



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 9

TÍTULO: Implementación del subsistema Gestión y Presentación de Contenidos de la Plataforma VideoWeb.

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

AUTOR: Yakelín de la Caridad Tejeda Pérez

TUTOR: Ing. Yunier Albrecht Delgado

CO-TUTOR(ES): Ing. Frank Torres Rodríguez

Ing. Angel Dayán Marín Abreu

Ciudad de La Habana, junio 2010.

AÑO 50 DE LA REVOLUCIÓN.

DATOS GENERALES

Generales de la investigación:

Título: Implementación del subsistema Gestión y Presentación de Contenido de la Plataforma VideoWeb.

Generales del diplomante:

Nombre y Apellidos: Yakelín de la Caridad Tejeda Pérez.

Sexo: F

Grupo: 9505

Correo electrónico: yltejeda@estudiantes.uci.cu.

Generales del tutor:

Nombre y apellidos: Yunier Albrecht Delgado.

Categoría docente: Instructor recién graduado

Grado científico:

Centro de trabajo: Universidad de las Ciencias Informáticas.

Cargo: Profesor.

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

Año de graduación: 2007

Institución donde se graduó: Universidad de las Ciencias Informáticas.

Correo electrónico: yalbrecht@uci.cu.

Teléfono del trabajo:

Teléfono particular: 835-8902

Generales del co-tutor:

Nombre y apellidos: Angel Dayán Marín Abreu.

Categoría docente: Instructor recién graduado

Grado científico:

Centro de trabajo: Universidad de las Ciencias Informáticas.

Cargo: Profesor.

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

Año de graduación: 2009

Institución donde se graduó: Universidad de las Ciencias Informáticas.

Correo electrónico: admarin@uci.cu.

Teléfono del trabajo: 835-8284

Teléfono particular: 835-8902

Generales del co-tutor:

Nombre y apellidos: Frank Torres Rodríguez.

Categoría docente: Instructor recién graduado

Grado científico:

Centro de trabajo: Universidad de las Ciencias Informáticas.

Cargo: Profesor.

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

Año de graduación: 2009

Institución donde se graduó: Universidad de las Ciencias Informáticas.

Correo electrónico: ftrodriguez@uci.cu.

Teléfono del trabajo: 835-8284

Teléfono particular: 835-8902

AGRADECIMIENTOS

A mi mamita y a mi papito por todo su amor, comprensión y esfuerzo realizado en estos 5 años, gracias a ellos he podido llegar donde estoy. Gracias por sus consejos oportunos y por siempre tener confianza en mí y darme ánimos para seguir adelante.

A mi hermanita querida, por ser la mejor hermana del mundo, siempre tan cariñosa y bondadosa. Has sido un punto de apoyo muy importante para mí y un maravilloso ejemplo a seguir. A mi Titi, que estuvo siempre pendiente de mí y a Rober, por los heladitos en todos sus viajes. Al amor de mi vida Jeans, por soportarme cuando tuve esos momentos de tanto estrés y estar ahí siempre para mí. Gracias mi cielo por toda tu comprensión y amor.

A mi Niñito, el mellizito, por ser esa persona tan especial y el mejor amigo que se puede tener. Eres el hermanito que no tuve, le agradezco a la vida por haberte conocido y permitirme compartir contigo estos 5 años de alegrías y tristezas. Te deseo lo más lindo de la vida.

A mis amigos Frank, Ángel, Yordany y Maquita, por todos los momentos buenos que compartimos en UCITVe y luego en VideoWeb. Son personas muy especiales y estoy muy feliz de haberlos conocido.

A mi tutor y amigo Yunier Albrecht por su paciencia, los ensayos antes de los cortes y la habilidad que tuvo para calmarme y darme fuerzas y ánimos para enfrentarme a cada uno de ellos. Siempre tendrás un lugar en mi corazón.

A mis co-tutores Ángel y Frank, sin su ayuda no hubiera sido posible que llegara hasta aquí. Muchísimas gracias, mi eterna gratitud hacia ustedes.

A Zulita y Annies, son muy especiales y unas magníficas amigas, estoy muy feliz por haberlas conocido. A Dayron (Dorito), llegar a conocerte y estar entre tus amigos ha sido muy lindo, eres la

AGRADECIMIENTOS

persona más bondadosa y desinteresada que he conocido jamás, por eso eres tan querido por todos, y me siento muy afortunada de haberte encontrado en mi vida. Todos ustedes son los mejores habaneros que he conocido jamás, sigan siendo así y les deseo toda la felicidad del mundo.

A los miembros del proyecto VideoWeb, en especial a Ruby que me soportó con mis quejas y escuchó en varias ocasiones la expo, dándome opiniones muy valiosas que me ayudaron a la hora de exponer y a Orlando quien me prestó muchas veces el tiempo de PC para que yo adelantara y trabajara en este documento.

A los muchachos de mi antiguo grupo Ela, José Enrique, Raciél, Darién, Los Pipis (Reinier y Alain), Lili, Margelis juntos dimos los primeros pasos aquí en la universidad, aquellos momentos quedaron grabados en mi mente.

A mis compañeros de aula de estos dos últimos años: Yvanosis que siempre tuvo un chiste y una mano amiga para mí, Dayami, Gilberto, Jean L., el Chino (Adrián), Pepe (Juan José) y todos los demás. Fue muy bueno haberlos conocido a todos.

A todos los miembros del tribunal por todos sus consejos y recomendaciones.

A la Revolución y a Fidel, por la maravillosa idea de crear esta escuela.

DEDICATORIA

Le dedico este trabajo a toda mi familia, en especial a mi mamá y a mi papá, que sin todo su apoyo no hubiera podido llegar hasta aquí. Les estaré eternamente agradecida a todos ustedes.

DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora del trabajo titulado: "Implementación del subsistema Gestión y Presentación de Contenidos de la Plataforma VideoWeb." y autorizo a la Facultad 9 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Yakelín de la Caridad Tejeda Pérez

Yunier Albrecht Delgado



VIII Jornada Científica Estudiantil

Se le otorga el presente
Diploma

A: *Yakelin Tejeda Pérez*
Rubisel Prieto Dupeyron

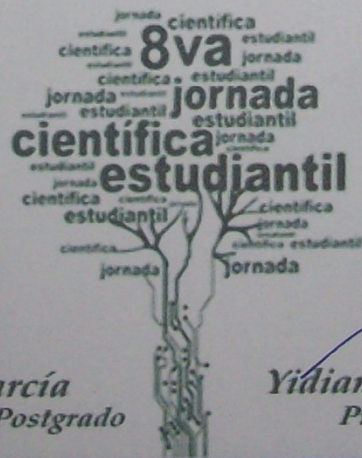


Del trabajo o actividad titulada:

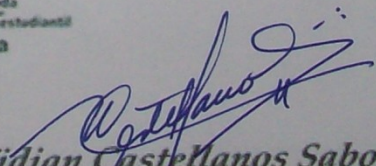
*Plataforma libre para la Transmisión de Contenidos
Audiovisuales en la Web*

por haber obtenido la condición de

Destacado



José Lavandero García
Vicerrector Investigación y Postgrado


Yidian Castellanos Saborí
Presidente FEU-UCI

Dado en La Habana a los ___ días del mes de mayo del 2010
"Año 52 de la Revolución"

DEFENSA DE TESIS

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: **Implementación del subsistema Gestión y Presentación de Contenido de la Plataforma VideoWeb.**

Autor(es): **Yakelin de la Caridad Tejeda Pérez**

El tutor del presente Trabajo de Diploma y líder de proyecto de la estudiante considera que durante el período el desarrollo de la Tesis la estudiante demostró alta responsabilidad ante las tareas. Se dio cumplimiento a los objetivos trazados con gran independencia y creatividad, logrando los resultados esperados y demostrando capacidad para apropiarse de los conocimientos necesarios para dar solución al problema planteado en la investigación. El trabajo posee alta calidad científico-técnica y es de gran importancia para el Departamento de Señales Digitales pues culmina el desarrollo de la primera versión de un producto informático contratado por un valor cercano al millón de dólares en los convenios de colaboración Cuba-Venezuela por lo que su resultado no solo representa mucho para Departamento y la Facultad 9 sino también para todos los cubanos. Como integrante fundadora del Proyecto Plataforma de VideoWeb destacar su esfuerzo durante todas las etapas del desarrollo de software y su activo desempeño cubriendo varios roles dentro del equipo de trabajo, características que le formaron y ayudaron a obtener los resultados que hoy muestra. En nombre de todo el equipo de proyecto la felicitación más que merecida y los deseos de que a partir de ahora en adelante siga cosechando éxitos.

Por todo lo anteriormente expresado considero que la estudiante está apta para ejercer como Ingeniero en Ciencias Informáticas y propongo que se le otorgue al Trabajo de Diploma, la calificación de 5 puntos.

Ing. Yunier Albrecht Delgado

24 de Junio de 2010

Firma

Fecha

RESUMEN

En la actualidad, el uso de la tecnología streaming para la transmisión de archivos multimedia por la web ha alcanzado un gran auge y aceptación entre los usuarios que utilizan este tipo de servicios. Debido a ello, muchas empresas e instituciones se han interesado en emplearla para transmitir información audiovisual dentro de sus redes internas.

Contar con una aplicación que brinde este tipo de servicios ofrece para cualquier entidad grandes oportunidades de mercado, es por ello que el proyecto VideoWeb, perteneciente al Departamento de Señales Digitales de la facultad 9 en la Universidad de Ciencias Informáticas (UCI), tiene la tarea de realizar una aplicación web sobre tecnologías libres que permita la administración, gestión y publicación de contenidos audiovisuales por la web, proporcionando un producto que ayude al país a alcanzar la independencia tecnológica que necesita y que pueda comercializarse posteriormente.

En el presente trabajo se detallan las herramientas y tecnologías que se utilizaron para el desarrollo del subsistema Gestión y Presentación de Contenidos de la Plataforma VideoWeb, así como el modelo de implementación y los diagramas de componentes que sentaron las bases para el posterior desarrollo del mismo. Finalmente se especifica el resultado arrojado por las pruebas realizadas después de la implementación de los módulos que componen el subsistema, las cuales validaron el correcto funcionamiento y cumplimiento de los requisitos establecidos para la Plataforma.

PALABRAS CLAVES: streaming, archivo multimedia, subsistema, gestión, publicación, tecnologías, Videoweb.

ÍNDICE DE FIGURAS

Figura 1. Patrón Arquitectónico Modelo - Vista – Controlador.....	14
Figura 2. Interfaz Darwing Streaming Server.....	17
Figura 3. Interfaz Mp4Live	18
Figura 4. Actividades y Artefactos del implementador.....	23
Figura 5. Modelo de Implementación.....	26
Figura 6. Subsistema de Gestión y Presentación de Contenidos.....	27
Figura 7. Paquete Gestión de Archivos Multimedia.....	28
Figura 8. Diagrama de Componentes Módulo FTPHTML.	29
Figura 9. Diagrama de componentes Módulo Tipología_AM.....	30
Figura 10. Diagrama de componentes Módulo Archivo Multimedia.	31
Figura 11. Diagrama de componentes Módulo Reproductor AM.....	32
Figura 12. Diagrama de componentes Módulo Archivo Multimedia_r	32
Figura 13. Diagrama de componentes Módulo Lista_Reproducción.	33
Figura 14. Interfaz gráfica del plugin Selenium.....	48

ÍNDICE DE TABLAS

Tabla 1. Tipos de Pruebas que se aplican al software	45
Tabla 2. Caso de Prueba: Adicionar archivo multimedia	49
Tabla 3. Caso de prueba: Crear publicación archivo multimedia	50
Tabla 4. Caso de prueba: Publicación archivo multimedia	51
Tabla 5. Caso de prueba: Dejar de publicar archivo multimedia	52
Tabla 6. Caso de prueba: Buscar archivo multimedia	53

ÍNDICE

Introducción	1
Capítulo 1. Fundamentación Teórica.....	5
Introducción.....	5
1.1 Procesos de Gestión y Presentación de contenidos en la Plataforma VideoWeb.	5
1.1.1 Descripción de los procesos de Gestión de Contenidos en la Plataforma VideoWeb.	5
1.1.2 Descripción de los procesos asociados a la Presentación de Contenidos en la Plataforma VideoWeb.	7
1.2 Tecnología para el Desarrollo de Aplicaciones Web.....	7
1.2.1 CMS Drupal.....	8
1.3 Lenguajes de programación	9
1.3.1 Lenguajes de programación y tecnologías del lado del cliente.....	9
1.3.2 Lenguajes de programación del lado del servidor.....	10
1.4 Entorno Integrado de desarrollo.....	11
1.4.1 Eclipse PDT.....	11
1.5 Sistema Gestor de Base de Datos.....	12
1.5.1 PostgreSQL.....	13
1.6 Patrón Arquitectónico Modelo Vista Controlador.....	14
1.7 Streaming de audio y video.....	15
1.7.1 Servidores Streaming.....	15
1.8 Metodología de desarrollo.....	18
1.8.1 Rational Unified Process (RUP).....	20
1.9 Lenguaje de Modelado. UML	20
1.10 Herramienta Case. Visual Paradigm	21
1.11 Servidor Web. Apache.....	21
1.12 Servidor FTP. VSFTPD	22
Conclusiones.....	22
Capítulo 2.....	23
Descripción de la solución Propuesta.....	23

Introducción.....	23
2.1 El Implementador del Sistema.....	23
2.2 Descripción del Subsistema Presentación y Gestión de Contenidos.....	24
Módulo FTPHTML.....	25
Módulo Tipología_AM.....	25
Módulo Archivo_Multimedia.....	25
Módulo Reproductor_AM.....	25
Módulo Lista_Reproducción.....	25
Módulo Archivo_Multimedia_r.....	25
2.3 Modelo de Implementación.....	26
2.3.1 Subsistema de Presentación y Gestión de Contenidos.....	26
2.3.2 Diagrama de componentes.....	28
<input type="checkbox"/> Módulo Tipología_AM.....	30
<input type="checkbox"/> Módulo Archivo Multimedia.....	31
<input type="checkbox"/> Módulo Reproductor AM.....	32
<input type="checkbox"/> Módulo Archivo Multimedia_r.....	32
<input type="checkbox"/> Módulo Archivo Multimedia_lista.....	33
2.4 Estándar de codificación.....	33
Conclusiones.....	38
Capítulo 3.....	40
Validación de la solución Propuesta.....	40
Introducción.....	40
3.1 Elementos fundamentales del proceso de prueba.....	40
3.2 Niveles de prueba.....	40
3.3 Tipos de prueba.....	42
3.4 Pruebas Funcionales.....	45
3.5 Automatización de las pruebas.....	45
Selenium IDE.....	46
3.6 Diseño de pruebas.....	48

ÍNDICE

Caso de prueba #1	49
Caso de prueba #2	50
Caso de prueba #3	51
Caso de prueba #4	52
Caso de prueba #5	53
Conclusiones	54
Conclusiones Generales	55
Recomendaciones	56
Bibliografía Citada	57
Bibliografía Consultada	59
Glosario de términos	61

INTRODUCCIÓN

Internet, conocida como la red de redes, se ha convertido en el medio de comunicación por excelencia en la actualidad. Gracias a ella las personas en todo el mundo mantienen una estrecha comunicación y disfrutan de múltiples servicios. La aparición de la Internet comercial trajo consigo una gran ola de digitalización, dentro de la que se vieron inmersos también los audiovisuales y su transmisión. Con el incremento del ancho de banda de las redes de telecomunicación aumentaron las prestaciones para brindar transmisión de audio y video por la web, para hacer más eficiente esta transmisión surgió la tecnología streaming, la cual revolucionó este servicio.

La tecnología de streaming se utiliza para aligerar la descarga y ejecución de audio y video en la web, ya que permite escuchar y visualizar los archivos mientras se están descargando. Si no se utiliza streaming para mostrar un contenido multimedia en la Red es preciso descargar primero el archivo entero al ordenador y más tarde ejecutarlo, para finalmente ver y oír lo que el archivo contenía. Sin embargo, el streaming permite que esta tarea se realice de una manera más rápida y que se pueda ver y escuchar su contenido durante la descarga (1). El uso del streaming permite además, un mejor aprovechamiento del ancho de banda de la red, por lo que se ha difundido rápidamente en la web y goza de gran aceptación entre los usuarios. Por ello varios sitios e instituciones como VideoWeb Perú¹, YouTube² y Tv.com³ brindan el servicio de transmisión de audio y video por la web mediante tecnología streaming.

En Cuba, la Universidad de las Ciencias Informáticas es una de las instalaciones que ha realizado aportes en la aplicación de esta tecnología, con el portal Inter-nos⁴. Esta solución fue realizada en el año 2003 y empezó su funcionamiento en el curso 2003-2004. El portal integra servicios de reproducción de audio y video en demanda, así como emisoras de radio y canales de televisión en vivo. Su principal función es la de contribuir al proceso docente educativo de la Universidad, publicando teleclases y materiales políticos y educativos que sirvan de soporte a la docencia. Promueve además la

¹ <http://www.videowebperu.com>

² <http://www.youtube.com/>

³ <http://www.tv.com/>

⁴ <http://inter-nos.uci.cu/>

INTRODUCCIÓN

cultura y el esparcimiento publicando películas, series, documentales, programas de radio, festivales de artistas aficionados, emisoras de radio y canales de televisión en vivo. Esta solución presenta como principales inconvenientes que dificultan su uso y comercialización que fue desarrollada completamente usando tecnologías propietarias como lenguaje de programación ASP, como gestor de base de datos Microsoft SQL Server 2000, como servidor streaming Windows Media Server y como servidor web Internet Information Services. Los procesos de publicación y administración de materiales y noticias resultan un poco engorrosos y son realizados por personal de soporte técnico calificado ya que la plataforma no cuenta con un sistema de gestión de contenidos que facilite estos procesos, los cuales en algunos casos se deben realizar de forma manual, copiando los archivos de video a publicar para una carpeta destinada para este fin en el servidor e interactuando directamente con el código de las páginas en las que se desea mostrar alguna información.

Debido al auge que ha tenido actualmente el uso de la tecnología streaming para la transmisión de archivos multimedia muchas empresas e instituciones se han interesado en emplearla para transmitir información dentro de sus redes internas. Por todas las oportunidades de mercado que ofrece tener una aplicación que brinde este tipo de servicios y por las deficiencias que presenta Internos, el proyecto VideoWeb tiene la tarea de realizar una aplicación que erradique todas esas deficiencias y así crear un producto que brinde independencia tecnológica y que pueda comercializarse posteriormente.

La Plataforma VideoWeb cuenta con dos subsistemas, el de Administración y el de Presentación y Gestión de Contenidos, este último todavía no se encuentra desarrollado debido a ello el presente trabajo está encaminado a resolver el **problema científico** de La Necesidad del subsistema Gestión y Presentación de Contenido de la Plataforma VideoWeb para la transmisión de audio y video mediante tecnología streaming utilizando software libre en el Departamento de Señales Digitales.

Para solucionar dicho problema se tiene como **objetivo general** de la investigación: Automatizar el subsistema Gestión y Presentación de Contenido de la Plataforma VideoWeb, por lo que el **objeto de estudio** serán los Procesos relacionados con la Gestión y Presentación de Contenidos de la Plataforma VideoWeb y el **campo de acción** estará dirigido a la Automatización de los procesos presentes en el subsistema Gestión y Presentación de Contenido de la Plataforma VideoWeb. Por tanto se tiene como **idea a defender** que “Con la implementación del subsistema Gestión y Presentación de Contenido y el

INTRODUCCIÓN

establecimiento de un estándar de codificación adecuado, se logrará el correcto funcionamiento de la Plataforma VideoWeb”.

Para darle cumplimiento al objetivo general se trazaron las siguientes tareas de investigación:

Tareas de la investigación:

- Caracterizar los procesos relacionados con la Gestión y Presentación de Contenido de la Plataforma VideoWeb.
- Caracterizar las herramientas y tecnologías de desarrollo establecidas en la arquitectura del producto.
- Determinar los estándares de codificación a utilizar en la implementación del subsistema.
- Construir el modelo de implementación.
- Implementar el 50% de los casos de uso críticos del sistema.
- Desarrollar los casos de prueba que certifiquen la veracidad de los algoritmos empleados.
- Documentar los resultados de las pruebas funcionales de caja negra del subsistema.

Con el correcto cumplimiento de las tareas se espera obtener los siguientes resultados o aportes:

- Documento de estándares de codificación.
- Modelo de Implementación.
- Implementación del 50% de los casos de uso críticos del sistema.

Para obtener los conocimientos necesarios que hagan posible el cumplimiento del objetivo trazado en el trabajo y cumplir las tareas propuestas, se lleva a cabo una investigación en la que se utilizan los siguientes métodos científicos:

INTRODUCCIÓN

Métodos Teóricos:

Analítico-Sintético: Mediante este método se analizaron los componentes de los procesos de gestión y presentación de contenidos audiovisuales, así como los documentos y artefactos que genera el desarrollador dentro del equipo de desarrollo para luego sintetizar todas estas partes y ver su funcionamiento como un todo haciendo énfasis en sus relaciones esenciales. Además se empleó este método para analizar los principales componentes y funcionalidades de soluciones similares que sirven de antecedentes a este trabajo.

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.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

Introducción

La implementación del subsistema de Gestión y Presentación de Contenidos de la Plataforma VideoWeb tiene como principales basamentos el análisis y diseño realizado en el proyecto VideoWeb, así como la arquitectura de software definida en el mismo.

En el presente capítulo se abordan una serie de conceptos asociados al dominio del problema que faciliten un mejor entendimiento del mismo, se describen los procesos asociados a la gestión y presentación de contenidos en la plataforma VideoWeb y de la tecnología streaming utilizada para brindar este servicio. Se abordan además, las principales tecnologías, herramientas y lenguaje de programación seleccionados.

1.1 Procesos de Gestión y Presentación de contenidos en la Plataforma VideoWeb.

1.1.1 Descripción de los procesos de Gestión de Contenidos en la Plataforma VideoWeb.

Los contenidos que se gestionan en la Plataforma VideoWeb son archivos multimedia y artículos de contenido (noticias y avisos). Los archivos multimedia pueden ser de: audio o video. La clasificación de los archivos multimedia de video estará definida de acuerdo a las necesidades del cliente. Los artículos de contenido pueden ser creados, revisados, publicados y eliminados. Los archivos multimedia pueden ser adicionados, publicados, des-publicados y eliminados.

➤ **Caracterización de los procesos asociados a la gestión de artículos de contenido.**

Crear artículos de contenido: Este proceso consiste en escribir la información asociada al artículo y darle el formato⁵ adecuado para su publicación.

⁵ Entiéndase por formato el tipo de fuente, tamaño, ortografía.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

Revisar artículos de contenido: Este proceso consiste en revisar el/los artículos de contenidos que se hayan creado para comprobar si tiene el formato adecuado y si el contenido está correcto.

Publicar artículos de contenido: Este proceso consiste en publicar los artículos de contenido que ya hayan sido revisados y estén listos para que los usuarios puedan visualizarlos.

Eliminar artículos de contenido: Este proceso consiste en eliminar los artículos de contenido cuyo tiempo de publicación haya expirado o que no se deseen tener disponibles para los usuarios.

➤ **Caracterización de los procesos asociados a la gestión de archivos multimedia.**

Adicionar archivo multimedia: Este proceso consiste en subir el archivo multimedia para el servidor de almacenamiento y llenar la información de los metadatos⁶ asociados al archivo para almacenarlos en la base de datos.

Modificar datos del archivo multimedia: Este proceso consiste en modificar la información de los metadatos asociados al archivo multimedia y guardar los cambios en la base de datos.

Eliminar archivo multimedia: Este proceso consiste en eliminar el archivo multimedia del servidor de almacenamiento de las medias, así como los datos asociados de la base de datos.

Crear publicación de archivo multimedia: Este proceso consiste en llenar los datos asociados a la publicación del archivo para ser almacenados en el servidor de base de datos. Los datos asociados a la publicación del archivo multimedia pueden ser modificados y eliminados.

Modificar datos asociados a la publicación de archivo multimedia: Este proceso consiste en modificar los datos asociados a la publicación del archivo multimedia y guardar la actualización en la base de datos.

Eliminar publicación de archivo multimedia: Este proceso consiste en eliminar la publicación de la base de datos. Para ello hay que tener en cuenta que si el archivo multimedia se encontraba publicado hay que eliminarlo, además, del servidor de streaming.

⁶ La primera acepción que se le dio (y actualmente la más extendida) fue la de dato sobre el dato, ya que proporcionan la información mínima necesaria para identificar un recurso. Puede incluir información descriptiva sobre el contexto, calidad y condición o características del dato (19).

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

Publicar archivo multimedia: Este proceso puede realizarse de dos formas: una para la *publicación de archivo multimedia en vivo* y otra para la *publicación de archivo multimedia bajo demanda*.

Publicación de archivo multimedia en Vivo: Este proceso consiste en capturar con el MP4Live la señal de audio o video, los datos de la captura se guardan en un fichero de configuración con extensión .sdp⁷, que es copiado posteriormente para el servidor de streaming.

Publicación de archivo multimedia bajo demanda: Este proceso consiste en subir el archivo multimedia almacenado en el servidor de medias para el servidor de streaming y cambiar el estado de la publicación a publicada.

1.1.2 Descripción de los procesos asociados a la Presentación de Contenidos en la Plataforma VideoWeb.

➤ **Caracterización de la Presentación de los artículos de contenido.**

Para la presentación de artículos de contenido en la Plataforma VideoWeb, se ubicará en la parte superior izquierda del cuerpo del contenido la imagen (si la lleva), en la parte superior estará el título y debajo el cuerpo, en la parte inferior derecha aparecerá la sección a la que pertenece el artículo.

➤ **Descripción de los procesos asociados a la Presentación de los archivos multimedia.**

La presentación de los archivos multimedia estará definida de acuerdo a la tipología⁸ a la que pertenezca. Los datos asociados a cada tipología serán definidos por el cliente.

1.2 Tecnología para el Desarrollo de Aplicaciones Web.

Hoy en día existen múltiples formas de desarrollar una aplicación web. Se pueden desarrollar desde cero, implementando todos los componentes sin reutilizar código existente, siguiendo estándares de desarrollo o libremente. Esta vía de solución trae consigo una gran demora y resulta compleja, sobre todo a la hora de desarrollar grandes sistemas, aunque posibilita adaptar las funcionalidades según las preferencias del equipo de trabajo. Para facilitar el desarrollo de estas aplicaciones, hacerlo más

⁷ “Session Description Protocol” (Protocolo de descripción de sesión).

⁸ Define el tipo o clasificación de los archivos multimedia (entiéndase serie, película, documental, etc.)

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

eficiente y cómodo, surgieron herramientas como los Frameworks y los CMSs. Estas herramientas brindan una base de partida, debido a que por lo general tienen una estructura sólida, que agiliza el proceso de construcción, lo que posibilita desarrollar aplicaciones complejas en un espacio de tiempo relativamente menor al que se emplea cuando se construyen desde cero. En la mayoría de las aplicaciones web que se necesitan es común encontrarse servicios y funcionalidades como el registro de usuarios, sesiones para brindar información o la manipulación de roles para controlar el acceso, las cuales ya están implementadas o resulta más fácil su implementación en Frameworks o CMSs.

1.2.1 CMS Drupal.

CMS acrónimo de Content Management System, conocido como sistema de gestión o administración de contenidos. En esencia los CMS se encargan de facilitar el trabajo de gestión y presentación de la información. Pueden brindar una serie de herramientas para interactuar de manera simple con la información o una amplia gama de funcionalidades y herramientas para posibilitar una avanzada manipulación de los datos. Los CMS se caracterizan por optimizar el proceso de publicación de contenido, haciéndolo de forma sencilla y rápida (2).

Características del Drupal.

El CMS Drupal presenta varias características que lo hicieron el más factible para la implementación de la Plataforma VideoWeb, ellas son: está desarrollado usando el lenguaje de programación PHP, lenguaje con el que el equipo de desarrollo está familiarizado. Esto favorece y agiliza el desarrollo y adaptación de componentes para el funcionamiento de la Plataforma VideoWeb. Soporta el PostgreSQL como gestor de base datos y el Apache como servidor web, además está realizado sobre la licencia GNU/GPL, todo muy conveniente para los intereses del proyecto. Unido a esto cuenta con gestión de privilegios granulares⁹, lo que posibilita una mayor especificación del acceso a los recursos. En cuanto al soporte existen comunidades internacionales que lo brindan y ponen a la disposición de todos numerosos módulos libres que sirven para ampliar las funcionalidades del CMS. Módulos como Views, Panels y CCK (Content Construction Kit) hacen del Drupal una herramienta potente y flexible. Posee un módulo de administración robusto que posibilita configurar y adaptar el sistema de forma detallada.

⁹ El sistema permite gestión de privilegios página por página o elemento por elemento y separar los privilegios para otras funcionalidades

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

1.3 Lenguajes de programación

Un **lenguaje de programación** es aquel elemento dentro de la informática que nos permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes (3).

1.3.1 Lenguajes de programación y tecnologías del lado del cliente.

Los lenguajes de programación del lado del cliente son aquellos que son independientes del servidor y que son interpretados directamente por los navegadores web. Los ordenadores de los clientes deben tener instalados las versiones de los navegadores y los pluggins necesarios para que puedan visualizar correctamente la página.

➤ HTML

HTML (Hyper Text Markup Language), es decir, lenguaje de definición por marcas para hipertexto, es un lenguaje muy sencillo que permite definir documentos hipertexto a base de ciertas etiquetas que marcan partes del documento dándoles una estructura o jerarquía, y que permite presentar 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 (4).

➤ JavaScript

Es un lenguaje de programación que aporta características dinámicas (datos variables en función del tiempo, interactividad con el usuario, personalización, etc.) a las páginas Web, escritas en lenguaje HTML. Fue creado para ampliar las funcionalidades del lenguaje HTML, su aplicación evita recargar el servidor con peticiones que se pueden resolver localmente.

➤ AJAX

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

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 (5).

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” (6).

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 (XSLTransformations).
- ✓ Recuperación de datos asíncrona mediante XMLHttpRequest (Extensible Markup Language /Hypertext Transfer Protocol).
- ✓ JavaScript para enlazarlo todo.

Esta tecnología es soportada por varios navegadores web como Internet Explorer 6+, FireFox 1.5+ (PC, Mac), Safari 2+ y Opera 9+ (PC, Mac). Su empleo en la Plataforma VideoWeb permitirá una mayor interactividad, usabilidad y una mayor estética visual.

1.3.2 Lenguajes de programación del lado del servidor.

Los lenguajes de programación del lado del servidor son aquellos que se ejecutan en el servidor web para la construcción de las páginas web antes de ser enviadas al navegador. Los códigos puestos en el servidor construyen las páginas atendiendo a solicitudes hechas por el cliente, pueden acceder a bases de datos, recursos en la red y ficheros en el servidor. Estos lenguajes son imperceptibles por el cliente,

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

cuando se realiza la petición al servidor este convierte todo el código a lenguaje html para que el navegador lo interprete, debido a ello estos lenguajes son mucho menos rígidos respecto al cambio de un navegador a otro o respecto a las versiones del mismo.

PHP es un lenguaje interpretado de propósito general, ampliamente usado y que está diseñado especialmente para el desarrollo web. Entre sus principales características se destaca que es libre, multiplataforma y cuenta con muchas funcionalidades de manera nativa. Tiene comportamiento modular, brinda la posibilidad de adicionar nuevas funcionalidades a través de nuevos módulos.

Permite la conexión a varios sistemas de bases de datos brindando múltiples funcionalidades, destacando en este aspecto la conectividad con PostgreSQL y MySQL, que son gestores muy utilizados en el desarrollo de aplicaciones web. Cuenta con amplia documentación, tanto en su página oficial como en distintos foros y publicaciones. Permite el empleo de Programación Orientada a Objeto, no requiere que se le especifique el tipo de datos de las variables y maneja excepciones a partir de la versión 5.0.

1.4 Entorno Integrado de desarrollo.

Un Entorno Integrado de Desarrollo o IDE (acrónimo en inglés de Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación. Pueden ser desarrollados para soportar uno o varios lenguajes de programación, fueron desarrollados para proveer marcos de desarrollo amigables para los lenguajes de programación.

1.4.1 Eclipse PDT

Para el desarrollo se escogió el Eclipse con el módulo PHP Development Tools (PDT) 2.0, debido a que es un IDE de código abierto y multiplataforma que permite el desarrollo en varios lenguajes de programación. Además cuenta con una comunidad de desarrollo que provee frecuentemente nuevas actualizaciones y mejoras al existente.

Eclipse cuenta con las siguientes características:

- ✓ Editor de texto.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

- ✓ Resaltador de sintaxis.
- ✓ Compilación en tiempo real.
- ✓ Control de versiones vinculado a CVS¹⁰.
- ✓ Pruebas unitarias con JUnit.
- ✓ Asistentes (wizards) para creación de proyectos, clases, pruebas, etc.
- ✓ Refactorización.

El PDT 2.0 es una herramienta de código abierto para el desarrollo en PHP que se ha hecho muy popular entre los desarrolladores de PHP ya que ofrece las funcionalidades básicas para la edición de código que ellos necesitan. Esta versión se enfoca en el apoyo a la programación orientada a objetos para PHP.

Entre las características de la versión 2.0 se encuentra:

- ✓ Rápida y fácil navegación por la jerárquica de objetos PHP.
- ✓ Método de navegación que permite una fácil búsqueda de código PHP.
- ✓ Editor sensible al contexto, provee resaltado de código, asistente de código y autocompletado de código.
- ✓ Soporte para el debug de código PHP.

1.5 Sistema Gestor de Base 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 (7).

En la actualidad existen un gran número de SGBD, todos tienen sus características propias y especificidades, sin embargo todos deben tener en cuenta los siguientes aspectos: abstracción de la información, la independencia de los datos, redundancia mínima, consistencia en los datos, seguridad,

¹⁰ Aplicación informática que implementa un sistema de control de versiones

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

integridad, respaldo y recuperación, tiempo de respuesta y control de concurrencia para que resulten sistemas robustos y confiables. Entre los SGBD bajo licencia libre se encuentran PostgreSQL y MySQL.

1.5.1 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 (8).

Entre sus características fundamentales se encuentran:

- ✓ Base de Datos Objeto relacional: Aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas.
- ✓ Alta concurrencia: Usando MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite acceso de lectura y escritura en una tabla de forma concurrente sin necesidad de bloqueos. Siempre se obtiene la última versión de datos que se ha actualizado.
- ✓ Integridad Referencial para garantizar la validez de los datos de la Base de Datos.
- ✓ Write Ahead Logging (WAL): Consiste en registrar los cambios antes de que estos sean escritos en la base de datos. Esto garantiza que en caso de que la BD se caiga, existirá un registro de las transacciones a partir del cual se puede restaurar la BD desde el punto en que se quedó.
- ✓ Lenguajes Procedurales: Tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Además tiene habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.
- ✓ Gran variedad de tipos de datos nativos: fecha, monetarios, elementos gráficos, MAC, IP, cadenas de bits, figuras geométricas, etc.
- ✓ Soporte SQL Comprensivo: Soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

Por todas las características anteriormente planteadas y por su alta velocidad de respuesta en bases de datos de gran tamaño, hacen a este SGBD robusto y estable. Por ello se escogió para su aplicación en la Plataforma VideoWeb, lo que permitirá tener un sistema de un alto nivel de escalabilidad.

1.6 Patrón Arquitectónico Modelo Vista Controlador.

El patrón Modelo Vista Controlador (MVC) separa los datos de una aplicación, la interfaz de usuario y la lógica del negocio en tres componentes distintos. Este patrón cumple con el patrón de diseño de alta cohesión. Cada una de las partes son independientes, la comunicación entre ellas es mediante interfaces, que abstraen sus estructuras internas. Esto permite desarrollar el modelo, la vista y la controladora de forma independiente, así como realizar modificaciones en sus partes, sin afectar a las demás.

En la figura 1 se muestra el funcionamiento de un sistema implementado usando el patrón arquitectónico MVC:

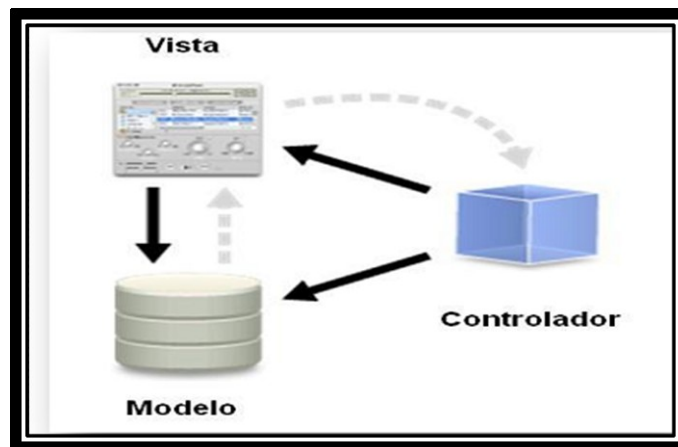


Figura 1. Patrón Arquitectónico Modelo - Vista – Controlador.

➤ Modelo.

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

➤ Vista.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

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.

Este patrón contribuye con el trabajo en equipo, pues posibilita la especialización del personal en un componente específico, siendo solo necesario conocer las interfaces de acceso disponibles de los demás componentes. También facilita la reutilización de las clases y el mantenimiento de los componentes.

1.7 Streaming de audio y video.

El streaming de archivos multimedia es una tecnología para la transmisión de contenidos audiovisuales desde un servidor hasta uno o varios ordenadores clientes a través de la red, empleando protocolos de transmisión optimizados para este fin que posibilitan un uso eficiente del ancho de banda disponible.

El streaming funciona de la siguiente manera. Primero el ordenador cliente se conecta con el servidor y éste comienza a enviarle el fichero. El cliente comienza a recibir el fichero y construye un buffer donde comienza a guardar la información. Cuando se ha llenado el buffer con una pequeña parte del archivo, el cliente lo empieza a mostrar y a la vez continúa con la descarga. El sistema está sincronizado para que el archivo se pueda ver mientras se descarga, de modo que cuando el archivo acaba de descargarse el fichero también ha acabado de visualizarse. Si en algún momento la conexión sufre descensos de velocidad se utiliza la información que hay en el buffer, de modo que se puede aguantar un poco ese descenso. Si la comunicación se corta demasiado tiempo, el buffer se vacía y la ejecución del archivo se cortaría también hasta que se restaurase la señal (1).

1.7.1 Servidores Streaming.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

Los servidores de streaming son los encargados de publicar los archivos multimedia y de atender las solicitudes realizadas por los usuarios. Pueden brindar dos tipos de servicios:

Bajo demanda: Este servicio permite al usuario hacer peticiones personalizadas al servidor, el cual levanta un hilo por cada usuario, lo que posibilita que el cliente tenga el control total sobre el flujo que se tiene asignado, permitiéndole adelantar, rebobinar, detener y reanudar la reproducción del archivo.

En vivo: El servidor transmite un flujo continuo de datos, al cual se conectan los usuarios. Esto posibilita que el servidor no tenga que levantar un flujo para cada usuario. En este servicio los usuarios no pueden modificar la velocidad de reproducción del archivo, rebobinarlo ni pararlo.

Actualmente existen varios servidores streaming bajo licencias libres que pueden usarse en la Plataforma, pero tienen algunas deficiencias que hacen que no suplan todas las necesidades de la misma. Entre los principales servidores streaming están: FFmpeg y FFServer, Darwin Streaming Server (DSS), VideoLan (VLC), Feng, MPEG4IP y MEC4.

➤ **Darwing Streaming Server**

Como servidor streaming para la Plataforma VideoWeb se seleccionó el Darwing Streaming Server (DSS) ya que presenta una serie de características que lo convierten en la mejor solución.

El DSS 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 (9).

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

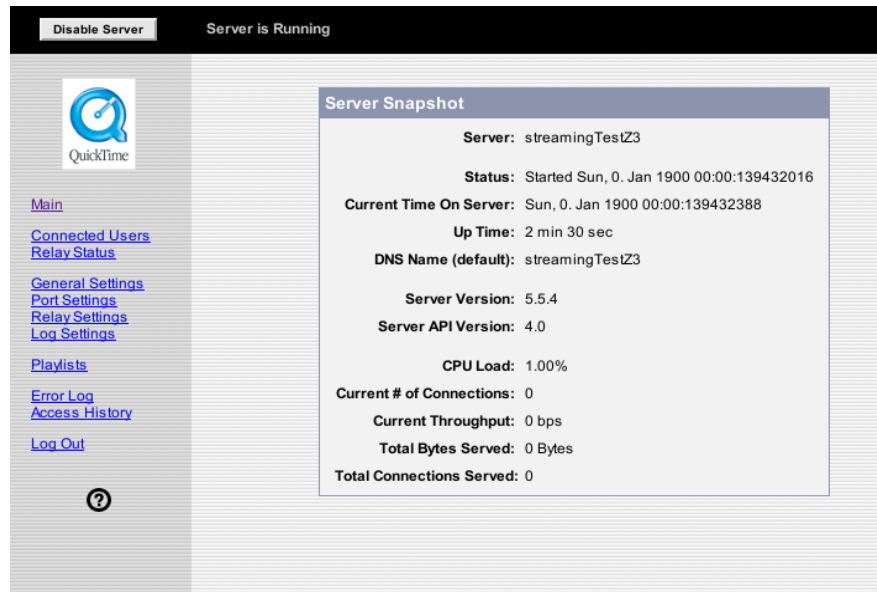


Figura 2. Interfaz Darwin Streaming Server

Permite un alto nivel de personalización y funciona en una gran variedad de plataformas como Windows, Linux y Solaris, permitiendo manipular el código fuente para satisfacer las necesidades de los desarrolladores. Es soportado por la comunidad del software libre y no es elegible para soporte técnico por Apple. Esta compañía mantiene un número de listas de discusión para usuarios y desarrolladores de Darwin Streaming Server para compartir ideas y discutir escenarios de despliegue. El DSS está disponible para descarga gratuita bajo la Apple Public Source License v2 (9). 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. Además, cuenta con una interfaz web que permite visualizar su estado y además posibilita su configuración y administración.

Aunque es extraordinario como servidor streaming de audio y video bajo demanda, 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 seleccionó 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.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

➤ Mp4Live

El Mp4Live es una herramienta disponible solamente para Linux que permite la captura y codificación de audio y video en tiempo real. Entre sus principales características están:

- ✓ Códec de video: mpeg4 (xvid, xvid-0.9.2 o xvid-1.0), H.261, mpeg2, h263, h264
- ✓ Códec de audio: AAC, MP3, Mpeg2, AMR NB and WB, G.711 alaw y ulaw, Mpeg1
- ✓ Filtros de video: desentrelazado.
- ✓ Interfaces de entrada de video: V4L (Video for Linux), V4L2.
- ✓ Interfaces de salida de audio: OSS (Open Sound System).
- ✓ Opciones de grabación: H.263, H.264, Mpeg4 video, Mpeg2 video, Mp3 audio, AAC audio, AMR audio, raw audio (PCM) y video (YUV).
- ✓ Opciones de transmisión: (RTP multidifusión o unidifusión) mpeg4 video, mpeg2 video, h261 video, h263 video, h264 video, G.711 audio, aac audio, mp3 audio, amr audio.

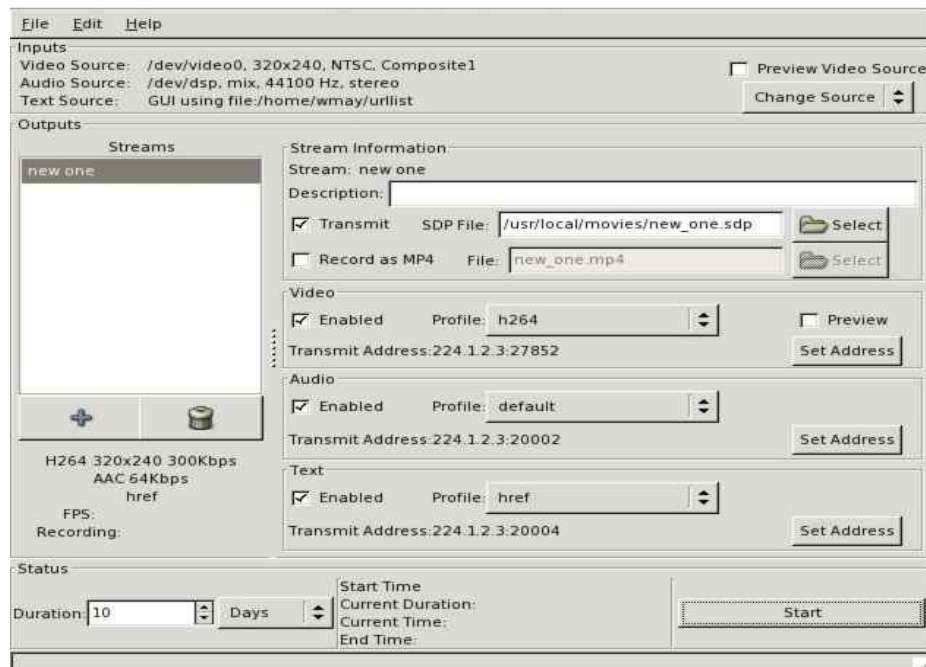


Figura 3. Interfaz Mp4Live

1.8 Metodología de desarrollo.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

Una metodología de desarrollo de software es un conjunto de pasos o procedimientos que guían el proceso de construcción de un software, permitiendo tener un producto de calidad y en el tiempo establecido. Entre las metodologías de desarrollo de software más difundidas en la actualidad podemos citar: Rational Unified Process (RUP), Extreme Programming (XP) y Microsoft Solution Framework (MSF).

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

1.8.1 Rational Unified Process (RUP).

El Proceso Unificado de Desarrollo es un proceso de desarrollo configurable que se adapta a través de los proyectos variados en tamaños y complejidad. 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. Los aspectos que definen el Proceso Unificado se resumen en tres frases claves: *dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental.*

Dirigido por casos debido a que estos representan los requisitos funcionales de la aplicación, guían el diseño, la implementación y prueba, es decir, guían el proceso de desarrollo. Ligado a los casos de usos está la arquitectura, que tiene influencia sobre ellos. La arquitectura debe posibilitar el desarrollo de todos los casos de usos y estos a su vez tienen que ajustarse a la arquitectura, por lo que la arquitectura y los casos de usos deben desarrollarse de forma paralela. El Proceso Unificado es iterativo incremental pues propone dividir el desarrollo del producto en varias iteraciones, en cada una de ellas se obtiene una serie de artefactos que servirán como base de la próxima. Así, en cada iteración se incrementa o modifican los artefactos, posibilitando entre otras cosas la identificación de riesgos, la corrección de errores, mejoras de las funcionalidades y refinar en sentido general el producto, hasta obtener el resultado esperado.

Por todo lo anteriormente expuesto se decidió utilizar RUP, además por ser una metodología que permite el modelado orientado a objeto y sin dudas por la gran cantidad de documentación y experiencia en el trabajo con ella que presenta el equipo de desarrollo lo que hace más factible su utilización.

1.9 Lenguaje de Modelado. UML

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 que esté basado en el paradigma de Programación Orientado a Objetos (10). Ofrece, además, un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

1.10 Herramienta Case. Visual Paradigm

Se seleccionó Visual Paradigm for UML 6.0 Enterprise Edition como herramienta para modelar todos los artefactos que se obtendrán a partir del análisis del negocio y el sistema por ser una herramienta multiplataforma, soporta el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue. Posibilita el modelado de base de datos, requerimientos, proceso de negocio, permite realizar todo tipo de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Las funcionalidades que brinda son numerosas como:

- ✓ Permite la captura, especificación y análisis de requisitos mediante la creación de Modelos de Análisis Textual.
- ✓ A partir de las descripciones contenidas en los análisis textuales se pueden identificar los elementos candidatos a incluir en los diagramas de casos de uso.
- ✓ Una vez identificados estos elementos candidatos se pueden utilizar para conformar los diagramas de casos de uso.
- ✓ Es una herramienta colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto.
- ✓ Permite control de versiones.
- ✓ Genera la documentación del proyecto automáticamente en varios formatos como Web o PDF.

Esta herramienta es de gran utilidad para el proyecto, por su nivel de integración con el entorno de desarrollo Eclipse PDT y por la posibilidad de trabajar ambos sobre la plataforma libre GNU/Linux, sistema operativo sobre el cual se desarrolla el proyecto.

1.11 Servidor Web. Apache

Como servidor web se propone Apache 2.2, servidor HTTP libre, de código abierto muy potente y usado en el mundo. Trabaja en múltiples plataformas como Unix (BSD, GNU/Linux.), Windows y Macintosh. Debido a su popularidad es fácil encontrar soporte ante cualquier duda o problema, lo que ha contribuido al mejoramiento de su seguridad y robustez.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

Tiene una arquitectura modular que permite adicionar o quitar de forma sencilla funcionalidades, permitiendo incrementar las prestaciones del servidor para numerosos usos. Se caracteriza por su gran escalabilidad, seguridad y rendimiento. Trabaja con varios lenguajes como Perl, Python y PHP lo que permite desarrollar aplicaciones web de gran calidad.

1.12 Servidor FTP. VSFTPD

Para realizar la manipulación de los archivos multimedia (copiarlos al servidor de media y al servidor de streaming) se decidió utilizar un servidor FTP, lo que permite agilizar el proceso y cargar con responsabilidades adicionales a los demás servidores. Para esta función se escogió el servidor VSFTPD (Very Secure FTP Daemons) que está liberado con licencia GPL para sistemas UNIX, incluyendo GNU/Linux, es seguro, estable, muy rápido y usado en muchos sitios los cuales dan buenas referencias de su comportamiento.

Conclusiones.

Con la descripción de los procesos asociados al subsistema Gestión y Presentación de Contenidos de la Plataforma VideoWeb se pudieron comprender para luego realizar una correcta implementación de los mismos. Caracterizar las herramientas, metodologías, tecnologías y lenguajes de programación que se utilizarán para la construcción de la Plataforma VideoWeb permitió realizar una adecuada selección de las mismas de acuerdo a los requerimientos establecidos para el sistema.

CAPÍTULO 2.

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

Introducción

En este capítulo se presenta una breve descripción del subsistema a implementar y los módulos que lo componen, se hace referencia, además, al modelo de implementación, estructurado a partir de los diagramas de componentes. Todo lo anteriormente planteado se definió a partir del diseño realizado a la Plataforma VideoWeb. También se muestra el estándar de codificación que se seleccionó para la implementación del producto y cuáles son las características y tareas que debe realizar un implementador de un producto de software.

2.1 El Implementador del Sistema.

El implementador de un producto de software debe poseer determinadas habilidades y realizar una serie de actividades que contribuyan a la realización del software como el desarrollo de los componentes de software y pruebas para verificar el correcto funcionamiento del mismo. La metodología seleccionada para el desarrollo del producto define cuales son las actividades y artefactos que genera el rol de implementador, ellas son:



Figura 4. Actividades y Artefactos del implementador.

De todas estas actividades y artefactos, para el desarrollo de la Plataforma VideoWeb, se definió que sólo se realizarán las actividades y artefactos siguientes:

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

Actividades:

- Analizar el comportamiento en tiempo de ejecución
- Ejecutar pruebas de desarrollador
- Implementar elementos de diseño
- Implementar la prueba de desarrollador
- Revisar el código

Artefactos:

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

Las habilidades y conocimientos apropiados para el implementador incluyen (11):

- Conocimiento del sistema o aplicación que se somete a prueba.
- Familiaridad con las herramientas de prueba y de automatización de prueba.
- Habilidades de programación.

2.2 Descripción del Subsistema Presentación y Gestión de Contenidos.

El subsistema Presentación y Gestión de Contenidos es el encargado de gestionar la presentación de los materiales a los usuarios a través de un reproductor embebido en el navegador o mediante un reproductor externo que tenga el usuario que soporte el protocolo RTSP¹¹, además se encargará de insertar nuevos archivos multimedia con todos sus datos asociados, modificarlos y eliminarlos; a partir de los cuales se podrán crear publicaciones, las que finalmente podrán ser visualizadas por los usuarios. Las funcionalidades de este subsistema se obtuvieron a partir del diseño del producto realizado en la fase de Elaboración.

¹¹ Real Time Streaming Protocol.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

Módulo FTPHTML.

Este módulo gestiona carpetas y ficheros de distintos servidores ftp. Sirve de interfaz a otros módulos para crear y eliminar carpetas, así como copiar y eliminar archivos multimedia de los servidores ftp y los distintos puntos de almacenamiento y de publicación en dichos servidores. Esto permite que la manipulación de los archivos se realice de forma rápida y segura. ([Ver Anexo 24](#))

Módulo Tipología_AM.

Permite la gestión de las clasificaciones de los materiales que se emplearán en la plataforma y posibilita definir de forma dinámica las tipologías asociadas a los materiales. ([Ver Anexo 1](#) y [Anexo 2.](#))

Módulo Archivo_Multimedia.

Interactúa con los módulos Tipología_AM y FTPHTML para gestionar la información de los archivos multimedia de acuerdo a la tipología a la que esté asociado y almacenarlo en el punto de almacenamiento que se seleccione. Permite además, la gestión de las publicaciones a partir de archivos previamente almacenados. (Ver [Anexos 3-17](#))

Módulo Reproductor_AM.

Se encarga de reproducir los archivos multimedia publicados así como las listas de reproducción creadas por los usuarios mediante un reproductor embebido en el navegador o mediante un reproductor externo que soporte el protocolo RTSP. ([Ver Anexos 18 - 19](#))

Módulo Lista_Reproducción.

Este módulo depende del módulo Archivo_Multimedia y se encarga de gestionar las listas de reproducción creadas por los usuarios. ([Ver Anexos 20-21](#))

Módulo Archivo_Multimedia_r.

Este módulo se encarga de gestionar la publicación de las señales en vivo o externas, tales como canales de tv, videoconferencias que se estén filmando en el momento o algún material que se esté capturando mediante un dispositivo de captura externo ([Ver Anexos 22-23](#)).

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

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 (12).

Este modelo muestra la distribución e interacción entre los subsistemas de la Plataforma VideoWeb teniendo en cuenta la arquitectura y el diseño del producto definidos en fases anteriores.

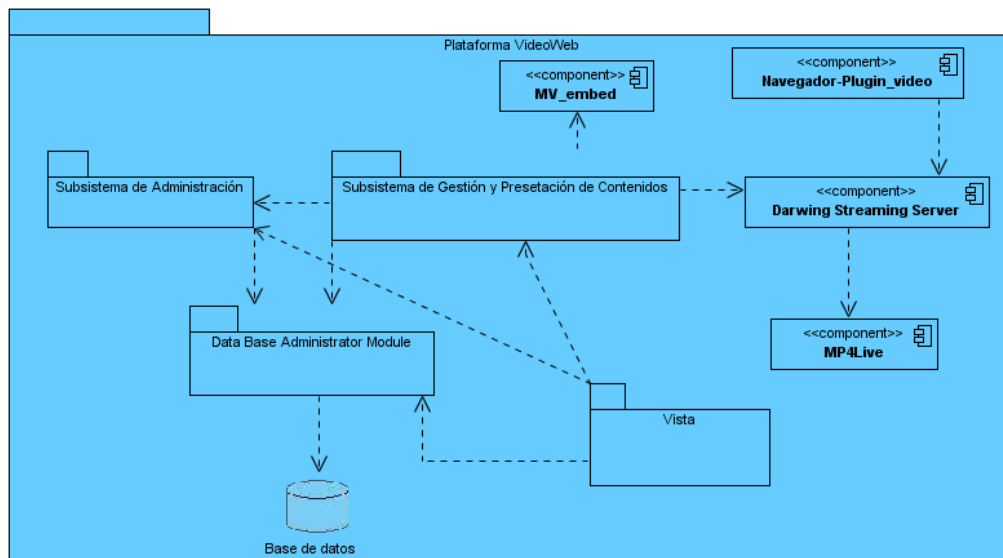


Figura 5. Modelo de Implementación

2.3.1 Subsistema de Presentación y Gestión de Contenidos.

Un subsistema de implementación es una recopilación de elementos de implementación y otros subsistemas de implementación; se utiliza para estructurar el modelo de implementación dividiéndolo en

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

componentes más pequeños (13). A continuación se muestra la distribución por paquetes del Subsistema de Gestión y Presentación de Contenidos de la Plataforma VideoWeb.

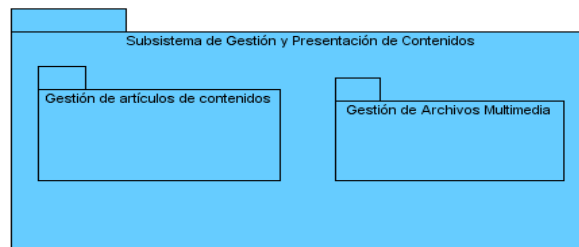


Figura 6. Subsistema de Gestión y Presentación de Contenidos

Para un mayor entendimiento de la composición del subsistema, se separaron por paquetes los módulos destinados a la gestión de artículos de contenidos y los destinados a la gestión de archivos multimedia. Los artículos de contenidos serán gestionados por el CMS, por lo tanto los módulos que se desarrollarán serán los relacionados con la gestión de los archivos multimedia. A continuación se muestra la distribución e interacción entre los módulos dentro del paquete de Gestión de Archivos Multimedia.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

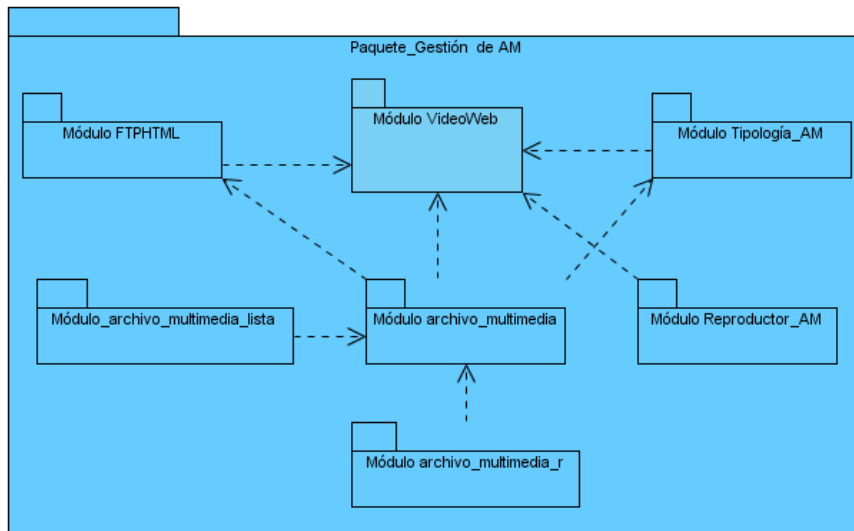


Figura 7. Paquete Gestión de Archivos Multimedia.

2.3.2 Diagrama de componentes.

El modelo de implementación se estructura a partir de los diagramas de componentes, por tal razón debe entender el funcionamiento interno de los módulos para luego asociarlos como un todo dentro de los subsistemas que lo 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 de software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, bibliotecas cargadas dinámicamente, entre otros. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente.” (14)

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

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

<<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.

➤ Módulo FTPHTML.

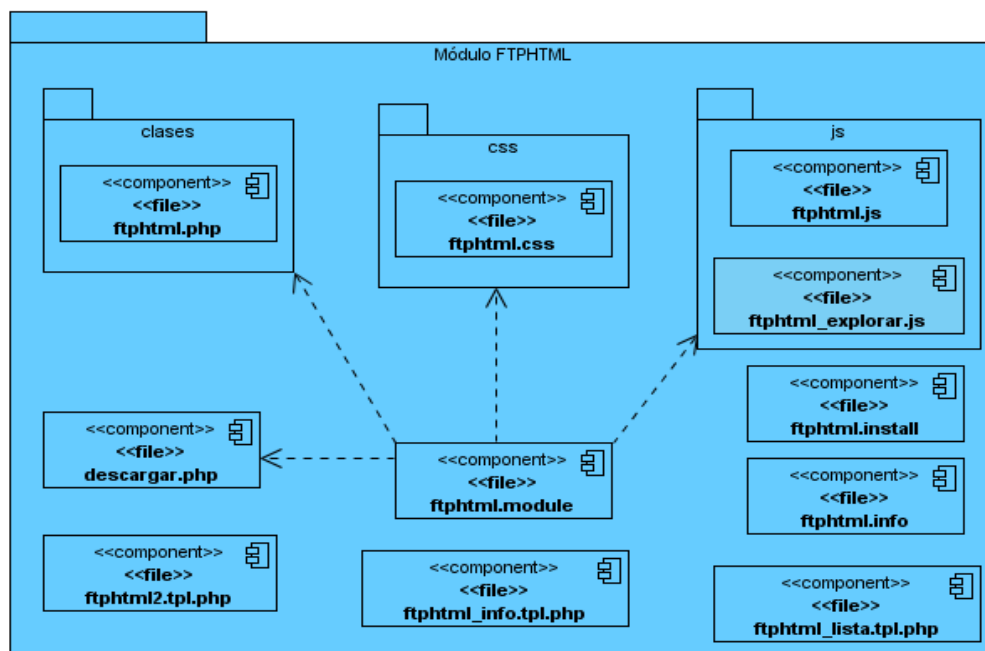


Figura 8. Diagrama de Componentes Módulo FTPHTML.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

➤ Módulo Tipología_AM

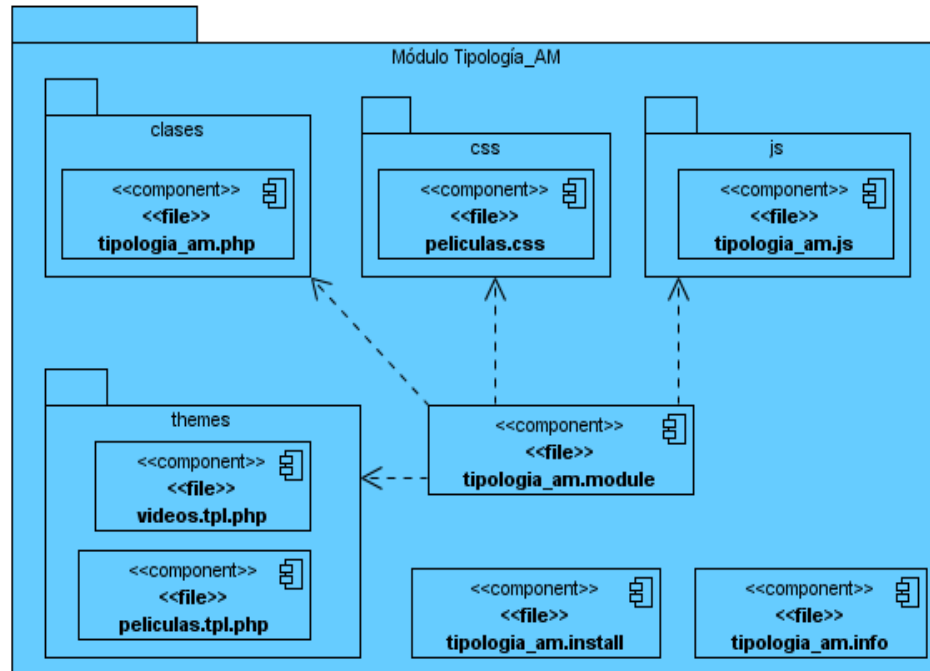


Figura 9. Diagrama de componentes Módulo Tipología_AM

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

➤ Módulo Archivo Multimedia.

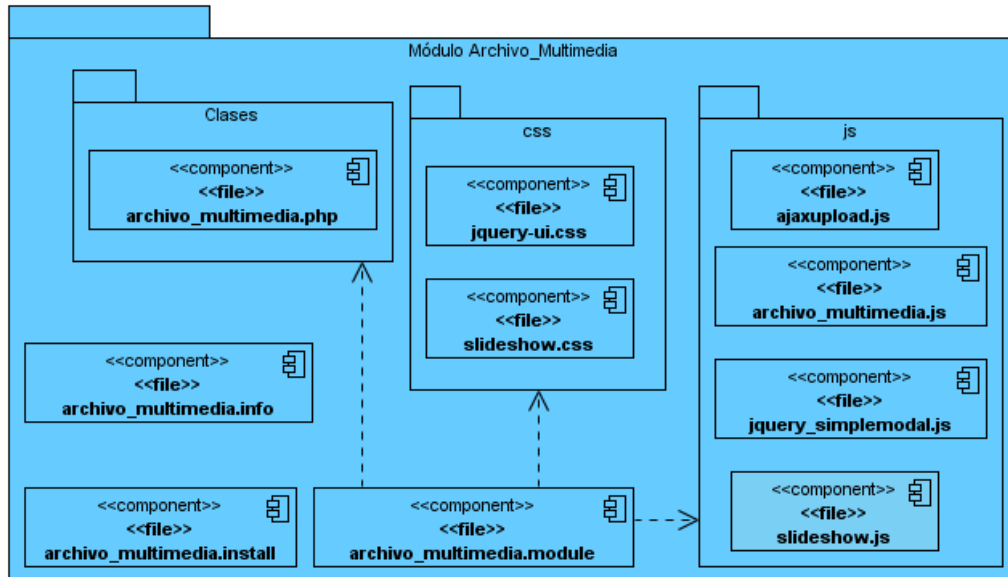


Figura 10. Diagrama de componentes Módulo Archivo Multimedia.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

➤ Módulo Reproductor AM.

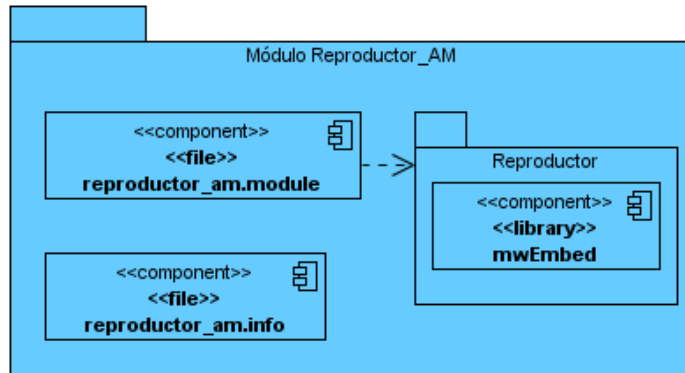


Figura 11. Diagrama de componentes Módulo Reproductor AM

➤ Módulo Archivo Multimedia_r

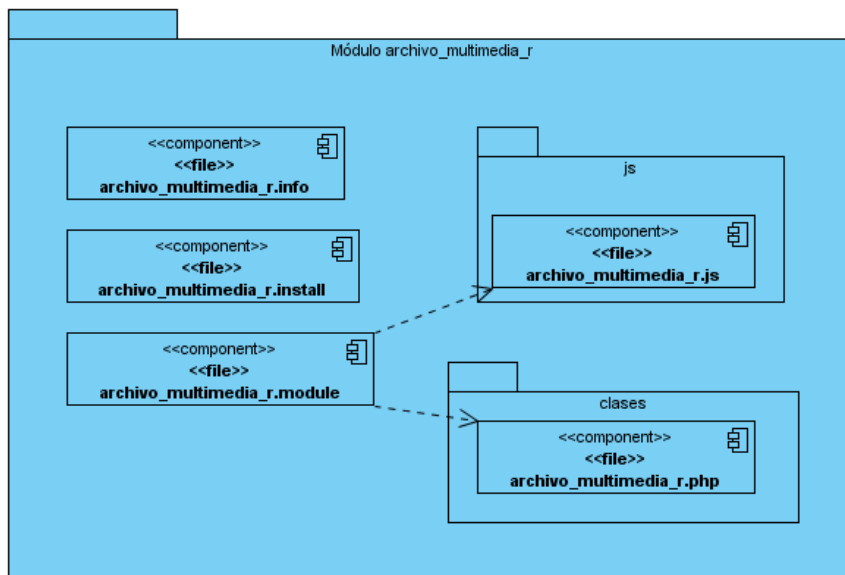


Figura 12. Diagrama de componentes Módulo Archivo Multimedia_r

➤ Módulo Archivo Multimedia_lista

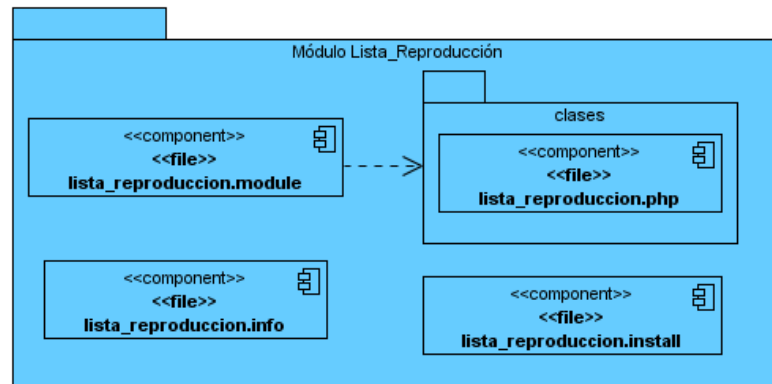


Figura 13. Diagrama de componentes Módulo Lista_Reproducción.

2.4 Estándar de codificación.

El estándar en el código de programación es muy importante en cualquier producto, principalmente si este involucra muchos desarrolladores. Esto asegura que el código sea de alta calidad, que contenga una cantidad baja de errores y sea fácil de mantener.

PRINCIPIOS GENERALES

- Los nombres de cada uno de los elementos del programa deben ser significativos; su nombre debe explicar en lo posible el uso del elemento.
- No manejar en los programas más de una instrucción por línea.
- Declarar las variables en líneas separadas.
- Añadir comentarios descriptivos junto a cada declaración de variables, si es necesario.
- La mayoría de los elementos se deben nombrar usando sustantivos.
- Los atributos deben comenzar con letra minúsculas y los métodos deben comenzar con letra mayúsculas.

Nomenclatura de los elementos.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

Elemento	Reglas de nombramiento
Clases, interfaces y archivos fuente	<p>Nombre sustantivo singular, con la primera letra en mayúscula y las demás en minúsculas.</p> <p>Si el nombre de la clase está comprimido en más de una palabra la primera letra debe de ser mayúscula. Letras mayúsculas seguidas no son permitidas.</p> <p>Ejemplo: “Zend_PDF” no es permitido mientras que “Zend_Pdf” sí lo es.</p>
Constantes	Nombre sustantivo en mayúsculas. Para separar palabras se usará el guión bajo: _
Paquetes y directorios	Nombre sustantivo singular en minúsculas

Comentarios

Cada programa deberá comenzar con un comentario que incluya:

- Objetivo, o problema que resuelve el programa.
- Algoritmo.

Cada función debe tener un encabezado que contenga:

- Objetivo de la función y no descripción del procedimiento.
- Comentarios de apoyo a variables, llamadas a función o inclusión de archivos que no sean obvios al proceso.
- Explicación de uso de argumentos (parámetros) no obvios.
- Explicación de uso de valores devueltos (de retorno).

Se utilizará dos tipos de comentarios.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

/* Se puede utilizar en el caso de un atributo, de un método o algún segmento de código */

Otros Comentarios Internos

//

//.....

Nombres de identificadores.

Se considera como identificador a los nombres de variables (arreglos, matrices, apuntadores), funciones, así como cualquier tipo de dato definido por el usuario (estructura, clase). Dichos identificadores deberán seguir las siguientes normas, además de las definidas por el propio lenguaje.

- Deberán tener un nombre significativo para que por su simple lectura, pueda conocerse su función, sin tener que consultar manuales o hacer demasiados comentarios.
- Para nombres que se usen con frecuencia o para términos largos, se recomienda usar abreviaturas estándar para que éstos tengan una longitud razonable. Si usa abreviaturas deben manejar la misma lógica en todo el programa.
- Evitar identificadores que comiencen con uno o dos caracteres de subrayado para evitar que se confundan con los que el compilador selecciona.

Identificadores de variables

Comenzarán siempre con la primera letra minúscula.

Para distinguir palabras dentro del nombre deberá emplearse un guión bajo (_).

Ejemplo:

temperatura_de_vapor

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

Indentación

- Define que debe usarse los sangrados para indentar el código y nunca espacios.

Saltos de Línea

- Añadir un salto de línea después del cierre de los paréntesis de los parámetros.
- Añadir un salto de línea después 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 de alta complejidad.

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.

String Literales

- Cuando se le asigna un texto literal (sin contenido de variables) se utilizarán comillas simples.

```
$a = 'Texto de ejemplo';
```

String Literales con apóstrofes

- Cuando se le asigna un texto literal con apóstrofes se utilizarán comillas dobles, este caso es principalmente en las sentencias SQL.

```
$sql = "SELECT `id`, `name` from `some_people` WHERE `name`='Fred' OR `name`='Susan';
```

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

Concatenación

- Para concatenar Strings se utilizará el operador "." (punto), con un espacio entre medio para mejorar la lectura.

```
$company = 'Zend' . ' ' . 'Technologies';
```

Arrays

En aquellos arrays de índices numéricos estos deberán ser números positivos, en la declaración de los valores del array se dejará un espacio en blanco luego de la coma para mejorar la lectura.

Clases

- En la declaración de la clase el nombre deberá seguir la convención ya expresada anteriormente, debe estar precedida por una documentación, luego la declaración y desarrollo.

```
/**  
 * Documentación del bloque aquí.  
 */  
class SampleClass {  
 // Todo el contenido de la clase aquí  
 // Identada por cuatro espacios vacíos  
 }
```

Control de flujo

- En las declaraciones if/then/else deberá tener un espacio antes y después del paréntesis condicional, en la misma línea se abre llave y se cierra en una línea diferente, lo mismo se aplica al elseif, a continuación un ejemplo que lo ilustra:

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

```
if (condition1 || condition2) {  
    action1;  
}  
elseif (condition3 && condition4) {  
    action2;  
}  
else {  
    defaultaction;  
}
```

- En el caso de un **Switch** la convención es la misma, adicionando que el break tiene una indentación mayor que el resto a continuación un ejemplo:

```
• switch (condition) {  
    case 1:  
        action1;  
        break;  
  
    case 2:  
        action2;  
        break;  
  
    default:  
        defaultaction;  
}
```

La opción default jamás deberá ser omitida en un Switch.

Conclusiones.

En el desarrollo del presente capítulo se describió el Subsistema de Gestión y Presentación de Contenidos y los módulos que los conforman con los casos de uso asociados a cada uno. Se presentó el modelo de implementación de la Plataforma VideoWeb, el empaquetamiento de los módulos y se mostró el funcionamiento interno de cada uno de ellos a través los Diagramas de Componentes. Además fue analizado y detallado el estándar de codificación utilizado en la implementación de los

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

módulos. Todo ello permitirá un mejor entendimiento entre los miembros del equipo de desarrollo y así lograr un producto de alta calidad.

CAPÍTULO 3.

VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

Introducción.

La prueba del software es un elemento crítico para garantizar la calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación de los elementos que componen la aplicación. Existen diversos tipos de pruebas, cada cual con sus objetivos y estrategia bien definidos, encaminadas a comprobar la calidad y los posibles errores que presenta el software. La creación de monótonos y extensos casos de pruebas ha sido reemplazada al transcurrir el tiempo por procesos automatizados que han ayudado a obtener productos de mayor calidad, a reducir costos y a ahorrar tiempo. En el presente capítulo se describirán las pruebas funcionales, así como las características de la herramienta utilizada para la automatización de las mismas.

3.1 Elementos fundamentales del proceso de prueba.

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

3.2 Niveles de prueba.

Prueba de desarrollador: La prueba de desarrollador indica los aspectos de diseño e implementación de las pruebas más adecuadas que debe llevar a cabo el equipo de desarrolladores, a diferencia de la prueba independiente. En la mayoría de los casos, la ejecución de la prueba se produce inicialmente con el grupo de pruebas de desarrollador que la diseñó e implementó; aunque es recomendable que los desarrolladores creen las pruebas de forma que estén disponibles para que las ejecuten grupos de pruebas independientes (15).

CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

Prueba independiente: Las pruebas independientes indican el diseño y la implementación de la prueba realizada más adecuadamente por alguien ajeno al equipo de desarrolladores. Puede considerarse esta distinción un súper conjunto, que incluye validación y verificación independientes. En la mayoría de los casos, la ejecución de la prueba se produce inicialmente con el grupo de pruebas independientes que la diseñó e implementó; aunque los verificadores independientes deberían crear sus pruebas de forma que estén disponibles para que las ejecuten los grupos de pruebas de desarrollador (15).

Prueba de unidad: La prueba de unidad se centra en la verificación de los elementos más pequeños del software que se puedan probar. Normalmente, las pruebas de unidad se aplican a componentes representados en el modelo de implementación para verificar que se cubren los flujos de control y los flujos de datos y que funcionan como se esperaba. Los detalles de la prueba de unidad se describen en la disciplina de implementación (15).

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

En algunos casos, los desarrolladores presuponen que otros grupos, como los verificadores independientes, son los encargados de realizar las pruebas de integración. Esta situación supone un riesgo para el proyecto de software y, en última instancia, para la calidad del software, puesto que:

- Las áreas de integración son un punto común de fallo del software.
- Las pruebas de integración realizadas por verificadores independientes suelen utilizar técnicas de caja negra y tratar componentes de software más grandes (15).

Prueba del sistema: Normalmente, la prueba del sistema se realiza cuando el software funciona en su totalidad. Un ciclo vital repetitivo permite que las pruebas del sistema se realicen mucho antes, en cuanto se hayan implementado subconjuntos bien formados del comportamiento de guiones de uso. Normalmente, el destino son los elementos en funcionamiento de extremo a extremo del sistema (15).

CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

Prueba de aceptación: La prueba de aceptación del usuario es la última acción de prueba antes de desplegar el software. El objetivo de la prueba de aceptación es comprobar si el software está preparado y lo pueden utilizar los usuarios para realizar las funciones y tareas para las que se diseñó (15).

Luego de haber analizado todos los niveles de prueba por los que puede pasar un software, destacando los principales objetivos que tienen en cada uno de estos, es necesario conocer cuáles son los tipos de pruebas que se aplican en cada uno de estos niveles.

3.3 Tipos de prueba.

Tipos de Pruebas que se aplican al software (16)

Dimensión de calidad	Tipo de pruebas
Funcionalidad	<i>Prueba de función:</i> pruebas que se centran en la validación de funciones del destino de la prueba, proporcionan los guiones de uso, los métodos y los servicios necesarios. Esta prueba se implementa y se ejecuta en diferentes destinos de la prueba, incluidas las unidades, las unidades integradas, las aplicaciones y los sistemas.
	<i>Prueba de seguridad:</i> pruebas que se centran en garantizar que los datos del destino de la prueba (o sistemas) sólo son accesibles para los actores a los que se dirigen. Esta prueba se implementa y ejecuta en varios destinos de la prueba.
	<i>Prueba de volumen:</i> pruebas que se centran en la verificación de la capacidad del destino de la prueba para manejar grandes cantidades de datos, ya sean de entrada y salida o residentes, en la base de datos. La prueba de volumen incluye estrategias de prueba como la creación de consultas que devolverán el contenido completo de la base de datos, o que tendrán tantas restricciones que no devolverán ningún dato, o en las que la entrada de datos tiene la cantidad

CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

	máxima de datos para cada campo.
Fiabilidad	<i>Prueba de integridad:</i> pruebas que se centran en la evaluación de la fuerza del destino de la prueba (resistencia a los errores) y la conformidad técnica del lenguaje, la sintaxis y la utilización de recursos. Esta prueba se implementa y se ejecuta en diferentes destinos de la prueba, incluidas las unidades y las unidades integradas.
	<i>Prueba de estructura:</i> pruebas que se centran en la evaluación de la adherencia del destino de la prueba a su diseño y formación. Normalmente, esta prueba se realiza en aplicaciones habilitadas para web y garantiza que todos los enlaces están conectados, se muestra el contenido adecuado y no hay ningún contenido huérfano.
	<i>Prueba de tensión:</i> se trata de un tipo de prueba de fiabilidad que se centra en la evaluación de cómo responde el sistema en circunstancias anormales. Las tensiones del sistema pueden ser cargas de trabajo extremas, memoria insuficiente, servicios y hardware no disponible o recursos compartidos limitados. Estas pruebas suelen realizarse para saber mejor cómo y en qué áreas fallará el sistema, de forma que se puedan planificar y presupuestar los planes de contingencia y el mantenimiento de las actualizaciones con bastante antelación.
	<i>Prueba de puntos de referencia:</i> se trata de un tipo de prueba de rendimiento que compara el rendimiento de un destino de la prueba nuevo o desconocido con una referencia conocida, carga de trabajo y sistema.
	<i>Prueba de contienda:</i> pruebas que se centran en la validación de la capacidad del destino de la prueba para manejar de forma aceptable

CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

	varias demandas del actor en el mismo recurso (registros de datos, memoria, etc.).
Rendimiento	<i>Prueba de carga:</i> se trata de un tipo de prueba de rendimiento que se utiliza para validar y evaluar la aceptabilidad de los límites operativos de un sistema bajo cargas de trabajo variables, mientras el sistema que se está probando permanece igual. En algunas variantes, la carga de trabajo permanece igual y se modifica la configuración del sistema que se está probando. Las medidas suelen tomarse en función del rendimiento de la carga de trabajo y el tiempo de respuesta de las transacciones en línea. Las variaciones de la carga de trabajo suelen incluir la emulación del pico y el promedio de cargas de trabajo que se producen dentro de la tolerancia operativa normal.
	<i>Perfil de rendimiento:</i> se trata de una prueba en la que se controla el perfil de tiempo del destino de la prueba, incluidos el flujo de la ejecución, el acceso de datos, las llamadas del sistema y de funciones para identificar y tratar los cuellos de botella de rendimiento y los procesos ineficaces.
	<i>Prueba de configuración:</i> pruebas que se centran en garantizar que las funciones del destino de la prueba son las adecuadas en diferentes configuraciones de hardware y software. Esta prueba también se puede implementar como una prueba de rendimiento del sistema.
Capacidad de soporte	<i>Prueba de instalación:</i> pruebas que se centran en garantizar que el destino de la prueba se instala correctamente en diferentes configuraciones de hardware y software, y en condiciones diferentes (como, por ejemplo, espacio de disco insuficiente o interrupciones de la alimentación). Esta prueba se implementa y ejecuta en aplicaciones

CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

y sistemas.

Tabla 1. Tipos de Pruebas que se aplican al software

Después de conocer cuáles son los niveles y tipos de pruebas que se le pueden realizar a un producto de software, ya se puede seleccionar cual es el tipo de prueba que se desarrollará para comprobar la calidad de la Plataforma VideoWeb. En el siguiente epígrafe se abordan las características e importancia de realizar pruebas funcionales, específicamente las de caja negra.

3.4 Pruebas Funcionales.

Se denominan pruebas funcionales o Functional Testing, a las pruebas de software que tienen por objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados, es común que este tipo de pruebas sean desarrolladas por analistas de pruebas con apoyo de algunos usuarios finales, esta etapa suele ser la última etapa de pruebas y al dar conformidad sobre esta el paso siguiente es el pase a producción (17).

A este tipo de pruebas se les denomina también pruebas de comportamiento o pruebas de caja negra, ya que los probadores o analistas de pruebas, no enfocan su atención a como se generan las respuestas del sistema, básicamente el enfoque de este tipo de prueba se basa en el análisis de los datos de entrada y en los de salida, esto generalmente se define en los casos de prueba preparados antes del inicio de las pruebas (17).

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

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructura de datos.
- Errores de rendimiento.
- Errores de inicialización y terminación.

3.5 Automatización de las pruebas.

CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

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.

Beneficios de las pruebas automatizadas.

- Rapidez: Las herramientas de pruebas automatizadas corren las pruebas significativamente más rápido que los probadores humanos.
- Fiabilidad: Las pruebas ejecutan precisamente las mismas operaciones cada vez que se ejecutan, eliminando el error humano.
- Repetición: Se puede probar cómo reacciona el software bajo repetidas ejecuciones de las mismas operaciones.
- Programable: Se pueden programar pruebas sofisticadas y complejas que muestren información oculta de la aplicación.
- Reusabilidad: Se pueden rehusar los scripts de las pruebas automatizadas, es decir las funciones creadas.

Finalmente se puede concluir que la principal razón para automatizar las pruebas es la reducción del tiempo de las mismas. Además al automatizar las actividades comunes que no requieren de inteligencia humana, los probadores y desarrolladores pueden dedicar mayor tiempo a pruebas más críticas y caminos más elaborados dejando los caminos básicos a las pruebas automatizadas. Para automatizar las pruebas al subsistema de Gestión y Presentación de Contenidos se seleccionó el Selenium IDE.

Selenium IDE

Selenium IDE es un plugin de Firefox que pertenece al juego de herramientas SeleniumHQ, permite realizar juegos de pruebas sobre aplicaciones web. Para ello realiza la grabación de la acción seleccionada (navegación por una página) en un "script", el cual se puede editar y parametrizar para adaptarse a los diferentes casos, y lo que es más importante su ejecución se puede repetir tantas veces como se quiera (18).

CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

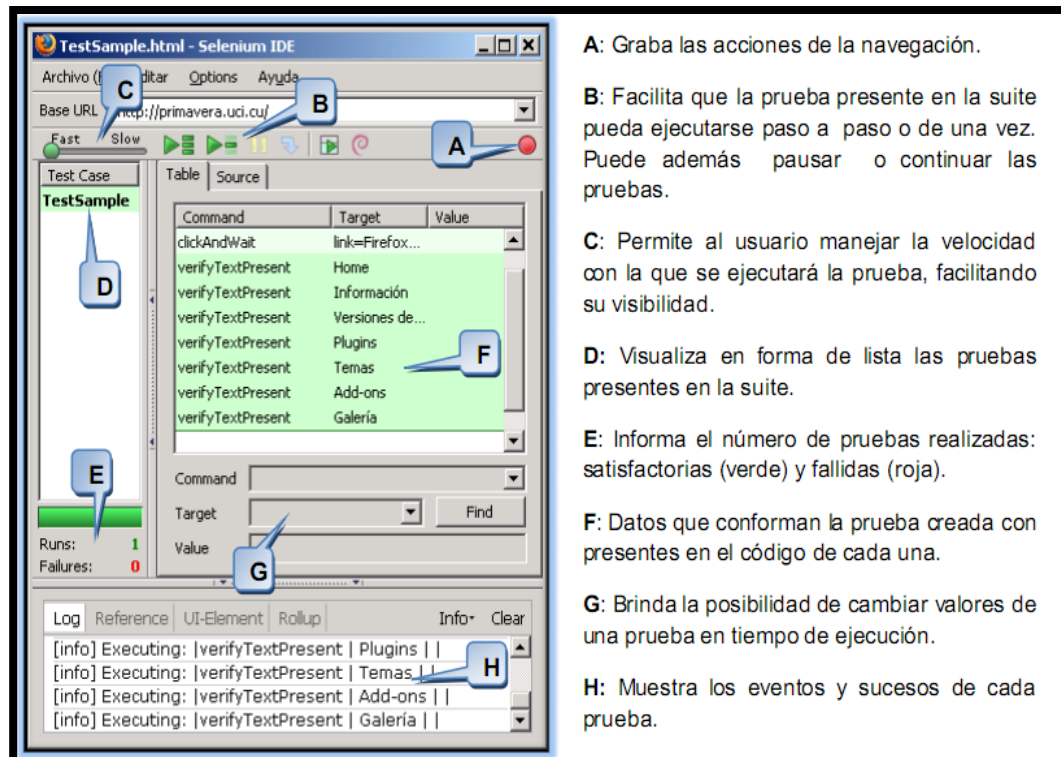
Esta herramienta permite al desarrollador web ahorrarse mucho esfuerzo (mucho esfuerzo = muchas horas) cada vez que se resuelve alguna incidencia o se genera una versión nueva. Para ello permite automatizar la realización de las pruebas ya sean o bien pruebas específicas (una acción en particular) o bien juegos de pruebas (un conjunto de acciones) (18).

Características:

- Facilidad de registro y ejecución de los test.
- Referencia a objetos DOM en base al ID, nombre o a través de XPath.
- Autocompletado para todos los comandos.
- Las acciones pueden ser ejecutadas paso a paso.
- Herramientas de depuración y puntos de ruptura (breakpoints).
- Los test pueden ser almacenados como HTML y scripts Ruby, entre otros formatos.
- Soporte para Selenium user-extensions.js.
- Ejecución en varios navegadores.
- Uso de diferentes API's en diferentes lenguajes (PHP, Ruby, JAVA, Javascript, etc).

A continuación se muestra la figura 14 que representa la interfaz del plugin Selenium IDE, así como una breve explicación de las partes que lo conforman.

CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.



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.

Figura 14. Interfaz gráfica del plugin Selenium.

3.6 Diseño de pruebas

Con el objetivo de validar el correcto funcionamiento de los módulos que componen el Subsistema de Gestión y Presentación de Contenidos se realizaron pruebas de caja negra con la ayuda del Selenium aprovechando las facilidades que ofrece. Para ello se seleccionaron algunos de los casos de uso críticos más importantes para la Plataforma.

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

Caso de prueba #1.

Nombre del Caso de Uso	Adicionar Archivo Multimedia
Precondiciones	Que el autor se haya autenticado. Que el archivo a adicionar esté en el formato de archivo requerido. Que exista un punto de almacenamiento con espacio libre disponible para contener el nuevo archivo.
Datos de entrada	Datos asociados a la tipología a la que pertenece el archivo multimedia, punto de almacenamiento del archivo multimedia.
Resultados esperados	<p>El sistema verifica el formato del archivo y los datos asociados a este, sube el archivo multimedia para el servidor de medias y crea la referencia al mismo en la base de datos. Muestra un mensaje informando al usuario que la operación se ha realizado correctamente.</p> <p>Si el archivo no se encuentra en el formato requerido, los datos introducidos son incorrectos o se dejó en blanco algún campo obligatorio el sistema muestra un mensaje de error.</p>
Resultados obtenidos	<p>La prueba es satisfactoria.</p> <p>La prueba es satisfactoria.</p>

Tabla 2. Caso de Prueba: Adicionar archivo multimedia.

Caso de prueba #2

Nombre del Caso de Uso	Crear Publicación Archivo Multimedia
Precondiciones	Que el autor se haya autenticado, que existan archivos multimedia previamente adicionados, que existan puntos de publicación de streaming disponibles.
Datos de entrada	Datos asociados a la publicación como nombre de la publicación, secciones y categorías a las cuales va a pertenecer, el punto de almacenamiento donde se va a guardar el archivo multimedia en el servidor de medias.
Resultados esperados	El sistema valida los datos entrados, en caso de ser correctos almacena los datos de la publicación en la base de datos, copia el archivo para el servidor de medias y muestra un mensaje indicando que la publicación se ha creado correctamente. Si se dejó algún campo requerido en blanco el sistema muestra un mensaje de error indicando que se deben llenar los campos obligatorios.
Resultados obtenidos	La prueba es satisfactoria. La prueba es satisfactoria.

Tabla 3. Caso de prueba: Crear publicación archivo multimedia

Pruebas realizadas con la ayuda del Selenium IDE.

Caso de prueba #3

Nombre del Caso de Uso Publicación Archivo Multimedia

Precondiciones Que el revisor se haya autenticado y que existan publicaciones de archivos multimedia cuyo estado sea no publicada, que exista espacio disponible en el servidor de streaming.

Datos de entrada Listado de archivo multimedia publicados, que exista alguno que no esté publicado y selección de uno de ellos.

Resultados esperados El sistema cambia el estado de la publicación de archivo multimedia a publicado, copia el archivo multimedia del servidor de medias para el servidor de streaming y muestra un mensaje indicando que el archivo fue publicado satisfactoriamente.

Resultados obtenidos La prueba es satisfactoria.

Publicar Archivo Multimedia	
open	/VideoWeb2/?q=media/gestionar
click	status_88
click	//button[@type='button']

Selenium TestRunner

Execute Tests

Fast Slow

Highlight elements

Elapsed: 00:01

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

Tools

Show Log

Tabla 4. Caso de prueba: Publicación archivo multimedia

Caso de prueba #4

Nombre del Caso de Uso	Dejar de Publicar Archivo Multimedia
Precondiciones	Que el revisor se haya autenticado y que existan publicaciones de archivos multimedia cuyo estado sea publicada.
Datos de entrada	Listado de archivos multimedia publicados.
Resultados esperados	El sistema cambia el estado de la publicación de archivo multimedia a no publicado, elimina el archivo multimedia asociado del servidor de streaming y muestra un mensaje indicando que la operación fue realizada correctamente.
Resultados obtenidos	La prueba es satisfactoria.

Resultado obtenido con el Selenium

The screenshot displays the Selenium TestRunner interface. On the left, a table lists the test steps:

Dejar de Publicar Archivo Multimedia	
open	/VideoWeb2/?q=media/gestionar
click	status_88
click	//button[@type='button']

On the right, the Selenium TestRunner window shows the following details:

- Execute Tests:** Includes play, pause, and refresh icons, and a speed slider from Fast to Slow.
- Highlight elements
- Elapsed: 00:01
- Tests** | **Commands**
- 1 run | 0 passed
- 0 failed | 0 failed
- 0 incomplete
- Tools: Show Log

Tabla 5.Caso de prueba: Dejar de publicar archivo multimedia

CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

Caso de prueba #5

Nombre del Caso de Uso Buscar Archivo Multimedia

Caso de Uso

Precondiciones Que el editor y el revisor se hayan autenticado.

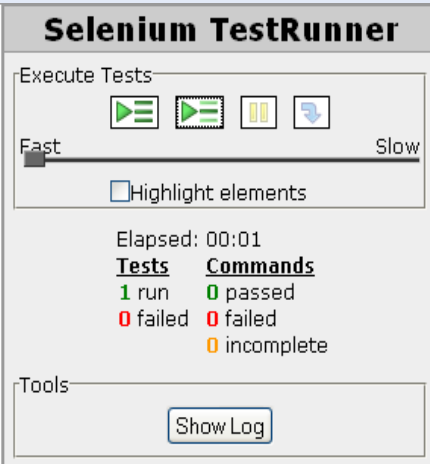
Datos de entrada Nombre del archivo que se desea buscar o la tipología a la que pertenece.

Resultados esperados El sistema busca en los metadatos de los archivos multimedia almacenados en la base datos y muestra el resultado en la lista de archivos multimedia debajo de las opciones de filtrado.

Resultados obtenidos La prueba es satisfactoria.

Resultado obtenido con el Selenium

Buscar Archivo Multimedia		
open	/VideoWeb2/?q=archivo_multimedia/gestionar	
type	buscar_key	colgando en
click	enviar_buscar	



The screenshot shows the Selenium TestRunner interface. It includes a 'Execute Tests' section with a speed slider set to 'Fast' and a 'Highlight elements' checkbox. Below this, it displays 'Elapsed: 00:01' and a summary of test results: 1 run, 0 passed, 0 failed, and 0 incomplete. A 'Show Log' button is visible in the 'Tools' section.

Tabla 6.Caso de prueba: Buscar archivo multimedia

Conclusiones.

Conocer cuáles son los tipos de pruebas que se le pueden realizar a un sistema informático permitió seleccionar la idónea para aplicarla al subsistema. El uso de la herramienta Selenium permitió ahorrar tiempo y esfuerzo, así como eliminar los errores humanos que pudieran suceder a la hora de ejecutar las pruebas. Con la realización de las pruebas se pudo validar el cumplimiento de los requisitos funcionales establecidos en la fase inicial del proceso de desarrollo del producto, constatando el correcto funcionamiento de los módulos que componen el subsistema Gestión y Presentación de Contenidos de la Plataforma VideoWeb.

CONCLUSIONES GENERALES

CONCLUSIONES GENERALES.

Con la caracterización de los procesos relacionados con la Gestión y Presentación de Contenidos de la Plataforma VideoWeb, se pudo comprender como funcionan y tener mayor claridad para implementar estas funcionalidades. Para desarrollar el sistema se caracterizaron las herramientas y tecnologías de desarrollo establecidas en la arquitectura del producto, lo que permitió sentar las bases para el posterior trabajo con las mismas. Durante el proceso de construcción del subsistema aumentó el dominio sobre el CMS Drupal, los servidores streaming que se utilizaron para la publicación de los archivos multimedia y los ftp para la manipulación y transferencia de los ficheros, también aumentaron los conocimientos sobre streaming de video por la web.

Para un mejor entendimiento entre los miembros del equipo de desarrollo se construyó el modelo de implementación y se estableció el estándar de codificación a seguir para el desarrollo del subsistema. El primero permitió modelar y comprender como interactúan los módulos y componentes del sistema; y el segundo, además de un entendimiento común, ayudó a obtener un código de alta calidad y buena legibilidad. Por último, para comprobar la calidad y correcto funcionamiento del sistema, se diseñaron y ejecutaron casos de prueba, los que arrojaron resultados satisfactorios, demostrando el cumplimiento de los requerimientos funcionales establecidos en la fase inicial del proceso de desarrollo del producto.

Finalmente todo lo anteriormente expuesto permitió culminar el subsistema para así completar las funcionalidades que debía tener la Plataforma VideoWeb y se logró de esta manera tener un producto realizado con herramientas libres que contribuirá a lograr la independencia tecnológica que tanto necesita el país.

RECOMENDACIONES

- ✚ Se recomienda que se continúe el desarrollo de la Plataforma VideoWeb y que en nuevas versiones se incorporen algunas funcionalidades como:
 - Visualizar los materiales audiovisuales que más hayan visto los usuarios.
 - Visualizar los materiales audiovisuales publicados recientemente según el criterio de un filtro de tiempo.
 - Visualizar los materiales audiovisuales de mayor puntuación y mayor aceptación por parte de los usuarios a través de las votaciones que se realizan.
 - Permitirle a un usuario notificar a otro que un nuevo contenido ha sido publicado.
 - Permitirle a un usuario compartir una lista de reproducción con otro usuario.
 - Permitirle a un usuario la posibilidad de aceptar, mezclar con una existente o rechazar una lista de reproducción que otro usuario desee compartirle.
 - Mostrarle al usuario los tipos de materiales audiovisuales que hayan sido de mayor preferencia para él.
- ✚ Realizarle otros tipos de pruebas al sistema.

BIBLIOGRAFÍA CITADA

1. **Marín Abreu, Angel Dayán.** *Desarrollo de la Plataforma VideoWeb. Rol Analista.* La Habana : s.n., 2009.
2. **Torres Rodríguez, Frank.** *Desarrollo de la Arquitectura de Software.* La Habana : s.n., 2009.
3. **definición.org.** *definición.org.* [En línea] [Citado el: 12 de Diciembre de 2009.] <http://www.definicion.org/lenguaje-de-programacion>.
4. **El lenguaje HTML.** [En línea] [Citado el: 12 de Diciembre de 2009.] http://www.xpps.net/contenido_xppsnet/areatec/HMTL.pdf.
5. **Orth, Steve.** *4DHispano.* [En línea] [Citado el: 13 de Diciembre de 2009.] <http://www.4dhispano.com/products/brochures/pdf/4dweb20pack-es.pdf>.
6. **Barral Precedo, David y García Chao, Asís.** [En línea] [Citado el: 13 de Diciembre de 2009.] http://stuff.gpul.org/2006_web/doc/2006_JWEB_03 AJAX.pdf.
7. **Gómez Ballester, Eva.** *Bases de Datos 1.* [En línea] [Citado el: 12 de Diciembre de 2009.] <http://www.alu.ua.es/j/jmr36/Conectate/Base%20Datos/Apuntes2006.pdf>.
8. **Domínguez Mora, Dunier.** *Implementación de los Subsistemas Web y Transferencia de la Plataforma de Transmisión Abierta para Radio y Televisión.* La Habana : s.n., 2009.
9. **Mac OS Forge.** *Darwin Streaming Server.* [En línea] Mac OS Forge, 16 de Mayo de 2008. [Citado el: 12 de Marzo de 2010.] <http://dss.macosforge.org/>.
10. **Alegsa.** *Diccionario Informático. Definición de UML.* [En línea] Alegsa. [Citado el: 15 de Marzo de 2010.] <http://www.alegsa.com.ar/Dic/uml.php>.
11. **Rational Unified Process.** [En línea] IBM, 2006. [Citado el: 8 de Febrero de 2010.] Ayuda.del.RUP.2007.Español/index.htm#core.base_rup/roles/rup_implementer_D145E819.html.
12. **Centro Nacional de Tecnologías de Información.** *Merinde.* [En línea] Centro Nacional de Tecnologías de Información. Venezuela, 2010. [Citado el: 25 de Febrero de 2010.] http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=96&Itemid=297.
13. **IBM Corporation.** *Rational Unified Process.* [En línea] 2007. [Citado el: 28 de Febrero de 2010.] Ayuda.del.RUP.2007.Español/core.base_rup/guidances/guidelines/important_decisions_in_implementation_BC39CD13.html.

BIBLIOGRAFÍA CITADA.

14. **Letelier Torres, Patricio.** Desarrollo de Software Orientado a Objeto usando UML. [En línea] [Citado el: 10 de Marzo de 2010.]
15. **IBM Corp.** Ayuda de Rational Unified Process 2007. *Niveles de prueba.* [En línea] 2006. [Citado el: 03 de Mayo de 2010.] Ayuda.del.RUP.2007.Español/core.base_rup/guidances/concepts/levels_of_test_8A878577.html.
16. **IBM Corporation.** Ayuda Rational Unified Process,2007. *Tipos de prueba.* [En línea] 2006. [Citado el: 3 de Mayo de 2010.] Ayuda.del.RUP.2007.Español/core.base_rup/guidances/concepts/types_of_test_8AB94831.html.
17. **Oré B., Ing. Alexander.** CalidadSoftware.com. *FUNCTIONAL TESTING - PRUEBAS FUNCIONALES.* [En línea] 2009. [Citado el: 4 de Mayo de 2010.] http://www.calidadyssoftware.com/testing/pruebas_funcionales.php.
18. **Madrid, Victor Javier.** AdictosAlTrabajo.com. *Autentia.* [En línea] 31 de Diciembre de 2008. [Citado el: 3 de Mayo de 2010.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=seleniumIDE>.
19. **Senso, Jose A. y Piñeiro, Antonio de la Rosa.** *El concepto de metadato. Algo más que descripción de recursos electrónicos.* Brasilia : s.n., 2003.

BIBLIOGRAFÍA CONSULTADA

1. **Marín Abreu, Angel Dayán.** *Desarrollo de la Plataforma VideoWeb. Rol Analista.* La Habana : s.n., 2009.
2. **Torres Rodríguez, Frank.** *Desarrollo de la Arquitectura de Software.* La Habana : s.n., 2009.
3. **definición.org.** *definición.org.* [En línea] [Citado el: 12 de Diciembre de 2009.] <http://www.definicion.org/lenguaje-de-programacion>.
4. **El lenguaje HTML.** [En línea] [Citado el: 12 de Diciembre de 2009.] http://www.xpps.net/contenido_xppsnet/areatec/HMTL.pdf.
5. **Orth, Steve.** *4DHispano.* [En línea] [Citado el: 13 de Diciembre de 2009.] <http://www.4dhispano.com/products/brochures/pdf/4dweb20pack-es.pdf>.
6. **Barral Precedo, David y García Chao, Asís.** [En línea] [Citado el: 13 de Diciembre de 2009.] http://stuff.gpul.org/2006_web/doc/2006_JWEB_03_AJAX.pdf.
7. **Gómez Ballester, Eva.** *Bases de Datos 1.* [En línea] [Citado el: 12 de Diciembre de 2009.] <http://www.alu.ua.es/j/jmr36/Conectate/Base%20Datos/Apuntes2006.pdf>.
8. **Domínguez Mora, Dunier.** *Implementación de los Subsistemas Web y Transferencia de la Plataforma de Transmisión Abierta para Radio y Televisión.* La Habana : s.n., 2009.
9. **Mac OS Forge.** *Darwin Streaming Server.* [En línea] Mac OS Forge, 16 de Mayo de 2008. [Citado el: 12 de Marzo de 2010.] <http://dss.macosforge.org/>.
10. **Alegsa.** *Diccionario Informático. Definición de UML.* [En línea] Alegsa. [Citado el: 15 de Marzo de 2010.] <http://www.alegsa.com.ar/Dic/uml.php>.
11. **Rational Unified Process.** [En línea] IBM, 2006. [Citado el: 8 de Febrero de 2010.] Ayuda.del.RUP.2007.Español/index.htm#core.base_rup/roles/rup_implementer_D145E819.html.
12. **Centro Nacional de Tecnologías de Información.** *Merinde.* [En línea] Centro Nacional de Tecnologías de Información. Venezuela, 2010. [Citado el: 25 de Febrero de 2010.] http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=96&Itemid=297.

13. **IBM Corporation.** Rational Unified Process. [En línea] 2007. [Citado el: 28 de Febrero de 2010.] Ayuda.del.RUP.2007.Español/core.base_rup/guidances/guidelines/important_decisions_in_implementation_BC39CD13.html.
14. **Letelier Torres, Patricio.** Desarrollo de Software Orientado a Objeto usando UML. [En línea] [Citado el: 10 de Marzo de 2010.]
15. **IBM Corp.** Ayuda de Rational Unified Process 2007. *Niveles de prueba.* [En línea] 2006. [Citado el: 03 de Mayo de 2010.] Ayuda.del.RUP.2007.Español/core.base_rup/guidances/concepts/levels_of_test_8A878577.html.
16. **IBM Corporation.** Ayuda Rational Unified Process,2007. *Tipos de prueba.* [En línea] 2006. [Citado el: 3 de Mayo de 2010.] Ayuda.del.RUP.2007.Español/core.base_rup/guidances/concepts/types_of_test_8AB94831.html.
17. **Oré B., Ing. Alexander.** CalidadSoftware.com. *FUNCTIONAL TESTING - PRUEBAS FUNCIONALES.* [En línea] 2009. [Citado el: 4 de Mayo de 2010.] http://www.calidadsoftware.com/testing/pruebas_funcionales.php.
18. **Madrid, Victor Javier.** AdictosAlTrabajo.com. *Autentia.* [En línea] 31 de Diciembre de 2008. [Citado el: 3 de Mayo de 2010.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=seleniumIDE>.
19. **Senso, Jose A. y Piñeiro, Antonio de la Rosa.** *El concepto de metadato. Algo más que descripción de recursos electrónicos.* Brasília : s.n., 2003.
20. Tu función. [En línea] 6 de Marzo de 2007. [Citado el: 13 de Diciembre de 2009.] <http://www.tufuncion.com/ventajas-ajax>.
21. MPEG4IP. [En línea] Diciembre de 2004. [Citado el: 13 de Diciembre de 2009.] <http://mpeg4ip.sourceforge.net/documentation/index.php?readme=mp4liveinternal>.
22. Standards, security and best practices. *Drupal.* [En línea] 2010. [Citado el: 25 de Enero de 2010.] <http://drupal.org/coding-standards>.
23. **Rojas, Johanna y Barrios, Emilio.** ESTRATEGIA DE APLICACIÓN DE LAS PRUEBAS. *Pruebas Funcionales.* [En línea] 2007. [Citado el: 2 de Abril de 2010.] <http://www.udistrital.edu.co/comunidad/grupos/arquisoft/fileadmin/Estudiantes/Pruebas/HTML%20-%20Pruebas%20de%20software/node36.html>.
24. **Lic. Carlos Galindo González, y Dr. Ramiro Pérez Vázquez.** Rational Unified Process® . *Rational Unified Process®* . 2009.

GLOSARIO DE TÉRMINOS.

Fichero .sdp: es un formato para describir parámetros de inicialización de streaming media. Ha sido publicado por el IETF como RFC 4566.

Mp4live: es la aplicación de la suite MPEG4IP que codifica y sirve la señal audiovisual al Darwin Streaming Server.

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

CVS: El **Concurrent Versions System** (CVS), o **Concurrent Versioning System**, es una aplicación informática que implementa un sistema de control de versiones: mantiene el registro de todo el trabajo y los cambios en los ficheros (código fuente principalmente) que forman un proyecto (de programa) y permite que distintos desarrolladores (potencialmente situados a gran distancia) colaboren.

Refactorización: es una técnica de la ingeniería de software para reestructurar un código fuente, alterando su estructura interna sin cambiar su comportamiento externo.

RSTP: El protocolo de flujo de datos en tiempo real (del inglés Real Time Streaming Protocol) establece y controla uno o muchos flujos sincronizados de datos, ya sean de audio o de video.