los componentes, relaciones, librerías y código fuente.

Subsistema de implementación: este artefacto consta de un conjunto de elementos de implementación. Estructura el modelo de implementación dividiéndolo en componentes más pequeños que se pueden integrar y probar de manera independiente. Los subsistemas de implementación se definen inicialmente en la vista de implementación y, por lo tanto, son de vital importancia ya que posibilita desglosar todo el sistema en partes más pequeñas e independientes para que su desarrollo sea menos complejo.

Prueba de desarrollador: proporciona a la implementación un subconjunto de pruebas necesarias, efectivas y eficaces, pensadas bajo las categorías siguientes: pruebas de unidad, parte de las pruebas de integración, y algunos aspectos de lo que se denomina pruebas del sistema. Esto posibilita a los

desarrolladores comprobar que el componente funciona correctamente antes de que se realicen más pruebas formales en él (16).

RUP guía a los equipos de proyecto en cómo administrar el desarrollo iterativo de un modo controlado mientras se balancean los requerimientos del negocio, el tiempo al mercado y los riesgos del proyecto. El proceso describe una serie de flujos de trabajos y fases que son su fundamento estructural; además define qué entregables producir, cómo desarrollarlos y también provee los patrones para el diseño de los mismos.

RUP utiliza como el lenguaje de modelado a Unified Model Language (UML) (por sus siglas en español Lenguaje Unificado de Modelado) para la representación de los diagramas. A continuación se describen las características de dicho lenguaje.

1.5 Lenguaje Unificado de Modelado (Unified Model Language UML)

Es un lenguaje visual orientado al modelado de sistemas. Facilita un vocabulario controlado por reglas y símbolos, con el fin de que todos los empleados de un proyecto obvien las ambigüedades y la dispersión conceptual. Ayuda al usuario a comprender la realidad de la tecnología y la posibilidad de que reflexione antes de invertir o gastar grandes cantidades en proyectos que no estén seguros en su desarrollo, disminuyendo así, el costo y tiempo destinado a la construcción de los elementos que confeccionarán el modelo. Dispone de un repertorio de unidades (clases, acciones, objetos, estados y casos de uso). La gramática define reglas de combinación que forman otras unidades más complejas (diagramas y modelos) y tiene una determinada escala de abstracción (17).

UML es el lenguaje de modelado de sistemas de software más popular en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Además de lo anterior, ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Es sin lugar a dudas una herramienta muy poderosa cuando se intenta crear un sistema complejo y se hace necesaria la comprensión general del mismo por parte de todas las personas involucradas.

Es un estándar de modelado utilizado en sistemas orientados a objetos. Es sin lugar a dudas una herramienta muy poderosa de la cual no se puede prescindir cuando se intenta crear un sistema complejo como es el desarrollo de los Subsistemas Web y Administración de la Programación, donde se hace necesaria la comprensión general del mismo por parte de todas las personas involucradas.

Para lo modelación de este lenguaje existe un grupo de herramientas case que permiten la ayuda en todos los aspectos durante el ciclo de vida de desarrollo de un software.

1.6 Herramientas Case para la Modelación

Las herramientas CASE permiten organizar y manejar la información de un proyecto informático, tornando más flexibles y comprensibles los sistemas para los participantes de un proyecto y mejorando además la comunicación entre los mismos. Por esta razón se puede decir que representan una forma a través de la cual se pueden modelar los procesos de negocio de las empresas.

Durante la etapa de desarrollo de la arquitectura de la plataforma se definió como Herramienta Case a utilizar el Visual Paradigm debido a todas las facilidades que el mismo le aporta al desarrollo de dicha plataforma, así como ser la herramienta por excelencia para ser utilizada en un ambiente de software libre. El uso de herramientas CASE para la modelación UML proporciona mayor rapidez y entendimiento en el desarrollo de software. Este tipo de aplicación se ha convertido en ayuda y apoyo imprescindible para los desarrolladores. A continuación se describen detalles específicos de la herramienta case utilizada.

1.6.1 Visual Paradigm

Visual Paradigm es una Suite de herramientas CASE profesionales que utiliza “UML”: como lenguaje de modelado. Este sistema está dotado de productos o módulos que facilitan el trabajo durante la confección de un software durante su ciclo de vida. Paradigm ofrece la posibilidad de representar de forma gráfica los diagramas permitiendo ver el sistema desde diferentes perspectivas. Soporta la ingeniería tanto directa como inversa.

Características principales:

- Soporte de UML versión 2.1.
- Modelado de procesos de negocio.
- Requisitos de gestión.
- Mapeo objeto-relacional capa generación.
- IDE⁹ grandes integraciones.
- Generación de código e Ingeniería Inversa.

⁹ **IDE** (acrónimo en inglés de integrated development environment): entorno de desarrollo integrado, es un programa compuesto por un conjunto de herramientas para un programador.

- Colaboración de los equipos (cliente).
- PDF, MS Word, HTML de generación de informes.
- Importación y exportación a XML e imagen.
- Forma Editor, diseñar su propia forma (18).

Además, de las características antes mencionadas se pueden mencionar otras que también son de gran importancia para el desarrollo de la plataforma: colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto, genera la documentación del proyecto automáticamente en varios formatos como Word o PDF y permite el control de versiones. Además, se destacan su robustez, usabilidad y portabilidad. De acuerdo con lo expuesto anteriormente Visual Paradigm es la herramienta correcta para la modelación de cualquier proyecto sobre software libre permitiendo de esta forma la independencia y soberanía tecnológica.

Un gran número de ventajas posee la herramienta Visual Paradigm, permitiendo tener un elevado índice de aceptación, además de acogerse a la política del software libre. Es fácil de usar, amigable, genera código, informes y posee editor de figuras. Presenta sincronización entre diagramas de entidad-relación y diagramas de clases, interoperabilidad con otras aplicaciones, facilidad para redactar especificaciones de casos de uso del sistema, generación de documentos y de código ORM (Object Relational Mapping) (en español Mapeo Objeto-Relacional), disponibilidad de múltiples versiones, para cada necesidad, disponibilidad de integrarse en los principales IDEs y en múltiples plataformas. Se integra con varias herramientas Java: Eclipse/IBM WebSphere, JBuilder, NetBeans, IDE, Oracle JDeveloper, BEA Weblogic.

Una vez definido un conjunto de tecnologías y herramientas que se utilizarán, es necesario caracterizar los distintos paradigmas de programación, de los cuales se abordará en el próximo epígrafe, para luego poder seleccionar el más eficiente para la implementación de los subsistemas.

1.7 Paradigma de la Programación

Los paradigmas de programación constituyen una visión particular o filosófica para la construcción del software, pues se definen como un conjunto de reglas a seguir que facilitan el trabajo en la programación. Establecen recomendaciones de expertos programadores, que tras años de labor, han acumulado una gran experiencia.

En el mundo actual existen diferentes tipos de paradigmas de programación, no es mejor uno que otro sino que cada uno tiene ventajas y desventajas. Utilizando como basamento lo antes expuesto se dará

una pequeña descripción del paradigma utilizado para la construcción de la Plataforma de Transmisión Abierta para Radio y Televisión.

A continuación se caracterizarán un grupo de paradigmas de la programación para su correcto entendimiento, comenzando por la programación orientada a objeto.

1.7.1 Programación Orientada a Objeto

Es un paradigma que emplea clases, objetos, métodos, mensajes, atributos para diseñar aplicaciones y programas de computadora. Se basa en varias técnicas de programación, incluyendo herencia, modularidad, polimorfismo, abstracción y encapsulamiento, las cuales aseguran la re-usabilidad de código (19). Lenguaje orientado a objetos completamente, ejemplo de este es el C++ (20). Por las ventajas que ofrece para la programación estructurada y el trabajo en equipo es que se escoge como paradigma a usar en la construcción del sistema.

Características fundamentales de la programación orientada a objeto (POO):

1. Encapsulamiento: es la ocultación de información. Significa mantener la información dentro del objeto y mantenerlo como una caja negra. Puede ser accedida por métodos.
2. Abstracción: es la capacidad de aislar y encapsular la información del diseño y la ejecución. Es la capacidad para identificar atributos y métodos.
3. Herencia: es la propiedad que permite a los objetos crearse a partir de otros objetos. Cada subclase comparte características comunes con la clase de la que deriva. Las clases originales son llamadas clases base y las nuevas clases creadas a partir de ellas clases derivadas. Una clase derivada puede ser también clase base dando lugar a una jerarquía de clases.
4. Polimorfismo: es la capacidad de que diferentes objetos reaccionen de distinta forma a un mismo mensaje. Es la capacidad de referirse a objetos de clases distintas en una jerarquía utilizando el mismo elemento de programa (método) para realizar la misma operación, pero de manera diferente.

Para el diseño de lenguaje de programación existen varios paradigmas, dentro de los que se encuentra la programación lógica, que a continuación se explica.

1.7.2 Programación Lógica

Colección de declaraciones lógicas que especifican las características que debe tener la solución buscada. Se basa en hechos y reglas. En la programación lógica se realiza un cálculo predictivo para obtener los resultados.

Algunas de sus características son:

- Un cómputo se expresa por la búsqueda de pruebas o por la definición de predicados recursivos.
- Los resultados intermedios se pasan mediante unificación.
- No hay memoria implícita.
- Control basado en recursión.

Es difícil de comprender y dominar. Su implementación puede resultar compleja cuando los hechos y reglas no son claramente visibles. El uso de este paradigma resulta realmente viable para aplicaciones que modelen la inteligencia artificial, bases de datos o procesos.

La programación imperativa es un paradigma de programación que describe la programación en términos del estado del programa y sentencias que cambian dicho estado, lo que permite en el próximo epígrafe argumentar sobre sus características.

1.7.3 Programación Estructurada

Es un paradigma de programación con el cual el programador elabora programas de organización clara, mediante el uso de tres estructuras básicas de control lógico (Secuencia, Selección, Iteración). Un programa estructurado está formado por funciones, segmentos, módulos y/o subrutinas, cada una con una sola entrada y una sola salida. Este paradigma posee un teorema estructural o teorema fundamental, el cual alega que cualquier programa, no importa el tipo de trabajo que ejecute, puede ser construido utilizando únicamente las tres estructuras básicas.

La programación funcional también se encuentra dentro de los paradigmas de programación que se explicó anteriormente, por lo que a continuación se explican las características de la programación funcional.

1.7.4 Programación Funcional

El paradigma funcional está basado en el modelo matemático de composición funcional. En este modelo, el resultado de un cálculo es la entrada del siguiente, y así sucesivamente hasta que una composición produce el valor deseado.

No existe el concepto de celda de memoria que es asignada o modificada. Más bien, existen valores intermedios que son el resultado de cálculos anteriores y las entradas a cálculos subsiguientes. Tampoco existen sentencias imperativas y todas las funciones tienen transparencia referencial.

La programación funcional incorpora el concepto de función como objeto de primera clase, lo que significa que las funciones se pueden tratar como datos (pueden pasar como parámetros, calculadas y devueltas como valores normales, y mezcladas en el cálculo con otras formas de datos) (21).

Los programas tienen alto grado de recursividad y el rendimiento puede resultar bajo. Las variables sólo pueden tener asignado un solo valor a lo largo de la ejecución, lo cual implica que no puede existir asignación destructiva. Su mayor aplicación reside en la implementación de sistemas matemáticos o que modelen procesos secuenciales a través de funciones de los cuales se obtiene un resultado final.

Luego de caracterizar un conjunto de paradigmas, a continuación se realizará una selección del paradigma a utilizar, argumentando las razones por la cual fue escogido.

1.7.5 Criterio de Selección del Paradigma de Programación.

Una vez realizado la caracterización de un grupo de paradigmas de programación, fue seleccionada la programación orientada a objeto para la implementación de los dos subsistemas. El mismo posee un grupo de ventajas que permitieron su elección, como son:

- Es intuitiva, describe un problema en términos similares a los que utiliza la mente humana.
- Permite construir soluciones más seguras y con un mantenimiento más sencillo.
- Escalabilidad de las aplicaciones.
- Fomenta la reutilización y extensión del código siendo de gran ayuda para los implementadores de sistemas software.
- Permite la creación de programas más complejos y agiliza su desarrollo.
- Proporciona la creación de programas visuales y de prototipos.
- Facilita el mantenimiento de los sistemas y el trabajo en equipo.

Después de analizar los distintos paradigmas de programación, se hace inevitable definir qué lenguaje de programación se utilizará para la implementación de los subsistemas propuestos. En el próximo epígrafe se explica en qué consiste un lenguaje de programación.

1.8 Lenguajes de Programación

Se puede definir como lenguaje de programación a un conjunto de símbolos, sintaxis y reglas semánticas que permiten la comunicación entre una persona y un ordenador. Constituyen una técnica estándar de comunicación para entregarle instrucciones a la computadora.

Pueden clasificarse en dos grupos principales:

- Lenguajes de bajo nivel: se acercan al funcionamiento de una computadora. El lenguaje representativo de esta categoría es el código máquina. Las instrucciones en este lenguaje están formadas por cadenas binarias y los programas son dependientes de la computadora donde se escribieron. Puede citarse también el lenguaje ensamblador el cual trabaja directamente con los registros de la computadora y usa acrónimos para las instrucciones.
- Lenguajes de alto nivel: son fáciles de aprender porque están formados por elementos de lenguajes naturales y al usarlos puede dar la sensación de que las computadoras los comprenden. Los programas no dependen del ordenador donde fueron escritos.

En la actualidad existen diferentes lenguajes de programación para desarrollar en la web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas. Desde los inicios de Internet, fueron apareciendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. A medida que pasó el tiempo, las tecnologías fueron desarrollándose y aparecieron nuevos problemas a dar solución. Estos problemas conllevaron al desarrollo de lenguajes de programación para páginas web dinámicas que permitieran interactuar con los usuarios.

En el desarrollo de una página web es de vital importancia, inicialmente definir los lenguajes que se utilizaran por parte del cliente y del servidor. A continuación se argumentará las características de la selección de los lenguajes de programación, tanto del lado del servidor como del cliente.

1.8.1 Lenguajes de Programación del Lado del Cliente

Los lenguajes de programación del lado cliente, en la mayoría de los casos, se utilizan en la construcción de las páginas web clientes que son interpretadas por el navegador. Para la implementación de los Subsistemas Web y Administración de la Programación, se tuvieron en cuenta algunos de ellos, los cuales son referenciados a continuación.

HTML

HTML (Lenguaje de Marcas de Hipertexto) es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la sintaxis y coloca instrucciones especiales que no muestra el navegador, aunque si indica cómo desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados. También indica cómo hacer un documento interactivo a través de ligas especiales de hipertexto, las cuales conectan diferentes documentos ya sea en la computadora u

otros recursos de Internet (22). Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el contenido con objetos tales como imágenes.

En un documento HTML se pueden ver tres partes bien diferenciadas:

- Una cabecera del tipo de documento, en ella se especifica el tipo de documento HTML.
- Una cabecera de documento, donde se especifica información del documento.
- El cuerpo del documento, donde se coloca toda la información que contiene la página web.

JavaScript

JavaScript es un lenguaje de programación interpretado, lo que significa que, no es necesario compilar los programas para ejecutarlos. Explicándolo de otra manera, los programas que emplean este lenguaje se pueden comprobar directamente en cualquier navegador sin necesidad de realizar otros procesos por medio. Se emplea principalmente para la creación de páginas web dinámicas.

Al ejecutarse en el lado cliente, evita la sobrecarga del servidor por peticiones que pueden tener respuestas locales. Los Subsistemas Web y Administración de la Programación se verán beneficiados por la característica mencionada anteriormente, siendo de vital importancia para su correcto funcionamiento. No se debe olvidar la vinculación existente entre el lenguaje JavaScript y el conjunto de tecnologías AJAX descritas anteriormente, por lo que prescindir de su utilización en la realización de este trabajo no es una opción.

Una vez analizadas todas estas características se decidió seleccionar el ExtJS por ser un nuevo framework, basado originalmente en YUI¹⁰ pero actualmente es independiente del framework que se utilice (incluso puede usarse sin frameworks). Ofrece una gran cantidad de componentes para crear interfaces de usuarios complejas.

ExtJS

ExtJS empezó siendo un conjunto de librerías y extensiones para YUI (Yahoo! User Inteface). Con el tiempo se convirtió en un Framework independiente y a principios de 2007 se creó una compañía para comercializar y darle soporte. En la actualidad es una librería JavaScript para la creación de aplicaciones enriquecidas del lado del cliente mucho más atractiva y proporcionando alto rendimiento. Sus características principales son: Gran desempeño, componentes de interfaz de usuario personalizable, con

¹⁰ **YUI**, conjunto de utilidades y controles, escrito en JavaScript, para la construcción de aplicaciones Web interactivas, usando técnicas como DOM, DHTML y AJAX.

buen diseño, simple de configurar y usar, ya que tiene muchos componentes independientes. No es más que una biblioteca de JavaScript que se utiliza para el desarrollo de aplicaciones webs interactivas usando tecnologías como AJAX y DOM. Es muy bueno por ser una librería JavaScript ligera y de alto rendimiento, permite crear aplicaciones complejas utilizando componentes predefinidos, compatible con la mayoría de los navegadores que permiten crear páginas e interfaces webs dinámicas.

- Internet Explorer 6+.
- Firefox 1.5+ (PC, Mac).
- Safari 2+.
- Opera 9+ (PC, Mac).

¿Qué ventajas ofrece ExtJS?

- Un modelo de componentes muy bien diseñado y extensible.
- Una API¹¹ muy fácil de usar.
- Compatibilidad de Browsers¹².
- Licencias comerciales y Open Source.

Las características y ventajas del framework ExtJS dentro del modelo de AJAX, permitirá poder afirmar el uso de estas herramientas para el desarrollo de los Subsistemas Web y Administración de la Programación de la PTARTV.

Ajax solo se puede utilizar como una opción más rápida y eficiente de navegación web, no como el único método de navegación. Para ser más específicos sólo debería utilizarse en webs que intentan ser "aplicaciones" como lo es la plataforma, e incluso en estos casos hay que tener en cuenta que el usuario tiene que acostumbrarse a este tipo novedoso de sitios. De todos modos es inevitable que la web quiera parecerse cada vez más a aplicaciones de escritorio.

Luego de haber explicado los lenguajes de programación a utilizar en la plataforma por parte del cliente, se comenzará a describir las características de los lenguajes de programación utilizados por parte del servidor.

¹¹ **API** (application programming interface): Interfaz de programación de aplicaciones por sus siglas en español, es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción

¹² **Browsers**: Navegador web que permite visualizar la información que contiene una página web.

1.8.2 Lenguaje de Programación del Lado del Servidor

Los lenguajes del lado servidor son aquellos que son reconocidos, ejecutados e interpretados por el propio servidor y que se remiten al cliente en un formato claro para él. Uno de los más usados actualmente es PHP, el cual fue escogido anteriormente por los analistas del sistema que se implementa; a continuación se presenta la validación de su uso por parte de los desarrolladores.

Un lenguaje que se ejecuta en el lado del servidor es independiente del cliente por lo que es mucho menos rígido respecto al cambio de un navegador a otro o respecto a las versiones del mismo.

PHP

PHP (Hypertext PreProcessor), es un lenguaje del lado del servidor, esto indica que funciona en un servidor remoto que procesa las páginas web antes de que sean enviadas al navegador del usuario. Este lenguaje fue creado para el desarrollo de páginas web dinámicas. Se puede incluir dentro del código HTML con facilidad, además permite una serie de funcionalidades extraordinarias, por las cuales se ha convertido en el favorito de millones de programadores en todo el mundo (23).

Dado que es un lenguaje abierto dando la posibilidad de modificar el código fuente y añadir nuevas funcionalidades ha tenido una rápida evolución y desarrollo. Actualmente se encuentra en su versión PHP5.

Una de sus grandes potencialidades es su soporte para una gran cantidad de bases de datos. De las cuales se pueden mencionar InterBase, mSQL, MySQL, Oracle, Informix, PostgreSQL, entre otras. PHP también ofrece la integración con varias bibliotecas externas, que dan al desarrollador la posibilidad de realizar cualquier tarea, desde generar documentos en PDF (acrónimo del inglés Portable Document Format, formato de documento portátil) hasta analizar código XML.

En la actualidad, el desarrollo en php se realiza en gran medida a través de frameworks de desarrollo que no son más que un conjunto de bibliotecas y clases que proveen al programador de una gran gama de funciones que hacen más rápido su trabajo.

Después de haber definido los lenguajes de programación que se utilizarán tanto por la parte del cliente como del servidor, es necesario explicar el framework que se empleará para el desarrollo de los dos subsistemas planteados anteriormente.

1.9 Framework

Los frameworks en el proceso de desarrollo de software representan una estructura que provee soporte funcional (programas, bibliotecas, lenguajes interpretados, entre otros) y organización del trabajo. Esta estructura permite además agilizar el proceso de creación y mantenimiento de las aplicaciones. En el desarrollo de la mayoría de los frameworks se utiliza la POO lo que permite la reutilización del código empleado. Muchas empresas han girado su vista hacia el uso de frameworks en la producción de software.

Se pueden encontrar frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, y para cualquier otro ámbito. También existen varios tipos de frameworks web: orientados a la interfaz de usuario, como Java Server Faces, orientados a aplicaciones de publicación de documentos, como Coocon, orientados a la parte de control de eventos, como Struts y algunos que incluyen varios elementos como Tapestry. La mayoría de frameworks web se encargan de ofrecer una capa de controladores de acuerdo con el patrón MVC¹³, ofreciendo mecanismos para facilitar la integración con otras herramientas para la implementación de las capas de negocio y presentación.

Durante el desarrollo de la PTARTV hasta el flujo de Análisis y Diseño se había propuesto para su implementación el uso del framework php Symfony por las diversas prestaciones que puede brindar para la correcta implementación de la plataforma.

1.9.1 *Symfony*

Symfony es un completo framework, desarrollado por Fabien Potencier, y diseñado para perfeccionar el desarrollo de aplicaciones web que separa la capa lógica de negocio, la capa lógica de servidor y la capa de presentación de una aplicación. Facilita diversas herramientas y clases enfocadas en reducir el tiempo de perfeccionamiento de un sistema complejo. Este framework informatiza las tareas más comunes, brindando al desarrollador la posibilidad de dedicarse por completo a los aspectos específicos de cada aplicación. Está desarrollado completamente con PHP 5, última versión de PHP que incorpora la programación orientada a objetos. Ha sido probado en numerosos proyectos reales y se utiliza en entornos web de comercio electrónico de primer nivel. Es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas (Unix, Linux) como en plataformas Windows.

¹³ **MVC:** Patrón modelo vista controlador.

La capa de internacionalización que incluye Symfony permite la traducción de los datos y de la interfaz, así como la adaptación local de los contenidos. Los formularios incluyen validación automatizada y relleno automático de datos. La gestión de la caché reduce el ancho de banda utilizado y la carga del servidor. La autenticación y la gestión de credenciales simplifican la creación de secciones restringidas y la gestión de la seguridad de usuario. Las interacciones con Ajax son muy fáciles de implementar mediante los helpers¹⁴ que permiten encapsular los efectos JavaScript compatibles con todos los navegadores en una única línea de código (24).

Para el desarrollo de los Subsistemas Web y Administración de la Programación se seleccionó como solución factible este framework puesto a que brinda una licencia de código abierto, funciona sobre diversas plataformas (como Windows y Linux), permite la migración del sistema gestor de base de datos sin tener que hacer cambios en el código fuente de la aplicación. Symfony a su vez sigue las mejores prácticas y patrones de desarrollo para la web, como la separación estricta de código, diseño y el patrón modelo vista controlador (MVC). Patrón del que a continuación se explican sus características y ventajas que permitieron su utilización para la implementación de los dos subsistemas.

1.10 Patrón Arquitectónico

El Model-View-Controller (Modelo-Vista-Controlador, en adelante MVC) se introdujo inicialmente en la comunidad de desarrolladores de Smalltalk-80. El MVC divide una aplicación interactiva en tres áreas: procesamiento, salida y entrada. Para ello, emplea las siguientes abstracciones:

- **Modelo** (Model): encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.
- **Vista** (View): muestra la información al usuario. Obtiene los datos del modelo. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.
- **Controlador** (Controller): reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas y otros. Los eventos son traducidos a solicitudes de servicio (“*service requests*”) para el modelo o la vista. El usuario interactúa con el sistema a través de los controladores (25).

En la figura 2 que se muestra a continuación se representa como se encuentra conformado el patrón arquitectónico Modelo-Vista-Controlador.

¹⁴ **Helpers:** (Ayudantes): son funciones php que devuelven código HTML, que te permiten facilitar el proceso de escribir las plantillas.



Figura 2. Representación del patrón arquitectónico Modelo Vista Controlador.

Este patrón es muy popular, pues ha sido portado a una gran cantidad de entornos y frameworks, entre los que se encuentran WinForms, *ASP.Net*, *Symfony*. Herramientas de programación visual como *Visual Basic*, *Visual Studio.Net* y otras emplean también alguna variante de este esquema. El MVC es utilizado en múltiples plataformas y lenguajes. Algunos de sus principales beneficios son:

- Menor acoplamiento.
- Mayor cohesión.
- Las vistas proveen mayor flexibilidad y agilidad.
- Se puede crear, añadir, modificar y eliminar nuevas vistas dinámicamente.
- Facilita el mantenimiento.
- Mayor escalabilidad.

El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo son los datos provenientes de la lógica de negocio recogidos en el controlador y el controlador invoca a la lógica de negocio, utilizando los objetos de negocio. La vista y el controlador constituyen la interfaz del usuario. Además, se pueden distinguir esencialmente en la creación de instancias exclusivas de las vistas, la alta cohesión y la escalabilidad, además de ser menos costoso y factible. Todo lo antes mencionado permite que se seleccione este patrón para el desarrollo de los Subsistemas Web y Administración de la Programación.

Para comenzar a realizar la implementación existen herramientas que le permiten a los desarrolladores tener facilidades en la elaboración del código fuente como es el caso de los entornos integrados de desarrollo.

1.11 Entorno Integrado de desarrollo

Entorno de Desarrollo Integrado ó Integrated Development Environment (IDE), es un programa de aplicación generalmente conformado por un editor de código, compilador, depurador y un constructor de

interfaz gráfica de usuario. Puede ser independiente o miembro de otras aplicaciones compatibles. Presentan un marco de trabajo amigable y es posible que soporten varios lenguajes de programación.

Luego de tener claro los lenguajes de programación a utilizar para la implementación de los subsistemas, es necesario definir el IDE que se utilizará, donde a continuación se realizará una caracterización del NetBeans 6.8 por ser el escogido para el desarrollo de los dos subsistemas.

1.11.1 NetBeans IDE 6.8

NetBeans se ha convertido en un IDE apto para la mayoría de los lenguajes de programación open source¹⁵ modernos. Es un entorno de desarrollo que posee código abierto, con todos los beneficios del software disponible en forma gratuita, logrando un gran éxito con una gran base de usuarios y una comunidad en constante crecimiento. A su vez es una herramienta confeccionada para escribir, compilar, depurar y ejecutar programas, logrando su facilidad de uso, su cumplimiento de regulaciones, sus perfiles de rendimiento, además de su flexibilidad entre plataformas. Se encuentra escrito en Java por lo que soporta el desarrollo de todos los tipos de aplicaciones Java, pero se puede utilizar para cualquier otro tipo de lenguaje de programación. Aparte de la filosofía de distribución y desarrollo que lo respalda, ofrece la posibilidad de crear nuevas aplicaciones multiplataforma.

De igual forma, NetBeans ofrece las ventajas de que Sun Microsystems lo ha creado, respaldado y ha abierto su código. La relación de Sun con el código básico de NetBeans se remonta a ocho años atrás aproximadamente y, dado que el IDE tenía el código abierto, en el año 2000, la compañía ha seguido desempeñando el papel de patrocinadora, lo cual garantiza que NetBeans es un ambiente de desarrollo empresarial con soporte total. Igualmente elimina la necesidad de los equipos de desarrollo que tienen que invertir demasiado tiempo manteniendo los modelos actualizados con revisiones exhaustivas que garanticen la actualización de indicadores y códigos de anotación. Estas características juntas pueden ahorrar al desarrollador muchas horas de trabajo y acelerar la disponibilidad de un nuevo programa (26).

Dentro de sus características presenta el control de versiones permitiéndole ser muy útil para los trabajos en equipos. La flexibilidad entre plataformas, el cumplimiento de UML y la capacidad de administrar la complejidad ayudan a garantizar que las aplicaciones cumplan con los requerimientos específicos del negocio. Permite crear aplicaciones web con PHP 5, un potente debugger¹⁶ integrado, viene con soporte

¹⁵ **Open source:** Código abierto.

¹⁶ **Debugger:** Es un programa que permite depurar o limpiar los errores de otro programa informático.

para Symfony, fácil de instalar y funciona en muchos sistemas operativos incluyendo Windows, Linux, Mac OS X y Solaris.

El servidor web se convierte en una herramienta fundamental para el desarrollo de los subsistemas, ya que sirve contenido estático a un navegador, permite cargar un archivo y lo sirve mediante la red al navegador de un usuario. De esta forma, es necesario definir un servidor web que permita una correcta implementación de los subsistemas.

1.12 Servidor Web

Un servidor web es un programa que implementa protocolos como HTTP y HTTPS (Hypertext Transfer Protocol Secure) (en español: Protocolo Seguro de Transferencia de Hipertexto). HTTP está diseñado para transferir los llamados hipertextos, páginas web o páginas HTML. Está disponible por medio de Internet, emplea normas en XML y sistema de mensajería. Además, no está vinculado a ningún sistema operativo o lenguaje de programación (27).

Un servidor web se encarga de mantenerse a la espera de peticiones llevada a cabo por un cliente HTTP que se suele conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. A modo de ejemplo, al hacer una solicitud mediante el navegador, éste realiza una petición HTTP al servidor de dicha dirección, el servidor responde al cliente enviando el código HTML de la página; el navegador (cliente), una vez recibido el código, lo interpreta y lo muestra en pantalla.

Apache es el servidor web hecho por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa, permitiendo de esta forma que se utilice para la realización de este trabajo, donde a continuación se explicarán sus características.

1.12.1 Apache

Su desarrollo comenzó en febrero de 1995, por Rob McCool, en una tentativa de mejorar el servidor existente en el NCSA (National Center for Supercomputing Applications) (en español: Centro Nacional de Aplicaciones de Supercomputación). Es un servidor web de código abierto cuya primera versión apareció en enero de 1996, el Apache 1.0. Hacia el 2000, el servidor web Apache era el más extendido en el mundo. El nombre "Apache" es un acrónimo de "a patchy server" -un servidor de remiendos-, es decir, un servidor construido con código preexistente y piezas y parches de código. Es la auténtica «kill app» del software libre en el ámbito de los servidores y el ejemplo de software libre de mayor éxito, por delante

incluso del kernel Linux. Desde hace años, más del 60% de los servidores web de Internet emplean Apache (28), donde en la figura 3 que se muestra a continuación se muestra el logo de este potente servidor.



Figura 3. Logo del Servidor Apache.

Apache presenta características importantes que la distinguen y que se emplean en el desarrollo de la plataforma, entre las que se encuentran:

- **Altamente configurable de diseño modular:** es muy sencillo ampliar las capacidades del servidor web Apache. Cualquiera que posea una experiencia decente en la programación de C++ o Perl puede escribir un módulo para realizar una función determinada. Esto significa que hay una gran cantidad de módulos Apache disponibles para su utilización.
- **Sencillo, con configuración basada en un poderoso archivo:** el servidor Apache no posee una interfaz de usuario gráfica para su administración. Sin embargo, es lo suficientemente flexible para permitirle repartir la configuración de su host virtual en múltiples archivos para no sobrecargar un único archivo con toda la gestión de las múltiples configuraciones de servidores virtuales.
- **Soporte de autenticación HTTP:** soporta autenticación básica basada en la web. Está también preparado para autenticación basada en la asimilación de mensajes, que es algo que los navegadores webs populares ya han implementado.
- **Soporte de scripts PHP:** este lenguaje de script ha comenzado a ser muy utilizado y ofrece un extenso soporte de PHP (29).

Dado el gran uso del servidor web Apache a nivel internacional, su integridad y robustez, así como las enormes posibilidades que brinda al estar implementado sobre software libre; resulta la opción viable para la PTARTV y en específico de los Subsistemas Web y Administración de la Programación.

Una base de datos (BD) forma parte indispensable de toda aplicación que necesite gestionar información. Las mismas permiten almacenar toda la información en memoria auxiliar, logrando un acceso directo a los datos y permitiendo una buena seguridad y protección de toda la información. Por lo que es necesario definir cuál sistema de gestión de base dato se utilizó para implementar las funcionalidades que pertenecen a los dos subsistemas.

1.13 Sistema de Gestión de Base de Datos

Un Sistema de Gestión de Base de Dato (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos (30).

En la implementación de los dos subsistemas se hace indispensable la utilización de un sistema de gestor de datos, lo que permite seleccionar PostgreSQL 8.4. En el próximo epígrafe se describen un conjunto de características que permitieron su elección para el desarrollo de los subsistemas.

1.13.1 PostgreSQL 8.4

PostgreSQL como SGBD está derivado del paquete Postgres escrito en Berkeley, distribuido bajo licencia BSD¹⁷(Berkeley Software Distribution) (en español, Distribución de Software Berkeley). Con más de una década de desarrollo tras él, PostgreSQL ha demostrado ser un gestor de bases de datos de código abierto muy avanzado, ofreciendo control de concurrencia multiversión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación como pueden ser C, C++, Java, Perl, TCL, Python, PHP, Ruby, Pike y C#. La figura 4 siguiente representa el logo del PostgreSQL.



Figura 4. Logo de PostgreSQL.

Su utilización se hace factible ya que es muy rápido, tiene buenas utilidades de administración, es confiable, robusto y fácil de usar tanto para volúmenes de datos grandes como pequeños, sin límites en los tamaños de registros. Consecuentemente tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios. La conectividad, velocidad y seguridad, lo hacen altamente conveniente para acceder a bases de datos en internet ya que tiene mejor control de acceso a usuarios.

¹⁷ **BSD:** Licencia de software libre.

La utilización de PostgreSQL como SGBD de los Subsistemas Web y Administración de la Programación y de forma general de la PTARTV obedece al propósito de la elaboración de un sistema con gran robustez y alto nivel de escalabilidad.

El sistema operativo es un software que actúa de interfaz entre los dispositivos de hardware y los programas usados por los usuarios de una computadora. En el siguiente epígrafe se realiza una descripción del sistema operativo seleccionado para la plataforma y el porqué de su selección haciendo énfasis en algunas de sus características.

1.14 Sistema Operativo

El sistema operativo es el programa más importante en una computadora. Para que los restantes programas de uso general funcionen en cada computadora es necesario incluir un sistema operativo. Es el encargado de ejecutar las tareas básicas, entre la que se mencionan, el 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 y otros. Además, se encarga de la seguridad, asegurándose de que usuarios no autorizados no tengan acceso al sistema.

En sistemas grandes, el sistema operativo tiene incluso mayor responsabilidad y poder, es como un policía de tráfico, se asegura de que los programas y usuarios que están funcionando al mismo tiempo no interfieran entre ellos. El sistema operativo también es responsable de la seguridad, asegurándose de que los usuarios no autorizados no tengan acceso al sistema (31).

Al abogarse por el uso del software libre para el desarrollo de la PTARTV y en específico de los Subsistemas Web y Administración de la Programación, se impone entonces la necesidad de una correcta selección de la distribución de Linux a utilizar.

1.13.1 Distribución de Linux

La distribución de Linux, no es más que una selección de programas y ficheros, organizados y preparados para su instalación. Estas distribuciones se pueden conseguir mediante Internet, o con la compra de sus CDs¹⁸, los cuales incluyen todo lo que se necesita para la instalación de un sistema Linux bastante completo (32). Existen muchas y variadas distribuciones de GNU/Linux, una de ellas es Ubuntu, la cual será empleada como sistema operativo en el desarrollo del trabajo, ya que en la actualidad la

¹⁸ **CDs**, (Compact Disc), soporte digital óptico utilizado para almacenar cualquier tipo de información (audio, fotos, video, documentos y otros datos).

mayoría de los usuarios se encuentran muy estrechamente relacionados con las interfaces gráficas y es precisamente esta distribución quien se encuentra más orientada al escritorio.

Algunas de las características que presenta GNU/Linux son:

- **Multitarea:** la palabra multitarea describe la habilidad de ejecutar varios programas al mismo tiempo. Linux utiliza la llamada multitarea preventiva, la cual asegura que todos los programas que se están utilizando en un momento dado serán ejecutados, siendo el sistema operativo el encargado de ceder tiempo de microprocesador a cada programa.
- **Multiusuario:** muchos usuarios usando la misma máquina al mismo tiempo.
- **Multiplataforma:** las plataformas en las que en un principio se puede utilizar Linux son 386, 486, Pentium, Pentium Pro, Pentium II, Amiga y Atari, también existen versiones para su utilización en otras plataformas, como Alpha, ARM, MIPS, PowerPC y SPARC.
- **Multiprocesador:** soporte para sistemas multiprocesadores, están disponibles para Intel y SPARC.
- **Todo el código fuente** está disponible, incluyendo el núcleo completo y todos los controladores (drivers), las herramientas de desarrollo y todos los programas de usuario; además todo ello se puede distribuir libremente. Hay algunos programas comerciales que están siendo ofrecidos para Linux actualmente sin código fuente, pero todo lo que ha sido gratuito sigue siendo gratuito (33).

Ubuntu es una distribución GNU/Linux que brinda las opciones de un sistema operativo centrado en ordenadores personales, aunque también ofrece soporte para servidores. Se ha convertido sin dudas en una de las distribuciones más importantes a nivel mundial. Tiene su base en Debian GNU/Linux y su principal objetivo es la facilidad y libertad de uso, la fluida instalación y los lanzamientos regulares.

Presenta una serie de ventajas como son:

- Es gratuito y no presenta ningún coste adicional para la edición empresarial.
- Usa lo mejor en infraestructura de traducciones y accesibilidad que la comunidad del software libre es capaz de ofrecer, permitiendo que sea utilizable por el mayor número de personas posible.
- Se publica de manera regular y predecible, se publica una nueva versión cada seis meses. Permite usar la versión estable actual o ayudar a mejorar la versión actualmente en desarrollo. Cada versión está soportada al menos durante 18 meses.
- Está totalmente comprometido con los principios del desarrollo de software de código abierto (34).

Debido a sus características, se convierte en una distribución orientada al escritorio, pero con bastante estabilidad. Fundamentalmente comparte las ventajas de Debian (exceptuando que tiene ligeramente menos paquetes, y que estos no están probados del todo), sumándole el hecho de tener una distribución bastante actualizada.

Por las razones anteriormente expuestas, Ubuntu se convierte en una solución robusta, para el desarrollo de este trabajo. Por tanto, será usado como sistema operativo para la PTARTV la última versión de la distribución Ubuntu.

1.15 Conclusiones Parciales

En el transcurso de este capítulo se abordaron varios temas que conforman la fundamentación teórica para el desarrollo de los Subsistemas Web y Administración de la Programación. Se explicó el objetivo de las funcionalidades de cada subsistema, entre ellos los procesos asociados con la televisión y la radio de la plataforma, así como el sistema de encuesta que brindará el portal web y la cartelera de la programación, el foro y las estadísticas de la visualización de los videos publicados. Se analizaron las características de un conjunto de tendencias y tecnologías, así como las herramientas asociadas a la posible propuesta para solucionar el problema científico que concierne la presente investigación. Dentro de las que se encuentra la arquitectura cliente-servidor y el conjunto de tecnologías Ajax para lograr una mayor rapidez en la navegación dentro de la plataforma. Se caracterizó la metodología de desarrollo de software, utilizando el Proceso Unificado (RUP) debido a la amplia documentación que brinda, lo cual es esencial por la concepción como producto de la Plataforma de Transmisión Abierta para Radio y Televisión. Se brindaron una serie de características de diversos paradigmas de programación como: Programación Lógica, Imperativa, Estructurada, Funcional y Orientada a Objeto, siendo esta última seleccionada para la implementación del grupo de funcionalidades. También caracterizando a continuación los lenguajes de programación tanto del lado del cliente como: HTML, JavaScript, ExtJS y del lado del servidor PHP. Se describió el framework utilizado Symfony, así como el patrón arquitectónico Modelo-Vista-Controlador y el entorno integrado de desarrollo NetBeans 6.8. Se detalló el servidor Web Apache, utilizado para la implementación de los subsistemas, analizando luego las características del Sistema Gestor de Base de Datos PostgreSQL en su versión 8.4, destacándose por su robustez, facilidades de uso y soporte. De esta forma, se facilitan los conocimientos teóricos necesarios para el entendimiento del presente trabajo.

Capítulo 2

Descripción de la Propuesta

CAPÍTULO 2

Descripción de la solución propuesta

Introducción

En este capítulo se realiza una breve descripción de las funcionalidades a implementar correspondiente a cada subsistema, así como los distintos módulos que lo conforman. Se muestra el modelo de implementación de la plataforma y los diagramas de componentes correspondientes a los Subsistema Web y Administración de la Programación, con el objetivo de que se comprendan cada una de las funciones que desempeña cada subsistema.

2.1 Descripción de los Subsistemas

Dentro de la Plataforma de Transmisión Abierta para Radio y Televisión existen ocho subsistemas: Administración de la Transmisión, Transferencia, Reporte, Gestión de Medias, Seguridad, Transmisión, Web y Administración de la Programación. Para lograr una total funcionalidad de la plataforma, es imprescindible culminar la implementación de algunas funcionalidades clave adicionadas en los dos últimos subsistemas relacionados anteriormente. El Subsistema de Administración de la Programación surgió la necesidad de reestructurarlo, provocando la necesidad de implementarlo de manera total. El Subsistema Web ya encuentra implementado los módulos NoticiasWeb, PublicacionWeb y SeccionWeb por lo que a continuación se describen los módulos restantes para la culminación de ambos subsistemas, los cuales se ha separado en módulos siguiendo la línea de desarrollo que propone el framework Symfony.

Dentro de la Plataforma de Transmisión Abierta para Radio y Televisión existen ocho subsistemas: Administración de la Transmisión, Transferencia, Reporte, Gestión de Medias, Seguridad, Transmisión, Web y Administración de la Programación. Para lograr una total funcionalidad de la plataforma, es imprescindible culminar la implementación de algunas funcionalidades claves adicionadas en los dos últimos subsistemas relacionados anteriormente. Cada una de ellas se ha separado en módulos siguiendo la línea de desarrollo que propone el framework Symfony y se relacionan a continuación.

2.1.1 Funcionalidades del Subsistema Web

La plataforma se comenzó a desarrollar basado en el framework Symfony, el cual en su estructura divide al sistema en módulos que agrupan funcionalidades con un alto grado de similitud o dependencias entre

ellas, todo esto para hacer el trabajo del desarrollador más fácil y comprensible. El Subsistema Web ya encuentra implementado los módulos NoticiasWeb, PublicacionWeb y SeccionWeb por lo que a continuación se describen los módulos restantes para la culminación del subsistema.

➤ **Módulo Encuestas**

Este módulo es el encargado de gestionar las encuestas que se publiquen en la plataforma. Mediante el uso de estas se podrá conocer el criterio de los usuarios acerca del tema en cuestión publicado. De esta manera, el administrador de la plataforma podrá tener conocimiento de las preferencias del cliente. Es importante señalar que las encuestas que gestionará la plataforma serán de complejidad baja, pues solamente estarán compuestas por una pregunta y varias respuestas asociadas con el objetivo de conocer puntos de vista del usuario sobre determinado material o cualquier otro elemento importante para la dirección.

➤ **Módulo Estadísticas**

Este módulo permitirá tener una constancia del rating¹⁹ de una determina media publicada en la web. El administrador de la plataforma podrá determinar cuáles fueron las medias más vistas por los usuarios, así como sus preferencias, siendo de vital importancia para la dirección al evaluar el total de medias observadas.

➤ **Módulo Foro**

Es de gran importancia que la plataforma cuente con un foro, ya que el usuario podrá tener una técnica de comunicación con otros usuarios o con el administrador del sistema, basada en un contenido de interés general. Este módulo logrará alcanzar un nivel mucho más elevado de interactividad con los usuarios de la plataforma en busca de retroalimentación.

➤ **Módulo Cartelera**

La cartelera permitirá que el usuario tenga la posibilidad de conocer la programación de los canales de radio y televisión realizada por el planificador o programador. La actualización de dicha cartelera será de forma dinámica lo cual contribuirá a eliminar errores humanos que pudieran realizarse durante el proceso y a la vez logrará una correspondencia con lo que se anuncia con antelación y lo que se está transmitiendo en determinado horario. Logrará de esta manera que el cliente nunca se encuentre desinformado con relación a las medias que la plataforma visualizará en una semana determinada.

¹⁹ **Rating:** Es una cifra que indica el porcentaje de visualización realizado por los usuarios acerca de una media.

Una vez explicadas las nuevas funcionalidades que se le agregaron al Subsistema Web, se comenzará a explicar las funciones que debe realizar el Subsistema de Administración de la Programación.

2.1.2 Funcionalidades del Subsistema Administración de la Programación

La tarea de este subsistema es poder realizar una programación de las medias que se visualizarán mediante la plataforma de forma automática. El Subsistema de Administración de la Programación surgió la necesidad de reestructurarlo, provocando la necesidad de implementarlo de manera total. El mismo se encuentra conformado por 4 módulos que se encarga de gestionar cada una de las prestaciones correspondientes a dicho subsistema.

➤ **Módulo Espacios Televisivos.**

Este módulo permite la gestión de espacios de televisión, que consiste en fijar momentos dentro de una programación televisiva en días determinados de la semana para que se transmitan espacios televisivos que deben obligatoriamente ser respetados, como por ejemplo el Noticiero Nacional de la Televisión Cubana.

➤ **Módulo Espacios Radiales.**

Los espacios radiales dentro de la plataforma desempeñan un papel muy parecido al de los espacios televisivos, diferenciándose que solo los espacios radiales son de audio. Por lo permite la gestión de espacios radiales, que consiste en fijar momentos dentro de una programación de radio en días determinados de la semana para que se transmitan espacios radiales que deben obligatoriamente ser respetados.

➤ **Módulo Programación Televisiva.**

Este módulo es el encargado de manejar todo lo relacionado con la programación de la televisión en la plataforma. Permitiendo la gestión de las programaciones de televisión que se transmitirán por los diferentes canales de televisión de la plataforma.

➤ **Módulo Programación Radial.**

La programación radial presenta la misma concepción de la programación televisiva, diferenciándose en las medias que se transmitirán, que solo serán de audio y el canal será uno específico para la radio. Por lo que permitirá manejar todo lo relacionado con la gestión de las programaciones de radio que se transmitirán por los diferentes canales de radio de la plataforma.

Una vez descritos cada uno de los módulos pertenecientes a su respectivo subsistema, se continuará con la descripción del modelo de implementación para la plataforma y se realizará una descripción más detallada para los subsistemas que son objeto de la presente investigación.

2.2 Modelo de Implementación

El modelo de implementación denota la implementación actual del sistema en términos de componentes y subsistemas de implementación (35). Describe como se organizan los elementos de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y cómo dependen los componentes entre sí, dentro de los cuales podemos encontrar: ficheros de código fuente, script, datos, archivos, ejecutables y directorios.

El principal objetivo del modelo de implementación es reconocer los componentes físicos de la implementación para que puedan gestionarse y comprenderse mejor. Las principales unidades de integración alrededor de las cuales se organizan los equipos se definen dentro de este modelo, así como las que se pueden versionar, desplegar y reemplazar separadamente.

El siguiente modelo de implementación representa componentes que hacen referencia a bibliotecas y aplicaciones externas del propio framework de desarrollo, utilizadas para la implementación de algunas funcionalidades de la plataforma como son: el uso de LDAP²⁰, para la autenticación mediante dominio, también se utiliza Propel para el mapeo de objetos a base de datos (conocido como ORM, de “object-relational mapping”) para PHP, se encarga de la generación del modelo, ya que crea el esqueleto o estructura básica de las clases y genera automáticamente el código necesario para la interacción con la base de datos, proporcionando persistencia para los objetos y un fácil servicio de consultas.

La Figura 5 a continuación muestra el modelo de implementación de la Plataforma de Transmisión Abierta para la Radio y la Televisión.

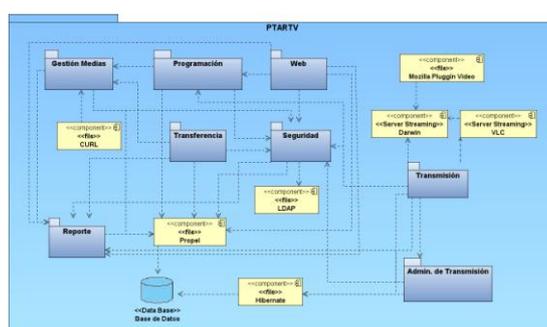


Figura 5. Modelo de Implementación de la PTARV.

A continuación se muestra como se encuentra elaborado la estructura del Subsistema Web y el de Administración de la Programación.

²⁰ **LDAP**, (*Lightweight Directory Access Protocol*), en español, Protocolo de Acceso a Directorios Ligeros, protocolo usado para acceder a "Servidores de Directorio".

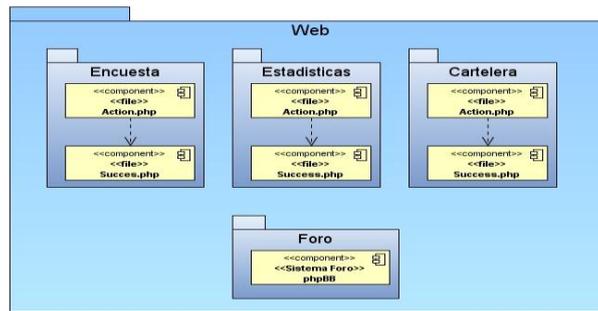


Figura 6. Estructura del Subsistema Web.



Figura 7. Estructura Subsistema Administración de la Programación.

Para lograr una mejor comprensión del modelo de implementación, se detalla cómo están divididos los Subsistema Web y Administración de la Programación a través de sus diagramas de componentes.

2.2.1 Diagramas de Componentes

Los diagramas de componentes muestran la estructura de componentes, incluyendo los clasificadores que especifican componentes y los artefactos que los implementan (36). La metodología RUP enfatiza en el uso del diagrama de componentes para modelar los subsistemas de implementación. Los componentes representan los elementos de un modelo dentro de un paquete, como son las clases en el modelo de diseño. Presentan estereotipos estándar como pueden ser:

- <<file>> ficheros de datos o código fuente.
- <<executable>> programas que se ejecutan en un nodo.
- <<table>> tabla de base de datos.
- <<library>> librerías estáticas o dinámicas.
- <<document>> documento.

A continuación se representan los diagramas de componentes relacionados con los subsistemas que se implementarán, detallando el flujo de trabajo de cada módulo perteneciente a cada subsistema.

Subsistema Administración de la Programación

Módulo Espacios Televisivos

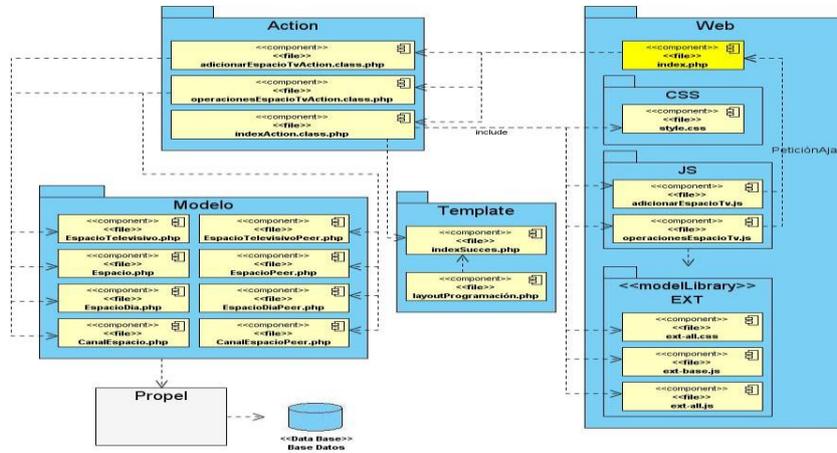


Figura 8. Diagrama de componentes de caso de uso Gestionar Espacio Televisivo.

Módulo Espacios Radiales

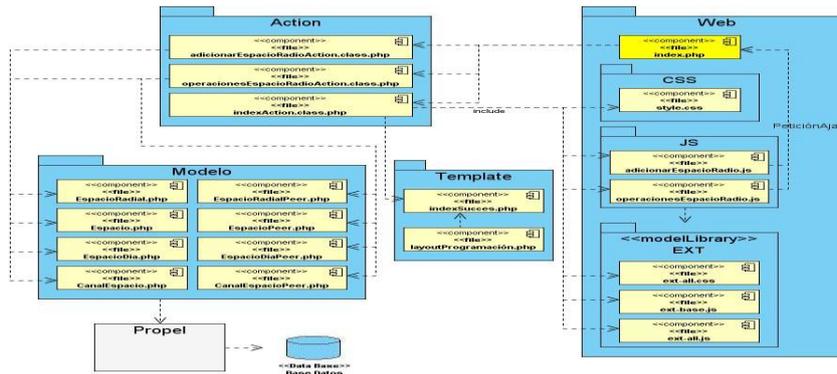


Figura 9. Diagrama de componentes de caso de uso Gestionar Espacio de Radio.

Módulo Programación de Televisión

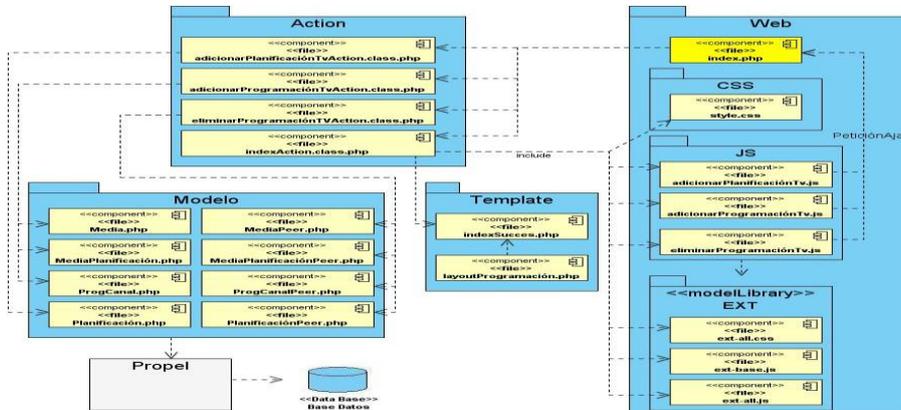


Figura 10. Diagrama de componentes de caso de uso Gestionar Programación de Televisión.

Módulo Programación Radial

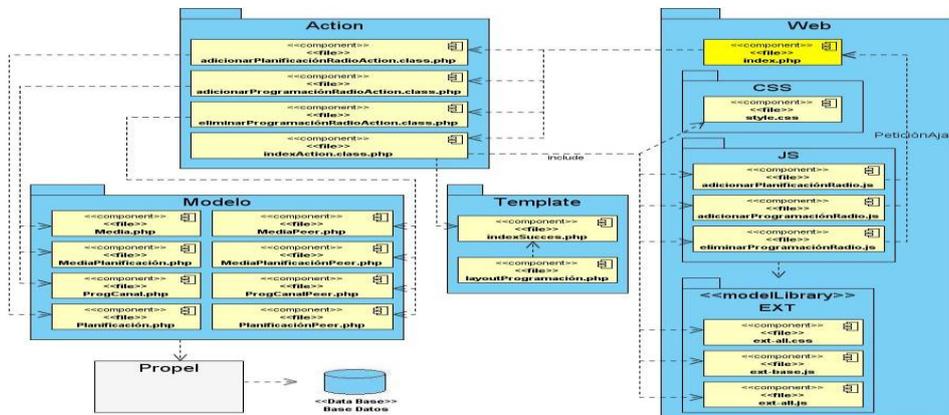


Figura 11. Diagrama de componentes de caso de uso Gestionar Programación de Radio.

Subsistema Web

Módulo Encuesta

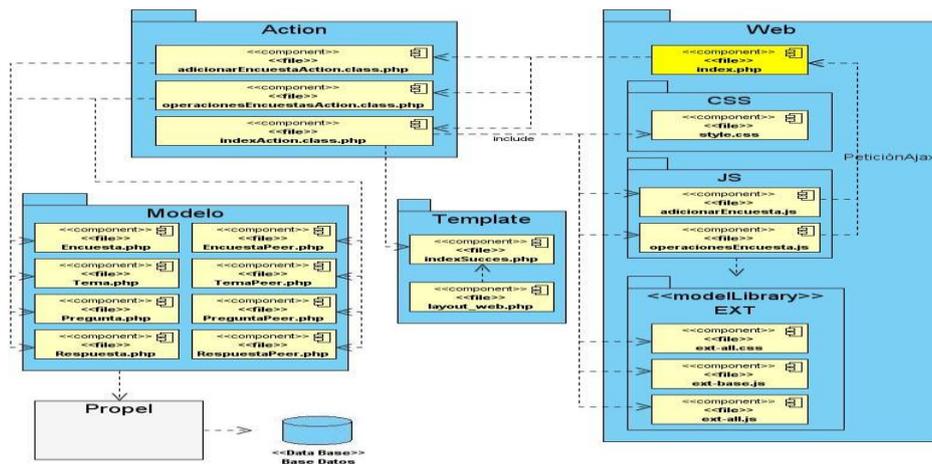


Figura 12. Diagrama de componentes del módulo Encuesta.

Módulo Cartelera

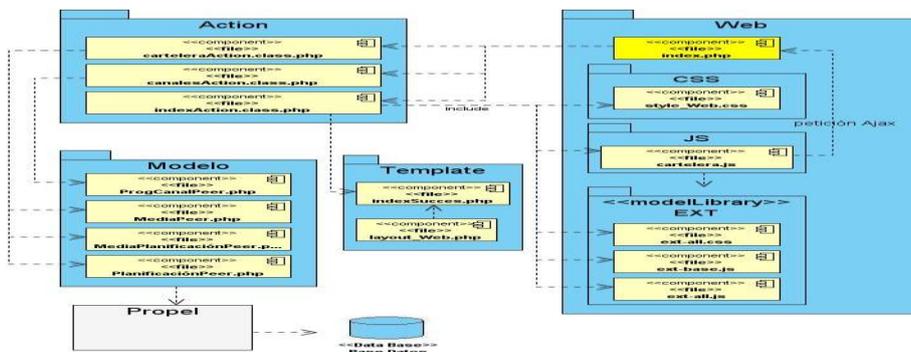


Figura 13. Diagrama de componentes del módulo Cartelera.

Módulo Estadísticas

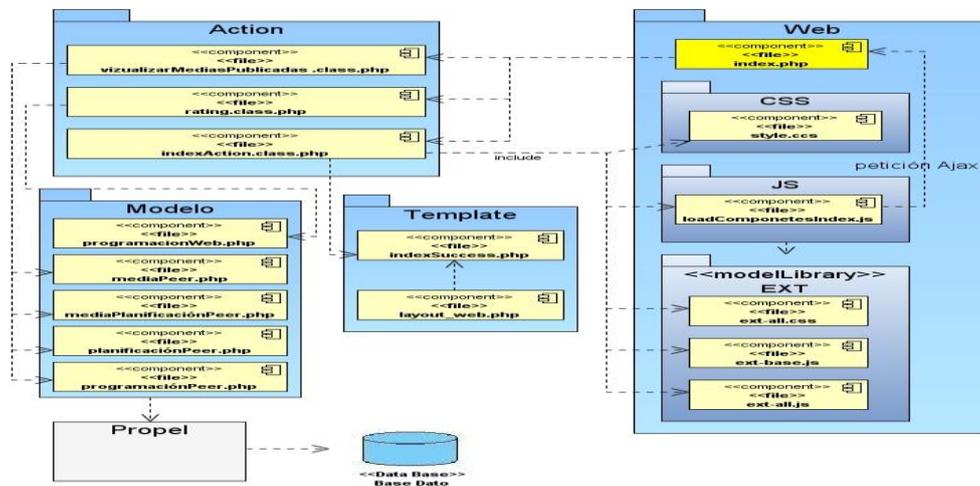


Figura 14. Diagrama de componentes del módulo Estadísticas.

Para la realización del módulo Foro se tubo en cuenta que en la etapa de desarrollo de la arquitectura de la plataforma se definió, utilizar el sistema de encuestas phpBB, su nombre es por la abreviación de PHP Bulletin Board. El mismo no es más que un sistema de foros gratuito basado en un conjunto de paquetes de código, programados en el lenguaje de programación web PHP. Es gratuito y de código abierto lo que permite la creación ilimitada de foros y subforos. Tiene un panel de administración fácil de utilizar, y un proceso de instalación fácil para poder disponer de un foro en cuestión de minutos. Funciona sobre bases de datos basadas en el lenguaje SQL como MySQL, PostgreSQL, Microsoft SQL Server, también sobre Oracle, donde almacena la información para poder recuperarla en cada petición del lenguaje. PhpBB presenta un grupo de características que le permiten convertirse en una solución factible a la hora de integrar un sistema de foro a la plataforma.

- Registro de usuarios y personalización de cada campo.
- Mensajes privados a múltiples usuarios y carpetas de mensajes.
- Búsqueda de temas y usuarios.
- Panel de administrador y de moderador por separado.
- Creación de encuestas con múltiples opciones.
- Perfil para cada usuario con sus datos personales.
- Los usuarios pueden tener amigos o enemigos, los mensajes de enemigos son ocultados automáticamente.
- Creación de grupos de usuarios, moderadores o administradores.
- Poder crear nuevos campos para el perfil de usuario.

Con un gran conjunto de características mencionadas anteriormente, y altamente configurable, junto con una interfaz compatible con los estándares, phpBB será un complemento ideal para la PTARTV, por lo que a continuación en la figura 15, se muestra el logo de phpBB.



Figura 15. Logo de phpBB.

Luego de explicar el flujo de trabajo de los módulos que pertenecen a cada subsistema mediante diagramas de componentes para un mejor entendimiento de los mismos, así como las características del phpBB utilizado para el módulo foro, se hace necesario caracterizar el estilo de programación a utilizar durante la etapa de implementación.

2.3 Estilos de Programación

Se define estilo de programación a la manera en que se le da un formato al código fuente, esto incluye la forma en que se colocan las llaves, cómo se indenta el código y dónde se emplean los paréntesis. Todo programador posee un estilo propio de escribir su código fuente, pero para poder lograr un buen estilo de programación deberá utilizar una estructura de código que sea fácil de entender, no solo para terceras personas, sino también para él mismo. Un buen estilo de programación consta de diversos criterios, entre los que se mencionan:

- Nombres significativos para variables, controles y procedimientos.
- Indentación (sangrías) y espacios apropiados en el código.
- Documentar el código (poner comentarios para aclarar).
- Módulos adaptables.
- Minimizar alcance de los datos hasta donde sea posible.

Nombres significativos: los nombres de variables, controles y procedimientos definidos por un usuario deben ser significativos. Con el fin de ayudar al usuario en la asignación de nombres se brinda un estándar de nombres convencionales.

Indentación y espacio apropiado en el código: la indentación se emplea para obtener una mejor visibilidad en el diseño de un programa. La misma muestra las líneas que están sujetas a otras. Por ejemplo, todas las líneas que forman el cuerpo de un ciclo serán indentadas con la instrucción principal del mismo.

Documentar el Código: las secciones de código más complicado e inusual deberán ser documentadas. Cada variable y arreglo tendrán que estar comentarios en el lugar que sean definidos para que su función pueda ser comprendida posteriormente.

Procedimientos Coherentes: cada procedimiento deberá ser diseñado para una tarea simple, pues si alguno de ellos maneja muchas tareas, es lógico que pueda ser difícil de entender y pueda ocurrir fácilmente un error.

Minimizar alcance de los datos hasta donde sea posible: las variables y los arreglos pueden ser accedidos mediante códigos en distintas partes de un programa y si son globales desde cualquier lugar. Esto sería lo ideal, pero si no hay cuidado pudiesen ocurrir algunos efectos extraños en otras partes del programa, como colocar un valor en una variable por error. Restringiendo el rango de acceso o el alcance de una variable o un arreglo se puede evitar este problema. Un alcance intermedio es el acceso necesario para una variable durante todo el simple módulo. Esto significa que cualquier procedimiento en el módulo puede tener acceso a la variable, pero los procedimientos en otros módulos tienen acceso denegado.

Mediante el empleo de un estilo de programación para la confección de los Subsistemas Web y Administración de la Programación es posible obtener un código más legible, permitiendo de esta forma la reducción de errores lógicos en el código y su reutilización.

En la actualidad existen varios estilos de programación desarrollados por diferentes autores reconocidos, estos son utilizados por una gran parte de los desarrolladores a nivel mundial debido a su fácil estructura y organización. Para la programación de los subsistemas se seleccionó el estilo Allman, también conocido como "ANSI style". Seguidamente se describen algunos criterios que lo conforman.

➤ Indentación

Plantea que se debe usar los sangrados para indentar el código, nunca espacios. Poner las llaves de control en la línea subsiguiente. En la figura 16 que se brinda a continuación se evidencia un ejemplo claro de la indentación.

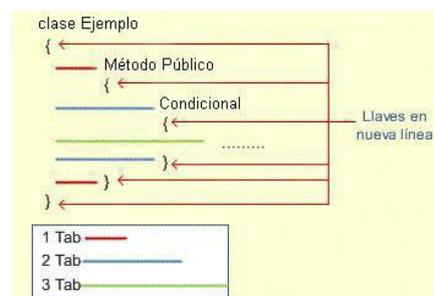


Figura 16. Ejemplo de Indentación.

➤ Saltos de Línea

Define que se debe añadir un salto de línea después del cierre de los paréntesis de los parámetros y detrás de un punto y coma, cuando termina la sentencia. En la figura 17 a continuación se muestra un claro ejemplo de los saltos de líneas.

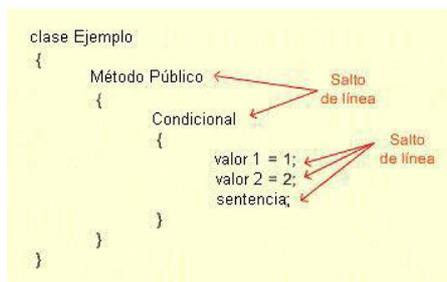


Figura 17. Ejemplo de Saltos de Líneas.

➤ Espacios y Líneas en Blanco

Este criterio plantea que se debe:

- ✓ Usar espacios en blanco para mejorar la legibilidad del código.
- ✓ Usar espacios en blanco a ambos lados del operador de símbolos, después de comas y después de las declaraciones.
- ✓ Usar líneas en blanco para separar trozos de código.
- ✓ Usar líneas en blanco antes de cada método dentro de la clase.

A continuación se representa con la figura 18 un claro ejemplo de los espacios y líneas en blanco.

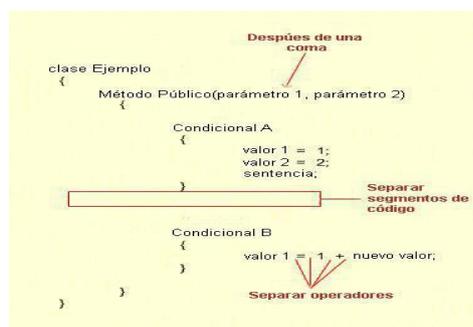


Figura 18. Ejemplo de Líneas y Espacios en Blanco.

El estilo de Allman se caracteriza por conservar un código limpio y claro. Las llaves de inicio y fin coinciden en la misma columna, permitiendo de esta forma que la identificación de cada bloque sea más fácil. Este presenta el inconveniente de que las llaves ocupan una línea enteramente, ocupando más espacio vertical de lectura.

Criterios de estándares para el estilo Allman.

1. Variables locales:

- Los nombres de algunas variables locales, como los iteradores o los contadores, pueden especificarse en minúscula y de forma abreviada, siempre que su contexto sea específicamente local y su lectura sea intuitiva. Ejemplos: \$cont, \$i, \$j.
- Al hacer asignaciones, debe existir un espacio a ambos lados del signo igual (=), esto funciona tanto para asignar un valor fijo, de otra variable o del resultado de una función.

2. Estructura de Control:

- Incluye if, for, while, switch, etc. Deben tener un espacio entre la palabra clave y el paréntesis de apertura, para diferenciarlos de las llamadas a funciones. Se recomienda encarecidamente, aunque no sea necesario, la utilización de llaves. Esto mejora la legibilidad y disminuye la posibilidad de errores lógicos al agregar nuevas líneas de código.

3. Indentación y largo de líneas:

- Indentar con 4 espacios, sin tabulador, para que cualquier editor de texto reconozca correctamente la indentación. Por otro lado, si bien existen editores que realizan corte automático de línea, es recomendable hacerlo en forma manual a los 75-80 caracteres.

4. Documentar el código:

- Se aconseja el uso de comentarios en línea para facilitar la comprensión del código, sobre todo en procedimientos complejos. Los comentarios pueden ser con fin documental o bien como 'ayuda-memoria'.
- Se recomienda utilizar los estilos de C (`/* */`) y C++ (`//`).

5. Llamadas a funciones:

- Las funciones deben ser llamadas sin espacio entre el nombre de la función, el paréntesis de apertura y el primer parámetro. En caso de varios parámetros, separar con espacios entre la coma y cada parámetro, y sin espacios entre el último parámetro, el paréntesis de cierre y el punto y coma.

6. Definición de clases:

- El nombre debe ser descriptivo, evitando abreviaturas.
- La llave de inicio de la clase se coloca en la línea siguiente, indentada correctamente.
- Todos los miembros de la clase deben ser privados, es decir, únicamente accesibles a través de métodos de la misma.
- Si el framework de desarrollo especifica convenciones de nombres de clases /métodos/ propiedades, respetarlas.

2.4 Conclusiones Parciales

Durante este capítulo se expusieron las distintas funcionalidades que presentan el Subsistema Web y el de Administración de la Programación, permitiendo detallar su funcionamiento a nivel de módulo como: Encuesta, Estadísticas, Foro y Cartelera, todos estos dentro del subsistema Web. Mientras que el Subsistema de Administración de la Programación se encuentra dividido en los módulos: Espacios Televisivos, Espacios Radiales, Programación Televisiva y Programación Radial. Inicialmente se representó el modelo de implementación, logrado de esa forma explicar la integridad de la plataforma a nivel de subsistema. Se mostró el funcionamiento interno de cada módulo a través de los diagramas de componentes, de esa manera se logra un correcto entendimiento del flujo trabajo en cada módulo. Después de haber sido analizado y detallado las características de los estilos de programación, se definió y caracterizó el estilo de programación Allman puesto en práctica en la implementación de los subsistemas, por lo que servirá de ayuda a otros desarrolladores a entender el código fuente ya que será limpio y claro.

Capítulo 3

Validación de la Solución Propuesta

CAPÍTULO 3

Validación de la solución propuesta

Introducción

En el desarrollo de aplicaciones web, las pruebas son de vital importancia, ya que aseguran la calidad de la aplicación. En el presente capítulo se aborda todo lo referente a las herramientas y elementos empleados en la validación de la aplicación con el objetivo de garantizar el correcto funcionamiento de la misma.

3.1 Automatización de Pruebas

Los desarrolladores de aplicaciones web emplean gran esfuerzo en probar correctamente una aplicación, pues resulta ser una tarea verdaderamente tediosa crear casos de prueba, ejecutarlos y analizar sus resultados. Debido a los constantes cambios que surgen en los requisitos de una aplicación y la refactorización continua del código es muy probable que aparezcan errores en la misma, es por ello que la automatización de pruebas es una opción muy útil para la creación de un entorno de desarrollo satisfactorio. Este tipo de pruebas obligan a los programadores a crear nuevas pruebas en un formato estandarizado y muy rígido que pueda ser procesado por un framework de pruebas.

Las pruebas automatizadas pueden reemplazar la documentación técnica de la aplicación, ya que ilustran de manera clara el funcionamiento de la aplicación. Estas pruebas comparan el resultado de un método con la salida esperada del mismo, es decir, evalúan “asertos”, que no son más que expresiones de tipo `$a == 2`. El valor de salida de un aserto puede ser `true` o `false`, lo que determina si la prueba tiene éxito o falla. La palabra “aserto” tiene un uso muy común en las técnicas de automatización de pruebas.

La automatización de pruebas es uno de los mayores avances en la programación. Concretamente en el desarrollo de las aplicaciones web, las pruebas aseguran la calidad de la aplicación incluso cuando el desarrollo de nuevas versiones es muy activo. Durante el próximo epígrafe se explicará los tipos de pruebas utilizadas en los subsistemas como son las unitarias y funcionales.

3.2 Pruebas Unitarias y Funcionales

Dependiendo del modelo del software empleado para el desarrollo de aplicaciones y su implementación estos dos tipos de pruebas pueden desempeñar un papel primordial. Para el caso de las pruebas

unitarias se analiza una porción de código que pueda ser analizada de manera aislada, como por ejemplo funciones y métodos, a los que después de pasarles parámetros de entrada se obtienen otros de salida claramente definidos, es decir, validan la forma en la que las funciones y métodos trabajan de manera particular. Este tipo de prueba se encarga de validar los casos de uso uno a uno, lo que significa que un único método puede necesitar varias pruebas unitarias si su funcionamiento varía en función del contexto.

Existen numerosos frameworks para crear pruebas unitarias, dentro de los que se destacan el PHPUnit y SimpleTest. En el caso particular de Symfony incluye su propio framework para pruebas llamado Lime, el cual se basa en la librería Test::More de Perl, además de ser compatible con TAP, lo que significa que los resultados de las pruebas se muestran con el formato definido en el “Test Anything Protocol”, creado para facilitar la lectura de los resultados de las pruebas.

Además de proporcionar soporte para pruebas unitarias, Lime tiene otras ventajas que lo caracterizan:

- Las pruebas de Lime son fáciles de leer y sus resultados también lo son, debido que utiliza diferentes colores para mostrar de forma clara la información más importante.
- El núcleo de Lime se valida mediante pruebas unitarias.
- Emplea el lenguaje PHP, el cual es muy rápido y está bien diseñado internamente.

Las pruebas funcionales evalúan la aplicación como un todo, comprueban que se alcancen determinados resultados tras realizar una serie de acciones sobre la aplicación y validan partes de las aplicaciones. Estas pruebas simulan la navegación del usuario, realizan peticiones y comprueban los elementos de respuesta, tal y como lo haría manualmente un usuario para validar que una determinada acción hace lo que se supone que debe que hacer. En ellas se ejecuta un escenario correspondiente a lo que se denomina un “caso de uso”.

Para este tipo de pruebas el framework Symfony dispone de un objeto especial, llamado sfBrowser, que actúa como un navegador que está accediendo a una aplicación, pero sin necesidad de utilizar un servidor web real. Este objeto permite el acceso directo a los objetos que forman cada petición (el objeto petición, el objeto sesión, el contexto y el objeto respuesta). También dispone de una extensión de esta clase llamada sfTestBrowser, que está especialmente diseñada para las pruebas funcionales y que tiene todas las características de sfBrowser, además de algunos métodos muy útiles para los asertos.

Para realizar pruebas de una manera correcta, es necesaria la utilización de herramientas que permiten reproducir las funcionalidades de cada subsistema, las mismas serán explicadas en el siguiente epígrafe.

3.2.1 Herramientas Empleadas en las Pruebas.

Los elementos descritos anteriormente que incluye Symfony para realizar pruebas unitarias y funcionales no son suficientes para la mayoría de los casos, sobre todo a la hora de probar las interacciones del lado cliente. El principal inconveniente que presentan estas técnicas es que no pueden simular el comportamiento de JavaScript. Si se definen interacciones muy complejas, como por ejemplo interacciones con Ajax, es necesario reproducir de forma exacta las pulsaciones de teclado que realiza el usuario y ejecutar los scripts de JavaScript. Normalmente, estas pruebas se hacen a mano, pero necesitan de la dedicación de mucho tiempo y son propensas a cometer errores.

Una solución a los problemas antes expuestos, es el empleo de Selenium, el cual no es más que un framework de pruebas escrito completamente en JavaScript. La principal ventaja del uso de esta herramienta es que permite realizar una serie de acciones en la página de la misma forma que las haría un usuario normal. Además de la ventaja de Selenium sobre el objeto sfBrowser de Symfony de ser capaz de ejecutar todo el código JavaScript de la página, incluidas las interacciones creadas con Ajax. Las pruebas de Selenium se escriben en HTML, por lo que cada caso de prueba consiste en una página HTML, con una tabla de 3 columnas: comando, destino y valor, la cual se muestra un ejemplo en la figura 19 que a continuación se brinda.

```

4 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
5 <link rel="selenium.base" href="" />
6 <title>Test Espacio Radio</title>
7 </head>
8 <body>
9 <table cellpadding="1" cellspacing="1" border="1">
10 <thead>
11 <tr><td rowspan="1" colspan="3">Test Espacio Radio</td></tr>
12 </thead><tbody>
13 <tr>
14 <td>open</td>
15 <td>/programacion</td>
16 <td></td>
17 </tr>
18 <tr>
19 <td>click</td>
20 <td>EspacioRadio</td>
21 <td></td>
22 </tr>
23 <tr>
24 <td>verifyTextPresent</td>
25 <td>Adicionar</td>
26 <td></td>
27 </tr>
28 </tbody></table>
29 </body>
30 </html>

```

Figura 19. Fragmento de caso de prueba desarrollado con Selenium.

Las pruebas de Selenium al ser creadas con HTML, arriban a que sea muy aburrida y tediosa la creación de cada prueba de manera manual. Para ello existe una extensión de Selenium para Firefox que permite grabar todos los movimientos y acciones realizadas sobre una página y guardarlos como una prueba. Una vez realizado todos los movimientos y añadidos todos los comandos, se pueden guardar en un archivo HTML para añadirlo al conjunto de pruebas. Este conjunto o suite de prueba se define en archivo TestSuite.html, el cual se encarga de guardar las referencias a cada uno de los archivos de pruebas

guardados independientemente y mostrarlos en una lista. Seguidamente se muestra a través de la figura 20 un ejemplo de un TestSuite.html.

```

1 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
2 <head>
3   <meta content="text/html; charset=UTF-8" http-equiv="content-type" />
4   <title>Test Suite</title>
5 </head>
6 <body>
7   <table id="suiteTable" cellpadding="1"
8     cellspacing="1" border="1"
9     class="selenium"><tbody>
10
11 <tr><td><b>Test Suite</b></td></tr>
12
13 <tr><td><a href="Espacio Radio">Espacio Radio</a></td></tr>
14
15 </tbody></table>
16 </body>
17 </html>
  
```

Figura 20. Fragmento del TestSuite.html.

La figura 21 a continuación muestra la interfaz gráfica del plugin Selenium para el navegador web Firefox y contiene además una breve explicación de las diferentes partes que la conforman.

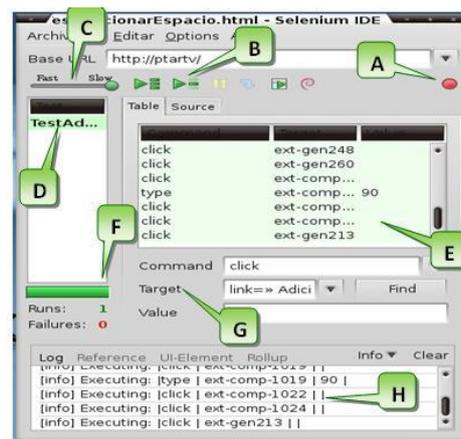


Figura 21. Elementos del Selenium IDE.

- A:** graba los movimientos de la navegación.
- B:** permite escoger que prueba presente en la *suite* ejecutar, una a una o todas a la vez. Pausar o continuar las pruebas y realizarlas paso a paso.
- C:** determinar la velocidad de ejecución de pruebas para que el usuario pueda apreciar la prueba.
- D:** suite de Pruebas, lista todas las pruebas presentes en la *suite*.
- E:** datos de las pruebas presentes en el código de cada una.
- F:** cantidad de pruebas realizadas: aceptadas (verde) y fallidas (rojo).
- G:** permite cambiar los valores de una prueba en tiempo de ejecución.
- H:** log. Registra los eventos y sucesos de cada prueba.

Después de seleccionar y caracterizar la herramienta necesaria para las pruebas de cada subsistema, se realizará en el próximo epígrafe un diseño de pruebas que nos permitan transformar los casos de uso en caso de prueba, para probar de esa forma que el sistema funcione correctamente como un todo.

3.3 Diseño de Pruebas

Para validar el conjunto de funcionalidades necesarias para cada subsistema se realizan algunas pruebas funcionales apoyadas en el uso de Selenium. Las pruebas que realiza son semejantes a las que haría cualquier usuario desde un navegador, con la ventaja de que las hace mucho más rápido y evita el trabajo repetitivo de probar una y otra vez lo mismo manualmente. A continuación se muestra la relación de varios casos de prueba implementados para la validación de la solución propuesta.

Caso de Prueba	Adicionar Encuesta
Condiciones	El usuario debe estar registrado en la base de datos como usuario del sistema. Y su rol debe ser el de Administrador Web.
Datos de Entrada	Nombre, tema, pregunta y respuestas de la encuesta.
Resultados esperados	La encuesta se guardará en la BD y se mostrarán sus datos en el panel que contiene las encuestas del sistema.
Resultados obtenidos	La prueba es satisfactoria. Se le informa al usuario que la encuesta se adicionó correctamente a la BD y se muestra en el panel que contiene las encuestas del sistema.
Observaciones	Al realizar la prueba con la herramienta Selenium se obtuvo el resultado correcto.

Tabla 1. Caso de Prueba # 1.

```

Test Adicionar Encuesta
open /noticiasWeb
click link=Encuesta
click //div[@id='menu4']/ul/li/a[2]/canvas
verifyTextPresent Nombre de Encuesta:
verifyTextPresent
verifyTextPresent Tema de Encuesta:
verifyTextPresent
verifyTextPresent Pregunta de Encuesta:
verifyTextPresent
verifyTextPresent Tipo de Respuestas:
verifyTextPresent
click ext-comp-1106
type ext-comp-1106 Nombre
type ext-comp-1110 Tema
type ext-comp-1107 Pregunta

```



Figura 22. Resultado de Caso de Prueba # 1.

Caso de Prueba	Crear Programación de Televisión
Condiciones	El usuario debe estar registrado en la base datos como usuario del sistema. Y su rol debe ser el de Administrador de Programación.
Datos de Entrada	Fecha, Canal.
Resultados esperados	La planificación de la programación quedará correctamente almacenada en la BD y luego se podrá visualizar en la sección de cartelera.
Resultados obtenidos	La prueba es satisfactoria. Se almacenó en la BD la planificación de la programación y los usuarios podrán observarla a través de la cartelera.
Observaciones	Al realizar la prueba con la herramienta Selenium se obtuvo el resultado correcto.

Tabla 2. Caso de Prueba # 2.

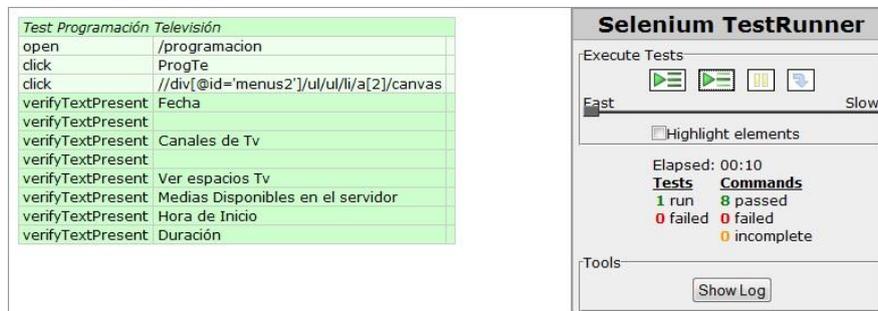


Figura 23. Resultado de Caso de Prueba # 2.

Caso de Prueba	Adicionar Espacio Radio
Condiciones	El usuario debe estar registrado en la BD y su rol debe ser el de Administrador de Programación.
Datos de Entrada	Espacio de radio.
Resultados	Que el espacio de radio quede correctamente adicionado.

esperados	
Resultados obtenidos	La prueba es satisfactoria. El espacio de radio resultó correctamente adicionado.
Observaciones	Al realizar la prueba con la herramienta Selenium se obtuvo el resultado correcto.

Tabla 3. Caso de Prueba # 3.

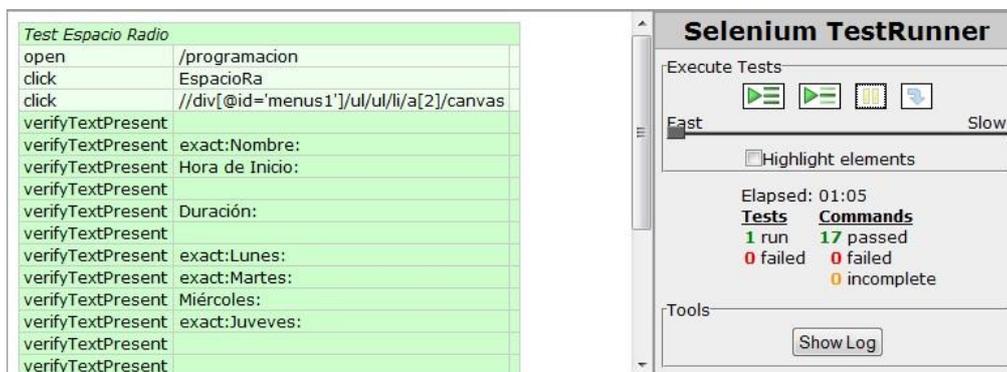


Figura 24. Resultado de Caso de Prueba # 3.

Caso de Prueba	Eliminar Espacio Radio
Condiciones	El usuario debe estar registrado en la BD y su rol debe ser el de Administrador de Programación.
Datos de Entrada	Debe existir en la plataforma un espacio de radio creado.
Resultados esperados	Que el espacio de radio quede correctamente eliminado.
Resultados obtenidos	La prueba es satisfactoria. El espacio de radio quedó correctamente eliminado.
Observaciones	Al realizar la prueba con la herramienta Selenium se obtuvo el resultado correcto.

Tabla 4. Caso de Prueba # 4.

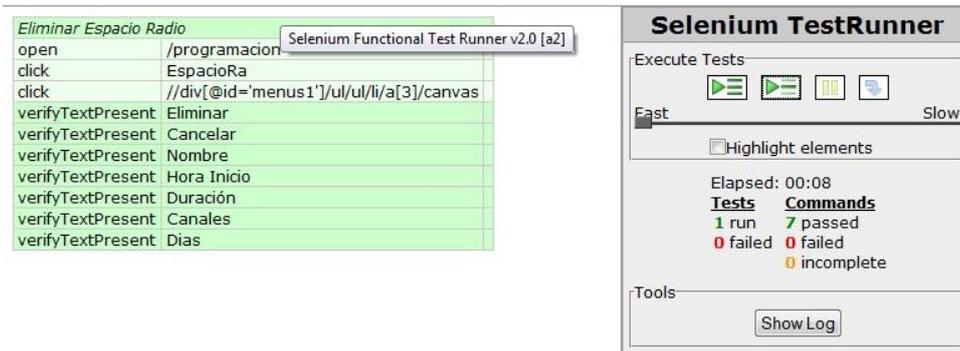


Figura 25. Resultado de Caso de Prueba # 4.

Caso de Prueba	Modificar Espacio Radio
Condiciones	El usuario debe estar registrado en la BD y su rol debe ser el de Administrador de Programación.
Datos de Entrada	Debe existir en la plataforma un espacio de radio creado, al cual se le realizarán las modificaciones.
Resultados esperados	Que el espacio de radio quede correctamente modificado.
Resultados obtenidos	La prueba es satisfactoria. El espacio de radio quedó correctamente modificado.
Observaciones	Al realizar la prueba con la herramienta Selenium se obtuvo el resultado correcto.

Tabla 5. Caso de Prueba # 5.

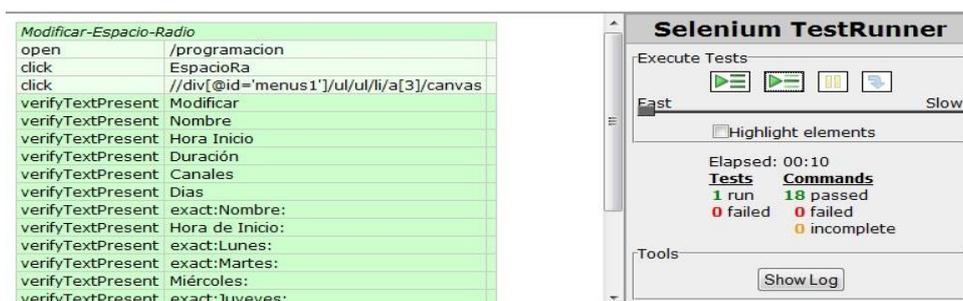


Figura 26. Resultado de Caso de Prueba # 5.

Caso de Prueba	Eliminar Programación de Televisión
Condiciones	El usuario debe estar registrado en la base datos como usuario del sistema. Y si rol debe ser el de Administrador de Programación.
Datos de Entrada	Debe existir en la plataforma una programación de televisión creada.
Resultados esperados	La programación de televisión quedará correctamente eliminada de la plataforma.
Resultados obtenidos	La prueba es satisfactoria. Se eliminó de la plataforma la programación de televisión.
Observaciones	Al realizar la prueba con la herramienta Selenium se obtuvo el resultado correcto

Tabla 6. Caso de Prueba # 6.

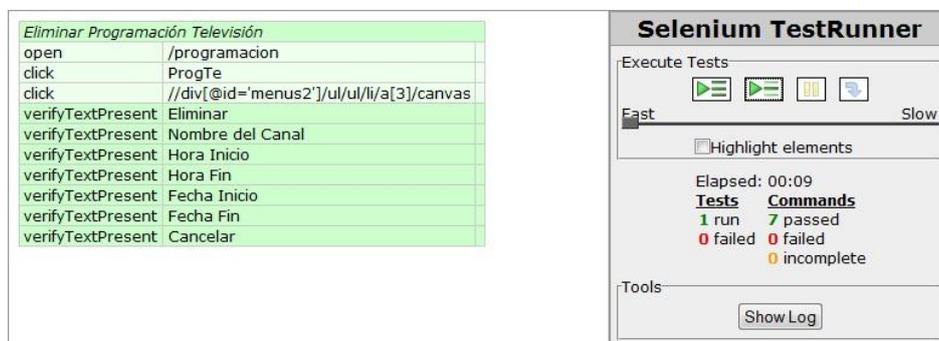


Figura 27. Resultado de Caso de Prueba # 6.

Luego de mostrar un grupo de ejemplos de casos de pruebas realizados a varios casos de uso es importante señalar que no fueron las únicas pruebas que se efectuaron. Por lo que a continuación se brinda una tabla con las estadísticas que brindaron el resultado de todas las pruebas realizadas a todo el conjunto de funcionalidades necesario para el desarrollo de este trabajo.

Nombre de Caso de Uso	Cantidad de Prueba	Pasos Necesarios	Pasos Fallidos	Resultado
Gestionar Encuesta.	4	39	0	Satisfactorio (100 %)
Visualizar Estadísticas.	1	3	0	Satisfactorio (100 %)
Visualizar Cartelera.	1	10	0	Satisfactorio (100 %)
Gestionar Espacio de Televisión.	4	54	0	Satisfactorio (100 %)
Gestionar Espacio de Radio.	4	49	0	Satisfactorio (100 %)
Gestionar Programación de Televisión.	5	33	0	Satisfactorio (100 %)
Gestionar Programación de Radio.	5	33	0	Satisfactorio (100 %)
Total	24	221	0	Satisfactorio (100 %)

Tabla 7. Resultado del las Pruebas de cada Caso de Uso.

Una vez realizado todas las pruebas necesarias para el grupo de funcionalidades de cada subsistema y representadas mediante una tabla, se puede observar los resultados satisfactorios obtenidos en las pruebas, demostrando de esa manera la correcta implementación de cada funcionalidad, sin encontrar fallos en el mismo, de esta manera la plataforma es cada vez más confiable y consistente. Hay que tener claro que inicialmente muchas de estas pruebas arrojaron una serie de errores, los cuales fueron solucionados y finalmente las pruebas realizadas una vez corregidos los errores, demostraron tener éxito.

3.4 Conclusiones Parciales

El objetivo principal de este capítulo fue precisamente conocer como se manejan los conceptos de pruebas en el contexto de la implementación de una aplicación. Por lo que se hizo referencia a los principales elementos relacionados con las pruebas unitarias y funcionales, realizadas a la fase de implementación y las herramientas empleadas, como lo fue Selenium. Se describieron los diseños de pruebas realizados a cada caso de uso, dentro de los que se encuentra Adicionar Encuesta, Crear Programación de Televisión y Adicionar Espacio Televisivo. Lo que permitió una vez realizadas las pruebas, demostrar el correcto funcionamiento de cada funcionalidad, ya que se obtuvo los resultados esperados, por lo que las pruebas realizadas fueron satisfactorias.

CONCLUSIONES GENERALES

El desarrollo de los subsistemas Web y Administración de la Programación brindó la posibilidad de documentar parte de las tareas propuestas para el desarrollo de la aplicación y la implementación de la Plataforma de Transmisión Abierta para Radio y Televisión. Se puede afirmar que con el desarrollo de esta investigación práctica:

- ✓ Se explicó el objetivo de las funcionalidades de cada subsistema, entre ellos los procesos asociados con la televisión y la radio de la plataforma, así como el sistema de encuesta que brindará el portal web y la cartelera de la programación, el foro y las estadísticas de la visualización de los videos publicados.
- ✓ Se analizaron las características de un conjunto de tendencias y tecnologías, así como las herramientas asociadas a la posible propuesta para solucionar el problema científico que concierne la presente investigación. Dentro de las que se encuentra la arquitectura cliente-servidor y el conjunto de tecnologías Ajax.
- ✓ Se caracterizó la metodología de desarrollo de software, utilizando el Proceso Unificado (RUP) debido a la amplia documentación que brinda, lo cual es esencial por la concepción como producto de la Plataforma de Transmisión Abierta para Radio y Televisión.
- ✓ Se brindaron una serie de características de diversos paradigmas de programación como: Programación Lógica, Imperativa, Estructurada, Funcional y Orientada a Objeto, siendo esta última seleccionada para la implementación del grupo de funcionalidades.
- ✓ Se caracterizó los lenguajes de programación tanto del lado del cliente como: HTML, JavaScript, ExtJS y del lado del servidor PHP, describiéndose el framework utilizado Symfony, así como el patrón arquitectónico Modelo-Vista-Controlador y el entorno integrado de desarrollo NetBeans 6.8.
- ✓ Se detalló el servidor Web Apache, utilizado para la implementación de los subsistemas, las características del Sistema Gestor de Base de Dato PostgreSQL en su versión 8.4, destacándose por su robustez, facilidades de uso y soporte.
- ✓ Se representó el modelo de implementación, logrado de esa forma explicar la integridad de la plataforma a nivel de subsistemas.
- ✓ Se mostró el funcionamiento interno de cada módulo a través de los diagramas de componentes, de esa manera se logra un correcto entendimiento del flujo trabajo en cada módulo.
- ✓ Se definió y caracterizó el estilo de programación Allman puesto en práctica en la implementación de los subsistemas.

- ✓ Se hizo referencia a los principales elementos relacionados con las pruebas unitarias y funcionales, realizadas a la fase de implementación y las herramientas empleadas, como lo fue Selenium.
- ✓ Se describieron y ejecutaron los casos de pruebas que certificaron el correcto funcionamiento de las funcionalidades desarrollada, ya que se obtuvo los resultados esperados.
- ✓ Se obtuvieron todos los artefactos necesarios para dar cumplimiento al objetivo general trazado.

De forma general se puede concluir que se cumplieron satisfactoriamente todos los objetivos enunciados desde el inicio de la investigación práctica, por lo que se le dio solución al problema científico planteado. Se incluyen a continuación una serie de recomendaciones que deben tenerse en cuenta para el trabajo futuro.

RECOMENDACIONES

Al término del presente trabajo de diploma el autor recomienda:

- ✓ Realizar un estudio de posibles mejoras para facilitar la adaptabilidad del sistema a las necesidades de diferentes clientes.
- ✓ Continuar trabajando en el perfeccionamiento e implementación de nueva servicios televisivos para los dos subsistemas, logrando de esta forma una plataforma con muchas más prestaciones.
- ✓ Actualizar la versión de framework Symfony utilizado, así como las librerías ExtJS.
- ✓ Aumentar la complejidad de sistemas de encuestas que forma parte la plataforma.

REFERENCIAS BIBLIOGRÁFICAS

1. **Ministerio de Educación, Política Social y Deporte.** Media. [En línea] 2008. [Citado el: 20 de Noviembre de 2009.] [Disponible en: <http://recursos.cnice.mec.es/media/television/bloque1/index.html#>].
2. **Evolución de las tecnologías de Telecomunicaciones.** [En línea] 2008 [Citado el: 22 de Noviembre de 2009] [Disponible en: http://www.coit.es/foro/pub/ficheros/evolucion_de_las_tecnologias_comunicacion_d80bbc52.pdf]
3. **Alejandra Padilla Juárez.** alipso.com. [En línea] 2008. [Citado el: 20 de Noviembre de 2009.] [Disponible en: <http://www.alipso.com/monografias2/EEpyAuZVZydkpvgjVU.shtml>].
4. **Miguel Yuste.** ELPAIS.COM. [En línea] 2007. [Citado el: 15 de Noviembre de 2009.] [Disponible en: http://www.elpais.com/articulo/sociedad/inventor/WWW/celebra/aniversario/CERN/elpepusoc/20090313elpepusoc_6/Tes].
5. **Manuel Ballesteros Corzo.** Control de un brazo robotico por medio de tecnologías web. Pamplona : s.n., 2008. [Disponible en: http://www.villanuevitas.galeon.com/diplomas/anexos_part2.pdf].
6. **Sergio Luján Mora.** Programación de aplicaciones web: historia, principios basicos y cliente web. [En línea] ISBN 84-8454-206-8, 2009. [Citado el: 4 de Noviembre de 2009.] [Disponible en : <http://www.editorial-club-universitario.es/libro.asp?ref=367>].
7. **Instituto Nacional de Estadística e Informática.** [En línea] 2009. [Citado el: 10 de Diciembre de 2009.] [Disponible en: <http://www1.inei.gob.pe/biblioineipub/bancopub/inf/Lib5038/defi.HTM>].
8. **MasterMagazine.** [En línea] 2007. [Citado el: 18 de Noviembre de 2009.] [Disponible en: <http://www.mastermagazine.info/termino/4294.php>].
9. **Librosweb.es.** [En línea] 2009. [Citado el: 20 de Noviembre de 2009.] [Disponible en: <http://librosweb.es/ajax/capitulo1.html>].
10. **MSDN.** [En línea] Microsoft Corporation, 2010. [Citado el: 16 de Noviembre de 2009.] [Disponible en: <http://msdn.microsoft.com/es-es/library/bb398874.aspx>].
11. **Maestros del web.** [En línea] 2008. [Citado el: 28 de Noviembre de 2009.] [Disponible en: <http://www.maestrosdelweb.com/editorial/ajax/>].
12. **Jesse James Garrett.** adaptive path. [En línea] 2009. [Citado el: 2 de Diciembre de 2009.] [Disponible en: <http://adaptivepath.com/ideas/essays/archives/000385.php>].
13. **Universidad de Murcia.** [En línea] 2008. [Citado el: 11 de Enero de 2010.] [Disponible en: <http://www.um.es/docencia/barzana/IAGP/lagp2.html>].
14. **Ivar Jacobson, Grady Booch y James Rumbaugh.** El Proceso Unificado de Desarrollo de Software. [En línea] Madrid: Pearson Educación, 2000. ISBN 84-7829-036-2. [Citado el: 27 de Noviembre de 2009] [Disponible en: <http://www.diazdesantos.es/libros/jacobson-ivar-el-proceso-unificado-de-desarrollo-de-software-L0000532300445.html>].

- 15. Patricio Letelier Torres.** Desarrollo de Software Orientado a Objeto usando UML. [En línea] 2009. [Citado el: 29 de Enero de 2010.] [Disponible en: http://www.slidefinder.net/c/curso_con_uml/3086359].
- 16. Johanna Rijas Rojas y Emilio José Barrios.** Investigación sobre estado del arte en diseño y aplicaciones de pruebas de software. [En línea] 2007. [Citado el: 30 de Enero de 2010.] [Disponible en: <http://www.udistrital.edu.co/comunidad/grupos/arquisoft/fileadmin/Estudiantes/Pruebas/HTML%20-%20Pruebas%20de%20software/index.html>].
- 17. Ivar Jacobson, Grady Booch y James Rumbaugh.** El Lenguaje Unificado de Modelado. Manual de Referencia. [En línea] 2000. ISBN 78-84-7829-087-1. [Citado el: 27 de Noviembre de 2009] [Disponible en: <http://www.agapea.com/libros/El-lenguaje-unificado-de-modelado-2ED-isbn-8478290761-i.htm>].
- 18. Visual Paradigm.** [En línea] 2008. [Citado el: 25 de Enero de 2010.] [Disponible en: <http://www.visual-paradigm.com/product/vpuml/enterpriseedition.jsp>].
- 19. Universidad de Burgos.** [En línea] 2009. [Citado el: 24 de Diciembre de 2009.] [Disponible en: http://pisuerga.inf.ubu.es/lsi/Invest/Java/Tuto/I_1.htm].
- 20. Universidad Tecnológica Nacional.** [En línea] 2009. [Citado el: 20 de Diciembre de 2009.] [Disponible en: <http://www.frt.utn.edu.ar/sistemas/paradigmas/page22.html>].
- 21. Historia de la Informática.** [En línea] Entorno Virtual de Aprendizaje, 2008. [Citado el: 25 de Enero de 2010.] [Disponible en: http://teleformacion.uci.cu/mod/resource/view.php?id=9287&subdir=/Tema_1].
- 22. Dr. Manuel Ortíz Martín.** Universidad Autonoma de Puebla. [En línea] 2009. [Citado el: 23 de Diciembre de 2009.] [Disponible en: http://www.cs.buap.mx/~mmartin/notas/BD_CS2004_v3.pdf].
- 23. Jose Antonio Gallego Vázquez.** Desarrollo Web con PHP y MySQL. [En línea] s.l. : Ediciones Anaya, 2003. 85-415-1525-5. [Citado el 20 de Diciembre de 2009] [Disponible en: http://libros.mysofa.es/libro/desarrollo_web_con_php_y_mysql].
- 24. DotBlog.** Symfony: framework por excelencia para PHP. [En línea] 2009. [Citado el: 24 de Diciembre de 2009.] [Disponible en: <http://www.dotconsulting.com.ar/blog/?p=6>].
- 25. Febe Ánge Ricardo.** Informática 2009. [En línea] 2009. [Citado el: 17 de Diciembre de 2009.] [Disponible en: http://www.informaticahabana.com/evento_virtual/files/MUL053.pdf].
- 26. Oracle.** [Online] 2009. [Cited: Enero 25, 2010.] http://www.sun.com/emrkt/innercircle/newsletter/latam/0207latam_feature.html.
- 27. Ethan Cerami.** Web Services Essentials. [En línea] 2002. ISBN 0-596-00224-6 [Citado el 15 de Diciembre del 2009] [Disponible en: <http://www.quedelibros.com/libro/21665/Web-Services-Essentials.html>].
- 28. Diseño de páginas Web, Alojamiento, Posicionamiento y Dominios en Alicante, España. ¿Qué es un Sistema Operativo?** [En línea] 2009. [Citado el: 18 de Enero de 2010.] [Disponible en: <http://www.masadelante.com/faq-sistema-operativo.htm>].

- 29. Computer Audio Video Systems Integrator.** cavsi. [En línea] 2009. [Citado el: 3 de Diciembre de 2009.] [Disponible en: <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>].
- 30. Linux-es.** [En línea] 2009. [Citado el: 18 de Enero del 2010.] [Disponible en: <http://www.linux-es.org/distribuciones/>].
- 31. Antonio Perpiñan.** GNU/Linux Básicamente.[En línea] 2007. ISBN 88-99999-99 -9.[Citado el: 20 de Enero del 2010] [Disponible en: http://www.codigolibre.org/index.php?option=com_content&view=article&id=5195&Itemid=33].
- 32. Ubuntu.** [En línea] 2009. [Citado el: 21 de Enero de 2010.] [Disponible en: <http://www.ubuntu.com/>].
- 33. Ivar Jacobson, Grady Booch y James Rumbaugh.** El Proceso Unificado de Desarrollo de Software. Madrid : Pearson Educación, [En línea] 2000. ISBN 84-7829-036-2.[Citado en 23 de Enero del 2010] [Disponible en: <http://www.libros-pdf.com.ar/%E2%80%9Ccel-proceso-unificado-de-desarrollo-de-software%E2%80%9D-pdf-100.html>].
- 34. Sparx Systems.** [En línea] 2007. [Citado el: 30 de Enero de 2010.] [Disponible en: http://www.sparxsystems.com.ar/resources/tutorial/uml2_componentdiagram.html].
- 35. J. Kabir Mohammed.** Servidor Apache 2. 2003. ISBN 9788441514683.[En línea] [Citado el 17 de Enero de 2010]. [Disponible en: <http://xbash.wordpress.com/2010/02/17/la-biblia-del-servidor-apache-mohammed-j-kabir/>].
- 36. Mastermagazine.** Definición de Apache. [En línea] 2009. [Citado el: 16 de Enero de 2010.] [Disponible en: <http://www.mastermagazine.info/termino/3866.php>].

BIBLIOGRAFÍAS

- 1. Fabien Potencier y François Zaninotto.** Symfony 1.2, la guía definitiva. [Online] editorial Apress (ISBN-13: 978-1590597866).
- 2. Mario Ramos Monso.** Programación PHP. Sitios Web dinámicos e interactivos (Español). [Online] MP Ediciones, 2004, 424; ISBN 987-526-202-1.
- 3. Rolando Alfredo Hernández León y Zayda Coello González.** El Paradigma Cuantitativo de la Investigación Científica. [Online] Ciudad de la Habana : EDUNTIV, 2002. ISBN 978-959-16-0343-2.
- 4. LARMAN CRAIG.** UML Y Patrones. Introducción Analisis y Diseño Orientado a Objetos. [Online] Prentice Hall Hispanoamericana. 978-970-17-0261-1.
- 5. McGraw - J.Frentzen Hill.** Superutilidades para JavaScript (Español). [Online] 1999, 549 p. ISBN 84-481-2124-4.
- 6. Cristian Darie.** AJAX and PHP. Building Responsive Web Applications (Inglés). [Online] Packt Publishing, 2006, 286 p. ISBN 1-904811-82-5. .
- 7. Departamento de Técnicas de Programación.** Entorno Virtual de Aprendizaje. Introducción a la Programación cliente - servidor. [En línea]
- 8. Ext JS.** [Online] 2006-2010 Ext JS, Inc.
- 9. Codigolinea.** Estilo de programación y convención de nombres. [Online] 2009.
- 10. Roger Pressman.** Ingeniería de software. Un enfoque práctico. [Online] McGraw-Hill. ISBN 84-481-3214-9.
- 11. Favio E. Miranda Perea.** Lenguajes de Programación y sus Paradigmas. [Online] 2008.
- 12. Departamento de Sistemas y Computación del Instituto Tecnológico La Paz.** Programación estructurada. [Online] 2009.
- 13. Equipo de desarrollo de PostgreSQL,** Manual de Usuario de PostgreSQL. [Online] 2009.

GLOSARIO DE TÉRMINOS

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

Radio: Tecnología que permite la transmisión de señales sonoras mediante la modulación de ondas electromagnéticas.

TIC: Tecnologías de la información y las comunicaciones.

UCI: Universidad de las Ciencias Informáticas.

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

Web: Servidor de información www. Se utiliza también para definir el universo www en su conjunto.

WWW (World Wide Web): Sistema de información distribuido con mecanismos de hipertexto. Es el universo de servidores HTTP, que permiten mezclar texto, gráficos y archivos de sonido juntos.

HTTP (Hyper Text Transfer Protocol): Es el protocolo usado en cada transacción de la World Wide Web.

CSS (Cascading Style Sheets): En español Hojas de Estilo en Cascada, ofrece la posibilidad de separar la estructura de un documento estructurado escrito en HTML o XML de su presentación.

DOM (Document Object Model): Provee un conjunto estándar de objetos para representar documentos HTML y XML, permitiendo su combinación y ofreciendo una interfaz estándar para el acceso y manipulación.

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

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

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

UML (Unified Modeling Language): Notación estándar para modelar objetos del mundo.

Herramientas CASE: Herramientas utilizadas para el desarrollo de proyectos de Ingeniería.

POO (Programación Orientada a Objeto): Es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas de ordenador

XML (Extensible Markup Language): Conocido como lenguajes de marcas. Es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C), no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

HTML: Lenguaje de marcado predominante para la construcción de páginas Web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

JavaScript: Lenguaje de programación interpretado, utilizado principalmente en páginas Web.

Navegador: Aplicación capaz de interpretar las órdenes recibidas en forma de código HTML fundamentalmente y convertirlas en las páginas que son el resultado de dicha orden, permitiendo así la comunicación con el servidor.

MVC (Modelo – Vista – Controlador): Es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

SQL: Structured Query Language (lenguaje de consulta estructurado), es un lenguaje computacional de Bases de Datos diseñado para la recuperación y gestión en sistemas de bases de datos relacionales.

Apache: Es un servidor web HTTP de código abierto para plataformas Unix.

GNU/Linux: Es uno de los términos empleados para referirse a la combinación del núcleo o kernel libre similar a Unix denominado Linux, que es usado con herramientas de sistema GNU.

Selenium: Es un entorno de desarrollo integrado para los scripts de Selenium que permite grabar fácilmente, rápidamente y reproducir las pruebas en el entorno real de que se utilice.