

Implementación de los Subsistemas Web y Transferencia de la Plataforma de Transmisión Abierta para Radio y Televisión

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

Autor:

Dunier Domínguez Mora

Tutor:

Ing. Yandy León Núñez

Ciudad de La Habana, junio de 2009

“Año del 50 aniversario del triunfo de la Revolución”



*“...si tuviereis fe como un grano de mostaza,
...nada os será imposible.”*

Mateo 17:20

A mi buen Jesús...

A mi hermosa Denisse...

A mis padres, abuelos y familia...

A todos los que estuvieron y están a mi lado...

AGRADECIMIENTOS

- ✓ A mi Dios por ser guía e inspiración de mi vida, por estar a mi lado cuando todos se han ido, por su fidelidad que nunca me ha dejado y su amor inigualable.
- ✓ A la Revolución y a la Universidad de las Ciencias Informáticas por la formación recibida y por brindar lo necesario para convertirme en un profesional.
- ✓ A mi tutor el Ingeniero Yandy León Núñez por creer siempre que se podía lograr este trabajo y por todas esas noches en las que no dormimos. Gracias mi hermano por tu ayuda y sacrificio, sin ti hubiera sido muy difícil este sueño.
- ✓ A mi novia Denisse, una mujer extraordinaria que llegó a mi vida en el tiempo indicado y que con su amor me ha ayudado a esforzarme y a seguir adelante. Gracias por todo tu apoyo mi pequeña.
- ✓ A mis padres Mireya y Ángel, los mejores padres del mundo. Nada iguala su dedicación y ayuda incondicional para que llegara a ser quien soy hoy. Gracias por sus consejos y enseñanzas.
- ✓ A mis abuelos Emma y Antonio que tanto amor me han dado. Ustedes son muy especiales para mí.
- ✓ A mi hermanito que también se ha preocupado por mí y que sé en lo profundo de su corazón quisiera hacer mucho más. Te quiero y oro a Dios por tu vida.
- ✓ A la Gallega y a esa familia hermosa por acogerme como un hijo. Gracias por sustentarme y brindarme su apoyo cuando más lo necesitaba.
- ✓ A dos personas que aunque se encuentren lejos físicamente, siempre han estado presentes y en los momentos más duros me han brindado excelentes consejos y la mejor ayuda, ellas son: Aylín León y Kenya Magdalena.
- ✓ A mis hermanos en Cristo por su apoyo incondicional y sus palabras de aliento. Todas las páginas de este trabajo no alcanzarían para nombrarlos.
- ✓ A los muchachos de UCITeVe por ser como una gran familia en la cual siempre se sonreía cada día.
- ✓ A mis compañeros y amigos, a todos los que compartimos sonrisas y tristezas. Gracias por su ayuda. Me emociona verlos hoy como profesionales.
- ✓ A cada profesor que durante estos 18 años de estudios ha sembrando lo que hoy puedo cosechar.

Muchas gracias...

OPINIONES Y AVALES

Se otorga el presente

Reconocimiento

A: Donier Dominguez Mora

Por haber resultado:

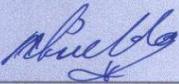
Relevante

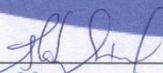
Con el Trabajo:

Plataforma de Transmisión Abierta
para Radio y Televisión

**5ta Jornada Científica
Facultad 9**

Dado a los 6 días del mes de mayo del 2009


Manuel E. Puebla Martínez
Vice Decano Producción


Frank E. Hernandez Sanchez
Presidente FEU Fac 9

Reconocimiento otorgado por la Facultad 9 de la UCI

OPINIONES Y AVALES



Reconocimiento otorgado por el Polo de Video y Sonido Digital de la UCI

OPINIONES Y AVALES

Universidad de las Ciencias
Informáticas
UCI

Universidad de las Ciencias Informáticas
Federación Estudiantil Universitaria

Otorga el presente:
Reconocimiento

*A: Plataforma de Transmisión Abierta
para la Radio y la Televisión.*

Por haber obtenido:
Destacado

"El futuro de Cuba debe ser necesariamente un futuro de hombres de ciencia..."
Fidel Castro

Dado a los 22 días del mes de mayo de 2009,
"Año del 50 Aniversario del Triunfo de la Revolución"

Alexander Rodríguez Bonet
Secretariado FEU-UCI

José Lavandero García
Vicerrador de Prog Especiales

7ma Jornada Científica Estudiantil

Reconocimiento otorgado por la Universidad de las Ciencias Informáticas

OPINIÓN DEL TUTOR SOBRE EL TRABAJO DE DIPLOMA

La Plataforma de Transmisión Abierta para Radio y Televisión pretende ser un sistema que unifique los servicios que hasta el momento se ofrecen en la Universidad de las Ciencias Informáticas, brindando soluciones mucho más eficientes y genéricas que puedan ser empleadas a su vez en cualquier entidad que las requiera. Dentro de esta plataforma, se identifican los subsistemas Web y Transferencia encargados entre otras cosas de dos funciones fundamentales para la misma:

1. Gestionar los procesos de visualización y publicación de contenidos para su posterior utilización por los clientes.
2. Conversión automática de los materiales a publicar y la organización física en los servidores que se encargarán de distribuirlos por la red.

Todo lo anterior basado en la premisa del uso de estándares abiertos, utilizando los avances de la arquitectura Cliente - Servidor, lo cual constituye una propuesta novedosa y de gran aporte para la Universidad y el país.

El trabajo realizado por el ponente es de vital importancia para el completamiento de los objetivos que desea alcanzar la plataforma y cumple de manera meritoria los requisitos necesarios y suficientes para una primera versión de la misma.

El documento de tesis presenta una excelente redacción, ortografía y claridad en las ideas expuestas, dando muestras de la preparación del estudiante en los temas tratados en cada uno de sus epígrafes. El cronograma de tareas trazado para la realización del mismo fue cumplido a totalidad y con un nivel de excelencia elevado, a pesar de la complejidad y desconocimiento de algunos temas, así como del escaso tiempo con que se contaba.

Durante el proceso de desarrollo del producto, el diplomante ha demostrado de manera inobjetable las siguientes características:

1. Alta responsabilidad y seriedad en el cumplimiento de las tareas indicadas por el tutor, la dirección del proyecto y del polo al que pertenece.

2. Gran sentido de pertenencia, respeto y de amor hacia el trabajo que se realiza.
3. Amplios conocimientos en el contenido estudiado, así como una gran capacidad de asimilación de nuevos temas, característica fundamental de un buen ingeniero de estos tiempos.
4. Creatividad en el desarrollo de nuevas alternativas a los problemas expuestos.
5. Disciplina, respeto, honestidad, consagración y sacrificio como características que lo han guiado durante todo el proceso.

Se recomienda el trabajo para ser presentado en otros eventos de carácter científico así como su continuidad y optimización para futuras versiones del proyecto.

Por todo lo expuesto anteriormente y considerando que se le dio cumplimiento a las tareas previstas, el tutor del presente trabajo de diploma propone la merecida calificación de 5 pts.

Tutor: Ing. Yandy León Nuñez

Firma

Fecha

ABSTRACT

The quick advance of communications and the development of compute networks have brought about considerable changes in the way of transmitting data. Radio and television have also experienced transformations due to the aforementioned event, for an increase in the ending and receiving of multimedia contents by means of computers is evidenced, providing more interactive and reliable services.

In this work we can find the bases of the implementation of Web and Transference of Radio and Television Open Transmission Platform subsystems. Such platform has as main goal the distribution of audiovisual contents making use of the signal transmission subsystems and streaming technology. The development Web and Transference subsystems emerges with the existence of the software design for these applications, but not with their way of programming.

This document comprehends the results of the investigation carried out in the study and analysis of technologies, programming techniques as well as the necessary tools for the development of the subsystems. The functionalities of the applications are described through the implementation model and the Components Diagram. After programming these applications its main functionalities are tested and aspects of great importance are recommended for the implementation of future versions.

KEYWORDS

Audio, Conversion, Platform, Streaming, Television, Transference, Transmission, Video, Web.

RESUMEN

El avance vertiginoso de las comunicaciones y el desarrollo de las redes computacionales han traído consigo cambios apreciables en las formas de transmisión de datos. La radio y la televisión también han experimentado transformaciones debido al suceso antes expuesto, pues se evidencia un aumento del envío y recepción de contenido multimedia haciendo uso de ordenadores, brindándose servicios más fiables e interactivos.

En el presente trabajo se muestran las bases de la implementación de los subsistemas Web y Transferencia de la Plataforma de Transmisión Abierta para Radio y Televisión. Dicha plataforma tiene como principal objetivo la distribución de contenidos audiovisuales haciendo uso de los sistemas de transmisión de señales y empleándose la tecnología streaming. El desarrollo de los subsistemas Web y Transferencia surge con la realidad de que existía el diseño de software de estas aplicaciones pero no la programación de los mismos.

Este documento recoge los resultados de la investigación realizada en el estudio y análisis de tecnologías, técnicas de programación y herramientas necesarias para el desarrollo de los subsistemas. Se describen las funcionalidades de las aplicaciones mostrándose a través del modelo de implementación y de los Diagramas de componentes. Luego de la programación de las aplicaciones se prueban sus principales funcionalidades y se recomiendan puntos de gran importancia para la implementación de versiones posteriores.

PALABRAS CLAVES

Audio, Conversión, Plataforma, Streaming, Televisión, Transferencia, Transmisión, Video, Web.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1.....	4
1.1 Introducción.....	4
1.2 Tendencias y tecnologías actuales.....	4
1.2.1 Modelo Cliente - Servidor.....	5
1.2.2 Aplicaciones Web.....	6
1.2.3 AJAX.....	8
1.2.3 Streaming de video	11
1.3 Paradigmas de la programación	13
1.3.1 Programación imperativa	14
1.3.2 Programación lógica	14
1.3.3 Programación funcional.....	15
1.3.4 Programación estructurada	15
1.3.5 Programación orientada a objetos.....	16
1.4 Lenguajes de programación	18
1.4.1 Lenguajes de programación del lado del cliente.....	19
1.4.2 Lenguajes de programación del lado del servidor	20
1.5 Entorno integrado de desarrollo.....	21
1.5.1 Eclipse PDT	21
1.6 Patrón arquitectónico.....	22
1.6.1 Modelo en capas.....	22
1.6.2 Modelo Vista Controlador.....	24
1.6.3 Fundamentación del patrón arquitectónico seleccionado	25
1.7 Framework	25
1.7.1 Symfony.....	26
1.8 Sistema de Gestión de Bases de Datos	27
1.8.1 PostgreSQL	28

1.9 Metodología de desarrollo de software	28
1.9.1 Proceso Unificado de Desarrollo (RUP)	29
1.10 Lenguaje Unificado de Modelado (UML)	30
1.11 Herramientas CASE para la modelación UML	31
1.11.1 Rational Rose Enterprise.....	31
1.11.2 Visual Paradigm	32
1.11.3 Fundamentación de la herramienta CASE seleccionada	32
1.12 Sistemas de streaming	33
1.12.1 FFmpeg y FFServer	33
1.12.2 VideoLAN.....	34
1.12.3 Darwin Streaming Server	35
1.12.4 MPEG4IP	36
1.12.5 Mp4live.....	36
1.12.6 Fundamentación de las aplicaciones de streaming seleccionadas	37
1.13 Servidor Web.....	38
1.13.1 Apache.....	38
1.14 Sistema operativo.....	39
1.14.1 UBUNTU Linux.....	40
1.15 Conclusiones.....	41
CAPÍTULO 2.....	42
2.1 Introducción.....	42
2.2 Descripción de los subsistemas.....	42
2.2.1 Subsistema Web	42
2.2.2 Subsistema de Transferencia.....	44
2.3 Modelo de implementación.....	45
2.4 Estándares de codificación.....	56
2.5 Conclusiones.....	58
CAPÍTULO 3.....	59
3.1 Introducción.....	59

3.2 Pruebas unitarias y funcionales	59
3.3 Automatización de pruebas	61
3.4 Herramientas para pruebas	61
3.5 Diseño de pruebas	64
3.6 Conclusiones.....	70
CONCLUSIONES	71
RECOMENDACIONES	72
REFERENCIAS BIBLIOGRÁFICAS	73
BIBLIOGRAFÍAS.....	76
ANEXOS.....	78
GLOSARIO	80

ÍNDICE DE FIGURAS

Figura 1	Modelo Cliente - Servidor.....	5
Figura 2	Esquema de AJAX.....	9
Figura 3	Uso de frameworks AJAX.....	9
Figura 4	Definición de streaming.....	11
Figura 5	Transferencia clásica de un fichero.....	12
Figura 6	Representación del funcionamiento de un sistema con arquitectura de tres capas.....	23
Figura 7	Representación del patrón arquitectónico Modelo Vista Controlador.....	24
Figura 8	Uso de los principales frameworks de php en el año 2007.....	26
Figura 9	Logo del Proceso Unificado de Desarrollo.....	29
Figura 10	Interfaz del Darwin Streaming Server.....	35
Figura 11	Interfaz del Mp4live.....	37
Figura 12	Logo del servidor Apache.....	39
Figura 13	Crecimiento del uso de la distribución Ubuntu.....	40
Figura 14	Modelo de implementación.....	46
Figura 15	Paquete de módulos del subsistema Web.....	46
Figura 16	Paquete de módulos del subsistema Transferencia.....	47
Figura 17	Diagrama de componentes del módulo Noticias Web.....	48
Figura 18	Diagrama de componentes del módulo Publicación Web.....	49
Figura 19	Diagrama de componentes del módulo Sección Web.....	50
Figura 20	Diagrama de componentes del módulo Cartelera.....	51
Figura 21	Diagrama de componentes del módulo Estadísticas.....	52
Figura 22	Diagrama de componentes del módulo Encuestas.....	53
Figura 23	Diagrama de componentes del módulo Foro.....	54
Figura 24	Diagrama de componentes del módulo Conversión.....	55
Figura 25	Código indentado.....	57
Figura 26	Fragmento de caso de prueba desarrollado con Selenium.....	62
Figura 27	Fragmento del TestSuite.html.....	63
Figura 28	Interfaz gráfica del plugin Selenium.....	64

Figura 29	Resultados del caso de prueba #1	66
Figura 30	Resultados del caso de prueba #2	67
Figura 31	Resultados del caso de prueba #3	68
Figura 32	Resultados del caso de prueba #4	69
Figura 33	Resultados del caso de prueba #5	70
Figura 34	La Web antes de AJAX	78
Figura 35	La Web con AJAX	78
Figura 36	Principales funcionalidades de los framework para php	79
Figura 37	Proceso Unificado de Desarrollo de Software (RUP)	79

INTRODUCCIÓN

En la actualidad hacer uso de los medios de comunicación puede resultar cotidiano y sencillo, pero en realidad no es un proceso simple el que existe detrás de cada transmisión. Las formas del hombre comunicarse han transcurrido por etapas como las escrituras jeroglíficas, la creación del papel y luego de la imprenta, hasta finalmente llegar al surgimiento de la radio, el cine y la televisión.

Durante la década de 1950 la nación cubana ocupó el tercer puesto en la realización de transmisiones televisivas en Latinoamérica convirtiéndose en pionera de este campo junto a Brasil y México. Fue además el segundo país en el mundo después de Estados Unidos en transmitir emisiones en colores. Desde el triunfo de la Revolución se ha procurado que estos medios de comunicación sean destinados a la formación cultural y cognoscitiva de la sociedad. El desarrollo acelerado que experimenta el mundo ha provocado grandes cambios en la infraestructura de transmisión de los medios de comunicación. El surgimiento de la informática, las redes de computadoras y de Internet han devenido en un crecimiento de su uso para la comunicación y como reemplazo a viejas técnicas analógicas de transmisión radial y televisiva.

La Universidad de las Ciencias Informáticas (UCI), nació como proyecto para la informatización de la sociedad cubana y la formación de profesionales altamente calificados en el desarrollo de soluciones de software. La UCI constituye un elemento fundamental para el desarrollo y puesta en marcha de sistemas que mejoren los procesos de comunicación.

La Facultad 9 de la UCI desde sus inicios contó con el desarrollo de proyectos orientados a servicios de televisión como UCITeVe. Con el transcurso de los años y la creación del Polo de Video y Sonido Digital en dicha Facultad, se han detectado una serie de procesos que demandan la automatización inmediata, y otros que aún estando automatizados, requieren de una mejora con el objetivo de hacer más eficientes las transmisiones. Ejemplo de lo anterior lo es el sistema televisivo de la UCI el cual consta con 24 canales y una emisora radial cuyo soporte de transmisión es un canal. Cada uno de estos canales se sustenta sobre un ordenador que reproduce un fichero de video o audio codificado en un formato determinado que puede ser AVI XviD, MOV, WMV, WMA, MP3, entre otros.

Mediante la red los usuarios de la UCI tienen acceso a diferentes sitios Web. Uno de los más visitados es Inter-Nos el cual brinda servicios de video a demanda como películas, series, programas grabados, entre otros. También brinda acceso a streaming de video de cuatro canales de Televisión Nacional. Pero tanto para la transmisión de los canales, como para la publicación en la Web de un material de audio o video, se necesita que dicho fichero esté codificado en el formato requerido.

En la actualidad existen sistemas propietarios muy profesionales para la codificación de medias, ejemplo de ello lo es el TMPGEnc 4.0 XPress, Amadis Video Converter Suite y el WinAVI Video Converter, por solo citar algunos. No es un secreto las limitaciones que representan para el país el uso de software propietario en procesos vitales como es precisamente la televisión y la radio.

La codificación del fichero y su transferencia al lugar de transmisión, es un proceso que requiere de una persona que utilice algún sistema de los antes mencionados, y luego realice la mayor parte del trabajo de forma manual, lo cual provoca errores humanos que se traducen en horas perdidas.

Aunque el sitio Inter-Nos permite la visualización de contenido multimedia, es un sistema desarrollado sobre plataformas propietarias y no integra todos los servicios televisivos que los usuarios requieren. La lógica de su implementación es analizada en la creación del subsistema Web, mientras que para el subsistema de Transferencia lo es la aplicación Qeven, la cual ofrece una solución gráfica al codificador de video Mencoder. Qeven no se ajusta a todos los requerimientos de codificación y transferencia de la Plataforma de Transmisión Abierta para Radio y Televisión. Esta solución gráfica es una aplicación de escritorio lo cual contrasta con la propuesta Web abordada en el presente trabajo. En el subsistema Transferencia se hará uso de las funcionalidades del codificador de audio y video FFmpeg.

Las razones antes expuestas han permitido identificar la necesidad de la implementación de los subsistemas Web y Transferencia, que permitan la gestión del contenido multimedia hacia el servidor de publicación, facilitando su visualización y la interacción del usuario, todo dentro de un sistema que unifique los servicios de transmisión radial, televisiva, de streaming y video a demanda. Por tanto, se define para la investigación el siguiente problema científico:

Inexistencia de la implementación de los subsistemas Web y Transferencia para gestionar el contenido multimedia hacia el servidor de publicación y permitir su visualización e interacción con el usuario en la Plataforma de Transmisión Abierta para Radio y Televisión (PTARTV).

Teniendo como base el problema planteado, se posee la perspectiva de construir subsistemas que permitan la transferencia y visualización de contenido multimedia en el sistema PTARTV. Lo anterior constituye un aporte práctico para la UCI, por lo cual se defiende la siguiente idea:

La implementación de subsistemas automatizados para la transferencia y visualización de contenido multimedia en la PTARTV, garantizará mayor rapidez y mejor funcionamiento en los procesos de transferencia, codificación y visualización de la información manejada.

En este trabajo es objeto de estudio la PTARTV y el campo de acción se centra en la automatización de los subsistemas Web y Transferencia.

Se propone como objetivo general implementar un prototipo funcional de los subsistemas Web y Transferencia del sistema PTARTV.

Para dar cumplimiento al objetivo propuesto se definen las siguientes tareas de investigación:

1. Valorar tendencias y tecnologías actuales.
2. Definir entornos de desarrollo integrado, frameworks y lenguajes de programación.
3. Seleccionar las herramientas para el trabajo con video a demanda y streaming en software libre.
4. Construir el modelo de implementación.
5. Implementar prototipos funcionales de los subsistemas Web y Transferencia.
6. Desarrollar los casos de prueba que certifiquen la veracidad de los algoritmos empleados.
7. Documentar los resultados de las pruebas unitarias y funcionales de los subsistemas.

Para obtener los conocimientos que posibiliten el cumplimiento del objetivo trazado, se lleva a cabo una investigación utilizándose algunos de los métodos científicos existentes, tanto teóricos como empíricos.

- *Analítico-Sintético*: Permite dividir la investigación en partes para su desarrollo y luego logra la unión a través de las relaciones existentes.
- *Histórico-Lógico*: Ayuda a caracterizar las soluciones existentes entendiendo su evolución y facilitando aprovechar puntos en común y conceptos teóricos que sean de relevancia.
- *Modelación*: Ofrece la posibilidad de crear abstracciones para explicar la realidad. Se hace visible en el trabajo al crear modelos como el de implementación.
- *Observación*: Facilita conocer el panorama real de una situación mediante la percepción directa.

CAPÍTULO 1

Fundamentación Teórica

1.1 Introducción

La implementación de los subsistemas Web y Transferencia dentro de la Plataforma de Transmisión Abierta para Radio y Televisión tiene como principal sustento el diseño realizado de dicha plataforma. Es necesario tener presente que se habla de subsistemas muy diferentes en cuanto a funcionalidad pero muy relacionados en su entorno. Resulta evidente que un diseño realizado hace ya algún tiempo sufra variaciones, pues en el campo de la informática ocurren muchas transformaciones en poco tiempo. Por tal razón se dejará siempre claro dónde se realizan modificaciones y cuál selección de tecnología resulta diferente a la propuesta anteriormente.

En el presente capítulo se definen una serie de conceptos necesarios para comprender el dominio del problema. Son abordadas las tendencias tecnológicas, técnicas y lenguajes de programación, metodología de desarrollo y herramientas. También se ofrece un análisis valorativo de los soportes utilizados para la implementación de los subsistemas.

1.2 Tendencias y tecnologías actuales

En medio de la revolución informática que ha vivido el mundo, los sistemas para computadoras se han orientado en gran escala al desarrollo de Aplicaciones Web. Teniendo en cuenta las funcionalidades que brindan tecnologías como: Cliente - Servidor, AJAX (Asynchronous JavaScript And XML), Streaming de audio y video, y la evolución cada vez mayor del tratamiento de contenido multimedia, resulta importante el uso de cada una de ellas para la creación de software orientado a la radio y la televisión.

Estas tecnologías están muy vinculadas a la solución de implementación de los subsistemas Web y Transferencia. Seguidamente se realiza el análisis valorativo de su selección y las ventajas que brindan para las aplicaciones a desarrollar.

1.2.1 Modelo Cliente - Servidor

La International Business Machines Corporation (IBM) define el modelo Cliente - Servidor como: "La 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 servicio hechos por estaciones de trabajo inteligentes o clientes, resultan en un trabajo realizado por otros computadores llamados servidores".[1]

El cliente es el que inicia un requerimiento de servicio. La ubicación de los datos o de las aplicaciones es indiferente. Mientras que por su parte el servidor puede ser cualquier recurso de cómputo cuya función sea responder al requerimiento del cliente. (**Figura 1**)

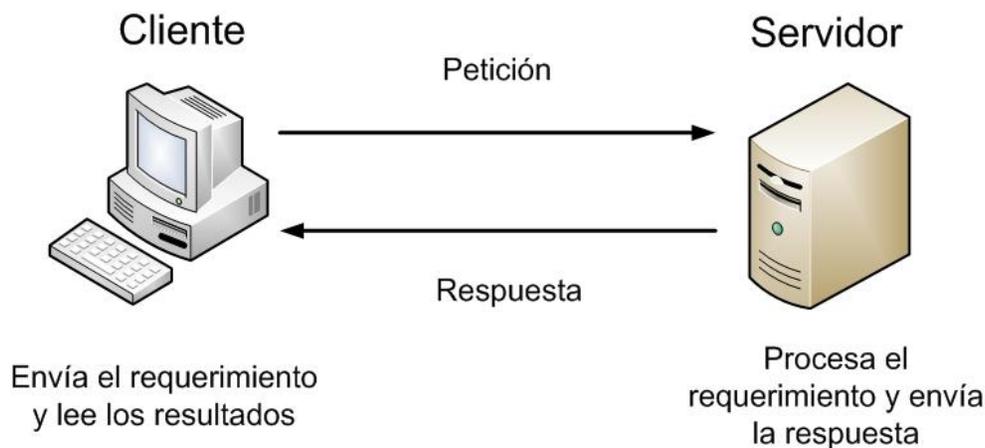


Figura 1. Modelo Cliente - Servidor

Entre las características del modelo Cliente - Servidor se pueden citar las siguientes:

- El cliente y el servidor pueden funcionar en una misma entidad o en entidades separadas.
- Para el cliente es indiferente la lógica del servidor, él solo necesita acceder a su interfaz externa.
- La ubicación física, el hardware que posee el equipo servidor y el sistema operativo que corre sobre él, no ofrecen problemas de comunicación para el cliente.
- Si ocurrieran modificaciones en el servidor estas afectarían en ninguna o poca escala al cliente.
- Un servidor ofrece servicios múltiples de forma simultánea.

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.

El uso del modelo Cliente - Servidor implica el manejo de errores y la implementación de estrategias para garantizar la integridad de los datos. Problemas en las redes de telecomunicación o en la transferencia de los datos pueden producir un mal funcionamiento del sistema.

1.2.2 Aplicaciones Web

El uso de los sistemas desktop o de escritorio trajo consigo que la información y la lógica se encontraran esparcidas en cada ordenador que las utilizara. Se originara duplicidad de datos si no existía unificación. Actualizar o corregir un sistema resultaba realmente complejo pues la modificación implicaba a cada ordenador. Estas situaciones propiciaron una nueva forma de concebir los sistemas.

En 1989 se dan los primeros pasos en la Web, lo cual se logra trabajando en la creación de una pequeña plataforma para la lectura de documentos científicos en el CERN (Centro Europeo de Investigación Nuclear). Los principios tecnológicos de esta plataforma incluyeron la creación de tres componentes: HTML (*HyperText Markup Language*), HTTP (*HyperText Transfer Protocol*) y URI (Universal Resource Identifier). El HTML era un lenguaje para escribir los documentos, y con el que se podían establecer los enlaces a otros documentos. HTTP, en cambio, era un protocolo para la transmisión de los documentos en formato HTML entre el servidor y el cliente. Y los URI eran los identificadores de los documentos que servían para establecer los enlaces.

Las aplicaciones Web tienen como arquitectura general la de un sistema Cliente - Servidor, donde tanto el navegador como el servidor (el servidor Web), y el protocolo mediante el cual se comunican son estándar, y no han de ser creados por el desarrollador. [2]

La parte del cliente de las aplicaciones Web está formada por el código HTML que forma la página Web, con opción a código ejecutable mediante los lenguajes script de los navegadores (*JavaScript*, *VBScript*, *PerlScript*) o mediante pequeños programas (*applets*) en Java. [3]

CAPÍTULO 1: Fundamentación Teórica

La parte del servidor está formada por un programa o script que es ejecutado por el servidor Web, y cuya salida se envía al navegador del cliente.

Ejemplos de aplicaciones Web:

- Señal ACN (Sistema de teletexto)
- Flickr (Administrador de fotografías)
- Gmail (Correo electrónico)
- Ebay (Catálogo y tienda electrónica)
- Bouchard Translator (Traductor en línea)

El auge de las aplicaciones Web se ve favorecido por las innegables ventajas que brindan. Pueden citarse algunas como:

- **Multiplataforma:** Las aplicaciones pueden ser utilizadas a través de variadas plataformas, tanto de hardware como de software apoyado en gran medida por el uso de estándares.
- **Actualización instantánea:** Dada la razón que los usuarios de la aplicación hacen uso de un sólo programa que interactúa directamente con el servidor, siempre utilizarán la versión más actualizada del sistema.
- **Acceso no fijo:** El usuario puede acceder a la aplicación usando cualquier equipo provisto de red y navegador, estando ubicado en cualquier lugar y suponiendo siempre que cuenta con los permisos de acceso necesarios.
- **Fácil integración:** Dado su basamento en protocolos estándares, la información manejada por el sistema puede ser accedida con mayor facilidad por otros sistemas.

El sistema PTARTV tendrá un mejor funcionamiento y brindará servicios optimizados al ser implementado como una aplicación Web. Para la automatización de los procesos televisivos que abarcan los subsistemas Web y Transferencia, las aplicaciones del tipo Cliente - Servidor son la solución más factible.

1.2.3 AJAX

Con el surgimiento de AJAX, las aplicaciones Web han pasado de versiones estáticas y ligeras de aplicaciones “reales” a aplicaciones muy completas. Ahora ofrecen elementos de interfaz más sofisticados y transferencia de datos asincrónica, las aplicaciones orientadas a la Web se aproximan mucho más a sus equivalentes de escritorio en términos de interfaz de usuario y de velocidad. Además, van más allá del navegador llegando a tiempos de ejecución independientes de la plataforma, smartphones y dispositivos móviles... Incluso en modo desconectado [4]

AJAX no es un programa, no es un archivo, no es un servidor, no es un módulo de apache, no es un script y no es una extensión de PHP (*PHP Hypertext Pre-processor*).

¿Qué es AJAX?

El término aparece en febrero de 2005 en el artículo “AJAX: A New Approach to Web Applications” publicado por Jesse James Garrett. Según el autor “AJAX no es una tecnología en sí mismo. En realidad, se trata de la unión de varias tecnologías que se desarrollan de manera autónoma y que se unen de formas nuevas y sorprendentes” [5]

Características de AJAX:

- Presentación basada en estándares XHTML (*Extensible Hypertext Markup Language*) y CSS (*Cascading Style Sheets*).
- Visualización e interacción dinámica utilizando el DOM (*Document Object Model*).
- Intercambio y manipulación de datos usando XML (*Extensible Markup Language*) y XSLT (*XSL Transformations*).
- Recuperación de datos asíncrona mediante XMLHttpRequest (*Extensible Markup Language / Hypertext Transfer Protocol*).
- JavaScript para enlazarlo todo.

Si se hiciera un esquema de AJAX podría quedar similar a lo mostrado por la **figura 2**.

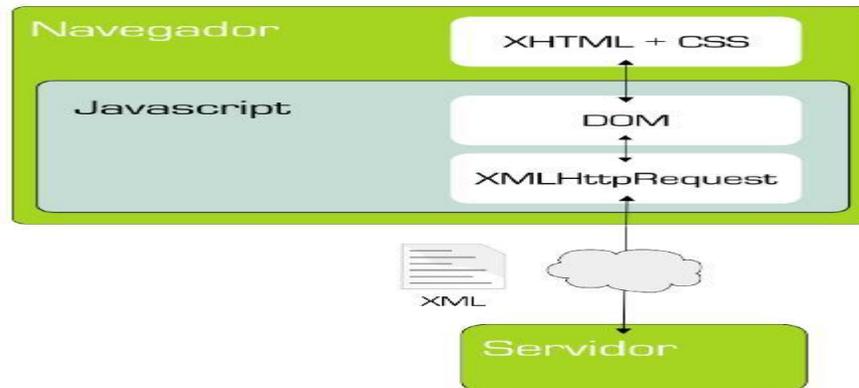


Figura 2. Esquema de AJAX

Indudablemente la Web ha tenido un antes (ver Anexo I) y un después (ver Anexo II) del uso de AJAX para la creación de Aplicaciones Ricas de Internet (RIAS, siglas en inglés).

Aunque existen diferentes frameworks para el uso de AJAX cada uno tiene sus peculiaridades. Al realizar una selección es importante tener en cuenta el sistema que se desarrollará, la complejidad del mismo y la habilidad de los desarrolladores. En la **figura 3** se muestra el porcentaje de uso de los diferentes frameworks.

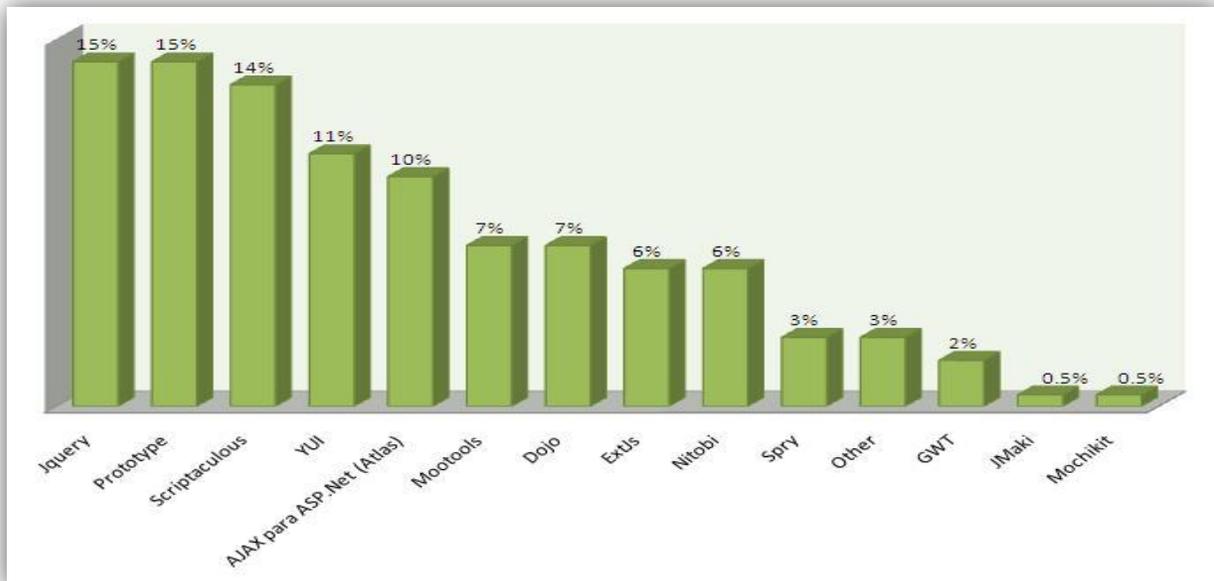


Figura 3. Uso de frameworks AJAX

Ext JS es un potente framework para el desarrollo de AJAX. Puede resultar un poco más complejo pero lo anterior se ve opacado por sus bondades:

- Modelo de Componentes.
- Modelo de Contenedores.
- Capas.
- Grid.
- PlantillasX.
- Vistas de Datos.
- Sirve de puente entre las librerías JS más usadas (Prototype, JQuery, YUI). Debido a que se inició como una extensión de YUI, esta presenta una cierta ventaja de compatibilidad respecto a las otras dos.

Soporta los navegadores:

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

Actualmente es usado por las siguientes compañías:

- Adobe
- Boeing Shared Services Group
- Borland
- Canon Canada, Inc.
- Cisco Systems, Inc.
- CNN
- IBM
- Siemens
- Sony Online Entertainment

Se puede afirmar que el uso del framework Ext JS de AJAX en el desarrollo de los subsistemas Web y Transferencia, permitirá que se logre simular una aplicación de escritorio que combine la interactividad necesitada, con los efectos visuales adicionales.

1.2.3 Streaming de video

“El streaming involucra el envío de información entre un servidor y un cliente a través de una red como Internet. El servidor fracciona la información en paquetes que se envían a través de la red. Los paquetes son ensamblados por el cliente, que reproduce la información recibida al mismo tiempo. El streaming se diferencia de una simple transferencia de ficheros en que el cliente reproduce la información mientras la está recibiendo en vez de esperar a recibirla completa antes de reproducirla. De hecho, un cliente de streaming podría no descargar el vídeo/audio, simplemente reproducir la información de cada uno de los paquetes e ir descartándolos.” [6]

En las **figuras 4 y 5** se muestran el streaming de video y la transferencia normal de ficheros respectivamente.



Figura 4. Definición de streaming.



Figura 5. Transferencia clásica de un fichero

Características de la Tecnología de Streaming:

- Se utiliza para sistemas multimedia distribuidos.
- Se fracciona la información para transmitirla.
- El envío de la información se realiza de forma temporizada.
- No es necesario que el cliente almacene toda la información que recibe.
- La reproducción puede comenzar instantes después del comienzo de la transmisión.

Algunos de los componentes que conforman esta tecnología son:

- **Servidor de Streaming:** Software encargado de atender las peticiones de los usuarios para acceder al contenido multimedia.

Funcionamiento del servidor:

- ✓ Cuando recibe una petición decide si la acepta. (Control de admisión)
- ✓ Establece una conexión con el cliente.
- ✓ Recibe las interacciones del cliente y actúa en la transmisión convenientemente.
 - Transmitiendo un flujo constante de información.
 - No transmitiendo.

- ✓ Finaliza la conexión cuando el cliente da por terminada la comunicación o cuando la información solicitada concluye.

Los servidores pueden brindar dos tipos de contenidos:

- ✓ **Bajo demanda o a la carta** (VOD, *siglas en inglés*): Permite al usuario hacer peticiones personalizadas de los ficheros almacenados en el servidor. Se realizan de manera individual, cada cliente que solicita una secuencia suele tener el control total de la misma pudiendo detenerla y reanudarla según su voluntad. Esto se debe a que se proporciona una sola ruta de acceso a los datos para cada cliente que solicita el contenido.
- ✓ **Difusión** (*broadcast*): Es un modo de transmisión de información donde un emisor envía información a una multitud de receptores de manera simultánea, sin necesidad de reproducir la misma transmisión por cada receptor. Es usada para transmitir video en vivo pues permite una audiencia de gran volumen. Los clientes que la reciben no pueden controlar el inicio del material, la velocidad de reproducción, ni rebobinar la secuencia. Es el servidor el que tiene el control de la secuencia.
- **Reproductor o Cliente**: Software cuyo propósito es la reproducción de los archivos suministrados por el servidor de Streaming. Es necesario que esté instalado en el ordenador para recibir las secuencias del servidor.

El streaming de audio y video se ha afianzado como la tecnología necesaria para la automatización de procesos de transmisión de radio y televisión, haciendo uso de redes de telecomunicación o cualquier otra infraestructura. Las ventajas y posibilidades que ofrece son esenciales para la solución del objetivo propuesto en el presente trabajo, acentuándose en la implementación del Subsistema Web. Pues este último debe cumplir con los requisitos de ofrecer la transmisión de televisión en vivo y el consumo de video a demanda.

1.3 Paradigmas de la programación

Los paradigmas de la programación pueden describirse como un conjunto de ideas que indican una forma de programar. A través de ellos se pueden obtener diversas formas de "ver" y "pensar" un

programa antes de escribirlo. Estas ideas han evolucionado a la par de los lenguajes de programación. A continuación se presenta una breve descripción de algunos de los paradigmas, haciendo énfasis en la Programación Orientada a Objetos (POO) como la solución para la implementación de los subsistemas Web y Transferencia.

1.3.1 Programación imperativa

La programación en este paradigma consiste en determinar qué datos son requeridos para el cálculo, asociar a esos datos unas direcciones de memoria, y efectuar paso a paso una secuencia de transformaciones en los datos almacenados, de forma que el estado final represente el resultado correcto.

En su forma pura este paradigma sólo soporta sentencias simples que modifican la memoria y efectúan bifurcaciones condicionales e incondicionales. Incluso cuando se añade una forma simple de abstracción procedimental, el modelo permanece básicamente sin cambiar. [7]

En la programación imperativa las funciones y datos se encuentran muchas veces entremezclados (código spaghetti). Resulta complejo razonar acerca del código. Se dificulta el manejo de múltiples instancias.

1.3.2 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.

1.3.3 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). [7]

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.

1.3.4 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.

Se obtiene un único bloque de programa, que cuando se hace demasiado grande puede resultar problemático su manejo. Detectar los errores no es tarea fácil. Aunque por mucho tiempo fue dominante como forma de concebir los programas para ordenadores fue superada con creces por la programación orientada a objetos.

1.3.5 Programación orientada a objetos

La POO, intenta simular el mundo real a través del significado de objetos que contienen las características y funciones. Se basa en la idea de un objeto, que es una combinación de variables locales y procedimientos llamados métodos que juntos conforman una entidad de programación. [8]

Clase: Es un tipo definido por el usuario que determina las estructuras de datos y las operaciones asociadas con ese tipo. Se puede especificar como una colección de objetos similares y un objeto constituye una instancia de la definición de una clase.

Encapsulación: Se usa para describir la combinación de estructuras de datos y de métodos que son manipulados por el objeto. La llamada a un objeto es lo que se denomina pasar un "aviso" a un objeto.

Abstracción: Es un término externo al objeto, que controla la forma en que es visto por los demás.

Modularidad: Un programa grande siempre será más complicado que la suma de varios programas pequeños, con lo que se considera ventajoso dividir un gran sistema en diversos módulos.

Jerarquía: Consiste en la clasificación y organización de las abstracciones según su naturaleza. El más claro ejemplo de jerarquía es la herencia.

Herencia: Se puede definir como una jerarquía de abstracciones y la relación entre clases, donde se comparte la estructura y el comportamiento de una o más clases consideradas como clases superiores.

Polimorfismo: Habilidad de llamar de la misma forma a dos métodos o procedimientos diferentes. El programa decide qué método utilizar según los parámetros que recibe. [9]

Un programa orientado a objetos es una colección de clases. Necesitará una función principal que cree objetos y comience la ejecución mediante la invocación de sus funciones miembro.

Esta organización conduce a separar partes diferentes de una aplicación en distintos archivos. La función principal también se pone en un archivo independiente. Se ensamblará el programa completo a partir de los archivos independientes en una única unidad.

Cuando se ejecuta ocurren tres acciones:

1. Se crean los objetos cuando se necesitan.
2. Los mensajes se envían desde los objetos y se reciben en otros.
3. Se borran los objetos cuando ya no son necesarios y se recupera la memoria ocupada por ellos.

1.3.5.1 Ventajas de la programación orientada a objetos

Aunque el paradigma de la POO no es infalible y tiene desventajas éstas son superadas por las posibilidades que brinda. Por tal razón serán éstas últimas el centro de atención.

- La POO constituye una evolución de otros paradigmas incorporando nuevos conceptos que le otorgan gran relevancia.
- Soporta sintáctica y semánticamente la unión entre los tipos abstractos de datos y sus operaciones (a esta unión se la suele llamar clase).
- Incorpora en su entorno de ejecución de mecanismos tales como el polimorfismo y el envío de mensajes entre objetos.
- 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.
- Relaciona el software con el mundo real y permite su evolución.
- Proporciona la creación de programas visuales y de prototipos.
- Facilita el mantenimiento de los sistemas y el trabajo en equipo.

1.4 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.

Si el lenguaje de programación no es código máquina entonces deberá ser compilado o interpretado.

La ejecución de un programa con compilador requiere de dos etapas:

- 1) Traducir el programa simbólico a código máquina
- 2) Ejecución y procesamiento de los datos.

Otros lenguajes de programación utilizan un programa intérprete o traductor, el cual analiza directamente la descripción simbólica del programa fuente y realiza las instrucciones dadas. La ventaja del proceso interpretado es que no necesita de dos fases para ejecutar el programa, sin embargo su inconveniente es que la velocidad de ejecución es más lenta ya que debe analizar e interpretar las instrucciones contenidas en el programa fuente.

1.4.1 Lenguajes de programación del lado del cliente

Un lenguaje del lado cliente es totalmente independiente del servidor. En el caso de una aplicación web esto permite que la página pueda ser albergada en cualquier sitio. Pero la página no se verá bien si el ordenador cliente no tiene instalado la versión correcta del navegador y los plugins¹ adecuados. El código, tanto del hipertexto como de los scripts, es accesible a cualquiera y esto puede afectar la seguridad.

1.4.1.1 HTML

“Es un lenguaje sencillo, permite definir documentos de hipertexto a base de ciertas etiquetas que marcan partes del documento dándoles una estructura o jerarquía. Presenta el texto de una manera estructurada y agradable, con enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido, video). El lugar donde se encuentra esta información puede ser el mismo documento o cualquier otro lugar de Internet.” [10]

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.

1.4.1.2 JavaScript

Los lenguajes Script han surgido para extender las capacidades del Lenguaje HTML. JavaScript es un lenguaje interpretado que añade dinamismo a las páginas web. Fue creado por Brendan Eich en la empresa Netscape Communications. El navegador tiene la responsabilidad de interpretar las sentencias. Se considera un lenguaje orientado a objetos y a eventos del usuario. Con su uso se pueden desarrollar scripts que den respuesta a dichos eventos.

¹ Programas que se relacionan con un sistema con el objetivo de añadirle funcionalidades.

Al ejecutarse en el lado cliente, evita la sobrecarga del servidor por peticiones que pueden tener respuestas locales. Los subsistemas Web y Transferencia 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 por lo que prescindir de su utilización en la realización de este trabajo no es una opción.

1.4.2 Lenguajes de programación del lado del servidor

La Programación del lado del servidor es una tecnología que consiste en el procesamiento de una petición de un usuario mediante el reconocimiento, ejecución e interpretación de un script en el servidor para generar páginas HTML dinámicamente como respuesta, las cuales son comprensibles para el cliente.

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.

1.4.2.1 PHP

PHP es un robusto lenguaje de programación del lado del servidor. Fue diseñado por Rasmus Lerdorf en 1994 como un CGI escrito en C que permitía la interpretación de un número limitado de comandos. Dada la aceptación del primer PHP y de manera adicional, su creador diseñó un sistema para procesar formularios al que le atribuyó el nombre de FI (Form Interpreter) y el conjunto de estas dos herramientas, sería la primera versión compacta del lenguaje: PHP/FI. [11]

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 (Portable Document Format) hasta analizar código XML. [12]

Resulta indiscutible que PHP es la solución por excelencia a la implementación de los subsistemas Web y Transferencia. El hecho de ser libre y poder utilizarlo sin necesidad de disponer de una licencia aumenta sus ventajas. PHP posee soporte para la gran mayoría de las plataformas existentes en la actualidad. Es eficiente para el desarrollo de aplicaciones Web y logra integrarse con un gran número de servidores, incluso propietarios. Su sintaxis clara y bien definida es lo que permite que sea sencillo de aprender y utilizar.

1.5 Entorno integrado de desarrollo

Integrated Development Environment (IDE) ó Entorno de Desarrollo Integrado. Los IDE son programas conformados por herramientas de gran necesidad para los programadores. Poseen editor de código, compilador, depurador y constructor de interfaz gráfica GUI. Pueden existir independientes o como parte de otras aplicaciones. Es posible que soporte un lenguaje de programación o varios.

1.5.1 Eclipse PDT

Eclipse es un entorno de desarrollo muy utilizado, se conoce generalmente por el desarrollo en Java. También se utiliza en las plataformas Cobol, C++, Jaba Mobile, PHP y Python. La comunidad Open Source estimula su desarrollo y sus proyectos se centran en la construcción de una plataforma abierta comprendida por frameworks extensibles, herramientas y versiones para la construcción.

El proyecto PDT (PHP Development Tools) ha tenido una gran respuesta por parte de los desarrolladores de PHP. Fue creado originalmente por IBM, actualmente está en manos de la Fundación Eclipse, una organización independiente sin ánimos de lucro que fomenta una comunidad de código abierto. Aunque consume muchos recursos de la computadora para su ejecución representa una buena solución por las razones siguientes:

- Es una plataforma de desarrollo muy estable y completamente libre.
- Presenta un excelente completamiento de código, lo que le facilita al programador no tener que memorizar todas las funciones que utilice el lenguaje de programación seleccionado.
- Permite la integración con el framework de desarrollo escogido para la implementación de la aplicación.

- Facilita la gestión y exploración de todas las clases generadas a lo largo de la edición del código o que se hayan importado de otra librería de PHP, estas clases las toma Eclipse y pueden ser usadas en todo el proyecto como si fuera parte de la librería estándar de PHP.
- Permite que se incorporen un número considerable de plug-ins en dependencia del tipo de aplicación que se quiera implementar.

1.6 Patrón arquitectónico

Los profesionales con experiencia resuelven, de forma mayormente intuitiva, muchos de los problemas de modelado de sistemas reales. El mejor profesional es el que reutiliza la misma solución retocada para resolver problemas similares en situaciones distintas. Estas experiencias, que engloban técnicas y criterios experimentales efectivos y probados, pueden ser difícilmente transmitidas formalmente a ingenieros noveles. [13]

Los Patrones pretenden ser la solución al problema de comunicación de experiencias planteado.

Cada patrón describe un problema que ocurre una y otra vez en mismo entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo siquiera en dos ocasiones de la misma forma. [14]

Esta idea de patrones, por su misma naturaleza genérica de Problema-Solución-Reutilización, se ha popularizado a nivel mundial. [13]

La utilización de patrones a nivel de arquitectura mejora también la concepción y entendimiento de los sistemas. Logra una estandarización para un común entendimiento entre los desarrolladores e ingenieros de software. Por estas razones se decide la aplicación de los mismos en el desarrollo del presente trabajo en aras de lograr un sistema robusto que cumpla con los objetivos propuestos para su realización.

1.6.1 Modelo en capas

El patrón de arquitectura en capas genera una organización con cierta jerarquía permitiendo que cada capa provea a la superior de los servicios que requiere. Aunque pueden implementarse sistemas con

dos, tres y n capas el ejemplo más representativo lo constituye el sistema con arquitectura de tres capas. **(Figura 6)**

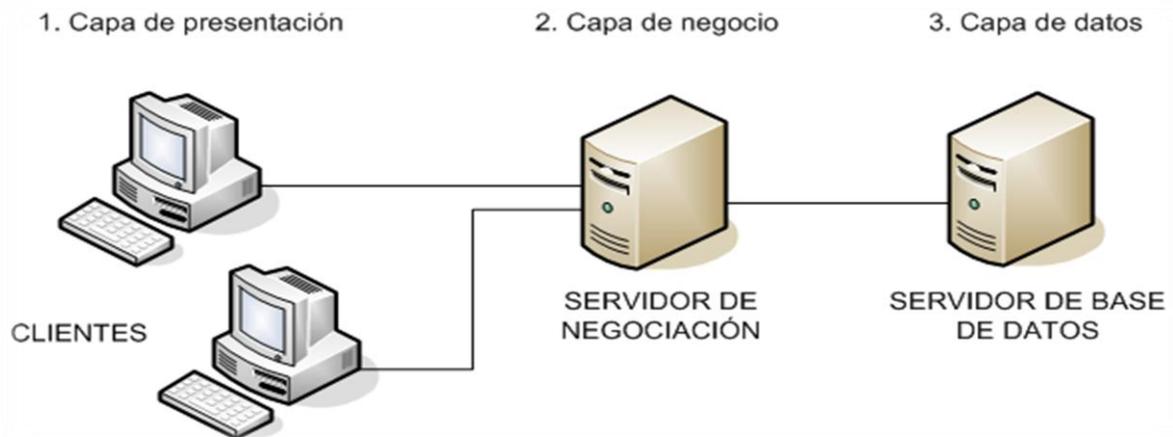


Figura 6. Representación del funcionamiento de un sistema con arquitectura de tres capas

Las tres capas planteadas son:

- ❖ Capa de Presentación.

Está habilitada para el manejo de las interfaces e interacción con los usuarios. Es la cara del sistema y debe ser funcional para lograr una buena comunicación entre el sistema y los agentes externos.

- ❖ Capa de Negocio (Capa Lógica).

La forman los elementos que dentro del software se encargan de la automatización de los procesos de negocio realizado por los usuarios. Se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. Contiene las reglas que deben cumplirse.

- ❖ La capa de Datos.

Reúne las capacidades que se ocupan del manejo de los datos. Puede estar formada por uno o más gestores de bases de datos. Recibe las órdenes de almacenamiento o recuperación de la información desde la capa de negocio.

Con el uso de este patrón arquitectónico se logra en cierta medida darle organización al software y ocultar las tecnologías usadas. En sistemas de gran tamaño las capas son casi siempre entidades muy complejas.

1.6.2 Modelo Vista Controlador

Fue introducido como parte de la versión Smalltalk-80 del lenguaje de programación Smalltalk. Se diseñó para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales son que el modelo, las vistas y los controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el modelo se refleje automáticamente en cada una de las Vistas. [15]

En la **figura 7** se puede observar el funcionamiento un sistema implementado usando el patrón arquitectónico Modelo Vista Controlador (MVC).

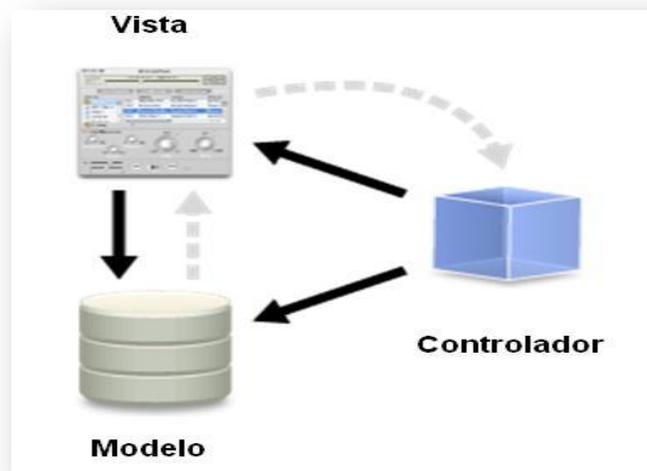


Figura 7. Representación del patrón arquitectónico Modelo Vista Controlador

❖ Modelo

Se responsabiliza del acceso a la capa de almacenamiento de datos. Lo más óptimo es que el modelo sea independiente del sistema de almacenamiento. Define las funcionalidades del sistema.

❖ Vista

Es la encargada de recibir datos del modelo y mostrarlos al usuario. Lleva un registro de su controlador asociado (pues generalmente lo instancia). Puede brindar el servicio de "Actualización ()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes). En el caso de una aplicación Web, la Vista es una página HTML con contenido dinámico sobre la cual el usuario puede realizar operaciones.

❖ Controlador

Recoge los eventos de entrada (un clic, un cambio en un campo de texto, entre otros). Estos eventos pueden derivar peticiones al modelo o a las vistas.

La separación de los datos de la representación visual de los mismos es una característica muy fuerte de este patrón. Con su uso se crea independencia de funcionamiento. Se facilita el mantenimiento en caso de errores. Permite la escalabilidad del sistema de ser necesario. Aunque la cantidad de archivos a mantener y desarrollar se incrementa considerablemente y la curva de aprendizaje resulta elevada respecto a otros patrones más sencillos son desventajas que se ven superadas fácilmente por sus prestaciones.

1.6.3 Fundamentación del patrón arquitectónico seleccionado

Durante el desarrollo de la PTARTV hasta el flujo de Análisis y Diseño se había propuesto para su implementación el uso del patrón arquitectónico en tres capas. Esta propuesta resulta inadecuada no solo por las ventajas que ofrece el patrón MVC (tratadas en el epígrafe anterior) frente al tres capas sino además porque la mayoría de los frameworks para php (será la vía de implementación del sistema como se observará más adelante) utilizan el MVC como patrón arquitectónico. Las razones antes expuestas resultan motivos decisivos para la selección de dicho patrón.

1.7 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. La **figura 8** muestra el uso de los principales frameworks de php en el año 2007.



Figura 8. Uso de los principales frameworks de php en el año 2007

Para la selección de un framework php es importante tener claro lo que se desea hacer y las prestaciones que puede brindar (ver Anexo III).

1.7.1 Symfony

Symfony es patrocinado por Sensio, una agencia web francesa. Su primer nombre fue Sensio Framework y sus clases contenían el prefijo sf. Cuando se decide lanzarlo como un framework open source, se acuerda llamarle Symfony.

Es un framework cuyo objetivo es optimizar el desarrollo de aplicaciones web. Utiliza el MVC por lo que separa la lógica de negocio de los datos y la presentación. Incorpora herramientas y clases que permiten reducir el tiempo empleado en la creación de una aplicación web compleja. Automatiza las

tareas comunes dándole la posibilidad al desarrollador de centrarse en los casos específicos de la aplicación.

Symfony está desarrollado usando PHP 5. Es usado actualmente en un gran número de proyectos y empresas como Yahoo ya han anunciado su migración. Su compatibilidad con gestores de bases de datos es alta incluyendo a los conocidos MySQL, PostgreSQL, Oracle y SQL Server.

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 helpers que permiten encapsular los efectos JavaScript compatibles con todos los navegadores en una única línea de código. [16]

Dos de las ventajas importantes de Symfony frente a otros frameworks son la calidad que brinda al código fuente y la cantidad de documentación disponible. El número de sus colaboradores ha crecido rápidamente y su labor va desde proponer parches y mejoras hasta detectar errores de la documentación. Symfony sin lugar a dudas se convierte en la solución factible para el desarrollo de los subsistemas Web y Transferencia de la PTARTV.

1.8 Sistema de Gestión de Bases de Datos

Un Sistema de Gestión de Base de Datos (SGBD) es un conjunto de programas que permite a los usuarios crear y mantener una Base de Datos (BD), por lo tanto, el SGBD es un software de propósito general que facilita el proceso de definir, construir y manipular la BD para diversas aplicaciones. Pueden ser de propósito general o específico [17]

Actualmente existe una amplia gama de SGBD con características propias, no obstante todos deben tener en cuenta los siguientes aspectos de manera general: abstracción de la información, la independencia de los datos, redundancia mínima, consistencia en los datos, seguridad, integridad, respaldo y recuperación, tiempo de respuesta y control de concurrencia. [18]

1.8.1 PostgreSQL

El SGBD relacional orientado a objetos conocido como PostgreSQL está derivado del paquete Postgres escrito en Berkeley, distribuida bajo licencia BSD. 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 y Python. [19]

La lista de lenguajes escritos anteriormente puede actualizarse añadiendo al Object Pascal, PHP, Ruby, Pike y C#.

Resulta importante señalar que mientras otros sistemas de bases de datos hacen uso de los bloqueos para controlar la concurrencia, el modelo multiversión (MVCC) o Control de Concurrencia Multiversión (Multiversion Concurrency Control) utilizado por PostgreSQL evita de manera elegante el problema de los bloqueos. En otras palabras, mientras se consulta una base de datos, cada transacción ve una imagen de los datos (una versión de la base de datos) como si fuera tiempo atrás, sin tener en cuenta el estado actual de los datos que hay por detrás.

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

1.9 Metodología de desarrollo de software

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 son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de los productos software. Es como un libro de recetas de cocina, en el que se van indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla. [20]

En la actualidad existen un gran número de propuestas que definen este marco de trabajo, se habla de metodologías tradicionales o robustas como Rational Unified Process (RUP), Microsoft Solutions Framework (MSF) y Métrica 3.0 o metodologías ágiles como Extreme Programming (XP), 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.

1.9.1 Proceso Unificado de Desarrollo (RUP)

El Proceso Unificado de Desarrollo tiene como fundamento la experiencia adquirida en el uso de la tecnología orientada a objetos y el desarrollo de software en una variedad de industrias encabezada por la compañía Rational, donde resaltan figuras como Grady Booch, James Rumbaugh e Ivar Jacobson. Es un proceso de desarrollo de software configurable que se adapta a través de los proyectos variados en tamaños y complejidad. Ha demostrado a través de los años su gran efectividad por lo que las grandes consultoras suelen recomendarlo incluso para proyectos delicados de misión crítica.



Figura 9. Logo del Proceso Unificado de Desarrollo

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. (Ver Anexo IV) El proceso describe qué entregables producir, cómo desarrollarlos y también provee los patrones.

Resulta importante resaltar que el proceso unificado es soportado por las principales herramientas que automatizan el modelado visual, la administración de cambios y las pruebas. Teniendo como basamento el tamaño del proyecto, el nivel de desarrollo crítico del mismo y el tiempo con el que se cuenta, RUP se convierte en la metodología a seguir para desarrollar los subsistemas Web y Transferencia.

1.10 Lenguaje Unificado de Modelado (UML)

A través de los años ha constituido un gran problema el entendimiento entre los propios desarrolladores y entre los desarrolladores y los clientes. La razón fundamental de esta problemática está dada en la abundancia de notaciones, metodologías y conceptos que hace que los desarrolladores de sistemas no se pongan de acuerdo en que es lo que realmente están elaborando. En busca de la Universalización y estandarización de las notaciones y procesos a utilizar, se conformó un consorcio liderado por la empresa Rational y por las principales empresas del mundo de la informática, quienes desarrollaron una notación llamada Lenguaje Unificado de Modelado.

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.

1.11 Herramientas CASE para la modelación UML

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) tienen como fin automatizar los aspectos claves de todo el proceso de desarrollo de un sistema.

De acuerdo con Kendall la ingeniería de sistemas asistida por ordenador es la aplicación de tecnología informática a las actividades, las técnicas y las metodologías propias de desarrollo, su objetivo es acelerar el proceso para el que han sido diseñadas, en el caso de CASE para automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas.

El uso de Herramientas CASE para la modelación UML proporciona mayor rapidez y entendimiento en el desarrollo de software. Este tipo de aplicaciones se han convertido en ayuda y apoyo imprescindible para los desarrolladores.

1.11.1 Rational Rose Enterprise

Rational Rose Enterprise constituye una herramienta poderosa para el modelado visual durante el ciclo de vida de un software. Rose propone cuatro tipos de modelo para la realización del diseño de un sistema, utilizando para ello una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software. Rational Rose Enterprise utiliza un proceso de desarrollo iterativo controlado, donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Ofrece la posibilidad generar código a partir de modelos Ada, ANSI C++, C++, CORBA, Java/J2EE, Visual C++ y Visual Basic. Proporciona mecanismos para la realización de la Ingeniería Inversa, es decir, a partir del código fuente de un programa obtener la información sobre su diseño.

Entre las características fundamentales del Rational Rose Enterprise se pueden citar:

- Cubre todo el ciclo de vida de un proyecto.
- Concepción y formalización del modelo.
- Construcción de los componentes.
- Transición a los usuarios.

- Certificación de las distintas fases.
- Capacidad para integrarse con cualquier sistema de control de versiones SCC-compliant, incluyendo a Rational ClearCase.

1.11.2 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. Es una herramienta colaborativa, por lo que soporta múltiples usuarios trabajando sobre el mismo proyecto; genera la documentación del proyecto automáticamente en varios formatos como Web o PDF, y permite control de versiones.

Entre sus características figuran:

- El uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de integrarse en los principales IDEs.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad en múltiples plataformas.

1.11.3 Fundamentación de la herramienta CASE seleccionada

Durante el desarrollo del Sistema PTARTV se utilizó como Herramienta Case el Rational Rose Enterprise hasta la fase de Análisis y Diseño. Un análisis detallado del desarrollo de este sistema, ha arrojado la realidad de que según las propuestas de software libre para la creación de dicha aplicación, la generación de código necesario y el buen entendimiento del equipo de desarrollo, se impone el uso de la herramienta CASE Visual Paradigm.

Visual Paradigm tiene un gran número de ventajas que le permiten tener un elevado índice de aceptación, además se acogerse a la política del software libre. Es fácil de usar, amigable, genera código, facilita la importación desde Rational Rose, genera informes, posee editor de figuras, etc. 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*); 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.

Aunque esta decisión implique la migración de los diagramas que se realizaron anteriormente, la redefinición de algunos términos y mayor costo de tiempo son efectos colaterales que hay que asumir en aras de obtener un sistema con toda la robustez que se desea.

1.12 Sistemas de streaming

Un sistema de streaming es una aplicación informática cuya función es realizar streaming de audio y video o de uno de los dos. Para una mayor comprensión de este tipo de aplicaciones se recomienda retomar el tópico 1.2.3 del presente trabajo donde se valora el funcionamiento de la tecnología streaming.

1.12.1 FFmpeg y FFServer

FFmpeg es una solución completa para grabar, convertir y transmitir audio y video por la red. Incluye la librería libavcodec. FFmpeg es desarrollada para Linux, pero puede ser compilado en la mayoría de los Sistemas Operativos, incluyendo Windows.

El proyecto está compuesto por varios componentes:

- FFmpeg: una herramienta de línea de comandos para convertir archivos multimedia entre distintos formatos.
- FFserver: un servidor streaming de multimedia para transmisiones en vivo.

- FFplay: un reproductor de multimedia simple basado en SDL (Simple Directmedia Layer, librería de bajo nivel para el acceso al audio, teclado, mouse, tarjetas gráficas, etc.) y las librerías FFmpeg.
- Libavcodec: es una librería contenedora de codificadores y decodificadores de audio y video.
- Libavformat: es una librería contenedora de multiplexadores y demultiplexadores para formatos contenedores de multimedia.

FFserver recibe archivos previamente almacenados o flujos de alguna instancia ffmpeg como entrada y los transmite utilizando RTP/RTSP/HTTP.

FFserver soporta una interfaz HTTP la cual muestra el estado actual del servidor. Simplemente hay que poner en el navegador web la dirección del flujo especial de estado especificado en el archivo de configuración.

El FFmpeg es una excelente herramienta para la conversión de formatos de video desde la línea de comandos o consola y por tal razón esta funcionalidad lo convierte en la base del desarrollo del subsistema de Transferencia. Cuando se transmite contenidos por la red haciendo uso del servidor se complica el proceso pues la configuración y el proceso de agregarle nuevos ficheros de entrada se realiza mediante la edición de un archivo de configuración llamado ffserver.conf. Los flujos de entrada son llamados semillas (feeds), y cada uno debe ser especificado en una sección <Feed> en el archivo de configuración lo cual hace este trabajo engorroso y complejo de automatizar.

1.12.2 VideoLAN

El VLC es capaz de transmitir todo lo que sea capaz de reproducir. El proceso de creación de un nuevo flujo a transmitir lleva una serie de pasos que deben ser realizados desde la interfaz (ya sea gráfica, web o telnet) que incluyen la selección del tipo de flujo (en demanda o en vivo), la selección del origen (dispositivo de captura, DVD, archivo previamente almacenado, etc.), selección del formato de salida (transcodificación en tiempo real a numerosos formatos).

VLS (VideoLAN Server): el cual puede hacer streaming de archivos MPEG-1, MPEG-2 y MPEG-4, DVDs, canales digitales de satélite, canales de televisión digital terrestre y videos en vivo en la red en multidifusión o unidifusión. La mayoría de las funcionalidades de VLS pueden ser encontradas en VLC.

La solución para streaming de VideoLan incluye el VLC media player: el cual puede ser utilizado como un servidor y como un cliente para hacer streaming y recibir streaming por la red.

Para que VLC resulte óptimo como servidor de Streaming para la PTARTV debe lograrse que sea capaz de leer, configurar y transmitir automáticamente los archivos que se encuentren almacenados en una carpeta determinada. Lo descrito anteriormente debe realizarse cuando se inicie el servidor y es necesario que detecte cuando se añade una nueva media a la carpeta. Debe tenerse en cuenta que es preciso mantener la instancia del VLC en ejecución de forma constante, brindando servicios. Esto es fundamental para una aplicación que funcione como servidor.

1.12.3 Darwin Streaming Server

Es la versión de código abierto de la tecnología del Apple's QuickTime Streaming Server que permite enviar un streaming de multimedia a clientes a través de internet utilizando los protocolos RTP y RTSP.

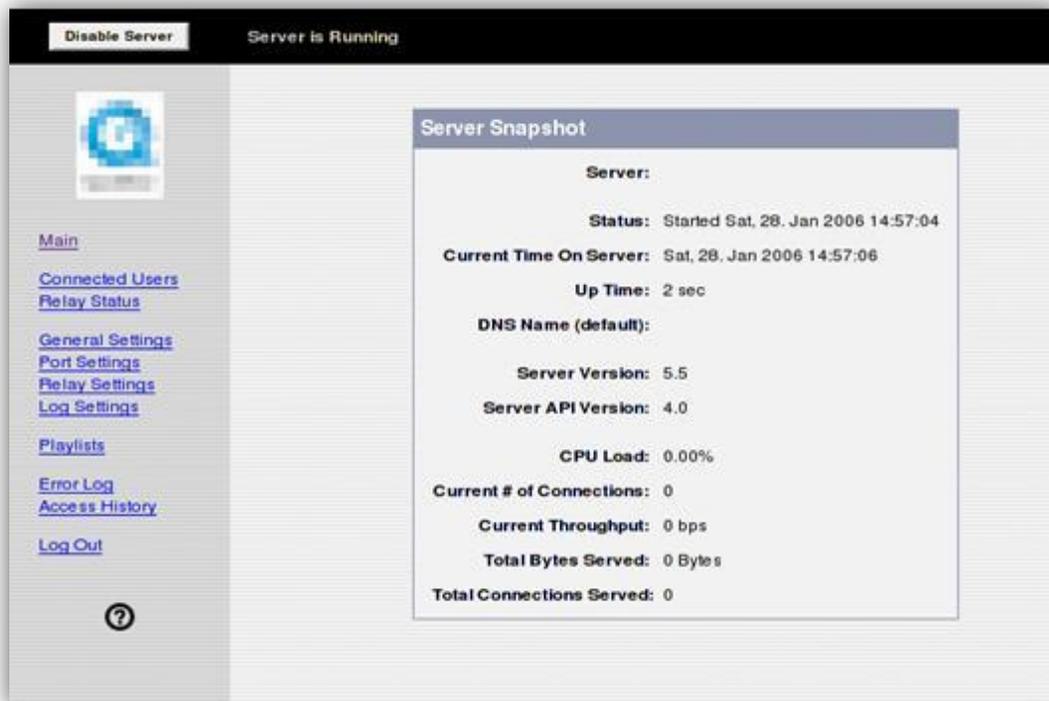


Figura 10. Interfaz del Darwin Streaming Server

Basado en el mismo código base del QuickTime Streaming Server, el Darwin Streaming Server proporciona un alto nivel de personalización y funciona en una gran variedad de plataformas permitiendo manipular el código fuente para satisfacer las necesidades de los desarrolladores. Darwin Streaming Server es un proyecto de código abierto pensado para desarrolladores que necesiten hacer streaming de archivos QuickTime o MPEG-4 en plataformas alternativas como Windows, Linux y Solaris; o para aquellos desarrolladores que necesiten extender y/o modificar el código del servidor streaming existente para satisfacer sus necesidades.

1.12.4 MPEG4IP

MPEG4IP provee un sistema completo para explorar archivos multimedia MPEG-4. Es una herramienta para hacer streaming de video y audio que es orientada a estándares y libre de protocolos y extensiones propietarias. El sistema incluye un codificador de audio MPEG-4 AAC y MP3, dos compresores de video MPEG-4, un creador de archivos MP4 con pistas (hints), un servidor basado en los estándares de la IETF (*Internet Engineering Task Force*) y un reproductor MPEG-4 que pueden entre los dos hacer streaming y reproducir desde un fichero local.

1.12.5 Mp4live

Mp4live solo está disponible para Linux. Soporta las siguientes características:

- Códec de video: MPEG-4 (XviD, XviD-0.9.2 o XviD-1.0), H.261, MPEG-2, H.263, H.264
- Códec de audio: AAC, MP3, MPEG-1, MPEG-2, AMR NB and WB, G.711 ALAW y ULAW.
- Filtros de video: desentrelazado.
- Interfaces de entrada de video: V4L (Video for Linux), V4L2.
- Interfaces de salida de audio: OSS.
- Opciones de grabación: H.263, H.264, MPEG-4 video, MPEG-2 video, MP3 audio, AAC audio, AMR audio, RAW audio (PCM) y video (YUV).
- Opciones de transmisión: (RTP multidifusión o unidifusión) MPEG-4 video, MPEG-2 video, H.261 video, H.263 video, H.264 video, G.711 audio, AAC audio, MP3 audio, AMR audio.

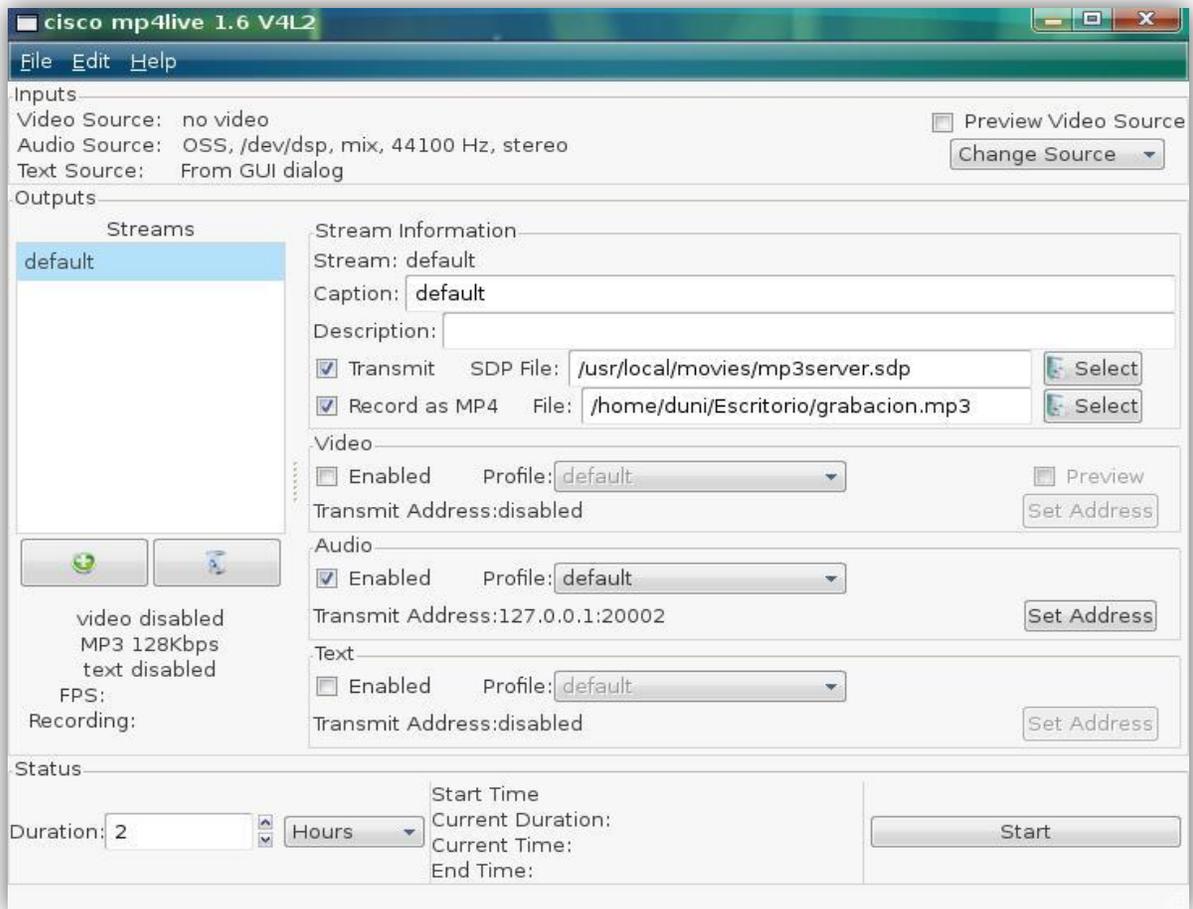


Figura 11. Interfaz del Mp4live

1.12.6 Fundamentación de las aplicaciones de streaming seleccionadas

Bajo licencias libres existen servidores streaming que pudieran usarse en el subsistema Web, pero presentan deficiencias en diferentes áreas lo que trae consigo que no exista una única herramienta que posea todos los requerimientos para satisfacer las necesidades.

Aunque durante el diseño de la PTARTV se había propuesto únicamente como server streaming al VLC, es una realidad según se planteó en el tópico 1.12.3 de presente trabajo que el mismo posee una serie de deficiencias hasta el momento no solucionadas.

Se propone usar al Darwin Streaming Server como servidor streaming de video bajo demanda.

Las ventajas que sustentan la selección anterior son:

- Darwin Streaming Server está disponible para descarga gratuita bajo la Apple Public Source License.
- Una vez instalado el servidor crea una carpeta en la cual se deben copiar los archivos en formato MP3 para audio y MOV o MP4 (luego de someterse a un proceso denominado hint) para video. Los ficheros que se encuentren en esta carpeta están publicados y pueden ser accedidos desde la red mediante el protocolo RTSP.
- El servidor cuenta con una interfaz web que permite visualizar su estado y además posibilita su configuración y administración.

Es una realidad que el Darwin Streaming Server es extraordinario como servidor streaming de video bajo demanda, pero por otra parte, no es capaz por sí solo de realizar el envío de flujo de difusión para la transmisión en vivo. Como solución a dicha problemática se propone al Mp4live como sistema auxiliar para la captura y codificación a MP4 en tiempo real permitiendo que el archivo sea transmitido por el Darwin como si fuera bajo demanda.

1.13 Servidor Web

El servidor Web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición.

El servidor Web será fundamental en el desarrollo de las aplicaciones del lado del servidor que se implementen pues funcionarán sobre él.

1.13.1 Apache

Servidor web de código abierto. 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). La 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. [21]



Figura 12. Logo del servidor Apache

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 Transferencia.

1.14 Sistema operativo

El sistema operativo es el programa (o software) más importante de un ordenador. Para que funcionen los otros programas, cada ordenador de uso general debe tener un sistema operativo. Los sistemas operativos realizan tareas básicas, tales como reconocimiento de la conexión del teclado, enviar la información a la pantalla, no perder de vista archivos y directorios en el disco, y controlar los dispositivos periféricos tales como impresoras, escáner, etc. [22]

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. [22]

Los sistemas operativos proporcionan una plataforma de software encima de la cual otros programas, llamados aplicaciones, puedan funcionar. Las aplicaciones se programan para que funcionen encima de un sistema operativo particular, por tanto, la elección del sistema operativo determina en gran medida las aplicaciones que puedes utilizar. [22]

Al abogarse por el uso del software libre para el desarrollo del sistema PTARTV y en específico de los subsistemas Web y Transferencia, se impone entonces la necesidad de una correcta selección del Sistema Operativo a utilizar como plataforma.

1.14.1 UBUNTU Linux

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.

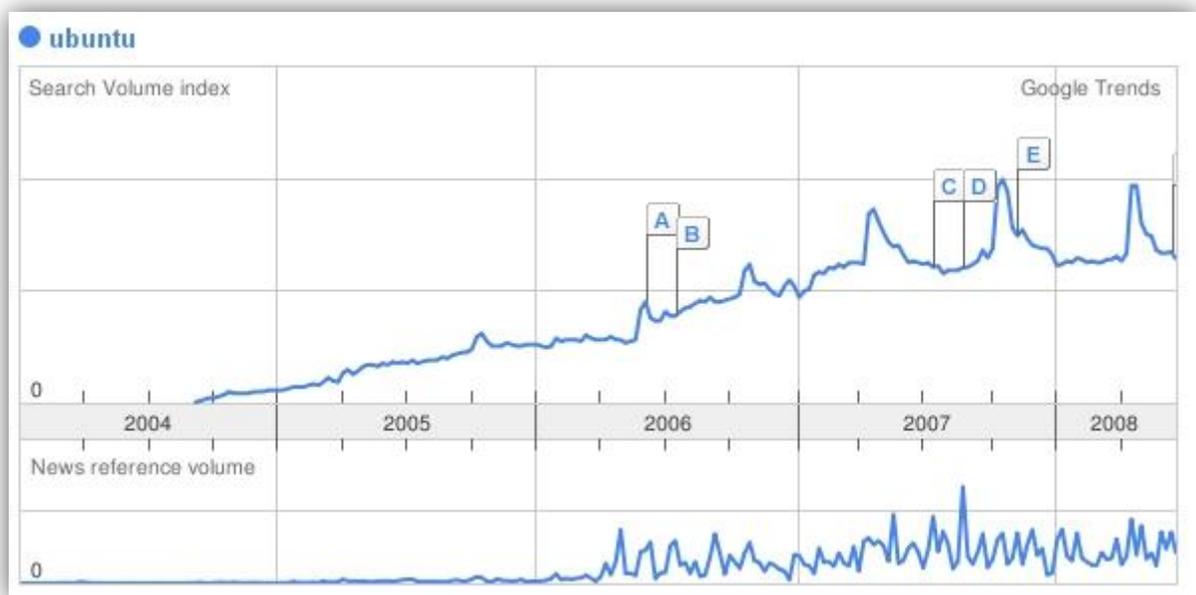


Figura 13. Crecimiento del uso de la distribución Ubuntu

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 (cada 6 meses: las versiones .04 en abril y las .10 en octubre).

Su principal patrocinador es Canonical Ltd., una empresa privada fundada y financiada por el empresario sudafricano Mark Shuttleworth. El nombre de la distribución proviene del concepto zulú y xhosa de Ubuntu, que significa "humanidad hacia otros" o "yo soy porque nosotros somos".

Requerimientos recomendados para el uso de Ubuntu

- Procesador: 700 MHz x86
- Memoria RAM: 384 MB
- Disco duro: 8GB
- Tarjeta de vídeo capaz de soportar resolución de 1024x768
- Tarjeta de sonido
- Conexión a internet

Requerimientos añadidos para efectos gráficos

- Procesador: 1.2 GHz x86
- Tarjeta de video soportada

La popularidad de Ubuntu es grande. En la actualidad Google usa en escritorio, su variante de esta distribución, bautizada como Goobuntu. Wikipedia migró sus servidores a Ubuntu Server en el 2008.

Ubuntu se convierte para las necesidades de este trabajo en una solución robusta. Por tanto, será usado como sistema operativo para la PTARTV la última versión de la distribución Ubuntu. Actualmente el sistema operativo se encuentra en su versión 9.4.

1.15 Conclusiones

En este capítulo se ha realizado un análisis de las tendencias tecnológicas, técnicas y lenguajes de programación, metodologías de desarrollo y herramientas que serán usados en el desarrollo del sistema PTARTV. Se han definido también una serie de conceptos necesarios para comprender el entorno del trabajo. Cuando las propuestas tecnológicas variaron respecto a las establecidas durante el diseño de la PTARTV se hizo énfasis en las razones que sustentan dichos cambios. Una selección racional de los elementos descritos anteriormente permitirá obtener un sistema con buena calidad.

CAPÍTULO 2

Descripción de la Solución Propuesta

2.1 Introducción

En este capítulo se presenta una breve descripción de los subsistemas a implementar y se hace referencia al modelo de implementación estructurado a partir de los diagramas de componentes. Todo lo anterior tiene su base en el diseño realizado de la Plataforma de Transmisión Abierta para Radio y Televisión. Se analizan también los estilos y estándares de codificación utilizados en la implementación.

2.2 Descripción de los subsistemas

Cuando se trabaja en un proyecto que tiene un gran número casos de uso existe la tendencia de dividirlo en aplicaciones más pequeñas que sean operativas por sí mismas a las cuales se les llama subsistemas. Estos subsistemas agrupan los casos de usos por áreas funcionales. La PTARTV se dividió en cinco subsistemas, siendo objeto de implementación del presente trabajo los subsistemas Web y Transferencia.

2.2.1 Subsistema Web

Antes de realizar cualquier análisis es importante comprender que el framework Symfony utilizado para el desarrollo de las aplicaciones estructura los sistemas en módulos. Por la razón antes expuesta se conforma el subsistema Web en siete módulos que permiten una implementación clara y correcta.

El subsistema Web es la interfaz externa de la PTARTV. Haciendo uso de él los usuarios que se conecten, podrán satisfacer sus preferencias televisivas consumiendo los servicios brindados por esta aplicación informática.

➤ **Módulo Noticias Web**

Este módulo es el encargado de gestionar las noticias que se publicarán en la PTARTV siendo de gran importancia para la información de los usuarios. El editor de noticias tendrá la posibilidad de adicionar, modificar o eliminar una noticia dada. Haciendo uso de este módulo los administradores podrán también mantener actualizado al público de sucesos radiales y televisivos de carácter nacional e internacional. Estas noticias podrán además llevar asociadas imágenes que añadan mayor calidad a la información que se desea comunicar.

➤ **Módulo Publicación Web**

La responsabilidad de este módulo es permitir adicionar, modificar o eliminar una publicación web. Se debe entender lo anterior como el acto mediante el cual se hace disponible para la visualización a través de la web de un archivo de audio o video. Una vez adicionada y visualizada la publicación, los usuarios podrán tener la posibilidad de reproducir la media asociada haciendo uso del reproductor embebido en el navegador.

➤ **Módulo Sección Web**

Con la utilización de este módulo se hace posible agrupar por categorías las medias que se visualizarán en el entorno web. Ejemplo de lo anterior sería si al entrar a la página y una vez registrado se va a la sección películas el usuario podrá visualizar las películas disponibles para la reproducción así como los datos asociados a la misma. Este módulo le añade un nivel mayor de organización al Subsistema Web.

➤ **Módulo Cartelera**

Haciendo uso de este módulo se informará a los usuarios la programación para los canales que son transmitidos por la PTARTV. La actualización de la cartelera será de manera automática lo cual contribuirá a eliminar errores humanos que pudieran introducirse en este proceso. Lo anterior permite que haya correspondencia entre lo que se anuncia con anterioridad y lo que se está transmitiendo en un determinado momento.

➤ **Módulo Estadísticas**

Mediante este módulo se podrá llevar un reporte estadístico del rating² de visualización de las medias publicadas en la web. Permitirá conocer cuáles son los materiales más visualizados y la generalidad de preferencias de los usuarios.

➤ **Módulo Encuestas**

Este módulo es responsable de gestionar las encuestas que se publiquen en la PTARTV. Haciendo uso de estas encuestas se logrará conocer los puntos de vista de los usuarios respecto a los programas que se transmiten, las publicaciones de la web o cualquier otro tema. Los administradores podrán conocer los gustos y necesidades de los usuarios. Con el uso de este módulo se logrará la retroalimentación en el sistema.

➤ **Módulo Foro**

Resulta de gran importancia permitirle al usuario la comunicación con otros usuarios o con los administradores del sistema. Por la razón antes dicha se le añade el módulo foro a la PTARTV, el cual permitirá gestionar los temas a discutir. Con la inserción de este módulo se aumenta la interactividad.

2.2.2 Subsistema de Transferencia

La responsabilidad de este subsistema es codificar los materiales de audio y video a los formatos requeridos para la transmisión y publicación dentro de la Plataforma de Transmisión Abierta para Radio y Televisión. Además, este subsistema se encargará del chequeo de la integridad de los ficheros codificados, permitiendo que sean copiados al servidor de medias y brindará un reporte del estado de la conversión.

➤ **Módulo Conversión**

Haciendo uso de las funcionalidades brindadas por este módulo la PTARTV podrá manejar los ficheros que serán transmitidos o publicados en un formato estándar, previamente seleccionado.

² Es un término usado para denominar el por ciento de visualizaciones realizadas por los receptores.

➤ Módulo Reporte de Conversión

Con la utilización de este módulo se podrá obtener un reporte del estado de la conversión, el cual servirá para tener conocimiento de los errores que pudieran suceder cuando se codifique un material. Los informes generados impedirán que una media con errores de conversión entre dentro del sistema.

2.3 Modelo de implementación

El modelo de implementación está comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes se pueden encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos. Este artefacto describe cómo se implementan los componentes, congregándolos en subsistemas organizados en capas y jerarquías, y señala las dependencias entre éstos. [23]

En el modelo de implementación de la PTARTV aparecen representados componentes externos al framework utilizado para el desarrollo de los subsistemas. Estos componentes son necesarios para la implementación de funcionalidades específicas. Ejemplo de lo anterior son: el LDAP³, para la autenticación mediante dominio, Propel es el framework ORM⁴ para PHP que facilita la interacción con la base de datos, CURL para el copiado de archivos a través de la red y mozilla-plugin-video para la reproducción de archivos multimedia.

La **figura 14** muestra el modelo de implementación de la PTARTV.

³ **LDAP** (Lightweight Directory Access Protocol, en español, Protocolo de Acceso a Directorios Ligeros) es un protocolo usado para acceder a "Servidores de Directorio".

⁴ **ORM** (Object-Relational Mapping, en español, Mapeo Relacional de Objetos) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.

CAPÍTULO 2: Descripción de la Solución Propuesta

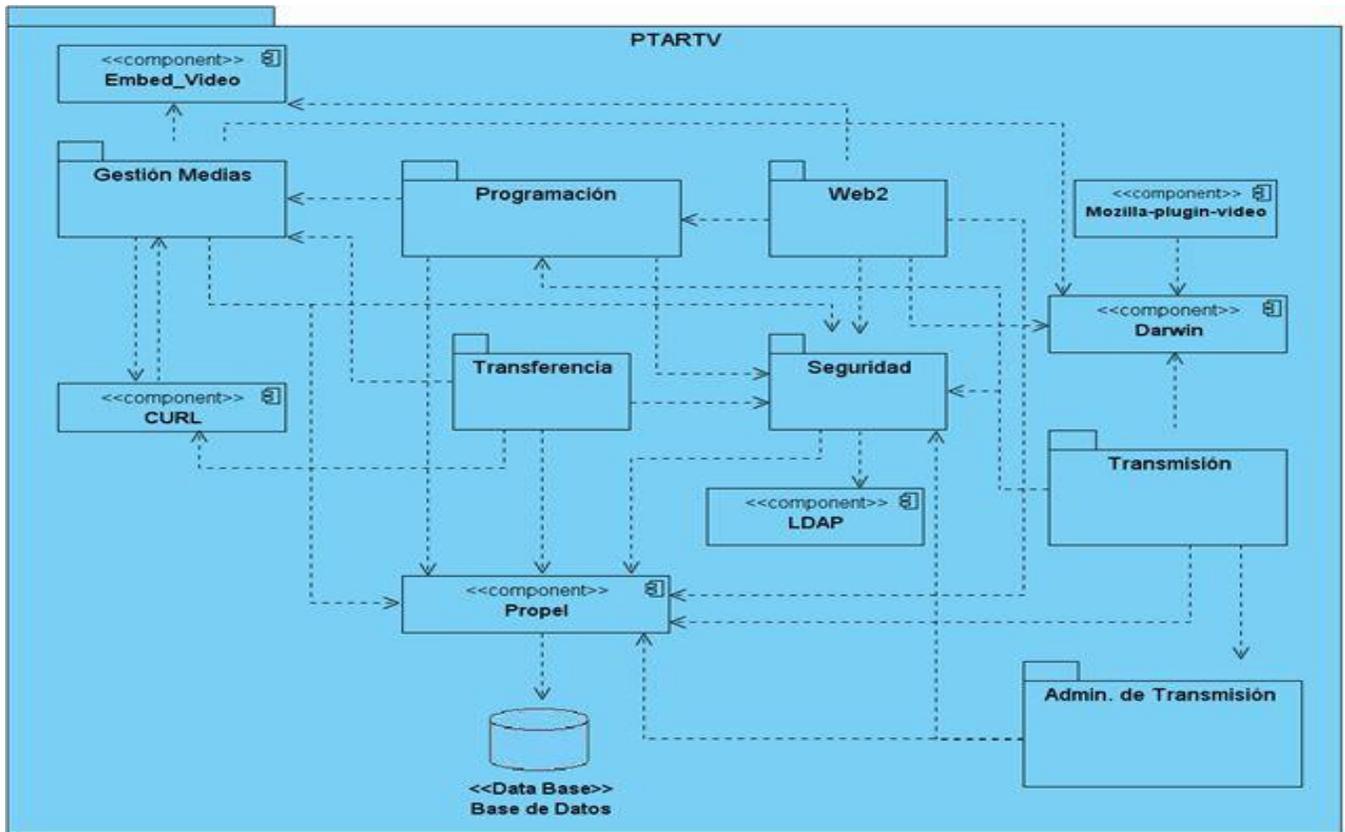


Figura 14. Modelo de implementación

Para una mejor comprensión del modelo anteriormente presentado se detalla el empaquetamiento de módulos por subsistemas.

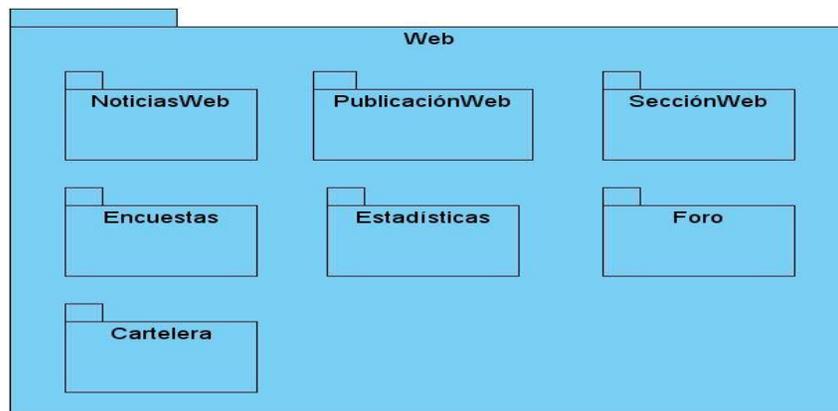


Figura 15. Paquete de módulos del subsistema Web

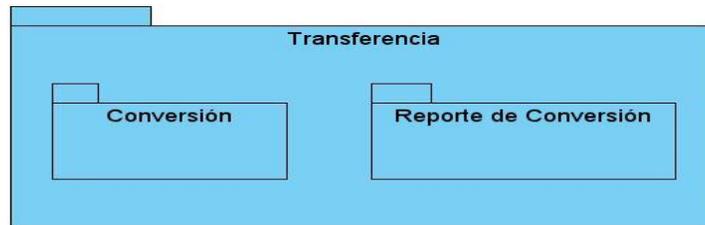


Figura 16. Paquete de módulos del subsistema Transferencia

2.3.1 Diagramas de componentes

El modelo de implementación se estructura a partir de los diagramas de componentes, por tal razón debe entenderse el funcionamiento interno de los módulos para luego asociarlos como un todo dentro de los subsistemas que componen. Al utilizar la metodología RUP debe conocerse que la misma hace énfasis en el uso de diagramas de componentes para modelar los subsistemas de implementación.

“Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Estos diagramas muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, bibliotecas cargadas dinámicamente, entre otros. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente.” [24]

Los principales estereotipos estándares que presentan los componentes dentro de un modelo son:

- `<<executable>>`
 - Representa programas que se ejecutan en un nodo.
- `<<file>>`
 - Son ficheros de datos o código fuente.
- `<<library>>`
 - Modelan librerías estáticas o dinámicas.
- `<<table>>`
 - Constituyen tablas de bases de datos.
- `<<document>>`
 - Simbolizan documentos.

➤ Subsistema Web

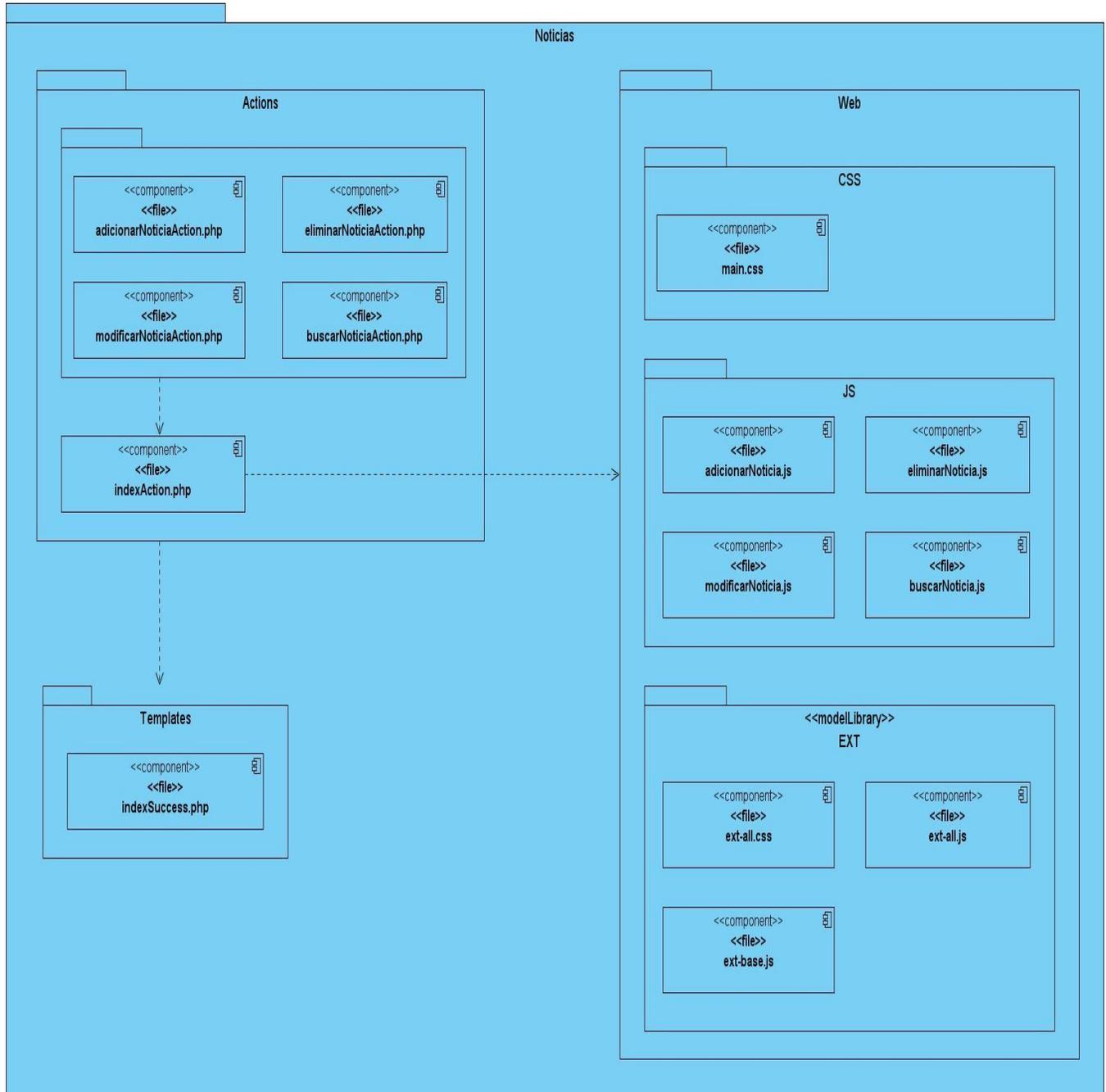


Figura 17. Diagrama de componentes del módulo Noticias Web

CAPÍTULO 2: Descripción de la Solución Propuesta

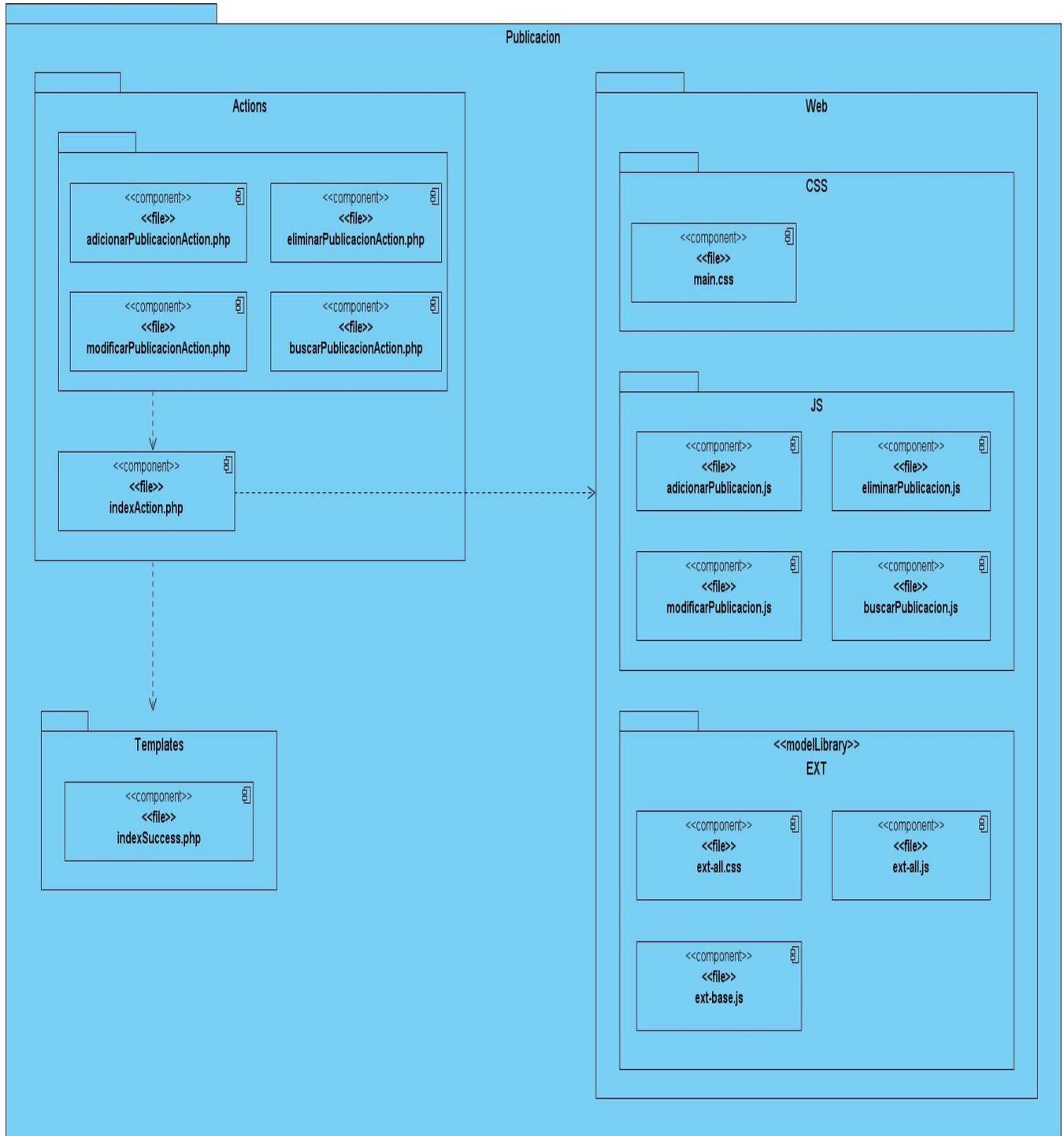


Figura 18. Diagrama de componentes del módulo Publicación Web

CAPÍTULO 2: Descripción de la Solución Propuesta

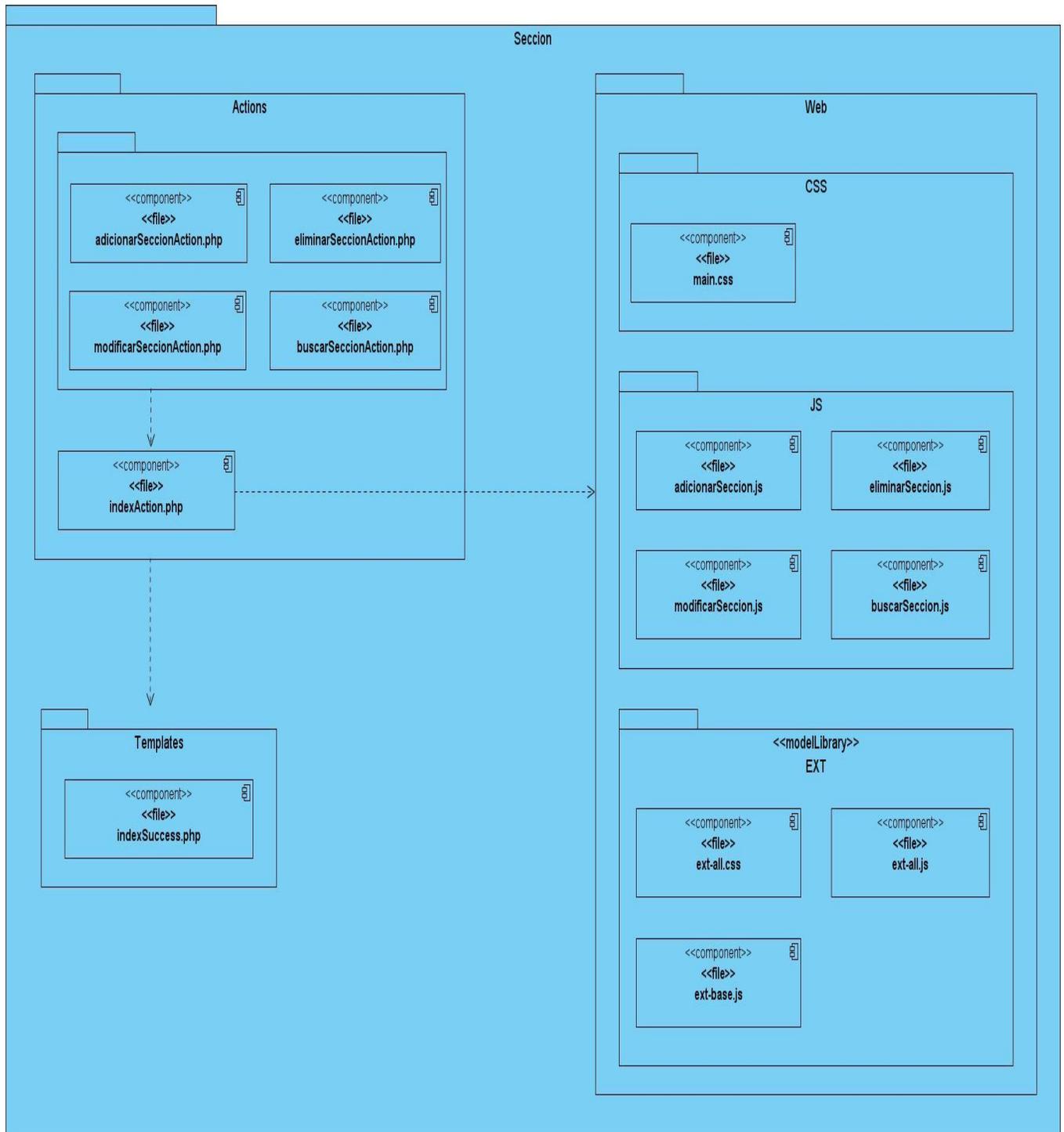


Figura 19. Diagrama de componentes del módulo Sección Web

CAPÍTULO 2: Descripción de la Solución Propuesta

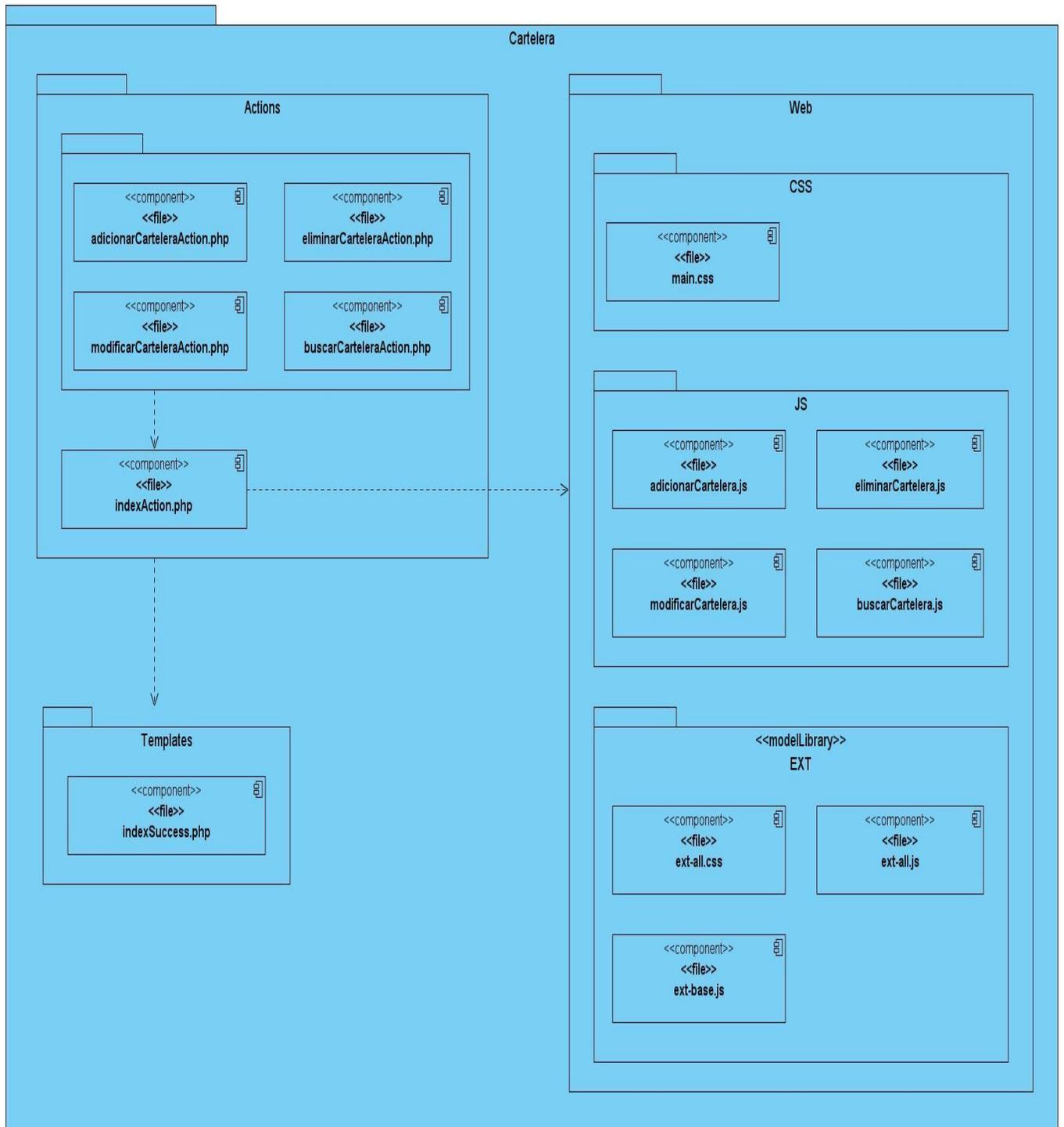


Figura 20. Diagrama de componentes del módulo Cartelera

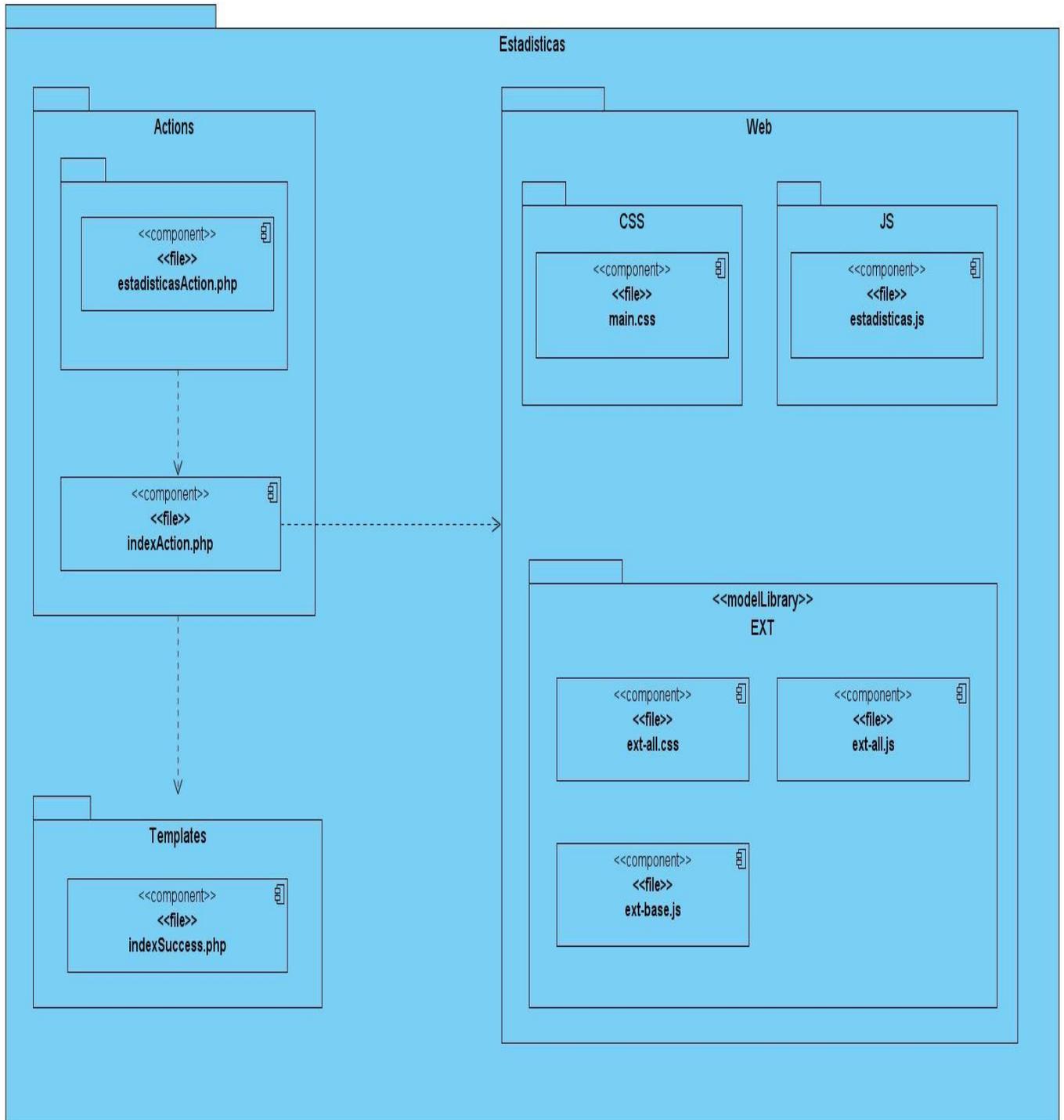


Figura 21. Diagrama de componentes del módulo Estadísticas

CAPÍTULO 2: Descripción de la Solución Propuesta

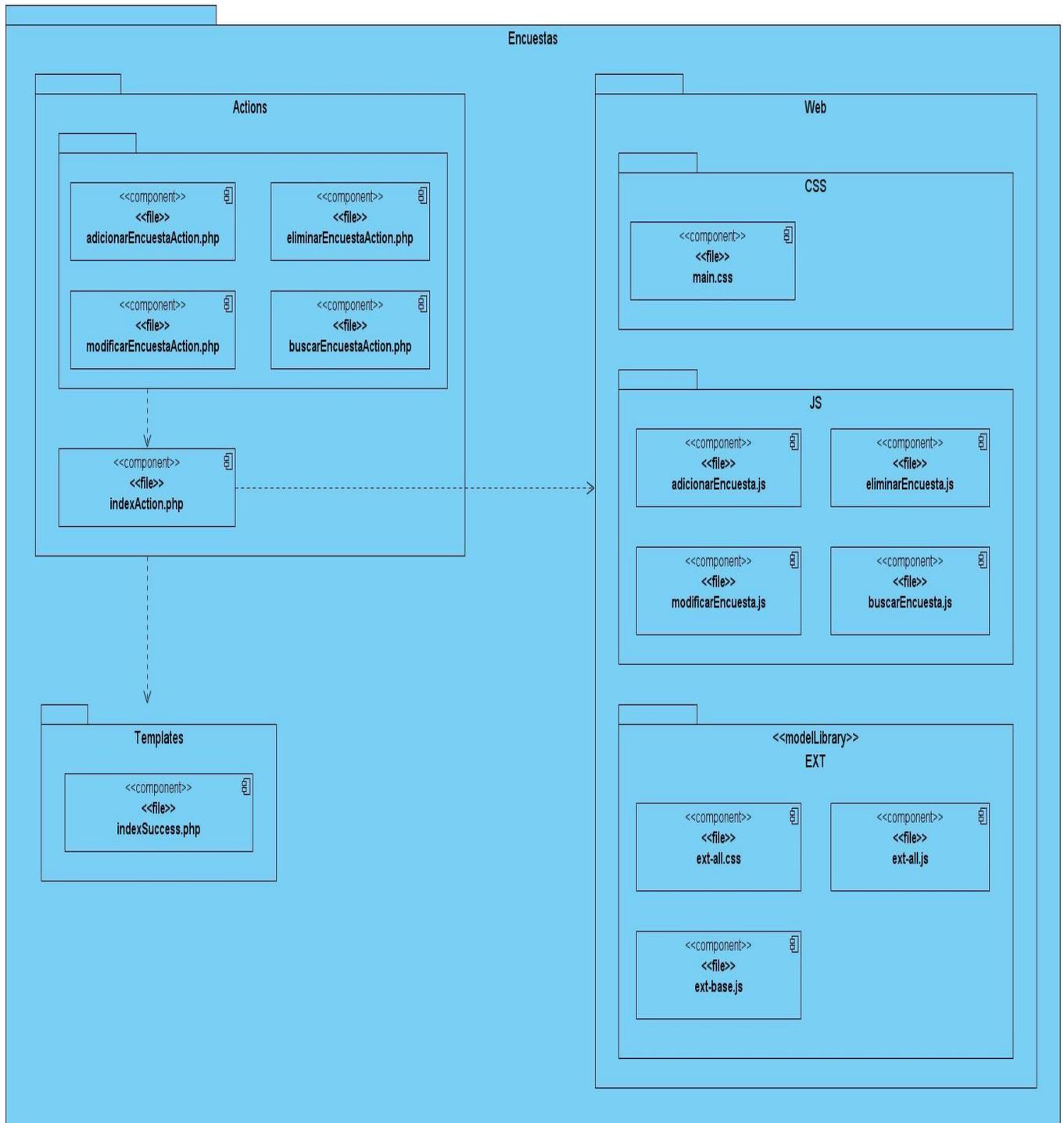


Figura 22. Diagrama de componentes del módulo Encuestas

CAPÍTULO 2: Descripción de la Solución Propuesta

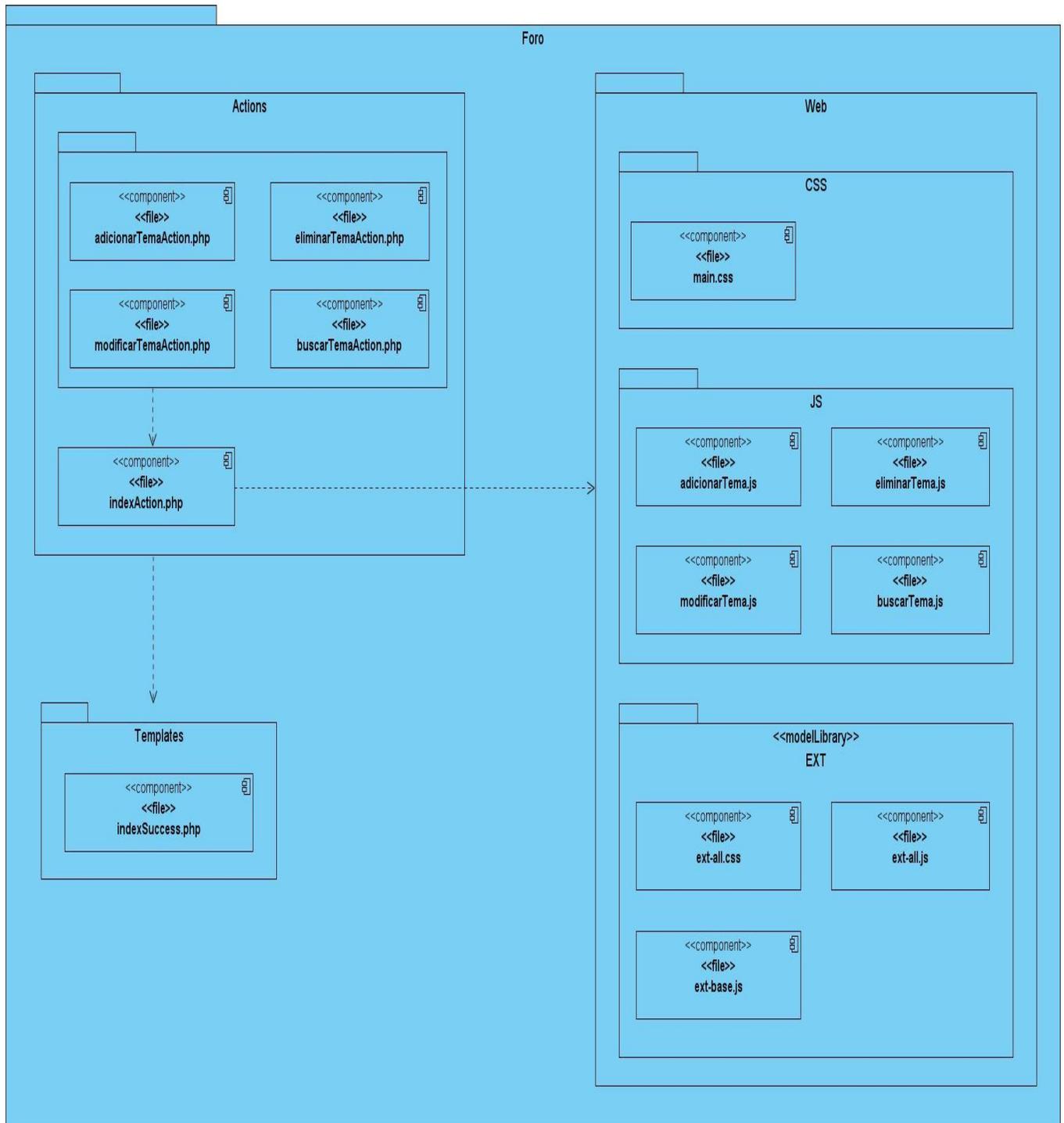


Figura 23. Diagrama de componentes del módulo Foro

➤ Subsistema de Transferencia

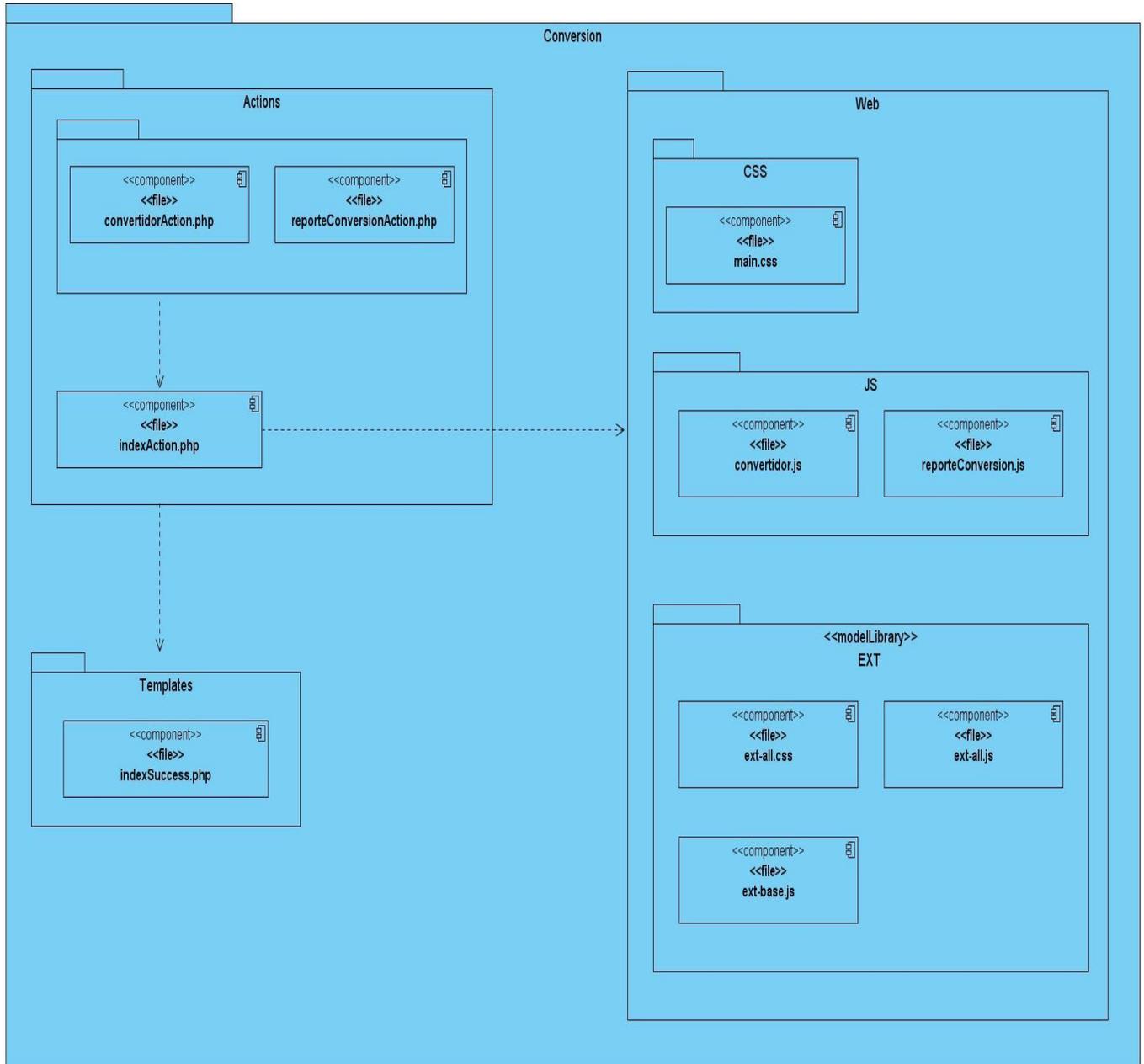


Figura 24. Diagrama de componentes del módulo Conversión

2.4 Estándares de codificación

Cuando se trabaja dentro de un equipo de desarrollo debe generalizarse un estándar de codificación de manera que exista similitud de formato entre el código de los implementadores. Es una realidad que cada programador tiene un estilo personal para escribir su código fuente pero al definirse un estándar para utilizar se debe acoplar dicho estilo personal con el de todo el equipo.

Se especifica como estándar de codificación al estilo de la sintaxis que se usará en la programación, lo anterior puede incluir la forma de estructurar el código fuente, la manera de colocar las llaves, la indentación del código, el empleo de los paréntesis, entre otros aspectos.

Al conformar un estándar de codificación deben tenerse en cuenta varios criterios entre los cuales se pueden mencionar:

- ✓ Nombres representativos para variables, controles y procedimientos.
- ✓ Indentación (sangrías) y espacios apropiados en el código.
- ✓ Documentación del código (poner comentarios para aclarar).
- ✓ Procedimientos Coherentes.

Nombres representativos: Cuando se definan nombres de variables, controles y procedimientos deben hacerse de forma representativa con el propósito de facilitar el entendimiento.

Indentación y espacios apropiados en el código: Visualizar un código fuente puede resultar complicado pero haciendo uso de la indentación se obtiene una mejor visibilidad pues se muestran las líneas que están sujetas a otras. Por ejemplo, todas las líneas que forman el cuerpo de una clase serán indentadas con la instrucción principal de la misma.

Documentación del código: Es una buena práctica documentar aquellas secciones de código más complicadas e inusuales. Al definir las variables y arreglos se deben documentar para que su función pueda ser comprendida posteriormente.

Procedimientos Coherentes: No debe añadirse demasiada complejidad a los procedimientos pues si manejan muchas tareas resulta natural que sean difíciles de entender y tengan una alta probabilidad de ocurrencia de errores.

CAPÍTULO 2: Descripción de la Solución Propuesta

A continuación se especifican los criterios de estandarización de código utilizados para el desarrollo de los subsistemas Web y Transferencia.

✓ Indentación

- Define que debe usarse los sangrados para indentar el código y nunca espacios. Poner las llaves de control en la línea subsiguiente.

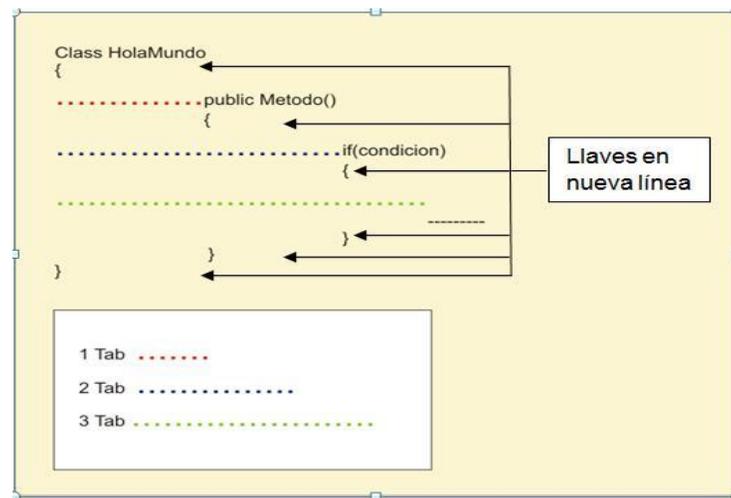


Figura 25. Código indentado

✓ Saltos de Línea

- Añadir un salto de línea después del cierre de los paréntesis de los parámetros.
- Añadir un salto de línea después de un punto y coma, cuando termina la sentencia.

✓ Espacios y líneas en blanco

- Usar espacios en blanco para mejorar la legibilidad del código.
- Usar espacios en blanco en ambos lados del operador de símbolos, después de comas y después de las declaraciones.
- Usar líneas en blanco para separar trozos de código.

✓ Longitud de la línea

- Evitar las líneas de más de 80 caracteres cuando se supere esta cifra se debe reorganizar el código usando algún principio descrito anteriormente.

CAPÍTULO 2: Descripción de la Solución Propuesta

✓ Apariencia de clases y funciones

- Usar la notación Camel Casing⁵ para nombrar las clases y las funciones. Ejemplo:

```
class conversionActions extends sfActions
{
    public function executeListadoNoticia()
    {
        ***
    }
}
```

✓ Apariencia de atributos

- Los atributos deben ser escritos en minúsculas y su nombre debe guardar relación con el valor que almacena. Ejemplo:

```
$nombre = $objeto->getNombre();
```

✓ Apariencia de constantes

- Las letras que conforman el nombre de las constantes se escribirán con letras mayúsculas.

2.5 Conclusiones

Durante el desarrollo del presente capítulo se expuso la descripción de los subsistemas Web, Transferencia y de los módulos que los conforman. Se presentó el modelo de implementación de la PTARTV, el empaquetamiento de los módulos por subsistemas y se mostró el funcionamiento interno de cada uno de los módulos a través de los Diagramas de componentes. Por último fue analizado y detallado el estándar de codificación utilizado en la implementación de los subsistemas.

⁵ Camel Casing es una norma de notación que establece que las palabras compuestas deben escribirse con mayúscula excepto la primera palabra. Ejemplo: indexHtml.

CAPÍTULO 3

Validación de la Solución Propuesta

3.1 Introducción

Un producto de software no puede alcanzar la calidad requerida sin pasar primero por un proceso exhaustivo de prueba y refinamiento. Cada prueba tiene su estrategia y propósito. La ausencia de defectos no puede ser asegurada a través de las pruebas, sino que solo se puede demostrar que existen errores en el software. La creación de extensos y monótonos casos de pruebas ha sido remplazada al transcurrir el tiempo por procesos automatizados. En este capítulo se aborda lo relacionado con las herramientas y procesos empleados para la validación de los subsistemas con la finalidad de garantizar un buen funcionamiento.

3.2 Pruebas unitarias y funcionales

Con base en el modelo de desarrollo de software utilizado para la creación de aplicaciones y la implementación realizada, las pruebas unitarias y funcionales tendrán en cada caso un grado mayor de relevancia sin dejar de ser importantes.

Las pruebas unitarias son dirigidas a analizar clases, métodos y porciones de código que puedan probarse de forma aislada. Estas pruebas validan a través de parámetros de entrada y parámetros de salida esperados como trabajan las funciones para cada caso en particular. Si un método varía su funcionamiento en dependencia del contexto deberá realizarse una prueba unitaria por cada caso.

El uso de las pruebas unitarias permitirá que:

Los errores sean más fáciles de localizar: bastará con ejecutar la batería de “Colecciones de pruebas”, y ver qué módulos no las pasan.

Los errores están más acotados: cuando un programa falla, muchas veces no se sabe de dónde pueden venir los problemas. Con las pruebas unitarias se consigue acotar los errores, sabiendo qué módulos no están pasando las pruebas.

CAPÍTULO 3: Validación de la Solución Propuesta

Se reducen los “efectos secundarios”: muchas veces, cuando se quiere arreglar algo bajo presión, se cometen otros errores, o no se tiene en cuenta ciertos aspectos, que hacen que el programa deje de funcionar por otro lado. Incluso a veces, es más peligroso arreglar un error que dejarlo como está, en el sentido de que se puede subsanar el error, pero generar otros distintos. Al aplicar las pruebas unitarias es más fácil controlar el programa pues se asegura que todo funciona tal y como se esperaba.

Se da más seguridad al programador: normalmente, la persona que ha programado un módulo no es la misma que la que debe corregir sus errores. Esto crea una sensación de inseguridad al programador pues a la hora de corregir un error, no tiene la certeza de que su corrección no va a afectar a otros módulos que desconoce. Las pruebas unitarias aseguran que una corrección no repercute en otros módulos, y permite al programador centrarse en su trabajo.

Las pruebas funcionales se hacen más sencillas: la mayoría de los aspectos individuales de cada unidad ya están probados a través de las pruebas unitarias. De este modo, las pruebas funcionales deben centrarse sólo en verificar la correcta cooperación de las distintas unidades, y en los funcionamientos generales del programa. [25]

Para la realización de pruebas unitarias pueden encontrarse una gran variedad de frameworks como el PHPUnit y el SimpleTest para solo citar algunos. Symfony posee su propio framework para pruebas llamado Lime, este se basa en la librería Test::More de Perl y es compatible con TAP(Test Anything Protocol) provocando que los resultados se muestren según este formato facilitando la lectura de los mismos.

Lime no solo proporciona soporte para pruebas unitarias sino que posee ventajas como:

- Las pruebas y los resultados son sencillos de leer.
- Se encuentra escrito en PHP y consta solamente de un archivo llamado lime.php y no posee dependencias.
- Se resalta en colores la información importante.

Las pruebas funcionales validan partes de las aplicaciones. Estas pruebas simulan la navegación del usuario, realizan peticiones y comprueban los elementos de la respuesta, tal y como lo haría manualmente un usuario para validar que una determinada acción hace lo que se supone que tiene que

hacer. En las pruebas funcionales, se ejecuta un escenario correspondiente a lo que se denomina un "caso de uso". [26]

El `sfBrowser` de `Symfony` actúa como un navegador que está accediendo a una aplicación, pero no requiere de un servidor web real. Este `sfBrowser` permite el acceso directo a los objetos que forman cada petición (el objeto petición, el objeto sesión, el objeto contexto y el objeto respuesta). `Symfony` 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 métodos útiles para los asertos, verificaciones y esperas.

3.3 Automatización de pruebas

Probar aplicaciones web resulta un proceso de gran esfuerzo. Crear casos de pruebas que certifiquen la veracidad de los algoritmos empleados es a menudo una tarea molesta, repetitiva y que debe incluir además ejecutar y analizar los resultados. Una aplicación puede variar o agregar requisitos, a lo que se añade la refinación continua del código, estas situaciones traen consigo la posibilidad de la aparición de errores. Por las razones antes expuestas la automatización de pruebas resulta muy útil para la creación de un sistema robusto.

La utilización del tipo de pruebas planteado ayuda al desarrollador a crear las mismas en un formato estándar y que puede ser procesado por un framework con esta finalidad.

El modo de funcionamiento de la automatización se basa en comparar un resultado con la salida esperada para ese método. Dicho de otra forma, se toman en cuenta los "asertos" (del inglés, "assertions"). El valor que toma como salida un aserto es `true` o `false`, lo que significa si la prueba tuvo éxito o falló.

3.4 Herramientas para pruebas

Cuando se trata de probar las interacciones del lado del cliente resultan insuficientes las herramientas que posee el `Symfony` para la realización de pruebas unitarias y funcionales. Lo anterior se sustenta en que estas técnicas no pueden modelar completamente el comportamiento de `JavaScript`. Al definirse en

CAPÍTULO 3: Validación de la Solución Propuesta

los subsistemas Web y Transferencia de la PTARTV interacciones complejas de AJAX, resulta necesario reproducir exactamente los eventos y luego ejecutar los scripts.

Una respuesta a las problemáticas anteriores lo constituye el uso de Selenium, que no es más que un framework de pruebas escrito en lenguaje JavaScript. Esta herramienta permite simular las acciones en la página como si fueran hechas por un usuario. Las pruebas de Selenium consisten en una página HTML, con una tabla de tres columnas: comando, destino y valor.

```
3 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
4 <head profile="http://selenium-ide.openqa.org/profiles/test-case">
5 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
6 <link rel="selenium.base" href="" />
7 <title>ReporteConversion</title>
8 </head>
9 <body>
10 <table cellpadding="1" cellspacing="1" border="1">
11 <thead>
12 <tr><td rowspan="1" colspan="3">ReporteConversion</td></tr>
13 </thead><tbody>
14 <tr>
15     <td>open</td>
16     <td>/conversion/index</td>
17     <td></td>
18 </tr>
19 <tr>
20     <td>click</td>
21     <td>link=> Reporte de Conversión</td>
22     <td></td>
23 </tr>
24 <tr>
25     <td>verifyTextPresent</td>
26     <td>Reportes de las medias convertidas en la plataforma</td>
27     <td></td>
28 </tr>
```

Figura 26. Fragmento de caso de prueba desarrollado con Selenium

Dado que las pruebas en Selenium son construidas usando el lenguaje HTML resultaría sumamente incomodo la creación de cada prueba de manera manual. Selenium posee una extensión para el navegador web Firefox que ofrece la posibilidad de grabar las acciones ejecutadas en una página, añadiendo los comandos necesarios para la prueba y guardándolas como un archivo HTML. Este

CAPÍTULO 3: Validación de la Solución Propuesta

fichero generado anteriormente se añade a la TestSuite.html en la cual se pueden agrupar las pruebas del Selenium para su posterior ejecución.

```
53 <body onload="filterTestsForBrowser()">
54
55   <table id="suiteTable"    cellpadding="1"
56     cellspacing="1"
57     border="1"
58     class="selenium">
59     <tbody>
60       <tr><td><b>Test Suite</b></td></tr>
61
62
63       <tr><td><a href="./TestPublicarNoticia.html">PublicarNoticia</a></td></tr>
64       <tr><td><a href="./TestAdicionarNoticia.html">AdicionarNoticia</a></td></tr>
65       <tr><td><a href="./ReporteConversion.html">TestReporteConversion</a></td></tr>
66       <tr><td><a href="./TestConvertirMedia2.html">TestConvertirMedia</a></td></tr>
67       <tr><td><a href="./TestAutenticarUsuario.html">TestAutenticarUsuario</a></td></tr>
68
69     </tbody>
70   </table>
71
72   <br />
73   <em>Not supported in this browser</em>
74   <table id="skippedTests" cellpadding="1"
75     cellspacing="1"
76     border="1"
77     class="selenium">
78     <tbody>
79       <tr><td><b>Skipped Tests</b></td></tr>
80     </tbody>
81   </table>
82
83 </body>
84 </html>
85
```

Figura 27. Fragmento del TestSuite.html

CAPÍTULO 3: Validación de la Solución Propuesta

La **figura 28** 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.

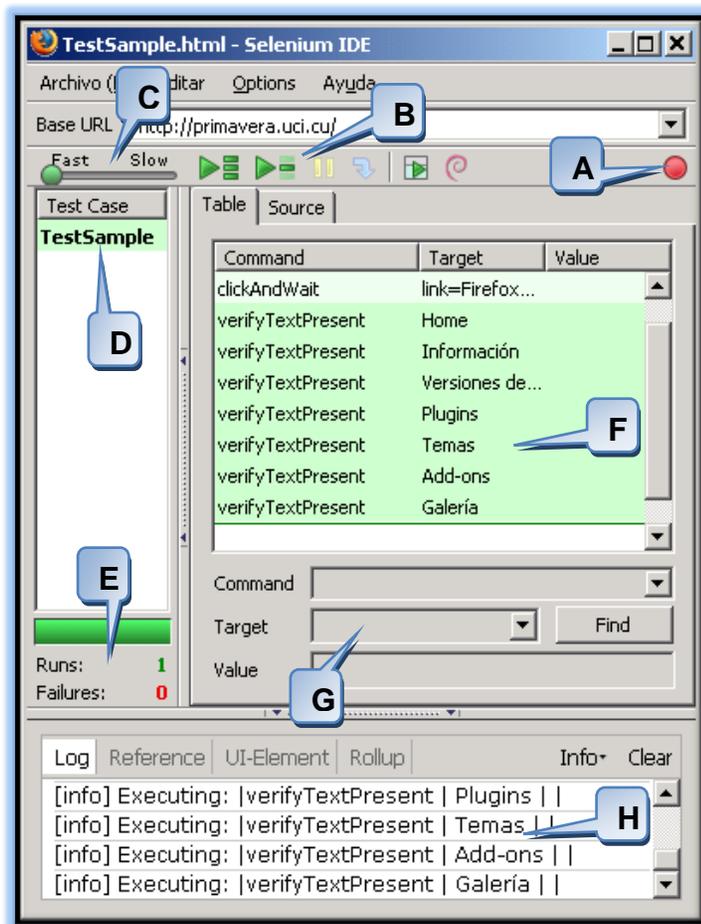


Figura 28. Interfaz gráfica del plugin Selenium

3.5 Diseño de pruebas

Con el objetivo de validar las funcionalidades y algoritmos utilizados para el desarrollo de los subsistemas, se realizan pruebas funcionales apoyadas en el uso del Selenium y las ventajas que el mismo ofrece.

A: Graba las acciones de la navegación.

B: Facilita que la prueba presente en la suite pueda ejecutarse paso a paso o de una vez. Puede además pausar o continuar las pruebas.

C: Permite al usuario manejar la velocidad con la que se ejecutará la prueba, facilitando su visibilidad.

D: Visualiza en forma de lista las pruebas presentes en la suite.

E: Informa el número de pruebas realizadas: satisfactorias (verde) y fallidas (roja).

F: Datos que conforman la prueba creada con presentes en el código de cada una.

G: Brinda la posibilidad de cambiar valores de una prueba en tiempo de ejecución.

H: Muestra los eventos y sucesos de cada prueba.

CAPÍTULO 3: Validación de la Solución Propuesta

A continuación se relacionan algunos casos de pruebas implementados con la ayuda de la herramienta mencionada.

Caso de prueba #1

Caso de prueba	Autenticar Usuario
Condiciones.	El usuario debe estar registrado en la base de datos como usuario del sistema. La contraseña de un usuario registrado debe corresponder con la esperada.
Datos de entrada	Usuario registrado y contraseña.
Resultados esperados	1. En caso de ingresar la contraseña correcta para un usuario registrado, el sistema debe mostrar la página de inicio del módulo determinado.
	2. Si se ingresa una contraseña incorrecta el sistema debe mostrar un texto de error y brindar la posibilidad de volver a ingresar sus datos.
Resultados obtenidos	1. La prueba es satisfactoria. Al autenticarse con un usuario registrado y la contraseña correcta, el sistema se redirecciona hacia la pagina del módulo deseado.
	2. La prueba es satisfactoria. Al autenticarse con un usuario registrado y la contraseña es incorrecta, el sistema muestra un mensaje de error diciendo que la contraseña es incorrecta y da la posibilidad de ingresar nuevamente los datos.
Observaciones	Al realizar la prueba con la herramienta Selenium se obtuvo el resultado correcto.

CAPÍTULO 3: Validación de la Solución Propuesta

TestAutenticarUsuario		
open	/login/index	
assertValue	ext-comp-1003	
assertValue	ext-comp-1004	
type	ext-comp-1003	Dunier
type	ext-comp-1004	
click	ext-gen22	
click	ext-gen82	
type	ext-comp-1004	dunier
click	ext-gen22	

Selenium TestRunner

Execute Tests

Fast
Slow

Highlight elements

Elapsed: 00:11

Tests	Commands
1 run	2 passed
0 failed	0 failed
	0 incomplete

Tools

View DOM
Show Log

Figura 29. Resultados del caso de prueba #1

Caso de prueba #2

Caso de prueba	Convertir Media
Condiciones	<p>El usuario debe estar registrado en la base de datos y su rol debe ser el de conversor.</p> <p>El servidor ftp debe encontrarse habilitado para que se puedan copiar las medias.</p> <p>En el servidor deben existir medias.</p>
Datos de entrada	Medias a convertir.
Resultados esperados	<ol style="list-style-type: none"> 1. Las medias quedarán correctamente convertidas en el formato mp4. 2. Se eliminarán las medias de la carpeta Conversión y pasarán convertidas a la carpeta Convertido.
Resultados obtenidos	<ol style="list-style-type: none"> 1. La prueba es satisfactoria. Las medias quedaron correctamente convertidas en el formato mp4, se eliminaron de la carpeta Conversión y se almacenaron en la carpeta Convertido.
Observaciones	La prueba con la herramienta Selenium obtuvo un resultado correcto.

CAPÍTULO 3: Validación de la Solución Propuesta

TestConvertirMedia		
open	/conversion/index	
click	link=» Medias en Conversión	
verifyTable	//div[@id='ext-gen70']/div/table.0.0	Nombre de la Media
verifyTextPresent	Nombre de la Media	
verifyTextPresent	47_3_NTSC_DV	
verifyTextPresent	Convertir Cancelar	
click	ext-gen99	
verifyTextPresent	Fin de la conversión	
verifyTable	//div[@id='ext-gen30']/div/table.0.0	Media Convertida
verifyTextPresent	Media Convertida	
verifyTextPresent	47_3_NTSC_DV	

Selenium TestRunner

Execute Tests

Fast
Slow

Highlight elements

Elapsed: 00:41

Tests	Commands
1 run	8 passed
0 failed	0 failed
	0 incomplete

Tools

View DOM
Show Log

Figura 30. Resultados del caso de prueba #2

Caso de prueba #3

Caso de prueba.	Reporte de conversión
Condiciones.	Deben haberse convertido las medias
Datos de entrada	Datos de las medias convertidas.
Resultados esperados	1. Reporte con la lista de todas las medias convertidas hasta la fecha.
Resultados obtenidos	1. La prueba es satisfactoria. Se le informa al usuario el reporte de las medias convertidas hasta la fecha.
Observaciones	Al realizar la prueba con la herramienta Selenium se obtuvo el resultado correcto.

CAPÍTULO 3: Validación de la Solución Propuesta

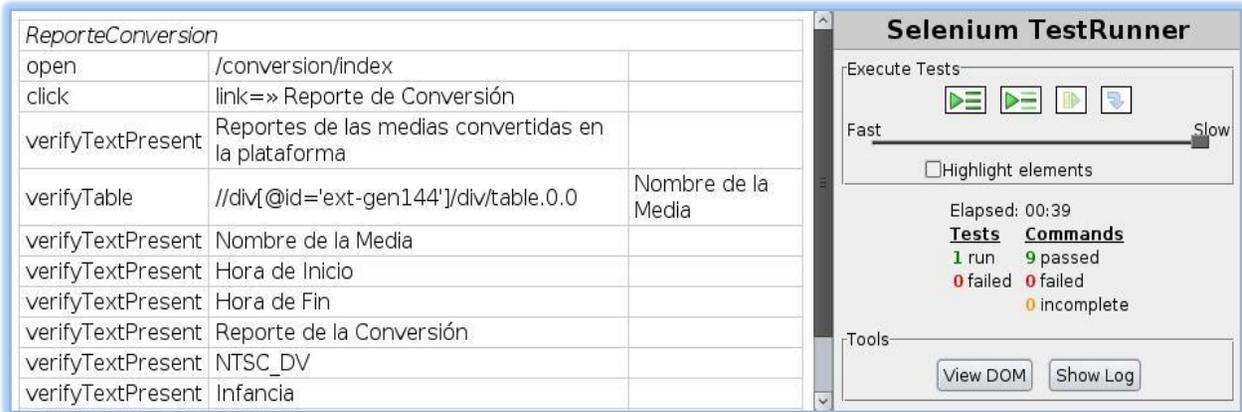


Figura 31. Resultados del caso de prueba #3

Caso de prueba #4

Caso de prueba	Adicionar Noticia
Condiciones	El usuario debe estar registrado en la base de datos y su rol ser el de Administrador Web.
Datos de entrada	Noticias redactadas.
Resultados esperados	1. La noticia quedará correctamente adicionada a la base de datos y se mostrarán sus datos en el panel que contiene las noticias del sistema.
Resultados obtenidos	1. La prueba es satisfactoria. Se le informa al usuario que la noticia se adicionó correctamente a la base de datos y se muestra en el panel que contiene las noticias del sistema.
Observaciones	Al realizar la prueba con la herramienta Selenium se obtuvo el resultado correcto.

CAPÍTULO 3: Validación de la Solución Propuesta

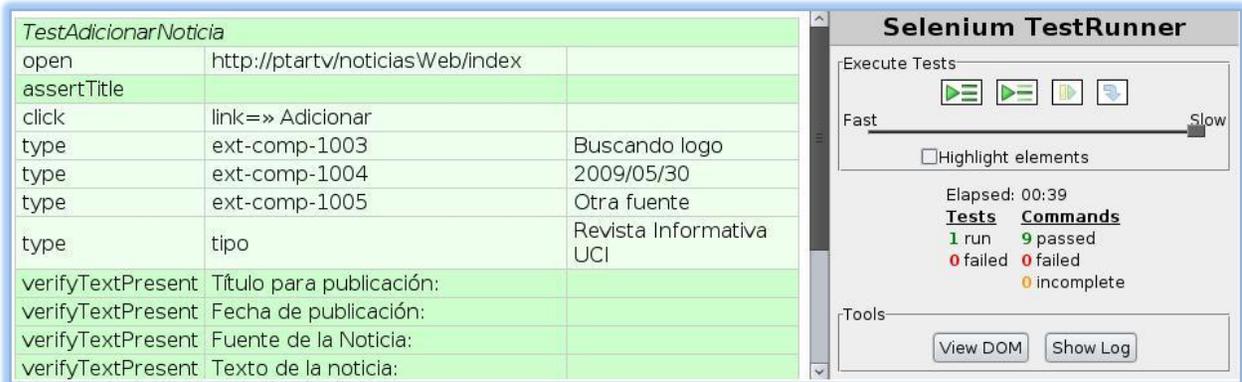


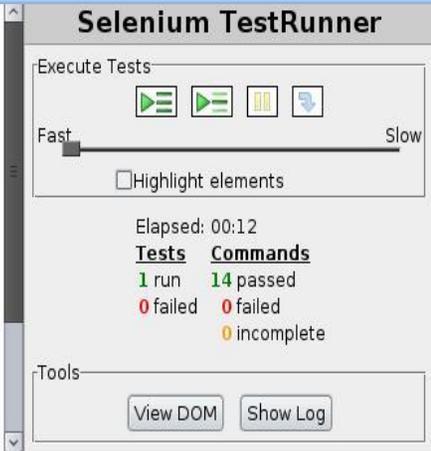
Figura 32. Resultados del caso de prueba #4

Caso de prueba #5

Caso de prueba.	Publicar Noticia
Condiciones.	Deben existir noticias almacenadas en la base de datos.
Datos de entrada	Noticias redactadas.
Resultados esperados	1. La noticia quedará publicada y se visualizará correctamente.
Resultados obtenidos	1. La prueba es satisfactoria. Se le informa al usuario que la noticia se publicó correctamente y se visualiza en la página de inicio de la PTARTV.
Observaciones	Al realizar la prueba con la herramienta Selenium se obtuvo el resultado correcto.

CAPÍTULO 3: Validación de la Solución Propuesta

TestPublicarNoticia	
open	http://ptartv/noticiasWeb/index
assertTitle	
verifyTextPresent	Noticias a publicar en el entorno Web
verifyTextPresent	Título
verifyTextPresent	Redactor
verifyTextPresent	Fecha
verifyTextPresent	Publicada
verifyTextPresent	Primero de Mayo
verifyTextPresent	yInunez
verifyTextPresent	2009-05-01
verifyTextPresent	Publicar
verifyTextPresent	-



The Selenium TestRunner interface displays the following information:

- Execute Tests:** Includes a play button, a list icon, a bar chart icon, and a refresh icon.
- Speed Control:** A slider from 'Fast' to 'Slow' with a 'Highlight elements' checkbox.
- Elapsed:** 00:12
- Tests:** 1 run
- Commands:** 14 passed, 0 failed, 0 incomplete
- Tools:** View DOM, Show Log

Figura 33. Resultados del caso de prueba #5

3.6 Conclusiones

En este capítulo se prestó atención a las características relacionadas con las pruebas para la fase de implementación y a la herramienta utilizada para el desarrollo de las mismas. Para lograr que los subsistemas sean lo más robustos posible es vital que sean sometidos a las pruebas pertinentes que confirmen las funcionalidades previstas. Resulta importante explicar que los casos de pruebas presentados no son los únicos diseñados para las aplicaciones sino que el objetivo es plasmar la forma de la realización de los mismos y el funcionamiento del Selenium como framework de pruebas.

CONCLUSIONES

Durante el desarrollo del trabajo de diploma y el tiempo dedicado a la investigación, implementación y prueba de la solución propuesta, se adquirieron conocimientos necesarios para el desarrollo del sistema y fundamentales para la formación profesional del autor. Luego de la realización de las actividades anteriores se logró:

- ✓ Obtener una valoración de las tendencias y tecnologías actuales que permitieran la correcta selección de entornos de desarrollo integrado, frameworks y lenguajes de programación.
- ✓ Seleccionar las herramientas necesarias para el trabajo con video a demanda y streaming en software libre.
- ✓ Estructurar, a partir de los Diagramas de componentes, el modelo de implementación de la PTARTV.
- ✓ Implementar prototipos funcionales de los subsistemas Web y Transferencia que cumplen con los requerimientos establecidos para el rol de implementador.
- ✓ Desarrollar y documentar los casos de prueba necesarios para verificar los algoritmos empleados.

Se confirma haber dado cumplimiento al objetivo general trazado desde el inicio del trabajo. La afirmación anterior se sustenta en el logro del desarrollo de sistemas automatizados para la transferencia y visualización de contenido multimedia en la PTARTV, lo cual garantiza mayor rapidez y mejor funcionamiento en los procesos de transferencia, codificación y visualización de la información.

RECOMENDACIONES

Al finalizar el presente trabajo se recomienda:

- ✓ Dar continuidad al desarrollo y soporte de los subsistemas desarrollados.
- ✓ Agregar funcionalidades que permitan la gestión y configuración del diseño gráfico de la Plataforma de Transmisión Abierta para Radio y Televisión y específicamente del subsistema Web por ser la interfaz visible a los usuarios. Lo anterior permitiría un mayor grado de personalización del sistema.
- ✓ El uso de un único servidor streaming en el Polo de Video y Sonido Digital de la Facultad 9 y también dentro de la Universidad que sea robusto y satisfaga las necesidades de las aplicaciones a desarrollar.
- ✓ La implementación de las deficiencias del VLC como servidor streaming para que sea viable su utilización pues cuenta con funcionalidades muy robustas.

REFERENCIAS BIBLIOGRÁFICAS

1. **Instituto Nacional de Estadística e Informática.** [En línea] 2009 [Citado el: 22 de enero de 2009] [Disponible en: <http://www1.inei.gov.pe/biblioineipub/bancopub/inf/Lib5038/defi.HTM>]
2. **Arsys Programación Web.** [En línea] 2007. [Citado el: 23 de enero de 2009] [Disponible en: <http://www.arsys.es/soporte/programacion/windows.htm>]
3. **Méndez, Néstor.** *Conceptos de Arquitectura Cliente - Servidor.* 2007. [Citado el: 23 de enero de 2009]
4. **Orth, Steve.** 4DHispano. [En línea] 2008 [Citado el: 23 de enero de 2009] [Disponible en: <http://www.4dhispano.com/products/brochures/pdf/4dweb20pack-es.pdf>]
5. **Garrett, Jesse James.** *Ajax: Un Nuevo acercamiento a las Aplicaciones Web.* [En línea] 18 de febrero de 2005. [Citado el: 23 de enero de 2009] [Disponible en: <http://mmugcba.com.ar/foro/viewtopic.php?id=36>]
6. **García Pañeda, Xabiel y Marín Prendes, Ignacio y Melendi Palacio, David.** *Introducción a los servicios de audio y video streaming.* Universidad de Oviedo. Departamento de Informática. [En línea] [Citado el: 24 de enero de 2009] [Disponible en: http://www.it.uniovi.es/old/material/cursos/streaming_EU_122004/Intro.pdf]
7. **Historia de la Informática.** *Que son los paradigmas.* Entorno Virtual de Aprendizaje. [En línea] 2008. [Citado el: Febrero, 2009] [Disponible en: http://teleformacion.uci.cu/mod/resource/view.php?id=9287&subdir=/Tema_1]
8. **Lenguajes de Programación.** [En línea] 2009. [Citado el: 22 de febrero de 2009] [Disponible en: <http://www.lenguajes-de-programacion.com/programacion-orientada-a-objetos.shtml>]
9. **Apunte Paradigmas de la Programación.** [En línea] 2009. [Citado el: 22 de febrero de 2009] [Disponible en: <http://www.cuys.com.ar>]

10. **El lenguaje HTML.** [En línea] 2009. [Citado el: 24 de febrero de 2009] [Disponible en: http://www.xpps.net/contenido_xppsnet/areatec/HMTL.pdf]
11. **Álvarez, M. A.** *Breve Historia de PHP.* [En línea] 2009. [Citado el: 24 de febrero de 2009] [Disponible en: <http://www.desarrolloweb.com/articulos/436.php>]
12. **Van Der Henst, Christian y Heredia Santos, Herminio** *¿Qué es el PHP?* [En línea] 2009 [Citado:24 de febrero de 2009] [Disponible en: <http://www.maestrosdelweb.com/editorial/phpintro>]
13. **Hurtado Jara, Omar.** *Evolución y Orientaciones de Patrones.* [Citado el: 28 de febrero de 2009]
14. **Alexander, Christopher y Ishikawa, Sara y Silverstein, Murray y Jacobson, Max y Fiksdahl-King, Ingrid y Angel, Shlomo.** *A Pattern Language: Towns/ Building/ Construction.* Oxford University Press. 1977. [Citado el: 28 de febrero de 2009]
15. **Centro Informático Científico de Andalucía Consejería de Innovación, Ciencia y Empresa.** *Arquitectura Modelo/Vista/Controlador* [En línea] 2009 [Citado: 6 de marzo de 2009] [Disponible en: http://www.cica.es/formacion/JavaTut/Apendice/arq_mvc.html]
16. **DotBlog.** *Symfony: framework por excelencia para PHP* [En línea] 2009 [Citado: 9 de marzo de 2009] [Disponible en: <http://www.dotconsulting.com.ar/blog/?p=6>]
17. **Gómez Ballester, Eva.** *Bases de Datos 1* [En línea] 2009 [Citado: 9 de marzo de 2009] [Disponible en: <http://www.alu.ua.es/j/jmr36/Conectate/Base%20Datos/Apuntes2006.pdf>]
18. **Computer Audio Video Systems Integrator.** *¿Qué es un Sistema Gestor de Bases de Datos o SGBD?* [En línea] 2009 [Citado: 9 de marzo de 2009] [Disponible en: <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>]
19. **Equipo de desarrollo de PostgreSQL.** *Manual de Usuario de PostgreSQL.* [En línea] 2009 [Citado: 10 de marzo de 2009] [Disponible en: <http://es.tldp.org/Postgresql-es/web/navegable/user/user.html>]

20. **Apuntes de Ingeniería del software. Sistemas Informáticos. Universidad de Murcia.** *Metodologías de desarrollo de software* [En línea] 2009 [Citado: 11 de marzo de 2009] [Disponible en: <http://www.um.es/docencia/barzana/IAGP/lagp2.html>]
21. **Mastermagazine.** *Definición de Apache* [En línea] 2009 [Citado: 14 de marzo de 2009] [Disponible en: <http://www.mastermagazine.info/termino/3866.php>]
22. **Diseño de páginas Web, Alojamiento, Posicionamiento y Dominios en Alicante, España.** *¿Qué es un Sistema Operativo?* [En línea] 2009 [Citado: 14 de marzo de 2009] [Disponible en: <http://www.masadelante.com/faq-sistema-operativo.htm>]
23. **Metodología de la Red Nacional de Integración y Desarrollo de Software Libre.** [En línea] 2009 [Citado: 5 de mayo de 2009] [Disponible en: <http://merinde.rinde.gob.ve>]
24. **Letelier Torres, Patricio.** *Desarrollo de Software Orientado a Objeto usando UML.* Departamento Sistemas Informáticos y Computación. Universidad Politécnica de Valencia. [Citado el: 11 de mayo de 2009].
25. **Comunidades de wikis libres para aprender.** *Ventajas de las pruebas unitarias.* [En línea] 2009 [Citado: 29 de mayo de 2009] [Disponible en: <http://www.wikilearning.com>]
26. **Symfony la guía definitiva.** *Capítulo 15. Pruebas unitarias y funcionales.* [En línea] 2009 [Citado: 29 de mayo de 2009] [Disponible en: http://www.librosweb.es/symfony_1_0/capitulo15/pruebas_funcionales.html]

BIBLIOGRAFÍAS

1. **Lima Córdova, Orestes y Suárez Contreras, Eridniel.** *Diseño de una Plataforma de Transmisión Abierta para la Radio y la Televisión.* 2008. [Consultado el: Enero, 2009]
2. **Hernández Bustio, José Andrés.** *Desarrollo del Canal Informativo del Ministerio del Poder Popular para la Energía y Petróleo de Venezuela: Subsistema de Transmisión.* 2008. [Consultado el: Enero, 2009]
3. **Departamento de Técnicas de Programación.** *Introducción a la Programación Cliente - Servidor.* Entorno Virtual de Aprendizaje. [En línea] 2009. [Consultado el: Enero, 2009] [Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=10530>]
4. **Pérez Abril, Enrique.** Zarza. [En línea] [Consultado el: Enero, 2009] [Disponible en: <http://zarza.usal.es/~fgarcia/doctorado/iweb/05-07/Trabajos/CalidadAplicsWeb.pdf>]
5. **Barral Precedo, David y García Chao, Asís.** [En línea] 2006. [Consultado el: Enero, 2009] [Disponible en: http://stuff.gpul.org/2006_web/doc/2006_JWEB_03_AJAX.pdf]
6. **García Pañeda, Xabiel y Marín Prendes, Ignacio y Melendi Palacio, David.** *Introducción a los servicios de audio y video streaming.* Universidad de Oviedo. Departamento de Informática. [En línea] [Consultado el: Enero, 2009] [Disponible en: http://www.it.uniovi.es/old/material/cursos/streaming_EU_122004/Intro.pdf]
7. **Instituto Nacional de Estadística e Informática.** [En línea] 2009. [Consultado el: Febrero, 2009] [Disponible en: <http://www1.inei.gob.pe/biblioineipub/bancopub/inf/Lib5038/ven1.HTM>]
8. **Ext JS.** [En línea] 2009. [Consultado el: Febrero, 2009] [Disponible en: <http://extjs.com/products/extjs>]
9. **Apunte Paradigmas de la Programación.** [En línea] 2009. [Consultado el: Febrero, 2009] [Disponible en: <http://www.cuys.com.ar>]

10. **Miranda Perea, Favio E.** *Lenguajes de Programación y sus Paradigmas*. 2008. [Consultado el: Febrero, 2009]
11. **Departamento de Sistemas y Computación del Instituto Tecnológico La Paz.** *Programación estructurada*. [En línea] 2009 [Consultado el: Febrero, 2009] [Disponible en: http://sistemas.itlp.edu.mx/tutoriales/pascal/u1_1_4.html]
12. **Salvador, J. R.** *Lamp y Eclipse una configuración para programar sitios web dinámicos con Linux*. [En línea] 2008. [Consultado el: Febrero, 2009] [Disponible en: <http://unreal-hosting.net/lampyeclipse.pdf>]
13. **Tecnología orientada a procesos de negocio.** [En línea] 2008. [Consultado el: Marzo, 2009] [Disponible en: <http://www.proactiva-calidad.com/java/patrones/mvc.html>]
14. **Equipo de desarrollo de PostgreSQL.** *Manual de Usuario de PostgreSQL*. [En línea] 2009 [Consultado el: 10 de marzo de 2009] [Disponible en: <http://es.tldp.org/Postgresql-es/web/navegable/user/user.html>]
15. **Historia de la Informática.** *Introducción al desarrollo del software*. Entorno Virtual de Aprendizaje. [En línea] 2009. [Consultado el: 10 de marzo de 2009] [Disponible en: http://teleformacion.uci.cu/mod/resource/view.php?id=9287&subdir=/Tema_2]
16. **Codigolinea.** *Estilo de programación y convención de nombres*. [En línea] 2009. [Consultado el: 12 de mayo de 2009] [Disponible en: <http://codigolinea.com/>]

ANEXOS

Anexo I. La Web antes de AJAX

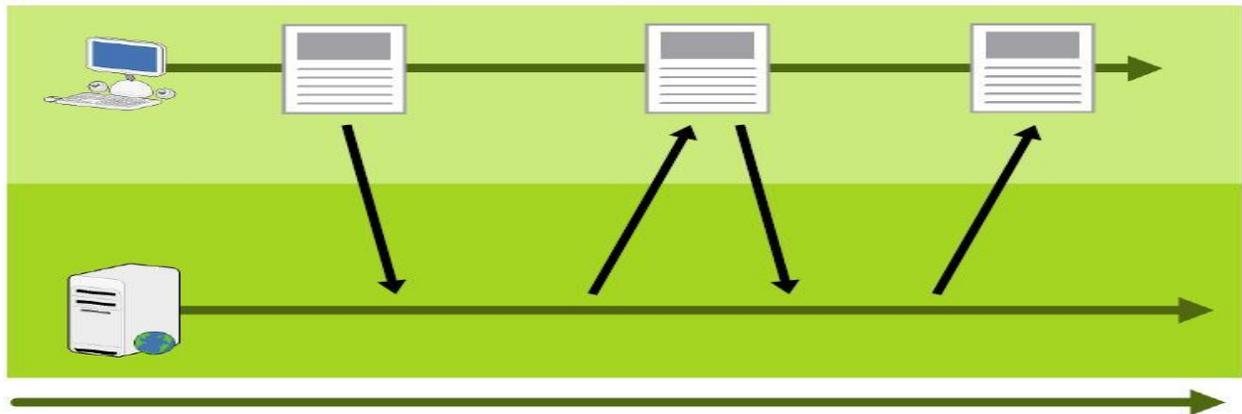


Figura 34. La Web antes de AJAX

Anexo II. La Web con AJAX

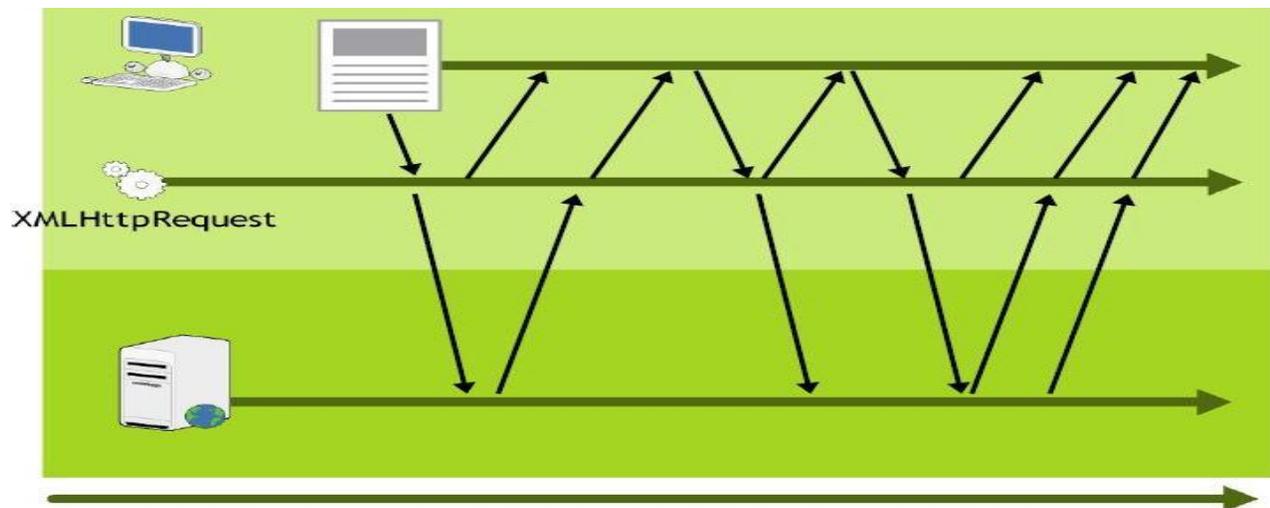


Figura 35. La Web con AJAX

Anexo III. Principales funcionalidades de los framework para php

Framework	PHP4	PHP5	MVC ¹	Multiple DB's ²	ORM ³	DB Objects ⁴	Templates ⁵	Caching ⁶	Validation ⁷	Ajax ⁸	Auth Module ⁹	Modules ¹⁰
Zend Framework	-	✓	✓	✓	-	✓	-	✓	✓	-	-	✓
CakePHP	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	-
Symfony Project	-	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	-
Seagull Framework	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
WACT	✓	✓	✓	✓	-	✓	✓	-	✓	-	-	-
Prado	-	✓	-	✓	-	-	✓	✓	✓	✓	✓	✓
PHP on TRAX	-	✓	✓	✓	✓	✓	-	-	✓	✓	-	-
ZooP Framework	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓
eZ Components	-	✓	-	✓	-	✓	✓	✓	✓	-	-	✓
CodeIgniter	✓	✓	✓	✓	-	✓	✓	✓	✓	-	-	✓

Figura 36. Principales funcionalidades de los framework para php

Anexo IV. Proceso Unificado de Desarrollo de Software (RUP)

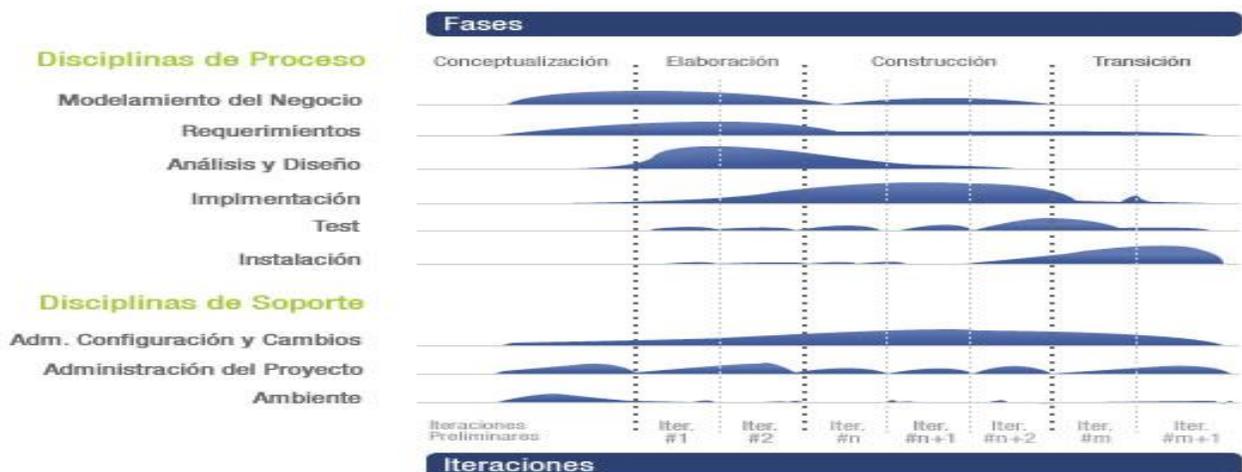


Figura 37. Proceso Unificado de Desarrollo de Software (RUP)

GLOSARIO

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

Audio: Señal, técnica o sistema electrónico relacionado con la grabación, reproducción y transmisión del sonido que normalmente se encuentra acotado al rango de frecuencias audibles para los seres humanos.

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

Conversión: Cuando se refiere a la conversión de video o audio, puede interpretarse como el hecho de tomar un fichero y cambiar su formato de codificación por otro seleccionado.

CSS: *Cascading Style Sheets* (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* (Modelo en Objetos para la representación de Documentos), 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.

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.

Multimedia: Se aplica a cualquier objeto que utilice simultáneamente diferentes formas de contenido informativo como texto, sonido, imágenes y video para informar o entretener al usuario.

Multiplataforma: Se utiliza el término para denominar a los programas, lenguajes de programación u otra clase de software que pueden brindar sus prestaciones funcionando sobre diversas combinaciones de hardware y software.

Protocolo: En informática es un método estándar que permite la comunicación entre procesos. Comprende un conjunto de reglas y procedimientos que deben respetarse para el envío y la recepción de datos a través de una red.

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

RTP: *Real Time Transport Protocol* (Protocolo de Transporte de Tiempo real), es un protocolo de nivel de sesión que se utiliza para transmitir información en tiempo real la cual puede ser audio y vídeo.

RTSP: *Real Time Streaming Protocol* (Protocolo de Flujo de Datos en Tiempo Real), es un protocolo no orientado a conexión y que en lugar de esto el servidor mantiene una sesión asociada a un identificador. RTSP establece y controla uno o muchos flujos sincronizados de datos pudiendo ser audio y video.

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

Smartphones: Teléfono inteligente que puede comunicarse a través de Wi-Fi, bluetooth, conexión a internet, envío de mensajería, e-mails, etc.

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.

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

Transmisión: Emitir la señal de radio o televisión desde una estación emisora a otra que actúa como receptora de la misma.

Transmisión Abierta: Señal de libre acceso por la cual no es necesario pagar para consumirla.

Video: Es una tecnología de captura electrónica, tratamiento, almacenamiento, transmisión, y reconstrucción de una sucesión de imágenes y sonidos que representan escenas en movimiento.

XHTML: *Extensible Hypertext Markup Language* (lenguaje extensible de marcado de hipertexto), es el lenguaje de marcado diseñado para sustituir a HTML como estándar para las páginas web. En la versión 1.0, XHTML es la versión XML de HTML, por lo que básicamente posee las mismas funcionalidades pero cumple las especificaciones más estrictas de XML.

XML: *Extensible Markup Language* (lenguaje extensible de marcado), es un metalenguaje extensible de etiquetas creado por el World Wide Web Consortium (W3C).

XMLHttpRequest: *Extensible Markup Language / Hypertext Transfer Protocol*, es empleado para realizar peticiones HTTP y HTTPS a servidores WEB. Para los datos transferidos se usa cualquier codificación de las siguientes: texto plano, XML, JSON (JavaScript Object Notation) y HTML.

XSLT: *Extensible Stylesheet Language Transformations* (Transformaciones XSL), es un estándar del W3C que presenta una forma de transformar documentos XML en otros que pueden o no ser XML. Las transformaciones del documento se llevan a cabo utilizando una o varias reglas de plantilla.